Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich–Mathematischen Gesamtfakultät
der
Ruprecht–Karls–Universität
Heidelberg

vorgelegt von
Dipl.-Inf. Stefan Weber
aus Heidelberg

Tag der mündlichen Prüfung: 20. März 2009

Discrete Tomography by Convex-Concave Regularization using Linear and Quadratic Optimization

1. Gutachter: **Prof. Dr. Christoph Schnörr**
2. Gutachter: **Prof. Dr. Joachim Hornegger**

To Pauline[1] and Attila[2].

[1] Pauline Weber (stillborn 21.12.2006).

[2] Prof. Attila Kuba (1953-2006),
Dept. of Image Processing and Computer Graphics, University of Szeged, Hungary.

Quidquid agis, prudenter agas et respice finem.[1]

[1] trans. "Whatever you do, do prudently, and be aware of the consequences."

Abstract

Tomography in its broadest sense concerns the reconstruction of cross section images which permits the visualization of the interior of objects. From the medical application, computerized tomography (CT) and magnetic resonance imaging (MRI) are well-known for delivering high quality images from inside the human body and, thus, support physicians at their diagnosis. Generally speaking, tomographic images depict the spatially varying density of objects; in the medical case for instance different sorts of tissue are distinguishable because of their diverse densities. Nevertheless, the values that density attains must not be constant, not even in the same class of tissue. Instead, it might continuously vary and therefore this type of tomography is referred to as continuous tomography.

In contrast to continuous tomography, discrete tomography concerns the reconstruction of objects that are made up from a few different materials, each of which comprising a homogeneous density distribution. Consequently, the involved densities can only embrace a certain set of discrete values. Such reconstruction scenarios arise for instance in quality protection by non-destructive testing where manufacturers seek for involvements or cracks inside their cast or metal parts. The reconstruction process assigns a single value from the discrete set to each spatial position which in the simplest case, single material and air (environment), yields a binary image with 0 corresponding to air (environment) and 1 to material. In fact the binary case is mostly studied in the relevant literature. There, it is typically also assumed that the discrete values are either known in advance and are, thus, directly accessible as a priori knowledge or that at least a reasonably good estimate can be provided. Similarly as continuous functions generalize discrete functions, discrete tomography can be perceived as a special case of continuous tomography. This, however, poses the question why to deal with a special case while there already exists a solution to the more general case? The answer is that algorithms which are specialized to the discrete problem have many advantages if both assumptions, (i) discreteness of the solution and (ii) discrete values are a priori known, hold for the underlying reconstruction problem. If they are met then specialized algorithms typically need a significant smaller amount of projection data since they explore the a priori knowledge in order to reduce the space of possible solutions. Further, due to the construction of the scanning device it might not be possible to perform a scan over a range of 180° which is, however, mandatory in order to apply the transform-based methods among the continuous reconstruction approaches. Specialized algorithms, conversely, do not underlie any restrictions on the scanning range and are, thus, applicable to limited angle tomography. Another advantage is that many discrete approaches allow to actively include additional a priori knowledge, e.g. the smoothness of the solution, into the reconstruction process which in turn makes even less projections necessary. Nonetheless, the restriction to a discrete set also entails some difficulties as it renders the reconstruction task into a combinatorial problem, in fact its $\mathbb{NP}$-completeness can be proven for more than two projections. As a result stochastic approaches, like simulated annealing, are frequently employed which can be applied straightforwardly due to their flexibility concerning objective criteria. It is known, however, that these approaches converge slowly if applied properly and their results cannot be reproduced due to their non-deterministic procedure.

This work addresses the development of new and theoretically sound algorithms which are suitable for reconstruction problems as they appear in discrete tomography. At this, we pursue a pure deterministic approach and our principal strategy is to solve a relaxation of the original problem at first. This part equals the minimization of a convex optimization problem which is good natured from a mathematical point of view and is efficiently solvable by linear or quadratic programming in our case. Subsequently, the relaxation is gradually released and a solution in the sense of the original problem is enforced. The non-linear subproblems which accrue during that process are rendered to sequences of convex optimization problems by means of an optimization approach based on d.c. ($\sim$ difference of convex functions) programming. This provably assures the convergence of our algorithms which is not obvious for this type of optimization strategy. In order to numerically demonstrate the performance of our algorithms we solve several reconstruction problems setup from different phantom images and a varying number of projections. At first, we consider reconstructions of binary images only but eventually extend our approach to the general discrete reconstruction problem such that it can be successfully applied to the non-binary case.

Zusammenfassung

Tomographie befasst sich im weitesten Sinne mit der Rekonstruktion von Querschnittsbildern aus Projektionen und ermöglicht es damit das Innere von Objekten visuell darzustellen. Aus dem medizinischen Anwendungsbereich sind die Computertomographie und die Kernspintomographie sehr bekannt, die in diesem Zusammenhang Schichtaufnahmen des menschlichen Körpers liefern und so den Arzt bei seiner Diagnose unterstützen. Im Allgemeinen gibt ein tomographisches Bild die unterschiedlichen, örtlichen Dichteverhältnisse wieder. So grenzen sich beispielsweise in einer medizinischen Aufnahme unterschiedliche Gewebsarten aufgrund ihrer verschiedenen Dichten gegeneinander ab. Der Wert der Dichte muss dabei, selbst innerhalb einer Gewebsart, nicht unbedingt gleichbleibend sein und darf sich kontinuierlich ändern. Diese Form der Tomographie wird daher auch als kontinuierliche Tomographie bezeichnet.

Im Gegensatz zur kontinuierlichen Tomographie befasst sich die diskrete Tomographie mit der Rekonstruktion von Objekten, die sich aus Materialien mit gleichbleibender (homogener) Dichteverteilung zusammensetzen. Die Dichte kann daher in der tomographischen Rekonstruktion nur bestimmte, diskrete Werte annehmen. Dies ist unter anderem interessant für die Qualitätssicherung durch zerstörungsfreies Prüfen, bei der beispielsweise Gussteile auf Einschlüsse oder Risse untersucht werden sollen. Im rekonstruierten Bild sind dann die entsprechenden Bereiche den jeweiligen Dichten zugeordnet, was im einfachsten Fall von nur einem Material und Luft (Umgebung) einem binären Bild entspricht; 0 entspricht Umgebung und 1 Material. Tatsächlich findet der binäre Fall in der einschlägigen Literatur die meiste Beachtung. Dort wird meist auch angenommen, dass die diskreten Werte entweder zuvor bekannt sind und daher als Vorwissen zur Verfügung stehen oder zumindest ein entsprechender Schätzwert bekannt ist. Ähnlich wie diskrete Funktionen eine echte Teilmenge kontinuierlicher Funktionen darstellen, kann man die diskrete Tomographie als Spezialfall der allgemeineren kontinuierlichen Tomographie verstehen. Dies berechtigt die Frage, warum man sich überhaupt mit einem Spezialfall beschäftigen soll, wo doch bereits eine Lösung für das allgemeinere Problem existiert. Die

Antwort darauf ist folgende: Algorithmen, die auf das diskrete Problem spezialisiert sind, weisen viele Vorteile auf, wenn die beiden Annahmen, (i) diskretwertige Lösung und (ii) Bekanntheit der diskreten Werte, auf das zugrundeliegende Rekonstruktionsproblem zutreffend sind. Falls dies gilt, dann kommen spezialisierte Algorithmen typischerweise mit signifikant weniger Projektionsdaten aus, da der Lösungsraum unter Ausnutzung des Vorwissens reduziert werden kann. Desweiteren ist es bei manchen Anwendungen aufgrund der Gerätekonstruktion nicht möglich Projektionen über einem Winkelbereich von 180° zu schießen, was unter den kontinuierlichen Rekonstruktionstechniken, insbesondere für die transformationsbasierten, eine notwendige Voraussetzung ist. Spezialisierte Algorithmen unterliegen diesbezüglich in den meisten Fällen keinerlei Beschränkung. Ein weiterer Vorteil der diskreten Ansätze liegt auch darin, dass weiteres Vorwissen, wie Glattheit der Lösung, aktiv in den Rekonstruktionsprozess mit eingebracht werden kann, wodurch weitere Projektionen eingespart werden können. Die Einschränkung auf eine diskrete Menge von zulässigen Dichtewerten bringt allerdings auch Schwierigkeiten mit sich, da durch sie das Rekonstruktionsproblem einen kombinatorischen Charakter erhält und man zeigen kann, dass es für mehr als zwei Projektionen $\mathbb{NP}$-vollständig ist. Aus diesem Grund kommen zur Lösung des Problems häufig stochastische Ansätze, wie beispielsweise Simulated Annealing zum Einsatz, die wegen ihrer hohen Flexibilität in Bezug auf das Optimierungskriterium direkt anwendbar sind. Falls richtig angewandt, sind diese bekanntermaßen langsam und erlauben aufgrund ihrer nicht-deterministischen Vorgehensweise keine Reproduktion der Ergebnisse.

Diese Arbeit beschäftigt sich mit der Entwicklung neuer, theoretisch fundierter Algorithmen für Rekonstruktionsprobleme wie sie typischerweise in der diskreten Tomographie auftreten. Wir verfolgen dabei einen rein deterministischen Ansatz, dessen grundlegende Strategie zunächst eine Vereinfachung (Relaxation) des ursprünglichen Problems löst. Dieser Teil entspricht der Minimierung eines konvexen Optimierungsproblems und ist daher aus mathematischer Sicht gutartig und mittels linearer oder quadratischer Programmierung effizient lösbar. Im Anschluss daran wird die Relaxation graduell aufgehoben und eine Lösung im Sinne des ursprünglichen, nicht–konvexen Problems erzwungen. Die dabei entstehenden Teilprobleme werden unter Einbeziehung eines auf d.c. ($\sim$ difference of convex functions) Programming basierenden Optimierungsansatzes jeweils in Sequenzen von konvexen Optimierungsproblemen überführt. Dadurch ist die Konvergenz unserer Algorithmen nachweislich gewährleistet, was für diese Optimierungsstrategie nicht offensichtlich ist. Um die Leistungsfähigkeit unserer Algorithmen numerisch zu demonstrieren, rekonstruieren wir verschiedene, zunächst binäre Phantombilder aus unterschiedlichen Projektionen. Anschließend erweitern wir unseren Ansatz auf allgemeine diskrete Rekonstruktionsprobleme, so dass er auch für den nicht-binären Fall angewendet werden kann.

Danksagung

Eine Promotion ist ein langer und mühseliger Weg, der bereits begonnen hat, lange bevor das erste Wort die erste Seite der Doktorarbeit ziert. Überhaupt ist eine solche Strecke, deren Bewältigung Jahre in Anspruch nimmt, nur zu schaffen, wenn man von vielen Menschen entsprechende Unterstützung erfährt. Ich möchte hier die Gelegenheit nutzen all denjenigen meinen tiefen Dank auszudrücken, die mir während meiner Zeit als Doktorand zur Seite standen und von denen einige diese Arbeit damit erst ermöglicht haben.

Zuallererst, möchte ich meine Dankbarkeit gegenüber Prof. Dr. Christoph Schnörr zum Ausdruck bringen. Bereits als Student hat er mein Interesse an der Bildverarbeitung und der Mustererkennung geweckt, wobei mich damals wie heute sein mathematischer Zugang zur Lösung derartiger Probleme in besonderem Maße beeindruckt wie anspricht. Sowohl seine Art des wissenschaftlichen Arbeitens als auch seine geradlinige Persönlichkeit haben mich stark beeinflußt und ich kann ihm nicht genug für all das danken, was ich von ihm gelernt habe.

Prof. Dr. Joachim Hornegger gilt mein Dank, daß er sich so bereitwillig als Zweitgutachter für meine Dissertation zur Verfügung gestellt und meine Arbeit seiner gründlichen Durchsicht unterzogen hat.

Meinem Zimmergenossen am Lehrstuhl und Mitstreiter auf dem Gebiet der diskreten Tomographie Thomas Schüle gilt mein Dank für die sehr angenehme Zusammenarbeit und die wertvollen, oft sehr langen fachlichen Diskussionen, die wir im Laufe der Jahre hatten und die uns so oft einen beachtlichen Schritt weitergebracht haben.

Ich bin nicht gutgläubig genug, um zu glauben, daß diese Arbeit vollständig frei von Fehlern sei, jedoch wurden große Anstrengungen unternommen diese auf ein unbedrohliches Ausmaß zu reduzieren. Hierbei wurde ich tatkräftig unterstützt von Dr. Paul Ruhnau, Dr. Karsten Mühlmann, Florian Becker und Jörg Kappes, von denen einige diese Arbeit komplett korrekturgelesen haben. Für ihre Mühe und ihren Einsatz möchte ich mich ganz herzlich bedanken.

Mit meinen Kollegen, die mich während meiner Zeit als Doktorand am Lehrstuhl begleitet haben, verbindet mich eine wunderbare, gemeinsame Zeit, für die ich sehr dankbar bin: Martin Bergtholdt, Dirk Breitenreicher, Prof. Dr. Daniel Cremers, Christian Gosch, Dr. Matthias Heiler, Yuan Jing, Rezaul Karim, Dr. Jens Keuchel, Dr. Timo Kohlberger, Jan Lellmann, Dr. Stefania Petra, Dr. Christian Schellewald, Stefan Schmidt, Annette Stahl, and Andriy Vlasenko. In diesem Zusammenhang möchte ich auch gerne Frau Rita Schieker erwähnen, die gute Seele unseres Lehrstuhls.

Meinen Eltern, Helga und Klaus Weber, möchte ich von Herzen danken, daß sie mich immer unterstützt haben und mir damit überhaupt ermöglicht haben diesen langen Weg zu gehen, so daß am Ende diese Arbeit erst entstehen konnte. Ohne sie wäre dies sicher nicht zu schaffen gewesen.

Nicht zuletzt gilt mein ganz besonderer Dank meiner Frau Carolin, die mir mit sehr viel Geduld den Rücken über diese lange Zeit freigehalten hat, damit ich weiter an dieser Arbeit schreiben konnte. Unserem Sohn Jonathan möchte ich danken, einfach weil er so ist wie er ist und er mir auf seine Weise immer wieder neue Kraft gibt.

Diese Arbeit widme ich meiner Tochter Pauline, der es traurigerweise nicht vergönnt war das Licht dieser Welt zu erblicken, und meinem Freund und Kollegen Prof. Dr. Attila Kuba, der das Gebiet der diskreten Tomographie wie kein anderer prägte und uns zu unser aller Bedauern nach langer Krankheit viel zu früh verlassen musste.

Contents

List of Algorithms

List of Figures

Notation

Throughout this work the following notation will be used, if not explicitly stated otherwise:

$\mathbb{N}$	Natural numbers, $0 \notin \mathbb{N}$.
$\mathbb{N}_0 := \mathbb{N} \cup \{0\}$	
$\mathbb{Z}$	Integers.
$\mathbb{Q}$	Rational Numbers.
$\mathbb{R}$	Real numbers.
$\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$	Extended real numbers.
$\mathbb{C}$	Complex numbers; symbol for the imaginary unit $\mathrm{i} := \sqrt{-1}$.
$A, B, C, ...$	Arbitrary sets.
x	Scalar value.
$\mathbf{x} := (x_1, ..., x_n)^\top$	Vector, typically $\mathbf{x} \in \mathbb{R}^n, n \in \mathbb{N}$.
$\mathbf{0} := (0, ..., 0)^\top =: \mathbf{0}_n$ $\mathbf{1} := (1, ..., 1)^\top =: \mathbf{1}_n$	Special vectors of size n containing only 0 or 1 entries. If size is not given explicitly it should be clear from the context.
$\mathbf{e}_i := (0, ..., 0, 1, 0, ..., 0)$	Vector with i-th component 1 and 0 otherwise.
$\langle \mathbf{a}, \mathbf{b} \rangle := \mathbf{a}^\top \mathbf{b} = \sum_{i=1}^n a_i b_i$	Scalar product of the vectors $\mathbf{a}$ and $\mathbf{b}$.
$\lVert \mathbf{x} \rVert_p := \sqrt[p]{\sum_{i=1}^n \lvert x_i \rvert^p}$	p-norm of a vector $\mathbf{x}$, $p \geq 1$, $p \in \mathbb{R}$.
$\lVert \mathbf{x} \rVert_1 = \sum_{i=1}^n \lvert x_i \rvert$	Manhattan norm.
$\lVert \mathbf{x} \rVert_2 =: \lVert \mathbf{x} \rVert$	Euclidean norm.
$\lVert \mathbf{x} \rVert_\infty := \max \{\lvert x_1 \rvert, ..., \lvert x_n \rvert\}$	infinity or maximum norm.

$\mathbf{A} := \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$	Matrix, $\mathbf{A} \in A^{m \times n},\ \ m, n \in \mathbb{N},\ \ a_{ij} \in A,\ \ \forall i, j.$
$\mathrm{diag}(\mathbf{a}) := \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix}$	Diagonal matrix.
$\mathbf{I} := \mathrm{diag}(\mathbf{1})$	Identity matrix.
$\lambda_k(\mathbf{A})$, $\lambda_{\min}(\mathbf{A})$	The k-th and the smallest eigenvalue of matrix $\mathbf{A}$.
$\mathbf{A} \succ 0$	Positive definiteness of matrix $\mathbf{A}$.
$\mathbf{A} \succeq 0$	Positive semidefiniteness of matrix $\mathbf{A}$.
$\delta(x) := \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases}$	Dirac delta function, see also appendix A.
$\mathcal{N}(i) := \{j \mid j \text{ is neighbor of } i\}$	Given a pixel i the set $\mathcal{N}(i)$ contains the indices to all its neighboring pixels j.
$\langle i, j\rangle \sim j \in \mathcal{N}(i)$	This notation is convenient when building the sum over all pairs of neighbors, e.g. $\sum_{\langle i,j\rangle} f(i,j) = \sum_i \sum_{j \in \mathcal{N}(i)} f(i,j)$.
$\mathbb{E}[X] := \sum_i p_i\, x_i$	Expectation value in case of a discrete random variable $X \in \{x_1, ...\}$ with probabilities p_i
$\mathbb{E}[X] := \int_{-\infty}^{\infty} x f(x)\, \mathrm{d}x$	and in case of a continuous random variable X with probability density function $f(x)$.

1 Introduction

Tomography literally descends from the Greek terms "tomos", transl. "slice", and "graphia", transl. "describing", and deals with the reconstruction of objects from which only some kind of projection data is known. Nowadays, tomographic imaging techniques impact diverse scientific fields and have been successfully applied to a variety of applications. For instance, computerized tomography (CT) is well-known in the context of medicine, as it provides insight into the human body without surgical intervention [63]. Tomographic techniques support archaeologists by gathering data from the inside of ancient excavations which would suffer tremendous damage otherwise due to their fragile condition, e.g. the investigation and 3D visualization of Egyptian mummies [90, 139]. A special type of tomography, called seismic tomography [72], is employed by geologists to examine the interior of the earth in search for raw materials, like oil deposits, and, as a final example, in non-destructive material testing highly reliable metal parts, like the turbine blades of an airplane, are reconstructed and scanned for cracks or involvements [6].

Projection data is often acquired by X-rays, like in computerized tomography where an X-ray emitting source and detectors are rotated around the patient. While X-rays travel through the human body they become partly absorbed depending on the density of the tissue at the current spatial position. The detectors finally measure the remaining radiation and, hence, the amount of absorbed intensity is known as the initial intensity is known as well. From this information, reconstruction algorithms retrieve an image of the spatial distribution of the density inside the object. Also based on X-rays, a slightly different type of projection is used in positron emission tomography (PET), a method which is, alongside CT, frequently applied in nuclear medicine. In contrast to computerized tomography a short-lived, radioactive tracer isotope which emits positrons is injected into the patient's blood circuit. Inside the body, the positrons annihilate with electrons, thereby, dispatching a pair of photons in opposite directions. The PET scanning device detects such pairs and reconstructs the position where the annihilation occurred. By this, physicians gather information about the metabolism inside the human brain or heart for instance. A tomographic technique, closely related to PET, is single photon emission computed tomography (SPECT) which also provides functional data of the metabolism. However, it differs as single photons and not pairs of photons are detected and used to retrieve the spatial information.

In contrast to the previous methods, magnetic resonance imaging (MRI) [117, 89] acquires projection data in completely different way, since it exploits the magnetic moment of the atomic nuclei with an odd number of protons and neutrons (non-zero spin), such as hydrogen and phosphorus. Therefore, the object is placed in a extremely strong (0.3 to 3 tesla) and uniform magnetic field which aligns the spins either parallel or anti-parallel to the magnetic field. The object is then excited by electromagnetic pulses perpendicular to the magnetic field leading some nuclei to temporarily enter a non-aligned high energy state. While realigning, the nuclei emit an electromagnetic signal which is measured by coils surrounding the object and permits the reconstruction of the nuclei's distribution.

Under mathematical considerations, tomographic reconstruction problems belong to the class of inverse problems which, informally speaking, are concerned with the determination of parameters in order to explain some observations or measurements. According to Hadamard [64] mathematical problems are termed well-posed if they fulfill the following criteria,

(i) a solution exists,

(ii) the solution is unique, and

(iii) the solution depends continuously on the data.

On the opposite, problems that do not meet these criteria are called ill-posed and their mathematical analysis is subtle [43]. Inverse problems are frequently ill-posed and arise, besides tomographic imaging, in many computer vision and image processing problems [37]. In contradiction to criterion (ii), the solution space of tomographic reconstruction problems can be huge, particularly, if only a small number of projections can be taken and/or the range from which the projections are gathered is restricted, the latter is usually termed limited angle tomography. In such cases, a large number of images can be in prefect agreement with the projection constraints while the individual images might not possess similar features and can, thus, appear quite differently. Assuming that some prior knowledge about the expected images is known in advance, a common remedy is to impose additional constraints on the space of solutions. Hence, solutions fitting that prior information are favored while others are suppressed. This process is typically referred to as regularization [128] and will play an important role concerning this work.

It is characteristical to all the tomographic techniques discussed so far that the reconstructed values may continuously range over the real numbers and, from this consideration, we might gather the problem addressed by these methods under the term continuous tomography. However, in other scenarios it might be known in advance that these values attain only a few different discrete values. For instance, if we think of material testing problems we have perfect knowledge about the object and the materials involved. Nevertheless, we do not know if there are any defects present in the object's structure and this is precisely what we are interested in. In a similar way, as discrete problems form a subclass of the more general continuous problems, one could certainly apply continuous reconstruction algorithms also to discrete problems. In this case, however, it is not guaranteed that the solution obtained by a continuous method is indeed valid within the discrete space of the problem, even if a lot of projections are taken. Even worse, for some applications the number of projections is crucial since more projections correspond to a higher X-ray exposure of the patient or the deterioration of a fragile material. Both, few projections and a limited range for projection acquisition, exclude continuous methods since they typically require a large amount of projection data and some of them even a full scan over 180°. Therefore, the demand for reconstruction methods which are specialized for the discrete case and their theoretical investigation is of broad interest and motivates the field of discrete tomography.

1.1 Thesis Statement and Contribution

This work is devoted to the development of efficient and theoretically sound reconstruction algorithms applicable for binary and, more generally, discrete tomography. This is a challenging task since methods suitable for discrete reconstruction problems and their theory are by no means settled and still reveal many interesting and surprising results.

At first, we consider relaxations of the problem based on linear programming and, therefore, avoid the problem's inherent combinatorial complexity. However, this approach requires an explicit rounding step afterwards which does not actively include the projection constraints and, thus, might lead to suboptimal results. In order to improve on this, we introduce a reconstruction framework based on convex-concave regularization and DC (difference of convex functions) programming which in turn implicitly performs the rounding. However, this leads to the minimization of a non-linear optimization problem which generally suffers from many local minima. In order to prevent our approach from getting trapped in a poor local optimum our optimization strategy is to relax the problem at first, such that it becomes convex and, thus, tractable. At this stage, the problem is readily solved to global optimality. In the subsequent stages, the relaxation is then gradually removed and, hence, the overall optimization process leads to an optimizer of the original, difficult problem. If the convex solution is in some sense close to a discrete solution, there is some hope that the final solution is also a proper solution of the original problem.

Similar strategies have been previously used in deterministic annealing [112] and by the graduated non-convexity (GNC) approach of Blake and Zisserman [15] which concerning image reconstruction has been applied in [101]. Convergence of this type of optimization is by no means obvious, it is however guaranteed for our approach due to the optimization method we adopt from DC programming [105, 106] leading, thus, to a sequence of convex subproblems. DC programming usually refers to algorithms for the global optimization of a certain class of non-convex functions [69]. Since global optimality is feasible only for small-sized problems we focus on the local optimization of large-scale problems.

Finally, we extend our reconstruction framework such that it can be successfully applied not only to binary but more generally to discrete tomography. We demonstrate that our algorithms perform competitive and yield high quality reconstructions in both binary and discrete tomography. Additionally, we apply our extended approach to image labeling and, thus, emphasize the relevance of our results to other optimization settings.

1.2 Related Work

Though discrete tomography[1] is still a young and active research field it roots back to mathematical problems concerning the determination of binary matrices from their row and column sums [115, 39, 40, 27, 28, 26, 29, 30]. In this context, the first discrete reconstruction algorithm has been proposed by Ryser who also introduced the concept of switching components [115] as elementary operations,

$$\begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & 1 \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline 0 & 1 \\ \hline 1 & 0 \\ \hline \end{array} \quad . \tag{1.1}$$

Both patterns, equation (1.1), share the same vertical (column sum) and horizontal projections (row sum) and can, thus, be interchanged if they occur in a binary matrix. He further showed that such matrices are uniquely determined by their row and column sum if no switching

[1] The name “discrete tomography” is due to Larry Shepp who organized the first meeting on the topic in 1994, see chapter 1 of [66].

components are present and any two images with the same row and column sum can be transformed into each other through a sequence of switchings. While former research was mainly motivated by mathematical interest concerning the consistency of such problems or the existence and uniqueness of a solution, this changed when attention was drawn to the reconstruction of materials at the atomic layer [125, 79], see also section 3.1.1.

Due to the combinatorial nature of the problem stochastic optimization approaches like simulated annealing (SA) [56] are frequently applied [38, 94, 68, 91, 98, 86] in order to minimize an estimation problem based on Markov-random field models.

Concerning the complexity of the problem, Gardner [52] proved that the discrete reconstruction problem becomes $\mathbb{NP}$-complete for more than two projections. Further $\mathbb{NP}$-completeness results including the cardinality of the discrete set can be found in [53, 41]. Nevertheless, the problem can be solved efficiently for two projections using a network flow formulation which is due to Gale [51], section 3.3.3. Interestingly, Gale's results lead to the same consistency condition as proved by Ryser [115]. Some approaches [116, 8, 9, 7] use this network flow approach and try to combine solutions obtained from several pairs of projections in order to have a solution for all projections.

More related to our work is the approach proposed by Censor [34], section 3.3.2, which is mounted on top of a non-binary iterative algorithm and steers the reconstruction process towards a binary solution. This steering procedure is, however, more heuristic than our approach.

For a detailed introduction to the field of discrete tomography, we kindly refer the reader to the books of Herman and Kuba [66, 67] and particularly recommend chapter 1 of [66] for a historical overview on the subject.

In chapter 6, we examine the case of image labeling for the evaluation of our multiclass approach. At this, we compare our proposed method to the α-expansion method from Boykov et al [23] which has become quite popular for the optimization of various computer vision problems, i.e. image segmentation [21] and restoration [19, 23], medical imaging [20], stereo [83] and motion [119]. Besides the α-expansion algorithm, we consider relaxations based on semidefinite programming (SDP) which have been applied to image labeling by Keuchel [76, 75]. SDP relaxations are well-known to provide reasonable approximations to combinatorial problems [137] but are inherently slow as they square the number of variables. Therefore, we also consider relaxations based on second-order cone programming (SOCP) [92] which have been proposed in [77] and [97]. In particular the latter approach has been used in connection with image labeling in [88].

1.3 Outline

Chapter 2 is devoted to continuous tomography and the reconstruction methods applied thereby. At this, we derive the Fourier slice theorem which forms the basis for all transform-based reconstruction methods. As a representative for this class, we investigate the filtered backprojection algorithm which is nowadays widely employed in real-world applications. Finally, we survey iterative reconstruction methods which in contrast to the transform-based methods pursue an algebraic reconstruction strategy.

In **chapter 3**, we introduce the field of discrete tomography restricted, however, to the case of binary tomography which is mostly addressed in the relevant literature. We overview reconstruction algorithms based on simulated annealing, binary steering, and network flow and compare them to conventional methods (chapter 2) in order to exemplificate their advantage with respect to binary tomography.

In **chapter 4**, we relax the original binary reconstruction problem in order to circumvent the problem's inherent combinatorial complexity. This gives rise to approximations by linear programming which in turn can be solved efficiently even for large problem instances. Due to the relaxation, however, an additional rounding step is necessary in order to finally obtain a discrete solution.

In **chapter 5**, we incorporate a binary concave regularizer which performs the rounding implicitly but involves the optimization of a non-convex functional. In order to proceed, we derive an optimization framework by means of a primal-dual subgradient method based on DC ($\sim$ difference of convex functions) programming. Concerning the LP-based algorithms (chapter 4), this leads to a sequence of linear programs where, however, the framework is not restricted to linear programming. In order to demonstrate the flexibility of our framework, we consider binary reconstructions under the assumption that projections are either only accessible from blurred objects or the projections themselves are blurred. Thus, the modified problem involves the estimation of a deblurring parameter within the overall reconstruction process. We show, nonetheless, that this problem can be cast into our framework and that our approach provides very promising reconstructions also for this type of problem.

Motivated by the results obtained in case of binary tomography so far, we develop in **chapter 6** a generalized approach for multiple discrete values. This extension is not straightforward and we, therefore, propose a new concave regularizer. However, we shown that our new regularizer reduces to our previous binary regularizer (chapter 5) in case of binary tomography. Hence, the multiclass approach properly generalizes the previous results. Besides, we are concerned with edge-preservation since we have to deal with blurring effects along object borders in the non-binary case. Finally, we derive global optimality conditions for our multiclass approach.

2 Computerized Tomography

Beginning with the discovery of X-rays by Conrad Wilhelm Röntgen (1845-1923) in 1895, it was not before 1972 till the first computed tomographic scanner was build by Sir Geofrey Hounsfield (1919-2004) and Allan MacLeod Cormack (1924-1998). They both independently developed some of the algorithms and shared the Nobel price for their pioneering work in 1972.

Since then, the typical setup of a scanning device includes a single or multiple X-ray sources on one side and detectors on the opposite side. Together, they are partly or fully rotated around the object while the source constantly emits X-rays towards the detectors. While travelling through the object, radiation becomes partly absorbed depending on the material's density at the current spatial position and, finally, when the X-rays arrive at the detectors, their remaining intensity is measured. Therewith and with the intensity emitted at the source the amount of radiation absorbed by the object follows immediately. Reconstruction algorithms explore this information in order to recover an image of the object's spatially varying density.

It is interesting that the mathematical foundation for tomographic reconstructions has already been laid in 1917 by Johann Radon (1887-1956) [109] who was even able to provide a solution to the reconstruction problem, decades before the first scanner was build. This is even more astonishing as Radon considered the problem from a pure mathematical point of view without any awareness of its practical implications.

This chapter begins with further information concerning the physical background of X-rays, section 2.1. Subsequently, section 2.2, we present the most prominent imaging geometries, a description of the regular pattern in which the X-rays are arranged. In section 2.3, we introduce two different types of reconstruction algorithms, the filtered backprojection and the algebraic reconstruction method (ART), each approaching the reconstruction problem from a different direction. The filtered backprojection, on one hand, exploits the result of the Fourier slice theorem and belongs to the class of transform-based methods. ART, on the other hand,

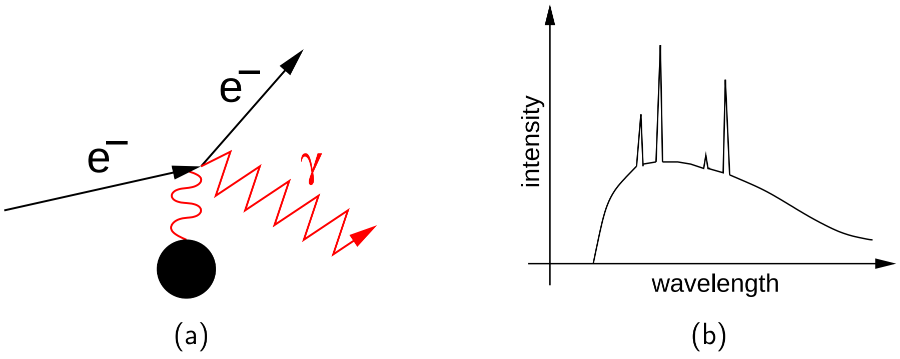


Figure 2.1: Radiation at various wavelengths is created by colliding electrons with atoms (a). This leads to the continuous part of the curve in figure (b) and is termed after the German expression "Bremsstrahlung". The discontinuities, called characteristic radiation, occur if the collision energy is sufficient to push an electron out of the inner shell whose place is immediately filled by another electron.

utilizes an algebraic formulation of the reconstruction problem and is, therefore, more related to the approaches typically used in discrete tomography.

2.1 Physics of Computerized Tomography

In contrast to microwaves and ultrasound, X-rays are ideally non-diffracting, i.e. they describe a straight line L while travelling from the source to the detector. On their way, some of the initial radiation becomes absorbed depending on the density at the current spatial position. Let I_0 denote the original intensity of an X-ray and let $\mu(\mathbf{x})$ be the function that relates the spatial position $\mathbf{x}$ to the attenuation of the object at $\mathbf{x}$. The remaining intensity I of the radiation which arrives at the detector is explained by the Beer-Lambert law,

$$I = I_0 \exp\left(-\int_L \mu(\mathbf{x}(\tau))\,\mathrm{d}\tau\right) \iff \log\frac{I_0}{I} = \log I_0 - \log I = \int_L \mu(\mathbf{x}(\tau))\,\mathrm{d}\tau. \qquad (2.1)$$

The process that recovers the attenuation function $\mu(\mathbf{x})$ from measurements I and I_0 is called reconstruction and is, in mathematical terms, an inversion of the integral operator on the right side of equation (2.1).

From a physical point of view, X-rays are electromagnetic radiation with a wavelength of about 10^{-10} meters. They are produced by accelerating electrons and colliding them with atoms, e.g. in medical applications metal atoms like Tungsten or Molybdenum are typically employed as target materials. Upon collision, the electrons emit energy in form of X-rays, see figure 2.1 (a) for an illustration of this process, and as they are differently decelerated their energy is emitted at different wavelengths. This leads to the continuous part of the distribution shown in figure 2.1 (b) and is termed after the German word "Bremsstrahlung". There is also a minimal wavelength corresponding to the maximal collision energy. This energy is reached if an electron is completely stopped and, thus, all its energy is converted into radiation. The discontinuous peaks in the distribution, figure 2.1 (b), are called characteristic radiation and occur if the energy of an electron is high enough to push out an electron from the inner shell of the target atom. The free position is immediately filled up with electrons from higher levels whereby X-ray photons are emitted. This kind of radiation must necessarily be discontinuous since there are only specific jumps possible.

2.2 Imaging Geometries

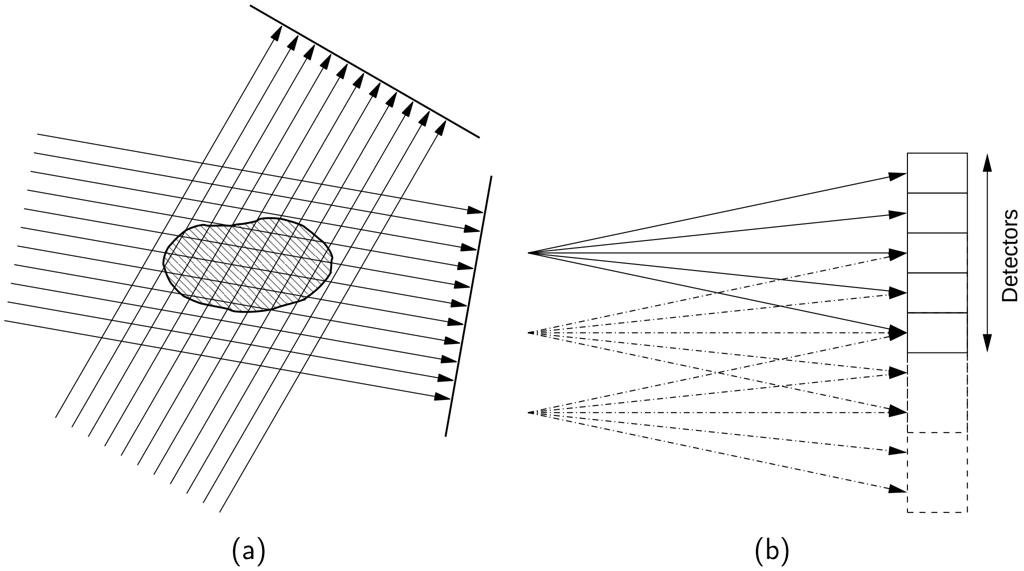


Figure 2.2: Parallelbeam geometry: (a) A set of parallel rays is shot through the object from different directions. (b) Illustration of a second-generation CT-scanner. Source and detectors are both moved (vertically) perpendicular to the projection direction (horizontally). Therefore, it is possible to measure not only the horizontal projection (first-generation) but at the same time also slightly rotated parallel projections.

During the scanning process, source and detector move around the object and the source emits X-rays towards the detector. For this purpose, the X-rays are typically arranged according to some regular pattern which is referred to as scanning or imaging geometry [99]. Among the

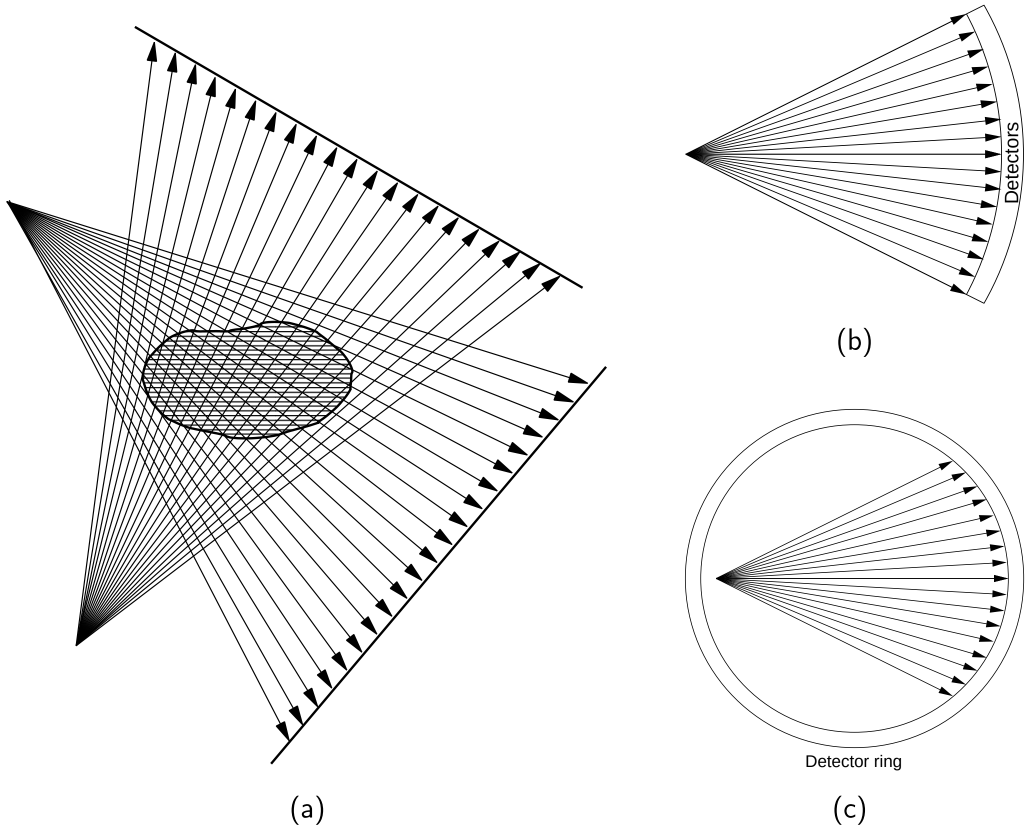


Figure 2.3: Fanbeam geometry: (a) In contrast to the parallelbeam geometry, where the source has to be moved by a small increment in a single projection, the fanbeam geometry shots a whole fan of rays and remains stationary within each projection. (b), (c) Illustration of a third and fourth generation scanner.

most prominent are the parallelbeam, fanbeam, and conebeam geometry.

In parallelbeam geometry a set of parallel X-rays is shot from each direction, figure 2.2. This geometry is implemented physically in the first-generation of scanners with a single source and a single detector cell [99]. For a fixed angle, source and detector are moved by a small amount perpendicular to the projection direction. By this, the object is scanned by a set of parallel rays and afterwards the procedure is repeated from a different angle, figure 2.2 (a).

In the second-generation of CT-scanners the single detector cell has been replaced by multiple detector cells [99] which allows to measure multiple projections at the same time, figure 2.2 (b). Besides that, the overall operation of the scanner remains the same as for the first-generation scanners.

The major drawback of the parallelbeam geometry is that the source has to be mechanically

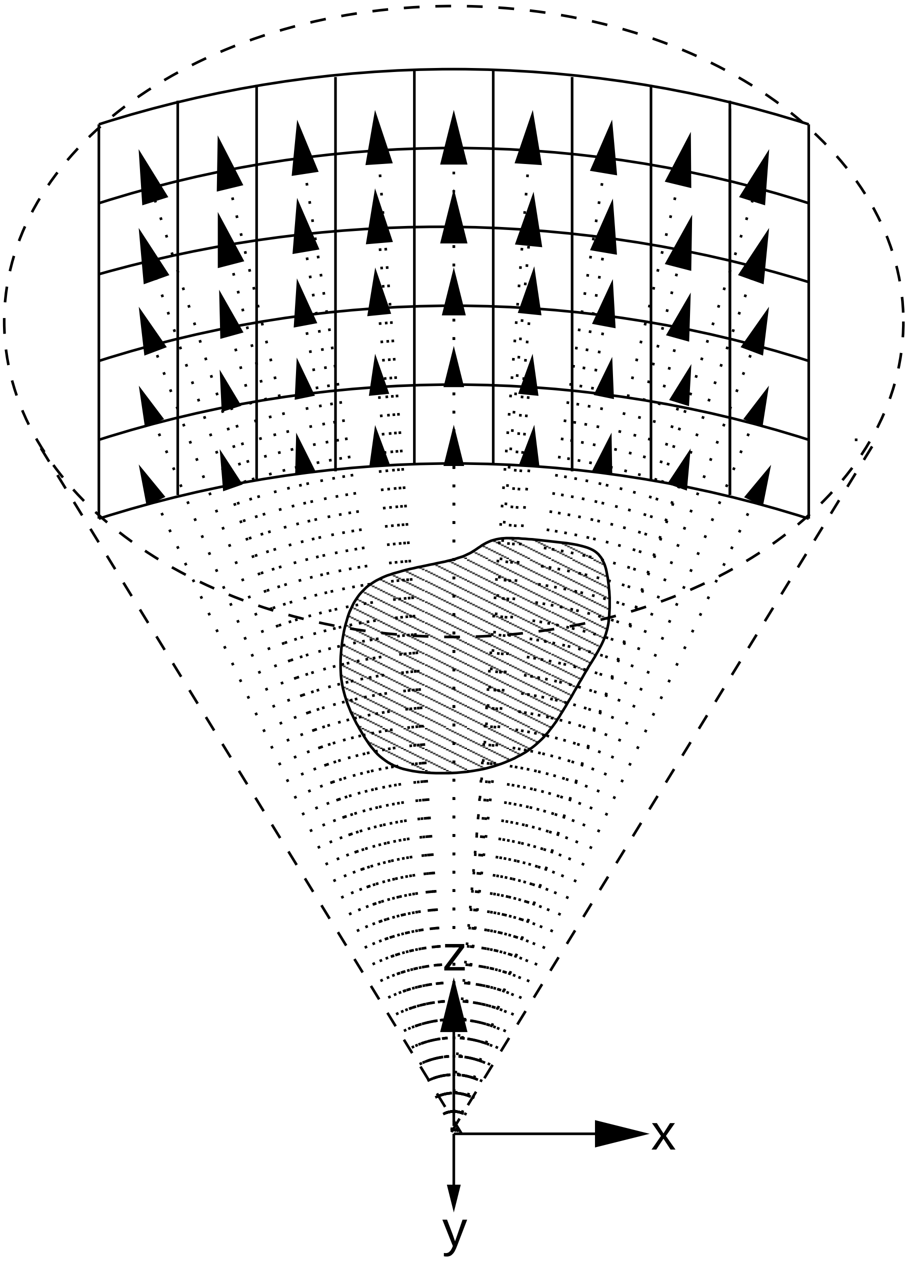


Figure 2.4: Conebeam geometry: The parallelbeam and fanbeam geometry aligns the rays within a plane and respectively the detectors are 1-dimensional arrays of detector cells. The conebeam geometry, in contrast, makes use of a 2-dimensional arrangement of detector cells and, therefore, aligns the rays within a cone.

moved to a new position for each single ray. This not only affects the quality of the projection but is time consuming as well. The fanbeam geometry, figure 2.3 (a), improves on both as each projection is acquired by a whole fan of rays shot from a stationary source. Further, refinements of the fanbeam geometry were introduced in the third and fourth generation of computer tomographs where either a rotating detector-source system, figure 2.3 (b), or a ring of stationary detectors and a rotating source, figure 2.3 (c), is used. Particularly, the latter enhances the imaging quality as the mechanical movement of the heavy detectors has been abandoned as well. However, a whole ring of detectors is quite expensive.

As the term conebeam suggests the source emits the X-rays in a cone shape. Thus, the detectors consist of a two-dimensional array of cells, in contrast to the previous geometries where either a single or a one-dimensional array is used. This leads to better reconstruction results as source and detector are usually not only rotated around the object, within the x-y-plane, but also moved along the rotation axis, z-axis. Due to the three-dimensional alignment of the rays, the conebeam geometry leads to a significant better resolution in the z-direction and is, therefore, mostly applied in modern medical CT-scanners.

2.3 Mathematics of Computerized Tomography

Today, the filtered backprojection is the method of choice for many CT applications. It is based on the Fourier slice theorem which yields a correspondence between projections in the spatial domain and subspaces in the frequency domain. In contrast, iterative reconstruction methods, like the algebraic reconstruction technique (ART), provide an algebraic approach to the reconstruction problem and are in this sense more closely related to typical discrete reconstruction methods.

This section introduces the Radon transform at first and provides the mathematical background of the filtered backprojection and ART subsequently. Further details, concerning CT and ART, can be found in [74] and [100, 99] where the latter cover the broad range of tomographic reconstruction methods, among them also non-linear tomography. Readers particularly interested in the filtered backprojection and its efficient implementation should consider [129] as well.

2.3.1 Radon Transform

Given a function $f : \mathbb{R}^2 \to \mathbb{R}$, the Radon transform of f is then defined as the integral of f restricted to a straight line. More specifically, let the minimal signed distance between the origin

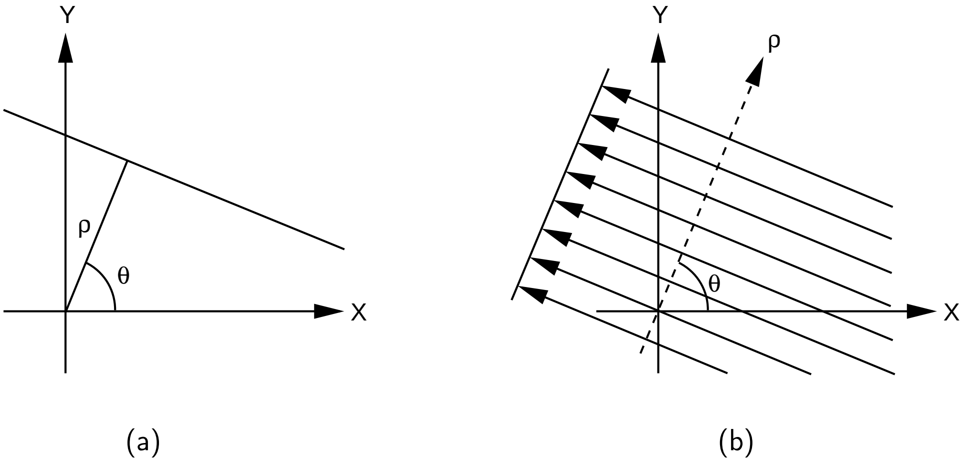


Figure 2.5: (a) A line in $\mathbb{R}^2$ defined by the angle $\theta \in [0, \pi[$ and the minimal distance to the origin $\rho \in \mathbb{R}$. (b) Given a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, the Radon transform of each line is defined as the integral of $f(x, y)$ restricted to that particular line. In view of the parallelbeam geometry, section 2.2, the family of all Radon transforms with fixed angle θ yields the projection of f under θ.

and that line be denoted by ρ and assume the line is parameterized by t and θ, figure 2.5 (a). The parameter space (ρ, t) and the coordinate space (x, y) are related to each other by a rotation of the coordinate system,

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \rho \\ t \end{pmatrix}. \tag{2.2}$$

A substitution of the arguments of f with equation (2.2) delivers the formal definition of the Radon transform,

$$\begin{aligned} \mathcal{R}(\rho, \theta)\,[f(x, y)] &:= \int_{-\infty}^{\infty} f(\rho\cos\theta - t\sin\theta, \rho\sin\theta + t\cos\theta)\,\mathrm{d}t && (2.3) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\,\delta(\rho - x\cos\theta - y\sin\theta)\,\mathrm{d}x\,\mathrm{d}y. && (2.4) \end{aligned}$$

Using

$$\int_{-\infty}^{\infty} f(x)\,\delta(ax + b)\,\mathrm{d}x = \frac{1}{|a|}\int_{-\infty}^{\infty} f\left(\frac{x - b}{a}\right)\delta(x)\,\mathrm{d}x = \frac{1}{|a|} f\left(\frac{-b}{a}\right), \tag{2.5}$$

equality of equation (2.3) and equation (2.4) holds for the case of $\sin\theta \neq 0$ due to

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ \delta(\rho - x\cos\theta - y\sin\theta)\ \mathrm{d}x\ \mathrm{d}y \quad = \tag{2.6}$$

$$\frac{1}{|\sin\theta|}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(x, -\frac{y}{\sin\theta} + \frac{\rho}{\sin\theta} - x\cot\theta\right)\ \delta(y)\ \mathrm{d}x\ \mathrm{d}y \quad = \tag{2.7}$$

$$\frac{1}{|\sin\theta|}\int_{-\infty}^{\infty} f\left(x, \frac{\rho}{\sin\theta} - x\cot\theta\right)\ \mathrm{d}x \quad = \tag{2.8}$$

$$\int_{-\infty}^{\infty} f\left(\rho\cos\theta - t\sin\theta, \rho\sin\theta + t\cos\theta\right)\ \mathrm{d}t, \tag{2.9}$$

while the case $\sin\theta = 0$ follows similarly. Further, equation (2.9) follows from equation (2.8) by substituting x with $\rho\cos\theta - t\sin\theta$.

Concerning the parallelbeam geometry from section 2.2, the Radon transforms of all lines with angle θ setup the projection of f under θ,

$$P_\theta(t) := \mathcal{R}(t,\theta). \tag{2.10}$$

A frequently employed representation of projections are sinograms which are obtained by stacking all projections. Hence, sinograms are plots of $P_\theta(t)$ in the variables θ and t where θ varies along the rows and t along the columns. Figure 2.7 (b) shows the sinogram of the Lena image, figure 2.7 (a).

2.3.2 Fourier Slice Theorem

The Fourier slice theorem is fundamental to all transform-based reconstruction techniques, such as the filtered backprojection, as it relates the Fourier transform[1] of an image to the Fourier transform of a projection obtained from this image.

Let us identify the Fourier transform of projection $P_\theta(t)$ with respect to parameter ν by

$$S_\theta(\nu) := \mathcal{F}[P_\theta(t)] = \int_{\infty}^{\infty} P_\theta(t)\exp\left(-\mathrm{i}\ 2\pi\nu t\right)\ \mathrm{d}t. \tag{2.11}$$

After supplementing equation (2.4) in (2.11) and rearranging terms we obtain

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ \left\{\int_{-\infty}^{\infty} \delta(t - x\cos\theta - y\sin\theta)\exp\left(-\mathrm{i}\ 2\pi\nu t\right)\ \mathrm{d}t\right\}\ \mathrm{d}x\ \mathrm{d}y. \tag{2.12}$$

[1] Note that there exist different definitions of the Fourier transform, throughout this work we use the definitions described in appendix B

Applying equation (2.5), finally, yields the Fourier slice theorem

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\ \exp\left(-\mathrm{i}\ 2\pi\nu\left(x\cos\theta + y\sin\theta\right)\right)\ \mathrm{d}x\ \mathrm{d}y = \tag{2.13}$$

$$\mathcal{F}[f(x,y)]\Big|_{\mathbf{u}=\nu(\cos\theta,\sin\theta)^\top}\ , \tag{2.14}$$

which reads in condensed form as

$$S_\theta(\nu) = \mathcal{F}[P_\theta(t)] = \mathcal{F}[f(x,y)]\Big|_{\mathbf{u}=\nu(\cos\theta,\sin\theta)^\top} = F(\nu\cos\theta, \nu\sin\theta). \tag{2.15}$$

Concrete, the Fourier slice theorem states that the 1-dimensional Fourier transform of the projection taken from an angle θ equals the slice obtained by restricting the 2-dimensional Fourier transform of the image to the line under angle θ. Although we specifically consider here projections obtained from an image using the parallelbeam geometry, analogous forms of the theorem can be found for projections of volumes and other imaging geometries.

2.3.3 Direct inverse Radon transform

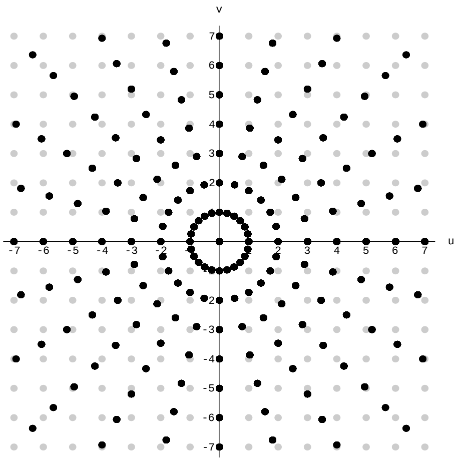


Figure 2.6: Direct inverse Radon transform: Given projection data $P_\theta(t)$ from a scan over 180°, first the 1-dimensional Fourier transform is applied to each projection, $S_\theta(\nu) := \mathcal{F}[P_\theta(t)]$. The Fourier slice theorem, equation (2.15), states that each $S_\theta(\nu)$ is a slice of the image in the Fourier domain, black dots. Thus, the idea of the direct inverse Radon transform is to obtain the 2-dimensional Fourier transform of the image by interpolating the polar grid, black dots, of $S_\theta(\nu)$ into the Cartesian grid, gray dots, (regridding). Unfortunately, this approach is numerically instable and the data in polar coordinates becomes sparser with increasing distance from the origin. Hence, this approach is not used for real applications.

The Fourier slice theorem, equation (2.15), provokes the idea of a simple reconstruction algorithm. Given some projection data $P_\theta(t)$ the Fourier slice theorem tells us that the Fourier transform of $P_\theta(t)$ equals a slice of the Fourier transformed image. Hence, we could interpolate the polar grid of $\mathcal{F}[P_\theta(t)]$ into the Cartesian grid of the frequency domain, this is called regridding. Afterwards, we simply apply the inverse Fourier transform and would expect a solution of the reconstruction problem. This procedure is known as direct inverse Radon transform for which it is, unfortunately, reported [74, 99, 31] that it is not advisable for at least two good reasons. First of all, regridding is numerically instable as it introduces errors leading to significant artifacts in the resulting image. Secondly, as can be seen in figure 2.6 the data available for the interpolation becomes sparser with increasing distance from the origin. Consequently, the approximation gets worse for higher frequencies and accuracy of finer details gets lost inevitably.

2.3.4 Filtered Backprojection

A slightly different, yet much more fruitful idea than the direct inverse Radon transform is to exchange the Cartesian coordinate system in the frequency domain of the Fourier transform through a polar coordinate system. This is achieved by a substitution of $\mathbf{u}$ with $(\nu\cos\theta, \nu\sin\theta)^\top$ in the formula of the 2-dimensional inverse Fourier transform, see appendix B. Then, the inverse Fourier transform reads

$$f(x,y) = \int_0^{2\pi}\int_0^\infty \nu F(\nu\cos\theta, \nu\sin\theta)\exp\left(\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right)\mathrm{d}\nu\ \mathrm{d}\theta. \tag{2.16}$$

By splitting the integral, equation (2.16), into $\theta \in [0,\pi]$ and $\theta \in [\pi, 2\pi]$,

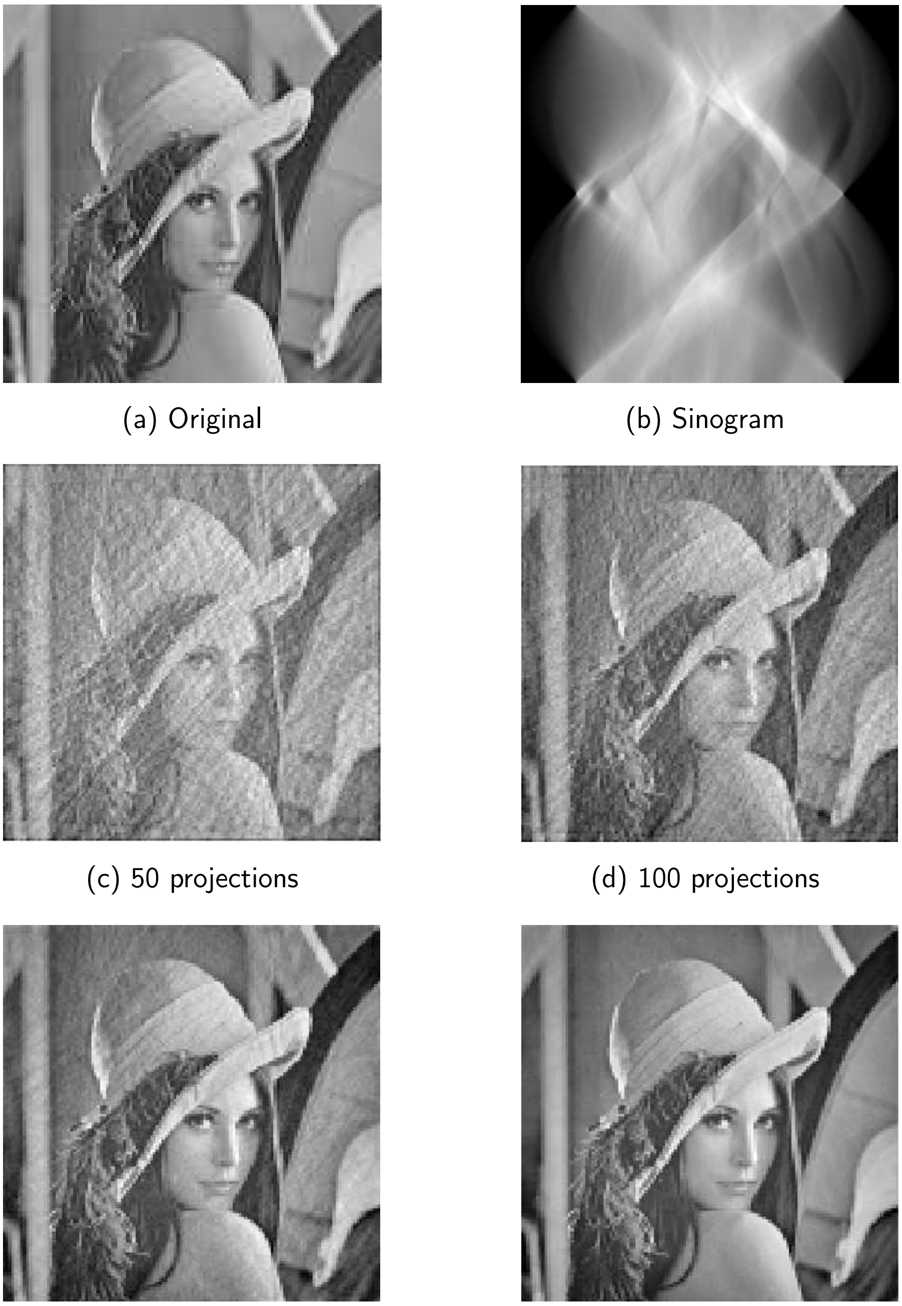
$$\begin{aligned} &\int_0^{\pi}\int_0^\infty \nu F(\nu\cos\theta, \nu\sin\theta)\exp\left(\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right)\mathrm{d}\nu\ \mathrm{d}\theta \quad + \qquad (2.17)\\ &\int_0^{\pi}\int_0^\infty \nu F(\nu\cos(\theta+\pi), \nu\sin(\theta+\pi))\exp\left(\mathrm{i}2\pi\nu(x\cos(\theta+\pi) + y\sin(\theta+\pi))\right)\mathrm{d}\nu\ \mathrm{d}\theta, \end{aligned}$$

and using the fact that

$$\begin{pmatrix} \nu\cos(\theta+\pi) \\ \nu\sin(\theta+\pi) \end{pmatrix} = \begin{pmatrix} -\nu\cos\theta \\ -\nu\sin\theta \end{pmatrix} \tag{2.18}$$

equation (2.16) can be written as

$$\begin{aligned} &\int_0^{\pi}\int_0^\infty \nu F(\nu\cos\theta, \nu\sin\theta)\exp\left(\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right)\mathrm{d}\nu\ \mathrm{d}\theta \qquad (2.19)\\ + &\int_0^{\pi}\int_0^\infty \nu F(-\nu\cos\theta, -\nu\sin\theta)\exp\left(-\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right)\mathrm{d}\nu\ \mathrm{d}\theta. \end{aligned}$$



(a) Original

(b) Sinogram

(c) 50 projections

(d) 100 projections

(e) 300 projections

(f) 1000 projections

Figure 2.7: (a) Original of the famous Lena image, 128×128. (b) Sinogram over 1000 projections, i.e. the projections are (vertically) stacked together such that each row contains a projection from a different angle θ. (c)-(f) Reconstructions obtained with our implementation of the filtered backprojection algorithm 1. For each experiment, the number of projections has been increased. However, projection were always taken equally spaced over 180 degrees: (c) 50, (d) 100, (e) 300, and (f) 1000 projections.

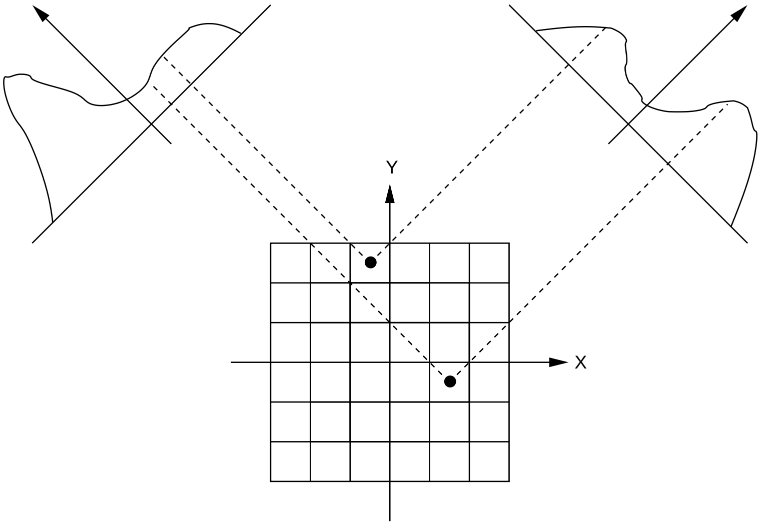


Figure 2.8: Schematic drawing of the filtered backprojection algorithm. In the “filtering” step, the algorithm computes the Fourier transform $\mathcal{F}[P_k]$ of each projection P_k and applies a highpass filter H in the frequency domain $H \cdot \mathcal{F}[P_k]$. Afterwards, the filtered frequency signals are transformed back to the spatial domain $\mathcal{F}^{-1}[H \cdot \mathcal{F}[P_k]]$ (inverse Fourier transform). For illustration purposes only two of these highpass filtered projections are depicted in this figure. The subsequent “backprojection” step projects each pixel (again, this is shown here only for two pixels) into all filtered projections, evaluates $\mathcal{F}^{-1}[H \cdot \mathcal{F}[P_k]]$ at the projected position, and sums the latter values.

Substituting and exchanging of integration borders yields

$$f(x, y) = \int_0^\pi \left[\int_{-\infty}^{\infty} |\nu| \, F(\nu \cos\theta, \nu \sin\theta) \exp\left(\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right) \mathrm{d}\nu \right] \mathrm{d}\theta. \qquad (2.20)$$

By virtue of the Fourier slice theorem, equation (2.15), we write

$$f(x, y) = \int_0^\pi \left[\int_{-\infty}^{\infty} |\nu| \, S_\theta(\nu) \exp\left(\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right) \mathrm{d}\nu \right] \mathrm{d}\theta \qquad (2.21)$$

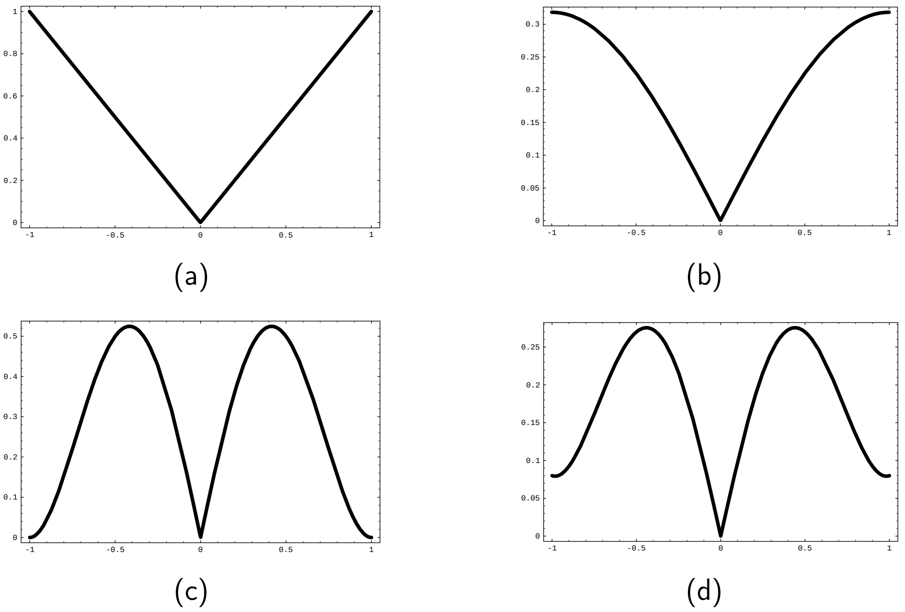
or alternatively

$$f(x, y) = \int_0^\pi Q_\theta(x\cos\theta + y\sin\theta) \, \mathrm{d}\theta \qquad (2.22)$$

with

$$Q_\theta := \int_{-\infty}^{\infty} |\nu| \, S_\theta(\nu) \exp\left(\mathrm{i}2\pi\nu(x\cos\theta + y\sin\theta)\right) \mathrm{d}\nu. \qquad (2.23)$$

Usually, equation (2.23) is referred to as filtering and equation (2.22) as backprojection part. Due to the convolution theorem which relates convolution and multiplication in spatial and



(a) (b)

(c) (d)

Figure 2.9: Filters: Let ν_u denote the length of the data array and ν_l the size of the filter, $\nu_l \leq \nu_u$. (a) Ram-Lak filter: $H(\nu) := |\nu|, \; |\nu| \leq \nu_l$; (b) Shepp Logan filter: $H(\nu) := \frac{1}{2}|\nu| \sin(\pi\nu/2\nu_l) / (\pi\nu 2\nu_l), \; |\nu| \leq \nu_l$; (c) Hann filter: $H(\nu) := |\nu| (1 + \cos(\pi\nu/\nu_l)), \; |\nu| \leq \nu_l$; (d) Generalized Hamming filter: $H(\nu) := |\nu| (\alpha + (1-\alpha)\cos(\pi\nu/\nu_l)), \; |\nu| \leq \nu_l,$ typically α ranges within $0.5 - 0.54$. Values of ν between $\nu_l < \nu_u$ are set to zero.

frequency domain, it is also possible to perform the filtering part in the spatial domain which is then called convolution backprojection [129]. Yet another alternative is filtering after back-projection [129].

According to equation (2.23) a highpass filter $|\nu|$, Ram-Lak filter, has to be applied to the projection data. For practical applications, this is, however, not the best choice since the Ram-Lak filter tends to amplify noise which is usually contained in the higher frequencies as well and, hence, other filters, figure 2.9, are often used instead [129]. Though we consider the filtered backprojection in our experiments only for noiseless projections, we also used other filters in order to get best possible results. However, as expected, these results were quite similar and, therefore, we provide only the results obtained from a varying number of projections in figure 2.7. All our experiments were performed with the filtered backprojection according to algorithm 1 which we implemented with Mathematica[2].

[2] Mathematica is a commercial computer algebra system available from Wolfram Research, www.wolfram.com.

Algorithm 1 Filtered Backprojection

```
Require: $P_1, ..., P_n$ projections
Require: $\phi_1, ..., \phi_n$ angles {corresponding to $P_k$}
Require: $\rho$ distance between rays {Assumed to be constant for all projections}
Require: $m$ number of rays per projection
Require: $I(x, y) := 0$ initialized for all $1 \leq x \leq sX,\ 1 \leq y \leq sY$
Require: $H$ highpass filter in frequency domain
  for $k := 1$ to $n$ do
    $\tilde{P}_k := \mathcal{F}^{-1}[H \cdot \mathcal{F}[P_k]]$
  end for
  for $y := 1$ to $sY$ do
    for $x := 1$ to $sX$ do
      for $k := 1$ to $n$ do
        $s := \frac{1}{2}(2x - sX - 1)\sin(\phi_k) - \frac{1}{2}(2y - sY - 1)\cos(\phi_k)$
        $t := \left(s + \frac{1}{2}\, m\, \rho\right)(m - 1)/(m\, \rho) + 1$
        $\lambda := t - \lfloor t \rfloor$
        $v := (1 - \lambda)\, \tilde{P}_k(\lfloor t \rfloor) + \lambda\, \tilde{P}_k(\lceil t \rceil)$
        $I(x, y) := I(x, y) + v$
      end for
    end for
  end for
```

2.3.5 Iterative Reconstruction Methods

In contrast to transform-based methods, like the filtered backprojection, iterative reconstruction methods pursue an algebraic approach to the reconstruction problem. By this, they are applicable even if it is not possible to acquire a large amount of data or projections over $180°$, both is mandatory for the filtered backprojection algorithm. However, due to their lack of accuracy and speed of convergence they are only used when both is not critical.

In the discretized image domain each pixel is associated with an unknown variable and each ray composes a linear equation. Thus, the whole reconstruction problem is described by a linear equation system. We will elaborate this further in section 3.2 since our approaches utilize the very same discretization model. For now, however, it is sufficient to consider the reconstruction problem as given by means of the linear equation system, $\mathbf{Ax} = \mathbf{b}$, $\mathbf{x} \in \mathbb{R}^n$, where we stress that it is not practicable to solve such systems directly as the number of unknowns becomes

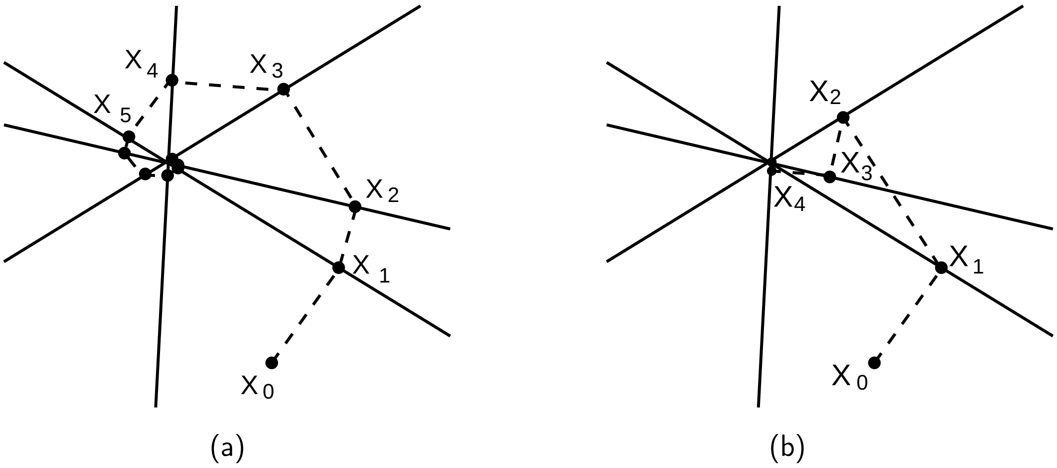


Figure 2.10: (a) ART (algebraic reconstruction technique) solves a linear equation system approximately. Starting from an initial point $\mathbf{x}_0$, ART proceeds by iteratively projecting onto the hyperplanes defined by the linear equation system. This results in a sequence of vectors $\mathbf{x}_i$ converging towards the solution of the equation system. As depicted in figure (b) the order in which the hyperplanes are visited influences the speed of convergence. The angle between two succeeding planes should preferably be orthogonal.

typically very large.

Based on the "method of projections", Kaczmarz [73] and Tanabe [127], the algebraic reconstruction technique (ART) is an iterative method for approximately solving linear equation systems. In general, a single equation $\mathbf{a}^\top \mathbf{x} = b$ describes a hyperplane $\{\mathbf{x} \mid \mathbf{a}^\top \mathbf{x} - b = 0\}$ in $\mathbb{R}^n$ and the solution space is the intersection of all such hyperplanes. If the intersection contains only a single point then this point is the unique solution of the linear equation system and if the intersection is empty there is no solution at all. The idea of ART is to consider each equation separately, instead of the whole equation system at once. Let us, therefore, assume that the row vectors of matrix $\mathbf{A}$ are uniquely enumerated by $\mathbf{a}_i^\top$. Starting with an initial guess $\mathbf{x}_0$, the ART method computes in the first iteration the projection onto the first equation which yields a new solution vector $\mathbf{x}_1$. In the next iteration, $\mathbf{x}_1$ is then projected onto the second equation and so on, see figure 2.10. The general formula for the projection of a point $\mathbf{x}_{i-1}$ onto equation i is derived below:

$$\begin{aligned}
&\mathbf{a}_i^\top \mathbf{x}_i = b_i \\
&\quad \mathbf{x}_i \overset{!}{=} \mathbf{x}_{i-1} + \lambda\, \mathbf{a}_i \iff \lambda = \frac{b - \mathbf{a}_i^\top \mathbf{x}_{i-1}}{\mathbf{a}_i^\top \mathbf{a}_i}\ , \quad \mathbf{a}_i^\top \mathbf{a}_i \neq 0 \\
&\quad \mathbf{x}_i = \mathbf{x}_{i-1} - \frac{\mathbf{a}_i^\top\ \mathbf{x}_{i-1} - b_i}{\mathbf{a}_i^\top\ \mathbf{a}_i}\, \mathbf{a}_i \qquad (2.24)
\end{aligned}$$

Alternatively to equation (2.24), one computes the update for $\mathbf{x}_i$ directly,

$$\Delta\mathbf{x}_i := \mathbf{x}_i - \mathbf{x}_{i-1} = \frac{b_i - \mathbf{a}_i^\top \, \mathbf{x}_{i-1}}{\mathbf{a}_i^\top \, \mathbf{a}_i} \, \mathbf{a}_i. \tag{2.25}$$

Once all equations have been visited, a new cycle starts by repeating the procedure initialized with the last vector $\mathbf{x}_k$. The overall ART algorithm is shown in algorithm 2. In case of a unique solution, convergence has been proven in [127] and, further, if an infinite number of solutions exists ART converges towards the solution which is closest to the initial starting point. If there is no solution at all, the ART method oscillates in a neighborhood of the hyperplanes' intersections. The order in which the equations are visited, greatly influences the speed of convergence. As illustrated in figure 2.10, it is favorable if the angle between two hyperplanes is large, i.e. ideally the planes are perpendicular. Hence, the ART procedure is accelerated by carefully choosing the order of the hyperplanes [70] in a preprocessing step. The reconstruction quality can be further improved by relaxing the update scheme to $\lambda\Delta\mathbf{x}_i$ where λ is chosen less than 1 or is a function decreasing towards 0 with increasing number of iterations. However, this usually deteriorates the overall convergence speed.

Algorithm 2 Algebraic Reconstruction Technique (ART)

```
Require: $\epsilon > 0$
Require: $c_{\max}$ maximal numbers of cycles.
  $\mathbf{x} := \mathbf{0}$
  $n :=$ Number of rows of $\mathbf{A}$.
  $\sigma :=$ Permutation such that $|\mathbf{a}_{\sigma_{i+1}}^\top \mathbf{a}_{\sigma_i}|$ becomes small.
  $\mathbf{A} :=$ Permute rows of $\mathbf{A}$ according to $\sigma$.
  $\mathbf{b} :=$ Permute components of $\mathbf{b}$ according to $\sigma$.
  $c := 1$
  $\mathbf{x}_o := \mathbf{x} + \mathbf{1}$
  while $\|\mathbf{x} - \mathbf{x}_o\|_2 \leq \epsilon$ and $c < c_{\max}$ do
    $\mathbf{x}_o := \mathbf{x}$
    for $i := 1$ to $n$ do
      $\mathbf{x}_i := \mathbf{x}_{i-1} - \mathbf{a}_i \, (\mathbf{a}_i^\top \, \mathbf{x}_{i-1} - b_i)/(\mathbf{a}_i^\top \, \mathbf{a}_i)$          cf. equation (2.24)
    end for
    $c := c + 1$
  end while
```

Originally proposed in [42], simultaneous reconstruction methods, like the simultaneous iterative reconstruction technique (SIRT) and simultaneous algebraic reconstruction technique (SART), improve ART by updating the new solution only after all equations have been visited,

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \lambda_i \sum_{i=1}^{n} \omega_i \frac{\mathbf{a}_i^\top \ \mathbf{x}_{i-1} - b_i}{\mathbf{a}_i^\top \ \mathbf{a}_i} \ \mathbf{a}_i. \tag{2.26}$$

In equation (2.26) the ω_i are weighting parameters with $\sum_i \omega_i = 1$. As a single ray traverses only a small fraction of pixels, matrix $\mathbf{A}$ is typically sparse and unpropitiously selected weighting parameters can lead to slow convergence. For this case, component averaging (CAV) [35] is a simultaneous reconstruction approach which is improved especially for sparse systems.

The common structure of all iterative reconstruction methods is that they first compute some kind of correction from the previous solution and use this correction in order to obtain a new solution, see figure 2.11.

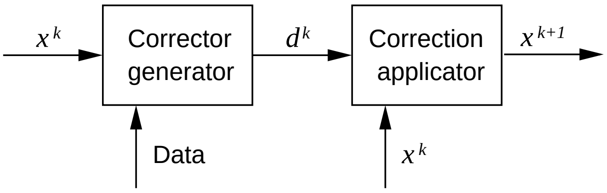


Figure 2.11: Iterative reconstruction methods: This drawing outlines the general structure of a single iteration. The current solution $\mathbf{x}^k$ is fed to the corrector generator which determines the correction vector $\mathbf{d}^k$ by means of given data. Afterwards, the correction applicator finds a new approximate solution $\mathbf{x}^{k+1}$.

3 Discrete Tomography

The continuous tomographic reconstruction methods from the previous chapter recover a real-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ from its projections. Assuming the image of f is some discrete set which is a priori known, e.g. the case $f : \mathbb{R}^n \rightarrow \{0, 1\}$ is widely concerned in literature [66], discrete tomography deals with this new type of reconstruction problem and algorithms designed for the discrete case posses several advantages compared to their continuous counterparts.

We have seen in the previous chapter that conventional tomographic methods typically require a lot of projection data in order to reveal satisfying results. For instance, compare the results in figure 2.7 where the Lena image has been reconstructed from a varying number of projections and the result obtained from as much as 300 projections is still not satisfying. Of course, this image is more complex than images which contain only a small number of values. However, the quality of the reconstruction only partly depends on the complexity of the image. In case of the filtered backprojection, a huge amount of projections is simply needed for the numerical approximation of the equations (2.22) and (2.23). Furthermore, a scan over $180°$ is prerequisite since the integration of θ in equation (2.22) ranges from 0 to π.

In order to demonstrate the performance of the filtered backprojection, algorithm 1, in case of binary tomography anyway, we conducted appropriate experiments with a varying number of projections ranging over $90°$ and $180°$. The results are shown in figure 3.1 and suggest that a large number of projections, ≈ 30 is necessary to be able to recognize the objects at least, and a scan over $180°$ is mandatory, as expected. Regarding discrete reconstruction algorithms, we stress the fact that it is possible to retrieve the original image from no more than three projections over $90°$.

Besides a formal definition of the reconstruction problem, this chapter presents potential applications for discrete tomography and different discrete reconstruction approaches. Among them are simulated annealing which has become popular [38, 94, 68, 91, 98, 86] due to the

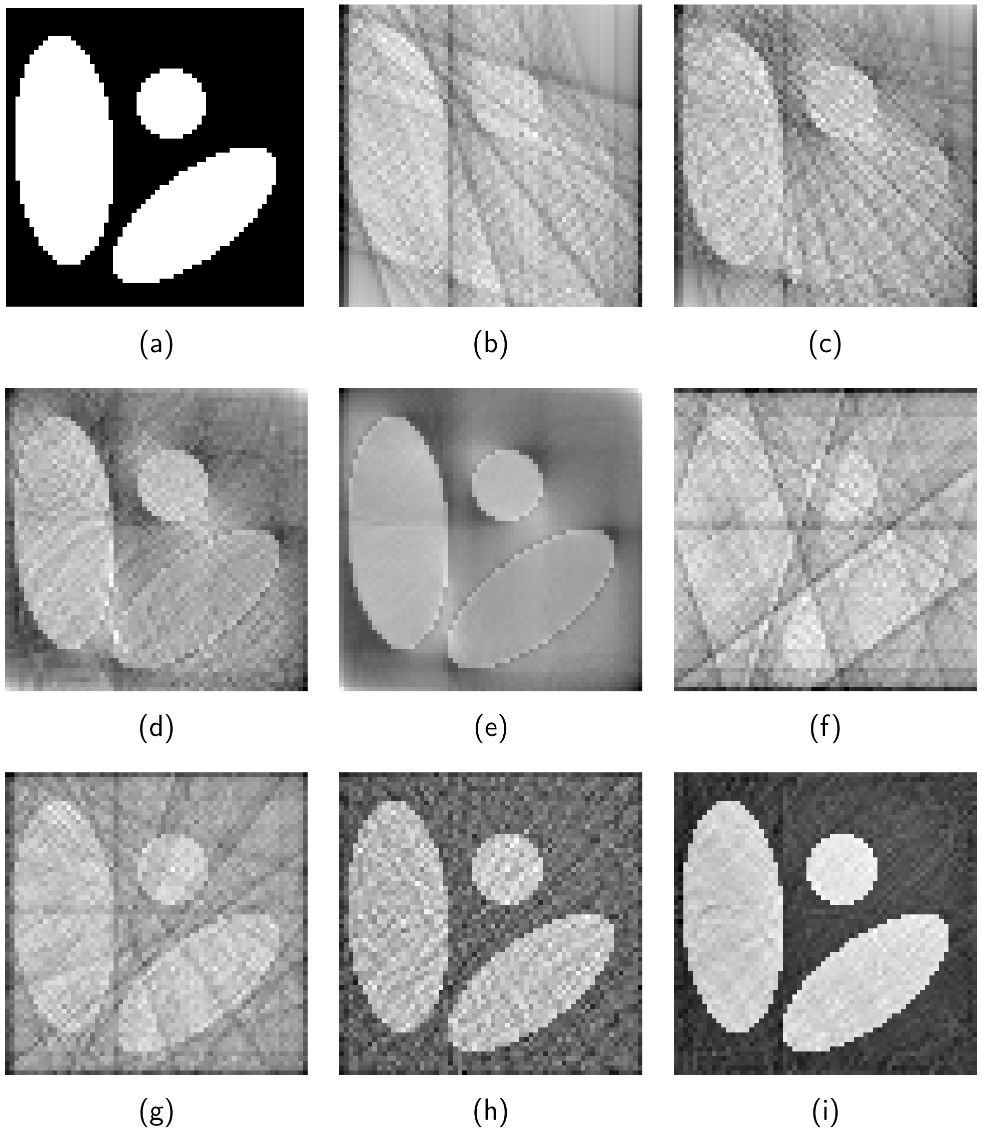


Figure 3.1: (a) Original image. (b)-(e) 5, 10, 30, 100 projections equally spaced over $90°$ and (f)-(i) over $180°$ respectively. The results demonstrate that the filtered backprojection does not provide accurate reconstructions for less than $180°$. Though the object is recognizable in (e) and maybe in (d) additional steps are necessary to retrieve the binary image. Concerning the results from $180°$ the reconstruction from 100 projections (i) yields a good reconstruction. Nonetheless, with algorithms specialized for discrete tomography image (a) can be perfectly reconstructed from no more than three projections, $0°$, $45°$, over $90°$.

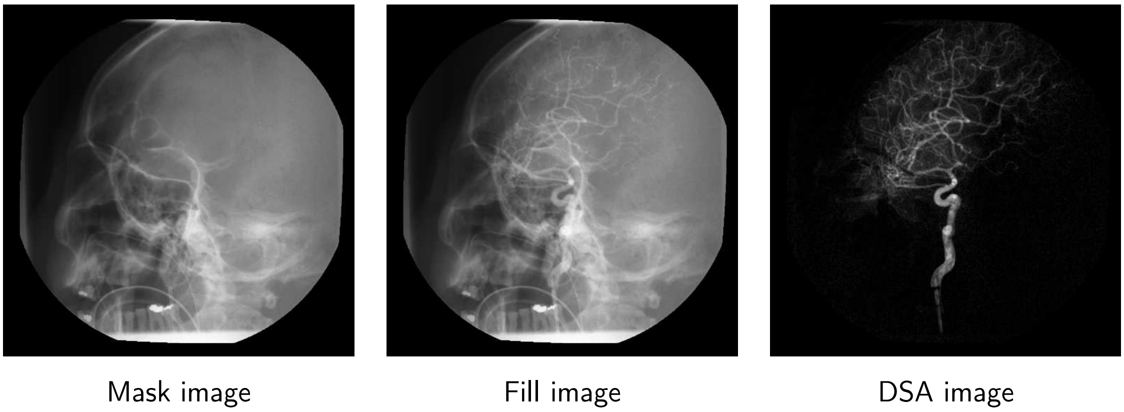
inherent combinatorial complexity of the problem. Concerning the filtered backprojection, we have seen already that it is not suitable for our needs, figure 3.1. The same is, however, not true for iterative reconstruction algorithms, like ART, since the idea to steer the overall continuous reconstruction process in some way towards a binary or discrete solution is tempting. Section 3.3.2 describes the binary steering approach proposed in [34] and shows that it indeed improves the results of ART for binary tomography. Finally, section 3.3.3 presents approaches based on network flows [51, 116, 8, 9, 7].

3.1 Applications

We present three selected applications for discrete tomography but emphasize that this is far from being complete. Discrete reconstruction problems belong to a broad class of optimization problems arising in a variety of contexts.

3.1.1 Electron Microscopy

The first application motivating and still impelling the field of discrete tomography is the reconstruction of crystalline structures. This is interesting in quality control where semiconductor materials for computer chips must possess certain features in order to work properly. A technique based on high resolution transmission electron microscopy, called QUANTITEM [125, 79] (QUANTITative analysis of the information provided by transmission Electron Microscopy), allows to measure the number of atoms lying on each line parallel to a set of directions. A natural way to define the problem mathematically is to assume that each atom is located at some integer position in $\mathbb{Z}^3$. Let $S \subset G \subset \mathbb{Z}^3$ where S is the set of positions occupied by an atom and G is a finite set such that only a finite number of positions must be considered. The reconstruction problem is to determine S from the projections provided by QUANTITEM. This can be regarded as a labeling problem where each element $\mathbf{z} \in G$ is assigned a label 0 or 1 depending whether $\mathbf{z} \notin S$ or $\mathbf{z} \in S$. It might also be tempting to use conventional reconstruction algorithms here but, unfortunately, only a small number of projections can be taken, i.e. 3,4, or 5, since the energy needed to acquire more projections would damage the object otherwise. Certainly, this amount of projection data is not sufficient for conventional algorithms and there is only hope for satisfying results by exploiting the discrete nature of the problem.



Mask image Fill image DSA image

Figure 3.2: The fill and the mask image show X-ray projections of a human head. In the fill image the vessels supplying the brain with blood are enhanced with contrast agent whereas the mask image is without. Digital subtraction angiography (DSA) subtracts the mask image from the fill image, thus, only the contrast agent inside the vessel remain.

3.1.2 Medical Imaging

In medical imaging, the X-ray expose of the patient is proportional to the amount of gathered projections and is, thus, a crucial parameter. A typical and frequently recurring task is to inspect the vascular system for pathological abnormalities, such could be a stenosis or an aneurysm, i.e. constriction and protuberance of a blood vessel respectively. Both are life-threatening if not treated properly and, in this context, digital subtraction angiography (DSA) is a well-known technique for separating the vessels and the remaining image parts. The key step is to take two X-ray images from the same perspective, one with contrast agent (fill image) which enhances the visual perception of the vessels and one without (mask image). Afterwards, both images are subtracted from each other and, thus, only the distribution of the contrast agent inside the vessel remains in the resulting image, figure 3.2. Under the assumption that the contrast agent is homogeneously and completely distributed within the organ, it is possible to reconstruct the distribution of the contrast agent as a binary volume from DSA projections. At this, 0 indicates the absence and 1 the presence of contrast agent at some spatial position.

Let μ_{Fill}, μ_{Mask} be functions describing the density of the volume filled with and without contrast agent, then the primary assumption, a homogeneous distribution of the contrast agent, can be written as

$$\mu_{Fill}(x) = \mu_{Mask}(x) + \rho\, \chi(x) \tag{3.1}$$

where χ is the binary indicator function describing the distribution of the contrast agent, scaled

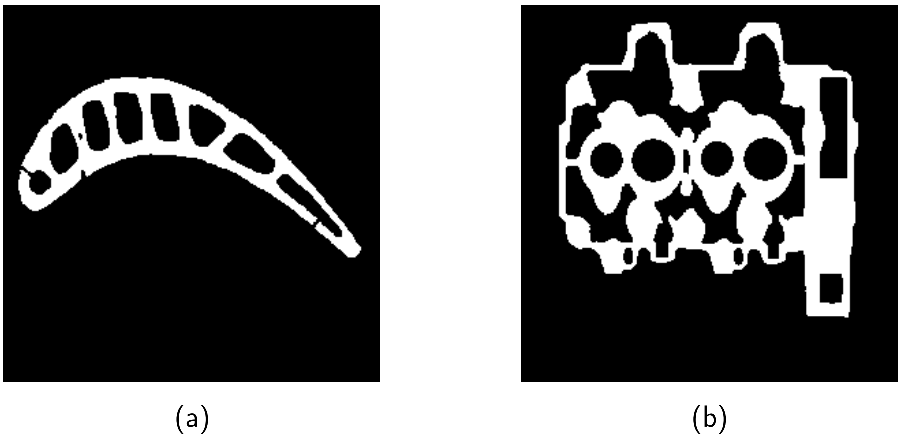


Figure 3.3: Non-destructive material testing: Cross sections through metal parts: (a) turbine blade and (b) cylinder block.

with the density of the contrast agent ρ. According to the Beer-Lambert law, equation (2.1), we have

$$\log \frac{I_0}{I} = \int_L \mu(x)\,\mathrm{d}x \tag{3.2}$$

where X-ray devices typically provide the value on the left of equation (3.2). The measured fill and mask projections are related to the projection of the binary volume by

$$\begin{aligned}\log \frac{I_0}{I_{Fill}} - \log \frac{I_0}{I_{Mask}} &= \int_L \mu_{Fill}(x)\,\mathrm{d}x - \int_L \mu_{Mask}(x)\,\mathrm{d}x \\ &= \int_L \mu_{Mask}(x) + \chi(x)\rho - \mu_{Mask}(x)\,\mathrm{d}x \\ &= \rho \int_L \chi(x)\,\mathrm{d}x.\end{aligned} \tag{3.3}$$

Surveys of discrete tomography in medical imaging can be found in [85] and [68]. A simulated annealing approach for the reconstruction of vascular structures has been proposed in [104] and [126] suggests a network flows approach in order to determine the structure of the left ventricle from two projections. The reconstruction of the heart chamber from biplane cardiac angiograms has been considered more recently in [108, 103]. Their algorithm reconstructs the volume slice by slice, thereby, propagating information not only from the previous slice but also over time as the heart muscle contracts and relaxes.

3.1.3 Non-destructive Material Testing

In material testing, tomographic imaging techniques are applied in order to find defects, like cracks or involvements, inside materials. Imagine for instance the turbine blades of an aircraft or other highly reliable products whose breaking resistance is very crucial. Due to the fact that tomographic methods are non-destructive the tested samples can be used further on and it is possible to test each piece individually and not only a set of random samples.

Material testing problems are just as made for discrete tomography since the object and the specific densities, respectively, are perfectly known in advance. Furthermore, it is even possible to use a specific prior model for the object itself since defects usually affect only small portion of the object. Figure 3.3 gives some impression of images as they might appear in material testing, a turbine blade on the left side and the cylinder block of a vehicle on the right side. For further details on the topic see for instance [84].

3.2 Problem Statement

In order to access the reconstruction problem algebraically the integral in equation (2.1) must be discretized properly. Therefore, two different discretization schemes are commonly used in discrete tomography, the fully-discrete and the image-based discretization scheme . Although we consider only the latter here, we emphasize that our algorithms are applicable in both cases, as we explain later on.

The fully-discrete scheme, figure 3.4 (a), involves discreteness in both, the image domain and the range, and originates from reconstruction problems arising in electron microscopy, section 3.1.1. Given finite subsets S and G of the integer lattice such that $S \subset G \subset \mathbb{Z}^n$. A single ray is conceived as a straight line and its corresponding projection counts the number of points in S traversed by that line. At this, the set G formally assures that only a finite number of points has to be considered, points outside of G do not contribute to the projections. Furthermore, lines are typically restricted to be discrete, i.e. lines with a rational slope traveling at least through one integer point in G. Given such projections and assuming S to be unknown the reconstruction problem seeks to recover S from the projection data. Note that the problem becomes trivial if the restriction to discrete lines is dropped, i.e. there exists a unique correspondence between points and lines in case of an irrational slope. Although this is sometimes mentioned in literature it is, nonetheless, of minor interest since it is impractical for real world applications.

In contrast to the fully-discrete scheme, the image-based discretization requires discreteness

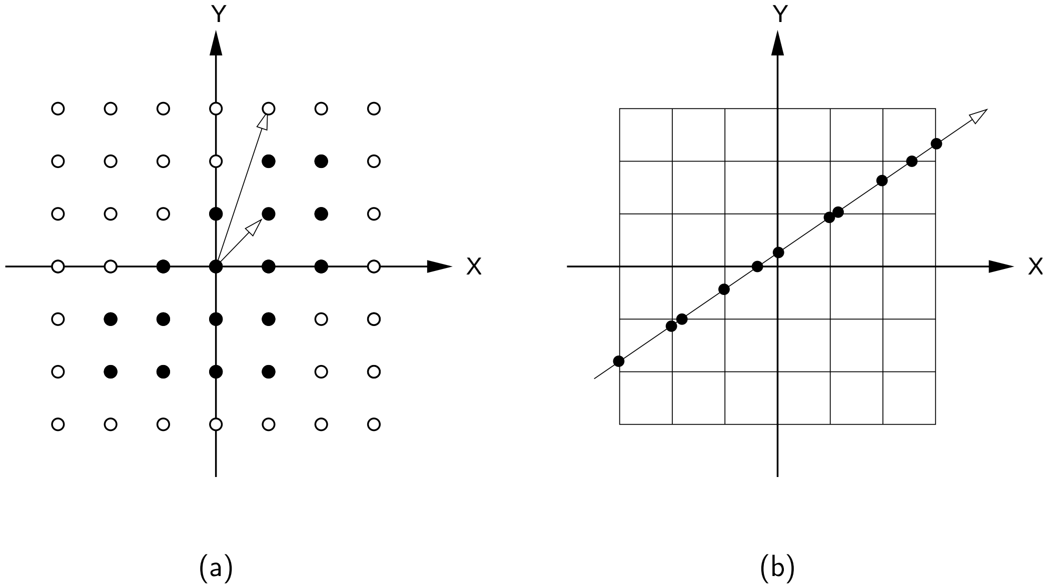


Figure 3.4: (a) Fully-discrete discretization scheme: Given two subsets S (black dots) and G (black and white dots) of the integer lattice $\mathbb{Z}^2$ with $S \subset G \subset \mathbb{Z}^2$. A projection is performed by counting the number of black dots along a line. Usually only lines with a rational slope and occupying at least one integer point in G, i.e. discrete lines, are permitted and it is assumed that points outside of G do not contribute to the projections. The reconstruction problem is then to recover the set S from the given projection data. (b) Image-based discretization scheme: The image domain is considered as tiled into squared areas (pixels) or mathematically Haar-basis functions. By this, the projection of a single ray corresponds to the integration over a piecewise constant function. This is depicted by the individual line-segments between each pair of succeeding black dots. As both schemes (a) and (b) are covered by the same algebraic formulation of the reconstruction problem our algorithms are independent of the underlying discretization.

only for the range, i.e. each pixel is associated with some unknown binary variable $x_i \in \{0, 1\}$. The continuous image domain is tiled into a set of Haar-basis functions, each amplified by the corresponding value x_i. Thus, the projection of a single ray corresponds to the integration along the path of the ray which is described by a piecewise constant function. Hence, the integral can be split into the sum of products $x_i \cdot a_i$ where a_i is the length of the ray's line-segment through pixel i. Denoting $a_i = 0$ if pixel i is not traversed at all, the projection of a single ray can be modelled more generally by $\mathbf{a}^\top \mathbf{x} = b$, where $\mathbf{x}$ is the vector over all binary variables. Stacking the equations of all rays together leads to a linear equation system and provides the following algebraic representation of the reconstruction problem,

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}\,, \quad \mathbf{x} \in \{0, 1\}^n. \tag{3.4}$$

Observe that the fully-discrete scheme also fits equation (3.4), i.e. matrix $\mathbf{A}$ is then the binary matrix with $a_{i,j} = 1$ if and only if ray i hits pixel j and $a_{i,j} = 0$ otherwise. The extension to three or more dimensions is obvious for both discretization schemes.

Throughout this work, we setup reconstruction problems using the parallelbeam geometry, section 2.2, but stress that other geometries are just as fine. The choice of geometry only affects the structure of matrix $\mathbf{A}$ in equation (3.4).

3.3 Optimization Approaches

We introduce three selected optimization strategies which are often used in order to solve discrete reconstruction problems: Simulated annealing, binary steering of non-binary iterative algorithms, and netflow approaches.

Simulated annealing is a non-deterministic sampling procedure frequently employed for the optimization of discrete or combinatorial problems. At this, it is quite flexible since it only demands the evaluation of the objective function, i.e. there is no need to evaluate the gradient or even the Hessian. On the opposite side, however, it known for its vast time consumption.

In the beginning of this chapter we demonstrated that the filtered backprojection algorithm is not adequate for discrete tomography, figure 3.1, since many projections and a scan over $180°$ are mandatory to obtain adequate results. Besides filtered backprojection, we also introduced iterative reconstruction techniques for continuous tomography in chapter 2 which utilize, if we neglect the binary constraint in equation (3.4) for the moment, the same algebraic formulation as discrete reconstruction algorithms. From this perspective, the question arises if these methods are extendible in a way that they additionally obey the binary constraint? The answer is yes, this has been done by Censor and Matej in [34] where a heuristic steering process is mounted on top of a non-binary iterative algorithm guiding the overall reconstruction process eventually to a binary solution.

The discrete reconstruction problem, equation (3.4), is $\mathbb{NP}$-complete for more than two projections but efficient algorithms exist in case of only two projections, i.e. the horizontal and the vertical projection or equivalently the row and the column sum. The key step of this approach is to formulate the problem as maximum flow problem on a netflow graph which provably reveals an integer solution.

3.3.1 Simulated Annealing (SA)

As the name implies, simulated annealing is inspired by a technique used in metallurgy for controlling the cool-down process of heated materials. At this, the goal is to increase the size of the crystals inside the material but preventing the appearance of defects in the crystalline structure at the same time. In the heated material atoms are free to wander randomly through states of higher energy and by cooling them down slowly, there is a chance that they will eventually find an arrangement at a low energy state.

Analogously, each step of the simulated annealing algorithm introduces small modifications in the current solution at random. Depending on the difference between the corresponding energy and an artificial temperature parameter T this random process does not only accept changes that decrease the energy but with a certain probability also changes that lead to an increase of the energy. As a consequence, simulated annealing can potentially escape local optima whereas local search algorithms might get trapped.

Simulated annealing is a generalization of a Monte Carlo method originally proposed by Metropolis et al [95] in order to search for equilibrium configurations of atoms at a given temperature level. Later a generalized version of the Metropolis' algorithm was published by Hastings [65], known as the Metropolis-Hastings algorithm, which is the foundation for the simulated annealing algorithm. The relation between the Metropolis-Hastings algorithm and mathematical minimization was first noted by Pincus [107]. However, it was Kirkpatrick et al [78] and Cerny [36] who both independently found that this forms the basis of a combinatorial optimization method. In computer vision, simulated annealing has been popularized by [56].

Markov Chains

We start with a preliminary excursion to Markov chains but confine ourselves to Markov chains defined on a finite state space $\mathcal{S}$ which is sufficient for all remaining considerations. A Markov chain (MC) is a sequence of random variables X_n, $n \in \mathbb{N}_0$, which can take on values from a finite state space $\mathcal{S}$. The random variables are equipped with the Markov property, that is

$$P(X_{n+1} = s \mid X_n = s_n, ..., X_0 = s_0) = P(X_{n+1} = s \mid X_n = s_n) \tag{3.5}$$

with $s, s_0, ..., s_n \in \mathcal{S}$. In other words this means that the probability of moving to a certain future state only depends on the present state and is, thus, independent from all past states. In case of a finite state space $\mathcal{S}$ the transition probability distribution can be represented by

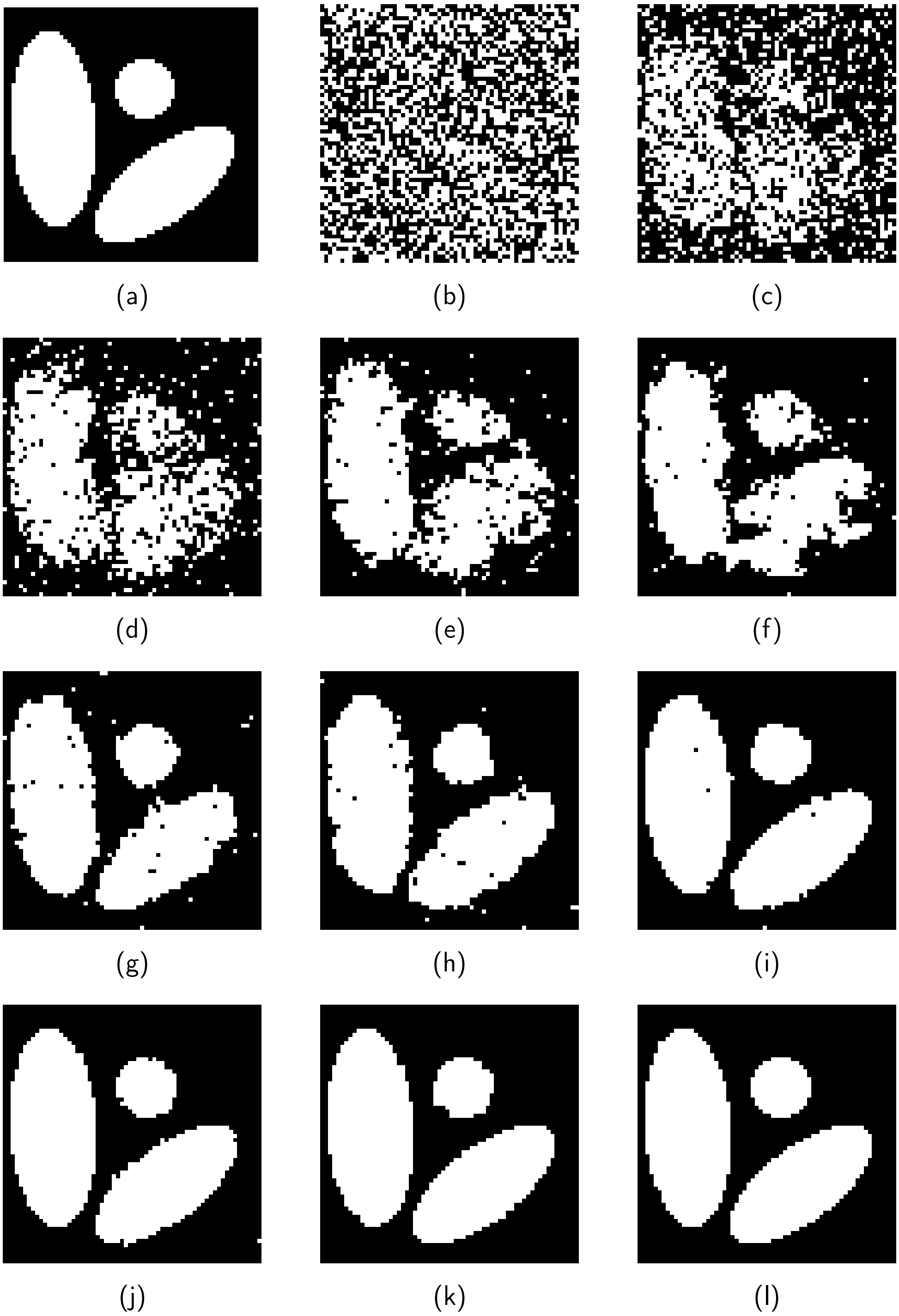


Figure 3.5: Image (a) shows the original image from which three projections, $0°$, $45°$, and $90°$ were taken. The remaining images (b)-(l) provide intermediate results obtained at decreasing temperature levels during the simulated annealing process. Though there are only three projections involved the reconstruction result (l) is much better than the results achieved with the filtered backprojection algorithm from many projections, cf. figure 3.1.

a matrix $\mathbf{P}$, called transition matrix, where the entries of $\mathbf{P}$ are defined by $p_{ij} := P(X_{n+1} = s_j \mid X_n = s_i)$. Consequently, $\mathbf{P}$ is a stochastic matrix, i.e. all rows of $\mathbf{P}$ sum to 1.

Important properties of Markov chains are irreducibility and periodicity . The first means that each state s is reachable from all other states in $\mathcal{S}$ otherwise the chain is reducible. The second property says that starting from a state s it is only possible to return to s in a periodic number of time steps, otherwise the Markov chain is considered aperiodic.

Starting from some initial distribution π_0 the stationary distribution π is a row vector satisfying

$$\pi = \pi\mathbf{P}, \tag{3.6}$$

which means that π is the eigenvector of $\mathbf{P}$ corresponding to the eigenvalue 1. In general neither existence nor uniqueness of a stationary distribution are guaranteed. However, in case of an irreducible and aperiodic Markov chain a unique stationary distribution exists. Additionally, the Perron-Frobenius theorem states convergence of $\mathbf{P}^k$ to a rank-one matrix where each row is the stationary distribution,

$$\lim_{k\to\infty} \mathbf{P}^k = \mathbf{1}\pi. \tag{3.7}$$

SA Approach

Let $\mathcal{S}$ be the finite set of feasible solutions and $c : \mathcal{S} \to \mathbb{R}_{\geq 0}$ be the cost function of a minimization problem. We further assume that there exists a neighborhood relation $\mathcal{N}$ such that $\mathcal{N}(\mathbf{x}) \subset \mathcal{S}$ contains all neighbors of $\mathbf{x} \in \mathcal{S}$ and $\mathcal{N}$ is symmetric, i.e. $\mathbf{y} \in \mathcal{N}(\mathbf{x})$ if and only if $\mathbf{x} \in \mathcal{N}(\mathbf{y})$. Assume that we are in state $\mathbf{x}$ the probability that we choose neighbor $\mathbf{y} \in \mathcal{N}(\mathbf{x})$ is covered by $P_\mathbf{x}(\mathbf{y})$ with $\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} P_\mathbf{x}(\mathbf{y}) = 1$. In order to move from state $\mathbf{x}$ we randomly select neighbor $\mathbf{y}$ with probability $P_\mathbf{x}(\mathbf{y})$. According to the Metropolis scheme, we compare $c(\mathbf{x})$ and $c(\mathbf{y})$ and move to state $\mathbf{y}$ if $c(\mathbf{y}) \leq c(\mathbf{x})$ holds. In case of $c(\mathbf{y}) > c(\mathbf{x})$ we accept the move with probability

$$\exp\left(-\frac{c(\mathbf{y}) - c(\mathbf{x})}{T}\right) \tag{3.8}$$

and reject otherwise. The parameter T introduces an artificial temperature controlling the acceptance of a move that increases the costs. If T is low such moves are unlikely to occur whereas for high temperatures $c(\mathbf{y}) - c(\mathbf{x})$ loses its influence on the probability, cf. equation (3.8), and suboptimal moves become more and more acceptable.

Suppose $\mathbf{X}_t$ is the state of the algorithm after t iterations, then $\mathbf{X}_t$ is a Markov chain with state space $\mathcal{S}$. The transition probabilities $P(\mathbf{X}_{t+1} = \mathbf{y} \mid \mathbf{X}_t = \mathbf{x})$ for $\mathbf{y} \in \mathcal{N}(\mathbf{x})$ are defined

Algorithm 3 Simulated Annealing (SA)

```
Require: $T_{\textsf{init}} > 0$ {Initial temperature}
Require: $0 < T_{\textsf{factor}} < 1$
Require: $N_{\textsf{attempts}} > 0$
Require: $N_{\textsf{accepts}} > 0$
Require: $0 < \min_{\textsf{efficiency}} \leq 1$
Require: $v_{\textsf{tolerance}} > 0$

  $\mathbf{x} :=$ Random($\{0,1\}^n$)
  $T := T_{\textsf{init}}$
  $History := \emptyset$
  $E := \|\mathbf{A}\mathbf{x} - \mathbf{b}\| + \alpha\, S(\mathbf{x})$
  $k := 1$
  $a := 0$ {Count accepted moves}

  repeat
    $p :=$ Random($\{0, ..., n-1\}$)
    $l :=$ Random($\mathcal{L}$)
    $\tilde{\mathbf{x}} := \mathbf{x}$
    $\tilde{\mathbf{x}}[p] := l$
    $E_{new} := \|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}\| + \alpha\, S(\tilde{\mathbf{x}})$

    {Accept move if ...}
    if $E_{new} < E$ or $\exp(-(E_{new} - E)/T) >$ Random($[0,1]$) then
      $\mathbf{x} := \tilde{\mathbf{x}}$
      $E := E_{new}$
      $History := History \cup \{E_{new}\}$
      $a := a + 1$
    end if

    {Adjust temperature}
    ... see algorithm 4 ...

    {Evaluate efficiency}
    if $k == N_{\textsf{attempts}}$ then
      $c_{\textsf{efficiency}} := a / N_{\textsf{attempts}}$
      $a := 0$
      $k := 1$
    else
      $k := k + 1$
    end if

  until $c_{\textsf{efficiency}} < \min_{\textsf{efficiency}}$
```

Algorithm 4 Simulated Annealing (SA): Adjust temperature

```
if $|History| == N_{accepts}$ then
  $v_{new}$ := Compute variance of $History$
  if $v < v_{new}$ or $|v - v_{new}| < v_{\mathsf{tolerance}}$ then
    $T = T \cdot T_{factor}$
  end if
  $v := v_{new}$
  $History := \emptyset$
end if
```

by

$$P(\mathbf{X}_{t+1} = \mathbf{y} \mid \mathbf{X}_t = \mathbf{x}) := \begin{cases} P_{\mathbf{x}}(\mathbf{y}) \exp\left(-c(\mathbf{y}) - c(\mathbf{x})/T\right) & \text{, if } c(\mathbf{y}) > c(\mathbf{x}) \\ P_{\mathbf{x}}(\mathbf{y}) & \text{, otherwise.} \end{cases} \tag{3.9}$$

Further, we define

$$\pi(\mathbf{x}) := \frac{\exp\left(-c(\mathbf{x})/T\right)}{\sum_{\mathbf{z}\in S} \exp\left(-c(\mathbf{z})/T\right)} \tag{3.10}$$

which is a probability distribution since $\pi(\mathbf{x}) \geq 0$ and $\sum_{x\in S} \pi(\mathbf{x}) = 1$.

Theorem 1 ([13]). *Suppose the Markov chain $\mathbf{X}_t$ is irreducible and that $P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{y}}(\mathbf{x})$ for every $\mathbf{x}$ and every $\mathbf{y}$ in $\mathcal{N}(\mathbf{x})$. Then, the vector with components $\pi(\mathbf{x})$, $\mathbf{x} \in \mathcal{S}$, defined according to equation (3.10) is the unique steady state distribution of the Markov chain $\mathbf{X}_t$.*

Consequently, the probability that state $\mathbf{X}_t$ equals a certain state $\mathbf{x}$ is approximately the steady state probability of $\pi(\mathbf{x})$ provided that one iterates for sufficiently long time t. As $\pi(\mathbf{x})$ falls exponentially with $c(\mathbf{x})$ and for $T \to 0$, cf. equation (3.10), almost all the steady state probability is concentrated on states at which $c(\mathbf{x})$ is globally minimized.

Reconstruction Algorithm

Consider the binary reconstruction problem as defined in equation (3.4) and let $X := \{0,1\}^n$ denote the space of all binary images. An adequate cost function on X is defined by $E(\mathbf{x}) := ||\mathbf{Ax} - \mathbf{b}||^2 + \alpha\ S(\mathbf{x})$ where $S(\mathbf{x})$ denotes an prior function with regularization parameter α, in our SA experiments

$$S(\mathbf{x}) := \sum_{\langle i,j\rangle} (x_i - x_j)^2 = \sum_i \sum_{j\in\mathcal{N}(i)} (x_i - x_j)^2. \tag{3.11}$$

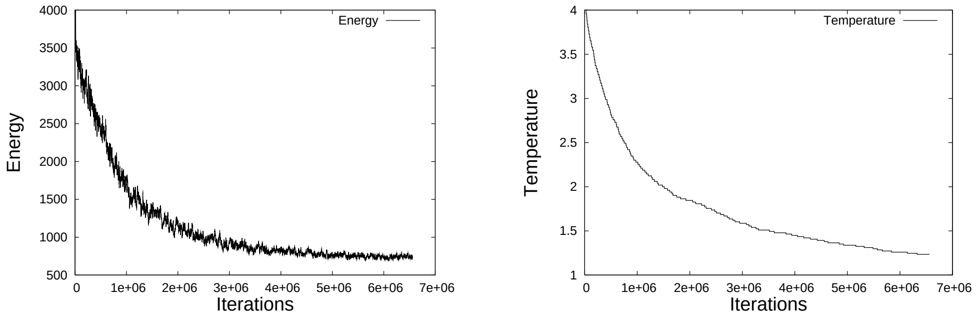


Figure 3.6: Referring to the discrete reconstruction experiment, figure 3.5, the left figure shows the behavior of the energy function while the right figure plots the temperature T correspondingly.

An implementation of the SA algorithm according to our previous discussion is shown in algorithm 3. In order to demonstrate the SA algorithm and to exemplify the benefits of discrete reconstruction algorithms we reconstructed again the phantom image previously used in connection with the filtered backprojection algorithm, figure 3.1. This time, however, we took only three projections equally spaced over 90°. Besides intermediate results at various temperature levels, the final result of this experiment is shown in figure 3.5 (l) which almost revealed the original image. This experiment shows that despite a small number of projections which were taken only over 90° it is, nonetheless, possible to achieve high quality reconstructions.

3.3.2 Binary Steering of Non-binary Iterative Algorithms

In case of iterative reconstruction methods, section 2.3.5, the algebraic formulation of the continuous and the discrete reconstruction problem, section 3.2, differ only in the binary constraint, equation (3.4). Therefore, it is tempting to apply iterative techniques which have been proven to be useful for the continuous case also in the discrete case. Before, however, modifications are necessary in order to enforce the binary constraint, $\mathbf{x} \in \{0,1\}^n$.

Censor and Matej [34] proposed a heuristic procedure which is mounted on top of an iterative non-binary reconstruction algorithm and steers the solution vector $\mathbf{x}$ towards $\mathbf{x} \in \{0,1\}^n$. Their method consists of two additional steps, the binarizer and the conflict settler, which are attached to the original algorithm as outlined in figure 3.8, compare figure 3.8 also to figure 2.11 where the outline of a general iterative reconstruction algorithm is shown. The algorithm uses three sequences, α_k, β_k, and t_k, of real numbers obeying $0 \leq \alpha_k < t_k$, $\alpha_k < \alpha_{k+1}$, $t_k < \beta_k \leq 1$, and $\beta_{k+1} < \beta_k$. A single iteration k of the modified algorithm starts with the binarizer which

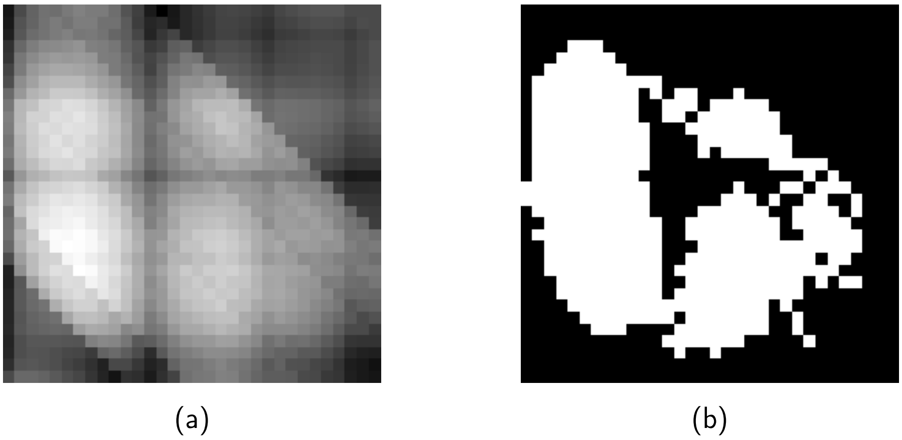


Figure 3.7: Reconstruction results obtained by ART, see algorithm 2, (a) without and (b) with binary steering. The binary reconstruction problem is similar to the one previously used for simulated annealing, figure 3.5, only the image size has been reduced.

according to

$$\tilde{x}_j^k := \begin{cases} 0, & \text{if} \quad x_j^k \leq \alpha_k \\ 1, & \text{if} \quad x_j^k \geq \beta_k \\ x_j^k, & \text{otherwise} \end{cases} \tag{3.12}$$

partially binarizes the solution $\mathbf{x}^k$ of the previous iteration. As the algorithm proceeds the α_k values increase and the β_k values decrease, thus, more and more components of $\mathbf{x}^k$ become binary. The next two steps, corrector generator and correction applicator, depend on the iterative algorithm employed for the reconstruction process, cf. figure 2.11. A conflict has occurred if a component y_j^k is larger than or equal to the threshold t_k while its corresponding component x_j^k was below α_k. Similarly, a conflict occurs if y_j^k is less than or equal to t_k but x_j^k was greater than β_k. In both cases no decision concerning this component is made and instead the conflict is settled by setting

$$x_j^{k+1} := \begin{cases} t_k - \epsilon, & \text{if} \quad x_j^k \leq \alpha_k \quad \text{and} \quad y_j^k \geq t_k \\ t_k + \epsilon, & \text{if} \quad x_j^k \geq \beta_k \quad \text{and} \quad y_j^k \leq t_k \\ y, & \text{otherwise} \end{cases} \tag{3.13}$$

In order to demonstrate the influence of the steering process we applied the ART method as described in section 2.3.5 with and without binary steering to a similar reconstruction experiment as previously used for simulated annealing, figure 3.5. As can be seen in figure 3.8 (a) the pure ART solution contains many fractional values and, thus, makes it hard to conclude which binary objects were present in the true image. On the other the side, figure 3.8 (b), the solution obtained with the binary steering process is much better than the pure ART solution. However, even with binary steering it cannot compare to the solution obtained by simulated annealing, figure 3.5 (l). The reason for this discrepancy between ART with binary steering and SA emerges from the spatial prior included in the SA algorithm. Concerning iterative methods it is difficult to additionally supplement them with a spatial smoothness prior which is a major drawback in view of images mainly consisting of homogeneous regions. Additionally, the steering process is a purely heuristic step with no deeper mathematical justification.

We kindly refer the more interested reader to [11] which includes a detailed evaluation of the binary steering process for discrete tomography and considers other iterative reconstructions methods, like component averaging (CAV) [35] which accommodates to the sparseness of matrix $\mathbf{A}$, equation (3.4).

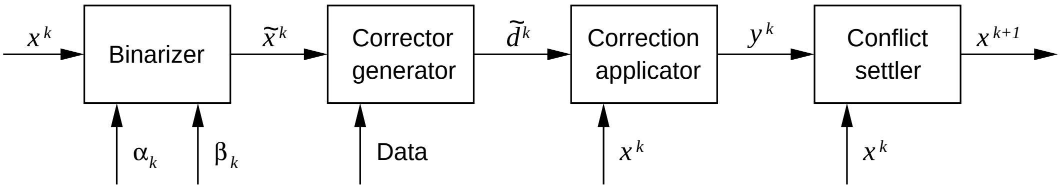


Figure 3.8: Outline of the overall steering process: The inner steps, corrector generator and correction applicator, are steps from the original non-binary algorithm, figure 2.11. The outer steps, binarizer and conflict settler, are introduced by the binary steering process.

3.3.3 Network Flow

The discrete reconstruction problem, equation (3.4), can be efficiently solved for two projections and is known to be $\mathbb{NP}$-complete otherwise. We study the case where the horizontal and the vertical projection are given and show how the reconstruction problem can be solved efficiently as a maximum-flow problem.

A flow network is a directed graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. There exist two distinguished vertices $s, t \in V$ called the source and the sink respectively. For convenience it is typically assumed that each vertex is on a path from the source to the sink and, thus, the graph is connected. For each edge $(u, v) \in E$ a capacity function $c(u, v) \geq 0$ assigns the maximal flow that can be passed through this edge. A flow, forwarded from the source towards the sink, is send over the network where the amount of flow is restricted by the individual edge capacities.

The net flow is a function $f : V \times V \to \mathbb{R}$ that provides the flow at each edge of G with $f(u,v) := 0$ if $(u,v) \notin E$. At this, f must not exceed the capacities of the edges,

$$f(u,v) \leq c(u,v), \tag{3.14}$$

and the flow from vertex u to vertex v is supposed to be the negative in the opposite direction,

$$f(u,v) = -f(v,u). \tag{3.15}$$

Further, except for the source s and the sink t, all flow that enters a vertex $u \in V - \{s,t\}$ must also leave u such that no flow remains in the vertices,

$$\sum_{v \in V} f(u,v) = 0. \tag{3.16}$$

As a consequence, the total flow emitted by the source,

$$|f| := \sum_{v \in V} f(s,v), \tag{3.17}$$

must finally enter the sink.

The residual network G_f of G induced by the flow f is defined by $G_f := (V, E_f)$ with the residual capacity

$$c_f(u,v) := c(u,v) - f(u,v),\ \forall (u,v) \in E, \tag{3.18}$$

and

$$E_f := \{(u,v) \in V \times V : c_f(u,v) > 0\}. \tag{3.19}$$

Max-Flow

We assume here that the horizontal and vertical projection are given but emphasize that the procedure is not necessarily restricted to these projections. Further, we consider in the following the small, 3×3, reconstruction problem provided in figure 3.9 with unknowns $x_i \in \{0,1\}$,$i \in \{1,...,9\}$, where the extension to larger problems is straightforward.

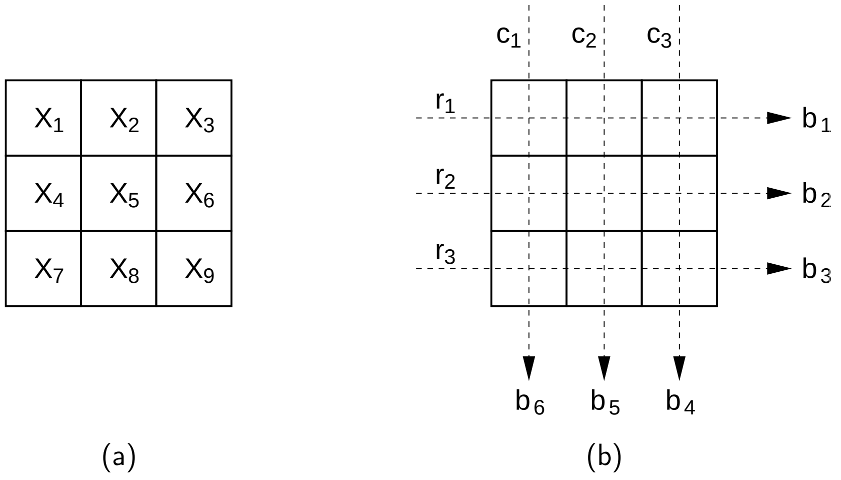


Figure 3.9: For illustration purposes we consider the following small 3×3 image, shown in figure (a), where each pixel is associated with an unknown variable $x_i \in \{0, 1\}$. Figure (b) depicts the imaging process, the horizontal and the vertical projection are taken as described in section 3.2. Hence, the projections equal the row and the column sums.

As described in section 3.2 the corresponding algebraic formulation leads to the following linear equation system,

$$\begin{pmatrix} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{pmatrix}. \tag{3.20}$$

Let $G := (V, E)$ be an undirected graph then G is called bipartite if and only if there exists a partition of $V = S \cup T$, $S \cap T = \emptyset$, such that for all $v_1, v_2 \in V$, $\{v_1, v_2\} \in E$, $v_1 \in S$ and $v_2 \in T$ or vice versa. The incidence matrix $\mathbf{A}_{inc}$ of G is defined by

$$(a_{v,e})_{v \in V, e \in E} := \begin{cases} 1 & \text{if } v \in e \\ \\ 0 & \text{if } v \notin e \end{cases}.$$

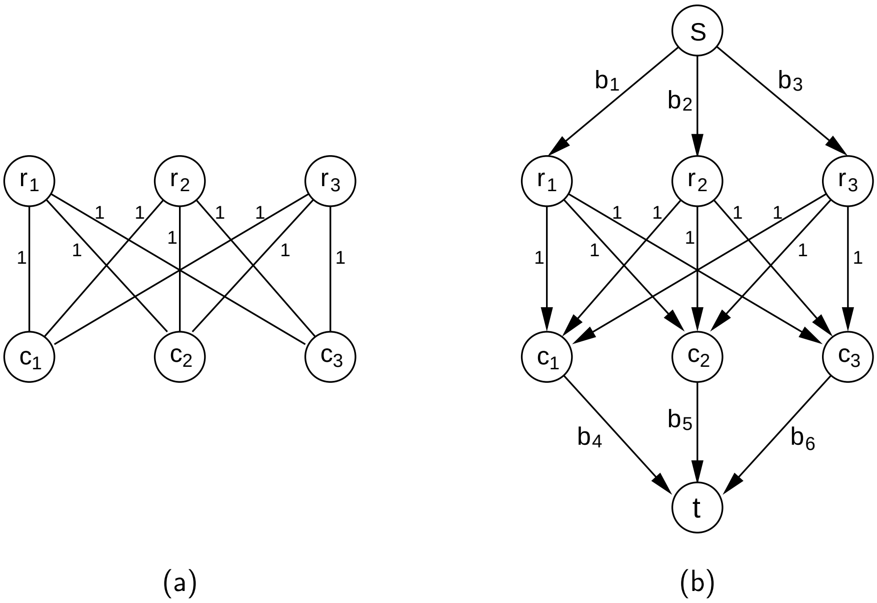


Figure 3.10: (a) Bipartite graph $G := (V, E)$ that represents the imaging process which is shown in figure 3.9 (b). The incidence matrix of G is the same as the projection matrix in equation (3.20). (b) The bipartite graph from the left side is embedded into a flow network in order to solve the reconstruction problem, figure 3.9(b).

With $S := \{r_1, r_2, r_3\}$ and $T := \{c_1, c_2, c_3\}$ the imaging process, figure 3.9, is represented as a bipartite graph, figure 3.10(a), and the corresponding incidence matrix is just the projection matrix shown in equation (3.20). Each edge of the bipartite graph corresponds to an unknown x_i. In order to actually solve the reconstruction problem the bipartite graph is embedded into the flow network, as shown in figure 3.10 (b), and the maximal amount of flow transported from the source to the sink yields a solution of the reconstruction problem. This problem can efficiently be solved for instance by the Ford-Fulkerson method [44] which iteratively seeks the residual network for a path in order to push more flow from the source to the sink. If such an augmenting path exists the residual network is adapted along this path and otherwise the algorithm stops.

Theorem 2 (Integrality theorem [44]). *If the capacity function c takes on only integral values, then the maximum flow f produced by the Ford-Fulkerson method has the property that $|f|$ is integer-valued. Moreover, for all vertices u and v, the value of $f(u, v)$ is an integer.*

Note that according to theorem 2 the integrality of the maxflow solution depends also on the optimization method. This becomes clear since we shall soon see, equation (3.23), that the maximum flow problem can be converted into a linear program with integer solutions at the vertices. If the solution of the problem is not unique there also exists non-binary solutions lying within the convex set spanned by the optimal binary vertices. It now depends on the optimization method itself which solution it reveals. See in this context also figure 4.3 in chapter 4 where we compare an interior point optimizer to the well-known simplex algorithm. Regarding the maximum flow problem, there meanwhile exist faster methods than the Ford-Fulkerson algorithm, like preflow-push algorithms [44], which, however, must not necessarily preserve integrality.

Binary reconstruction approaches based on network flows have been considered in various papers [116, 8, 9, 7]. The basic network flow approach is limited to two projections which is insufficient even for object of moderate complexity. Therefore, efforts have been made in order to overcome this systematic drawback by combining reconstruction results obtained from pairs of projections. In [116] a heuristic based on an additional network is used merging pairwise reconstructions. Another heuristic is proposed by Batenburg [8, 9, 7] which additionally incorporates spatial smoothness while merging different reconstructions. We compare and comment our results in chapter 4 to the results reported in [7].

In view of the reconstruction approach that will be introduced in the next chapter it is worth taking a look at network flows from a slightly different perspective. Therefore, we define the constraint matrix $\mathbf{C}$ and the constraint vector $\mathbf{d}$ of the bipartite graph G,

$$\mathbf{C} := \begin{pmatrix} \mathbf{A}_{inc} \\ -\mathbf{I} \\ \mathbf{I} \end{pmatrix} \quad \text{and} \quad \mathbf{d} := \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix}, \tag{3.21}$$

and consider the polytope

$$\tilde{P} := \{\mathbf{x} \mid \mathbf{C}\,\mathbf{x} \leq \mathbf{d}\}. \tag{3.22}$$

A matrix is said to be totally unimodular if the determinant of every square submatrix is either -1, 0, or 1. The following theorem states the total unimodularity of matrix $\mathbf{C}$, equation (3.21),

Theorem 3 ([120]). *A graph $G := (V, E)$ is bipartite if and only if its constraint matrix is totally unimodular (see appendix C for a proof).*

Since the right hand side vector $\mathbf{d}$, equation (3.21), is integral the following theorem holds,

Theorem 4 ([120]). *Let matrix $\mathbf{A}$ be totally unimodular and $\mathbf{b}$ be an integer vector, then the vertices of the polytope*

$$P := \{\mathbf{x} \mid \mathbf{A}\,\mathbf{x} \leq \mathbf{b}\}$$

are all integer vectors (see appendix C for a proof).

By this, the maximum-flow approach for two projections can be equivalently written as the following maximization problem

$$\max_{\mathbf{x}} \{\mathbf{1}^\top \mathbf{x} \mid \mathbf{x} \in \tilde{P}\}. \tag{3.23}$$

Equation (3.23) motivates the use of linear programming in connection with binary reconstruction problems and we take on this in the next chapter.

Graph Cuts

Graph cuts [60] are global optimization techniques for binary functions, equation (3.24), based on netflows. Therefore, the energy function is represented as flow network and a cut of the graph yields the minimal energy.

$$E(\mathbf{x}), \quad \mathbf{x} \in \{0,1\}^n. \tag{3.24}$$

Let $G(V, E)$ be a flow network, a cut (S, T) is a partition of V into S and $T = V - S$ such that $s \in S$ and $t \in T$. Further, if f is a flow in G then the flow between S and T is defined as $f(S, T)$ and respectively the capacity $c(S, T)$. One of the fundamental results in combinatorial optimization is the max-flow min-cut theorem 5 which connects the maximal flow problem from the previous section with the minimal cut of a flow network.

Theorem 5. *[Max-flow min-cut theorem [44]] If f is a flow in a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:*

(i) f is a maximum flow in G.

(ii) The residual network G_f contains no augmenting paths.

(iii) $|f| = c(S, T)$ for some cut (S, T) of G.

Computer vision algorithms based on graph cuts have been suggested and successfully applied in [19, 23, 22, 82]. In the typical outline of a binary image labeling problem all pixels are associated with nodes which are connected according to some neighborhood structure $\mathcal{N}$. Additionally, all these nodes are connected to the source and sink which correspond to the binary labels and weights reflecting the individual terms of the energy function are assigned to the edges. Figure 3.11 depicts the overall graph structure. Due to the structure of the graph a cut implies a one to one correspondence with a binary image, $\mathbf{x} \in \{0,1\}^n$. A pixel is assigned 1, respectively

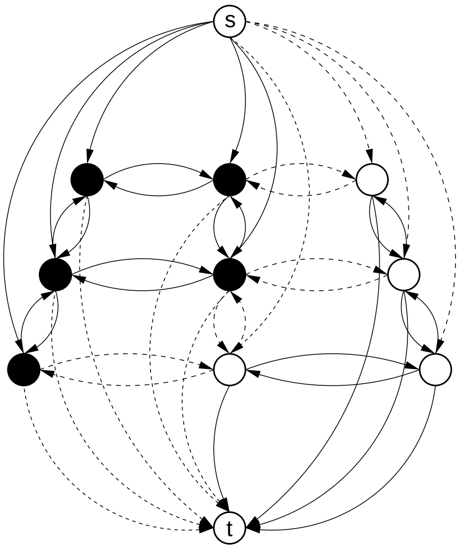


Figure 3.11: Graph cuts flow graph: The pixels correspond to the intermediate nodes which are internally connected according to the neighborhood structure induced by the image. Additionally, the intermediate nodes are connected to the source s and sink t. Dotted edges denote edges that belong to the cut. If we assume the source labels a pixels 0 and the sink 1 then a pixel receives label 1 if it is connected to the sink and the edge to the source is in the cut.

0, if the edge between its node and s respectively t is in the cut. One of both edges must be in the cut since a connection between the source and the sink would exist otherwise. Further, it is not possible that both edges are in the cut since then the cut cannot be minimal anymore.

Kolmogorov and Zabih [82] investigated the types of energy functions that can effectively minimized by graph cuts. In particular, they consider the class that includes all binary functions of type

$$E_2(\mathbf{x}) := \sum_i \mathcal{E}_i(x_i) + \sum_{i<j} \mathcal{E}_{i,j}(x_i, x_j), \tag{3.25}$$

where $\mathcal{E}_i(\cdot)$ and $\mathcal{E}_{i,j}(\cdot,\cdot)$ are arbitrary terms depending on a single variable or two variables respectively. Their main result, concerning equation (3.25), shows that graph cuts are applicable if each term $\mathcal{E}_{i,j}$ obeys

$$\mathcal{E}_{i,j}(0,0) + \mathcal{E}_{i,j}(1,1) \leq \mathcal{E}_{i,j}(0,1) + \mathcal{E}_{i,j}(1,0). \tag{3.26}$$

In [82] this condition is referred to as regularity and is equivalent to submodularity. The presence of non-regular terms imposes negative edge weights which in turn prohibits the use of maxflow approaches.

Concerning the tomographic reconstruction problem, equation (3.4), in terms of a quadratic data term

$$E(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\,\mathbf{b}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{b}. \tag{3.27}$$

Matrix $\mathbf{Q} := \mathbf{A}^\top \mathbf{A}$ contains only non-negative entries since matrix $\mathbf{A}$ is non-negative, see section 3.2. The positive off-diagonal entries of $\mathbf{Q}$, $\mathcal{E}_{i,j}(x_i, x_j) := q_{i,j} x_i x_j$, $0 < q_{i,j}$, $i \neq j$, lead to $0 + q_{i,j} \nleq 0 + 0$ and, thus, to the non-regularity of $E(\mathbf{x})$, compare equation (3.26). For this reason, it is not possible to apply graph cuts directly which otherwise would contradict the $\mathbb{NP}$-completeness of the reconstruction problem anyway.

In order to optimize non-regular functions with graph cuts at least approximately different approaches have been proposed in [113], [110] and more recently in [81]. We will focus on [81] here since it seems to be the most promising approach.The key step is to introduce additional binary variables y_i for all x_i and to transform the original energy function $E(\mathbf{x})$, equation (3.27), into

$$\begin{aligned}\tilde{E}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\Big(&\sum_i \mathcal{E}_i(x_i) + \sum_i \mathcal{E}_i(1 - y_i) + \\ &\sum_{i,j} \mathcal{E}_{i,j}(x_i, x_j) + \sum_{i,j} \mathcal{E}_{i,j}(y_i, y_j) + \\ &\sum_{i,j} \mathcal{E}_{i,j}(x_i, 1 - y_j) + \sum_{i,j} \mathcal{E}_{i,j}(1 - y_i, x_j)\Big).\end{aligned} \tag{3.28}$$

If $x_i = 1 - y_i$ holds then the pair (x_i, y_i) is called *consistent* and *inconsistent* otherwise. It is clear that the minimum of $E(\mathbf{x})$, equation (3.27), and $\tilde{E}(\mathbf{x}, \mathbf{y})$, equation (3.28), coincides if all variables i are consistent. However, it is not possible to enforce consistency as this would introduce non-regular terms again. Therefore, depending on the structure of the optimization problem there is hope that inconsistency affects only a small number of variables. In this context, it has been proven [17] that if (x_i, y_i) is consistent it is also optimal, this property is called *persistence*. Consequently, the number of inconsistent variables gives some hint about the distance between the approximate solution and the optimal solution. Unfortunately, the approximation $\tilde{E}(\mathbf{x}, \mathbf{y})$, equation (3.28), can achieve much smaller energies than $E(\mathbf{x}^*)$ at the global optimal solution $\mathbf{x}^*$. We will demonstrate that this leads to problems when energy, equation (3.28), is applied to discrete reconstruction problems. Therefore, we reconsider the imaging geometry as shown in figure 3.9, i.e. we are given the row and column sum, and assume that the projections are taken from the following binary image:

$$\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \tag{3.29}$$

Clearly, this reconstruction problem is trivial and its solution is unique. However, if we formulate the problem according to equation (3.28) and seek for all its global optimal solutions by brute force computation we obtain three different optima. The top row images, equation (3.30), show the original variables $\mathbf{x}$ and the bottom row images, equation (3.31), the corresponding complementary variables $\mathbf{y}$. The approximation is not unique anymore, all three solutions are equally good with respect to equation (3.28) but only the solution in the middle column fulfills the projection constraints of the original problem.

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \tag{3.30}$$

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 0 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \tag{3.31}$$

We also performed larger experiments using again the reconstruction problem from figure 3.5. The results are shown in the figure 3.12 where the top row shows, as previously, the original binary variables $\mathbf{x}$ and the bottom row the complementary variables $\mathbf{y}$. In contrast to the smaller experiment the computation of all global optimal solutions is prohibitive due to the size of the problem. However, as can be clearly seen the results are far away from the true image and the energy of the reconstructed images is significantly below the energy of the groundtruth image, table 3.1. Therefore, we must conclude that the introduction of complementary variables does not make graphcuts applicable for the reconstruction of binary images.

3.4 Error Measures for Binary Images

We introduce different error measurements in this section which will be used in the following chapters for the evaluation our algorithms. At this, there are two aspects that we are interested in: first, we want the reconstructions to be in good agreement with the projections, i.e.

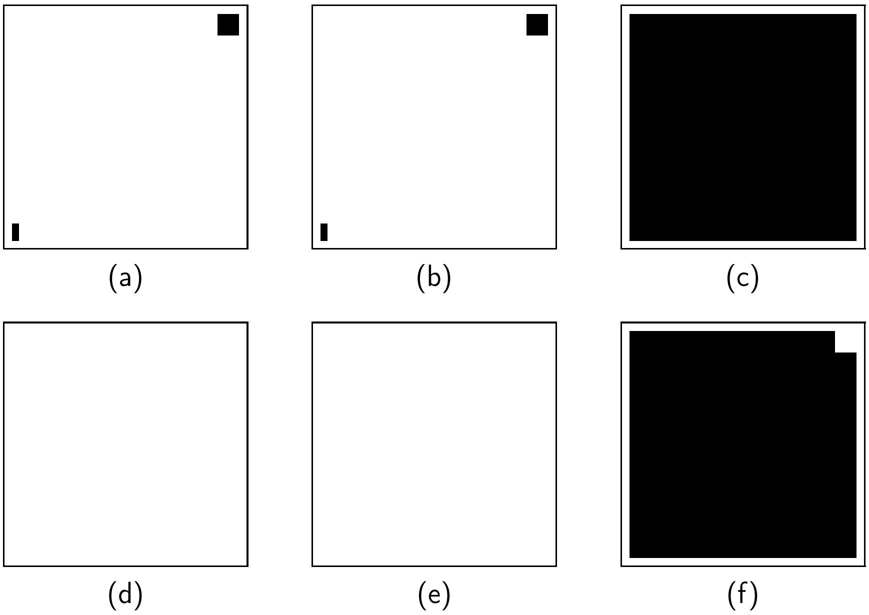


Figure 3.12: Reconstruction results: The reconstruction problem shown and described in figure 3.5 was formulated with the approximate energy, equation (3.28), in order to overcome non-regular terms and optimized via graph cuts. As can be seen, the reconstruction is far away from the true image which shows three ellipses. However, the energy of the reconstructed images is significantly below the energy of the groundtruth image due to the construction of the approximate energy.

$\mathbf{A}\ \mathbf{x} = \mathbf{b}$, and, second, we want them to be close to the groundtruth images used for problem generation. Due to the ill-posedness of reconstruction problems there might, however, exist different images which perfectly agree with the projection constraints. Our measurements give considerations to both facets.

Let $\mathbf{x}^*$ be the groundtruth image from which the reconstruction problem, $\mathbf{A}\ \mathbf{x}^* = \mathbf{b}$, has been setup as described in section 3.2. Assume, further, that $\mathbf{x} \in \{0,1\}^n$ is a binary solution to the problem obtained by some optimization algorithm. To analyze the quality of $\mathbf{x}$ with respect to the reconstruction problem and the groundtruth $\mathbf{x}^*$ we employ the following error measurements,

$$Err_1(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \ , \tag{3.32}$$

$$Err_2(\mathbf{x}) := \frac{1}{\sum_{i=1}^{n} x_i} \|\mathbf{x} - \mathbf{x}^*\|_1 \ . \tag{3.33}$$

Additionally to the error measures, Err_1 and Err_2, we comprise two error measurements commonly applied in object matching for the comparison of binary images.

smoothness	energy reconstruction	energy groundtruth
0	-157796	0.015625
1	-157777	656.016
5	-157716	3280.02

Table 3.1: Comparison of the energy values: We applied graph cuts to the reconstruction problem shown and described in figure 3.5 using the approximate energy function, equation (3.28). Comparing the energy of the optimal graph cut solution (middle column) to the groundtruth solution (right column) shows that the energy of the global optimum is significantly below the energy of the groundtruth image. However, the reconstruction is quite different from the original image, figure 3.12.

In many computer vision tasks it is necessary to find a certain object in an arbitrary image, in order to detect pedestrians in a street scene for instance [54, 55]. At this, it is popular to represent the object edges and the image edges as binary images and to compare both at several positions [16, 71]. The object is found if both, object edges and image edges, match at this position. So far, the procedure would lead to poor results since it relies on the fact that the edges must be perfectly aligned. For this reason, a distance transform is firstly applied to the object edges, then for any point in the edge image the distance to its closest object point is known. Formally, let P denote the set of object points the distance for an arbitrary point x is defined by

$$\mathcal{D}_P(x) := \min_{y \in P} \|x - y\|. \tag{3.34}$$

The distance transform of an binary image A provides a grayvalued image with $\mathcal{D}_p(x)$ for each pixel $x \in A$. We denote this briefly by $\mathcal{D}(A)$ and provide an example in figure 3.13. Based on the distance transform, the Chamfer [5, 16] and the Hausdorff [2, 3, 71] distance are frequently applied for image comparison.

The Hausdorff distance is defined [71] as

$$H(A, B) := \max\{h(A, B), h(B, A)\} \tag{3.35}$$

with

$$h(A, B) := \max_{x \in A} \min_{y \in B} \|x - y\| = \max_{x \in A} \mathcal{D}_B(x). \tag{3.36}$$

The function $h(A, B)$ is called direct Hausdorff distance and measures the distance from a point x that is farthest away from any point in B. While $h(A, B)$ is non-symmetric in general, i.e. $h(A, B) \neq h(B, A)$, the Hausdorff distance $H(A, B)$ is symmetric again.

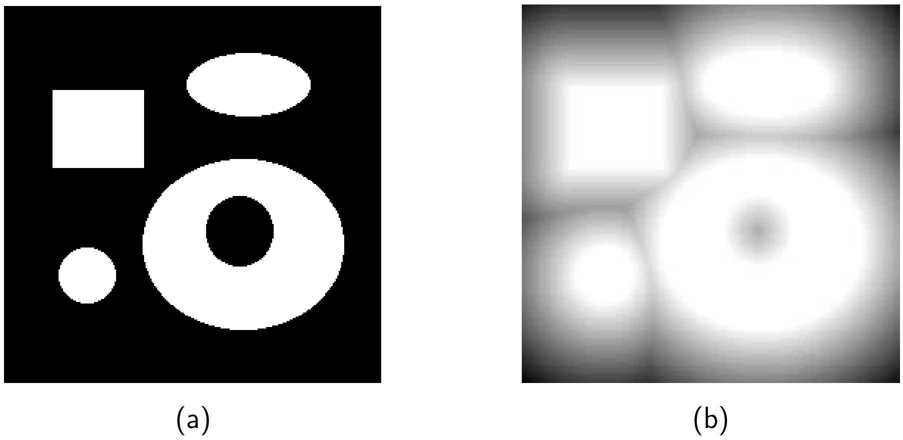


Figure 3.13: Distance transform: Image (a) provides a binary test image with black pixels corresponding to background and white to object respectively. The distance transform of image (a) is shown in (b); darker grayvalues indicate greater distances from the objects.

In contrast to the Hausdorff distance which aims at the worst mismatched point the Chamfer distance overlays the binary image and the distance transform and averages the distances of all edge pixels. In [16] the root mean square has been proposed for averaging

$$c(A,B) := \frac{1}{3}\sqrt{\frac{\sum_{i,j}(A_{i,j}\cdot \mathcal{D}(B)_{i,j})^2}{\sum_{i,j} A_{i,j}}}. \tag{3.37}$$

which is non-symmetric, analogous to $h(A,B)$. The symmetric Chamfer distance is defined by

$$C(A,B) := \frac{c(A,B)+c(B,A)}{2}. \tag{3.38}$$

According to the definition above, we associate the Hausdorff distance, equation (3.35), and the Chamfer distance, equation (3.38), with the following error measures

$$Err_3(\mathbf{x},\mathbf{x}^*) := H(\mathbf{x},\mathbf{x}^*) \tag{3.39}$$

$$Err_4(\mathbf{x},\mathbf{x}^*) := C(\mathbf{x},\mathbf{x}^*). \tag{3.40}$$

4 Linear Optimization

Due to the inherent combinatorial complexity, the binary reconstruction problem, equation (3.4), belongs to the class of $\mathbb{NP}$-complete problems, section 1.2, and is, therefore, hard to optimize. In this situation, it is often beneficial to relax the constraints of the original problem such that the resulting problem is less difficult. If the relaxed problem does not differ too much there is hope that a solution of the relaxation is also a good approximation to the original problem.

Since we consider the case of binary tomography in this chapter a natural relaxation is to replace the constraint $\mathbf{x} \in \{0,1\}^n$ by $\mathbf{x} \in [0,1]^n$. This allows a formulation of the relaxed problem in terms of linear programming which in turn can be globally optimized, even for large instances, by modern interior point solvers. Within this context, we additionally introduce a smoothness prior which favors spatially homogeneous reconstructions and consider different rounding strategies which must be applied to eventually yield a binary solution.

4.1 Linear Programming (LP)

Although there exist different definitions for linear programming problems the following is usually referred to as the standard form of a linear program,

$$\min\ \mathbf{c}^\top \mathbf{x} \quad \text{subject to} \quad \mathbf{A}\,\mathbf{x} = \mathbf{b}\ ,\ \mathbf{x} \geq \mathbf{0}, \tag{4.1}$$

where $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{A}$ is a $\mathbb{R}^{m \times n}$ matrix. Other definitions involve maximization instead of minimization, inequality constraints, or allow negative components in $\mathbf{x}$. However, it is not difficult to see that all these formulations are essentially equal. By changing the sign of $\mathbf{c}$, the minimization problem converts into a maximization problem, $\min \mathbf{c}^\top \mathbf{x} = \max -\mathbf{c}^\top \mathbf{x}$. Furthermore, given that some or all rows $\mathbf{a}_i^\top$ of matrix $\mathbf{A}$ are associated with inequalities, $\mathbf{a}_i^\top \mathbf{x} \leq b_i$ or $\mathbf{a}_i^\top \mathbf{x} \geq b_i$, then, through the introduction of slack variables λ_i, each inequality is replaced by $\mathbf{a}_i^\top \mathbf{x} + \lambda_i = b_i$, $\lambda_i \geq 0$, or respectively $\mathbf{a}_i^\top \mathbf{x} - \lambda_i = b_i$, $\lambda_i \geq 0$, which again yields a linear program in standard form. A similar trick is used if some or all components of $\mathbf{x}$ are free, i.e. $x_i \in \mathbb{R}$ instead of $x_i \geq 0$, then each free variable x_i is replaced by $\lambda_1 - \lambda_2$ with $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$.

If a vector $\mathbf{x}$ satisfies the constraints $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq 0$, it is called a feasible point and the set of all feasible points defines the feasible set. Let S be the feasible set of the linear program $\mathcal{P}$ then there are three possible cases concerning the solvability of $\mathcal{P}$:

- $S = \emptyset$: $\mathcal{P}$ is not feasible.
- $S \notin \emptyset$, $\nexists \min\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{x} \in S\}$: $\mathcal{P}$ is feasible but unbounded.
- $S \notin \emptyset$, $\exists \min\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{x} \in S\} = \mathbf{c}^\top \mathbf{x}^*$: $\mathcal{P}$ is feasible and $\mathbf{x}^*$ is an optimal point.

Associated with every linear program is the dual linear program. Equation (4.2) shows the dual of the standard linear program, equation (4.1), then also called primal program.

$$\max\ \mathbf{b}^\top \mathbf{y} \quad \text{subject to} \quad \mathbf{A}^\top \mathbf{y} \leq \mathbf{c} \tag{4.2}$$

There exist strong connections between primal and dual problems. Given feasible vectors for both of them, the dual objective lower bounds the primal objective, and, vice versa, the primal upper bounds the dual, see equation (4.3).

$$\mathbf{c}^\top \mathbf{x} \geq \left(\mathbf{A}^\top \mathbf{y}\right)^\top \mathbf{x} = \mathbf{y}^\top (\mathbf{A}\,\mathbf{x}) = \mathbf{y}^\top \mathbf{b} \tag{4.3}$$

Further, the values of both objective functions, equation (4.1) and equation (4.2), coincide if the points for the primal $\mathbf{x}^*$ and the dual $\mathbf{y}^*$ are optimal points, $\mathbf{c}^\top \mathbf{x}^* = \mathbf{b}^\top \mathbf{y}^*$. Let $\mathcal{P}$ be a primal linear program in standard form and $\mathcal{D}$ its dual then the following holds:

- If $\mathcal{P}$ is feasible and unbounded then $\mathcal{D}$ is not feasible.
- If $\mathcal{P}$ is feasible and bounded then so is $\mathcal{D}$.

Concerning the optimization of linear programs, the simplex algorithm, invented by George Dantzig in 1947, has been the first applied algorithm and, at this, marks a breakthrough in modern optimization, see [45] for a historical overview. Starting at some arbitrary corner point, also called vertex or extreme point, of the polytope, $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, the algorithm proceeds to one of the neighboring vertices as long as it leads to an improvement of the objective function. The algorithm terminates once it reaches a vertex where no further improvement is possible. Such a point must necessarily be globally optimal due to convexity of the polytope. Though the simplex algorithm has exponential worst-case complexity it performs much better in practice.

In 1984 Karmarkar introduced interior-point algorithms which have a polynomial-time worst-case complexity. These methods rely on barrier functions which attend a finite value within the polytope but increase towards infinity near the borders of the polytope. Starting from a point inside the polytope, this point is pushed in a direction where the objective function improves while the barrier functions ensure that the point is kept inside the polytope. If there is no further improvement for a certain number of iterations the interior-point algorithm terminates. In contrast to the simplex method which always returns a vertex of the polytope, interior-point solvers are more likely to pick a solution in the center of the polygon spanned by the optimal vertices. A detailed introduction to interior-point algorithms and linear programming can be found in [138].

4.2 LP Relaxations

Linear programming in connection with binary tomography has been firstly suggested in [1] and corresponding LP-based relaxations have been proposed in [50] and [62]. In this section, we introduce the relaxations [50, 62] besides others and incorporate, at this, a smoothness prior which favors spatially homogeneous solutions.

4.2.1 FSSV Approach

Fishburn, Schwander, Shepp, and Vanderbei [50] proposed the following LP-relaxation which we call $(FSSV)$:

$$(FSSV) \qquad \min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{0}^\top \mathbf{x}, \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad 0 \leq x_j \leq 1, \quad \forall j \tag{4.4}$$

As can be seen, this is a degenerated form of a linear program since the target function is constantly zero. Nevertheless it, provides a sufficient condition for the feasibility of the reconstruction problem. If the linear program returns a binary solution then this is certainly a solution of the reconstruction problem while otherwise it is unclear if the problem is feasible or not. The major drawback of this approach is that solutions must fulfill the projections constraints exactly which renders this approach very sensitive to noise. For this reason, we consider this approach theoretically but exclude it from our experiments later on. An evaluation of $(FSSV)$ can be found in [133].

4.2.2 Best Inner Fit Approach (BIF)

Equation (3.23) from the last chapter motivates the best inner fit criterion and establishes a close connection to network flow approaches. This criterion has been first applied to binary tomography by Gritzmann et al [61] using, at this, a greedy optimization strategy in order to avoid fractional solutions:

$$(BIF) \qquad \max_{\mathbf{x}\in\mathbb{R}^n} \mathbf{1}^\top \mathbf{x}, \quad \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad 0 \leq x_j \leq 1, \quad \forall j \tag{4.5}$$

In contrast to the $(FSSV)$ approach the (BIF) criterion is less sensitive to noise since each projection ray is allowed to either fit or under fit its constraint. Thus, the feasibility of the linear program is obvious since $\mathbf{x} = 0$ is a feasible point.

Concerning reconstructions from two projections[1], we have seen in section 3.3.3 that the (BIF) and max flow approach are closely related since solutions obtained for this specific reconstruction setup are guaranteed to be binary due to the total unimodularity of the constraint matrix.

4.2.3 Norms

Besides the approaches in section 4.2.1 and 4.2.2, we consider relaxations that minimize the difference between $\mathbf{A} \cdot \mathbf{x}$ and $\mathbf{b}$ according to some norm $\| \cdot \|_p$. At this, it is possible to give a closed form for $\mathbf{x}$ in case of the $\mathcal{L}_2$-norm,

$$\underset{\mathbf{x}}{\text{argmin}} \ \{E(\mathbf{x}) := \ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2\} = \tag{4.6}$$

$$\underset{\mathbf{x}}{\text{argmin}} \ \left\{E(\mathbf{x})^2 = \ \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{b}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{b}\right\} \tag{4.7}$$

$$\nabla E(\mathbf{x})^2 = 2\mathbf{A}^\top \mathbf{A}\mathbf{x} - 2\mathbf{A}^\top \mathbf{b} \overset{!}{=} \mathbf{0} \iff \mathbf{x} = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b}. \tag{4.8}$$

However, equation (4.8) is of limited gain since the inversion of $\mathbf{A}^\top\mathbf{A}$ is prohibitively expensive for larger problems, silently assuming that the inverse actually exists. In the subsequent chapters, we will revise the quadratic minimization problem, equation (4.7), but it is of less interest in view of linear programming.

Concerning the $\mathcal{L}_1$- and $\mathcal{L}_\infty$-norm, it is well-known that they can be well approximated via linear programming.

[1] Note that not only the projections of the row and the column sum are suitable but any pair of projections which leads row-wise to constant entries in matrix $\mathbf{A}$. By this, one can factor out this entry for each row/equation independently, divide the corresponding projection value b on the right hand side through it, and, thus, obtains a binary matrix $\mathbf{A}$ and an integral vector $\mathbf{b}$.

To begin with the $\mathcal{L}_1$-norm, the relaxed reconstruction problem reads

$$\min_{\mathbf{x}\in[0,1]^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_1. \tag{4.9}$$

In order to approximate the absolute value of each component in vector $(\mathbf{A}\mathbf{x}-\mathbf{b})$, we first introduce corresponding slackvariables $\mathbf{s}^\top := (s_1,...,s_m)$. The objective function minimizes the sum over all slackvariables and, hence, approximates equation (4.9) implicitly. Let $\tilde{\mathbf{x}}^\top := (\mathbf{x}^\top\mathbf{s}^\top)$ and $\tilde{\mathbf{c}}^\top := (\mathbf{0}^\top\mathbf{1}^\top)$, the linear program approximation of equation (4.9) is given by

$$\min_{\mathbf{x}\in\mathbb{R}^n} \tilde{\mathbf{c}}^\top\tilde{\mathbf{x}} \quad \text{subject to} \quad \begin{pmatrix} \mathbf{A} & -\mathbf{I} \\ -\mathbf{A} & -\mathbf{I} \end{pmatrix}\tilde{\mathbf{x}} \le \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}, \ 0 \le x_i \le 1 \quad \forall i. \tag{4.10}$$

In case of the $\mathcal{L}_\infty$-norm,

$$\min_{\mathbf{x}\in[0,1]^n} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_\infty, \tag{4.11}$$

only a single slackvariable s is necessary approximating the maximal disagreement of a component in $(\mathbf{A}\mathbf{x}-\mathbf{b})$. Similarly to the $\mathcal{L}_1$ reconstruction, equation (4.10), the linear program is described in terms of $\tilde{\mathbf{x}}^\top := (\mathbf{x}^\top s)$ and $\tilde{\mathbf{c}}^\top := (\mathbf{0}\ 1)$,

$$\min_{\mathbf{x}\in\mathbb{R}^n} \tilde{\mathbf{c}}^\top\tilde{\mathbf{x}} \quad \text{subject to} \quad \begin{pmatrix} \mathbf{A} & -\mathbf{1} \\ -\mathbf{A} & -\mathbf{1} \end{pmatrix}\tilde{\mathbf{x}} \le \begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}, \ 0 \le x_i \le 1 \quad \forall i. \tag{4.12}$$

In retrospect of minimizing the $\mathcal{L}_\infty$-norm and $\mathcal{L}_1$-norm, it seems natural to deliberate the differences between several choices of $p \in \{1,2,\infty\}$. Let us, therefore, consider the individual residuals

$$r_i(\mathbf{x}) := y_i - \mathbf{a}_i^\top\mathbf{x}, \quad \forall i = 1,...,m\ , \tag{4.13}$$

and suppose that the r_i are double-sided exponentially distributed, ,

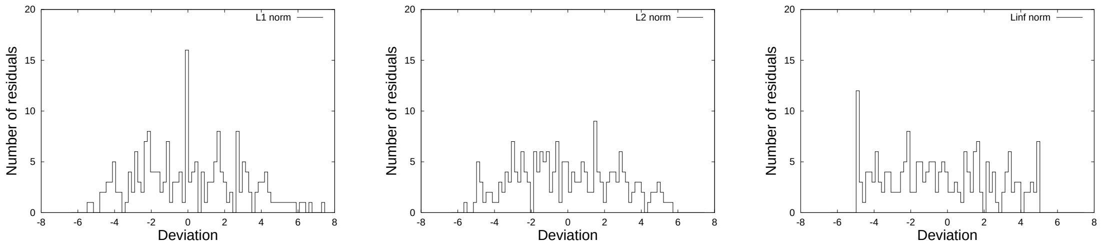
$$p_a(\mathbf{z}) := \frac{1}{2a}\exp\left(-\frac{|\mathbf{z}|}{a}\right). \tag{4.14}$$

Accordingly, their likelihood function $\mathcal{L}(\mathbf{y}|\mathbf{x})$ is defined by

$$\mathcal{L}(\mathbf{y}|\mathbf{x}) := \prod_{i=1}^m p_a(r_i(\mathbf{x})) = \left(\frac{1}{2a}\right)^m \exp(-\frac{1}{a}\sum_{i=1}^m |r_i(\mathbf{x})|) \tag{4.15}$$

and, respectively, the log likelihood function

$$\ln\mathcal{L}(\mathbf{y}|\mathbf{x}) = -(2+\ln\, a)\, m - \frac{1}{a}\sum_{i=1}^m |r_i(\mathbf{x})|. \tag{4.16}$$



$\mathcal{L}_1$ $\mathcal{L}_2$ $\mathcal{L}_\infty$

Figure 4.1: We minimized $\|\mathbf{Ax} - \mathbf{b}\|_p$ for $p \in \{1, 2, \infty\}$ with $\mathbf{A} \in [0,1]^{200\times 30}$ and $\mathbf{b} \in [0, 10]^{200}$ randomly generated. All three plots show histograms of the individual residuals r_i. In case of the $\mathcal{L}_1$-norm the residual has the widest spread and a strong peak at 0. On the other side, the $\mathcal{L}_\infty$-norm gives the smallest interval containing all r_i. The $\mathcal{L}_2$-norm is in between $\mathcal{L}_1$ and $\mathcal{L}_\infty$.

Maximizing the log likelihood function, equation (4.16), with respect to $\mathbf{x}$ yields the $\mathcal{L}_1$-estimate as maximum likelihood (ML) estimate

$$\max_{\mathbf{x}} \ln \mathcal{L}(\mathbf{y}|\mathbf{x}) = \max_{\mathbf{x}} -\frac{1}{a}\sum_{i=1}^{m} |r_i(\mathbf{x})| = \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{Ax}\|_1 \,. \tag{4.17}$$

Similarly, if we assume a normal distribution,

$$p_\sigma(\mathbf{z}) := \frac{1}{\sqrt{(2\pi\sigma^2)}} \exp(-\frac{1}{2\sigma^2}\mathbf{z}^2), \tag{4.18}$$

the $\mathcal{L}_2$-estimate is obtained as maximum likelihood estimate,

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{Ax}\|_2 \,. \tag{4.19}$$

Finally, in case of a uniform distribution,

$$p_a(\mathbf{z}) := \begin{cases} \frac{1}{2a} & -a \leq \mathbf{z} \leq a \\ \\ 0 & \text{otherwise} \end{cases}, \tag{4.20}$$

any $\mathbf{x}$ that satisfies

$$\|\mathbf{y} - \mathbf{Ax}\|_\infty \leq a \tag{4.21}$$

is a maximum likelihood estimate.

To illustrate this further, we randomly generated a matrix $\mathbf{A}$ and a vector $\mathbf{b}$ and set up optimization problems of the form $\min_{\mathbf{x}} \|\mathbf{Ax}-\mathbf{b}\|_p$ with $p \in \{1, 2, \infty\}$. We optimized the problems with the suggested relaxations, where in case of $p = 2$ we used the pseudoinverse instead of a direct matrix inversion since $\mathbf{A}^\top\mathbf{A}$ is likely to be singular. Figure 4.1 shows the distribution (histogram) of the residuals r_i and, as can be seen, the results nicely reflect the distributions assumed for the maximum likelihood estimation.

4.3 Regularization

The approaches introduced in section 4.2 compute approximations to the binary reconstruction problem. At this, their results can significantly differ from the underlying image due to the ill-posedness of problem. Since we are particularly interested in objects that exhibit some kind of spatial homogeneity we focus on the regularization of the former approaches in favor of more coherent solutions [133, 135, 136, 121].

We explicitly demonstrate the regularization in case of the $(FSSV)$ and (BIF) approach and stress that the procedure is analogous for the norm approximations, equations (4.10) and (4.12). In reference to the notation used in former sections, we denote the corresponding regularized LP relaxations by $(reg - FSSV)$, $(reg - BIF)$, $(reg - \mathcal{L}_1)$, and $(reg - \mathcal{L}_\infty)$.

4.3.1 Smoothness Prior

A natural way to derive a smoothness term is to consider proper discretizations of smoothness functionals in the continuous space $\mathbb{R}^m$

$$\|L(f)\|_2^2 \,. \tag{4.22}$$

At this, the operator L measures the spatial variation of the attenuation function f. The simplest example for a suitable L is the gradient operator

$$L(f) = \nabla f = (\frac{\partial}{\partial x_1}f, ..., \frac{\partial}{\partial x_m}f)^\top. \tag{4.23}$$

We confine ourselves to the simplest choice, equation (4.23), which, by discretizing equation (4.22), leads to the well–known 5–point stencil for the Laplacian. At the functional level this simply corresponds to summing up the differences at adjacent pixel positions, in our case the 4-neighborhood of a pixel,

$$\sum_{\langle j,k \rangle} (x_j - x_k)^2 \,, \tag{4.24}$$

where only horizontal and vertical (not diagonal) positions are considered as nearest neighbors. In order to derive a suitable LP–relaxation based on [80], we replace the squared terms in equation (4.24) by their absolute values,

$$\sum_{\langle j,k\rangle} |x_j - x_k| \,, \tag{4.25}$$

which in view of the original integer constraint $x_j \in \{0,1\}$ is a reasonable approximation. Following the general approach described in [80], we introduce a vector $\mathbf{z}$ with auxiliary variables $\mathbf{z}^\top = (\dots, z_{\langle j,k\rangle}, \dots)$ and approximate each term $|x_j - x_k|$ by minimizing $z_{\langle j,k\rangle}$ subject to $z_{\langle j,k\rangle} \geq x_j - x_k$ and $z_{\langle j,k\rangle} \geq x_k - x_j$ in the linear program.

4.3.2 Regularized FSSV

Revising the $(FSSV)$ approach from section 4.2.1 the objective function supplemented with the smoothness prior, equation (4.24), reads

$$(reg - FSSV) \qquad \min_{\mathbf{x}\in[0,1]^n} \mathbf{0}^\top\mathbf{x} + \frac{\alpha}{2}\sum_{\langle j,k\rangle} |x_j - x_k| \,, \quad \mathbf{Ax} = \mathbf{b} \,, \tag{4.26}$$

and therewith the corresponding linear program comes out as

$$\min_{\mathbf{x},\mathbf{z}} \mathbf{0}^\top\mathbf{x} + \frac{\alpha}{2}\sum_{\langle j,k\rangle} z_{\langle j,k\rangle} \quad \text{subject to} \qquad \begin{aligned} &\mathbf{Ax} = \mathbf{b} \\ &z_{\langle j,k\rangle} \geq x_j - x_k \\ &z_{\langle j,k\rangle} \geq x_k - x_j \\ &0 \leq x_j \leq 1 \,, \quad \forall j. \end{aligned}$$

4.3.3 Regularized BIF

In case of the (BIF) approach, section 4.2.2, we proceed analogously to section 4.3.2, i.e. we consider

$$(reg - BIF) \qquad \min_{\mathbf{x}\in[0,1]^n} -\mathbf{1}^\top\mathbf{x} + \frac{\alpha}{2}\sum_{\langle j,k\rangle} |x_j - x_k| \,, \quad \mathbf{Ax} \leq \mathbf{b}, \tag{4.27}$$

and obtain the corresponding linear program as

$$\min_{\mathbf{x},\mathbf{z}} -\mathbf{1}^\top \mathbf{x} + \frac{\alpha}{2} \sum_{\langle j,k \rangle} z_{\langle j,k \rangle} \quad \text{subject to} \quad \begin{aligned} &\mathbf{Ax} \leq \mathbf{b} \\ &z_{\langle j,k \rangle} \geq x_j - x_k \\ &z_{\langle j,k \rangle} \geq x_k - x_j \\ &0 \leq x_j \leq 1\,, \quad \forall j. \end{aligned}$$

Referring to the total unimodularity of $\mathbf{A}$ which guaranteed a binary solutions for certain pairs of projections, section 4.2.2 and 3.3.3, it is worth mentioning that adding the smoothness constraints $z_{\langle j,k \rangle} \geq x_j - x_k$ and $z_{\langle j,k \rangle} \geq x_k - x_j$ necessarily leads to a submatrix

$$\det \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = 2$$

and, thus, deteriorates the total unimodularity of the constraint matrix of $(reg - BIF)$.

4.4 Rounding

Due to the linear programming relaxation, we only obtain a fractional solution $\mathbf{x} \in [0,1]^n$ so far, instead of $\mathbf{x} \in \{0,1\}^n$. In order to finally achieve a reasonable binary approximation of the original problem, we consider randomized and deterministic rounding procedures applied to the fractional LP solution afterwards.

4.4.1 Randomized Rounding

Probably the simplest randomized or probabilistic rounding procedure is to consider the fractional values of the solution $\mathbf{x}$ as some kind of assignment probabilities, i.e. pixel i receives label 1 with probability x_i and label 0 with probability $1 - x_i$. By choosing a random value $\gamma \in [0,1]$ individually for each pixel, one assigns 1 if $\gamma \leq x_i$ and 0 otherwise. It is, however, reported by Kleinberg and Tardos [80] that this rounding scheme leads to poor results and, instead, they suggest the rounding procedure which is adapted in algorithm 5 for our needs. Again, the fractional values are considered as assignment probabilities but in contrast to the simple approach above several pixels are assigned during a single iteration.

Let I denote the set of pixels, $i \in \{1, ..., n\}$ and consider all pixels as unassigned initially, a single iteration of the algorithm randomly selects a label $l \in \{0,1\}$ and a value $\gamma \in [0,1]$, and assigns label l to all unassigned pixels i if $\gamma \leq x_{i,l}$ with $x_{i,0} := 1 - x_i$ and $x_{i,1} := x_i$. The algorithm terminates if there are no unassigned pixels left, i.e. $Q = \emptyset$.

Algorithm 5 Randomized Rounding [80]

```
$\mathcal{L} := \{0, 1\}$
$I := \{1, ..., n\}$
$Q := I$
while $Q \neq \emptyset$ do
    $l :=$ random($\mathcal{L}$)
    $\gamma$ :=random($[0, 1]$)
    for all $i \in Q$ do
        if $l = 0$ and $\gamma \leq 1 - x_i$ then
            $x_i := 0$
            $Q := Q - \{i\}$
        else if $l = 1$ and $\gamma \leq x_i$ then
            $x_i := 1$
            $Q := Q - \{i\}$
        end if
    end for
end while
```

With respect to algorithm 5, the work of Kleinberg and Tardos [80] includes the following results which are adapted to our binary case here.

Lemma 6 ([80]). *The probability that an unassigned pixel i is assigned label l in a given iteration is precisely $x_{i,l}/2$, the probability that pixel i is assigned any label is precisely $1/2$. Further, over all iterations, the probability that a pixel i is assigned label l is precisely $x_{i,l}$.*

Proof. We uniformly choose a label l with probability $1/2$ and the probability that pixel i is assigned label l is $x_{i,l}/2$. Hence, the probability that pixel i is assigned to any label is $\sum_{l \in \{0,1\}} x_{i,l}/2 = 1/2$ and that pixel i is assigned label l over all iterations is

$$\frac{1}{2}\, x_{i,l} \sum_{p=0}^{\infty} \frac{1}{2^p} = x_{i,l}.$$

□

Following [80], two neighboring pixels i and j are separated by an iteration if they were unassigned before and exactly one of them is assigned during that iteration. It is worth noting that if two neighboring pixels receive different labels then they were separated during some iteration, the opposite is obviously not true since the rejected pixel might obtain the same label during an iteration later on.

Lemma 7 ([80]). *Let pixel i and j be neighbors, the probability that both pixels are separated during an iteration is exactly $|x_{i,1} - x_{j,1}|$. The probability that the two pixels i and j are assigned differently by the process is at most $2|x_{i,1} - x_{j,1}|$.*

Proof. Without loss of generality we assume $x_{i,l} \geq x_{j,l}$ for some selected label l and $\gamma \in [0,1]$ chosen uniformly. Case (i) $\gamma > x_{i,l}$, both pixels remain unassigned. Case (ii) $x_{i,l} \geq \gamma > x_{j,l}$, label l is assigned to pixel i but not to pixel j. Case (iii) $x_{j,l} \geq \gamma$, label l is assigned to both pixels. Consequently, only case (ii) separates pixels and occurs with probability $|x_{i,l} - x_{j,l}|$.

The probability that exactly one of the objects is assigned during an iteration is covered by

$$P_S := \frac{1}{2} \sum_{l \in \{0,1\}} |x_{i,l} - x_{j,l}| = \frac{|x_{i,0} - x_{j,0}| + |x_{i,1} - x_{j,1}|}{2} = |x_{i,1} - x_{j,1}|.$$

Further, let P_A be the probability that both pixels are assigned in the current iteration and P_R be the probability that both pixels are rejected, obviously $P_A + P_S + P_R = 1$. One observes that the overall process will surely not separate the pixels i and j if they receive their label assignment in the same iteration and the probability for this case is

$$P_A \sum_{p=0}^{\infty} (P_R)^p = \frac{P_A}{1 - P_R} = \frac{1 - P_S - P_R}{1 - P_R} = 1 - \frac{P_S}{1 - P_R} = 1 - \frac{P_S}{P_A + P_S}.$$

Thus, the probability that pixel i and j are separated by the process is upper-bounded by

$$\frac{P_S}{P_A + P_S}$$

where $P_A + P_S$ is the probability that at least one of both pixels is assigned. This is at least the probability that one pixel is assigned which is $1/2$ according to lemma 6. Consequently, the probability that two pixels i and j are assigned to different labels is at most $2|x_{i,1} - x_{j,1}|$. □

Lemma 7 makes precise that the randomized procedure, algorithm 5, preserves spatial smoothness to some extent. For practical purposes the rounding procedure is repeated several times and the binarization with the lowest Err_1 measure, section 3.4, is kept as result.

4.4.2 Deterministic Rounding

In [133, 135, 136, 121], we obtained reasonable binarizations using a simple deterministic rounding scheme which thresholds the fractional solution $\mathbf{x}$ with a certain predefined global threshold parameter t, i.e. if $x_i \leq t$ then $x_i = 0$ and $x_i = 1$ otherwise.

A more sophisticated deterministic method has been proposed by Kleinberg and Tardos [80] which is a derandomized version of the probabilistic algorithm 5, see our adaption in algorithm 6. The outline of the algorithm is quite simple. Consider the set of all fractional values occurring in the fractional LP solution and assume all elements of this set arranged in a list with increasing order. Note that each grayvalue appears only once in the list although it might appear several times in the fractional solution. We obtain the set of all thresholds as mean values of any pair of grayvalues succeeding in the list. By this, all thresholds leading to different binarizations are considered which certainly includes the results of the randomized algorithm 5. Each threshold is applied to the fractional solution and the rounding with the lowest Err_1 measure, section 3.4, is returned.

Algorithm 6 Deterministic Rounding [80]

```
Require: $\mathbf{x}$ - fractional solution.
Require: $\mathbf{g}$ - contains the different fractional values of $\mathbf{x}$ increasingly ordered.

$T := \{(g_i + g_{i+1})/2,\ 0 \leq i < n\}$
$\hat{\mathbf{x}}$ := threshold $\mathbf{x}$ with arbitrary $t \in T$.
$e_{\min} := Err_1(\hat{\mathbf{x}}) = \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_2$

for all $t \in T$ do
    $\hat{\mathbf{x}}$ := threshold $\mathbf{x}$ at $t$.
    $e := Err_1(\hat{\mathbf{x}})$
    if $e < e_{\min}$ then
        $e_{\min} := e$
        $\hat{\mathbf{x}}_{\min} := \hat{\mathbf{x}}$
    end if
end for

return $\hat{\mathbf{x}}_{\min}$ and $e_{\min}$
```

For practical implementations, this procedure might be considered as too expensive since in worst case all pixels contain different values, hence, a large number of thresholds must be applied. From our experience, this should, however, not be the case as it contradicts the influence of the smoothness prior. Besides, our experiments, section 4.5, indicate that the optimal threshold is significantly close to $t = 0.5$ which equals the fixed predefined global threshold we used in [133, 135, 136, 121].

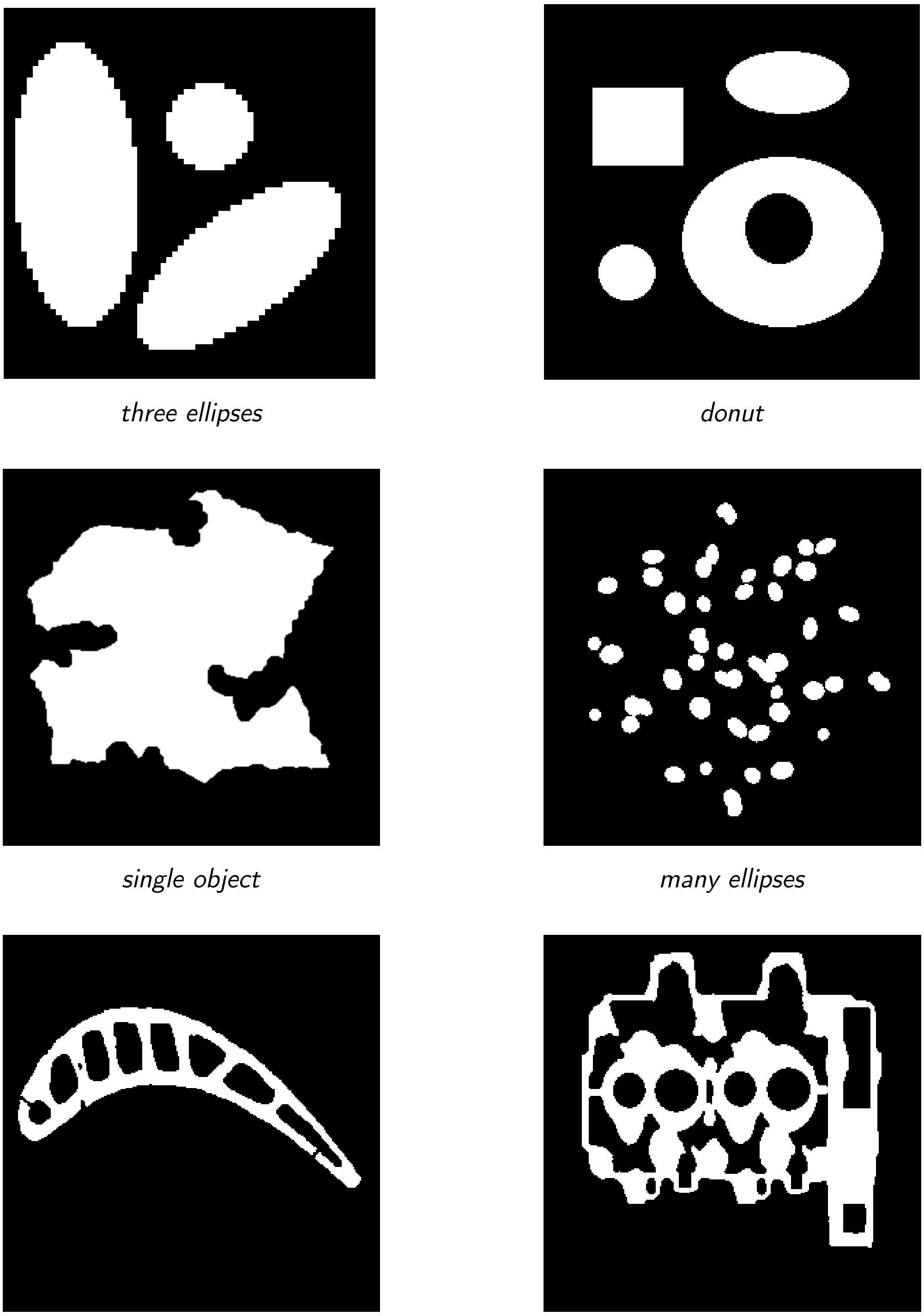
4.5 Numerical Evaluation

We evaluate the approaches $(reg - BIF)$, $(reg - \mathcal{L}_1)$, and $(reg - \mathcal{L}_\infty)$ for scenarios involving noiseless and noisy projections. We, however, exclude the $(reg - FSSV)$ approach from our experiments as it is clearly inferior, the equality constraint prevents its application to inconsistent equation systems which occur in many real world situations due to the presence of noise. Instead, we refer the interested reader to [133] for a comparison of $(reg - FSSV)$ and $(reg - BIF)$. Furthermore, we do not evaluate the enhancing influence of the smoothness prior on reconstruction results since this has been extensively studied in [133, 135, 136, 121].

Using the parallel beam geometry and the image-based discretization scheme, section 3.2, we setup different reconstruction problems from the images shown in figure 4.2. In each projection the distance between two closest rays is set to the side length of a pixel which is assumed to be 1 and, consequently, the horizontal and the vertical projection coincide with the row and the column sum.

For the implementation of our algorithms, we used the gcc C++ compiler and the C interface to CPLEX, a commercial optimization package from the ILOG[2] company which provides a simplex and an interior point solver for linear programming. Depending on the chosen solver, not only the speed of the reconstruction process can differ remarkably, interior point methods are typically faster than simplex optimizers, but also the reconstruction results which we demonstrate in figure 4.3. As can be seen there, the total unimodularity of matrix A guarantees that all vertices of the polytope are integer, section 3.3.3, and in case of the simplex algorithm the solution is an optimal vertex (or basic solution) of the polytope, section 4.1. However, as the solution of the reconstruction problem is not unique, there also exist other basic solutions and interior point solvers typically locate a non-basic solution somewhere in the middle of the polygon spanned by the optimal vertices. Of course, this solution must not be binary anymore, even in case of total unimodularity. To have a basic solution again, it is possible to use the crossover method, also part of the CPLEX package, which transforms a non-basic to a basic

[2] www.ilog.com



three ellipses

donut

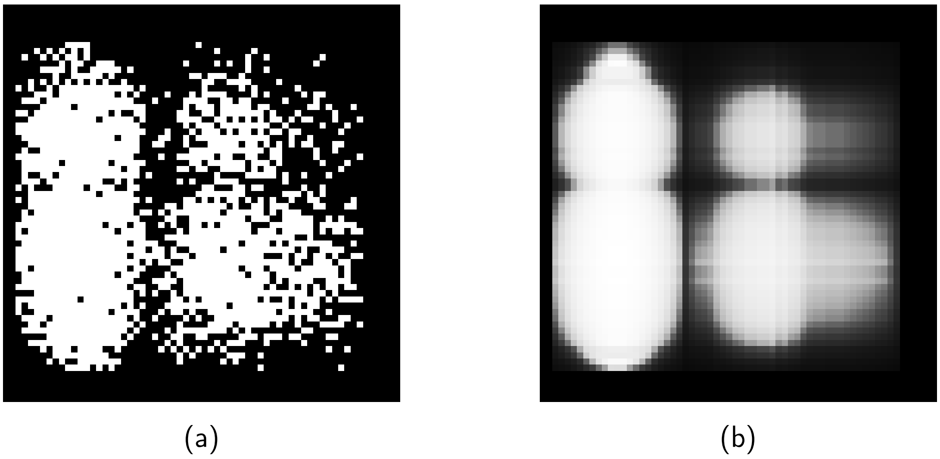
single object

many ellipses

turbine

cylinders

Figure 4.2: Different phantom images which are used throughout this work for the generation of reconstruction problems. Thanks are due to Joost Batenburg [7] who kindly provided us with the *single object*, *many ellipses*, *turbine*, and *cylinders* image.



(a) (b)

Figure 4.3: Using the horizontal and the vertical projection (rows and columns sums), a reconstruction problem has been setup from the *three ellipses* image and was solved with (a) the simplex algorithm and (b) the interior point solver. Due to the unimodularity of the problem the simplex solution is binary while the interior point solution is still fractional.

CPLEX parameter	Value
CPX_PARAM_LPMETHOD	Barrier optimizer (CPX_ALG_BARRIER)
CPX_PARAM_BARSTARTALG	Dual is 0 (default)
CPX_PARAM_BARCROSSALG	No crossover (-1)
CPX_PARAM_BAREPCOMP	Convergence tolerance (1e-3)

Table 4.1: Settings used in connection with the CPLEX interior point solver.

solution. This is, however, a time-consuming process and abolishes the gain of using faster interior point solvers.

The simplex algorithm is only preferable if total unimodularity is exploited which is not the case for us since two projections are certainly not sufficient, even for moderate complex objects, and we prefer to optimize over all projection at once. Besides, our regularization term does not preserve total unimodularity, as can be seen in section 4.3.3. For these reasons, we decided to use the CPLEX interior point solver for optimization purposes with the settings and options summarized in table 4.1.

Thoroughly, projection angles are selected at equally spaced positions over $180°$ (excluding the projection at $180°$). Hence, it is sufficient to refer to the number of projections without explicitly specifying the actual angles. Note that we use a range of $180°$ here in order to be comparable to the results published in [7]. The number of projections is usually chosen very

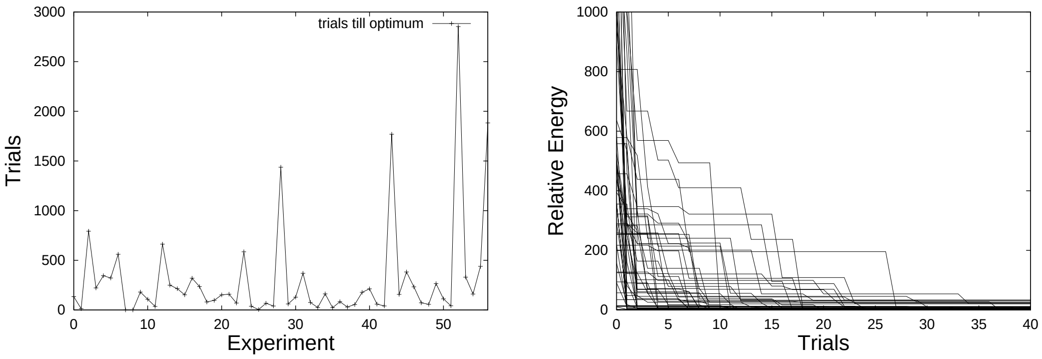


Figure 4.4: Evaluation of the randomized rounding procedure, algorithm 5: (left) Number of trials needed in order to achieve the same result as the deterministic rounding, algorithm 6. On average only a few trials were needed except in a few cases (outliers). (right) In each run the currently best rounding obtained by the randomized algorithm is kept. This plot shows the decrease of the relative energy, defined by *currently best randomized energy - best deterministic energy*, over all experiments. As can be seen, in all cases only a few trials are necessary in order to reach a sufficient rounding.

small such that the reconstruction problems becomes even more difficult. However, we stress that our approaches reveal the groundtruth image from more projections. Some results of the noiseless experiments can be found in figures 4.8 – 4.11 where the fractional solution is always shown in the first and third row and the corresponding binary solution is placed just below in the second and fourth row. In case of the noisy experiments, a normally distributed error $\mathcal{N}(\mu := 0, \sigma)$ has been added to the projections in advance, results of these experiments can be found in figure 4.16.

The rounding of the fractional solution was performed by first computing the optimal deterministic rounding with algorithm 6 and, afterwards, starting the randomized rounding, algorithm 5, until it achieves the same result. Figure 4.4 (left) plots the number of trials needed by the randomized procedure in order to find the same result as its deterministic counterpart and shows that on average only a few trials were necessary. The right side of figure 4.4 plots the relative energy (currently best randomized energy - optimal deterministic energy) and it can be seen that the energy in all experiments has been reduced significantly after a few trials (≈ 40). Therefore, the randomized method is advisable if a potentially suboptimal but still reasonable rounding is sufficient.

Referring to the list of thresholds found and used by the deterministic rounding procedure, algorithm 5, we consider the binary images corresponding to each threshold and plot the energy,

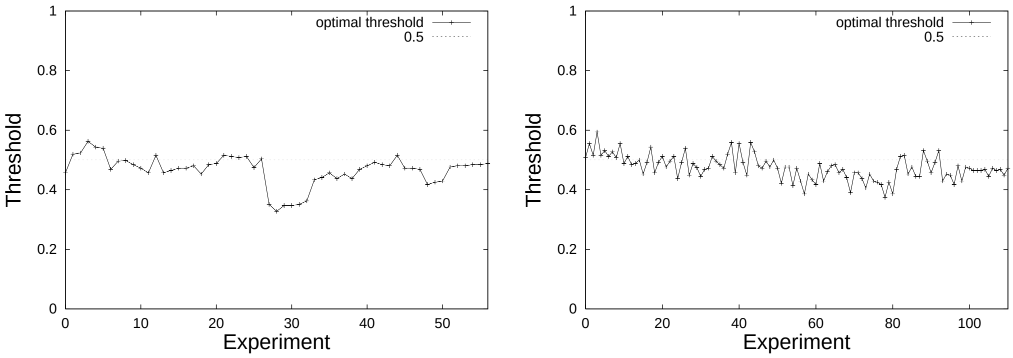


Figure 4.5: Optimal thresholds found by the deterministic rounding procedure, algorithm 5: (left) Plot of the optimal deterministic thresholds against a fixed threshold at 0.5 over all noiseless experiments. As can be seen, there is a correlation between both, except for a deviation in the middle which is caused by reconstructions from an insufficiently small number of projections. By increasing the number of projections, however, the optimal threshold moves towards 0.5 again. (right) The experiments from noisy projections additionally confirm this correlation. Thus, we conclude that a reasonable rounding can be achieved by simply applying a fixed threshold at 0.5.

figures 4.6 and 4.7, which is just the error measure Err_1 of the binary image from section 3.4. In nearly all cases the optimal threshold is close to 0.5, see figure 4.5 (left), except for a disagreement in the middle part which is caused by reconstructions of the *many ellipses* image from an extremely small number of projections. However, as can be seen, figure 4.7 (left side), the optimal threshold also inclines towards 0.5 in this experiments if the number of projections increases. Our experiments from noisy projections additionally support this correlation of the optimal deterministic threshold with 0.5, figure 4.5 (right). Therefore, we can expect a reasonable rounding by simply applying a fixed threshold at 0.5 [133, 135, 136, 121].

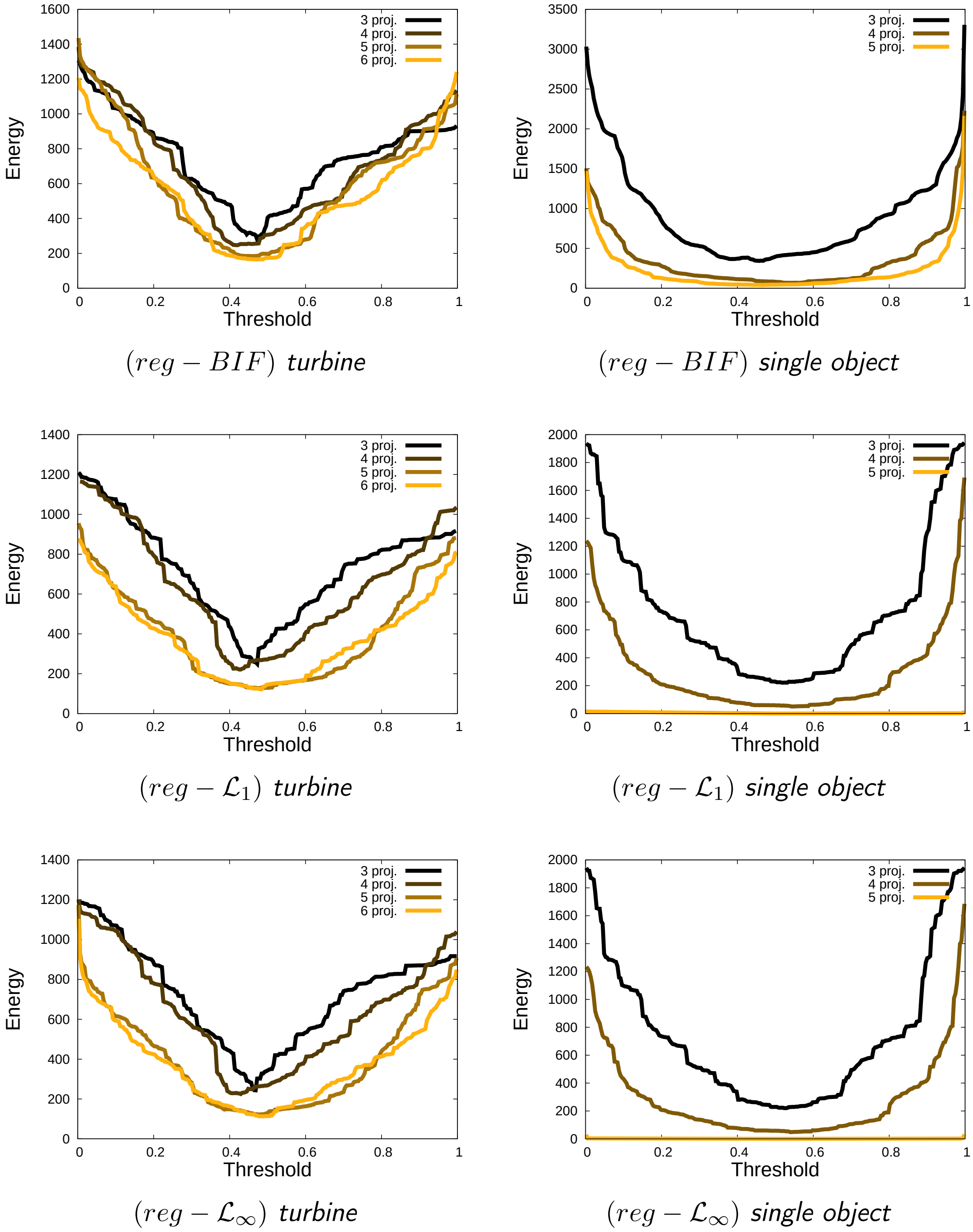
In order to evaluate the quality of the reconstructions with respect to the groundtruth phantom images and the projection constraints we apply the error measures from section 3.4. Figures 4.12 – 4.15 plot the measurements for the noiseless and 4.17 – 4.20 for the noisy experiments. Apparently, the $(reg - BIF)$ approach performs worst in the noiseless case while $(reg - \mathcal{L}_1)$ and $(reg - \mathcal{L}_\infty)$ yield comparable results. In some experiments the results of $(reg - \mathcal{L}_1)$ are better while in others $(reg - \mathcal{L}_\infty)$ is ahead, nevertheless, the difference is quite small. Regarding the experiments from noisy projections the $(reg - \mathcal{L}_1)$ approach delivered the best results with respect to Err_1, Err_2, and Err_4 in most cases. However, there is no clear preferable approach with respect to the Err_3 measure.

Concerning the time consumption, table 4.2, the $(reg - BIF)$ approach is clearly favorable in comparison to $(reg-\mathcal{L}_1)$ and $(reg-\mathcal{L}_\infty)$. In the *turbine* experiment the $(reg-\mathcal{L}_\infty)$ approach performed comparable to $(reg-\mathcal{L}_1)$ but in others, particularly the *more ellipses* and *cylinders* experiments, it was clearly inferior.

A comparison of our $(reg-BIF)$ approach and a network flow approach, section 3.3.3, has been conducted in [7]. According to this, both algorithms deliver reconstructions of comparable quality but $(reg-BIF)$ seems to be inferior concerning the time consumption. Some of our own experiments are quite similar, i.e. same reconstruction phantom, same projections, however, our machine was a bit stronger (3 GHz Intel Pentium 4) than the one used in [7] (2.4 GHz). In view of our own experiments the network approach is expectedly faster but, nevertheless, the gap appears to be smaller, even if we take into account that the experiments were performed on different machines. A possible explanation could be the settings used for the CPLEX optimizer which greatly affect the time consumption of the algorithm. The default setting leads CPLEX to optimize until a tolerance is reached which is simply unreasonable for reconstruction purposes.

	Nr. projections	$(reg - BIF)$ [sec]	$(reg - \mathcal{L}_1)$ [sec]	$(reg - \mathcal{L}_\infty)$ [sec]
turbine	3	60	573.5	652
	4	56	691	756
	5	63.5	1277	918
	6	77	1375	1042.5
single object	3	220.5	813	764
	4	195	1357	1217.5
	5	297	1632	1244
many ellipses	3	80	429	391
	4	46	632.5	726
	5	73	688	1133
	6	70	901	1397
	7	139	1032	1368
	8	82.5	1577	1960
cylinders	3	152.5	485	641
	4	131	748	1071
	5	167	1212	912
	6	194.5	1514	1594
	8	238	1925	2471
	10	380	960.5	3359

Table 4.2: This table lists the time consumption of the experiments in seconds and, as can be seen, the $(reg - BIF)$ approach is significantly faster than $(reg - \mathcal{L}_1)$ or $(reg - \mathcal{L}_\infty)$. In case of $(reg - \mathcal{L}_1)$ and $(reg - \mathcal{L}_\infty)$, the first approach needs less computational time than the latter, especially in the experiments with the *many ellipses* and the *cylinders* phantom images.



$(reg - BIF)$ *turbine*

$(reg - BIF)$ *single object*

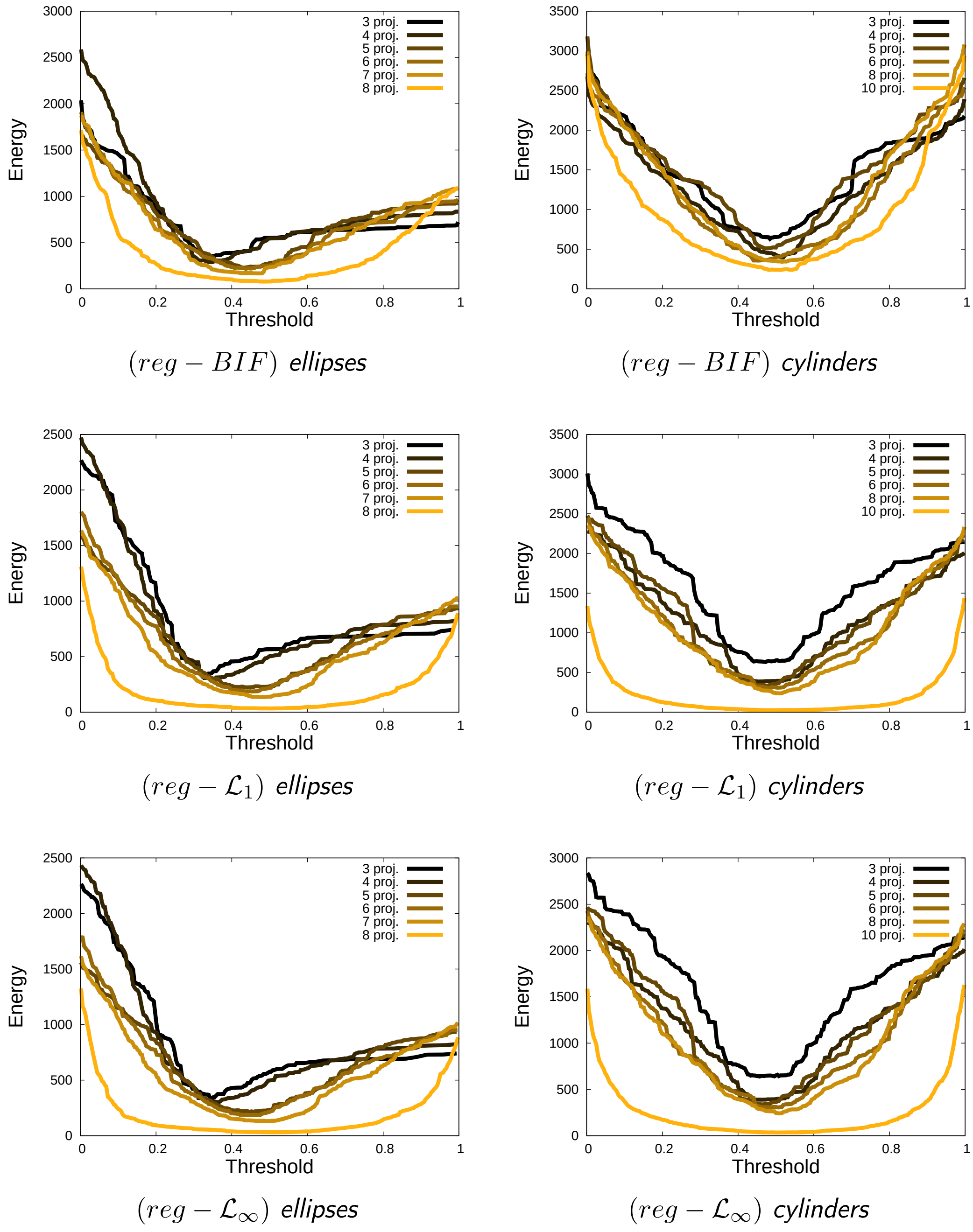
$(reg - \mathcal{L}_1)$ *turbine*

$(reg - \mathcal{L}_1)$ *single object*

$(reg - \mathcal{L}_\infty)$ *turbine*

$(reg - \mathcal{L}_\infty)$ *single object*

Figure 4.6: Deterministic rounding: For each threshold the energy (Err_1), section 3.4, of the resulting binary image is plotted. As can be observed, the optimal threshold leading to the image with the lowest energy is closely located around 0.5.



$(reg - BIF)$ *ellipses*

$(reg - BIF)$ *cylinders*

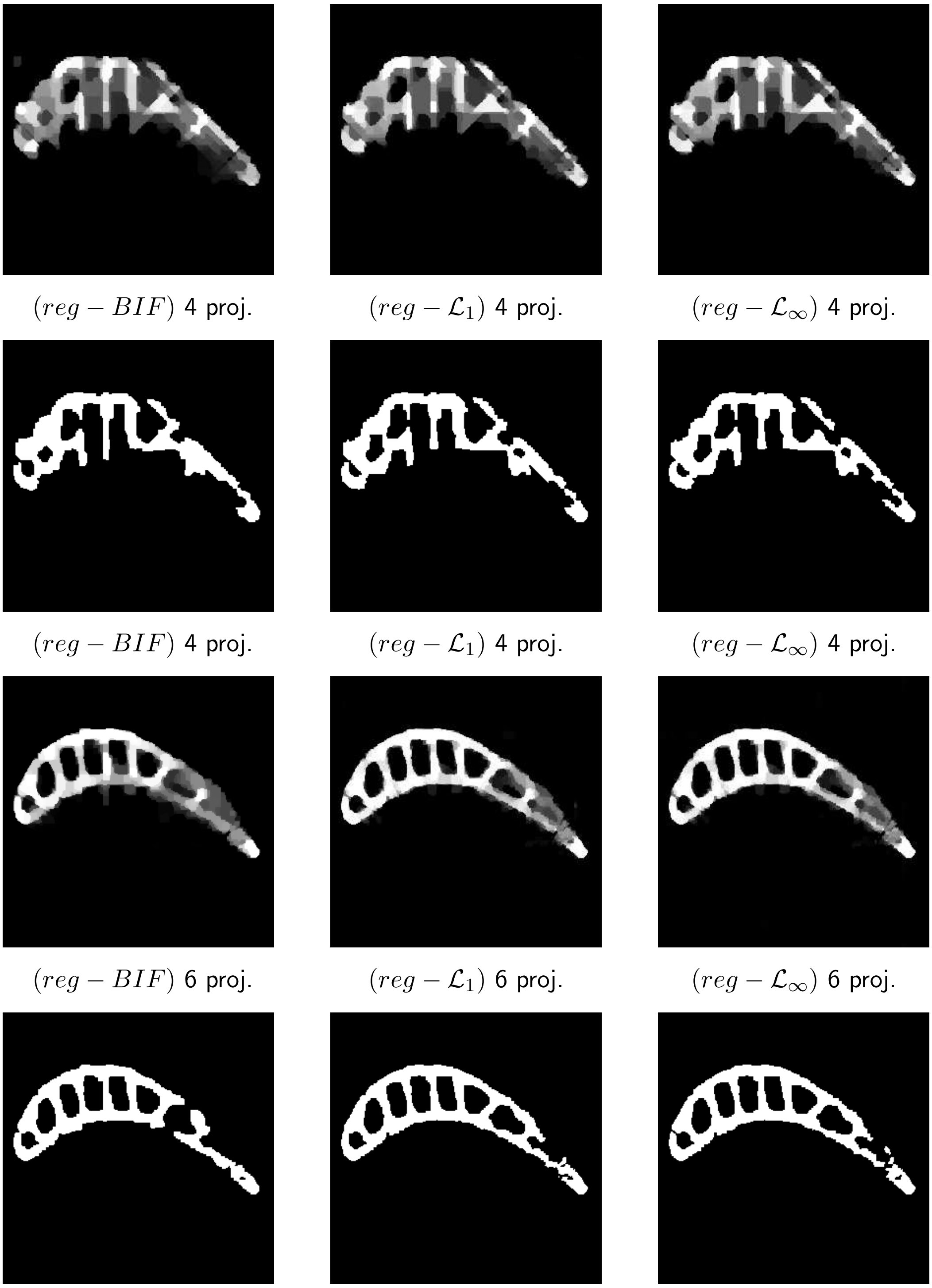
$(reg - \mathcal{L}_1)$ *ellipses*

$(reg - \mathcal{L}_1)$ *cylinders*

$(reg - \mathcal{L}_\infty)$ *ellipses*

$(reg - \mathcal{L}_\infty)$ *cylinders*

Figure 4.7: Deterministic rounding: For each threshold the energy (Err_1) of the resulting binary image is plotted. As can be observed, the optimal threshold leading to the image with the lowest energy is closely located around 0.5. The largest deviation is observed for the experiments from a very small number of projections on the left side. However, as can be seen the optimal threshold moves again towards 0.5 for an increasing number of projections.



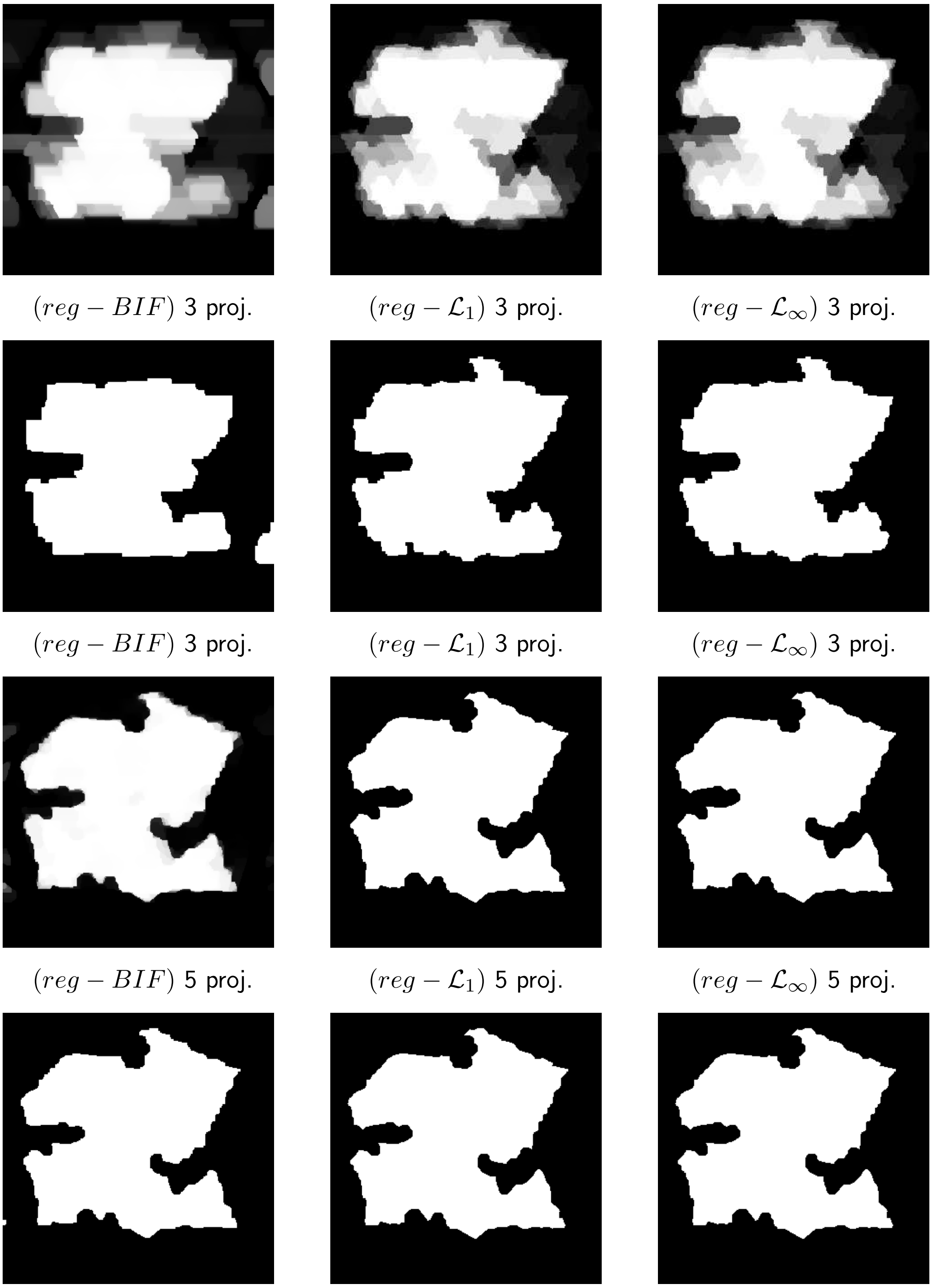
$(reg - BIF)$ 4 proj. $(reg - \mathcal{L}_1)$ 4 proj. $(reg - \mathcal{L}_\infty)$ 4 proj.

$(reg - BIF)$ 4 proj. $(reg - \mathcal{L}_1)$ 4 proj. $(reg - \mathcal{L}_\infty)$ 4 proj.

$(reg - BIF)$ 6 proj. $(reg - \mathcal{L}_1)$ 6 proj. $(reg - \mathcal{L}_\infty)$ 6 proj.

$(reg - BIF)$ 6 proj. $(reg - \mathcal{L}_1)$ 6 proj. $(reg - \mathcal{L}_\infty)$ 6 proj.

Figure 4.8: Reconstructions of the *turbine* image without noise. Throughout these experiments the regularization parameter α was set to 0.5 for $(reg - BIF)$ and $(reg - \mathcal{L}_1)$ and in case of $(reg - \mathcal{L}_\infty)$ to 0.001.



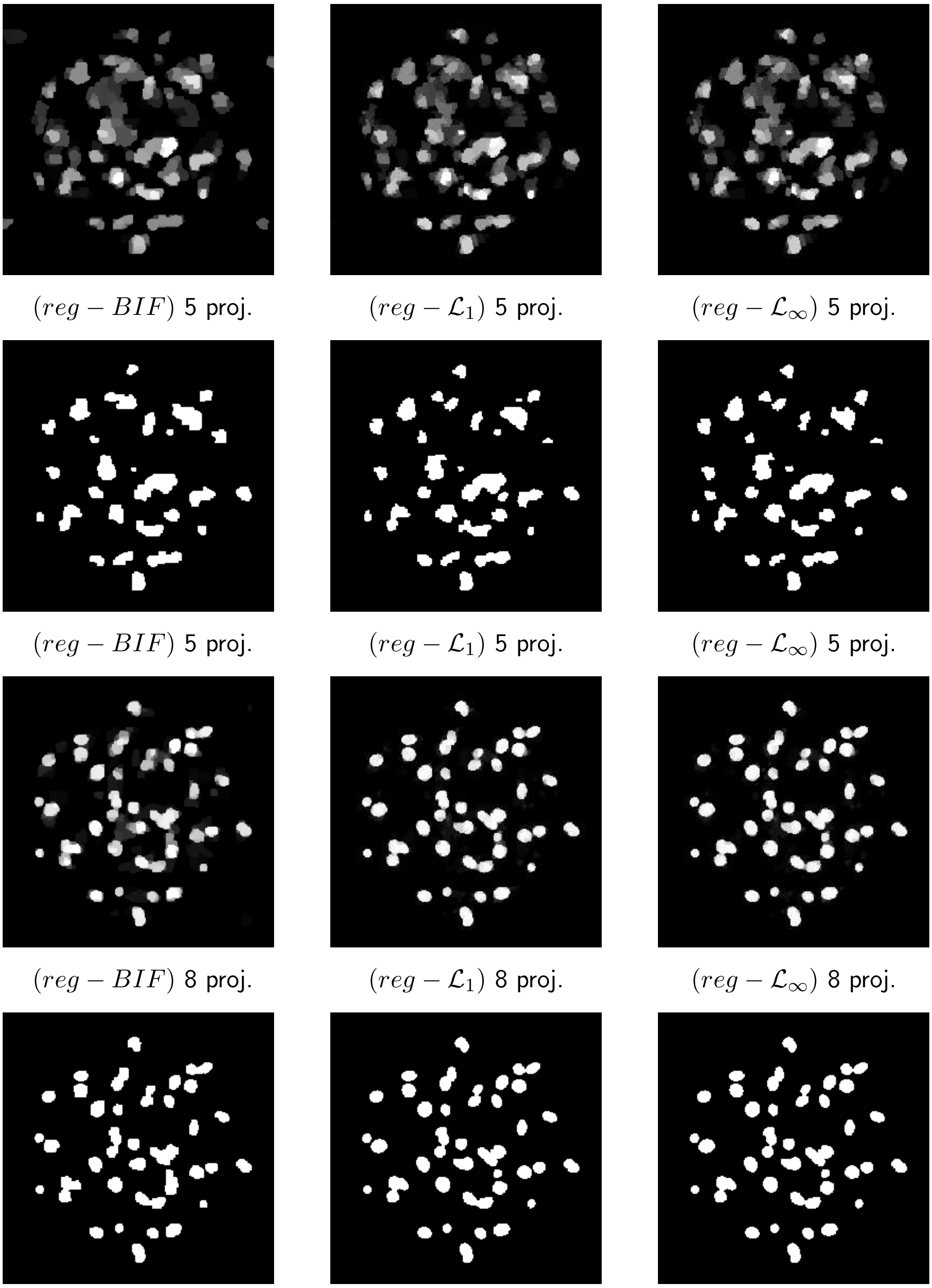
$(reg - BIF)$ 3 proj. $(reg - \mathcal{L}_1)$ 3 proj. $(reg - \mathcal{L}_\infty)$ 3 proj.

$(reg - BIF)$ 3 proj. $(reg - \mathcal{L}_1)$ 3 proj. $(reg - \mathcal{L}_\infty)$ 3 proj.

$(reg - BIF)$ 5 proj. $(reg - \mathcal{L}_1)$ 5 proj. $(reg - \mathcal{L}_\infty)$ 5 proj.

$(reg - BIF)$ 5 proj. $(reg - \mathcal{L}_1)$ 5 proj. $(reg - \mathcal{L}_\infty)$ 5 proj.

Figure 4.9: Reconstructions of the *single object* image without noise. Throughout these experiments the regularization parameter α was set to 0.5 for $(reg-BIF)$ and $(reg-\mathcal{L}_1)$ and in case of $(reg-\mathcal{L}_\infty)$ to 0.001.



$(reg - BIF)$ 5 proj.

$(reg - \mathcal{L}_1)$ 5 proj.

$(reg - \mathcal{L}_\infty)$ 5 proj.

$(reg - BIF)$ 5 proj.

$(reg - \mathcal{L}_1)$ 5 proj.

$(reg - \mathcal{L}_\infty)$ 5 proj.

$(reg - BIF)$ 8 proj.

$(reg - \mathcal{L}_1)$ 8 proj.

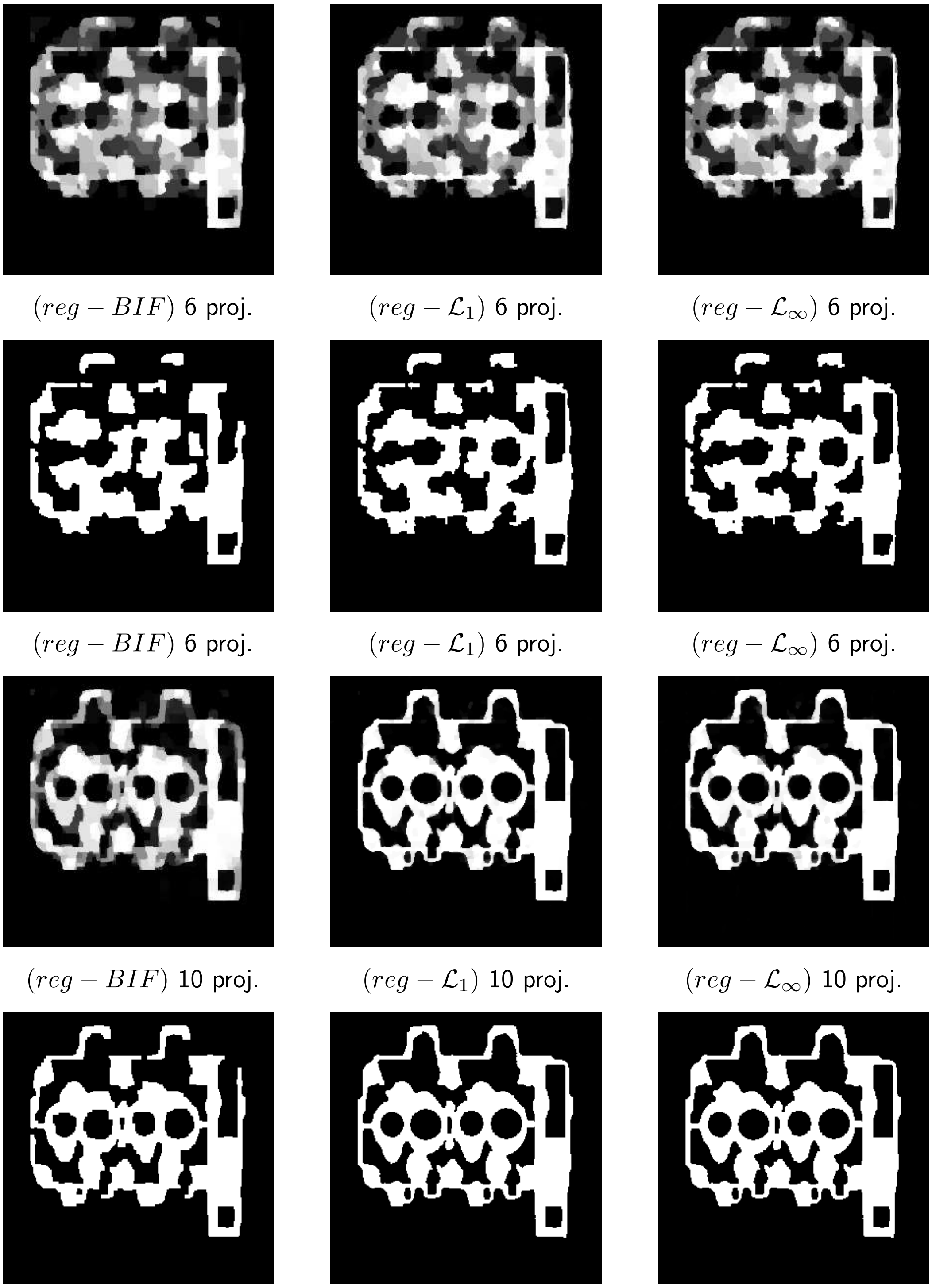
$(reg - \mathcal{L}_\infty)$ 8 proj.

$(reg - BIF)$ 8 proj.

$(reg - \mathcal{L}_1)$ 8 proj.

$(reg - \mathcal{L}_\infty)$ 8 proj.

Figure 4.10: Reconstructions of the *many ellipses* image without noise. Throughout these experiments the regularization parameter α was set to 0.5 for $(reg-BIF)$ and $(reg-\mathcal{L}_1)$ and in case of $(reg-\mathcal{L}_\infty)$ to 0.001.



$(reg - BIF)$ 6 proj. $(reg - \mathcal{L}_1)$ 6 proj. $(reg - \mathcal{L}_\infty)$ 6 proj.

$(reg - BIF)$ 6 proj. $(reg - \mathcal{L}_1)$ 6 proj. $(reg - \mathcal{L}_\infty)$ 6 proj.

$(reg - BIF)$ 10 proj. $(reg - \mathcal{L}_1)$ 10 proj. $(reg - \mathcal{L}_\infty)$ 10 proj.

$(reg - BIF)$ 10 proj. $(reg - \mathcal{L}_1)$ 10 proj. $(reg - \mathcal{L}_\infty)$ 10 proj.

Figure 4.11: Reconstructions of the *cylinders* image without noise. Throughout these experiments the regularization parameter α was set to 0.5 for $(reg - BIF)$ and $(reg - \mathcal{L}_1)$ and in case of $(reg - \mathcal{L}_\infty)$ to 0.001.

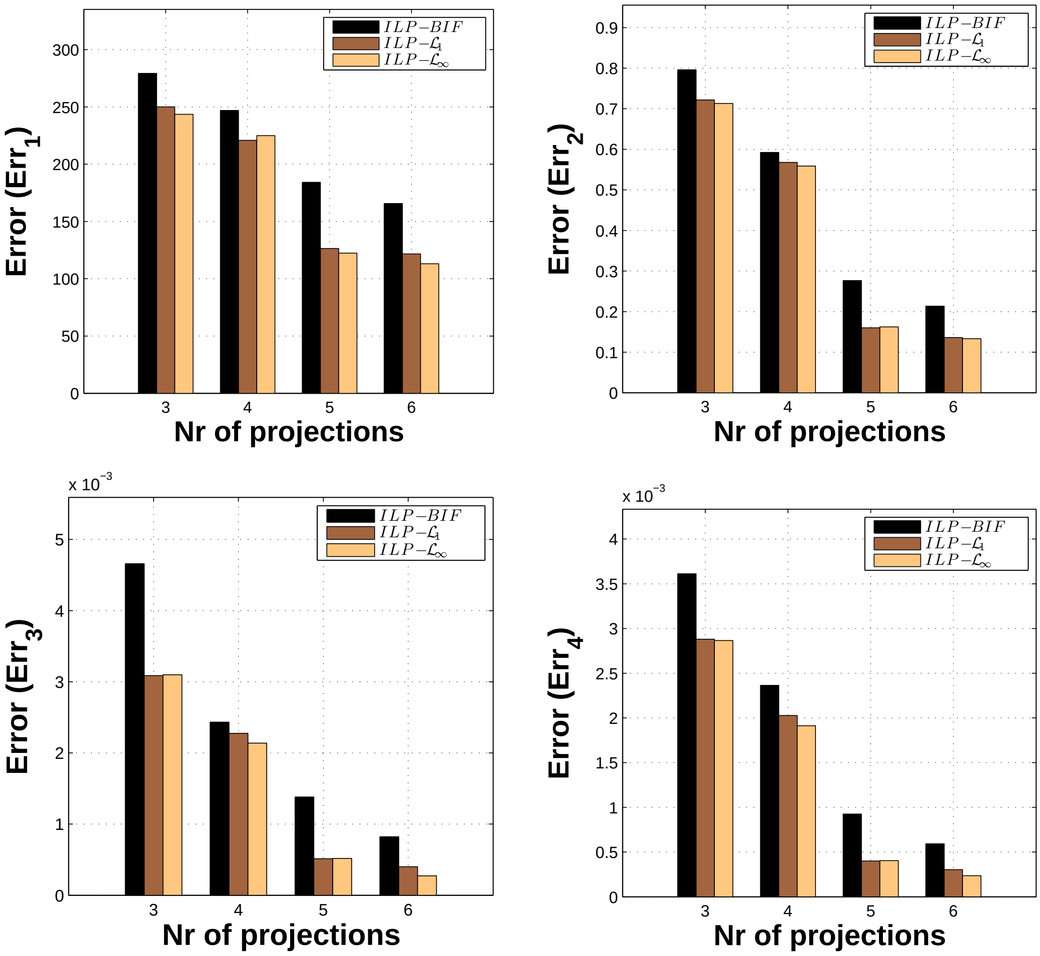


Figure 4.12: Error measurements, section 3.4, for the *turbine* experiment without noise.

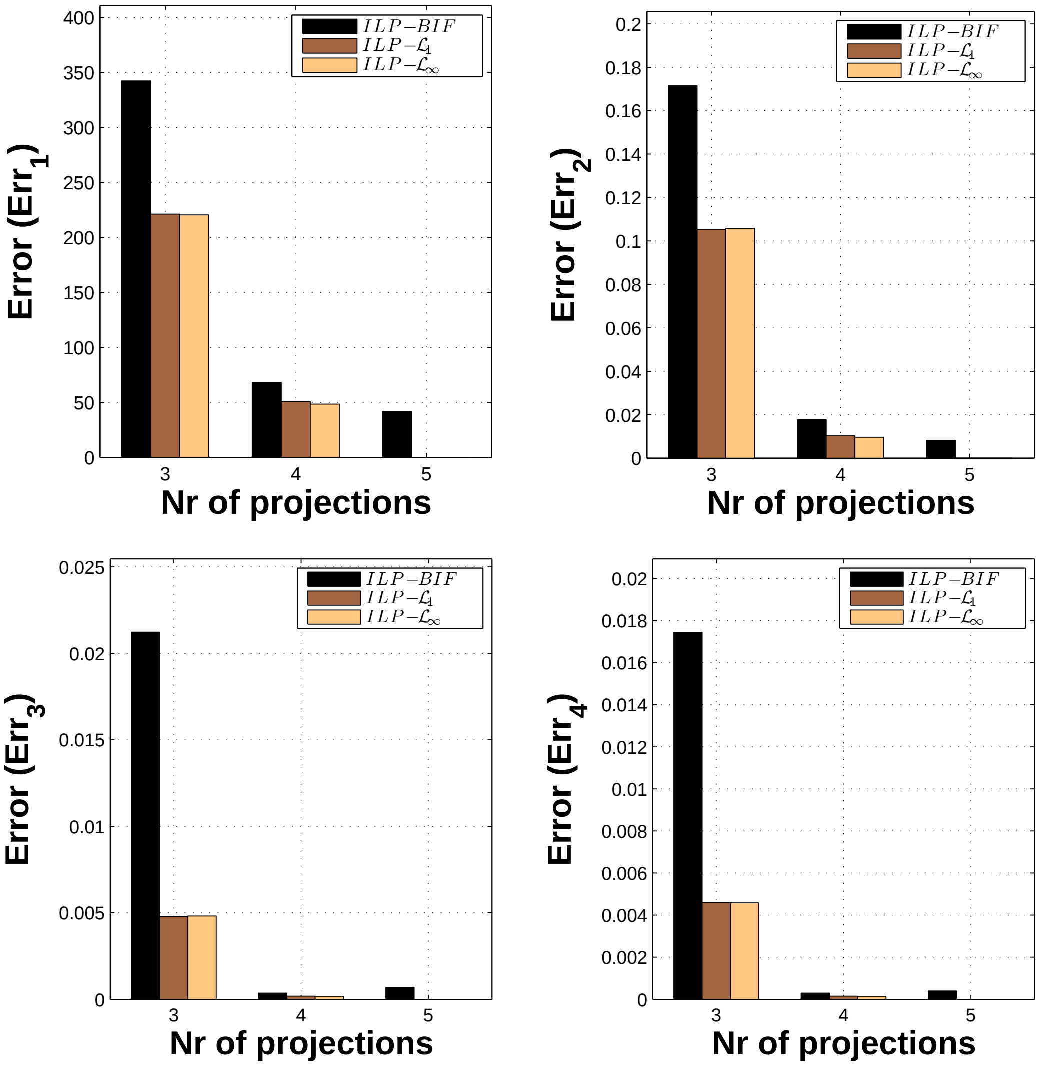


Figure 4.13: Error measurements, section 3.4, for the *single object* experiment without noise.

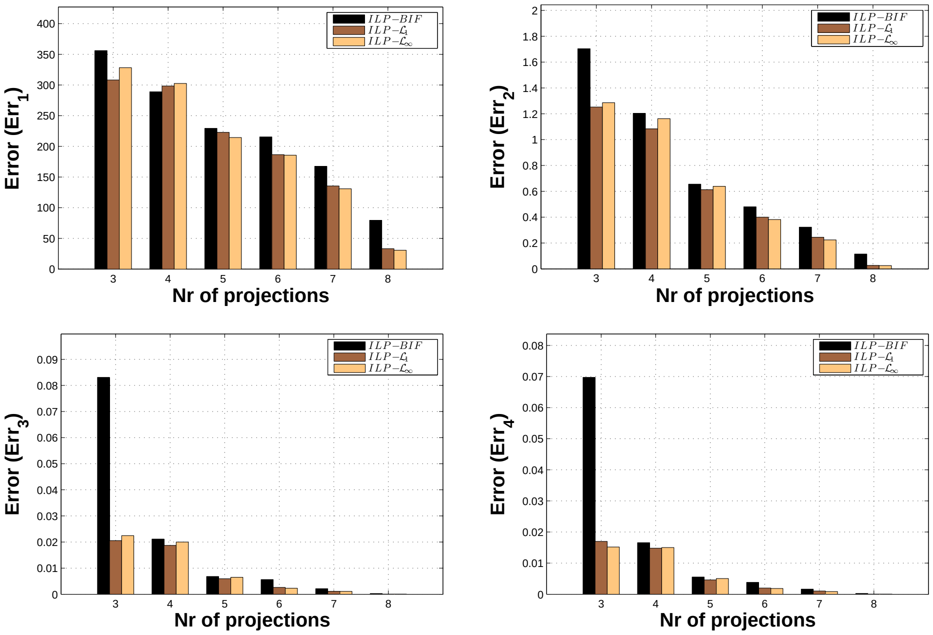


Figure 4.14: Error measurements, section 3.4, for the *many ellipses* experiment without noise.

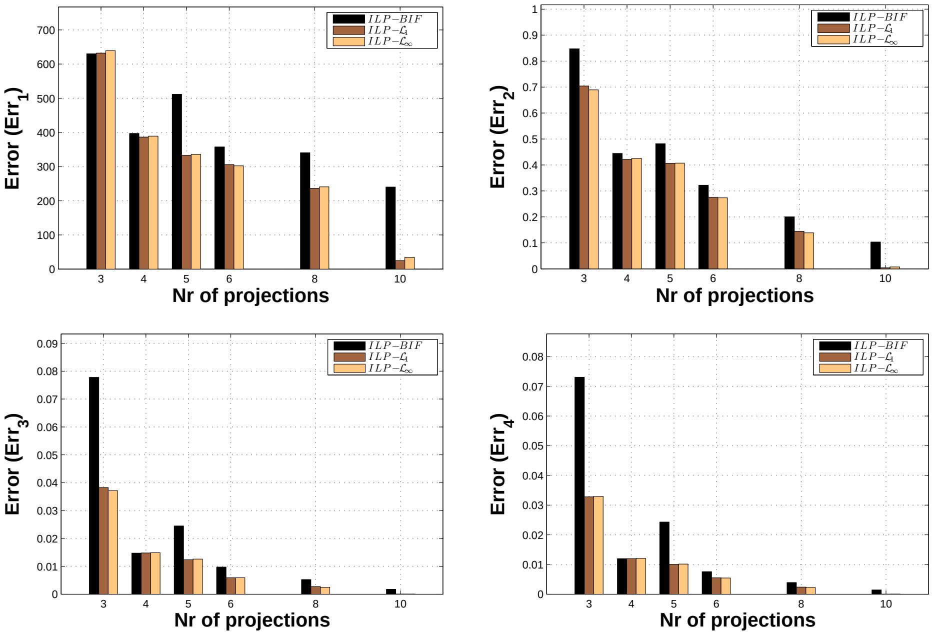
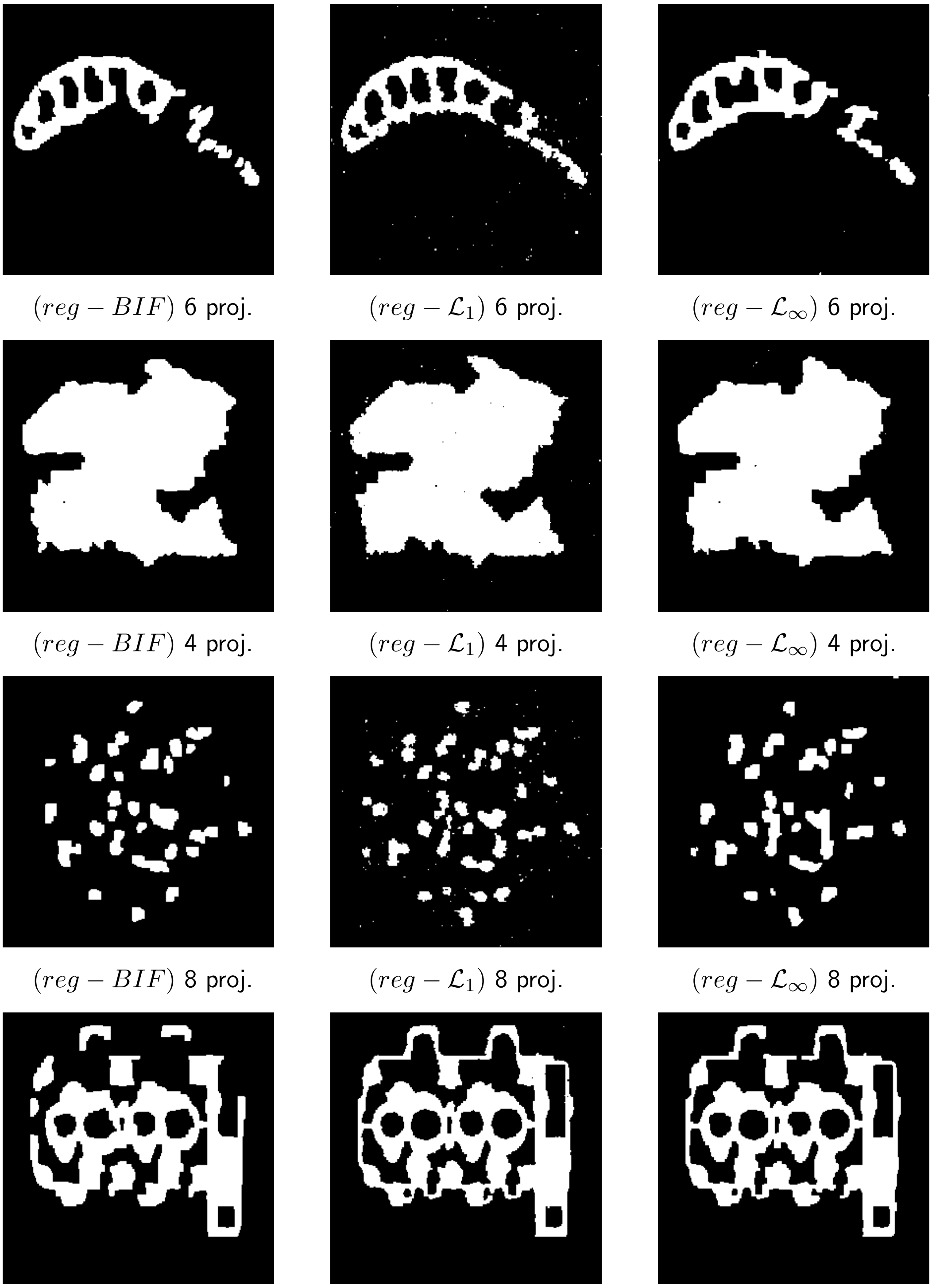


Figure 4.15: Error measurements, section 3.4, for the *cylinders* experiment without noise.



$(reg - BIF)$ 6 proj. $(reg - \mathcal{L}_1)$ 6 proj. $(reg - \mathcal{L}_\infty)$ 6 proj.

$(reg - BIF)$ 4 proj. $(reg - \mathcal{L}_1)$ 4 proj. $(reg - \mathcal{L}_\infty)$ 4 proj.

$(reg - BIF)$ 8 proj. $(reg - \mathcal{L}_1)$ 8 proj. $(reg - \mathcal{L}_\infty)$ 8 proj.

$(reg - BIF)$ 10 proj. $(reg - \mathcal{L}_1)$ 10 proj. $(reg - \mathcal{L}_\infty)$ 10 proj.

Figure 4.16: Reconstructions results from noisy projections, $\mathcal{N}(\mu := 0, \sigma)$ added to projection vector $\mathbf{b}$: For $(reg - BIF \mid reg - \mathcal{L}_1 \mid reg - \mathcal{L}_\infty)$, α was set to (1.0 | 1.0 | 0.005) in case of *turbine* $(\sigma := 3.0)$, *single object* $(\sigma := 3.0)$,*many ellipses* $(\sigma := 3.0)$, and to (0.75 | 0.75 | 0.0025) for *cylinders* $(\sigma := 1.0)$.

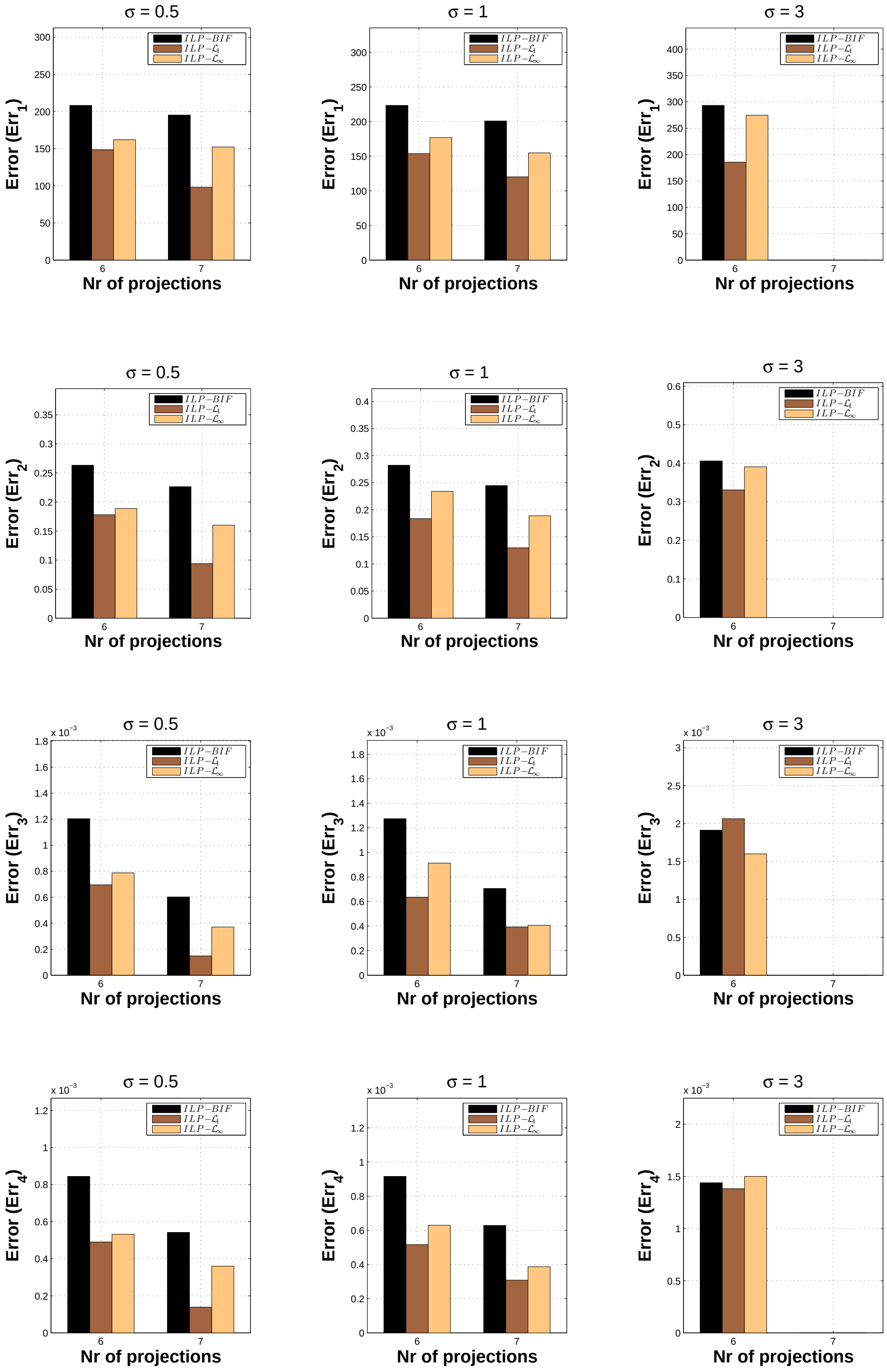


Figure 4.17: Error measurements (noisy projections): *turbine* (the reconstructions from 7 projections were not performed for $\sigma := 3.0$).

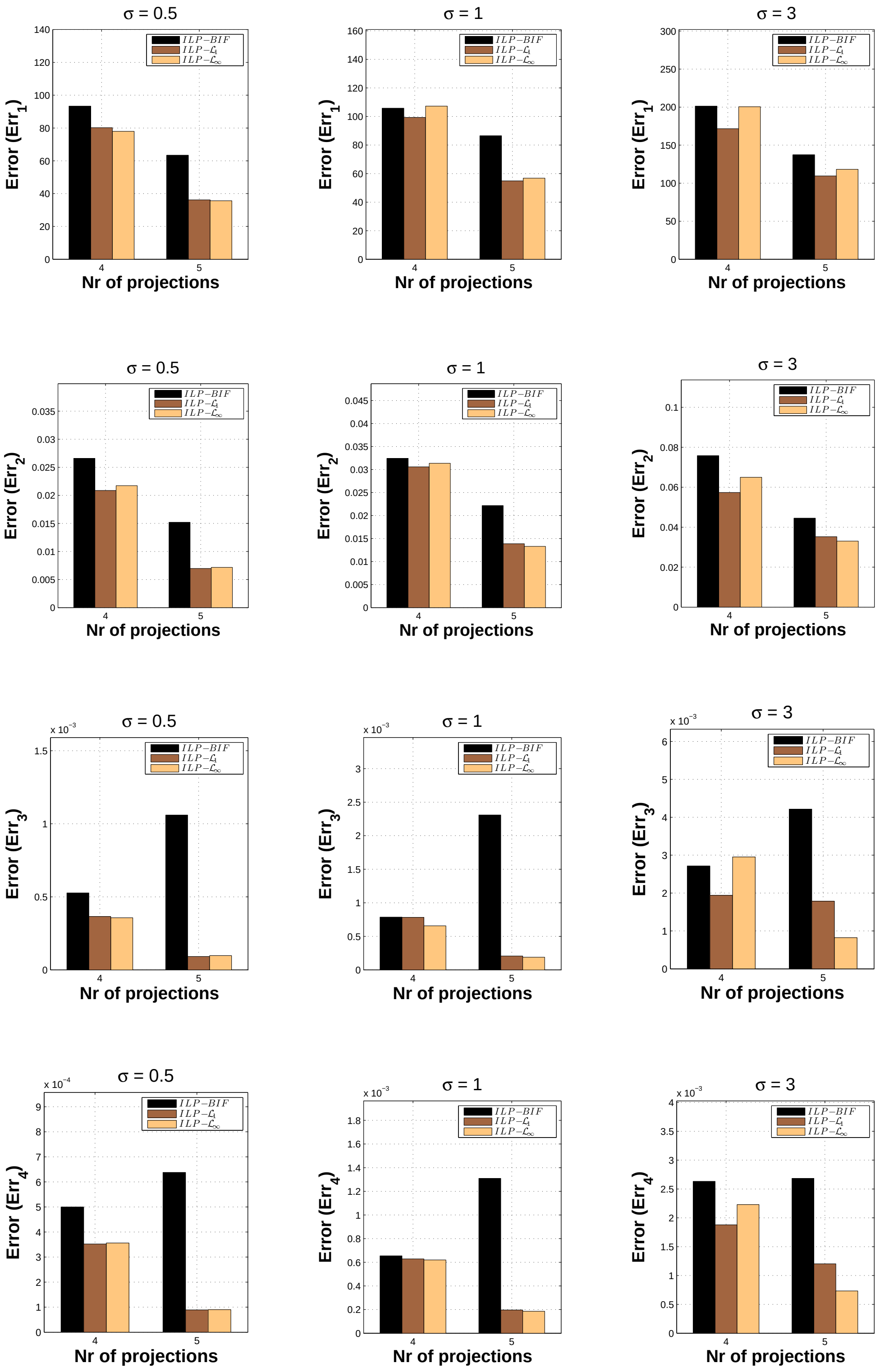


Figure 4.18: Error measurements (noisy projections): *single object*.

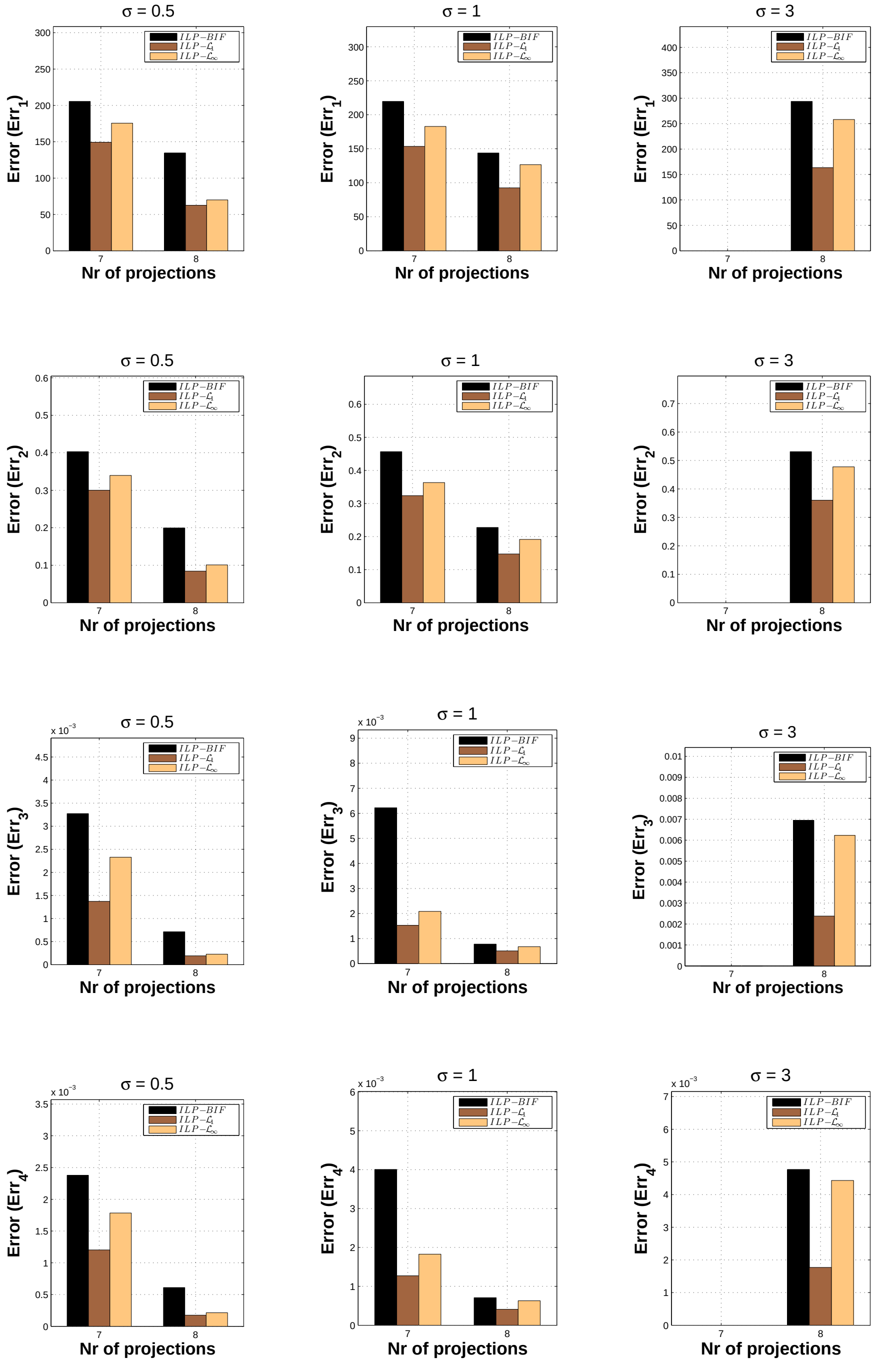


Figure 4.19: Error measurements from noisy projections: *many ellipses* (reconstructions from 7 projections were not performed for $\sigma := 3.0$). Similar to the other results obtained from noisy projections, the $(reg - \mathcal{L}_1)$ approach typically yields best results with respect to Err_1, Err_2, and Err_4. In case of the Err_3 measure there is no clear favorite.

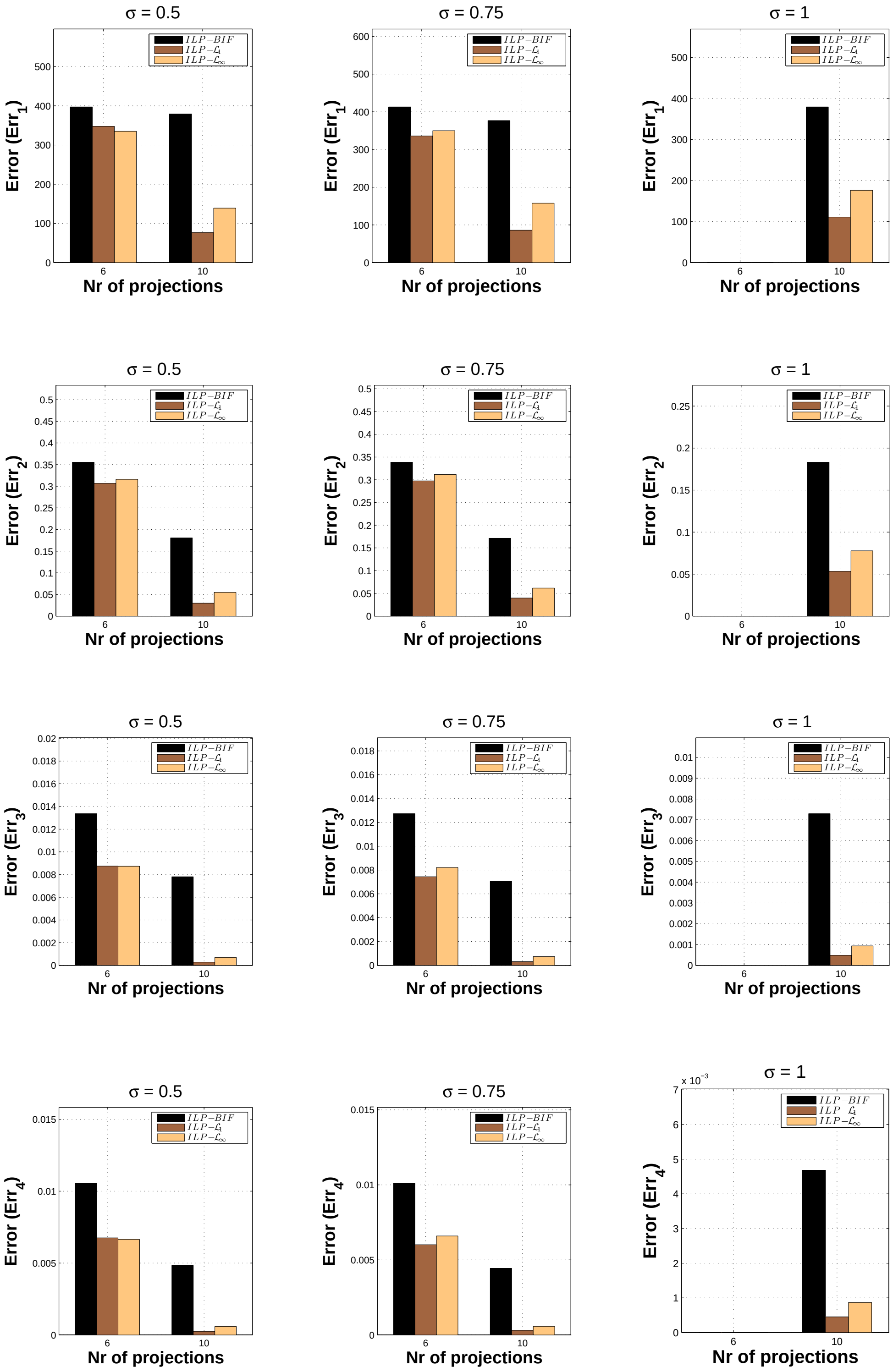


Figure 4.20: Error measurements of the experiments from noisy projections: *cylinders* (the reconstructions from 6 projections were not performed for $\sigma := 1.0$).

5 DC Framework

In the previous chapter we investigated different linear programming approaches which solve a relaxed form of the originally binary constrained reconstruction problem. As this yields only a fractional approximation, i.e. $\mathbf{x} \in [0,1]^n$, a postprocessing step is necessary in order to round the fractional solution to $\mathbf{x} \in \{0,1\}^n$. At this, the drawbacks of this approach are that there is no interference between the rounding and the optimization process and that the binary constraint is not further enforced within the actual optimization process. We address these disadvantages of the former approaches in the present chapter in which we systematically develop a mathematically sound optimization framework that implicitly performs the rounding step and is guaranteed to converge.

Additionally, our optimization framework is quite general and extensible, as for instance the approaches from the former chapter, $(reg{-}BIF)$, $(reg{-}\mathcal{L}_1)$, and $(reg{-}\mathcal{L}_\infty)$, can be "plugged-in", except $(reg{-}FSSV)$. In case of $(reg{-}FSSV)$ the equality constraint is troublesome, e.g. rounding obviously contradicts the equality constraint if there exists no feasible binary solution at all. However, the framework is by no means restricted to linear programming but any other convex constrained optimization technique is just fine. For this reason, we also use quadratic programming and demonstrate its expandability in a scenario where the reconstruction process involves the estimation of an unknown parameter, section 5.5. We therefore include the well-known expectation-maximization (EM) algorithm into our framework and remark that under certain circumstances the M-step can be solved analytically. This case has been considered in [123] where it is assumed that the absorption parameters are not exactly 0 and 1 and must be estimated during the reconstruction process. The situation dealt with in section 5.5 is, however, different as there is no analytical solution of the M-step and, thus, it must be numerically approximated. From this perspective, the approach taken here is more general and should, therefore, be applicable to an even larger class of optimization problems.

5.1 Convex and Concave Functions

In mathematics, convex and concave functions constitute a class of functions which are exceptionally valuable for optimization since global optimality instantly follows from local optimality and the uniqueness of the global optimum follows from strict convexity or strict concavity. This allows to use elaborate local optimization techniques and still to obtain global optimal solutions.

As a few basic facts from convex analysis are necessary concerning our work we provide the necessary definitions and details in this section. However, excellent introductions on the subject can be found in [111] and [18].

Throughout this chapter we shall assume S to be a convex set, i.e. for any two points $\mathbf{u}, \mathbf{v} \in S$ the line segment defined by $(1-\lambda)\ \mathbf{u} + \lambda\ \mathbf{s}$, $\lambda \in [0,1]$, belongs to S. Let f be a function, $f : S \rightarrow \overline{\mathbb{R}}$, then f is convex if and only if $f((1-\lambda)\ \mathbf{x} + \lambda\ \mathbf{y}) \leq (1-\lambda)\ f(\mathbf{x}) + \lambda\ f(\mathbf{y})$ for all $\lambda \in [0,1]$ and $\mathbf{x}, \mathbf{y} \in S$. Conversely, f is concave if and only if $-f$ is convex. Further, if the inequality holds strict f is called strictly convex or strictly concave respectively. A convex function f is called proper if its effective domain $\operatorname{dom} f := \{\mathbf{x} \in S \mid f(\mathbf{x}) < +\infty\}$ is non-empty and the restriction of f to $\operatorname{dom} f$ is finite.

Based on differentiability convexity and concavity can be further characterized, assume therefore f to be differentiable on $\operatorname{dom} f$. From the first-order condition we have, f is convex if and only if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^{\top}(\mathbf{y} - \mathbf{x}), \forall\ \mathbf{x}, \mathbf{y} \in \operatorname{dom} f \tag{5.1}$$

and, thus, the first-order approximation of f is a global underestimator and, vice versa, for concave f a global overestimator.

If f is twice differentiable on $\operatorname{dom} f$ then the second-order condition yields, f is convex if and only if the Hessian of f is positive semidefinite,

$$\nabla^2 f(\mathbf{x}) \succeq 0, \forall \mathbf{x} \in \operatorname{dom} \mathrm{f}, \tag{5.2}$$

and f is strictly convex if positive definiteness, $\nabla^2 f(\mathbf{x}) \succ 0$, holds for all $\mathbf{x} \in \operatorname{dom} \mathrm{f}$.

In case f is non-differentiable at $\mathbf{x}_0$ the subgradient of f at $\mathbf{x}_0$ is a vector $\mathbf{v}$ such that

$$f(\mathbf{x}) \geq f(\mathbf{x}_0) + \langle \mathbf{v}, \mathbf{x} - \mathbf{x}_0 \rangle\ , \ \forall \mathbf{x}, \tag{5.3}$$

and can, thus, be considered as a generalized gradient. The set of all subgradients of f at $\mathbf{x}_0$ constitutes the subdifferential $\partial f(\mathbf{x}_0)$ of f at $\mathbf{x}_0$.

The conjugate function of f, not necessarily convex, is defined by

$$f^*(\mathbf{y}) := \sup_{\mathbf{x}\in S} \{\langle \mathbf{x}, \mathbf{y}\rangle - f(\mathbf{x})\} = - \inf_{\mathbf{x}\in S} \{f(\mathbf{x}) - \langle \mathbf{x}, \mathbf{y}\rangle\} \tag{5.4}$$

and is also referred to as the Legendre-Fenchel transform of f. This establishes a one to one correspondence in the class of proper, lower-semicontinuous, and convex functions, as stated by the next theorem.

Theorem 8 ([111]). *Let f be proper, lower-semicontinuous, and convex then it holds that*

$$f = f^{**} = (f^*)^*.$$

Given a convex function f, we start with rearranging the definition of subdifferentials in order to derive equation (5.5) and (5.6),

$$\begin{aligned}\partial f(\mathbf{x}_0) &= \{\mathbf{y} \mid f(\mathbf{x}) \geq f(\mathbf{x}_0) + \langle \mathbf{y}, \mathbf{x} - \mathbf{x}_0\rangle,\ \forall \mathbf{x}\}\\ &= \{\mathbf{y} \mid f(\mathbf{x}) \geq f(\mathbf{x}_0) + \langle \mathbf{y}, \mathbf{x}\rangle - \langle \mathbf{y}, \mathbf{x}_0\rangle,\ \forall \mathbf{x}\}\\ &= \{\mathbf{y} \mid \langle \mathbf{y}, \mathbf{x}_0\rangle - f(\mathbf{x}_0) \geq \langle \mathbf{y}, \mathbf{x}\rangle - f(\mathbf{x}),\ \forall \mathbf{x}\}.\end{aligned}$$

Note, that the inequality becomes equal if $\mathbf{x} = \mathbf{x}_0$ and, thus, we write

$$\begin{aligned}\partial f(\mathbf{x}_0) &= \{\mathbf{y} \mid \langle \mathbf{y}, \mathbf{x}_0\rangle - f(\mathbf{x}_0) = \sup_{\mathbf{x}\in S}\{\langle \mathbf{y}, \mathbf{x}\rangle - f(\mathbf{x})\}\}\\ &= \{\mathbf{y} \mid \langle \mathbf{y}, \mathbf{x}_0\rangle - f(\mathbf{x}_0) = f^*(\mathbf{y})\}\\ &= \{\mathbf{y} \mid \langle \mathbf{y}, \mathbf{x}_0\rangle - f^*(\mathbf{y}) = f(\mathbf{x}_0)\}.\end{aligned}$$

By means of conjugate functions, the equation above reads

$$f^{**}(\mathbf{x}_0) := \sup_{y\in S}\{\langle \mathbf{y}, \mathbf{x}_0\rangle - f^*(\mathbf{y})\} \overset{!}{=} f(\mathbf{x}_0)$$

where we have

$$f^{**}(\mathbf{x}_0) = f(\mathbf{x}_0)$$

due to theorem 8. Consequently, $\mathbf{y}$ is a subgradient of f at $\mathbf{x}_0$ if and only if

$$\partial f(\mathbf{x}_0) = \operatorname*{argmax}_{\mathbf{y}\in S}\{\langle \mathbf{y}, \mathbf{x}_0\rangle\ - f^*(\mathbf{y})\} \tag{5.5}$$

and similarly we derive

$$\begin{aligned}\partial f^*(\mathbf{y}_0) &= \{\mathbf{x} \mid f^*(\mathbf{y}) \geq f(\mathbf{y}_0) + \langle \mathbf{x}, \mathbf{y} - \mathbf{y}_0\rangle,\ \forall \mathbf{y}\}\\ &= \operatorname*{argmax}_{\mathbf{x}\in S}\{\langle \mathbf{y}_0, \mathbf{x}\rangle\ - f(\mathbf{x})\}. \end{aligned}\tag{5.6}$$

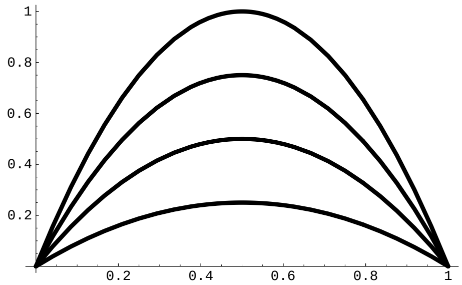


Figure 5.1: Concave regularizer: Considering a single variable x, the plot shows the term $\mu\, x\, (1-x)$, $0 \leq x \leq 1$, for increasing values of μ. During the minimization process, values of x within $(0,1)$ become more and more penalized as μ increases.

5.2 Convex-Concave Regularization

Convex-concave regularization is the crucial step that is necessary to include the binary constraints into the overall optimization process. Let us, therefore, consider binary optimization problems of the following type,

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \{0,1\}^n, \tag{5.7}$$

where $f(\mathbf{x})$ is assumed to be a convex function. Obviously, problem (5.7) is equivalent to

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \quad \text{subject to} \;\; & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \\ & \mathbf{x}^\top (\mathbf{1}-\mathbf{x}) = 0 \end{aligned} \tag{5.8}$$

where $\mathbf{x}^\top (\mathbf{1}-\mathbf{x}) = 0$ describes the binary constraint and entails difficulties on the optimization of (5.8). Corresponding to problem (5.8) we obtain the Lagrangian function $E(\mathbf{x};\mu)$ by adding the tedious constraints as concave penalty term to the objective function,

$$k(\mathbf{x};\mu) := \frac{\mu}{2}\, \langle \mathbf{x}\,, \mathbf{1}-\mathbf{x} \rangle \tag{5.9}$$

$$\min_{\mathbf{x}\in\mathbb{R}^n} E(\mathbf{x};\mu) :=\; g(\mathbf{x}) + k(\mathbf{x};\mu) \quad \text{subject to} \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \tag{5.10}$$

where μ penalizes non-binary components in $\mathbf{x}$. If $\mathbf{x}$ is binary then the penalty term $k(\mathbf{x};\mu)$ vanishes and $E(\mathbf{x};\mu)$ equals the original objective function $f(\mathbf{x})$.

The key of our optimization approach is to start from the convex solution of problem (5.10), i.e. $\mu = 0$, and then to gradually include the binary constraint by increasing μ, figure 5.1. This is further justified by theorem 9 which explains the connection between problem (5.7) and (5.10).

Theorem 9 ([57, 69]). *Suppose that $f(\mathbf{x})$ is twice continuously differentiable on $[0,1]$, then there exists $\mu^* \in \mathbb{R}$ such that for all $\mu > \mu^*$*

(i) problem (5.7) and the minimization of $E(\mathbf{x};\mu)$, equation (5.10), are equivalent.

(ii) $E(\mathbf{x};\mu)$ is concave on $[0,1]^n$.

This connection between integer programming and concave maximization is well-known. Theorem 9(i) states that for sufficiently large μ the global optima of problem (5.7) and (5.10) coincide. However, due to theorem 9(ii) not only the global optimum satisfies the binary constraints but any vertex of $[0,1]^n$ is a local minimum and, thus, the profit of our approach depends on how severely local minima affect the concave minimization problem.

5.3 DC Programming

The presence of convex and concave terms in functional $E(\mathbf{x};\mu)$, equation (5.10), suggests a mathematical programming method concerned with the minimization of the difference of two convex functions (difference of convex functions $\sim$ d.c. programming). Such approaches are typically used in connection with global optimization [69] which is, however, only permissive for small scale problems. Our focus, in contrast, is on the local optimization of large scale problems which has been considered in context of d.c. programming in [105, 106].

Let S be a convex set and $f : S \to \overline{\mathbb{R}}$ be a lower semicontinuous (lsc), proper, and convex function, then f is said to be d.c. decomposable if and only if there exist two lsc., proper, and convex functions $g, h : S \to \overline{\mathbb{R}}$ such that

$$f(\mathbf{x}) = g(\mathbf{x}) - h(\mathbf{x}). \tag{5.11}$$

Let us now consider the following optimization problem where the objective function $f(\mathbf{x})$ is assumed to be d.c. decomposable

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} g(\mathbf{x}) - h(\mathbf{x}) \tag{5.12}$$

and

$$\operatorname{dom} g \subset \operatorname{dom} h \quad \text{and} \quad \operatorname{dom} h^* \subset \operatorname{dom} g^* \tag{5.13}$$

is required in order to have a finite minimum. Problem (5.12) is equivalent to the general d.c. program

$$\inf_{\mathbf{x}}\{g(\mathbf{x}) - h(\mathbf{x})\} \tag{5.14}$$

for which its dual is obtained by means of conjugate functions,

$$\begin{aligned}\inf_{\mathbf{x}} \{g(\mathbf{x}) - h(\mathbf{x})\} &= \inf_{\mathbf{x}} \left\{g(\mathbf{x}) - \sup_{\mathbf{y}} \{\langle \mathbf{x}, \mathbf{y}\rangle - h^*(\mathbf{y})\}\right\} \\ &= \inf_{\mathbf{x}} \left\{g(\mathbf{x}) + \inf_{\mathbf{y}} \{h^*(\mathbf{y}) - \langle \mathbf{x}, \mathbf{y}\rangle\}\right\} \\ &= \inf_{\mathbf{x}} \inf_{\mathbf{y}} \ \{g(\mathbf{x}) - [\ \langle \mathbf{x}, \mathbf{y}\rangle - h^*(\mathbf{y})]\} \\ &= \inf_{\mathbf{y}} \left\{h^*(\mathbf{y}) + \inf_{\mathbf{x}} \{g(\mathbf{x}) - \langle \mathbf{x}, \mathbf{y}\rangle\}\right\} \\ &= \inf_{\mathbf{y}} \left\{h^*(\mathbf{y}) - \sup_{\mathbf{x}} \{\langle \mathbf{x}, \mathbf{y}\rangle - g(\mathbf{x})\}\right\} \\ &= \inf_{\mathbf{y}} \{h^*(\mathbf{y}) - g^*(\mathbf{y})\}. \end{aligned} \tag{5.15}$$

In order to optimize problem (5.14) we adopt the following primal-dual subgradient method, algorithm 7, from [105, 106]

Algorithm 7 DC Algorithm (DCA)

```
Choose $\mathbf{x}^0 \in \operatorname{dom} g$ arbitrary
for k = 0,1,... do
    $\mathbf{y}^k \in \partial h(\mathbf{x}^k)$                (y-step)
    $\mathbf{x}^{k+1} \in \partial g^*(\mathbf{y}^k)$            (x-step)
end for
```

where the investigation of the algorithm in [106] includes the following properties:

Theorem 10 ([106]). *Assume $g, h : \mathbb{R}^n \to \overline{\mathbb{R}}$ to be proper, lower-semicontinuous and convex, and $\operatorname{dom} g \subset \operatorname{dom} h$, $\operatorname{dom} h^* \subset \operatorname{dom} g^*$. Then*

(i) the sequences $\{\mathbf{x}^k\}, \{\mathbf{y}^k\}$ are well-defined, cf. x- and y-step of algorithm 7.

(ii) the sequence $\{g(\mathbf{x}^k) - h(\mathbf{x}^k)\}$ is decreasing and so is $\{f(\mathbf{x}^k)\}$.

(iii) every limit point $\mathbf{x}^*$ of $\{\mathbf{x}^k\}$ is a critical point of $g - h$ which means that $\mathbf{0} \in \partial f(\mathbf{x}^*)$.

Starting from an initial point $\mathbf{x}^0 \in \operatorname{dom} g$, the algorithm develops sequences $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ which converge to a locally optimal duality pair of the primal (5.14) and the dual problem (5.15). Given $\mathbf{x}^k \in \partial g^*(\mathbf{y}^{k-1})$, the y-step replaces $g^*(\mathbf{y})$ in the dual problem with its affine minorization at $\mathbf{y}^{k-1}$,

$$\inf_{\mathbf{y}}\{h^*(\mathbf{y}) - g^*(\mathbf{y})\} \ \leq \ \inf_{\mathbf{y}}\{h^*(\mathbf{y}) - \left[g^*(\mathbf{y}^{k-1}) + \langle \mathbf{x}^k, \mathbf{y} - \mathbf{y}^{k-1}\rangle\right]\}\ ,$$

and, thus, amounts, by virtue of equation (5.5), in a convex program

$$\mathbf{y}^k \in \partial h(\mathbf{x}^k) = \underset{\mathbf{y}}{\operatorname{argmin}} \{h^*(\mathbf{y}) - \Big[g^*(\mathbf{y}^{k-1}) + \langle \mathbf{x}^k, \mathbf{y} - \mathbf{y}^{k-1} \rangle \Big]\} \tag{5.16}$$

$$= \underset{\mathbf{y}}{\operatorname{argmin}} \{h^*(\mathbf{y}) - \langle \mathbf{x}^k, \mathbf{y} \rangle\}. \tag{5.17}$$

In return, the x-step uses $\mathbf{y}^k \in \partial h(\mathbf{x}^k)$ in order to have an affine minorization of the primal problem at $\mathbf{x}^k$,

$$\inf_{\mathbf{x}} \{g(\mathbf{x}) - h(\mathbf{x})\} \ \leq \ \inf_{\mathbf{x}} \{g(\mathbf{x}) - \Big[h(\mathbf{x}^k) + \langle \mathbf{y}^k, \mathbf{x} - \mathbf{x}^k \rangle \Big]\} \ ,$$

and similarly leads, by means of equation (5.6), to a convex program

$$\mathbf{x}^{k+1} \in \partial g^*(\mathbf{x}^k) = \underset{\mathbf{x}}{\operatorname{argmin}} \{g(\mathbf{x}) - \Big[h(\mathbf{x}^k) + \langle \mathbf{y}^k, \mathbf{x} - \mathbf{x}^k \rangle \Big]\} \tag{5.18}$$

$$= \underset{\mathbf{x}}{\operatorname{argmin}} \{g(\mathbf{x}) - \langle \mathbf{y}^k, \mathbf{x} \rangle\}. \tag{5.19}$$

5.4 Iterating Linear Programs (ILP)

We supplement the linear programming approaches from the previous chapter with the convex-concave regularization from section 5.2 and proceed with the derivation of the d.c. based reconstruction framework for binary tomography.

5.4.1 Approach

Let us consider the general form of a reconstruction approach based on linear programming with the concave regularizer, equation (5.9), added to the objective,

$$\min_{\mathbf{x} \in [0,1]^n, \mathbf{s} \in \mathbb{R}^m_{\geq 0}, \{z_{\langle i,j \rangle}\}} \tilde{\mathbf{c}}^\top \tilde{\mathbf{x}} + \frac{\alpha}{2} \sum_{\langle i,j \rangle} z_{\langle i,j \rangle} + \frac{\mu}{2} \langle \mathbf{x}, \mathbf{1} - \mathbf{x} \rangle \tag{5.20}$$

$$\text{subject to} \quad \tilde{\mathbf{A}}\, \tilde{\mathbf{x}} \leq \tilde{\mathbf{b}}\,, \quad z_{\langle i,j \rangle} \geq x_i - x_j\ ,\ z_{\langle i,j \rangle} \geq x_j - x_i.$$

At this, we define $\tilde{\mathbf{x}}^\top := (\mathbf{x}^\top, \mathbf{s}^\top)$ such that the problem description in equation (5.20) fits the approaches $(reg - BIF)$, $(reg - \mathcal{L}_1)$, and $(reg - \mathcal{L}_\infty)$ from chapter 4.

Further, we put

$$\mathbf{z} := (\tilde{\mathbf{x}}^\top, \dots, z_{\langle i,j \rangle}, \dots)^\top \tag{5.21}$$

and rewrite all constraints of (5.20)

$$0 \leq x_i \leq 1\ , \quad \tilde{\mathbf{A}}\,\tilde{\mathbf{x}} \leq \tilde{\mathbf{b}}\,, \quad z_{\langle i,j\rangle} \geq x_i - x_j\ ,\ z_{\langle i,j\rangle} \geq x_j - x_i$$

as

$$\hat{\mathbf{A}}\,\mathbf{z} \leq \hat{\mathbf{b}}\ . \tag{5.22}$$

Using the notation

$$\delta_C(\mathbf{z}) := \begin{cases} 0 & , \mathbf{z} \in C \\ +\infty & , \mathbf{z} \notin C \end{cases} \tag{5.23}$$

for the indicator function of a convex set C, problem (5.20) reads:

$$\min_{\mathbf{z}} f(\mathbf{z})\ ,$$

where, confer definition (5.21),

$$f(\mathbf{z}) := \tilde{\mathbf{c}}^\top \tilde{\mathbf{x}} + \frac{\alpha}{2}\sum_{\langle i,j\rangle} z_{\langle i,j\rangle} + \frac{\mu}{2}\langle \mathbf{x}, \mathbf{1} - \mathbf{x}\rangle + \delta_K(\hat{\mathbf{b}} - \hat{\mathbf{A}}\mathbf{z})\ , \tag{5.24}$$

and K is the standard cone of non-negative vectors.

Although the d.c. decomposition of a function is not unique in general a straightforward decomposition in case of f, equation (5.24), is readily found by

$$g(\mathbf{z}) = \tilde{\mathbf{c}}^\top \tilde{\mathbf{x}} + \frac{\alpha}{2}\sum_{\langle i,j\rangle} z_{\langle i,j\rangle} + \delta_K(\hat{\mathbf{b}} - \hat{\mathbf{A}}\mathbf{z})\ , \tag{5.25}$$

$$h(\mathbf{z};\mu) = -\frac{\mu}{2}\langle \mathbf{x}, \mathbf{1} - \mathbf{x}\rangle = \frac{\mu}{2}\langle \mathbf{x}, \mathbf{x} - \mathbf{1}\rangle\ . \tag{5.26}$$

Obviously, both functions $g(\mathbf{z})$ and $h(\mathbf{z};\mu)$ are convex and $g(\mathbf{z})$ is non-smooth because of δ_K. Note that this does not prevent the application of algorithm 7 from section 5.3 as smoothness of g or h is not required.

Furthermore, the assumptions of theorem 10 hold because of $\operatorname{dom} g \subset \operatorname{dom} h$, and $g^*(\mathbf{y}) = \sup_{\mathbf{z}} \big\{\langle \mathbf{z}, \mathbf{y}\rangle - g(\mathbf{z})\big\} < \infty$ for any finite vector $\mathbf{y}$. Hence, the y-step of algorithm 7 reads

$$\mathbf{y}^k = \nabla h(\mathbf{z}^k;\mu) = \mu(\mathbf{x}^k - \frac{1}{2}\mathbf{1}) \tag{5.27}$$

due to

$$\partial h(\mathbf{z}_0;\mu) = \big\{\nabla h(\mathbf{z}_0;\mu)\big\}$$

if h is differentiable [111].

Since g is proper, lower-semicontinuous, and convex, the x-step of algorithm 7 leads to

$$\mathbf{z}^{k+1} \in \operatorname*{argmin}_{\mathbf{z}} \{ g(\mathbf{z}) - \langle \mathbf{y}^k, \mathbf{z} \rangle \}$$

where we finally compute $\mathbf{z}^{k+1}$ by inserting $\mathbf{y}^k$ from equation (5.27) and by virtue of (5.25), (5.22), and (5.21) as minimizer of the following linear program:

$$\begin{aligned} \mathbf{z}^{k+1} := & \min_{\mathbf{x} \in [0,1]^n, \mathbf{s} \in \mathbb{R}^m_{\geq 0}, \{z_{\langle i,j \rangle}\}} \langle\, \tilde{\mathbf{c}} \,,\, \tilde{\mathbf{x}} \,\rangle \;+\; \frac{\alpha}{2} \sum_{\langle i,j \rangle} z_{\langle i,j \rangle} \;-\; \left\langle \mu \, (\mathbf{x}^k - \frac{1}{2}\mathbf{1}), \mathbf{x} \right\rangle \\ & \text{subject to} \quad \hat{\mathbf{A}}\, \mathbf{z} \leq \hat{\mathbf{b}}. \end{aligned} \tag{5.28}$$

The complete reconstruction algorithm, shown in algorithm 8, consists of two loops where the outer loop increases the penalty parameter μ and the inner loop executes the DC algorithm 7. The procedure terminates if all components of $\mathbf{x}$ are sufficiently close to a binarization, i.e. $\forall i, \; \min\{x_i, 1 - x_i\} < \varepsilon$.

Algorithm 8 DC-Framework with Linear Programming (ILP)

```
Require: Choose $\mathbf{z}^0 \in \operatorname{dom} g$ arbitrary.
Require: $0 < \Delta\mu$
Require: $0 < \epsilon$
  $\mu := 0$
  repeat
    $k := 0$
    repeat
      Compute $\mathbf{z}^{k+1}$ by means of the linear program (5.28) with current value of $\mu$.
      $k := k + 1$
    until $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2 < \epsilon$                    (DC-loop)
    $\mu := \mu + \Delta\mu$
  until $\nexists\, \mathbf{x}_i^k \in [\epsilon, 1 - \epsilon]$                  ($\mu$-loop)
```

Note that in case of $\mu := 0$, we minimize the original linear program from chapter 4, whereas $\mu > 0$ shifts the current iterate in the direction of the negative gradient of the "binarization" functional, equation (5.9). While this is an intuitively clear modification of the linear programming approaches from the former chapter, convergence of algorithm 8 is not obvious but is proven by theorem 10. In correspondence to the regularized linear programming approaches from chapter 4, we denote the approaches which include the convex-concave regularization by $(ILP - BIF)$, $(ILP - \mathcal{L}_1)$, and $(ILP - \mathcal{L}_\infty)$.

5.4.2 Numerical Evaluation

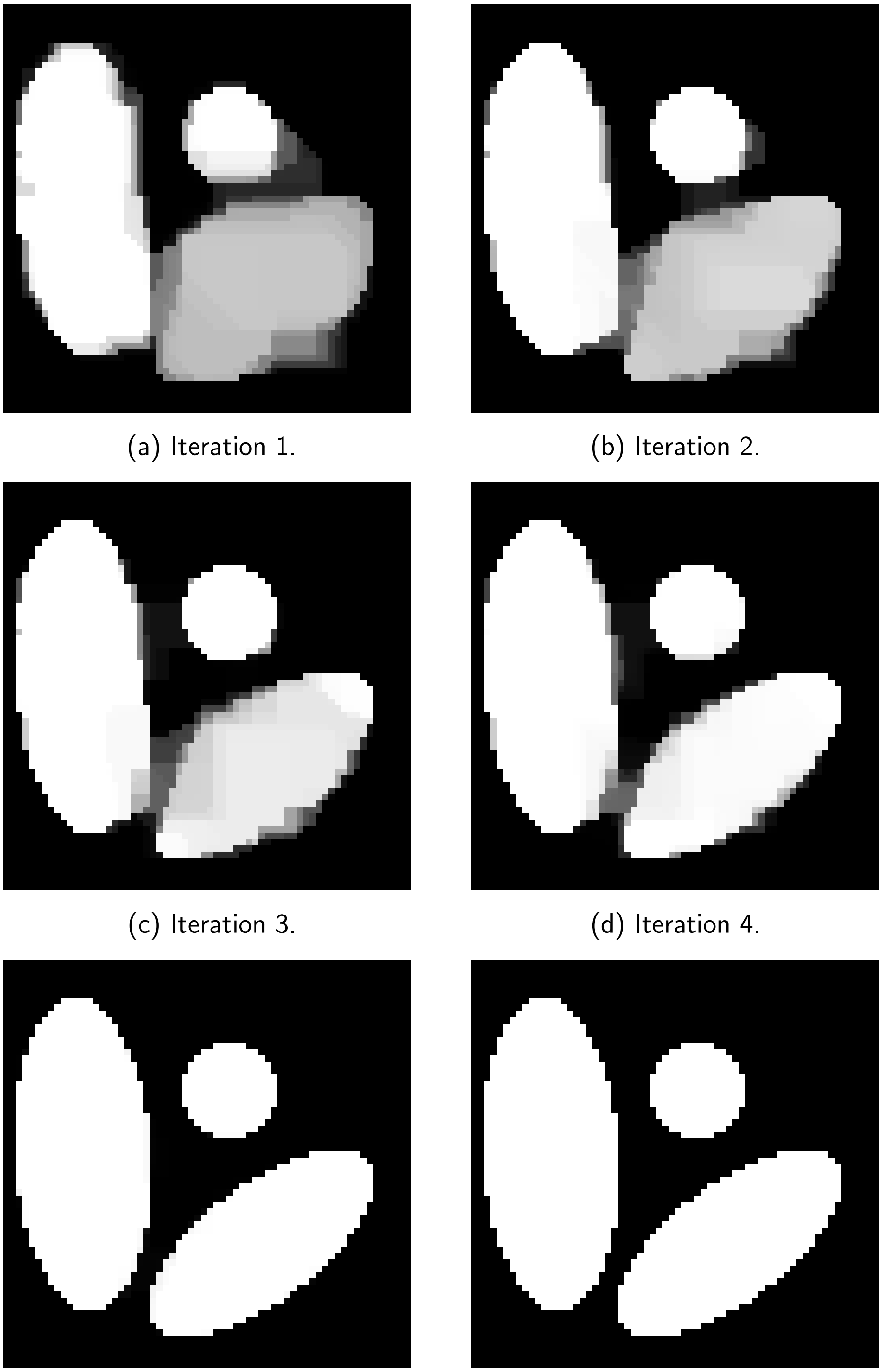
To provide an illustration of the convex-concave regularizer we present intermediate results in the figures 5.2 and 5.3 which were obtained with $(ILP - BIF)$. In both cases the images in (a) show the convex solution which equals the fractional solution found by $(reg - BIF)$, section 4. In the succeeding images (b)–(e), it can be seen how the individual components of the solution vector $\mathbf{x}$ are gradually pushed towards 0 or 1, as the concave regularizer, equation (5.9), becomes more and more involved, until all entries of $\mathbf{x}$ are binary, image (e). In view of the convex solution, shown in figure 5.3 (a), which is used in connection with $(reg - BIF)$ as input for the rounding procedure, it seems unlikely to achieve a decent reconstruction result with the $(reg - BIF)$ approach. In contrast, figure 5.3 (b)–(e) shows that the $(ILP - BIF)$ approach is still able to obtain a very nice reconstruction result, e.g. the ring-shaped object is closed during the iterations.

We evaluate the proposed ILP reconstruction methods with respect to the linear programming approaches from section 4 and simulated annealing, algorithm 3, in that we repeatedly solved the reconstruction problems from the previous chapter. The results in case of noiseless projections can be found in the figures 5.4–5.7 and in case of noisy projections in figure 5.12. It can be seen there that the ILP methods achieve better results compared to the linear programming approaches. Further, we obtained best results with $(ILP - \mathcal{L}_1)$ which usually performed better than simulated annealing and the other ILP approaches with respect to Err_1. For a comparison, we present all error plots similarly to the previous chapter in the figures 5.10–5.11 (noiseless projections) and figures 5.13–5.14 (noisy projections).

Concerning the number of iterations needed by the ILP approaches in order to achieve a binary solution, the figures 5.8–5.9 plot the number of undecided pixels, that is the amount of $x_i^k \in [\epsilon, 1 - \epsilon]$ in iteration k. As can be seen, the number of iterations becomes smaller with increasing number of projections and is typically smallest for $(ILP - \mathcal{L}_\infty)$ and largest for $(ILP - BIF)$. Nevertheless, in any case the amount of undecided pixels drops significantly after a few iterations. The total number of iterations that was used in each experiment can be found in table 5.1 along with the average time consumption of a single iteration.

[iterations]/[seconds]	Nr. projections	$(ILP - BIF)$	$(ILP - \mathcal{L}_1)$	$(ILP - \mathcal{L}_\infty)$
turbine	3	73 / 74	70 / 701	12 / 1736
	4	83 / 71	82 / 794	5 / 1343
	5	93 / 82	8 / 871	6 / 1587
	6	12 / 70	5 / 997	5 / 1490
single object	3	75 / 313	77 / 859	6 / 1769
	4	5 / 155	3 / 852	5 / 1340
	5	5 / 202	1 / 1632	1 / 1244
ellipses	3	77 / 84	84 / 526	10 / 607
	4	76 / 54	95 / 648	6 / 688
	5	83 / 65	116 / 916	5 / 1187
	6	78 / 72	7 / 647	6 / 1353
	7	14 / 115	4 / 963	4 / 1597
	8	4 / 73	3 / 1117	6 / 2053
cylinders	3	84 / 219	78 / 550	6 / 1899
	4	84 / 171	91 / 719	7 / 2012
	5	115 / 257	108 / 987	6 / 1232
	6	114 / 266	139 / 1317	6 / 2557
	8	50 / 254	5 / 1593	6 / 2751
	10	5 / 347	3 / 978	9 / 2695

Table 5.1: This table lists the total number of iterations for each experiment and the average time consumption of a single iteration in seconds.



(a) Iteration 1.

(b) Iteration 2.

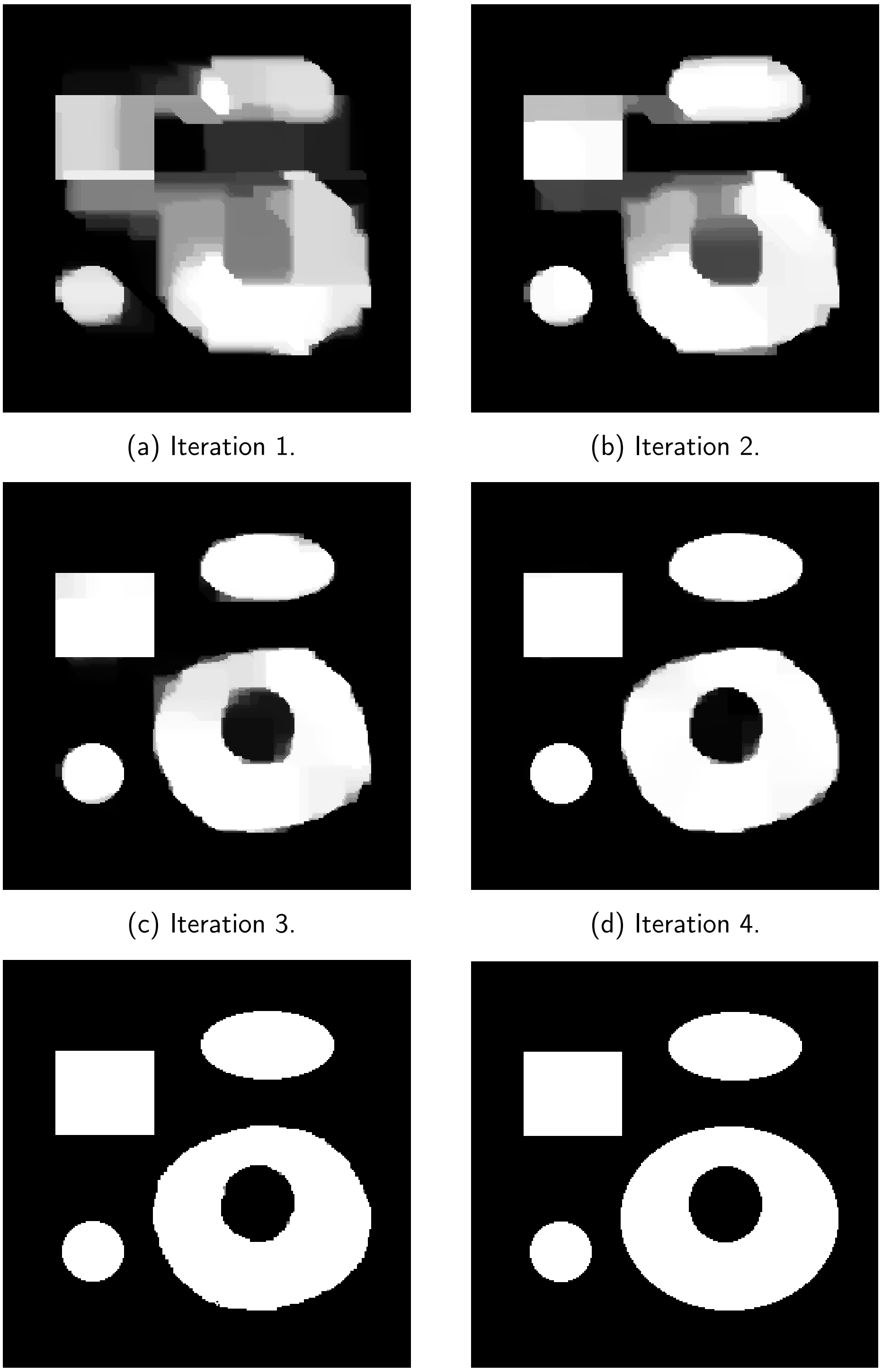
(c) Iteration 3.

(d) Iteration 4.

(e) Iteration 5.

(f) Original.

Figure 5.2: Reconstruction with $(ILP - BIF)$ from three projections, $0°$, $45°$, and $90°$ which were taken from the image shown in figure (f). After 5 iterations already, algorithm 8 returns a solution, figure (e), equal to the groundtruth image.



(a) Iteration 1.

(b) Iteration 2.

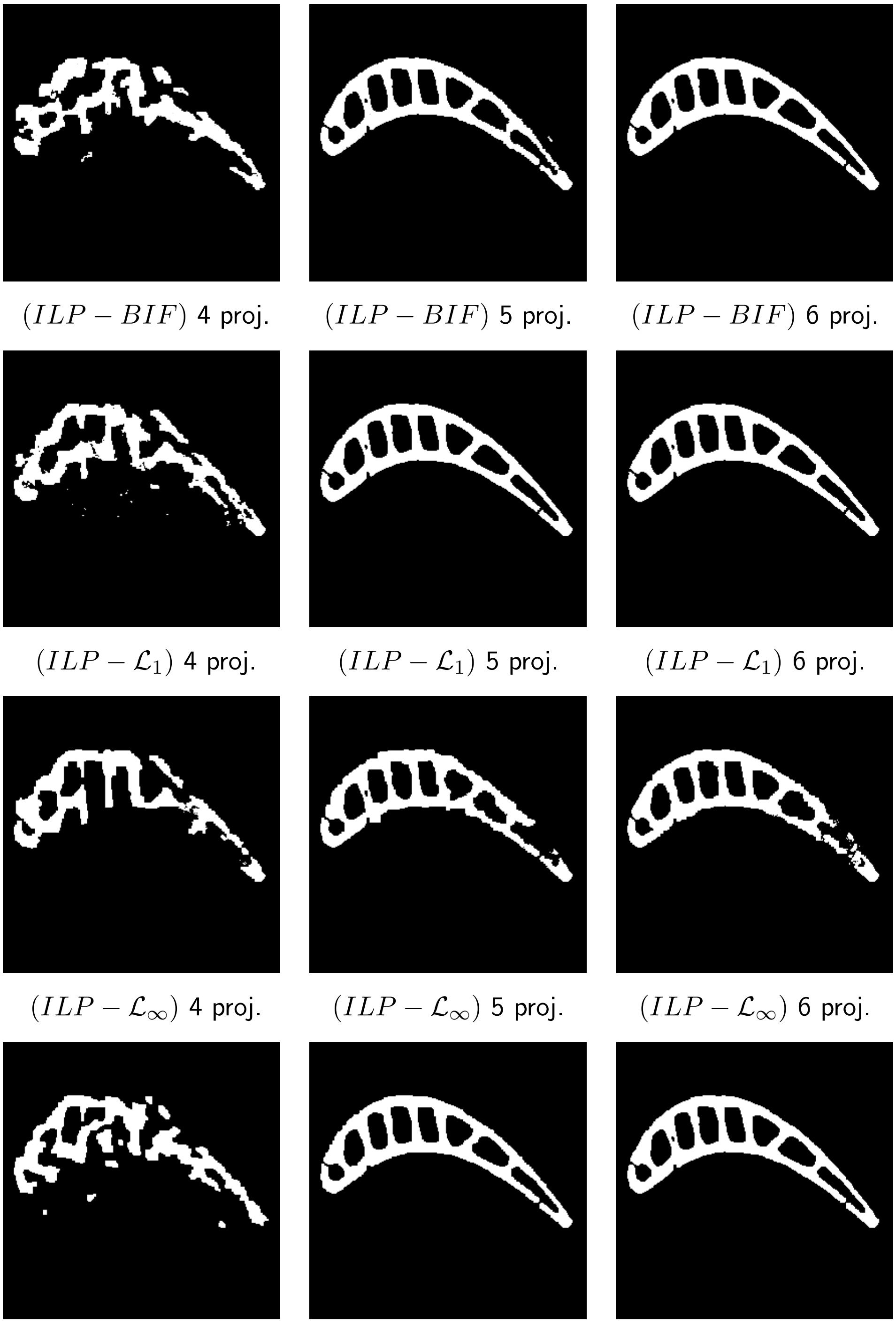
(c) Iteration 3.

(d) Iteration 4.

(e) Iteration 36.

(f) Original.

Figure 5.3: Similar to figure 5.2, the images (a)–(e) show results at different iteration steps of algorithm 8. The reconstruction problem has been set up from three projections, $0°$, $45°$, and $90°$, taken from the image shown in (f).



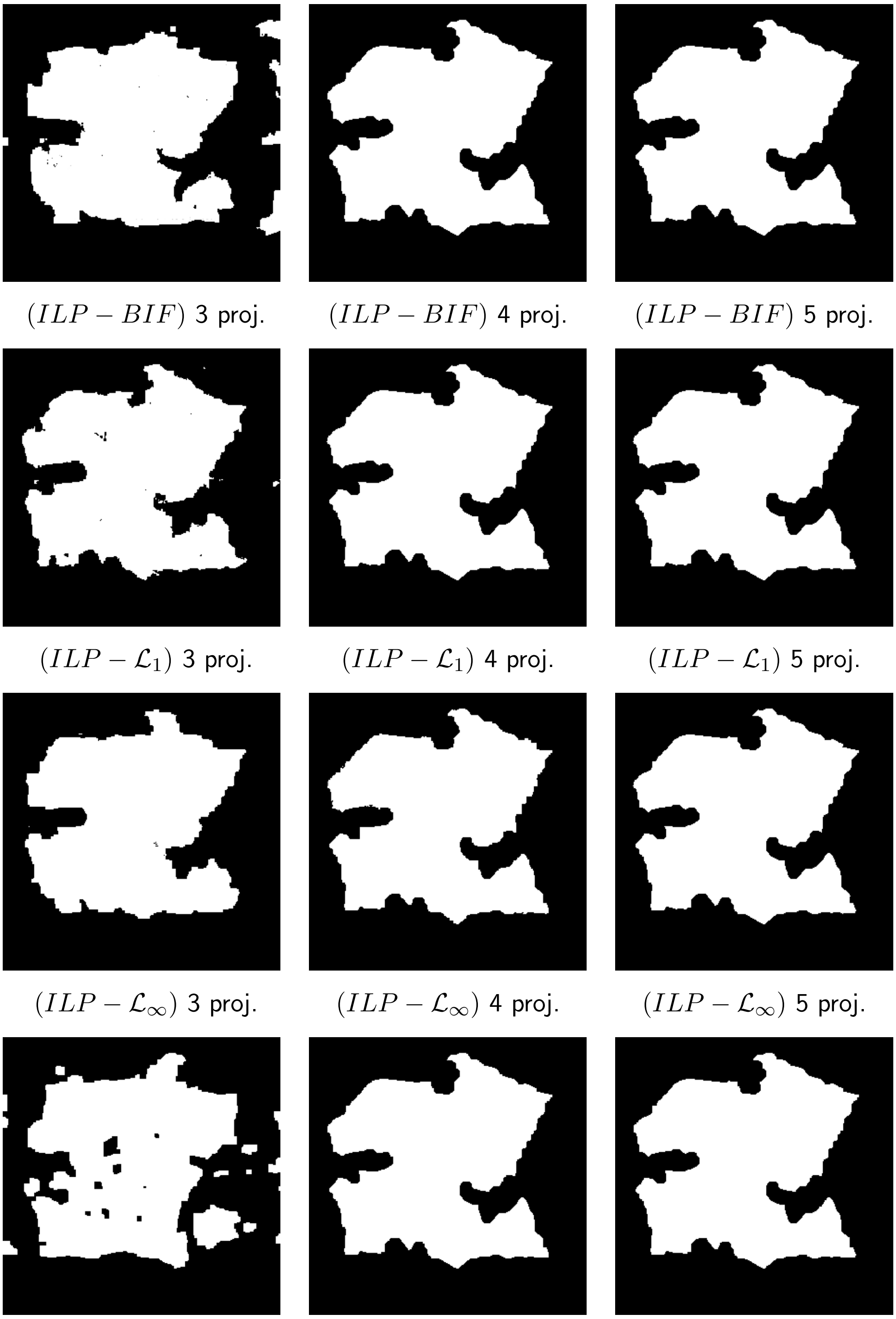
$(ILP - BIF)$ 4 proj. $(ILP - BIF)$ 5 proj. $(ILP - BIF)$ 6 proj.

$(ILP - \mathcal{L}_1)$ 4 proj. $(ILP - \mathcal{L}_1)$ 5 proj. $(ILP - \mathcal{L}_1)$ 6 proj.

$(ILP - \mathcal{L}_\infty)$ 4 proj. $(ILP - \mathcal{L}_\infty)$ 5 proj. $(ILP - \mathcal{L}_\infty)$ 6 proj.

(SA) 4 proj. (SA) 5 proj. (SA) 6 proj.

Figure 5.4: The figure shows reconstruction results obtained with the ILP approaches using the *turbine* image with various noiseless projections. We used the same problem settings as previously in connection with the linear programming experiments, figure 4.8, i.e. the same reconstruction problems and the same choices for the smoothness parameter: $\alpha := 0.5$ in case of $(ILP - BIF)$ and $(ILP - \mathcal{L}_1)$ and $\alpha := 0.001$ for $(ILP - \mathcal{L}_\infty)$.



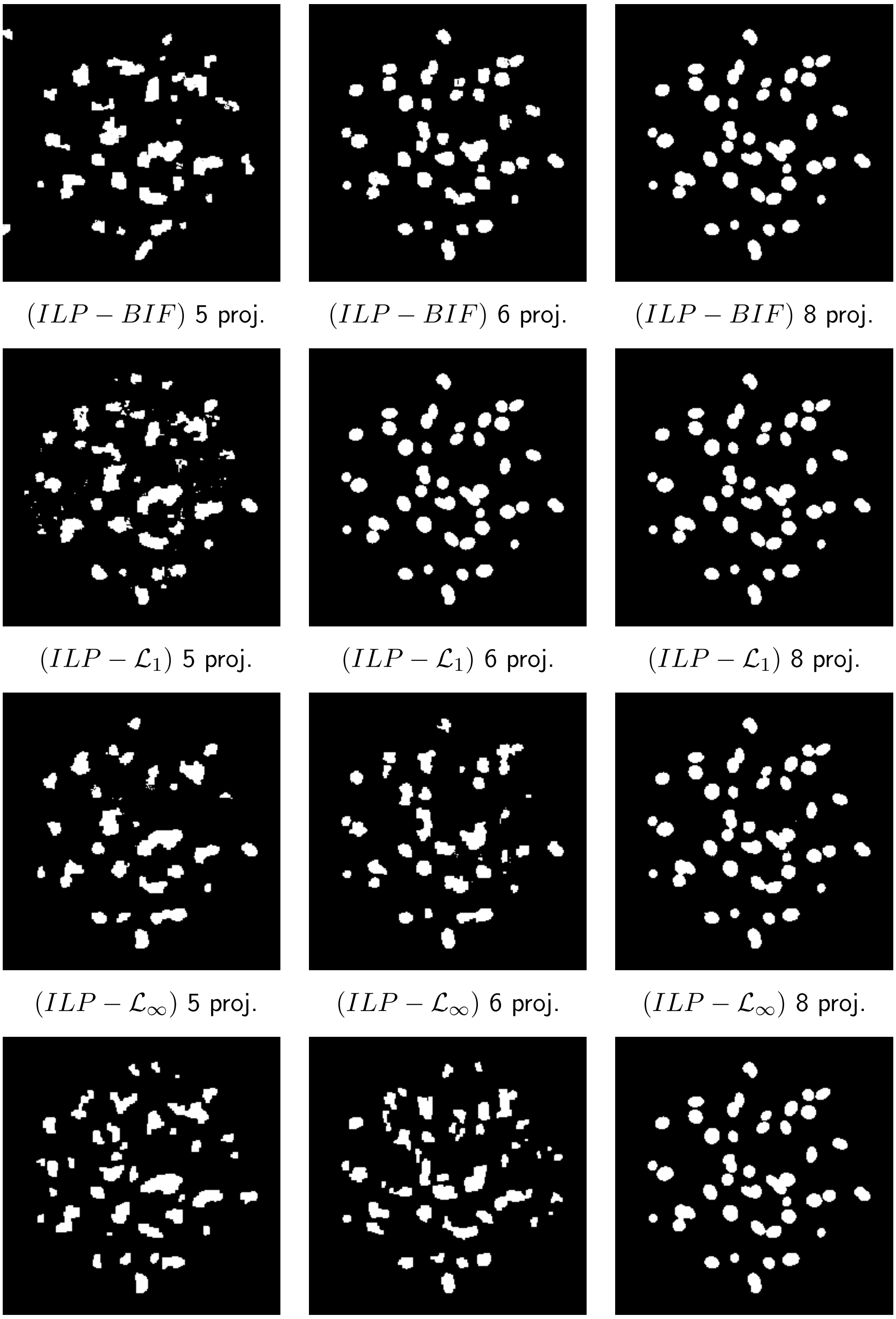
$(ILP - BIF)$ 3 proj. $(ILP - BIF)$ 4 proj. $(ILP - BIF)$ 5 proj.

$(ILP - \mathcal{L}_1)$ 3 proj. $(ILP - \mathcal{L}_1)$ 4 proj. $(ILP - \mathcal{L}_1)$ 5 proj.

$(ILP - \mathcal{L}_\infty)$ 3 proj. $(ILP - \mathcal{L}_\infty)$ 4 proj. $(ILP - \mathcal{L}_\infty)$ 5 proj.

(SA) 3 proj. (SA) 4 proj. (SA) 5 proj.

Figure 5.5: In comparison to figure 4.9, this figure shows the reconstruction results obtained by the ILP approaches for the *single object* image from noise-free projections. Analogously, the regularization parameter α has been set to 0.5 for $(ILP - BIF)$ and $(ILP - \mathcal{L}_1)$ and in case of $(ILP - \mathcal{L}_\infty)$ to 0.001 throughout these experiments.



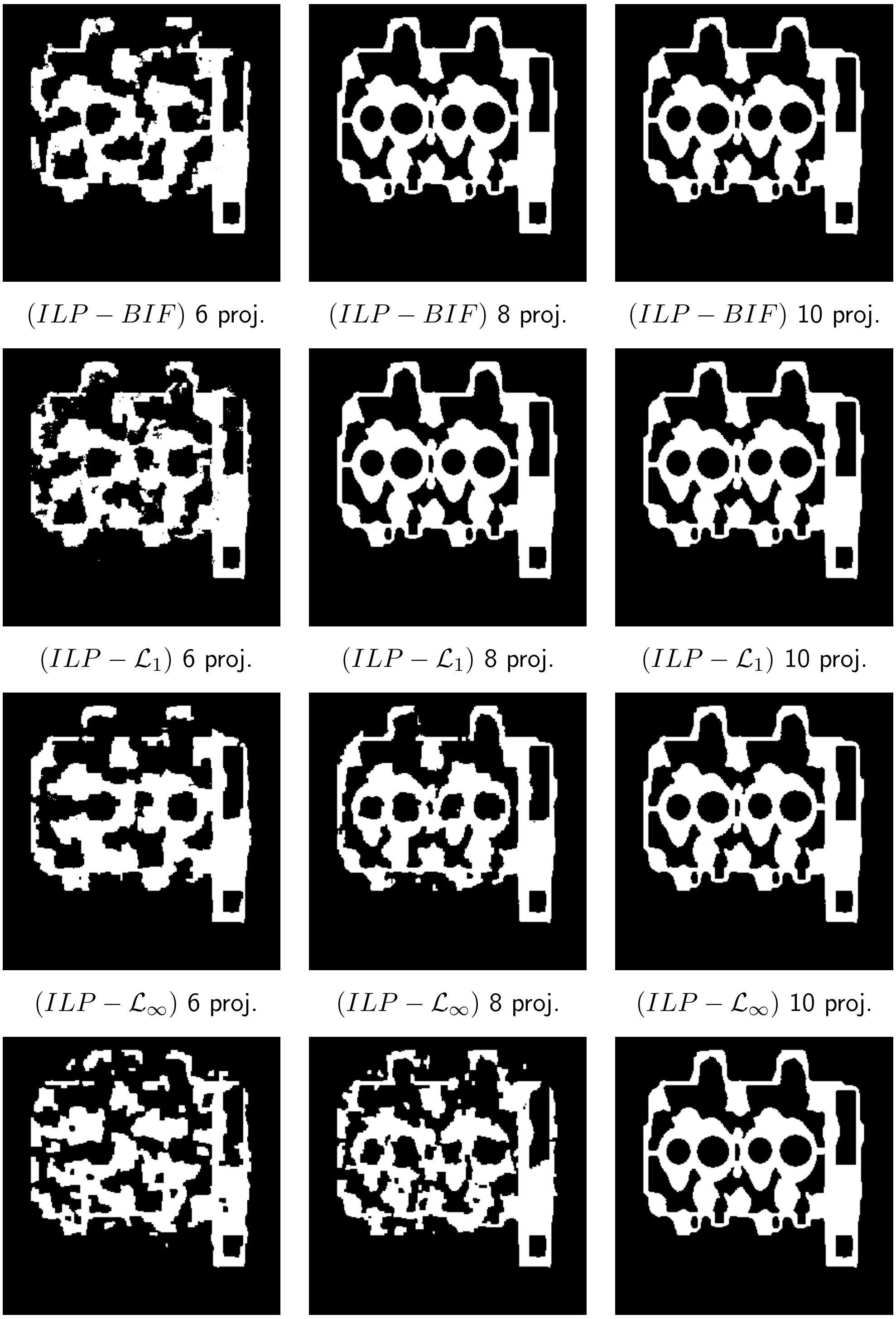
$(ILP - BIF)$ 5 proj. $(ILP - BIF)$ 6 proj. $(ILP - BIF)$ 8 proj.

$(ILP - \mathcal{L}_1)$ 5 proj. $(ILP - \mathcal{L}_1)$ 6 proj. $(ILP - \mathcal{L}_1)$ 8 proj.

$(ILP - \mathcal{L}_\infty)$ 5 proj. $(ILP - \mathcal{L}_\infty)$ 6 proj. $(ILP - \mathcal{L}_\infty)$ 8 proj.

(SA) 5 proj. (SA) 6 proj. (SA) 8 proj.

Figure 5.6: In comparison to figure 4.10, this figure shows the reconstruction results obtained by the ILP approaches for the *many ellipses* image from noise-free projections. Analogously, the regularization parameter α has been set to 0.5 for $(ILP - BIF)$ and $(ILP - \mathcal{L}_1)$ and in case of $(ILP - \mathcal{L}_\infty)$ to 0.001 for all experiments.



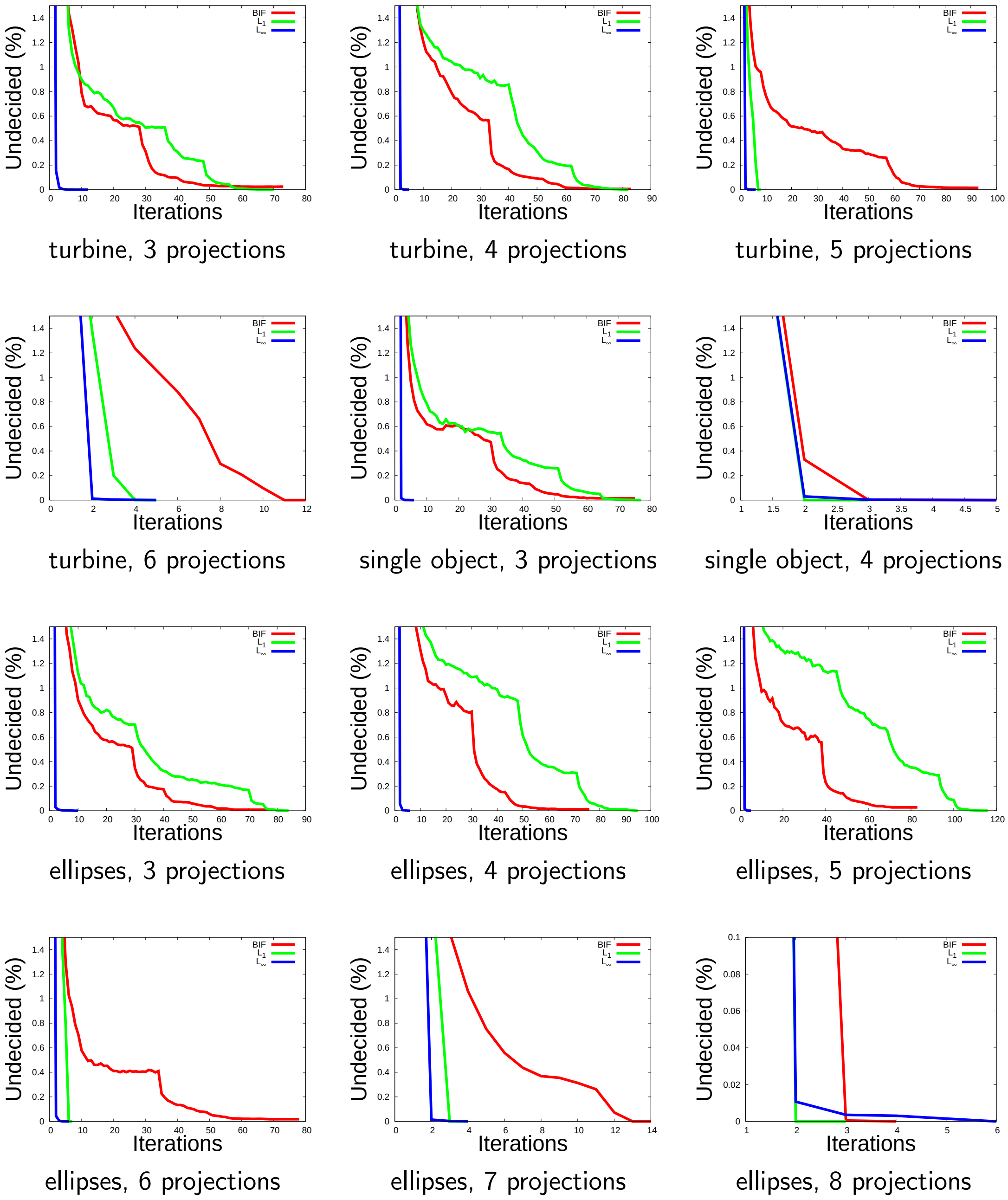
$(ILP - BIF)$ 6 proj. $(ILP - BIF)$ 8 proj. $(ILP - BIF)$ 10 proj.

$(ILP - \mathcal{L}_1)$ 6 proj. $(ILP - \mathcal{L}_1)$ 8 proj. $(ILP - \mathcal{L}_1)$ 10 proj.

$(ILP - \mathcal{L}_\infty)$ 6 proj. $(ILP - \mathcal{L}_\infty)$ 8 proj. $(ILP - \mathcal{L}_\infty)$ 10 proj.

(SA) 6 proj. (SA) 8 proj. (SA) 10 proj.

Figure 5.7: In comparison to figure 4.11, this figure shows the reconstruction results obtained by the ILP approaches for the *cylinders* image from noise-free projections. Analogously, the regularization parameter α has been set to 0.5 for $(ILP - BIF)$ and $(ILP - \mathcal{L}_1)$ and in case of $(ILP - \mathcal{L}_\infty)$ to 0.001.



turbine, 3 projections

turbine, 4 projections

turbine, 5 projections

turbine, 6 projections

single object, 3 projections

single object, 4 projections

ellipses, 3 projections

ellipses, 4 projections

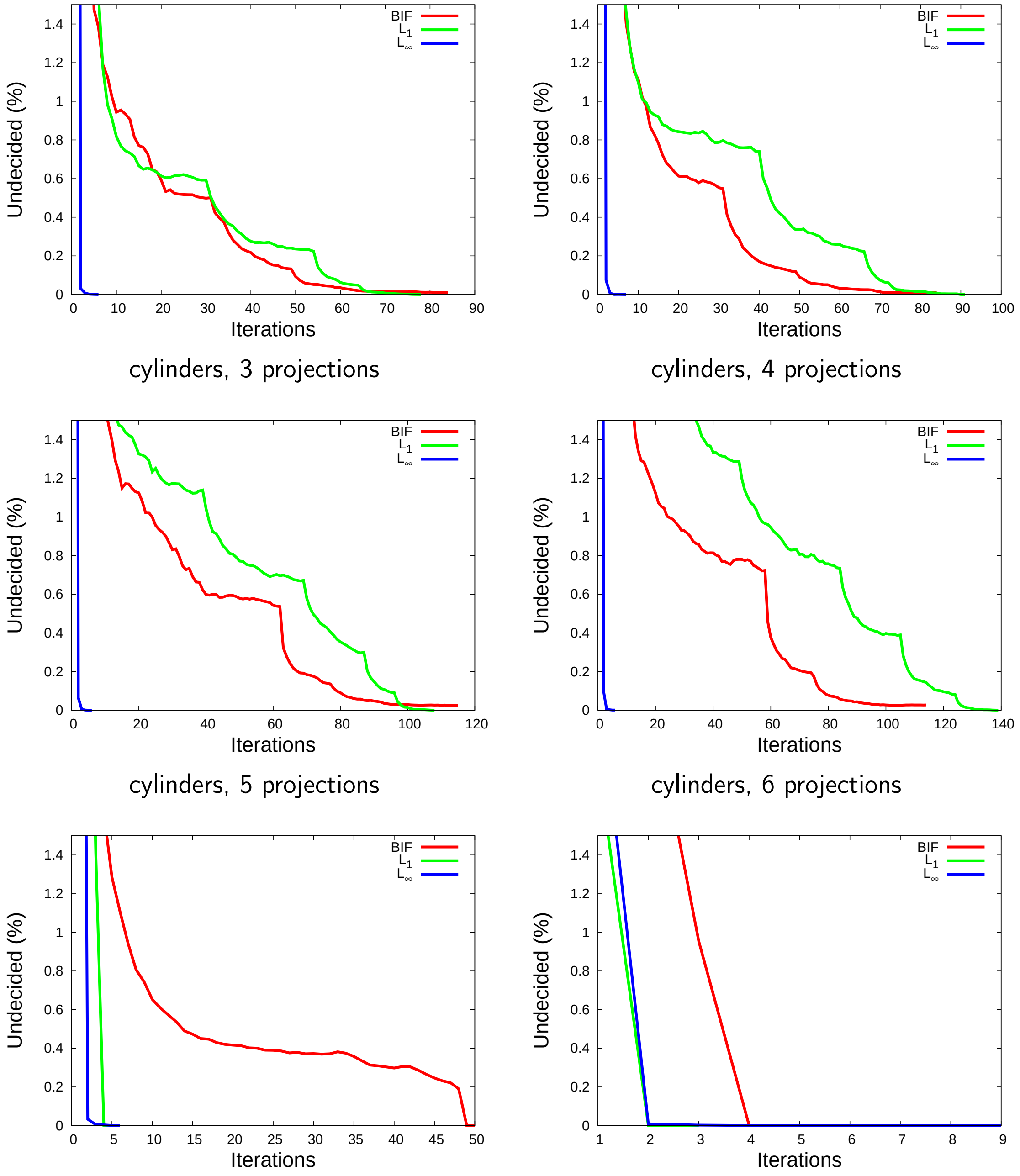
ellipses, 5 projections

ellipses, 6 projections

ellipses, 7 projections

ellipses, 8 projections

Figure 5.8: These plots present the amount of undecided pixels, i.e. pixels which have no binary decision yet, for the *turbine*, *single object*, and *many ellipses* experiment.



cylinders, 3 projections

cylinders, 4 projections

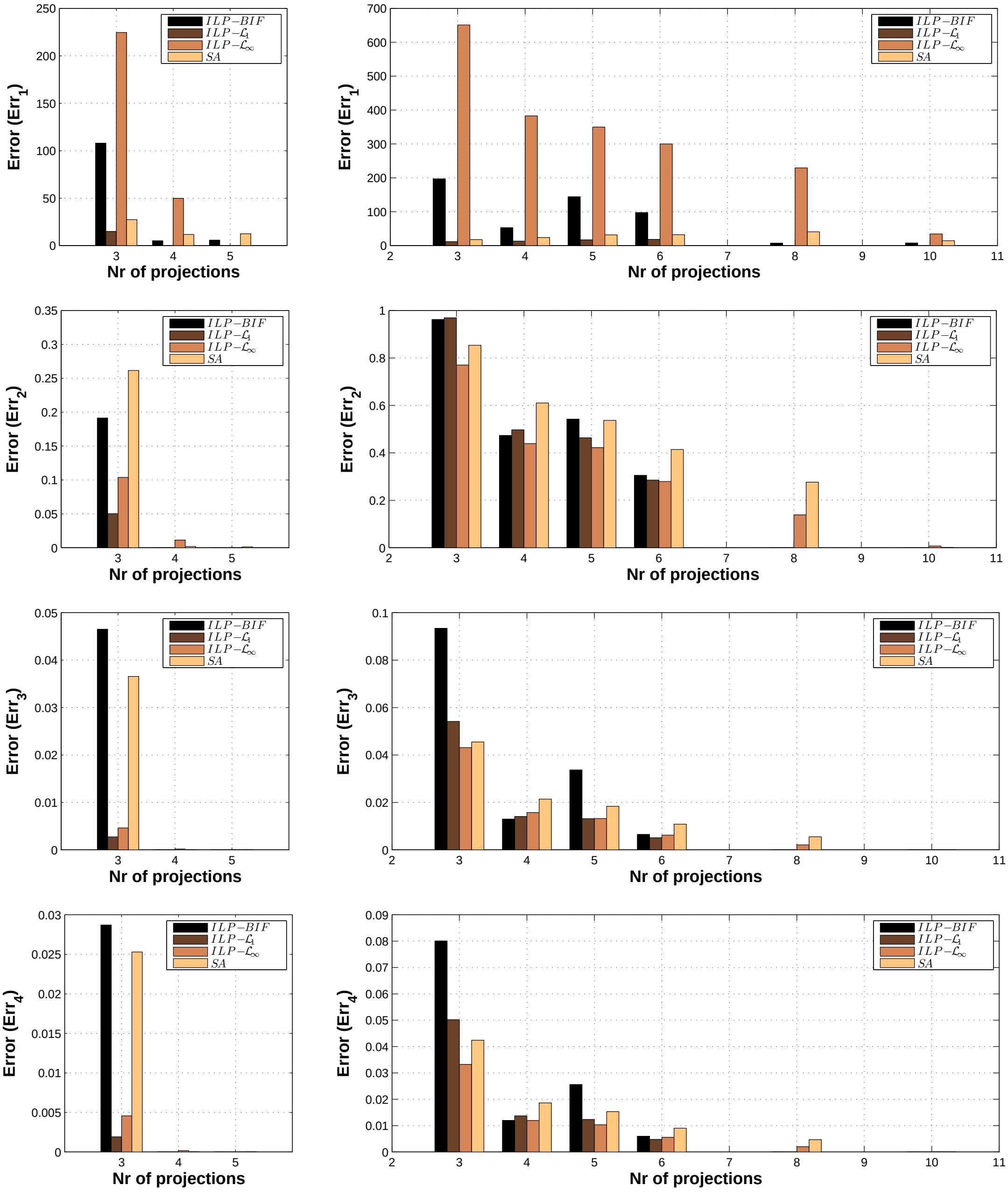
cylinders, 5 projections

cylinders, 6 projections

cylinders, 8 projections

cylinders, 10 projections

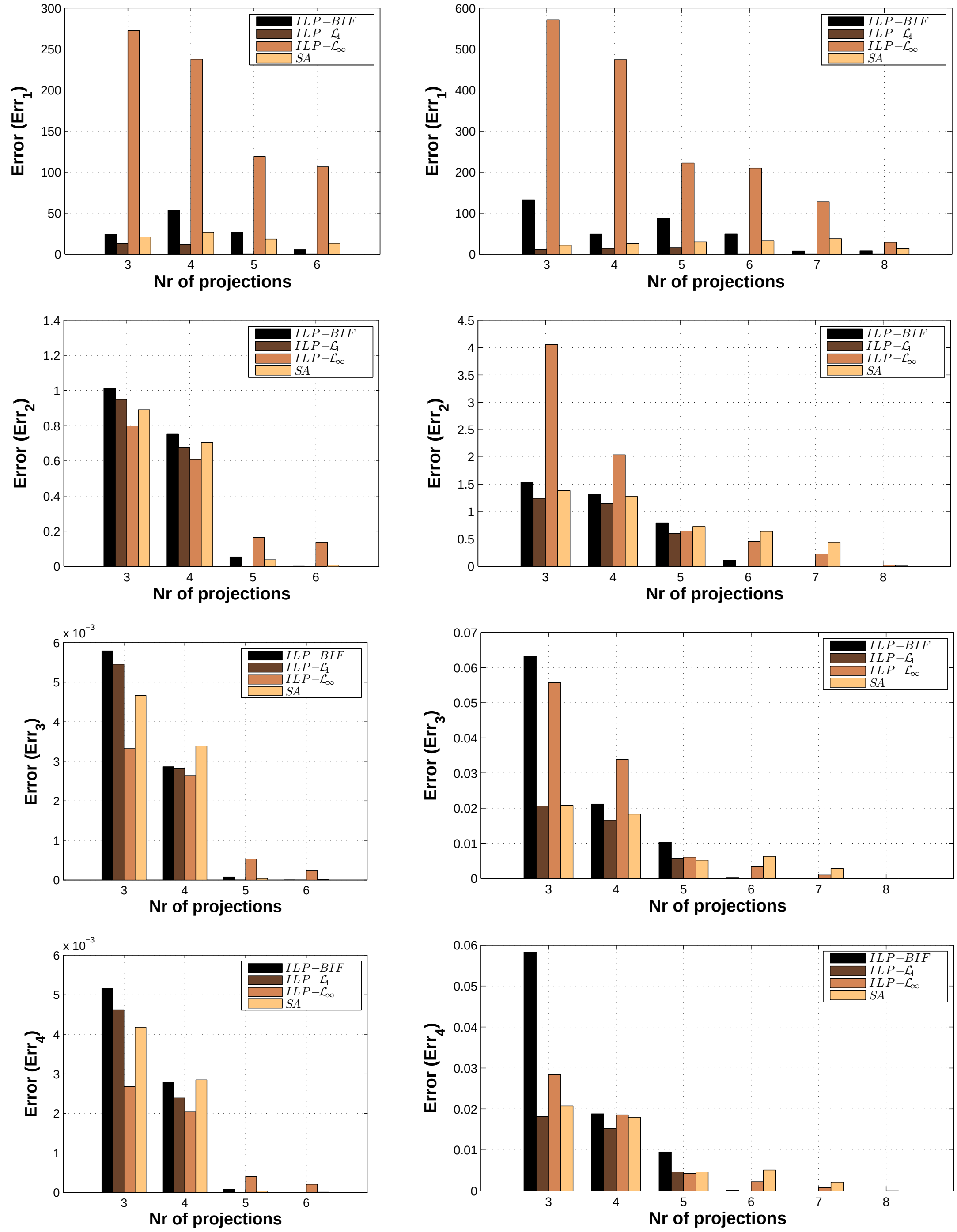
Figure 5.9: These plots present the amount of undecided pixels, i.e. pixels which have no binary decision yet, for the *cylinders* experiment.



single object

cylinders

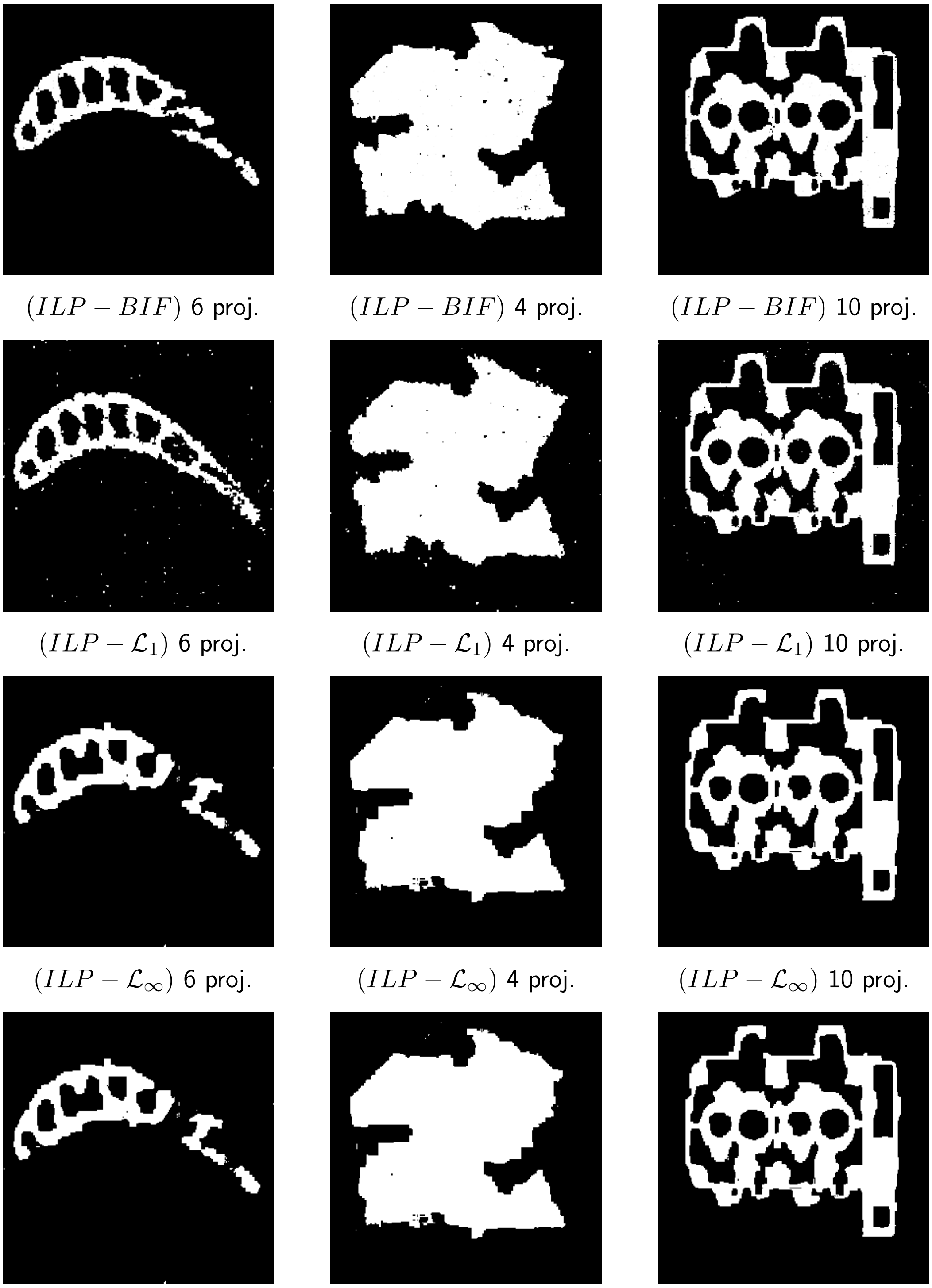
Figure 5.10: Error measurements (without noise): (left) *single object* and (right) *cylinders*.



turbine

many ellipses

Figure 5.11: Error measurements (without noise): (left) *turbine* and (right) *many ellipses*.



$(ILP - BIF)$ 6 proj. $(ILP - BIF)$ 4 proj. $(ILP - BIF)$ 10 proj.

$(ILP - \mathcal{L}_1)$ 6 proj. $(ILP - \mathcal{L}_1)$ 4 proj. $(ILP - \mathcal{L}_1)$ 10 proj.

$(ILP - \mathcal{L}_\infty)$ 6 proj. $(ILP - \mathcal{L}_\infty)$ 4 proj. $(ILP - \mathcal{L}_\infty)$ 10 proj.

(SA) 6 proj. (SA) 4 proj. (SA) 10 proj.

Figure 5.12: Reconstructions results from noisy projections, $\mathcal{N}(\mu := 0, \sigma)$ added to projection vector $\mathbf{b}$: For $(ILP-BIF \mid ILP-\mathcal{L}_1 \mid ILP-\mathcal{L}_\infty)$, α was set to (1.0 | 1.0 | 0.005) in case of *turbine* ($\sigma := 3.0$), *single object* ($\sigma := 3.0$),*many ellipses* ($\sigma := 3.0$), and to (0.75 | 0.75 | 0.0025) for *cylinders* ($\sigma := 1.0$).

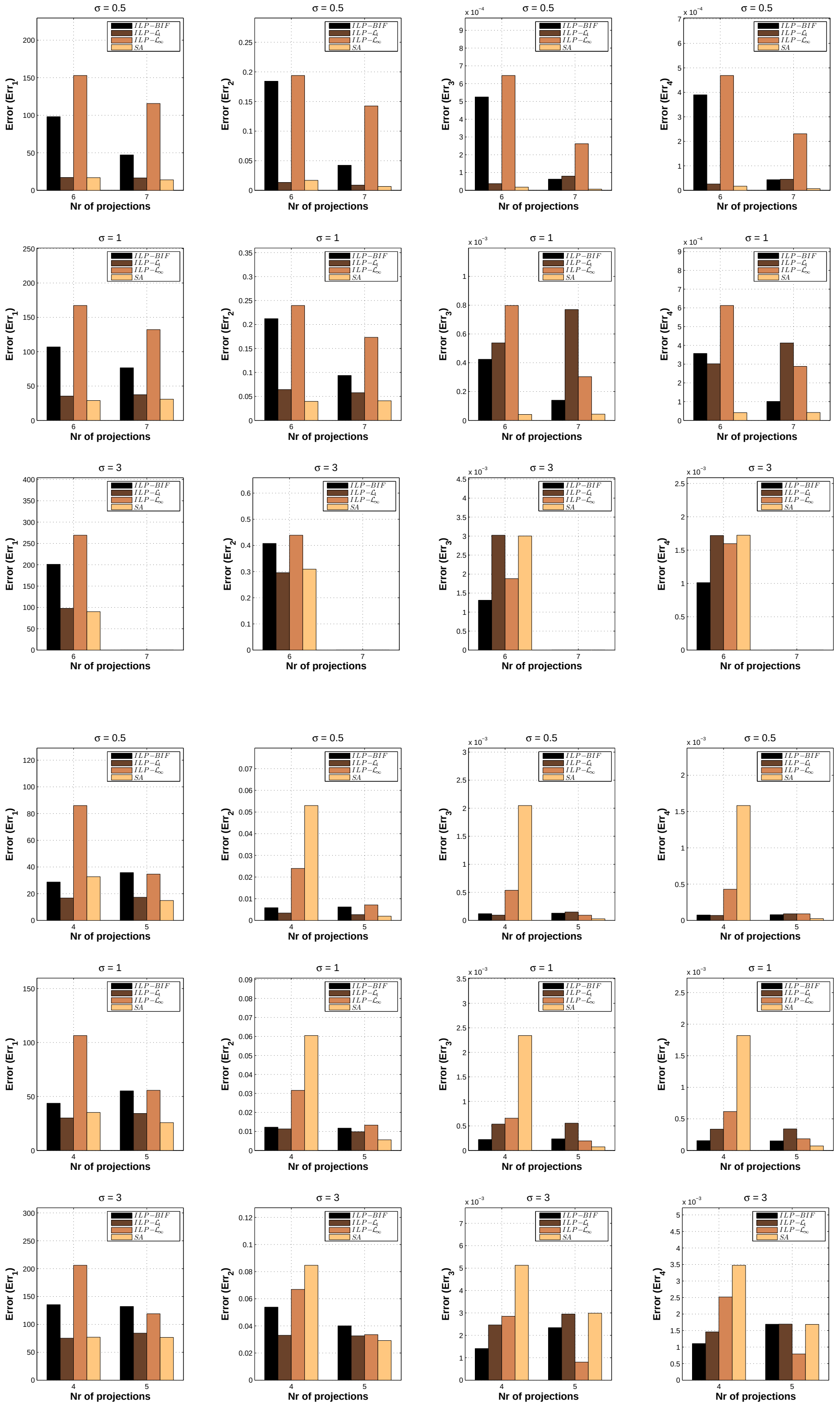


Figure 5.13: Error measurements (noisy projections): (top) *turbine* and (bottom) *single object*.

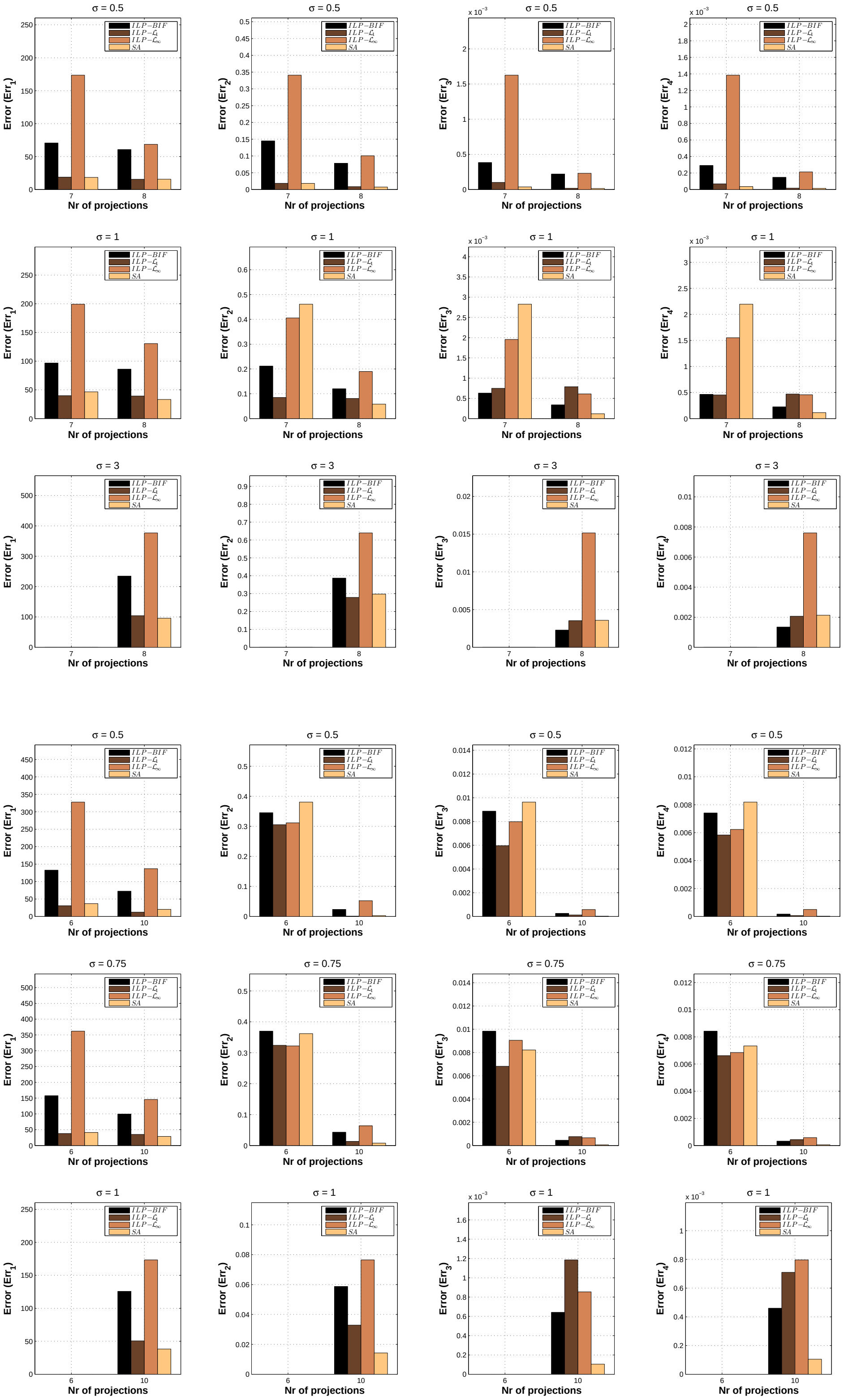


Figure 5.14: Error measurements (noisy projections): (top) *many ellipses* and (bottom) *cylinders*.

5.5 Binary Tomography with Deblurring

So far we considered only linear programming in order to solve the convex optimization problem within the d.c. based optimization framework but stressed that it is by no means restricted it. In this section, we demonstrate its flexibility in two different ways: i) we include quadratic programming for the optimization in the x-step and ii) we supplement our d.c. framework with an additional expectation maximization (EM) step which estimates a hidden parameter during the reconstruction process.

5.5.1 Motivation

It is a general characteristic of imaging systems that the acquired images are some distorted versions of the ideal images of real objects. The distortion is due to physical limitations, e.g. finite resolution in space and time, non-uniform sensitivity in the field of view, etc. In many cases the distorted image can be modeled as the convolution of the ideal image with some function describing the distortion [59].

The situation is the same in tomography since the pixel values in the projection images are usually only some approximation of the line integrals to be measured by a perfect imaging system in an ideal physical situation. In different application areas of tomography there are several correction methods to improve the quality of the reconstructed images. The correction strategies can be roughly divided into two classes. The first class contains the methods aiming to correct the projection data before reconstruction (let us call them preprocessing) and then the reconstruction is performed from the corrected projection data. The second class is the family of special methods which include the correction into the reconstruction process. We believe that both strategies can be useful. If the correction can be done as a preprocessing step before reconstruction then one of the methods from the first class is preferable. However, there are situations when the correction is impossible or too complicated before reconstruction, e.g. scatter correction in CT or in SPECT, then the correction during the reconstruction can still give a good solution.

The distortion of tomographic images has become an important issue also in case of binary tomography since it is getting to be applied in several areas. There are several publications discussing different corrections in DT, e.g. in X-ray and neutron tomography [4, 87], and electron microscopy [33]. Concerning our work, we deal with the general distortion model when the distortion can be described by the convolution with a Gaussian kernel $G_\sigma(\cdot)$. If the parameter σ is known in advance then the correction (deconvolution) can be done as a preprocessing step before the reconstruction. However, if the parameter is not known then we are going to show that there is still a way to perform binary tomography by including this parameter as an unknown value to be determined. To motivate our approach we present some reconstructions, see figure 5.15, without deblurring.

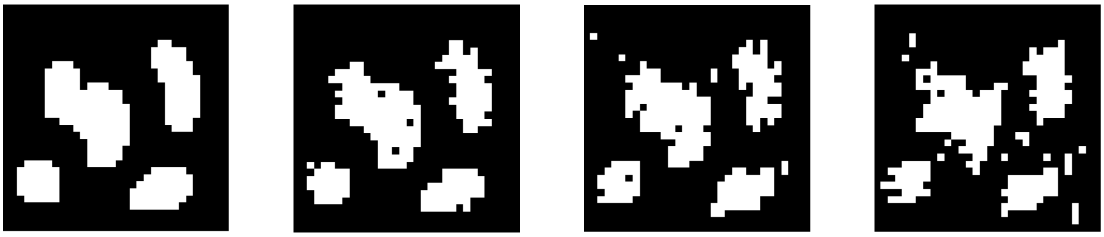

(a) Original (b) $\sigma = 0.5$ (c) $\sigma = 1.0$ (d) $\sigma = 2.0$

Figure 5.15: Reconstruction without deblurring fails: Panel (a) shows an object which was blurred with a Gaussian convolution kernel $\mathbf{G}_\sigma$ at three different scales $\sigma \in \{0.5, 1.0, 2.0\}$, and then projected along 5 directions $0°$, $22.5°$, $45°$, $67.5°, 90°$. Panels (b)-(d) show the reconstruction results without deblurring. The performance considerably deteriorates for increasing σ. Note that the original object (a) can be reconstructed without any error from three projections.

5.5.2 Problem Statement

Let $\mathbf{G}_\sigma$ denote the matrix that represents the linear mapping of some data by convolving it with an isotropic Gaussian kernel and scale-parameter σ. We generalize the reconstruction problem along two directions:

(i) Reconstruction from projections of blurred objects:
The corresponding generalization of the reconstruction problem reads

$$\mathbf{A}\,\mathbf{G}_\sigma\,\mathbf{x} = \mathbf{b}\,, \qquad \mathbf{A} \in \mathbb{R}^{m \times n}\,,\ \mathbf{x} \in \{0,1\}^n\,,\ \mathbf{b} \in \mathbb{R}^m. \tag{5.29}$$

(ii) Reconstruction from blurred projection data:
The corresponding generalization of the reconstruction problem reads

$$\mathbf{G}_\sigma \mathbf{A}\,\mathbf{x} = \mathbf{b}\,, \qquad \mathbf{A} \in \mathbb{R}^{m \times n}\,,\ \mathbf{x} \in \{0,1\}^n\,,\ \mathbf{b} \in \mathbb{R}^m. \tag{5.30}$$

For notational simplicity, we use in both cases the same symbol $\mathbf{G}_\sigma$, although $\mathbf{G}_\sigma$ denotes a block-circulant matrix in (5.29) corresponding to the convolution of multi-dimensional data $\mathbf{x}$, whereas $\mathbf{G}_\sigma$ represents the one-dimensional convolution of the projection data in (5.30).

5.5.3 Quadratic Optimization and DC Framework

In [122, 118] we investigated the reconstruction problem as defined in section 5.4 and considered, at this, the family of quadratic minimization problems

$$\begin{aligned}\min_{\mathbf{x}\in[0,1]^n} E(\mathbf{x};\mu) &:= \frac{1}{2}\left[\|\mathbf{A}\,\mathbf{x}-\mathbf{b}\|^2+\alpha\,\|\mathbf{L}\,\mathbf{x}\|^2+\mu\,\langle\mathbf{x},\mathbf{1}-\mathbf{x}\rangle\right] \\ &= \frac{1}{2}\,\mathbf{x}^\top\mathbf{A}^\top\mathbf{A}\,\mathbf{x}-\mathbf{b}^\top\mathbf{A}\,\mathbf{x}+\frac{1}{2}\,\mathbf{b}^\top\mathbf{b}+\frac{\alpha}{2}\,\mathbf{x}^\top\mathbf{L}^\top\mathbf{L}\,\mathbf{x}+\frac{\mu}{2}\,\mathbf{x}^\top(\mathbf{1}-\mathbf{x}) \\ &= \frac{1}{2}\,\mathbf{x}^\top(\mathbf{A}^\top\mathbf{A}+\alpha\,\mathbf{L}^\top\mathbf{L}-\mu\,\mathbf{I})\,\mathbf{x}+\frac{1}{2}\left(\mu\,\mathbf{1}^\top-2\mathbf{b}^\top\mathbf{A}\right)\mathbf{x}+\frac{1}{2}\,\mathbf{b}^\top\mathbf{b}\end{aligned} \tag{5.31}$$

where each row of matrix $\mathbf{L}$ measures the difference $x_i - x_j$ of spatially neighboring pixels i and j, i.e. $i \in \mathcal{N}(j)$ and $j \in \mathcal{N}(i)$ due to symmetry. Consequently, the second term of $E(\mathbf{x};\mu)$ corresponds to the smoothness prior

$$\|\mathbf{L}\,\mathbf{x}\|^2 = \sum_{\langle i,j\rangle}(x_i-x_j)^2 \tag{5.32}$$

with regularization parameter α.

In view of equation (5.31) we may also write

$$E(\mathbf{x};\mu) := \frac{1}{2}\,\mathbf{x}^\top\mathbf{Q}_\mu\,\mathbf{x}+\mathbf{q}_\mu^\top\mathbf{x}+c \tag{5.33}$$

where $\mathbf{Q}_\mu$, $\mathbf{q}_\mu$, and c are defined according to

$$\mathbf{Q}_\mu := \mathbf{A}^\top\mathbf{A}+\alpha\,\mathbf{L}^\top\mathbf{L}-\mu\,\mathbf{I} \tag{5.34}$$

$$\mathbf{q}_\mu^\top := \frac{\mu}{2}\,\mathbf{1}^\top-\mathbf{b}^\top\mathbf{A} \tag{5.35}$$

$$c := \frac{1}{2}\,\mathbf{b}^\top\mathbf{b}. \tag{5.36}$$

Similarly to section 5.4, we equivalently express the minimization of functional (5.31) over the convex set of feasible solutions $[0,1]^n$ with the indicator function $\delta_{[0,1]^n}(\mathbf{x})$, equation (5.23),

$$\min_{\mathbf{x}\in\mathbb{R}^n} E(\mathbf{x};\mu) := \frac{1}{2}\left[\|\mathbf{A}\,\mathbf{x}-\mathbf{b}\|^2+\alpha\,\|\mathbf{L}\,\mathbf{x}\|^2+\mu\,\langle\mathbf{x},\mathbf{1}-\mathbf{x}\rangle\right]+\delta_{[0,1]^n}(\mathbf{x}) \tag{5.37}$$

and find a proper d.c. decomposition by

$$g(\mathbf{x}) := \frac{1}{2}\left[\|\mathbf{A}\,\mathbf{x}-\mathbf{b}\|^2+\alpha\,\|\mathbf{L}\,\mathbf{x}\|^2\right]+\delta_{[0,1]^n}(\mathbf{x}) \tag{5.38}$$

$$h(\mathbf{x};\mu) := \frac{\mu}{2}\,\langle\mathbf{x},\mathbf{x}-\mathbf{1}\rangle. \tag{5.39}$$

We stress again that such decompositions are not unique in general and in case of functional $E(\mathbf{x};\mu)$ a different decomposition has been used in [122, 134] leading to a special instance of the Goldstein-Levitin-Polyak gradient projection method in the x-step.

Proceeding with the decomposition in (5.38) and (5.39), the complete reconstruction approach reads as shown in algorithm 9. The regularization parameter μ must initially be chosen smaller than the smallest eigenvalue $\lambda_{\min}(\mathbf{Q}_0)$ of matrix $\mathbf{Q}_0$ since the quadratic problem, equation (5.31) and (5.33), remains strictly convex then and, thus, minimizing still yields the global optimum.

In case of $\mathbf{x} \neq \mathbf{0}$ the smoothness term, equation (5.32), becomes only zero for constant $\mathbf{x}$, i.e. $\mathbf{x} = c\,\mathbf{1}$, $c \neq 0$, which is, however, not true for $\mathbf{A}^\top\mathbf{A}$. Consequently, the positive definiteness of $\mathbf{Q}_0$ follows and thus $0 < \lambda_{\min}(\mathbf{Q}_0)$. Increasing parameter μ during the iterations shifts the eigenvalues of $\mathbf{Q}_0$ in negative direction, as can be seen from equation (5.34), and matrix $\mathbf{Q}_\mu$ becomes indefinite for $\lambda_{\min}(\mathbf{Q}_\mu) < 0 < \lambda_{\max}(\mathbf{Q}_\mu)$ which is equivalent to $\lambda_{\min}(\mathbf{Q}_0) < \mu < \lambda_{\max}(\mathbf{Q}_0)$.

For practical implementations the smallest eigenvalue of $\mathbf{Q}_0$ is found numerically using either the power or the inverse power iteration, see appendix E. While the inverse iteration is able to locate the smallest eigenvalue and its corresponding eigenvector directly the power method computes only the absolute largest eigenpair of a matrix. However, by means of a simple trick it can be used to find the smallest eigenpair as well and thus avoids the linear equation systems which must be solved in case of the inverse power iteration. Therefore, we start with computation of the absolute largest eigenvalue of $\mathbf{Q}_0$ and have that $0 < \lambda_{\min}(\mathbf{Q}_0) < \lambda_{\max}(\mathbf{Q}_0)$ due to the positive definiteness of $\mathbf{Q}_0$. If we apply the power iteration to matrix $(\lambda_{\max}(\mathbf{Q}_0)\mathbf{I} - \mathbf{Q}_0)$ afterwards we obtain $(\lambda_{\max}(\mathbf{Q}_0) - \lambda_{\min}(\mathbf{Q}_0))$ as absolute largest eigenvalue and, thus, $\lambda_{\min}(\mathbf{Q}_0)$.

We consider problem (5.33) here only insofar as it necessary to cover the subject of this section. For more details, we would like to encourage the reader to have a view at [122, 118, 124]. Particularly [124] includes many more interesting aspects of the d.c. optimization framework in terms of quadratic optimization which are not in the scope of this work, like more sophisticated update schemes for μ and the selection of individual penalty parameters μ_i for each variable x_i.

Coming back again to the scenario of blurred image or projection data as described in section 5.5.2, the reconstruction problem (5.31) generalizes to

$$\min_{\mathbf{x}\in[0,1]^n} E_\sigma(\mathbf{x};\mu) := \frac{1}{2}\left[D(\mathbf{x};\sigma) + \alpha\, S(\mathbf{x}) + \mu\, \langle \mathbf{x}, \mathbf{1} - \mathbf{x}\rangle\right] \tag{5.40}$$

Algorithm 9 DC-Framework with Quadratic Programming

```
Require: Choose $\mathbf{x}^0$ arbitrary                         (our choice: $\mathbf{x}^0 := (\frac{1}{2}, ..., \frac{1}{2})^\top$)
Require: $0 < \Delta\mu$                                          (our choice: $\Delta\mu \in (0, 0.5]$)
Require: $0 < \epsilon$                                           (our choice: $10^{-4} \leq \epsilon \leq 10^{-2}$)
  $\mu < \lambda_{\min}(\mathbf{Q}_0)$
  repeat
    $k := 0$
    repeat
      $\mathbf{y}^k := \nabla h(\mathbf{x}^k; \mu) = \mu(\mathbf{x}^k - \frac{1}{2}\mathbf{1})$          {See equation (5.27).}
      $\mathbf{x}^{k+1} := \operatorname*{argmin}_{\mathbf{x} \in [0,1]^n} \{g(\mathbf{x}) - \langle \mathbf{y}^k, \mathbf{x} \rangle\}$
      $k := k + 1$
    until $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2 < \epsilon$                                   (DC-loop)
    $\mu := \mu + \Delta\mu$
  until $\nexists\ \mathbf{x}_i^k \in [\epsilon, 1 - \epsilon]$                                   ($\mu$-loop)
```

where either

$$D(\mathbf{x}; \sigma) := \|\mathbf{A}\ \mathbf{G}_\sigma\ \mathbf{x} - \mathbf{b}\|^2 \quad \text{or} \quad D(\mathbf{x}; \sigma) := \|\mathbf{G}_\sigma \mathbf{A}\ \mathbf{x} - \mathbf{b}\|^2 \tag{5.41}$$

and $S(\mathbf{x})$ indicates a smoothness prior, e.g. equation (5.32). The optimization of $E_\sigma(\mathbf{x}; \mu)$ in (5.40) is complicated through the unknown scale-parameter σ of the convolution operator $\mathbf{G}_\sigma$. A common and natural approach to solve this problem is to apply the well-known expectation-maximization (EM) [46, 93] iteration to the probabilistic interpretation of the data term $\mathbf{D}(\mathbf{x}; \sigma)$ as a likelihood term, provided this is computationally feasible. We briefly introduce the EM algorithm in the next section and elaborate our approach afterwards.

5.5.4 Expectation-Maximization (EM)

Let $\mathcal{X}$ be a random set of samples drawn from a distribution $\mathcal{P}$ parameterized with Θ where $\mathcal{X}$ is usually denoted as measured or observed data. The likelihood function is defined by

$$\mathcal{L}(\Theta|\mathcal{X}) := \mathcal{P}(\mathcal{X}|\Theta) \tag{5.42}$$

and is considered as a function of Θ where $\mathcal{X}$ is fixed. The maximum likelihood (ML) problem seeks for parameters Θ^* such that the likelihood function becomes maximal,

$$\Theta^* := \operatorname*{argmax}_{\Theta} \mathcal{L}(\Theta|\mathcal{X}), \tag{5.43}$$

at which often the log likelihood function $\log \mathcal{L}(\Theta|\mathcal{X})$ is used instead of $\mathcal{L}(\Theta|\mathcal{X})$. Given some estimate Θ_k of the parameters the idea is to compute an update Θ such that

$$\log \mathcal{L}(\Theta|\mathcal{X}) > \log \mathcal{L}(\Theta_k|\mathcal{X}) \tag{5.44}$$

which is equivalent to the maximization of

$$\log \mathcal{L}(\Theta|\mathcal{X}) - \log \mathcal{L}(\Theta_k|\mathcal{X}) = \log \mathcal{P}(\mathcal{X}|\Theta) - \log \mathcal{P}(\mathcal{X}|\Theta_k). \tag{5.45}$$

At this, the EM-algorithm additionally includes the case where some data is hidden or missing. Let us, therefore, denote the hidden parameters with $\mathcal{Y}$, then the total probability $\mathcal{P}(\mathcal{X}|\Theta)$ is obtained by marginalizing over $\mathcal{Y}$,

$$\mathcal{P}(\mathcal{X}|\Theta) = \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta). \tag{5.46}$$

For the subsequent derivation of equation (5.47) we take advantage of $\sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) = 1$ and use Jensen's inequality which proves that for a convex function f defined on an interval I, $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n \in I$, and $\lambda_1, \lambda_2, ..., \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$, the following inequality holds

$$f\left(\sum_{i=1}^{n} \lambda_i \mathbf{x}_i\right) \leq \sum_{i=1}^{n} \lambda_i f(\mathbf{x}_i).$$

Using Jensen's inequality, a lower bound on equation (5.45) is found by

$$\begin{aligned}
&\log \mathcal{L}(\Theta|\mathcal{X}) - \log \mathcal{L}(\Theta_k|\mathcal{X}) = \log \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta) - \log \mathcal{P}(\mathcal{X}|\Theta_k) \\
&\quad = \log \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta) \frac{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)}{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)} - \log \mathcal{P}(\mathcal{X}|\Theta_k) \\
&\quad = \log \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \left(\frac{\mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta)}{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)}\right) - \log \mathcal{P}(\mathcal{X}|\Theta_k) \\
&\quad \geq \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log \left(\frac{\mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta)}{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)}\right) - \log \mathcal{P}(\mathcal{X}|\Theta_k) \\
&\quad = \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log \left(\frac{\mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta)}{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)}\right) - \log \mathcal{P}(\mathcal{X}|\Theta_k) \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \\
&\quad = \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log \left(\frac{\mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta)}{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)\ \mathcal{P}(\mathcal{X}|\Theta_k)}\right) \\
&\quad =: \Delta(\Theta|\Theta_k)
\end{aligned} \tag{5.47}$$

which in turn yields a lower bound $B(\Theta|\Theta_k)$ on $\log \mathcal{L}(\Theta|\mathcal{X})$ defined by

$$\log \mathcal{L}(\Theta|\mathcal{X}) \geq B(\Theta|\Theta_k) := \log \mathcal{L}(\Theta|\mathcal{X}) + \Delta(\Theta|\Theta_k). \tag{5.48}$$

One observes that

$$\begin{aligned}
B(\Theta_k|\Theta_k) &= \log \mathcal{L}(\Theta_k|\mathcal{X}) + \Delta(\Theta_k|\Theta_k) \\
&= \log \mathcal{L}(\Theta_k|\mathcal{X}) + \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log\left(\frac{\mathcal{P}(\mathcal{X}|\mathbf{y},\Theta_k)\ \mathcal{P}(\mathbf{y}|\Theta_k)}{\mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k)\ \mathcal{P}(\mathcal{X}|\Theta_k)}\right) \\
&= \log \mathcal{L}(\Theta_k|\mathcal{X}) + \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log\left(\frac{\mathcal{P}(\mathcal{X},\mathbf{y}|\Theta_k)}{\mathcal{P}(\mathcal{X},\mathbf{y}|\Theta_k)}\right) \\
&= \log \mathcal{L}(\Theta_k|\mathcal{X}) + \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log 1 \\
&= \log \mathcal{L}(\Theta_k|\mathcal{X})
\end{aligned} \tag{5.49}$$

which states that the lower bound touches $\log \mathcal{L}(\Theta|\mathcal{X})$ at Θ_k. When maximizing $B(\Theta|\Theta_k)$, consequently, the new choice Θ_{k+1} is at least as good as the previous parameters Θ_k and, thus, the EM procedure converges towards a fix point, see [93] for detailed convergence results. For the maximization of $B(\Theta|\Theta_k)$, all terms independent of Θ are not essential and can therefore be omitted. Hence, we have

$$\begin{aligned}
\Theta_{k+1} :=& \operatorname*{argmax}_{\Theta} \{B(\Theta|\Theta_k)\} \\
=& \operatorname*{argmax}_{\Theta} \left\{ \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log\left(\mathcal{P}(\mathcal{X}|\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y}|\Theta)\right) \right\} \\
=& \operatorname*{argmax}_{\Theta} \left\{ \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log\left(\frac{\mathcal{P}(\mathcal{X},\mathbf{y},\Theta)\ \mathcal{P}(\mathbf{y},\Theta)}{\mathcal{P}(\mathbf{y},\Theta)\ \mathcal{P}(\Theta)}\right) \right\} \\
=& \operatorname*{argmax}_{\Theta} \left\{ \sum_{\mathbf{y}\in\mathcal{Y}} \mathcal{P}(\mathbf{y}|\mathcal{X},\Theta_k) \log\left(\mathcal{P}(\mathcal{X},\mathbf{y}|\Theta)\right) \right\} \\
=& \operatorname*{argmax}_{\Theta} \left\{ \mathbb{E}_{\mathbf{y}|\mathcal{X},\Theta_k} \left\{ \log\left(\mathcal{P}(\mathcal{X},\mathbf{y}|\Theta)\right) \right\} \right\}.
\end{aligned} \tag{5.50}$$

The EM algorithm iteratively evaluates the last expression of equation (5.50) which is the estimation (E-step) and the maximization (M-step) of the conditional expectation, as shown in algorithm 10.

Algorithm 10 EM Algorithm

```
Require: $\mathcal{X}$
Require: $\Theta_0$ - initial guess on the parameters.
repeat
    Setup the conditional expectation $\mathbb{E}_{\mathbf{y}|\mathcal{X},\Theta_k}\{\log(\mathcal{P}(\mathcal{X},\mathbf{y}|\Theta))\}$    (E-step)
    $\Theta_{k+1} \leftarrow \operatorname{argmax}_\Theta \{\mathbb{E}_{\mathbf{y}|\mathcal{X},\Theta_k}\{\log(\mathcal{P}(\mathcal{X},\mathbf{y}|\Theta))\}\}$    (M-step)
until $\Theta_{k+1} = \Theta_k$
```

5.5.5 Approach

Data Term and Scale Estimation

We regard the minimization of $E_\sigma(\mathbf{x};\mu)$ in (5.40) as maximum-a-posteriori (MAP) estimation of $\mathbf{x}$, given the data $\mathbf{b}$:

$$\mathcal{P}(\mathbf{x}|\mathbf{b}) \propto \exp\big(-E_\sigma(\mathbf{x};\mu)\big) \propto \mathcal{P}(\mathbf{b}|\mathbf{x})\mathcal{P}(\mathbf{x}) \tag{5.51}$$

$$\mathcal{P}(\mathbf{b}|\mathbf{x}) \propto \exp\big(-D(\mathbf{x};\sigma)\big) \tag{5.52}$$

$$\mathcal{P}(\mathbf{x}) \propto \exp\big(-\alpha S(\mathbf{x}) + \mu\frac{1}{2}\langle\mathbf{x},\mathbf{x}-\mathbf{1}\rangle\big) \tag{5.53}$$

The normalizing term in (5.51) is missing since it only depends on $\mathbf{b}$ and, hence, is unessential for estimating $\mathbf{x}$. The data likelihood $\mathcal{P}(\mathbf{b}|\mathbf{x})$ is unknown due to the dependency of the data term $D(\mathbf{x};\sigma)$ on the unknown parameter σ. Given some estimate $\hat{\mathbf{x}}$, the continuous counterpart of equation (5.50) reads

$$Q(\mathbf{x}|\hat{\mathbf{x}},\mathbf{b}) := \int_{\mathbb{R}_+} \mathcal{P}(\sigma|\mathbf{b},\hat{\mathbf{x}}) \log \mathcal{P}(\mathbf{b},\sigma|\mathbf{x})\, \mathrm{d}\sigma. \tag{5.54}$$

To compute (5.54), the first term under the integral is evaluated via Bayes' rule

$$\mathcal{P}(\sigma|\mathbf{b},\hat{\mathbf{x}}) = \frac{\mathcal{P}(\mathbf{b}|\sigma,\hat{\mathbf{x}})\mathcal{P}(\sigma|\hat{\mathbf{x}})}{\mathcal{P}(\mathbf{b}|\hat{\mathbf{x}})}.$$

The denominator does not depend on σ and is, therefore, unessential for the marginalization of σ in equation (5.54). The first term of the numerator is given by the data term $\mathcal{P}(\mathbf{b}|\sigma,\hat{\mathbf{x}}) = Z^{-1}\exp(-D)$, where Z is a normalizing constant. Furthermore, it is reasonable to assume independency $\mathcal{P}(\sigma|\mathbf{x}) = \mathcal{P}(\sigma)$ and, thus, we obtain

$$\mathcal{P}(\sigma|\mathbf{b},\hat{\mathbf{x}}) \propto \frac{1}{Z}\exp\big(-D(\hat{\mathbf{x}};\sigma)\big)\,\mathcal{P}(\sigma). \tag{5.55}$$

For the second term under the integral in (5.54), we compute

$$\log \mathcal{P}(\mathbf{b}, \sigma|\mathbf{x}) \ \propto\ \log \mathcal{P}(\mathbf{b}|\sigma, \mathbf{x}) + \log \mathcal{P}(\sigma) \ \propto\ -D(\mathbf{x}; \sigma) + \log \mathcal{P}(\sigma) \tag{5.56}$$

using again $\mathcal{P}(\sigma|\mathbf{x}) = \mathcal{P}(\sigma)$, and dropping the normalizing constant of the first term on the right, as explained above after the equations (5.51)–(5.53). Furthermore, we can drop the last term $\log \mathcal{P}(\sigma)$ in (5.56) because it neither depends on $\mathbf{x}$, nor does it contribute to the averaging of $D(\mathbf{x}; \sigma)$ with respect to σ. As a result, we insert the remaining term $-D(\mathbf{x}; \sigma)$, together with (5.55), into (5.54) and denote the resulting expression again with Q:

$$Q(\mathbf{x}|\hat{\mathbf{x}}, \mathbf{b}) := \int_{\mathbb{R}_+} \frac{1}{Z} \exp\big(-D(\hat{\mathbf{x}}; \sigma)\big)\, p(\sigma) \big(-D(\mathbf{x}; \sigma)\big)\, \mathrm{d}\sigma \tag{5.57}$$

This expression clearly shows how the unknown dependency on σ of the objective criterion (5.40) is dealt with. Given a current estimate $\hat{\mathbf{x}}$ and a prior distribution $\mathcal{P}(\sigma)$, the unknown data term $D(\mathbf{x}; \sigma)$ is replaced by maximizing the average (5.57). Consequently, we replace functional $E_\sigma(\mathbf{x}; \mu)$ in (5.40) with the approximation

$$E(\mathbf{x}; \hat{\mathbf{x}}, \mu) := -Q(\mathbf{x}|\hat{\mathbf{x}}, \mathbf{b}) + \alpha\, S(\mathbf{x}) - \mu \frac{1}{2} \langle \mathbf{x}, \mathbf{x} - \mathbf{1} \rangle. \tag{5.58}$$

In practice, we choose the prior $\mathcal{P}(\sigma)$ to be uniform within a reasonable interval $[\sigma_{\min}, \sigma_{\max}]$, and $\hat{\mathbf{x}}$ is the current estimate on $\mathbf{x}$. $Q(\mathbf{x}|\hat{\mathbf{x}}, \mathbf{b})$ is then evaluated by computing the one-dimensional integral (5.57) numerically using the trapezoidal rule, see figure 5.16.

Smoothness Term

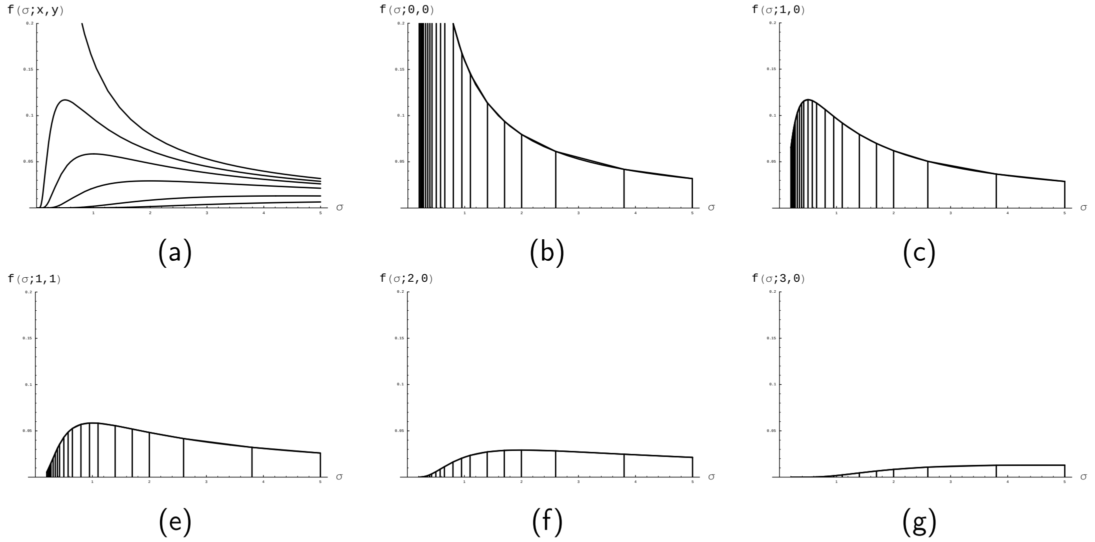
As smoothness prior $S(\mathbf{x})$ in (5.58), we choose a discrete approximation of the total-variation (TV) measure

$$\int_\Omega |\nabla \mathbf{x}|\; d\Omega$$

of $\mathbf{x}$ (here temporarily regarded as a function), whose edge-preserving properties are well-known in image processing [114]. This prior has also been successfully used in connection with discrete tomography, see [32].

5.5.6 Optimization

Following section (5.5.3), we include the convex set of feasible solutions into functional $E(\mathbf{x}; \hat{\mathbf{x}}, \mu)$, equation (5.58), by using the indicator function $\delta_{[0,1]^n}(\mathbf{x})$, equation (5.23), and consider the



(a) (b) (c)

(e) (f) (g)

Figure 5.16: In order to numerically approximate the integration in (5.57), we consider a $n \times n$ Gaussian filtermask as a function of σ. For instance, (a) plots the behavior of some mask entries, where the highest function corresponds to the central mask position. We use a subdivision scheme to find suitable support points for all functions such that the approximation becomes finer in areas of higher curvature and coarser otherwise. The plots (b)-(g) depict the approximation of the functions in (a). As can be seen there, the support points become very dense for small values of σ which makes it reasonable to choose $0 < \sigma_{\min}$ in order to reduce the number of support points. This cut-off is justified as the blurring effect caused by very small values of σ does hardly impact any reconstruction algorithm.

following d.c. decomposition

$$g(\mathbf{x};\hat{\mathbf{x}}) := -\,Q(\mathbf{x}|\hat{\mathbf{x}},\mathbf{b}) + \alpha\, S(\mathbf{x}) + \delta_{[0,1]^n}(\mathbf{x}) \tag{5.59}$$

$$h(\mathbf{x};\mu) := \mu\,\frac{1}{2}\langle \mathbf{x}, \mathbf{x}-\mathbf{1}\rangle. \tag{5.60}$$

The full reconstruction algorithm is listed in algorithm 11 where the estimation of the unknown scale-parameter σ through the EM-iteration, section 5.5.5, is done as part of the reconstruction algorithm. The global optimum of the convex optimization problem can be computed using any method. However, in view of the simple structure of the box-constraints $\mathbf{x} \in [0,1]^n$, we used the spectral projected gradient algorithm proposed in [14], appendix D, in our implementation.

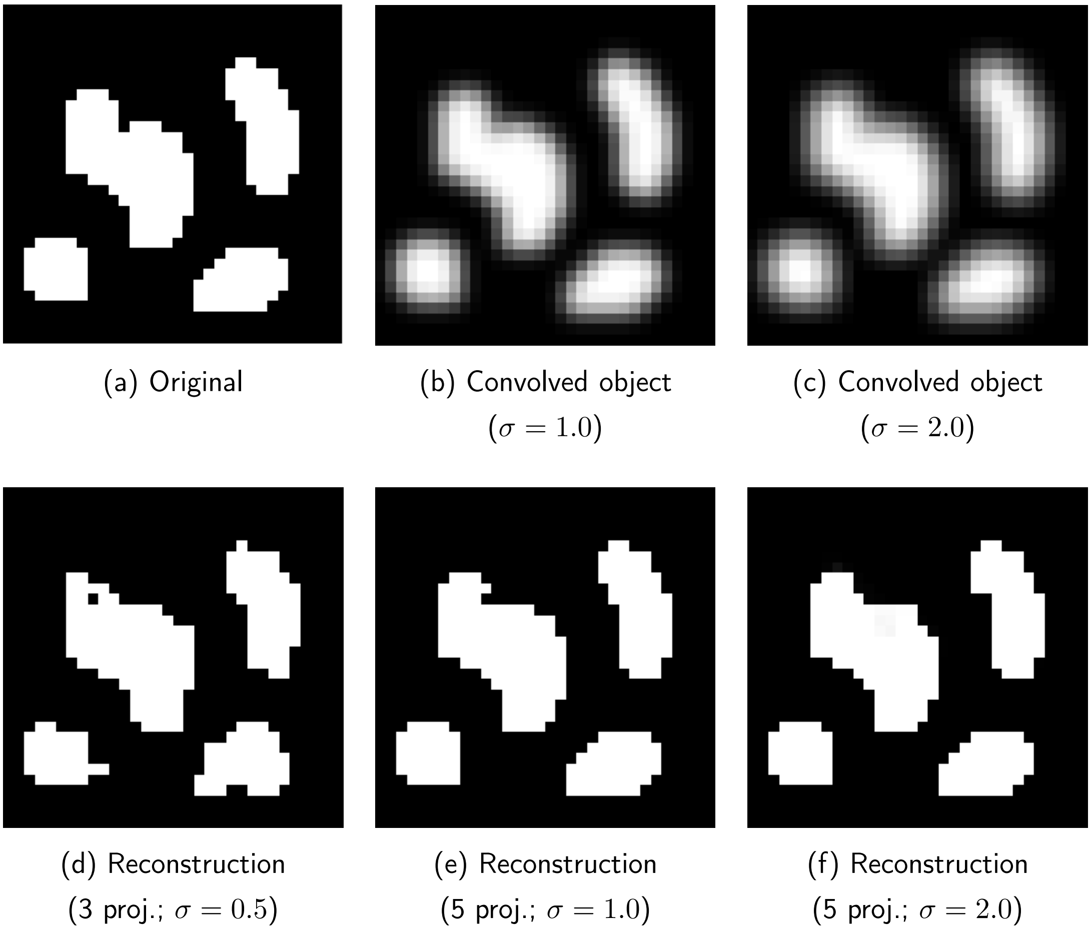
Algorithm 11 DC-Framework with Quadratic Programming and Deblurring

```
Require: Choose $\mathbf{x}^0$ arbitrary                (our choice: $\mathbf{x}^0 := (\frac{1}{2}, ..., \frac{1}{2})^\top$)
Require: $0 < \Delta\mu$                                (our choice: $\Delta\mu \in (0, 0.5]$)
Require: $0 < \epsilon$                                 (our choice: $10^{-4} \leq \epsilon \leq 10^{-2}$)
  $\mu := 0$
  repeat
    $k := 0$
    repeat
      $\mathbf{y}^k := \nabla h(\mathbf{x}^k; \mu) = \mu\,(\mathbf{x}^k - \frac{1}{2}\mathbf{1})$
      $l := 0$
      $\hat{\mathbf{x}}^0 := \mathbf{x}^k$
      repeat
        $\hat{\mathbf{x}}^{l+1} := \underset{\mathbf{x} \in [0,1]^n}{\operatorname{argmin}} \left\{ g(\mathbf{x}; \hat{\mathbf{x}}^l) - \langle \mathbf{y}^k, \mathbf{x} \rangle \right\}$
        $l := l + 1$
      until $||\hat{\mathbf{x}}^l - \hat{\mathbf{x}}^{l-1}||_2 < \epsilon$            (EM-loop)
      $\mathbf{x}^{k+1} := \hat{\mathbf{x}}^l$
      $k := k + 1$
    until $||\mathbf{x}^k - \mathbf{x}^{k-1}||_2 < \epsilon$                (DC-loop)
    $\mu := \mu + \Delta\mu$
  until $\nexists\, \mathbf{x}_j^k \in [\epsilon, 1 - \epsilon]\,,\ j = 1, \ldots, n$      ($\mu$-loop)
```

5.5.7 Evaluation

In figure 5.15, we showed that the binary reconstruction fails in case of blurred objects. We repeated the experiment, however, this time taking deblurring into account. The results shown in figure 5.17 reveal that our novel reconstruction algorithm copes with both problems, deblurring by scale-parameter estimation and binary reconstruction, at the same time. Further experiments showed, that the original object can be reconstructed in case of $\sigma = 1.0$ even from four projections ($0°$, $45°$, $90°$, and $135°$).

Concerning reconstructions from blurred projections, the upper-left image shown in figure 5.18 was projected along four directions $0°$, $45°$, $90°$, $135°$. Panel (b) shows these projections for illustration, and panel (c) its blurred version ($\sigma = 1.5$). The latter data was used to compute the reconstruction shown in panel (f). Panels (d) and (e) show the reconstruction results for $\sigma = 1.0$ with and without deblurring, respectively. While the latter result clearly shows


(a) Original

(b) Convolved object ($\sigma = 1.0$)

(c) Convolved object ($\sigma = 2.0$)

(d) Reconstruction (3 proj.; $\sigma = 0.5$)

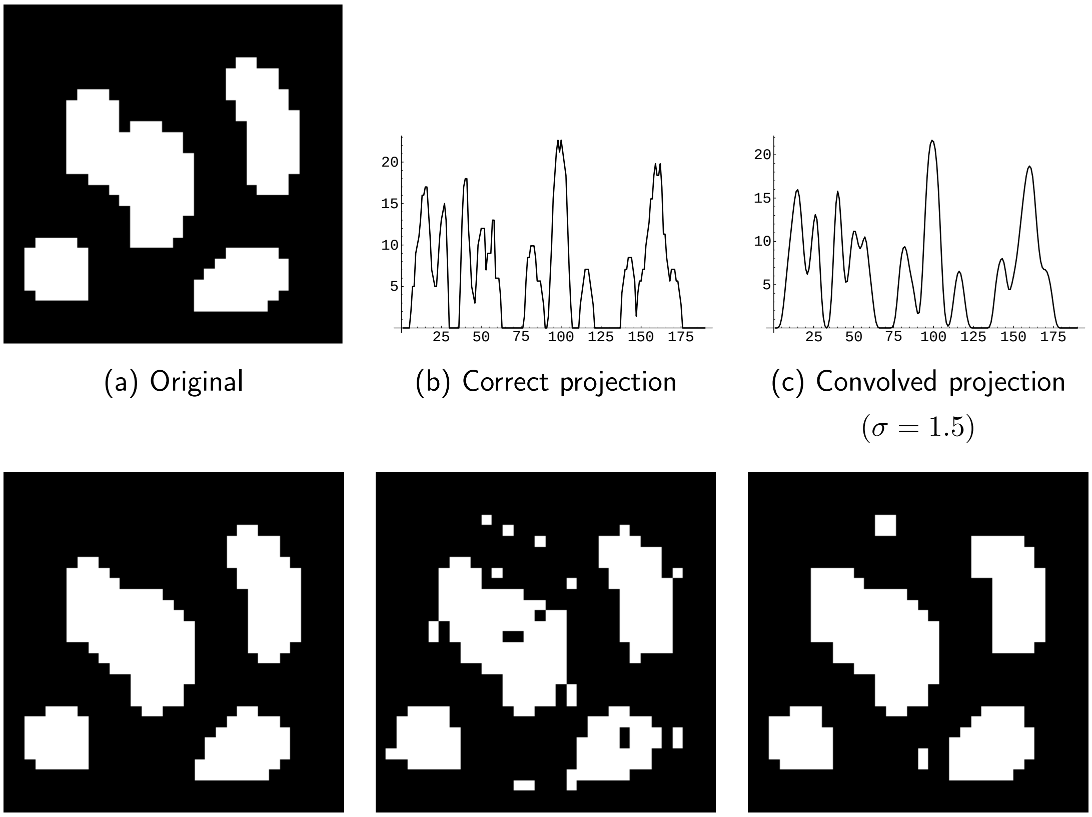
(e) Reconstruction (5 proj.; $\sigma = 1.0$)

(f) Reconstruction (5 proj.; $\sigma = 2.0$)

Figure 5.17: Reconstruction from blurred objects: (a) Original image, 32×32. (b) and (c): original image convolved with different Gaussian kernels, $\sigma \in \{1.0, 2.0\}$. 5 projections were taken for both images ($0°$, $22.5°$, $45°$, $67.5°$, $90°$). Figures (d) and (f) show the corresponding results of our reconstruction algorithm. Since we obtained for $\sigma = 0.5$ the original image we present in this case the reconstruction from only three projections, $0°$, $45°$, and $90°$. Throughout the experiments the smoothing parameter α was set to 0.01.

the ill-posedness of the combined deblurring-reconstruction problem, the results (d) and (f) demonstrate the stability of our reconstruction algorithm even under such severe conditions. Additional reconstruction results from blurred projections can be found in figures 5.20 and 5.21.

To illustrate the deblurring process further, figure 5.19 depicts the expressions $\exp(-D(\hat{x};\sigma))/Z$ and $D(\hat{x};\sigma)$, respectively, as a function of σ during the experiment in figure 5.18 (c). It can be clearly seen that the former expression peaks most around the correct value $\sigma = 1.5$, whereas the latter term attains its global minimum there. Our experiments also revealed that reconstructions from blurred projections are typically more difficult than from projections of blurred objects.



(a) Original

(b) Correct projection

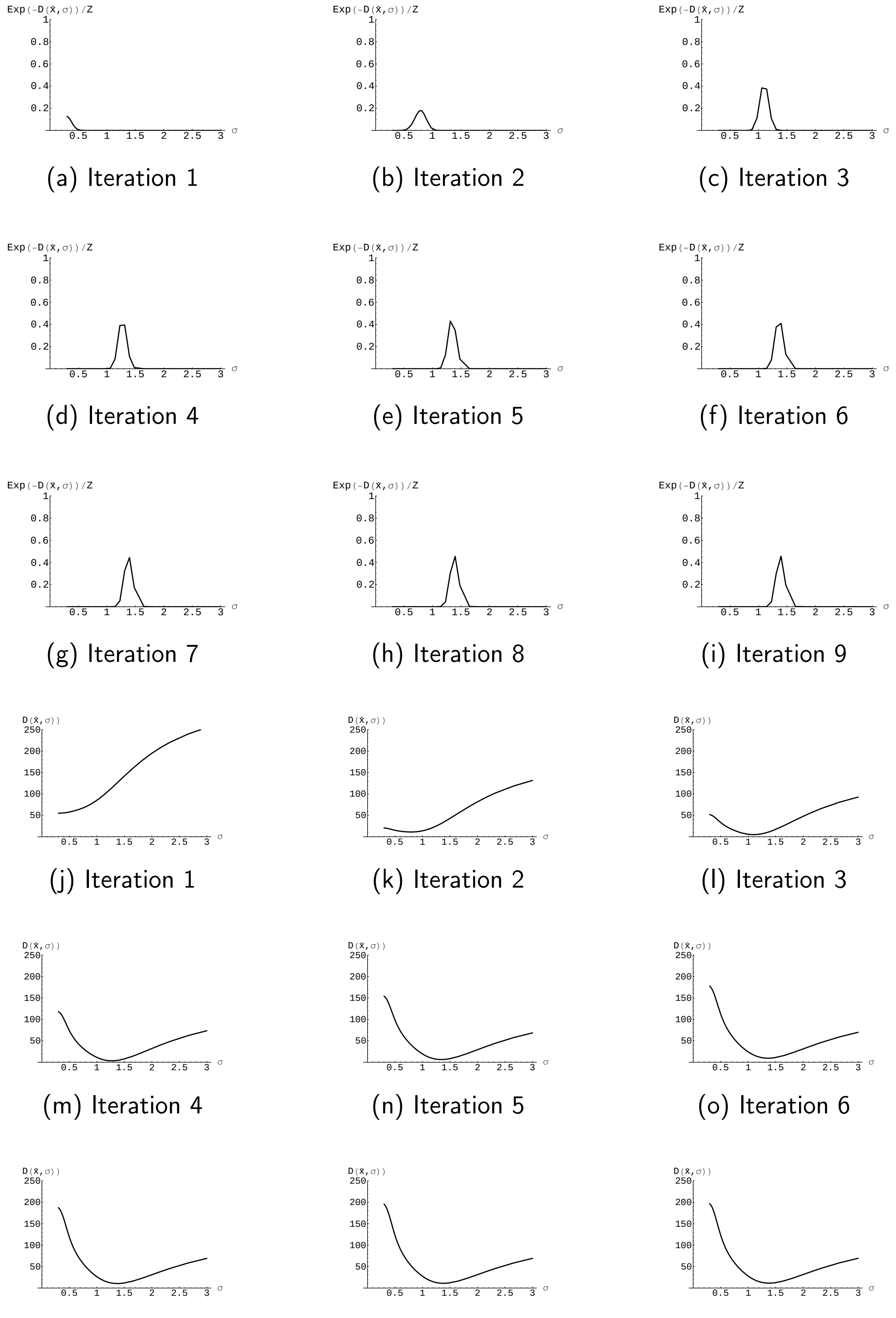
(c) Convolved projection ($\sigma = 1.5$)

(d) With deblurring ($\sigma = 1.0, \alpha = 0.01$)

(e) Without deblurring ($\sigma = 1.0, \alpha = 0.01$)

(f) With deblurring ($\sigma = 1.5, \alpha = 0.05$)

Figure 5.18: Reconstruction from blurred projections: Projections at $0°$, $45°$, $90°$, and $135°$ were taken from the image shown in panel (a) and convolved with a Gaussian kernel, $\sigma = 1.5$. Panels (b) and (c) show the correct projections and the blurred projections, respectively. Panel (d) and (e) show the reconstruction from projection data blurred with $\sigma = 1.0$ with and without taking deblurring into account. Panel (f) shows the reconstruction result for $\sigma = 1.5$ with $\alpha = 0.05$.



(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

(d) Iteration 4

(e) Iteration 5

(f) Iteration 6

(g) Iteration 7

(h) Iteration 8

(i) Iteration 9

(j) Iteration 1

(k) Iteration 2

(l) Iteration 3

(m) Iteration 4

(n) Iteration 5

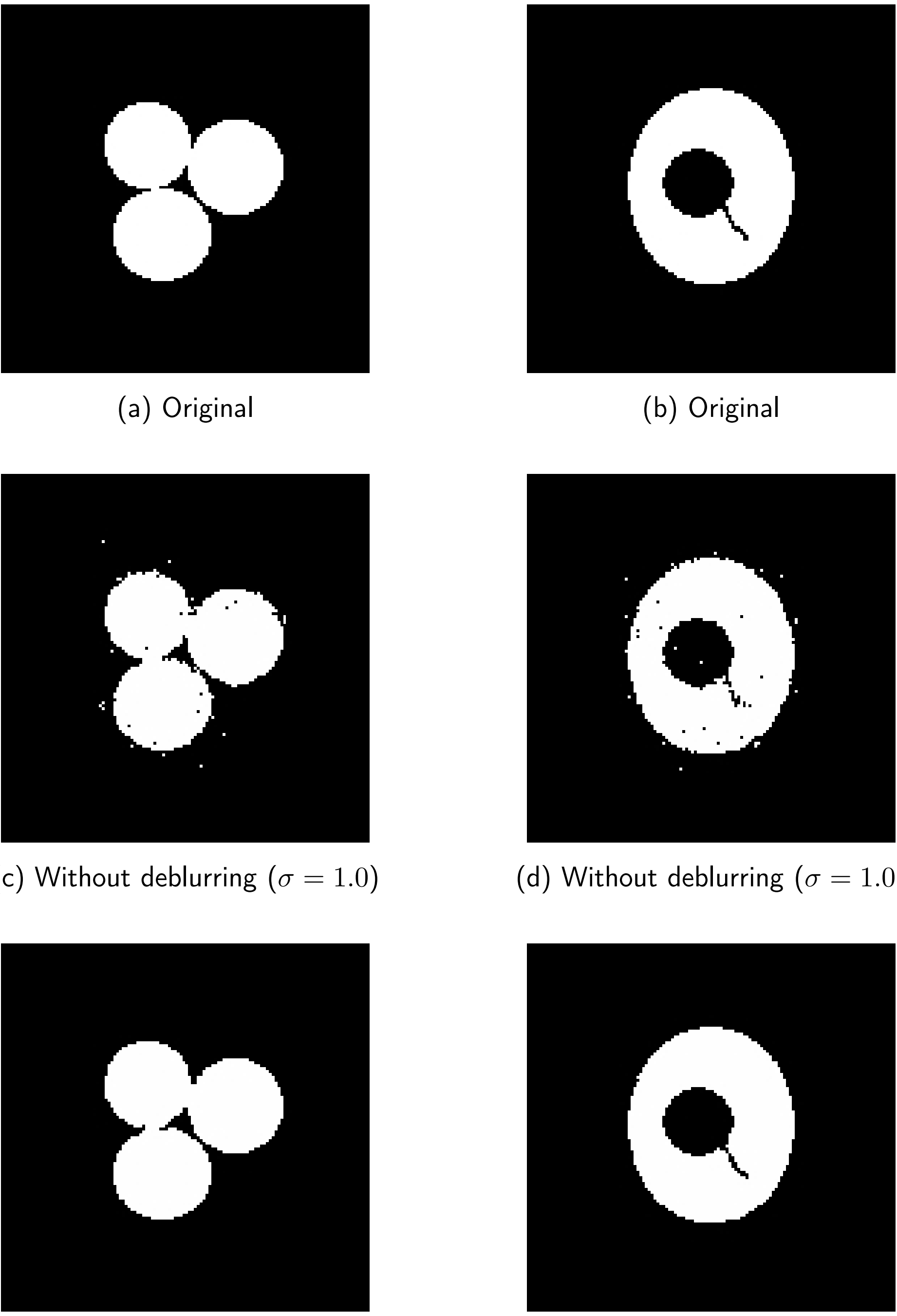
(o) Iteration 6

(p) Iteration 7

(q) Iteration 8

(r) Iteration 9

Figure 5.19: The $\exp(-D(\hat{x};\sigma))/Z$ term, upper half, and $D(\hat{x};\sigma)$ term, lower half, as a function of σ at various iterations. While the former term peaks most near the correct value $\sigma = 1.5$, the latter attains its global minimum there.



(a) Original

(b) Original

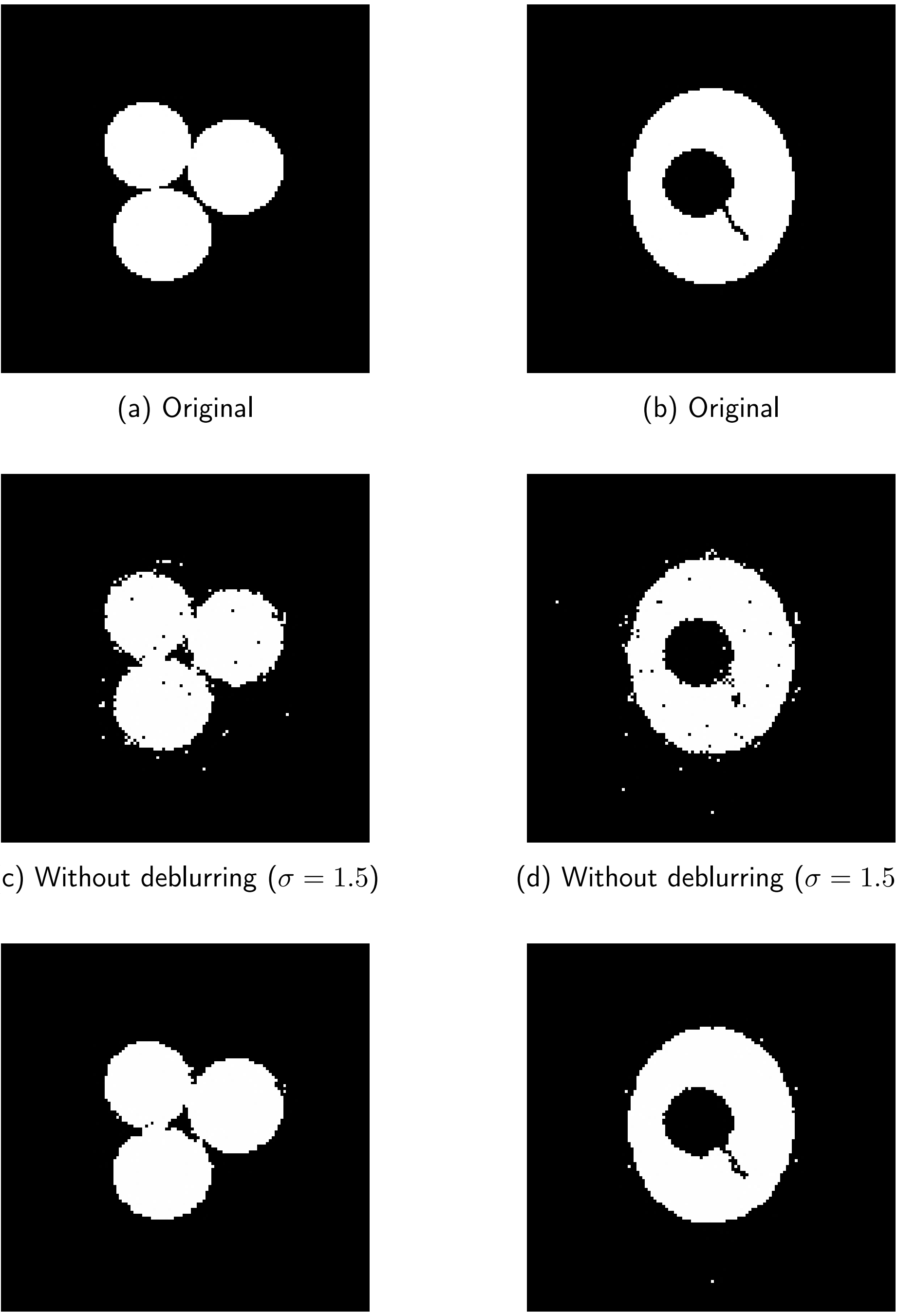
(c) Without deblurring ($\sigma = 1.0$)

(d) Without deblurring ($\sigma = 1.0$)

(e) With deblurring ($\sigma = 1.0$)

(f) With deblurring ($\sigma = 1.0$)

Figure 5.20: Reconstruction from blurred projections: (a),(b) Original image, 128×128. For both images, reconstruction problems were set up from 5 projections, $0°$, $36°$, $72°$, $108°$, and $144°$, each blurred with a Gaussian kernel, $\sigma = 1.0$. (c),(d) Reconstruction without deblurring. (e),(f) Reconstruction with deblurring.



(a) Original

(b) Original

(c) Without deblurring ($\sigma = 1.5$)

(d) Without deblurring ($\sigma = 1.5$)

(e) With deblurring ($\sigma = 1.5$)

(f) With deblurring ($\sigma = 1.5$)

Figure 5.21: Reconstruction from blurred projections: (a),(b) Original image, 128×128. From both images reconstruction problems were setup from 5 projections, $0°$, $36°$, $72°$, $108°$, and $144°$, and the projections were convolved with a Gaussian kernel, $\sigma = 1.5$. (c),(d) Reconstruction without deblurring. (e),(f) Reconstruction with deblurring.

6 Multi-Valued Discrete Reconstruction

In the preceding chapters we considered the reconstruction of binary objects and successfully derived a family of algorithms based on convex-concave regularization and d.c. programming. Therefore, the question about an extension of our approach to the general case of discrete tomography, i.e. reconstruction of objects made up from multiple discrete values, seems to arise naturally. Besides, the fact that the related literature is mostly concentrated on binary tomography endows us with further motivation to put our efforts in this research direction.

A first attempt to the multi-valued problem has been published in [123], where, in analogy to the binary case, the space between any two consecutive discrete values is relaxed to an interval. The concave regularizer, equation (5.9), is placed in each interval, and starting from the convex solution the intermediate values are pushed towards the boundaries of the corresponding interval. This method is easy to implement and does not need additional variables but includes a heuristic step which allows the pixels to move from one interval to another. Consequently, a pixel has to travel through several intervals in order to reach more distant values and is likely to get stuck.

Regarding to this previous method we pursue a different strategy here which does not require any heuristic step but involves additional variables. As we will see, this new method indeed generalizes the family of d.c.-based algorithms to multiple classes. In addition, we show that the global optimality conditions proposed in [10] for the binary quadratic programming, section 5.5.3, can be adapted to the multiclass case as well.

The formal description of the reconstruction problem we use here allows our approach to become more appealing to a larger class of optimization problems, for instance image labeling. Therefore, we consider the image labeling problem besides the reconstruction problem in the evaluation part of this chapter, section 6.5.1, and compare our results to labeling algorithms based on graphcuts [23, 24, 82], section 3.3.3, semidefinite programming (SDP) [76, 75], and second-order cone programming (SOCP) [97, 88].

6.1 Problem Statement

We consider objects that are composed of different, not necessarily two, homogeneous materials and conceive the corresponding densities as associated with prototypical grayvalues $g_j \in \mathbb{R}$, $j \in \{1, ..., l\}$ in the image domain. We synonymously refer to the grayvalues g_j as labels and, thus, define the set of labels as $\mathcal{L} := \{g_1, ..., g_l\}$ and the label vector as $\mathbf{g} := (g_1, ..., g_l)^\top$. Additionally, it is convenient to assume that $g_j < g_k$ for $j < k$.

A single pixel $i \in \{1, ..., n\}$ is represented by a vector $\mathbf{x}_i := (x_{i,1}, ..., x_{i,l})^\top$ where each component corresponds to a label. If pixel i attains label j we want component $x_{i,j}$ to be 1 and all others to be 0. Consequently, we want $\mathbf{x}_i$ to be a canonical basis vector of $\mathbb{R}^l$ and we refer to the set of all 0-1 representations as $\Omega_{\{0,1\}} := \{\mathbf{e}_j \;|1 \leq j \leq l\} \subset \{0, 1\}^l$.

The entire image is described by a set of binary vectors $\mathbf{x}_i^\top$, $i \in \{1, ..., n\}$, which can be written in form of the assignment matrix $\mathbf{X}$ simply by stacking all $\mathbf{x}_i^\top$ on top of each other,

$$\mathbf{X} := \begin{pmatrix} x_{1,1} & \cdots & x_{1,l} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,l} \end{pmatrix} =: (\mathbf{x}_{*,1}, ..., \mathbf{x}_{*,l}), \tag{6.1}$$

where the columns of $\mathbf{X}$ are denoted by $\mathbf{x}_{*,j}$.

Using this notation, we state the multiclass reconstruction problem,

$$\mathbf{A}\,\mathbf{X}\,\mathbf{g} = \mathbf{b}, \quad \mathbf{x}_i \in \Omega_{\{0,1\}},\ \forall i, \tag{6.2}$$

in analogy to the binary reconstruction problem, section 3.2.

Sometimes, however, it is convenient to gather all unknowns $x_{i,j}$ in a single vector,

$$\mathbf{x} := (x_{1,1}, \ldots, x_{1,l}, x_{2,1}, \ldots, x_{2,l}, \ldots\ldots, x_{n,1}, \ldots, x_{n,l})^\top, \tag{6.3}$$

which we call assignment vector. Hence, the reconstruction problem, equation (6.2), can be equivalently written in terms of the assignment vector $\mathbf{x}$,

$$\mathbf{A}\,\mathbf{G}\,\mathbf{x} = \mathbf{b}, \quad \mathbf{x}_i \in \Omega_{\{0,1\}},\ \forall i, \tag{6.4}$$

where $\mathbf{G} := \mathrm{diag}(\mathbf{g}^\top)$ is a n-blockdiagonal matrix.

6.2 Approach

6.2.1 Relaxation

In order to avoid the inherent combinatorial complexity of the problem, equation (6.2) and equation (6.4), we relax the difficult constraint $\mathbf{x}_i \in \Omega_{\{0,1\}}$ to

$$\mathbf{x}_i \in \Omega_1 := \{\mathbf{x}_i \in \mathbb{R}^l \mid \mathbf{x}_i \geq 0 \text{ and } \sum_{j=1}^{l} x_{i,j} = 1\} \tag{6.5}$$

which is known as the standard simplex, canonical simplex, or probability simplex. In other words, this is the convex set spanned by the standard basis vectors or more importantly for us the elements of $\Omega_{\{0,1\}}$. Notice, that this is also equivalent to consider $\mathbf{x}_i$ as the Barycentric coordinates with respect to the vectors in $\Omega_{\{0,1\}}$.

We can easily imagine this set in case of only three labels, see figure 6.1 where black dots depict the elements of the discrete set $\Omega_{\{0,1\}}$ and the triangular area depicts the relaxed space Ω_1.

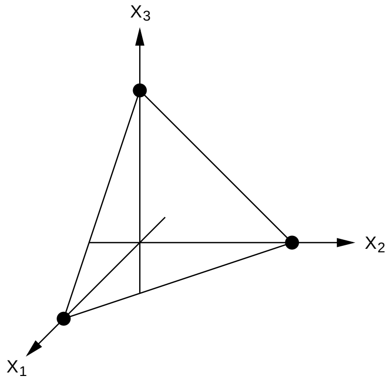


Figure 6.1: Besides the formal discussion of our approach, we subsequently consider the case of three labels for illustration purposes. In this scenario Ω_1, equation (6.5), equals the triangular area spanned by the unit vectors in $\mathbb{R}^3$, depicted by the black dots.

Concerning the entire set of pixels the overall space is defined by $\Omega_1^n := \Omega_1 \times \cdots \times \Omega_1$ which inherits convexity from Ω_1, equation (6.5).

6.2.2 Concave Regularization

Binary Concave Regularizer

It is tempting to apply the concave regularization term, equation (5.9), from the previous chapter also in the multiclass scenario,

$$\langle \mathbf{x}, \mathbf{1} - \mathbf{x} \rangle, \quad \mathbf{x} \in \Omega_1^n. \tag{6.6}$$

As we will explain here this attempt does, however, not yield the desired result. For illustration purposes we examine the regularizer in a scenario involving only three labels, $\mathcal{L} := \{g_1, g_2, g_3\}$, but it should become clear that this does not limit our considerations. Figure 6.2 (a-c) shows the concave regularizer, equation (6.6), for a single pixel i. Starting in figure 6.2(a), μ is set to zero $\mathbf{x}_i$ and no penalty is imposed on $\mathbf{x}_i$. By increasing μ, $\mathbf{x}_i$ is pushed towards the vertices of Ω_1, figure 6.2(b,c). Akin to the binary case, the regularizer vanishes if $\mathbf{x}_i \in \Omega_{\{0,1\}}$ and therewith does not alter the value of an energy functional at the vertices of Ω_1.

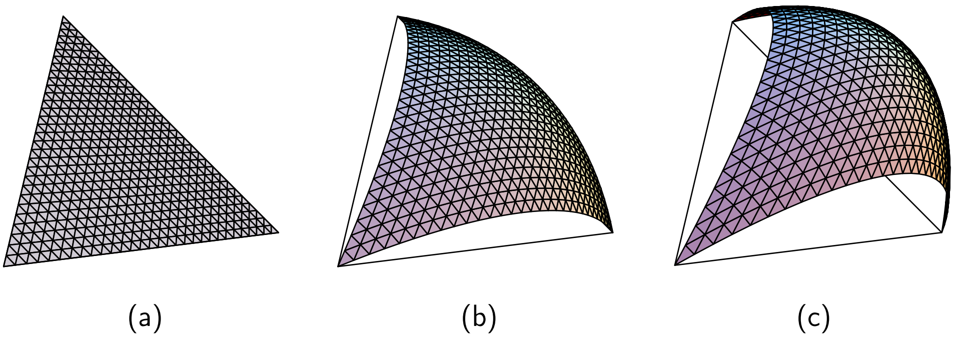


Figure 6.2: Concerning a single pixel in case of three labels, figures (a)-(c) show the concave regularizer which is derived from the binary case, equation (6.6), for increasing values of μ. For illustrative purposes the function is projected onto the set Ω_1, compare figure 6.1.

Concerning equation (6.2) a potential data term $D(\mathbf{x})$ typically measures the difference between $\mathbf{A}\ \mathbf{G}\ \mathbf{x}$ and $\mathbf{b}$. For simplicity we set $\mathbf{A} := \mathbf{I}$ and $D(\mathbf{x}) := \|\mathbf{G}\ \mathbf{x} - \mathbf{b}\|_2^2$ and focus on a singe pixel i with b being its corresponding component in $\mathbf{b}$. By this, we are concerned with $(\langle \mathbf{g}, \mathbf{x}_i \rangle - b)^2$ which is plotted in figure 6.3 for different values of b. As can be observed, $D(\mathbf{x})$ attains a unique minimum if $b = g_1$, figure 6.3 (a), and $b = g_3$, figure 6.3 (c). However, in case of $b = g_2$ a linear subspace exists where $D(\mathbf{x})$ becomes minimal, figure 6.3 (b), since g_2 can be linearly combined, $g_2 = \langle \mathbf{g}, \mathbf{x}_i \rangle$, $\mathbf{x}_i \in \Omega_1$. In general any value between $g_1 < g_l$ is represented by the intersection of an affine subspace with Ω_1, figure 6.4.

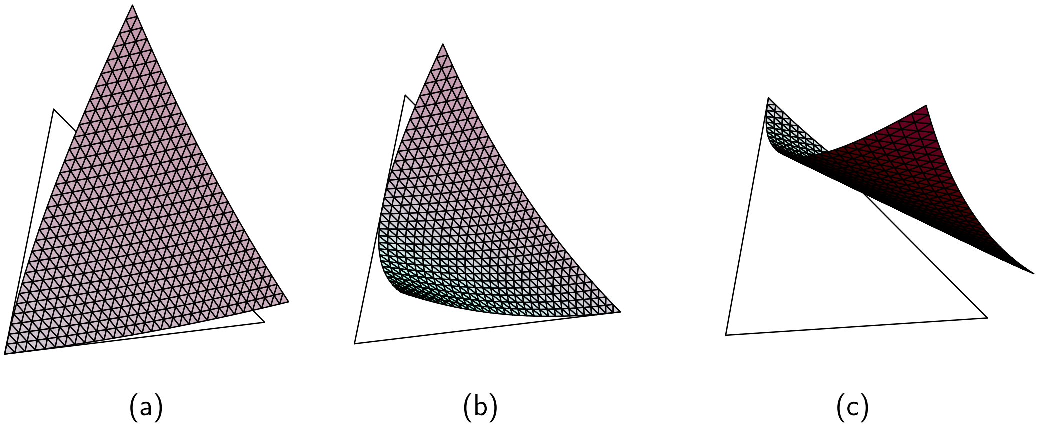


Figure 6.3: Similar to figure 6.2, we consider a single pixel i and a gray value $\tilde{g}$, $g_1 \leq \tilde{g} \leq g_3$, and assume that both are related due to the simplified data term $D_{\tilde{g}}(\mathbf{x}_i) := (\langle \mathbf{g}, \mathbf{x}_i \rangle - \tilde{g})^2$. The plots illustrate the data term in case of (a) $\tilde{g} := g_1$, (b) $\tilde{g} := g_2$, and (c) $\tilde{g} := g_3$. As can be seen in (a) and (c) the minima are unique with respect to Ω_1, figure 6.1, and coincide with the correct corner. In (b), however, there is a one-dimensional subspace of Ω_1 where $D_{g_2}(\mathbf{x})$ becomes minimal.

Imagine now, the data term in figure 6.3 (b) superimposed with the concave regularizer rearing up as shown in figure 6.1. As the data term becomes minimal within a linear subspace the concave regularizer is allowed to push $\mathbf{x}_i$ either towards the vertex representing g_2 as desired or towards the opposite direction, i.e. the edge between the vertices g_1 and g_3. If the latter case happens the optimizer gets trapped in a local optimum from which it becomes unlikely to escape. Consequently, pixel i will be incorrectly labeled either g_1 or g_3.

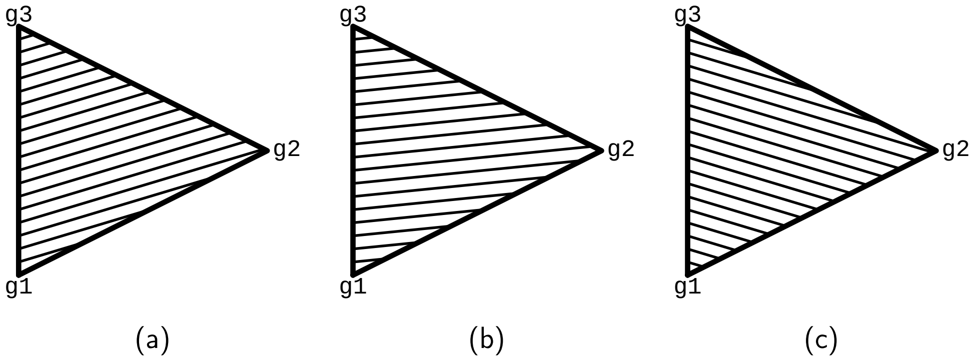


Figure 6.4: Figures (a)-(c) show the set Ω_1 in the three label case, $g_1 < g_2 < g_3$, see also figure 6.1. The hatched lines indicate the affine subspaces which proceed parallel to the lines and are aligned depending on the ratio between the grayvalues g_1, g_2, g_3. Each subspace uniquely represents a grayvalue g, $g_1 < g < g_3$.

Multiclass Concave Regularizer

As described above, the convex-concave regularization of the multiclass case is difficult as linear combinations emerge. In order to cope with this situation, we introduce a novel regularizer as replacement for the concave regularizer, equation (6.6),

$$\mu \sum_{i=1}^{n} \sum_{j=1}^{l} c(g_j, \langle \mathbf{g}, \mathbf{x}_i \rangle)\, x_{i,j} \; . \tag{6.7}$$

Depending on $\mathcal{L}$, each position $\mathbf{x}_i \in \Omega_1$ is associated with some affine subspace which represents some value $\tilde{g} := \langle \mathbf{g}, \mathbf{x}_i \rangle$, $g_1 \leq \tilde{g} \leq g_l$. Our new regularizer, equation (6.7), includes a cost function $c(\cdot)$ which measures the difference between $\tilde{g}$ and each label in $\mathcal{L}$. As a suitable choice for $c(\cdot)$ we confine ourselves to

$$c(g_j, \langle \mathbf{g}, \mathbf{x}_i \rangle) := (g_j - \langle \mathbf{g}, \mathbf{x}_i \rangle)^2. \tag{6.8}$$

As can be seen in equation (6.7), the costs associated with label g_j are further linked to the corresponding component in $\mathbf{x}_i$. Thus, it becomes preferable to assign higher values to $x_{i,j}$ if g_j is similar to $\tilde{g}$ and lower values if g_j differs from $\tilde{g}$. By increasing μ the cost function $c(\cdot)$ becomes more and more concentrated on the label that fits best.

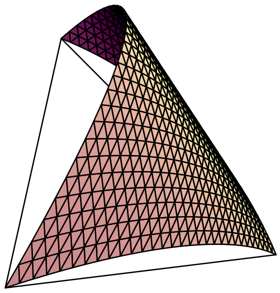


Figure 6.5: This plot shows the multiclass regularizer, equations (6.7) and (6.8), in case of three labels. Comparing this to the regularizer originating from the binary case in figure 6.2 it becomes perceivable how the new regularizer overcomes the difficulties discussed in section 6.2.2.

Let $\mathbf{x} \in \Omega_{\{0,1\}}$, without limitation $\mathbf{x} = \mathbf{e}_i$, then we have

$$\sum_{j=1}^{l} c(g_i, \langle \mathbf{g}, \mathbf{x}_i \rangle) x_{i,j} = (g_i - \langle \mathbf{g}, \mathbf{e}_i \rangle)^2 = (g_i - g_i)^2 = 0. \tag{6.9}$$

Thus, added to an energy function $E(\mathbf{x})$ our new regularizer vanishes and does not alter the value of $E(\mathbf{x})$ if $\mathbf{x} \in \Omega_1^n$ attains a 0-1 representation.

Concerning concavity, we derive for our particular choice of $c(\cdot)$, equation (6.8),

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{l} c(g_j, \mathbf{g}^\top \mathbf{x}_i)\; x_{i,j} = \sum_{i=1}^{n}\sum_{j=1}^{l} \Big(g_j - \mathbf{g}^\top \mathbf{x}_i\Big)^2 \; x_{i,j} = \\
&\sum_{i=1}^{n}\sum_{j=1}^{l} \Big(g_j^2 - 2 g_j \mathbf{g}^\top \mathbf{x}_i + (\mathbf{g}^\top \mathbf{x}_i)^2\Big) \; x_{i,j} = \\
&\sum_{i=1}^{n}\sum_{j=1}^{l} \Big(g_j^2 x_{i,j} - 2 g_j \; \mathbf{g}^\top \mathbf{x}_i \; x_{i,j} + (\mathbf{g}^\top \mathbf{x}_i)^2 x_{i,j}\Big) = \\
&\sum_{i=1}^{n} \left(\sum_{j=1}^{l} g_j^2 x_{i,j} - \sum_{j=1}^{l} 2 g_j \; \mathbf{g}^\top \mathbf{x}_i \; x_{i,j} + \sum_{j=1}^{l} (\mathbf{g}^\top \mathbf{x}_i)^2 x_{i,j}\right) = \\
&\sum_{i=1}^{n} \left(\sum_{j=1}^{l} g_j^2 x_{i,j} - 2\; \mathbf{g}^\top \mathbf{x}_i \sum_{j=1}^{l} g_j \; x_{i,j} + (\mathbf{g}^\top \mathbf{x}_i)^2 \sum_{j=1}^{l} x_{i,j}\right) = \\
&\sum_{i=1}^{n} \left(\sum_{j=1}^{l} g_j^2 x_{i,j} - 2\; (\mathbf{g}^\top \mathbf{x}_i)^2 + (\mathbf{g}^\top \mathbf{x}_i)^2\right) = \\
&\sum_{i=1}^{n} \left(\sum_{j=1}^{l} g_j^2 x_{i,j} - (\mathbf{g}^\top \mathbf{x}_i)^2\right) \qquad (6.10)
\end{aligned}$$

By exploiting the sum constraint from Ω_1, equation (6.5), in the transformations above, our regularizer reduces to a new function

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{l} g_j^2 x_{i,j} - (\mathbf{g}^\top \mathbf{x}_i)^2\right) \qquad (6.11)$$

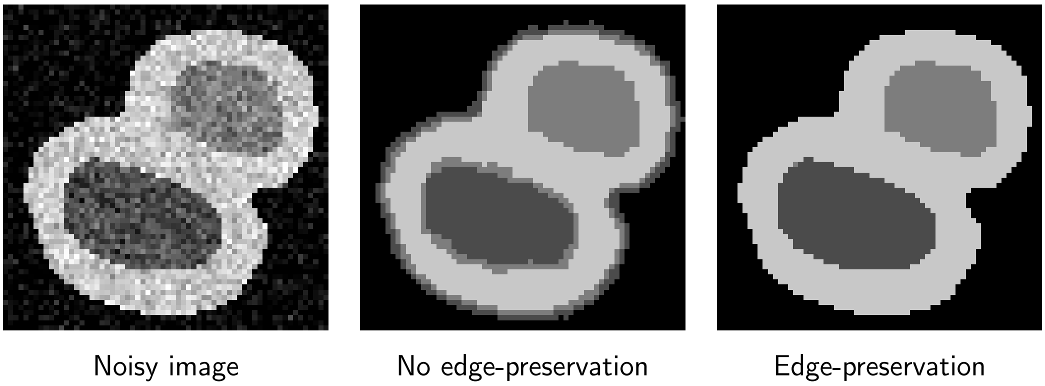
which equals our regularizer on Ω_1 and can be different outside of Ω_1. Consequently, concavity of function (6.11) on $\mathbb{R}^{n \times l}$ directly implies concavity of our regularizer on Ω_1. Therefore, we consider the Hessian of function (6.11) which is a n-blockdiagonal matrix with each block containing $-2\;(\;\mathbf{g}\;\mathbf{g}^\top)$. For an arbitrary $\mathbf{z} \in \mathbb{R}^l$, we have that

$$-2\;\mathbf{z}^\top(\;\mathbf{g}\;\mathbf{g}^\top)\;\mathbf{z} = -2\;(\mathbf{z}^\top \mathbf{g})^2 \leq 0$$

which shows the negative semidefiniteness of the Hessian matrix and, thus, implicitly shows that our novel regularizer, equation (6.7), is indeed concave on Ω_1.

6.2.3 Discontinuity-preserving Smoothness Priors

It is well-known that some smoothness priors, like for instance $\sum_{\langle i,j\rangle}(x_i - x_j)^2$, tend to blur the borders of objects. Therefore, edge or discontinuity preserving smoothness priors are usually applied in order to circumvent this effect, see figure 6.6. The image on the right side originally contained four different grayvalues but has been corrupted by noise. The problem is to remove the noise in order to recover the original image. This is a typical computer vision task which we will give a closer look in section 6.5.1 as it is quite related to our discrete tomography problem.



Noisy image — No edge-preservation — Edge-preservation

Figure 6.6: The left image shows an image which originally contained four graylevels, see figure 6.11 (a). However, the image is moderately distorted with noise and a typical computer vision task is to recover the original image, see section 6.5.1. Using a smoothness prior without edge-preservation yields unsatisfying results as it blurs the borders of different regions, middle image. This effect is circumvented if a edge- or discontinuity-preserving smoothness prior is used instead, right image.

As long as we considered binary images, like in the previous chapters, the blurring effect could be neglected since a single pixel was represented by a single variable and the concave prior decided the variable to be either 0 or 1. With respect to our approach we suggest two priors here which preserve discontinuities and suit our optimization framework.

A common way to introduce edge-preservation is to truncate the smoothness prior after a certain level such that larger deviations between neighboring pixels are constantly penalized. These priors are called robust priors and are non-convex due to the truncation. Hence, the applicability of robust priors is restricted, however, we will see that they fit quite naturally into our d.c. framework.

We consider the robust smoothness prior $S_I(\mathbf{x})$ defined by

$$S_I(\mathbf{x}) := \sum_{\langle i,j\rangle} f_c(\langle \mathbf{g}, \mathbf{x}_i\rangle - \langle \mathbf{g}, \mathbf{x}_j\rangle), \quad \text{with} \quad f_c(t) := \begin{cases} \frac{1}{c^2}\, t^2 & \text{if} \quad |t| < c \\ 1 & \text{otherwise} \end{cases}, \tag{6.12}$$

which is a truncated quadratic function [15] where parameter c adjusts the truncation. As illustrated in figure 6.7, $S(\mathbf{x})$ is d.c. decomposable and a formal decomposition is given in equation (6.13). It is incorporated into our d.c. framework simply by adding each term to the appropriate part of the energy functional, i.e. either $g(\mathbf{x})$ or $h(\mathbf{x})$. At this, the fact that the decomposition, equation (6.13), includes a non-differentiable term, $h_c(t)$, is not obstructive since the d.c. framework is based on subdifferentials.

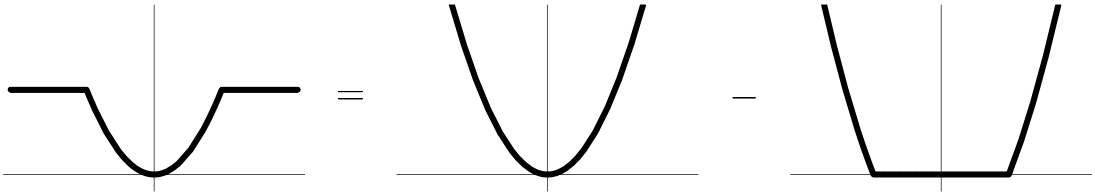


Figure 6.7: This figure illustrates the d.c. decomposition of the robust smoothness prior $S(\mathbf{x})$, equation (6.12), shown on the left side. As can be seen there, the prior is decomposed into the difference of two convex functions which is formally shown in equation (6.13). Note that our d.c. framework is able to handle this decomposition although the function on the right side is non-differentiable at c.

$$f_c(t) \;:=\; g_c(t) - h_c(t) \qquad \text{with} \qquad \begin{aligned} g_c(t) &:= \tfrac{1}{c^2}t^2 \\ h_c(t) &:= \tfrac{1}{c^2}\max\{0, t^2 - c^2\} \end{aligned} \tag{6.13}$$

The robust smoothness prior penalizes deviations in the grayvalues and, hence, implicitly acts on $\mathbf{x}$. Alternatively, $\mathbf{x}$ is explicitly included if we use the assignment matrix $\mathbf{X}$, equation (6.1), that finally contains a binary image in each column $\mathbf{x}_{*,i}$. Hence, we can apply the smoothness prior $\|\ \mathbf{L}\ \mathbf{x}\ \|$ from the previous chapter independently to the columns of $\mathbf{X}$,

$$S_{II}(\mathbf{x}) := \sum_{j=1}^{l} \|\ \mathbf{L}\ \mathbf{x}_{*,j}\|^2 = \sum_{j=1}^{l} \mathbf{x}_{*,j}^\top\ \mathbf{L}^\top\mathbf{L}\ \ \mathbf{x}_{*,j} =: \|\tilde{\mathbf{L}}\ \mathbf{x}\|^2. \tag{6.14}$$

As all terms in $S_{II}(\mathbf{x})$ are convex, there is no need for a d.c. decomposition and we can simply add the whole term to $g(\mathbf{x})$, unlike $S_I(\mathbf{x})$.

6.2.4 Optimization

With respect to multiclass reconstructions, we minimize the following energy functionals,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^{nl}} E(\mathbf{x};\mu) := \frac{1}{2}\Big[\|\mathbf{A}\;\mathbf{G}\;\mathbf{x}-\mathbf{b}\|^2 + \alpha\; S(\mathbf{x}) + 2\,\lambda\,(\langle\mathbf{x},\mathbf{x}\rangle - n)\Big] \\ + \mu\sum_{i=1}^{n}\sum_{j=1}^{l}(g_j - \langle\mathbf{g},\mathbf{x}_i\rangle)^2\; x_{i,j} \qquad (6.15) \\ \text{subject to}\quad \mathbf{E}\,\mathbf{x} = \mathbf{1_n}, \quad \mathbf{x}\geq\mathbf{0}, \end{aligned}$$

where $\mathbf{E} := \operatorname{diag}(\mathbf{1_l}^\top)$ is a n-blockdiagonal matrix and $S \in \{S_I, S_{II}\}$, see equations (6.12) and (6.14). We also write

$$E_I(\mathbf{x};\mu) \quad \text{and} \quad E_{II}(\mathbf{x};\mu) \qquad (6.16)$$

if we want to distinguish between the functionals $E(\mathbf{x};\mu)$ supplemented with either S_I or S_{II}.

As d.c. decomposability is required for S_I, we generally write

$$S(\mathbf{x}) =: S_g(\mathbf{x}) - S_h(\mathbf{x}) \qquad (6.17)$$

which in terms of the smoothness priors from section 6.2.3 reads

$$\begin{aligned} S_I(\mathbf{x}) \quad &\to \quad S_g(\mathbf{x}) := \frac{1}{c^2}\|\,\mathbf{L}\;\mathbf{G}\;\mathbf{x}\|^2 && (6.18) \\ & \qquad S_h(\mathbf{x}) := \frac{1}{c^2}\sum_{\langle i,j\rangle}\max\{0, (\langle\mathbf{g},\mathbf{x}_i\rangle - \langle\mathbf{g},\mathbf{x}_j\rangle)^2 - c^2\} && (6.19) \\ S_{II}(\mathbf{x}) \quad &\to \quad S_g(\mathbf{x}) := \sum_{j=1}^{l}\|\,\mathbf{L}\;\mathbf{x}_{*,j}\|^2 && (6.20) \\ & \qquad S_h(\mathbf{x}) := 0. && (6.21) \end{aligned}$$

The term $\lambda\;(\langle\mathbf{x},\mathbf{x}\rangle - n)$ in functional, equation (6.15), is redundant with respect to the constraints. It is, however, important for the optimization as we will explain later.

By virtue of the equations (6.10) and (6.17), functional $E(\mathbf{x};\mu)$, equation (6.15), is d.c.

decomposable according to

$$E(\mathbf{x};\mu) = g(\mathbf{x}) - h(\mathbf{x};\mu) \quad \text{with}$$

$$g(\mathbf{x}) := \frac{1}{2}\Big[\|\mathbf{A}\ \mathbf{G}\ \mathbf{x} - \mathbf{b}\|^2 + \alpha\ S_g(\mathbf{x}) + 2\ \lambda\ (\langle \mathbf{x}, \mathbf{x}\rangle - n)\Big] \tag{6.22}$$

$$h(\mathbf{x};\mu) := \frac{\alpha}{2} S_h(\mathbf{x}) - \mu \sum_{i=1}^{n}\sum_{j=1}^{l}(g_j - \langle \mathbf{g}, \mathbf{x}_i\rangle)^2\ x_{i,j} \tag{6.23}$$

and, thus, fits the d.c. framework in section 5.4.

With respect to the (y-step), we compute the partial derivative at $x_{s,t}$ of the convex-concave regularizer in equation (6.23) by

$$\begin{aligned}
&\frac{\partial}{\partial x_{s,t}}\left[-\mu\sum_{i=1}^{n}\sum_{j=1}^{l}(g_j - \mathbf{g}^\top \mathbf{x}_i)^2\ x_{i,j}\right] = \frac{\partial}{\partial x_{s,t}}\left[-\mu\sum_{j=1}^{l}(g_j - \mathbf{g}^\top \mathbf{x}_s)^2\ x_{s,j}\right] = \\
&\frac{\partial}{\partial x_{s,t}}\left[-\mu\left((g_t - \mathbf{g}^\top \mathbf{x}_s)^2\ x_{s,t} + \sum_{j=1, j\neq t}^{l}(g_j - \mathbf{g}^\top \mathbf{x}_s)^2\ x_{s,j}\right)\right] = \\
&-\mu\left(2(g_t - \mathbf{g}^\top \mathbf{x}_s)(-g_t)\ x_{s,t} + (g_t - \mathbf{g}^\top \mathbf{x}_s)^2 + \sum_{j=1, j\neq t}^{l} 2(g_j - \mathbf{g}^\top \mathbf{x}_s)(-g_t)\ x_{s,j}\right) = \\
&-\mu\left((g_t - \mathbf{g}^\top \mathbf{x}_s)^2 + \sum_{j=1}^{l} 2(g_j - \mathbf{g}^\top \mathbf{x}_s)(-g_t)\ x_{s,j}\right) = \\
&-\mu\left((g_t - \mathbf{g}^\top \mathbf{x}_s)^2 + 2(-g_t)\Big(\sum_{j=1}^{l} g_j\ x_{s,j} - \mathbf{g}^\top \mathbf{x}_s(\sum_{j=1}^{l}\ x_{s,j})\Big)\right) = \\
&-\mu\left((g_t - \mathbf{g}^\top \mathbf{x}_s)^2 + 2(-g_t)\Big(\mathbf{g}^\top \mathbf{x}_s - \mathbf{g}^\top \mathbf{x}_s\Big)\right) = -\mu\ \Big(g_t - \mathbf{g}^\top \mathbf{x}_s\Big)^2
\end{aligned} \tag{6.24}$$

and, thus, obtain the subgradient of $h(\mathbf{x};\mu)$ with

$$\partial h(\mathbf{x};\mu) = \frac{\alpha}{2}\partial S_h(\mathbf{x}) - \mu\Big(g_j - \mathbf{g}^\top \mathbf{x}_i\Big)^2_{1\leq i\leq n,\ 1\leq j\leq l}. \tag{6.25}$$

In reference to algorithm 7, section 5.3, the (y-step), equation (6.26), and (x-step), equation (6.27), of the multiclass d.c. approach read

$$\begin{aligned}\mathbf{y}^k &:= \partial h(\mathbf{x}^k; \mu) && (6.26)\\ \mathbf{x}^{k+1} &:= \underset{\mathbf{x}\in\Omega_1}{\text{argmin}} \Big\{ g(\mathbf{x}) - \langle \mathbf{y}^k, \mathbf{x} \rangle \Big\} && (6.27)\\ &= \underset{\mathbf{x}\in\Omega_1}{\text{argmin}} \Big\{ \frac{1}{2} \Big[\|\mathbf{A}\ \mathbf{G}\ \mathbf{x} - \mathbf{b}\|^2 + \alpha\ S_g(\mathbf{x}) + 2\ \lambda\ (\langle \mathbf{x}, \mathbf{x} \rangle - n) \Big] - \langle \mathbf{y}^k, \mathbf{x} \rangle \Big\}.\end{aligned}$$

The (x-step), equation (6.27), involves the minimization of a quadratic optimization problem. As previously in the binary case, section 5.5 and [123], we can use the spectral projected gradient (SPG) method, appendix 15, which requires only the evaluation of the function, the gradient, and the projection onto the convex set. The latter is trivial in the binary case because of the simple box constraints but is more intricated in the multiclass case due to the shape of Ω_1. Therefore, appropriate projection methods have been proposed in [96] and [131] which we both implemented and tested. Summarizing our experience, the convergence behavior of the SPG algorithm was comparable for both methods but was too slow for our needs in total. According to [12] and [102] the profit of projection methods in optimization strongly depends on the complexity of the underlying projection method. Thus, we compared the time spent in the projection methods against the remaining part of SPG and found that about 50% was needed only for the projections. For this reason we decided not to pursue SPG any further for the multiclass approach and to use a different optimization strategy instead.

Reconsidering the constraints introduced by the convex set Ω_1, we observe that they partly consist of equality and inequality constraints,

$$\mathbf{E}\ \mathbf{x} = \mathbf{1}_n \qquad \text{and} \qquad \mathbf{x} \geq \mathbf{0}. \tag{6.28}$$

By introducing the augmented Lagrangian function $L_{c^j}(\mathbf{x};\ \mathbf{y}^k, \lambda^j)$, equation (6.29), we simplify the constraints above and, hence, the optimization problem involved in the (x-step), equation (6.27),

$$\begin{aligned}L_{c^j}(\mathbf{x};\ \mathbf{y}^k, \xi^j) := \min_{\mathbf{x}\geq\mathbf{0}} \frac{1}{2} \Big[\|\mathbf{A}\ \mathbf{G}\ \mathbf{x} - \mathbf{b}\|^2 + \alpha\ S_g(\mathbf{x}) + 2\ \lambda\ (\langle \mathbf{x}, \mathbf{x} \rangle - n) \Big] - \langle \mathbf{y}^k, \mathbf{x} \rangle \\ +(\xi^j)^\top\ (\ \mathbf{E}\ \mathbf{x} - \mathbf{1}_n) + \frac{c^j}{2} \|\mathbf{E}\ \mathbf{x} - \mathbf{1}_n\|^2,\end{aligned} \tag{6.29}$$

where ξ^j denotes the vector of Lagrange multipliers. Increasing the penalty parameter c^j enforces the equality constraints and updates ξ^j according to

$$\xi^{j+1} := \xi^j + c^j\ (\mathbf{E}\ \mathbf{x} - \mathbf{1}_n), \tag{6.30}$$

as recommended in [12]. Notice that $L_{c^j}(\mathbf{x};\ \mathbf{y}^k, \xi^j)$ is a quadratic minimization problem over the cone of non-negative vectors, i.e. $\mathbf{x} \in \mathbb{R}^{ln}_{\geq 0}$,

$$L_{c^j}(\mathbf{x};\ \mathbf{y}^k, \xi^j) := \min_{\mathbf{x} \geq \mathbf{0}} \frac{1}{2} \mathbf{x}^\top \mathbf{Q}\, \mathbf{x} - \mathbf{q}^\top \mathbf{x} + c, \tag{6.31}$$

with

$$\mathbf{Q}_I := \mathbf{G}^\top \mathbf{A}^\top \mathbf{A}\ \mathbf{G} + \alpha\ \mathbf{G}^\top \mathbf{L}^\top \mathbf{L}\ \mathbf{G} + 2\,\lambda\,\mathbf{I} + c^j\ \mathbf{E}^\top \mathbf{E} \tag{6.32}$$

$$\mathbf{Q}_{II} := \mathbf{G}^\top \mathbf{A}^\top \mathbf{A}\ \mathbf{G} + \alpha\ \tilde{\mathbf{L}}^\top \tilde{\mathbf{L}} + 2\,\lambda\,\mathbf{I} + c^j\ \mathbf{E}^\top \mathbf{E} \tag{6.33}$$

$$\mathbf{q} := \mathbf{b}^\top \mathbf{A}\ \mathbf{G} + \langle \mathbf{y}^k, \mathbf{x} \rangle - (\xi^j)^\top \mathbf{E} + c^j\ \mathbf{1}_n^\top \mathbf{E} \tag{6.34}$$

$$c := \frac{1}{2} \mathbf{b}^\top \mathbf{b} - \lambda\, n - (\xi^j)^\top \mathbf{1}_n + \frac{c^j}{2} \mathbf{1}_n^\top \mathbf{1}_n, \tag{6.35}$$

where the choice of smoothness prior, S_I or S_{II}, affects matrix $\mathbf{Q}$ and, thus, leads to $\mathbf{Q}_I$ and $\mathbf{Q}_{II}$ respectively. For the sake of simplicity, we sometimes write $\mathbf{Q}$ and distinguish between $\mathbf{Q}_I$ and $\mathbf{Q}_{II}$ only if necessary. Also, we do not explicitly denote that $\mathbf{Q}$ depends on the parameters α, λ, and c^j.

Concerning the minimization of the quadratic optimization problem, equation (6.31), we use the MPRGP (modified proportioning with reduced gradient projections) optimizer, see algorithm 12, which has been proposed by Dostál and Schöberl in [49]. MPRGP is explicitly adviced and applied by the authors for quadratic auxiliary problems in augmented Lagrangian type algorithms [48, 47] which are restricted to the non-negative cone. At this, MPRGP requires matrix $\mathbf{Q}$ to be positive definite. This is exactly were parameter λ in functional, equation (6.15), enters the stage.

Assuming for the moment that $\lambda = 0$, in case of $\mathbf{Q}_{II}$ the smoothness prior $\mathbf{x}^\top \tilde{\mathbf{L}}^\top \tilde{\mathbf{L}}\ \mathbf{x}$ is 0 iff all columns of $\mathbf{X}$ are constant, e.g. $\mathbf{X} = (\alpha_1 \mathbf{1}_n, ..., \alpha_l \mathbf{1}_n)$. Additionally, $\mathbf{x}^\top \mathbf{E}^\top \mathbf{E}\ \mathbf{x} = 0$ holds iff $\sum_{j=1}^{l} \alpha_j = 0$. Equation $\mathbf{x}^\top \mathbf{G}^\top \mathbf{A}^\top \mathbf{A}\ \mathbf{G}\ \mathbf{x}$ is at least zero if $\mathbf{G}\mathbf{x} = \mathbf{0}$ and, thus, we obtain $\sum_{j=1}^{l} g_j \alpha_j = 0$. Summarizing, we have two equations and at least two unknowns. Consequently, if $\lambda = 0$ the smallest eigenvalue of matrix $\mathbf{Q}_{II}$ is 0 and we have positive semidefiniteness. However, matrix $\mathbf{Q}_{II}$ becomes positive definite for $\lambda > 0$ where λ equals the smallest eigenvalue. Similar in case of $\mathbf{Q}_I$, $\mathbf{x}^\top \mathbf{G}^\top \mathbf{A}^\top \mathbf{A}\ \mathbf{G}\,\mathbf{x}$ and $\mathbf{x}^\top \mathbf{G}^\top \mathbf{L}^\top \mathbf{L}\ \mathbf{G}\,\mathbf{x}$ are zero at least if $\mathbf{G}\mathbf{x} = \mathbf{0}$. This leads to $\mathbf{g}^\top \mathbf{x}_i = 0$, for all i, and from $\mathbf{x}^\top \mathbf{E}^\top \mathbf{E}\ \mathbf{x} = 0$ we obtain $\mathbf{1}_l^\top \mathbf{x}_i = 0$, for all i. By this, we have $2 \cdot n$ equations and $l \cdot n$, $l \geq 2$, unknowns. Thus, matrix $\mathbf{Q}_I$ is positive semidefinite but becomes positive definite for $\lambda > 0$ and λ is the smallest eigenvalue. Hence, we conclude that positive definiteness holds for our quadratic functionals, equation (6.31), iff $\lambda > 0$.

Algorithm 12 MPRGP (Modified Proportioning w. Reduced Gradient Projections) [49]

```
Require: Quadratic optimization problem, $\min_{\mathbf{x}} \frac{1}{2}\,\mathbf{x}^\top \mathbf{Q}\,\mathbf{x} - \mathbf{q}^\top \mathbf{x}$, subject to $\mathbf{x} \geq \mathbf{0}$.
Require: $\mathbf{Q}$ is symmetric positive definite
Require: $\Gamma > 0$, $\bar{\alpha} \in (0, \|\mathbf{Q}\|^{-1}]$, and $\epsilon > 0$.
Require: $\mathbf{x}^0 \geq \mathbf{0}$                                {Initial $\mathbf{x}$.}
{See equations (6.36) et seq. for the definitions of $\varphi$, $\tilde{\varphi}$, $\beta$, $\nu$, and $P_{\geq 0}\cdot$.}
$\mathbf{r} := \mathbf{Q}\,\mathbf{x}^0 - \mathbf{q}$                    {Residual of the gradient.}
$\mathbf{p} := \varphi(\mathbf{x}^0)$                                    {Direction}
$k := 0$
while $\|\nu(\mathbf{x}^k)\| > \epsilon$ do
  if $\|\beta(\mathbf{x}^k)\|^2 \leq \Gamma\,\tilde{\varphi}(\mathbf{x}^k)\,\varphi(\mathbf{x}^k)$ then
    $\alpha_{cg} := \mathbf{r}^\top \mathbf{p} / \mathbf{p}^\top \mathbf{p}$           {Compute conjugate gradient step size.}
    $\alpha_{\max} := \max_{\alpha}\{\mathbf{x}^k - \alpha\,\mathbf{p} \geq 0\}$         {Compute maximal step size.}
    if $\alpha_{cg} < \alpha_{\max}$ then

      {Conjugate gradient step.}
      $\mathbf{x}^{k+1} := \mathbf{x}^k - \alpha_{cg}\,\mathbf{p}$                 {Update $\mathbf{x}$.}
      $\mathbf{r} := \mathbf{Q}\,\mathbf{x}^{k+1} - \mathbf{q}$                   {Update residual of the gradient.}
      $\gamma := \varphi(\mathbf{x}^{k+1})^\top \mathbf{Q}\,\mathbf{p} / \mathbf{p}^\top \mathbf{p}$
      $\mathbf{p} := \varphi(\mathbf{x}^{k+1}) - \gamma\,\mathbf{p}$              {Update direction.}

    else

      {Expansion step.}
      $\mathbf{x}^{k+\frac{1}{2}} := \mathbf{x}^k - \alpha_{\max}\,\mathbf{p}$
      $\mathbf{x}^{k+1} := P_{\geq 0}\big(\mathbf{x}^{k+\frac{1}{2}} - \bar{\alpha}\,\varphi(\mathbf{x}^{k+1})\big)$    {Update $\mathbf{x}$.}
      $\mathbf{r} := \mathbf{Q}\,\mathbf{x}^{k+1} - \mathbf{q}$                   {Update residual of the gradient.}
      $\mathbf{p} := \varphi(\mathbf{x}^{k+1})$                                 {Update direction.}

    end if
  else

    {Proportioning step.}
    $\mathbf{d} := \beta(\mathbf{x}^k)$
    $\alpha := \mathbf{r}^\top \mathbf{d} / \mathbf{d}^\top \mathbf{Q}\,\mathbf{d}$
    $\mathbf{x}^{k+1} := \mathbf{x}^k - \alpha\,\mathbf{d}$                       {Update $\mathbf{x}$.}
    $\mathbf{r} := \mathbf{Q}\,\mathbf{x}^{k+1} - \mathbf{q}$                     {Update residual of the gradient.}
    $\mathbf{p} := \varphi(\mathbf{x}^{k+1})$                                   {Update direction.}

  end if
  $k := k + 1$
end while
return $\mathbf{x} := \mathbf{x}^k$                                      {Return solution.}
```

Let $g(\mathbf{x}) := \mathbf{Q}\,\mathbf{x} - \mathbf{q}$ be the gradient of equation (6.31). The MPRGP optimizer, algorithm 12, involves the following definitions,

$$\varphi_i(\mathbf{x}) := \begin{cases} g_i(\mathbf{x}) & \text{if} \quad x_i \neq 0 \\ 0 & \text{if} \quad x_i = 0 \end{cases} \tag{6.36}$$

$$\tilde{\varphi}_i(\mathbf{x}) := \min\{x_i/\bar{\alpha}, \varphi_i(\mathbf{x})\} \tag{6.37}$$

$$\beta_i(\mathbf{x}) := \begin{cases} 0 & \text{if} \quad x_i \neq 0 \\ \min\{g_i(\mathbf{x}), 0\} & \text{if} \quad x_i = 0 \end{cases} \tag{6.38}$$

$$\nu(\mathbf{x}) := \varphi(\mathbf{x}) + \beta(\mathbf{x}) \tag{6.39}$$

$$P_{\geq 0}(\mathbf{x}) := \begin{cases} x_i & \text{if} \quad x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{6.40}$$

where φ_i is the free gradient, $\tilde{\varphi}$ the reduced free gradient, β the chopped gradient, ν the projected gradient, and $P_{\geq 0}$ the projection on the non-negative cone. For further details we kindly refer the more interested reader to [49, 48, 47].

The overall multiclass d.c. approach is summarized in algorithm 13 and leads to the following interpretation. As μ is initially set to 0, the concave prior does not interfere with the first iterations of the inner loops and the algorithm yields the convex solution of the problem. Afterwards, μ is incremented in the outer loop, $\mu > 0$, and the subsequent y-step computes costs for all $x_{i,j}$ depending on the grayvalue $\mathbf{g}^\top \mathbf{x}_i^k$ obtained from the previous solution,

$$\mu \left(g_j - \mathbf{g}^\top \mathbf{x}_i^k\right)^2, \tag{6.41}$$

compare equations (6.25) and (6.26) and see also figure 6.8. During the next x-step, the optimizer favors components $x_{i,j}$ of $\mathbf{x}$ if their label g_j matched the value of the pixel in the previous solution, and penalizes them otherwise.

6.3 Relation to the Binary Case

In section 5.5.3 we briefly presented a quadratic optimizer for the binary case, see Schüle [124] for a through introduction. Originally, this suggested a quadratic formulation for multiple classes which, however, failed as described in section 6.2.2. From this background, the question arises how the behavior of both, i.e. the original binary optimizer and the MC-DCA applied to binary problems, differs.

Algorithm 13 Multiclass DC Algorithm (MC-DCA)

```
Require: $\mathbf{x}^0 \in \Omega_1$                      {for example $\mathbf{x}^0 := \frac{1}{l}(1, ..., 1)^\top$.}
Require: $\Delta\mu > 0$
Require: $\epsilon_{DC}, \epsilon_L > 0$
Require: $c_0 \geq 0$
Require: $\Delta c > 0$
$i := 0$                                                  {Outer loop counter.}
$j := 0$                                                  {Penalty counter.}
$\mu^0 := 0$
repeat
    $k := 0$                                              {Inner loop counter.}
    repeat
        $\mathbf{y}^k := \nabla h(\mathbf{x}^k; \mu^i)$    {(y-step); equation (6.26).}

        $l := 0$                                          {BEGIN (x-step); equation (6.27).}
        $\hat{\mathbf{x}}^0 := \mathbf{x}^k$
        $\lambda^0 := (0, ..., 0)^\top$
        repeat
            $\lambda^{l+1} := \lambda^l + c^j\ (\mathbf{E}\,\hat{\mathbf{x}}^l - \mathbf{1}_n)$    {Update Lagrange multiplier $\lambda$.}

            {MPRGP optimizer, algorithm 12, with starting point $\hat{\mathbf{x}}^k$.}
            $\hat{\mathbf{x}}^{l+1} := \min_{\mathbf{x} \geq 0}\{L_{c^j}(\mathbf{x};\ \mathbf{y}^k, \lambda^{l+1})\}$

            $c^{j+1} := c^j + \Delta c$                   {Update penalty parameter $c$.}
            $j := j + 1$
            $l := l + 1$
        until $\|\mathbf{E}\,\hat{\mathbf{x}}^l - \mathbf{1}_n\|_2 < \epsilon_L$
        $\mathbf{x}^{k+1} := \hat{\mathbf{x}}^l$           {END: (x-step); equation (6.27).}

        $k := k + 1$
    until $\|\mathbf{x}^k - \mathbf{x}^{k-1}\|_2 < \epsilon_{DC}$
    $\mu^{i+1} := \mu^i + \Delta\mu$                      {Update $\mu$.}
    $i := i + 1$
    $\mathbf{x}^0 := \mathbf{x}^k$
until $\nexists\, \mathbf{x}_j^k \in [\epsilon_{DC}, 1 - \epsilon_{DC}],\ \forall j$
return $\mathbf{x} := \mathbf{x}^k$                        {Return solution.}
```

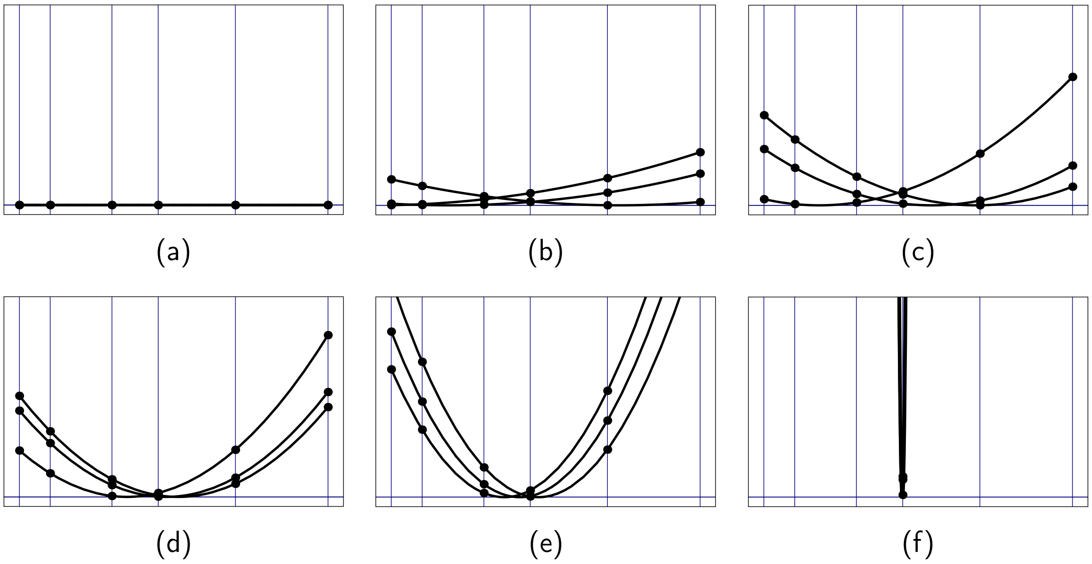


Figure 6.8: Concerning a single pixel i the plots illustrate the costs assigned by the cost function $c(\cdot)$, equation (6.8), for increasing values of μ and the resulting behavior of our approach. The x-axis (bottom) ranges from the left to the right over the interval of possible grayvalues $[g_1, g_l]$ and a vertical line is drawn at each position $g_j \in \mathcal{L}$. Consider plot (a) which shows the first iteration of the algorithm. All costs are 0 since $\mu = 0$ and the algorithm computes the convex solution leading pixel i to attain some grayvalue in $[g_1, g_l]$. In the next inner pass of the algorithm, plot (b), costs (depicted by black dots) are assigned for each label g_j by the y-step depending on the grayvalue of pixel i in the previous solution. However, pixel i is able to change its value even to distant grayvalues since only moderate costs have been assigned. During the following passes the costs are increased by μ and thus pixel i becomes more localized, plots (c)-(e). At some point, it is preferable in the (x-step) to stay at the current grayvalue g_j since the costs for moving, even to a close grayvalue g_{j-1} or g_{j+1}, become too expensive, plot (f).

We apply the multiclass functional in equation (6.15) supplemented with smoothness prior $S_{II}(\mathbf{x})$, equation (6.14), to a binary problem. At this, we exploit that $\mathbf{x}_{*,1} = 1 - \mathbf{x}_{*,2}$ and, thus, we drop the variables $\mathbf{x}_{*,1}$,

$$\begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{6.42}$$

and

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} \\ & \vdots & \\ x_{n,1} & x_{n,2} \end{pmatrix} = \begin{pmatrix} 1 - x_{1,2} & x_{1,2} \\ & \vdots & \\ 1 - x_{n,2} & x_{n,2} \end{pmatrix}. \tag{6.43}$$

Note, that vector $\mathbf{x}_{*,2}$ here is equal to the solution vector $\mathbf{x}$ in the binary case, see previous chapters.

Using equation (6.42) and (6.43), yields

$$\mathbf{G}\,\mathbf{x} = \mathbf{x}_{*,2} \tag{6.44}$$

and, consequently, the data and smoothness term of our multiclass functional reduce to

$$\|\mathbf{A}\,\mathbf{G}\,\mathbf{x} - \mathbf{b}\|^2 = \|\mathbf{A}\,\mathbf{x}_{*,2} - \mathbf{b}\|^2 \quad \text{and} \quad \|\mathbf{L}\,\mathbf{G}\,\mathbf{x}\|^2 = \|\mathbf{L}\,\mathbf{x}_{*,2}\|^2 \tag{6.45}$$

which are exactly the same for the binary optimizer, section 5.5.3. Concerning our concave regularizer, section 6.2.2, we derive

$$\begin{aligned} \sum_{i=1}^{n}\sum_{j=1}^{l} c(g_j, \mathbf{g}^\top \mathbf{x}_i)\, x_{i,j} &= \sum_{i=1}^{n}\sum_{j=1}^{l} (g_j - \mathbf{g}^\top \mathbf{x}_i)^2\, x_{i,j} \\ &= \sum_{i=1}^{n} x_{i,2}^2\,(1 - x_{i,2}) + (1 - x_{i,2})^2\, x_{i,2} \\ &= \sum_{i=1}^{n} (1 - x_{i,2})(x_{i,2}^2 + (1 - x_{i,2})\, x_{i,2}) \\ &= \sum_{i=1}^{n} (1 - x_{i,2})\, x_{i,2} \\ &= \langle \mathbf{x}_{*,2}, \mathbf{1} - \mathbf{x}_{*,2} \rangle \end{aligned} \tag{6.46}$$

and, hence, obtain the concave regularizer used for the binary case. Only the term $\lambda(\langle \mathbf{x}, \mathbf{x} \rangle - n)$ remains which was introduced in order to ensure the positive definiteness of matrix $\mathbf{Q}$, equation (6.31). However, concerning equation (6.45) matrix $\mathbf{Q}$ is only positive semidefinite for trivial problems, i.e. constant images. Thus, we can confidently choose $\lambda = 0$ and obtain

$$\frac{1}{2}\Big[\|\mathbf{A}\ \mathbf{G}\ \mathbf{x} - \mathbf{b}\|^2 + \alpha\|\mathbf{L}\ \mathbf{G}\ \mathbf{x}\|^2\Big] + \mu \sum_{i=1}^{n}\sum_{j=1}^{l} c(g_j, \langle \mathbf{g}, \mathbf{x}_i\rangle)\ x_{i,j} = \\ \frac{1}{2}\Big[\|\mathbf{A}\ \mathbf{x}_{*,2} - \mathbf{b}\|^2 + \alpha\|\mathbf{L}\ \mathbf{x}_{*,2}\|^2\Big] + \mu\ \langle \mathbf{x}_{*,2}, \mathbf{1} - \mathbf{x}_{*,2}\rangle. \qquad (6.47)$$

This result ensures that both functionals are the same in the binary case and, thus, the multiclass optimizer is indeed a generalization of the binary quadratic optimizer from section 5.5.3.

6.4 Global Optimality Conditions

Concerning the optimization of quadratic problems with binary constraints,

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\ \mathbf{x} - \mathbf{q}^\top \mathbf{x}, \quad \mathbf{x} \in \{0,1\}^n, \qquad (6.48)$$

global optimality conditions, theorem 11, have been proposed in [10].

Theorem 11 ([10]). *Assume $\mathbf{x} \in \{0,1\}^n$ then referring to the quadratic minimization problem in equation (6.48) the following inequalities, to be read elementwise, provide sufficient (SC) and necessary (NC) conditions for the global optimality of $\mathbf{x}$.*

$$\textit{(SC):} \qquad \frac{1}{4}\lambda_{\min}(\mathbf{Q})\ \mathbf{1} \quad \geq \quad (\operatorname{diag}(\mathbf{x}) - \frac{1}{2}\mathbf{I})(\mathbf{Q}\ \mathbf{x} - \mathbf{q}) \qquad (6.49)$$

$$\textit{(NC):} \qquad \frac{1}{4}\operatorname{diag}(\mathbf{Q})\ \mathbf{1} \quad \geq \quad (\operatorname{diag}(\mathbf{x}) - \frac{1}{2}\mathbf{I})(\mathbf{Q}\ \mathbf{x} - \mathbf{q}) \qquad (6.50)$$

While these conditions can straightforwardly be checked in case of the binary quadratic optimizer, section 5.5.3, it is different in the multiclass case due to the restriction $\mathbf{x} \in \Omega^n_{\{0,1\}}$. For this reason, we follow the ideas of [10] and elaborate analogous conditions for the multiclass problem.

With respect to (SC) and our multiclass problem, we consider the quadratic minimization problem,

$$\min_{\mathbf{x}} E(\mathbf{x}) \quad \text{with} \quad E(\mathbf{x}) := \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\ \mathbf{x} - \mathbf{q}^\top \mathbf{x}, \quad \mathbf{x} \in \Omega^n_{\{0,1\}}, \qquad (6.51)$$

and associated with $E(\mathbf{x})$ the Lagrangian function $L(\mathbf{x}; c)$,

$$L(\mathbf{x}; c) := E(\mathbf{x}) + c\ \|\mathbf{E}\ \mathbf{x} - \mathbf{1}\|^2 \quad \text{s.t.} \quad \mathbf{x} \in \{0,1\}^{nl}, \qquad (6.52)$$

which yields a quadratic optimization problem with binary constraints again. Observe that $\|\mathbf{E}\ \mathbf{x} - \mathbf{1}\|^2$ penalizes solutions $\mathbf{x} \notin \Omega^n_{\{0,1\}}$ for $c > 0$. Thus, the key is to select a fixed $\hat{c}$ deteriorating all $\mathbf{x} \notin \Omega^n_{\{0,1\}}$ such that the global optimum $\mathbf{x}^* \in \{0,1\}^{nl}$ of $\min_{\mathbf{x}} L(\mathbf{x};\hat{c})$ is guaranteed to be in $\Omega^n_{\{0,1\}}$.

Lemma 12. *Let $\hat{\mathbf{x}} \in \Omega^n_{\{0,1\}}$ and $\mathbf{x}^*$ be the global minimum of $L(\mathbf{x};\hat{c})$, $\mathbf{x} \in \{0,1\}^{nl}$, then $\mathbf{x}^* \in \Omega^n_{\{0,1\}}$ for all $\hat{c} > E(\hat{\mathbf{x}})$.*

Proof. Given $\mathbf{x} \in \{0,1\}^{nl}$ and let $\mathbf{X}$ be the corresponding assignment matrix, equation (6.1). Since $\mathbf{x}$ is an arbitrary binary vector, $\mathbf{X}$ might not contain exactly a single 1 entry per row and $\|\mathbf{E}\ \mathbf{x} - \mathbf{1}\|^2$ measures this disagreement. For any binary vector $\mathbf{x} \notin \Omega^n_{\{0,1\}}$ the following lower bound holds

$$1 \leq \|\mathbf{E}\,\mathbf{x} - \mathbf{1}\|^2 \tag{6.53}$$

since matrix $\mathbf{X}$ must contain at least one row with at least two 1 entries or at least one row with only zero entries.

Consider any $\mathbf{x} \notin \Omega^n_{\{0,1\}}$, by means of inequality (6.53) and $\hat{c} > E(\hat{\mathbf{x}})$ we have

$$L(\mathbf{x};\hat{c}) = E(\mathbf{x}) + \hat{c}\ \|\mathbf{E}\,\mathbf{x} - \mathbf{1}\|^2 \geq E(\mathbf{x}) + \hat{c} > E(\hat{\mathbf{x}}) = L(\hat{\mathbf{x}};\hat{c}) \geq L(\mathbf{x}^*;\hat{c})$$

and $\mathbf{x}^* \in \Omega^n_{\{0,1\}}$ follows. □

Note that for practical purposes we get a reasonable small lower bound without any additional effort simply by taking $\hat{c} > E(\mathbf{x})$ where $\mathbf{x}$ is the solution to be checked for global optimality. If $\hat{c}$ is chosen with respect to lemma 12 we obtain sufficient conditions for the multiclass problem by applying (SC) of theorem 11 to $L(\mathbf{x};\hat{c})$.

Unfortunately, this does not work out analogously for the necessary conditions (NC) in theorem 11 since they check if the inversion of a single component in $\mathbf{x}$ leads to a lower minimum. If so, $\mathbf{x}$ obviously cannot be globally optimal while otherwise it might be. As a consequence, (NC) applied in the same way as (SC) will always be true because flipping a single component of $\mathbf{x} \in \Omega^n_{\{0,1\}}$ must lead to $\mathbf{x} \notin \Omega^n_{\{0,1\}}$ which will never attain a lower minimum due to lemma 12. Therefore, we follow the lead of Beck and Teboulle [10] for (NC) and derive necessary conditions, theorem 13, which are appropriate for the multiclass problem.

Theorem 13. *Given the quadratic minimization problem,*

$$\min_{\mathbf{x}} P(\mathbf{x}) := \frac{1}{2}\,\mathbf{x}^\top \mathbf{Q}\,\mathbf{x} - \mathbf{q}^\top \mathbf{x}, \quad \mathbf{x} \in \Omega^n_{\{0,1\}}, \tag{6.54}$$

with real and symmetric matrix $\mathbf{Q}$*. If* $\mathbf{x} \in \Omega^n_{\{0,1\}}$ *is globally optimal for problem (6.54), then*

$$\frac{1}{2}\, Q_{j,j} - \sum_{k=a(j)}^{b(j)} \Big(Q_{j,k} - \frac{1}{2}\, Q_{k,k} \Big)\, x_k \quad \geq \quad \Big(\sum_{k=a(j)}^{b(j)} x_k\, \mathbf{e}_k - \mathbf{e}_j \Big)^\top \Big(\mathbf{Q}\, \mathbf{x} - \mathbf{q} \Big)$$

with

$$a(j) := \lfloor \frac{j-1}{l} \rfloor \cdot l + 1 \quad \textit{and} \quad b(j) := \lceil \frac{j}{l} \rceil \cdot l$$

holds for all $j \in \{1, ..., l \cdot n\}$*.*

Proof. We have that if $\mathbf{x} \in \Omega^n_{\{0,1\}}$ is globally minimal then

$$P(\mathbf{x}) \leq P(\mathbf{z}), \quad \forall\, \mathbf{z} \in \Omega^n_{\{0,1\}}. \tag{6.55}$$

With respect to $\mathbf{x}$ we consider

$$\mathbf{z}_j := (\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\, \mathbf{e}_k + \mathbf{x}) \in \Omega^n_{\{0,1\}} \tag{6.56}$$

which changes the labeling of a single pixel in $\mathbf{x}$ for each $j \in \{1, ..., l \cdot n\}$, e.g. $j = 1$ leads to $\mathbf{z}_1 = (1\ 0\ 0\ ...\ 0\ \mathbf{x}_2^\top\ \mathbf{x}_3^\top\ ...\ \mathbf{x}_n^\top)^\top$, $j = 2$ to $\mathbf{z}_2 = (0\ 1\ 0\ ...\ 0\ \mathbf{x}_2^\top\ \mathbf{x}_3^\top\ ...\ \mathbf{x}_n^\top)^\top$, ... , $j = n \cdot l$ to $\mathbf{z}_{n \cdot l} = (\mathbf{x}_1^\top\ \mathbf{x}_2^\top\ ...\ \mathbf{x}_{n-1}^\top 0\ ...\ 0\ 1)^\top$. Consequently, the following inequalities provide necessary conditions for the global optimality of $\mathbf{x} \in \Omega^n_{\{0,1\}}$,

$$\begin{aligned} P(\mathbf{x}) \ \leq \ P(\mathbf{z}_j) \ &= \frac{1}{2}\, (\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\, \mathbf{e}_k + \mathbf{x})^\top \mathbf{Q}\, (\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\, \mathbf{e}_k + \mathbf{x}) \\ &\quad - \mathbf{q}^\top \Big(\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\, \mathbf{e}_k + \mathbf{x} \Big), \quad \forall j. \end{aligned} \tag{6.57}$$

The remaining part of the proof concerns the simplification and rearrangement of equation (6.57).

We focus on the quadratic term of $P(\mathbf{z}_j)$ and obtain

$$
\begin{aligned}
&\frac{1}{2}\,(\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k + \mathbf{x})^\top \mathbf{Q}\,(\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k + \mathbf{x}) = \\
&\frac{1}{2}\,\Big[\mathbf{e}_j^\top \mathbf{Q}\,\mathbf{e}_j + \Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big)^\top \mathbf{Q}\,\Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big) + \mathbf{x}^\top \mathbf{Q}\,\mathbf{x}\Big] \\
&- \mathbf{e}_j^\top \mathbf{Q}\,\Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big) + \mathbf{e}_j^\top \mathbf{Q}\,\mathbf{x} - \mathbf{x}^\top \mathbf{Q}\,\Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big) = \\
&\frac{1}{2}\,\Big[Q_{j,j} + \sum_{k=a(j)}^{b(j)} x_k\,Q_{k,k} + \mathbf{x}^\top \mathbf{Q}\,\mathbf{x}\Big] \\
&- \sum_{k=a(j)}^{b(j)} x_k\,Q_{j,k} + \mathbf{e}_j^\top \mathbf{Q}\,\mathbf{x} - \mathbf{x}^\top \mathbf{Q}\,\Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big)
\end{aligned}
\tag{6.58}
$$

thereby exploiting that

$$
(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k)^\top \mathbf{Q}\,(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k) = \sum_{k=a(j)}^{b(j)} x_k\,Q_{k,k} \tag{6.59}
$$

since $x_{k_1}x_{k_2} = 0$ for $k_1 \neq k_2$, $k_1, k_2 \in \{a(j), ..., b(j)\}$. Expressing equation (6.57) with (6.58) leads to

$$
\begin{aligned}
\frac{1}{2}\,\mathbf{x}^\top \mathbf{Q}\,\mathbf{x} - \mathbf{q}^\top \mathbf{x} \quad \leq \quad & \frac{1}{2}\Big[Q_{j,j} + \sum_{k=a(j)}^{b(j)} x_k\,Q_{k,k} + \mathbf{x}^\top \mathbf{Q}\,\mathbf{x}\Big] \\
& - \sum_{k=a(j)}^{b(j)} x_k\,Q_{j,k} + \mathbf{e}_j^\top \mathbf{Q}\,\mathbf{x} - \mathbf{x}^\top \mathbf{Q}\,\Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big) \\
& - \mathbf{q}^\top\Big(\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k + \mathbf{x}\Big)
\end{aligned}
$$

where $P(\mathbf{x})$ chancels out on both sides. Thus, we have

$$
\begin{aligned}
&\mathbf{x}^\top \mathbf{Q}\,\Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big) - \mathbf{e}_j^\top \mathbf{Q}\,\mathbf{x} + \mathbf{q}^\top\Big(\mathbf{e}_j - \sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k\Big) \quad \leq \\
&\frac{1}{2}\Big[Q_{j,j} + \sum_{k=a(j)}^{b(j)} x_k\,Q_{k,k}\Big] - \sum_{k=a(j)}^{b(j)} x_k\,Q_{j,k}.
\end{aligned}
$$

which finally completes the proof,

$$\frac{1}{2}\,Q_{j,j} - \Big(\sum_{k=a(j)}^{b(j)} Q_{j,k} - \frac{1}{2}\,Q_{k,k}\Big)\,x_k \quad \geq \quad \Big(\sum_{k=a(j)}^{b(j)} x_k\,\mathbf{e}_k - \mathbf{e}_j\Big)^{\top}\Big(\mathbf{Q}\,\mathbf{x} - \mathbf{q}\Big).$$

□

6.5 Numerical Evaluation

The d.c. based extension to multiple labels opens our algorithms to a much broader class of problems, besides the tomographic reconstruction of discrete images. In many computer vision tasks, for instance, one wishes to find an assignment between pixels and labels such that each pixel uniquely corresponds to a label (labeling problem). This describes, for instance, the situation in image denoising where an image, containing a certain set of grayvalues, has been corrupted by noise and, therefore, one must remove the noise in order to restore the true image. Image denoising is related to our problem, in fact, it can be understood as a special instance of the discrete reconstruction problem.

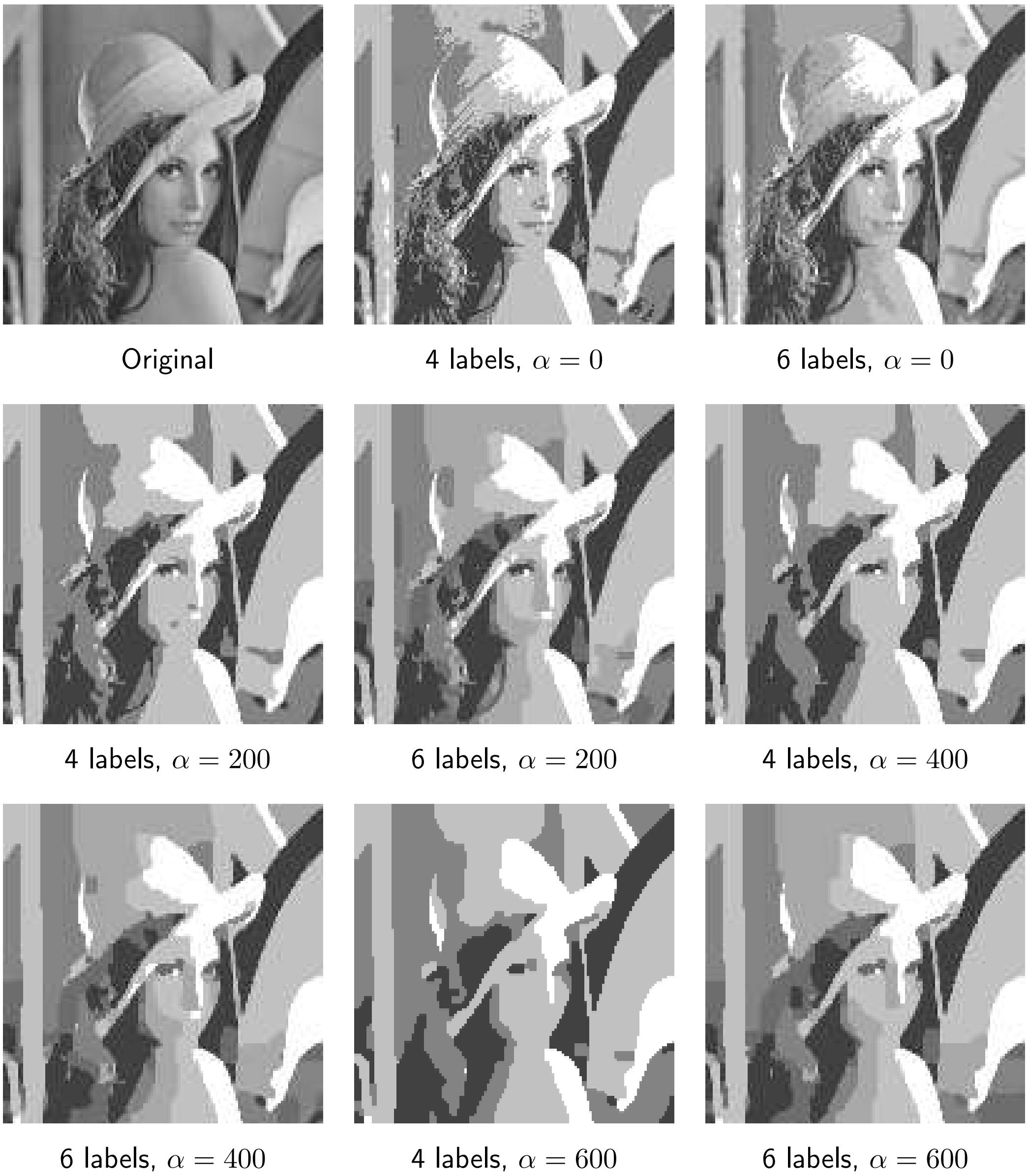
We consider the problem of image denoising first since it allows us to study the behavior of our algorithm and the smoothness priors without any additional effects caused by a very small number of projections. At this, we compare our approach to different algorithms well-known in the computer vision community. The second part of the evaluation is then, of course, devoted to the tomographic reconstruction of multiclass images.

6.5.1 Image Labeling

Image labeling, or image denoising in our particular case, is closely related to the tomographic reconstruction problem. Imagine a noisy image, $n \times m$, embedded into the three dimensional space, i.e. its dimension is then $n \times m \times 1$, and take a single projection parallel to the z-axis such that each ray passes exactly a single pixel. The corresponding projection matrix $\mathbf{A}$ contains a single 1 entry per row since the side length of a squared pixel or respectively cubic voxel is assumed to be 1. Particularly, $\mathbf{A}$ becomes the identity matrix if we enumerate the rays and pixels accordingly and, thus, the data term in equation (6.15) reduces to

$$\frac{1}{2}\|\mathbf{G}\,\mathbf{x} - \mathbf{b}\|^2. \tag{6.60}$$

Note that all other terms of equation (6.15) are not affected by this and, hence, remain unchanged.



Original

4 labels, $\alpha = 0$

6 labels, $\alpha = 0$

4 labels, $\alpha = 200$

6 labels, $\alpha = 200$

4 labels, $\alpha = 400$

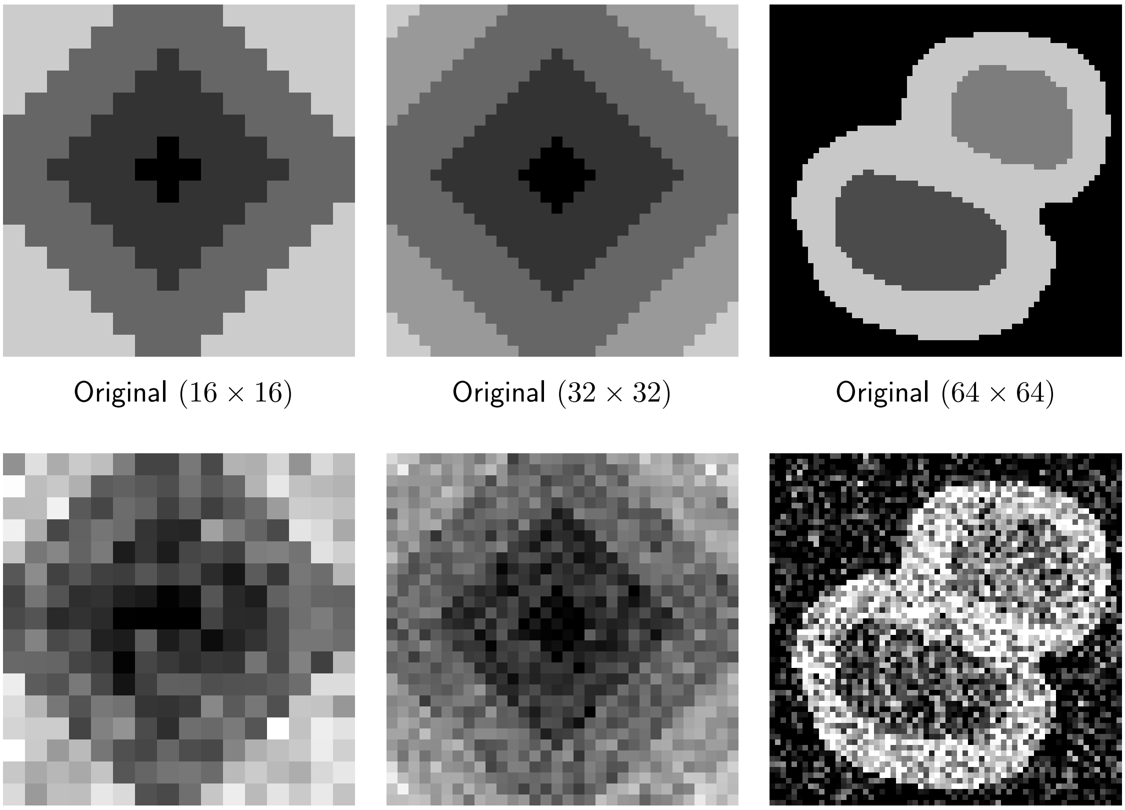
6 labels, $\alpha = 400$

4 labels, $\alpha = 600$

6 labels, $\alpha = 600$

Figure 6.9: **Image labeling:** Regarding energy functional E_{II}, equations (6.16) and (6.15), this figure shows the labeling results of the MC-DCA applied to the Lena image (128×128; upper left) with 4 and 6 predefined labels. This results nicely show the increasing influence of the smoothness prior for different values of α.

In case of image denoising, we compare our approach to the α-expansion algorithm of Boykov et al [23] which is a popular energy minimization technique in computer vision. It can be shown that its solution is within a certain range of the global optimum [23]. The expansion algorithm has been successfully applied to image segmentation [21] and restoration [19, 23] as well as to medical imaging [20], stereo [83] and motion [119]. The algorithm iteratively selects a label and tries to expand the set of pixels assigned to this label. Though the overall problem is non-binary, each step of the α-expansion algorithm yields a binary minimization problem which is solved by a single graph cut [60], see section 3.3.3. The algorithm terminates if no further improvement is possible.



Original (16×16) Original (32×32) Original (64×64)

Noisy (16×16) Noisy (32×32) Noisy (64×64)

Figure 6.10: **Image labeling:** This figures shows the groundtruth and noisified images. The (16×16) image on the lower left is used in connection with the SDP relaxation which is, compared to the (32×32) test image for SOCP, reduced in size in order to have reasonable computation times. Finally, the α-expansion approach is evaluated with the image on the lower right.

In contrast to the α-expansion which directly acts on the discrete label set we assign a binary vector to each pixel where a single 1 entry indicates the label associated with the pixel. A similar description is used for instance by Keuchel [76] and [75] who suggests a relaxation based on semidefinite programming (SDP) in order to approximatively solve the image labeling problem. It is known [137] that such relaxations are able to find good approximations to combinatorial problems, however, the application of semidefinite programming is limited to small size problems only since it squares the number of variables. In order to circumvent this drawback, relaxations based on second-order cone programming (SOCP) [92] have been proposed in [77] and [97] where the latter has been applied to computer vision problems in [88]. In contrast to our MC-DCA approach, semidefinite and second-order cone programming relaxations do not provide hard decisions, instead, they deliver a soft assignment, i.e. the vector must be positive and sum to one but is not necessarily binary. Hence, a rounding step must be applied afterwards to eventually yield a binary solution, i.e. in our SDP and SOCP experiments we assign the maximal component to 1 and all others to 0. Note, that this is somewhat similar to the linear programming relaxations we considered in chapter 4 which originally motivated us to include the rounding step in the overall optimization process and, thus, led us to our d.c. framework, chapter 5.

For evaluation purposes we consider the semidefinite [76] and the second-order cone programming relaxation [97, 88] besides the α-expansion algorithm [23]. We implemented the α-expansion algorithm in C++ including the graph cut C library gratefully provided by Vladimir Kolmogorov[1]. For the second-order cone programming approach of [97, 88] we setup a matlab program invoking the external mosek[2] solver to perform the actual optimization task. Similarly in case of semidefinite programming [76] where also a matlab script was used calling the PENSDP[3] solver.

Our first experiment concerns the behavior of the smoothness prior S_{II}, equation (6.14), for different values of α. Therefore, we applied our MC-DCA approach with a set of either 4 or 6 labels to the Lena image, 128×128, shown in figure 6.9, where the labels were just roughly picked from the histogram of the Lena image but without any deeper justification. This scenario does not involve any noise which has been added to the original image. However, the original image contains much more grayvalues besides the prototypical labels, thus, we simply considered these grayvalues as corrupted by noise. As can be seen in figure 6.9 which shows the results of this experiment, all labels are present in the labeling images. Additionally, one can nicely see the influence of the smoothness prior for different weighting factors α.

[1] http://www.adastral.ucl.ac.uk/~vladkolm/software.html

[2] www.mosek.com

[3] http://www.penopt.com/

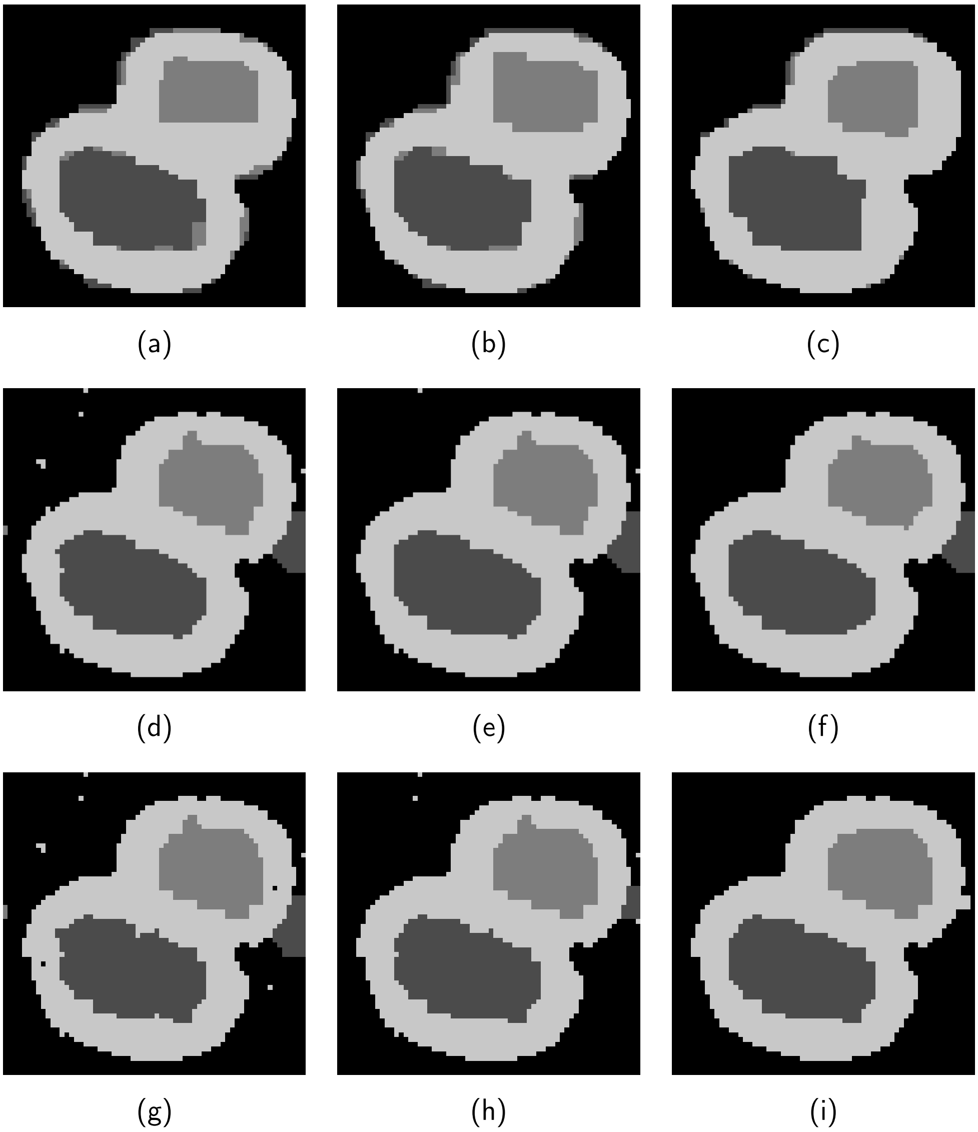
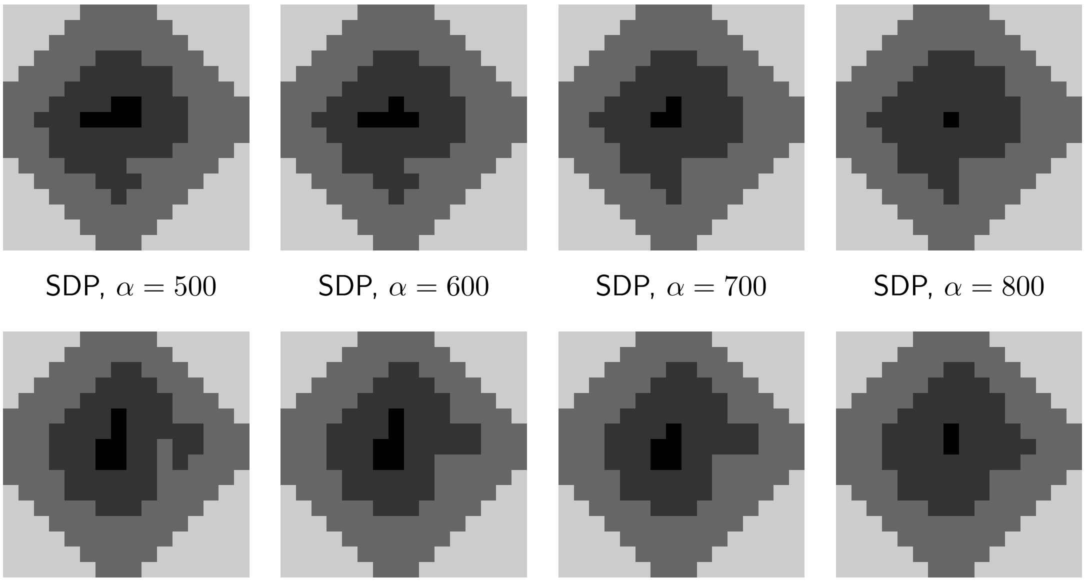


Figure 6.11: **Image labeling:** (a)–(c) MC-DCA with smoothness prior S_I and parameters $c := 10$ and $\alpha \in \{10, 12, 14\}$ from left to right. (d)–(f) MC-DCA with smoothness prior S_{II} and $\alpha \in \{3500, 4250, 5750\}$. (g)–(i) Graphcuts with $\mathcal{L}_2$-norm smoothness prior truncated at $c := 50$, equation (6.12), and regularization parameter $\alpha \in \{4, 5, 6\}$. It can be seen that the smoothness prior S_I is too sensitive in case of heavy noise while S_{II} is robust. Further, MC-DCA supplemented with S_{II} leads to good results, also compared to the results of the α-expansion algorithm.

The figures 6.11, 6.12, and 6.13 show the results of the α-expansion, SDP, and SOCP approaches compared to the MC-DCA. As can be seen, our approach yields in all cases comparable results.



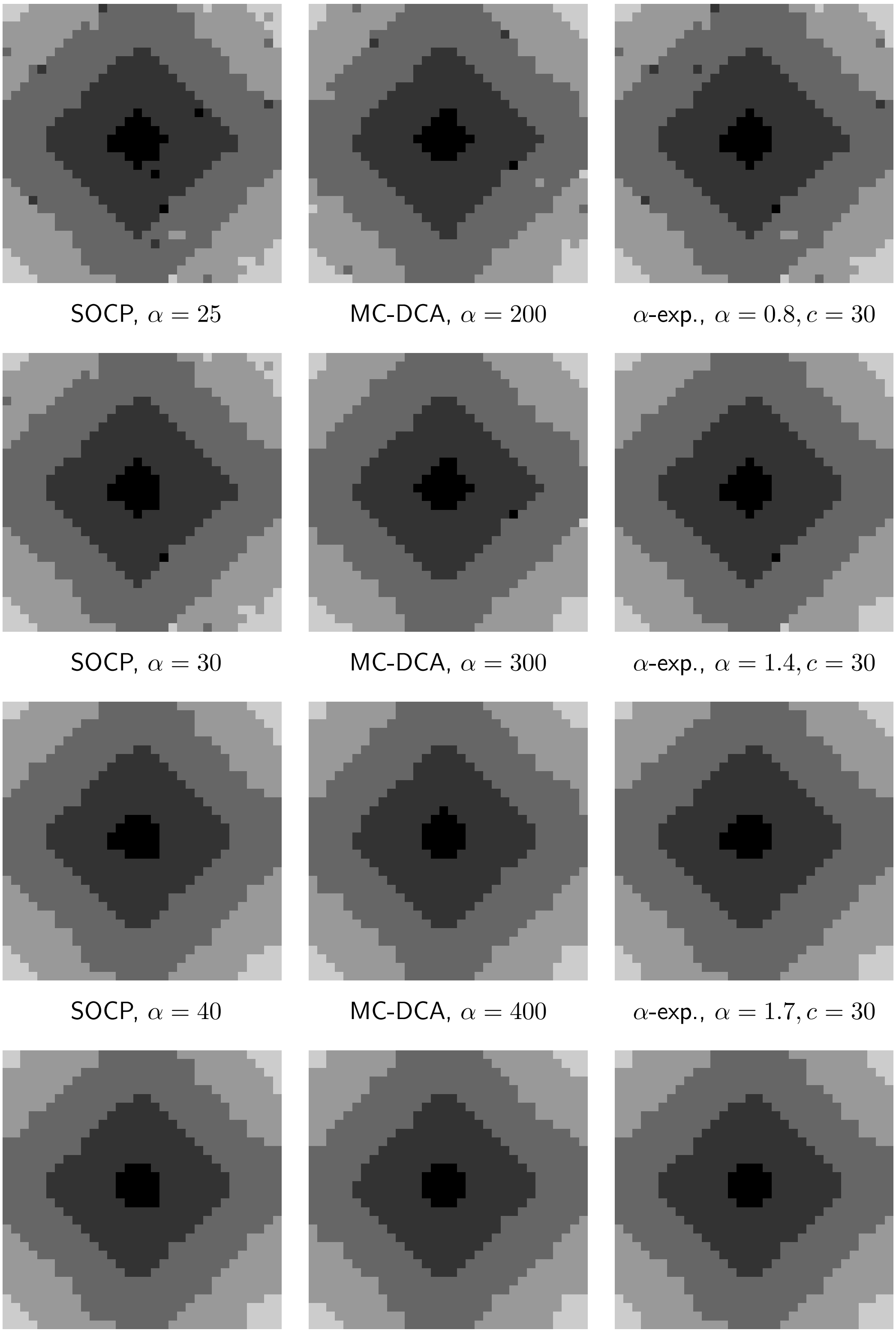
SDP, $\alpha = 500$ SDP, $\alpha = 600$ SDP, $\alpha = 700$ SDP, $\alpha = 800$

MC-DCA, $\alpha = 225$ MC-DCA, $\alpha = 275$ MC-DCA, $\alpha = 325$ MC-DCA, $\alpha = 350$

Figure 6.12: **Image labeling:** Results of SDP and MC-DC algorithm.

6.5.2 Image Reconstruction

In order to get a first impression of the MC-DC approach with respect to image reconstruction we conduct a small experiment and present some of its intermediate solutions, figure 6.14. The problem was setup from the image shown in the upper left corner of figure 6.14 by taking the horizontal, vertical, and one diagonal projection. The middle image in the upper row of figure 6.14 shows the convex solution obtained right after the first iteration of the algorithm. As can be seen, the content of the image is hard to perceive from the convex solution, even if we know the four different graylevels involved in the true image. However, as the number of iterations increase, more and more details become recognizable until we finally end up with a solution containing only the a priori supplied graylevels. The final solution furthermore happened to be the original image in this case. Note that we originally considered both functionals, equations (6.15) and (6.16), for this experiment. However, E_I was clearly inferior compared to E_{II} since it was not possible to reconstruct the original image from three projections using E_I. Another point that makes functional E_{II} preferable is that the concave part consists only of the concave regularizer which vanishes for all $\mathbf{x} \in \Omega^n_{\{0,1\}}$ and the convex part contains the quadratic form we wish to minimize. Thus, we can apply the global optimality conditions from section 6.4 to E_{II}. For this reasons, we consider only functional E_{II} for the remaining reconstruction experiments.



SOCP, $\alpha = 25$

MC-DCA, $\alpha = 200$

α-exp., $\alpha = 0.8, c = 30$

SOCP, $\alpha = 30$

MC-DCA, $\alpha = 300$

α-exp., $\alpha = 1.4, c = 30$

SOCP, $\alpha = 40$

MC-DCA, $\alpha = 400$

α-exp., $\alpha = 1.7, c = 30$

SOCP, $\alpha = 50$

MC-DCA, $\alpha = 500$

α-exp., $\alpha = 1.9, c = 30$

Figure 6.13: **Image labeling:** Results of SOCP, α-expansion, and MC-DC algorithm.

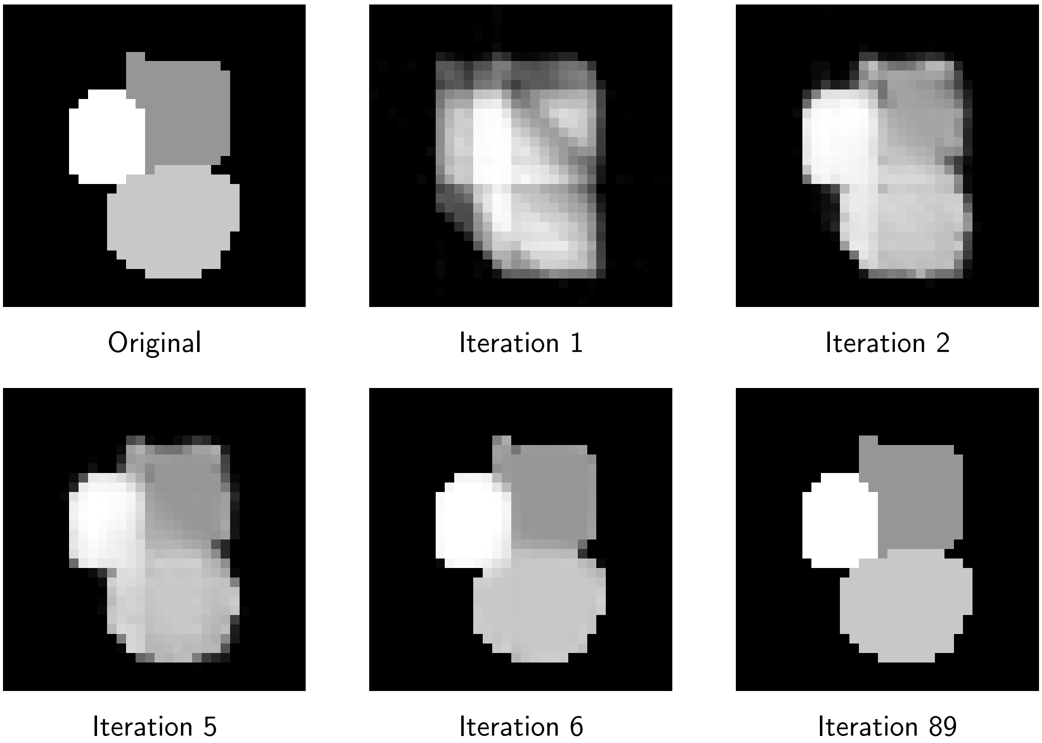
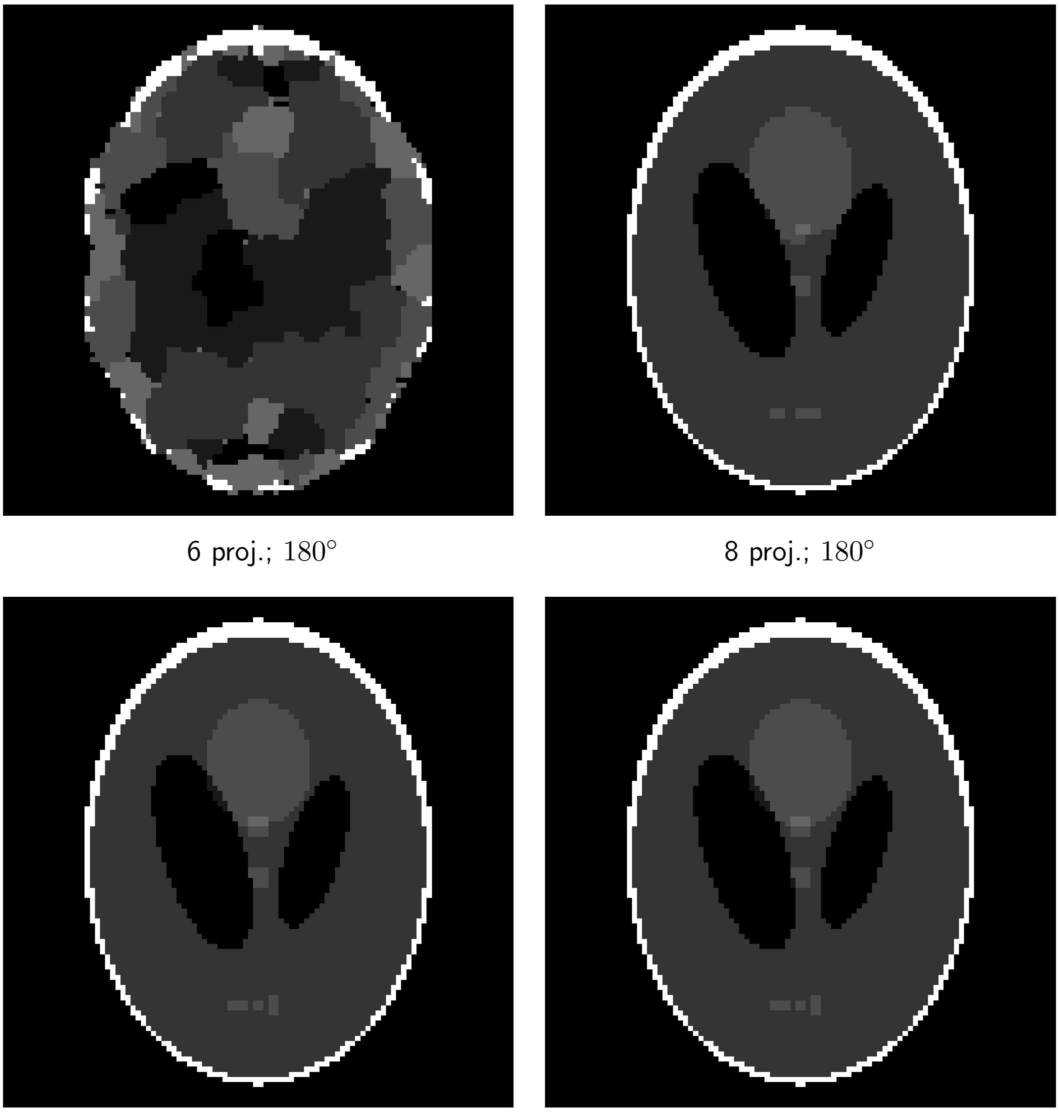


Figure 6.14: **Image reconstruction:** We setup a reconstruction problem using three projections, i.e. $0°$, $45°$, and $90°$, from the image shown in the upper left corner. The image at iteration 1 shows the convex solution of the relaxed problem while the remaining images were obtained at various iterations afterwards.

For our next experiment we took a varying number of projections from the well-known Shepp-Logan phantom image which was of size 100×100 and contained 6 different grayvalues. The phantom can be seen in the lower right corner of figure 6.15 since the reconstruction shown there led to the very same. As previously, projections were taken equally spaced over either $90°$ or $180°$ degree. Clearly, the problems over a range of $90°$ are much more difficult which reflects in the number of projections that were necessary in order to reach similar results between the $90°$ and $180°$ experiments, figures 6.15 and 6.16.

As mentioned in the introductory part of this chapter, we proposed a different approach to the multiclass reconstruction problem in [123] which is also based on d.c. programming but still quite different from the MC-DCA presented here. The former approach places a binary regularizer in each subinterval $[g_i, g_{i+1}]$ of $[g_1, g_l]$ and then starts to push each variable towards the borders of a subinterval. The advantage of this method is that it always uses a single variable per pixel, no matter how many labels are involved in the problem. On the other side, however,



6 proj.; 180°

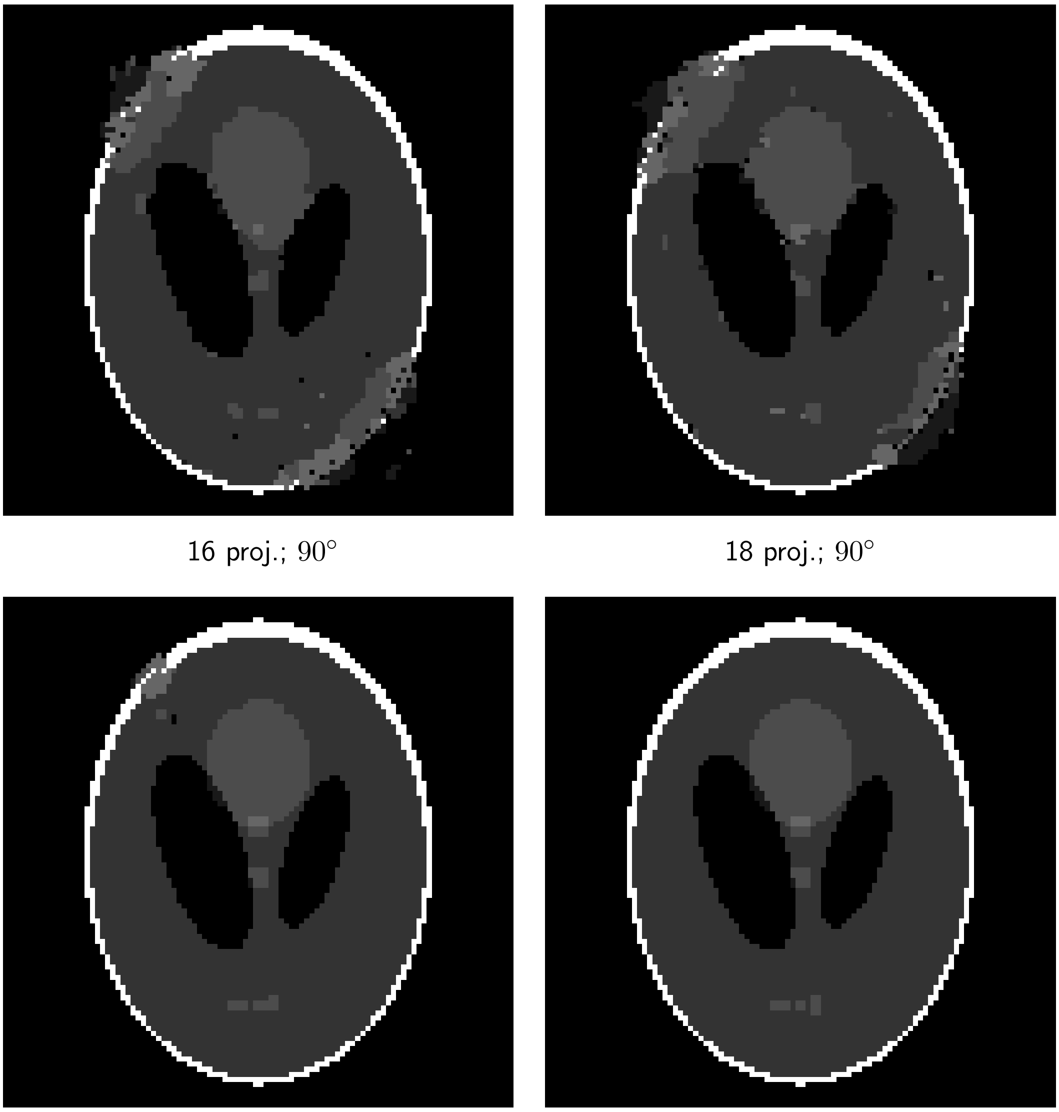
8 proj.; 180°

14 proj.; 180°

18 proj.; 180°

Figure 6.15: **Image reconstruction:** Using projections over 180° we setup different reconstruction problems for the Shepp-Logan phantom; the result in the lower right corner equals the original. The results were obtained from our MC-DCA optimizing functional E_{II}, equations (6.15) and (6.16).

a heuristic procedure must be invoked which allows to exchange variables between different subintervals. Its evaluation in [123] includes the same experiments over 180° we conducted here with the Shepp-Logan phantom. Comparing both shows that our new approach already yields high quality reconstructions from 8 projections, second image from the left in the top row



16 proj.; 90°

18 proj.; 90°

20 proj.; 90°

22 proj.; 90°

Figure 6.16: **Image reconstruction:** Using projections over 90° we setup different reconstruction problems for the Shepp-Logan phantom; the result in the lower right corner equals the original. The results were obtained from our MC-DCA optimizing functional E_{II}, equations (6.15) and (6.16).

of figure 6.15, while our former approach was still not able to reconstruct the original image from 64 projections.

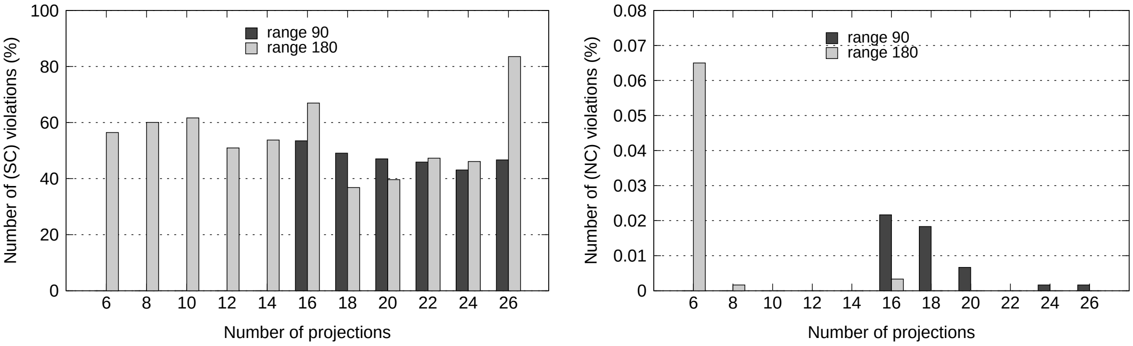


Figure 6.17: These figures plot the number of violated sufficient (SC) and the necessary conditions (NC), section 6.4, for the Shepp-Logan experiments over $90°$ and $180°$. Since we did not perform any reconstructions over $90°$ with less than 16 projections the first half of the dark bars is left out. As can be seen in the left plot, the sufficient condition does not drop when increasing the number of projections, even if the results become close or equal to the groundtruth image. We have observed a similar behavior also in the binary case where it is possible to apply the optimality conditions without further modifications, section 6.4. This indicates that the sufficient condition itself is too strict for our purposes. In contrast to (SC) the violated necessary conditions (NC), shown on the right side, decrease as the amount of projections growth and finally become zero.

7 Conclusion and Further Work

In the present work, we considered the problem of binary tomography and more generally discrete tomography, section 3.2. Related to the optimization of such problems, stochastic sampling methods like simulated annealing, section 3.3.1, have become quite popular due to the inherent combinatorial complexity. Although it can be proven that simulated annealing locates a globally optimal solution it is less advantageous in practice. If applied properly the computational time exceeds the time spent for a complete search and, further, its probabilistic procedure prohibits the reproduction of results.

In contrast, we pursued a different optimization strategy since we focused on continuous relaxations and concentrated on purely deterministic optimization approaches. Motivated by the work of [50, 62] and [80], we introduced different linear programming relaxations in chapter 4 and, as we are primarily interested in spatially coherent objects, we supplemented an additional prior which favors smooth solutions. Therewith, we efficiently computed a globally optimal solution of the relaxed problem and applied, subsequently, different rounding schemes in order to obtain a binary solution as approximation to the original problem. Our results, section 4.5, indicate that our algorithms yield reasonable approximations of the binary problem and, furthermore, a comparison in [7] shows that the results of $(reg - BIF)$ are comparable to their network flow approach.

Considering the evaluation, section 4.5, of the threshold-based rounding procedures, section 4.4.1, it is interesting that the optimal threshold seems to be systematically distributed around $t = 0.5$. This can be further exploited as it renders the rounding scheme to a simple threshold operation of the fractional solution at 0.5 which is cheap concerning computational times and trivial to implement. The investigation of this observation from a theoretical point of view will be part of our further work.

Although the results of the linear programming relaxations are quite promising a drawback of our approach so far has been the decoupling of the optimization and the rounding stage. Therefore, we proposed an optimization framework based on d.c. programming in chapter 5 which performs the rounding within the optimization process and is, further, general enough to fit the linear programming approaches from chapter 4. A comparison of the results obtained with the new implicit, section 5.4.2, and the old explicit rounding step, section 4.5, shows that the novel approaches improve the results, even from a smaller number of projections. Further, our experiments show that the d.c. and linear programming based reconstruction algorithms yield comparable results to simulated annealing if the latter is run for a long period of time and carefully cooled down. Concerning the d.c. framework with quadratic programming and simulated annealing the same conclusion can be found in [132].

The d.c. based linear programming approaches consist of a sequence of linear programs which are subsequently solved. Using, at this, interior point solvers is fast for the individual linear programs but contains redundancy when iteratively solving similar problems. Therefore, so called warm-start strategies [140] utilize that the subsequent linear program is just a perturbed version of the previous problem in order to reduce the number of interior point iterations. Note that this perfectly suits our situation as the constraint set remains the same throughout all iterations, only the objective vector changes. So far, we do not exploit any accelerations but expect a significant reduction of the reconstruction time and, thus, include such considerations in our future work.

In section 5.5, we additionally extended our iterative reconstruction framework for binary tomography with an Expectation-Maximization (EM) step to improve its behavior in the presence of degradations during data acquisition and considered, at this, quadratic programming instead of linear programming, see also [124]. For evaluation purposes we defined two different degradation models. The same reconstruction algorithm can be applied to either of them which accurately estimates an unknown scale-parameter σ, during the reconstruction. Our results show that our approach stabilizes the reconstruction process in the presence of degradations.

Regarding the Q function in the EM-step, further work includes an adaptive sampling strategy of the supporting points. This is important for two reasons: First, it is expected to produce a more accurate approximation of the integral especially in areas where the true σ is suspected. Second, it should also reduce the number of supporting points since we can skip areas which are of low interest. The latter is expected to further speed up our algorithm.

We suppose that our approach is sufficiently general to be applied to other combined reconstruction and missing parameter estimation scenarios as well. This will also be subject to our future work.

In chapter 6, we successfully extended our so far binary d.c. based reconstruction framework to the case of multiple discrete values. We, therefore, proposed a novel concave regularization term which yields, if applied to binary tomography, our former binary regularizer as special case, section 6.2.2 and 6.3.

In contrast to the binary case, the multiclass scenario requires to treat blurring effects caused by smoothing over the borders of objects properly. For this reason, we introduced two different, discontinuity-preserving smoothness priors, S_I and S_{II}, which are suitable for the multiclass optimization framework, section 6.2.3. Our results indicate that S_{II} is preferable as it is more robust than S_I and, thus, typically yields better results.

In [10] sufficient and necessary conditions have been proposed regarding the global optimality of quadratic minimization problems subject to binary constraints. We showed in section 6.4 that these can be adapted such that they also fit the multiclass problem. Similarly to the results published concerning binary tomography [118], the sufficient condition is too strict for our needs while the necessary condition applies, figure 6.17.

We numerically evaluated our multiclass algorithm with respect to image reconstruction and image labeling, as the later can be perceived as a special case of the reconstruction problem. At this, we compared our approach to state of the art labeling algorithms, namely the graph cuts based α-expansion algorithm [23], the second-order cone programming (SOCP) relaxation proposed in [97, 88], and the semidefinite programming (SDP) relaxation in [76]. Section 6.5.1, shows that our approach achieves competitive results and is, further, applicable to large scale problem instances, in contrast to (SDP). Concerning image reconstructions, we demonstrated in section 6.5.2 that our approach yields very promising results and improves the results of our former multiclass d.c. approach which has been proposed in [123]. We believe that our multiclass framework can be usefully applied also to other discrete optimization scenarios, besides image labeling and image reconstruction. Therefore, our future work also concerns its application to related optimization problems.

Speaking about the DC framework in general, we have still open questions concerning the optimal selection of the penalty parameter μ: Is there an optimal schedule for μ such that we can guarantee to obtain the best solution achievable with our DC framework? Note that this is somewhat similar to the cooling schedule in simulated annealing algorithm. Another related and open question is to somehow bound the distance to the globally optimal discrete solution. Can we guarantee in advance that we miss the globally optimal solution at most by a certain amount? As these questions are extremely difficult, answers to any of them are quite precious and will be part of our future considerations.

A Dirac Delta distribution

From a mathematical point of view the Dirac delta function cannot be considered as a function due to its singularity, instead it is called a distribution.

A distribution g generalizes the concept of a function and is introduced by

$$\{g(t), \phi(t)\} \to Z$$

where an arbitrary function ϕ from a prescribed set of test functions is mapped to a number Z. The mapping itself is usually written in terms of an integral

$$\langle g(t), \phi(t) \rangle := \int_{-\infty}^{\infty} g(t)\phi(t) \, \mathrm{d}t = Z$$

which is, however, not meant in the sense of Riemannian integration.

Particularly, the Dirac delta distribution δ maps any continuous (test) function ϕ to

$$\int_{-\infty}^{\infty} \phi(t)\delta(t) \, \mathrm{d}t = \phi(0).$$

B Fourier transform

The definition of the Fourier and the inverse Fourier transform might slightly vary between different books. Nevertheless, we stick to the following definitions concerning our work.

Continuous Fourier transform

Let $f : \mathbb{R}^n \to \mathbb{C}^n$ be a function, then its Fourier transform (FT) $\mathcal{F}$ and its inverse Fourier transform (IFT) $\mathcal{F}^{-1}$ are defined by

$$F(\mathbf{u}) = \mathcal{F}\left[f(\mathbf{x})\right] := \int_{\mathbb{R}^n} f(\mathbf{x}) \exp\Big(-\mathrm{i}\, 2\pi\, \mathbf{u}^\top \mathbf{x}\Big)\, \mathrm{d}\mathbf{x} \tag{B.1}$$

$$f(\mathbf{x}) = \mathcal{F}^{-1}\left[F(\mathbf{u})\right] := \int_{\mathbb{R}^n} F(\mathbf{u}) \exp\Big(\mathrm{i}\, 2\pi\, \mathbf{u}^\top \mathbf{x}\Big)\, \mathrm{d}\mathbf{u}. \tag{B.2}$$

Discrete Fourier transform

Let $x := \{x_i \in \mathbb{C}\}$, $i \in \{0, ..., N-1\}$, be a sequence of complex numbers then the discrete Fourier transform (DFT) transforms x into another sequence of complex numbers $X = \{X_s \in \mathbb{C}\}$, $s \in \{0, ..., N-1\}$, and vice versa the inverse discrete Fourier transform. The transformations for a one-dimensional arrays are defined by

$$X_s = \mathcal{F}\left[x\right] := \sum_{j=0}^{N-1} x_j \exp\left(-\mathrm{i}2\pi \frac{js}{N}\right) \tag{B.3}$$

$$x_j = \mathcal{F}^{-1}\left[X\right] := \frac{1}{N} \sum_{s=0}^{N-1} X_s \exp\left(\mathrm{i}2\pi \frac{js}{N}\right) \tag{B.4}$$

and for a two-dimensional arrays $x := \{x_{j,k} \in \mathbb{C}\}$, $j \in \{0, ..., N-1\}$, $k \in \{0, ..., M-1\}$, respectively as

$$X_{s,t} = \mathcal{F}\left[x\right] \quad := \quad \sum_{k=0}^{M-1}\sum_{j=0}^{N-1} x_{j,k} \exp\left(-\mathrm{i}2\pi\left(\frac{js}{N}+\frac{kt}{M}\right)\right) \tag{B.5}$$

$$x_{j,k} = \mathcal{F}^{-1}\left[X\right] \quad := \quad \frac{1}{NM}\sum_{t=0}^{M-1}\sum_{s=0}^{N-1} X_{s,t} \exp\left(\mathrm{i}2\pi\left(\frac{js}{N}+\frac{kt}{M}\right)\right) \tag{B.6}$$

See for instance [25] for a comprehensive introduction to the continuous and the discrete Fourier transform.

C Total Unimodularity and Bipartite Graphs

Definition 14. *A matrix $\mathbf{A}$ is said to be totally unimodular if the determinant of every square submatrix is either -1, 0, or 1.*

Theorem 4 [120] *Let matrix $\mathbf{A}$ be totally unimodular and $\mathbf{b}$ be an integer vector, then the vertices of the polytope*

$$P := \{\mathbf{x} \mid \mathbf{A}\,\mathbf{x} \leq \mathbf{b}\}$$

are all integer vectors.

Proof. Assume $\mathbf{z}$ to be a vertex of P, then there exists a subset of inequalities in P, defined by $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$, such that $\mathbf{z}$ is the unique solution of $\tilde{\mathbf{A}}\,\mathbf{z} = \tilde{\mathbf{b}}$. Matrix $\tilde{\mathbf{A}}_{i,k}$ is obtained by eliminating the i-th row and the k-th column from $\tilde{\mathbf{A}}$, the cofactor matrix is then defined by

$$\mathrm{cof}_{i,k}\tilde{\mathbf{A}} := (-1)^{i+k} \det \tilde{\mathbf{A}}_{i,k}.$$

We solve $\tilde{\mathbf{A}}\,\mathbf{z} = \tilde{\mathbf{b}}$ by means of Cramer's rule for each component of $\mathbf{z}$,

$$z_k = \frac{\sum_i \tilde{b}_i \, \mathrm{cof}_{i,k}\tilde{\mathbf{A}}}{\det \tilde{\mathbf{A}}} = \begin{cases} \sum_i \tilde{b}_i \, \mathrm{cof}_{i,k}\tilde{\mathbf{A}} & \text{if } \det \tilde{\mathbf{A}} = 1 \\ \\ -\sum_i \tilde{b}_i \, \mathrm{cof}_{i,k}\tilde{\mathbf{A}} & \text{if } \det \tilde{\mathbf{A}} = -1. \end{cases}$$

Note, that the case $\det \tilde{\mathbf{A}} = 0$ cannot occur since $\tilde{\mathbf{A}}$ is non-singular.

Consider $\sum_i \tilde{b}_i \, \mathrm{cof}_{i,k}\tilde{\mathbf{A}}$, by the total unimodularity of $\tilde{\mathbf{A}}$ we have that $\mathrm{cof}_{i,k}\tilde{\mathbf{A}} \in \{-1, 0, 1\}$. Since $\mathbf{b}$ is integer it follows that the sum and thus z_k must be integer. □

Definition 15. *The incidence matrix $\mathbf{A}_{inc}$ of a graph $G := (V, E)$ is defined by*

$$(a_{v,e})_{v \in V, e \in E} := \begin{cases} 1 & \text{if } v \in e \\ \\ 0 & \text{if } v \notin e \end{cases}$$

Definition 16. *The constraint matrix $\mathbf{A}$ of a graph $G := (V, E)$ is defined by*

$$\mathbf{A} := \begin{pmatrix} \mathbf{A}_{inc} \\ -\mathbf{I} \\ \mathbf{I} \end{pmatrix}$$

Theorem 3 [120] *A graph $G := (V, E)$ is bipartite if and only if its constraint matrix is totally unimodular.*

Proof. Let $\mathbf{A}$ be the constraint matrix of the bipartite graph G.

"$\Rightarrow$"

Consider any $k \times k$ submatrix $\tilde{\mathbf{A}}$ of $\mathbf{A}$, we prove by induction on k that $\tilde{\mathbf{A}}$ is totally unimodular. For $k = 1$ this is trivially true. Assume the induction hypothesis is true for all submatrices of size $(k-1) \times (k-1)$.

(i.) If there exists a row in $\tilde{\mathbf{A}}$ with only zero entries we have $\det \tilde{\mathbf{A}} = 0$.

(ii.) If there exists a row in $\tilde{\mathbf{A}}$ with only a single non-zero entry $a \in \{-1, 1\}$ then by means of Laplace's formula we expand around a and apply the induction hypothesis on the resulting $(k-1) \times (k-1)$ submatrix. Hence we have $\det \tilde{\mathbf{A}} \in \{-1, 0, 1\}$.

(iii.) Every row of $\tilde{\mathbf{A}}$ has more than one non-zero entry. Since the lower part of $\mathbf{A}$ contains only a negative and a positive identity matrix $\tilde{\mathbf{A}}$ must be entirely from the upper part of $\mathbf{A}$, i.e. $\mathbf{A}_{inc}$, and thus all its entries are either 0 or 1.

By the fact that G is bipartite we partition the rows of $\tilde{\mathbf{A}}$ into two sections corresponding to the partitioning of the vertices V. Consider the row sum vector $\mathbf{r}$ and the column sum vector $\mathbf{c}$ of $\tilde{\mathbf{A}}$, it holds that

$$\sum_{i=1}^{k} r_i = \sum_{i=1}^{k} c_i \tag{C.1}$$

since $\tilde{\mathbf{A}}$ is a square matrix with binary entries. In case of a bipartite graph G each column of $\mathbf{A}_{inc}$ contains exactly one entry in each partition. From $c_i \leq 2$, $r_i \geq 2$, and equation (C.1) it follows that $c_i = r_i = 2$. Therefore each column of $\tilde{\mathbf{A}}$ contains a 1 entry in the upper section and 1 entry in the lower section. If we add the row vectors in each section we obtain the $\mathbf{1}$ vector in both cases. Using the coefficient 1 for the vectors in one section and -1 for the vectors in the other section we construct a linear combination which equals $\mathbf{0}$. Consequently, the rows of $\tilde{\mathbf{A}}$ are linearly dependent and $\det \tilde{\mathbf{A}} = 0$.

"$\Leftarrow$"

Let $\mathbf{A}$ be totally unimodular and assume that G is not bipartite, then G contains an odd cycle. In contradiction to the total unimodularity of $\mathbf{A}$ the submatrix corresponding to the odd cycle has determinant 2. □

Corollary 17. *A graph $G := (V, E)$ is bipartite if and only if its incidence matrix $\mathbf{A}$ is totally unimodular.*

Proof. Consequence of theorem 3. □

D Spectral Projected Gradients (SPG)

Considering optimization problems of the following type

$$(P) \qquad \min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \Omega$$

where Ω is a convex and closed set. It is further required that f has continuous partial derivatives on an open set containing Ω and we denote the gradient of f with g. The following spectral projected gradient (SPG) method, algorithm 15, has been proposed in [14] for the optimization of problem (P).

Algorithm 14 Line Search (called from SPG algorithm 15)

```
$f_{\max} := \max\{f(\mathbf{x}^{k-j}) \mid 0 \leq j \leq \min\{k, m-1\}\}$
$\mathbf{x}_+ := \mathbf{x}^k + \mathbf{d}^k$
$\delta := \langle g(\mathbf{x}^k), \mathbf{d}^k \rangle$
$\alpha := 1$
while $f(\mathbf{x}_+) > f_{\max} + \alpha\,\gamma\,\delta$ do
  $\alpha_{temp} := -\frac{1}{2}\,\alpha^2\,\delta/(f(\mathbf{x}_+) - f(\mathbf{x}^k) - \alpha\,\delta)$
  if $\alpha_{temp} \geq \sigma_1$ and $\alpha_{temp} \leq \sigma_2\,\alpha$ then
    $\alpha := \alpha_{temp}$
  else
    $\alpha := \alpha/2$
  end if
  $\mathbf{x}_+ := \mathbf{x}^k + \alpha\,\mathbf{d}^k$
end while
return $\alpha^k := \alpha$
```

Regarding the parameters involved in algorithm 15 the following values were suggested in [14] and worked without any difficulty in our case:

- $m := 10$
- $\lambda_{\min} := 10^{-3}$
- $\lambda_{\max} := 10^{3}$
- $\lambda^0 := \min\{\lambda_{\max}, \max\{\lambda_{\min}, 1/\|P_\Omega(\mathbf{x}^0 - g(\mathbf{x}^0)) - \mathbf{x}^0\|_\infty\}\}$
- $\gamma := 10^{-4}$
- $\sigma_1 := 0.1$
- $\sigma_2 := 0.9$

Algorithm 15 Spectral Projected Gradients (SPG)

```
Require: $\mathbf{x}^0 \in \mathbb{R}^n, \quad m \geq 1, \quad \lambda_{\min} > 0, \quad \lambda_{\max} > \lambda_{\min}, \quad \gamma \in (0,1)$
Require: $0 < \sigma_1 < \sigma_2 < 1, \quad \lambda^0 \in [\lambda_{\min}, \lambda_{\max}]$
  $\mathbf{x}^0 := P_\Omega(\mathbf{x}^0)$
  $k := 0$
  while Stopping criterion is not satisfied do
    $\mathbf{d}^k := P_\Omega(\mathbf{x}^k - \lambda^k g(\mathbf{x}^k)) - \mathbf{x}^k$
    Compute $\alpha^k$ using line search algorithm 14.
    $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha^k \mathbf{d}^k$
    $\mathbf{s}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$
    $\mathbf{y}^k := g(\mathbf{x}^{k+1}) - g(\mathbf{x}^k)$
    $\beta^k := \langle \mathbf{s}^k, \mathbf{y}^k \rangle$
    if $\beta^k \leq 0$ then
      $\lambda^{k+1} := \lambda_{\max}$
    else
      $\lambda^{k+1} := \min\left(\lambda_{\max}, \max\left(\lambda_{\min}, \langle \mathbf{s}^k, \mathbf{s}^k \rangle / \beta^k\right)\right)$
    end if
    $k := k + 1$
  end while
  $\mathbf{x}^* := \mathbf{x}^k$
  return $\mathbf{x}^*$
```

E Power Iteration and Inverse Iteration

We briefly introduce two algorithms useful in certain situations for the numerical computation of eigenvalues and eigenvectors of a matrix $\mathbf{A}$. For further details, we refer the reader to [130] or [58].

It is required that the matrix is symmetric and real, $\mathbf{A} = \mathbf{A}^\top \in \mathbb{R}^{n \times n}$, which ensures real eigenvalues $\lambda_1, ..., \lambda_n$ and the existence of a complete set of orthogonal eigenvectors, $\mathbf{v}_1, ..., \mathbf{v}_n$, $\mathbf{v}_i^\top \mathbf{v}_j = 0$, $i \neq j$, $\|\mathbf{v}_i\| = 1\ \forall i$.

The first algorithm, typically referred to as power iteration or power method, can be applied if only the absolute largest eigenvalue and its corresponding eigenvector are needed.

Algorithm 16 Power Iteration

```
Require: $\mathbf{v}^0$ some initial vector with $\|\mathbf{v}^0\| = 1$
for all $k = 1, 2, ...$ do
    $\mathbf{w} = \mathbf{A}\mathbf{v}^{k-1}$
    $\mathbf{v}^k = \mathbf{w}/\|\mathbf{w}\|$
    $\lambda^k = (\mathbf{v}^k)^\top \mathbf{A}\mathbf{v}^k$
end for
```

At this, the sequence λ^k converges towards $|\lambda_i| \geq |\lambda_j|$, $\forall j$, and $\mathbf{v}^k$ to the corresponding eigenvector $\mathbf{v}_i$ respectively.

The second algorithm, known as inverse iteration, permits the computation of an eigenvalue closest to an given estimate μ. However, it requires the solution of a linear equation system during each iteration. Analog to the first algorithm, the desired eigenvalue and corresponding eigenvector are given by the final values of λ^k and $\mathbf{v}^k$.

Algorithm 17 Inverse Iteration

```
Require: $\mu \in \mathbb{R}$ estimate of the eigenvalue
Require: $\mathbf{v}^0$ some initial vector with $\|\mathbf{v}^0\| = 1$
for all $k = 1, 2, ...$ do
    Solve $(\mathbf{A} - \mu\,\mathbf{I})\,\mathbf{w} = \mathbf{v}^{k-1}$ for $\mathbf{w}$
    $\mathbf{v}^k = \mathbf{w}/\|\mathbf{w}\|$
    $\lambda^k = (\mathbf{v}^k)^\top \mathbf{A}\mathbf{v}^k$
end for
```

Bibliography

[1] R. Aharoni, G. T. Herman, and A. Kuba. Binary vectors partially determined by linear equation systems. *Discrete Math.*, 171(1-3):1–16, 1997.

[2] A. J. Baddeley. An error metric for binary images. In W. Förstner and S. Ruwiedel, editors, *Robust Computer Vision: Quality of Vision Algorithms, Proceedings, International Workshop on Robust Computer Vision, Bonn 1992*, pages 59–78. Wichmann Verlag, Karlsruhe, 1992.

[3] A. J. Baddeley. Errors in binary images and L^p version of the Hausdorff metric. *Nieuw Archief voor Wiskunde*, 10:157–183, 1992.

[4] M. Balaskó, A. Kuba, A. Nagy, Z. Kiss, L. Rodek, and L. Ruskó. Neutron-, gamma-, and X-ray three-dimensional computed tomography at the Budapest research reactor site. *Nuclear Instruments and Methods in Physics Research A,*, 542:22–27, 2005.

[5] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf. Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching. Technical Report 153, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, 1977.

[6] J. Baruchel, P. Merle, G. Peix, and E. Maire. *X-Ray Tomography in Material Science*. Kogan Page, 2000.

[7] Joost K. Batenburg. *Network Flow Algorithms for Discrete Tomography*. PhD thesis, University of Leiden, Netherlands, 2006.

[8] K. J. Batenburg. Reconstructing binary images from discrete X-rays, 2004.

[9] K. J. Batenburg. A new algorithm for 3D binary tomography. *Electronic Notes in Discrete Mathematics*, 20:247–261, 2005.

[10] Amir Beck and Marc Teboulle. Global Optimality Conditions for Quadratic Optimization Problems with Binary Constraints. *SIAM J. Optim.*, 11(1):179–188, July 2000.

[11] T. Beck. 3D-Reconstruction of geometrically limited objects with discrete tomography. Master's thesis, Friedrich-Alexander University Erlangen-Nürnberg, Lehrstuhl für Künstliche Intelligenz, Inst. für Mathem. Maschinen und Datenverarbeitung, 2002. (in german).

[12] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1995.

[13] Dimitris Bertsimas and Robert Weismantel. *Optimization over Integers*. Dynamic Ideas, Belmont, Massachusetts, ISBN: 0-9759146-2-6, May 2005.

[14] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813: SPG - software for convex-constrained optimization. *ACM Transactions on Mathematical Software*, 27:340–349, 2001.

[15] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.

[16] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):849–865, 1988.

[17] E. Boros and P.L. Hammer. Pseudo-Boolean Optimization. *Discrete Appl. Math.*, 123:155–225, 2002.

[18] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[19] Y. Boykov, O. Veksler, and R. Zabih. Markov Random Fields with Efficient Approximations. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[20] Yuri Boykov and Marie-Pierre Jolly. Interactive Organ Segmentation using Graph Cuts. In *Proc. of MICCAI'00, LNCS 1935*, pages 276–286, 2000.

[21] Yuri Boykov and Vladimir Kolmogorov. Computing Geodesics and Minimal Surfaces via Graph Cuts. In *Proc. European Conf. Computer Vision (ECCV)*, pages 232–248, 1998.

[22] Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. on PAMI*, 2002.

[23] Yuri Boykov and Olga Veksler. Fast Approximate Energy Minimization via Graph Cuts. *ICCV'99*, 1999.

[24] Yuri Boykov, Olga Veksler, and Ramin Zabih. A New Algorithm for Energy Minimization with Discontinuities. In *EMMCVPR '99: Proceedings of the Second International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 205–220, London, UK, 1999. Springer-Verlag.

[25] Ronald N. Bracewell. *The Fourier Transform and Its Applications*. New York: McGraw-Hill, 3 edition, 1999.

[26] Chad Brewbaker. Enumerating (0,1) Matrices Uniquely Reconstructable From Their Row and Column Sum Vectors, October 2003.

[27] R.A. Brualdi. Matrices of zeros and ones with fixed row and column sum vectors. *Lin. Algebra and Its Appl.*, 33:159–231, 1980.

[28] Richard A. Brualdi. Minimal nonnegative integral matrices and uniquely determined (0,1)-matrices. *Linear Algebera and its Application*, 341:351–356, 2002.

[29] Richard A. Brualdi and Geir Dahl. Matrices of zeros and ones with given line sums and a zero block. *Linear Algebra and its Application*, 371:191–207, 2003.

[30] Richard A. Brualdi and Suk-Geun Hwang. A Bruhat order for the class of (0,1)-matrices with row sum vector R and column sum vector s. *Electronic Journal of Linear Algebra*, 12:6–16, December 2004.

[31] Thorsten M. Buzug. *Einführung in die Computertomographie*. Springer, Berlin, 2004.

[32] T.D. Capricelli and P.L. Combettes. Parallel block-iterative reconstruction algorithms for binary tomography. *Electr. Notes in Discr. Math.*, 20:263–280, 2005.

[33] J. M. Carazo, C. O. Sorzano, E. Rietzel, R. Schröder, and R. Marabini. *Discrete Tomography. Foundations, Algorithms, and Applications (Eds.: Herman, G. T., Kuba, A.)*, chapter Discrete tomography in electron microscopy, pages 405–416. Birkhäuser, Boston, MA, 1999.

[34] Y. Censor. Binary steering in discrete tomography reconstruction with sequential and simultaneous iterative algorithms. *Lin. Algebra and its Appl.*, 339:111–124, 2001.

[35] Y. Censor, D. Gordon, and R. Gordon. Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems. *Parallel Computing*, 27:777–808, 2001.

[36] V. Cerny. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.

[37] B. Chalmond. *Modeling and Inverse Problems in Image Analysis*. Springer, Berlin, 2003.

[38] M. T. Chan, G. T. Herman, and E. Levitan. Bayesian image reconstruction using image-modeling Gibbs priors. *Int. J. Imag. Syst. Technol.*, 9:85–98, 1998.

[39] Shi-Kuo Chang. The reconstruction of binary patterns from their projections. *Communications of the ACM*, 14:21–25, 1971.

[40] Shi-Kuo Chang. Algorithm 445. Binary pattern reconstruction from projections. *Commun. ACM*, 16:185–186, 1973.

[41] Marek Chrobak and Christoph Dürr. Reconstructing Polyatomic Structures from Discrete X-Rays: NP-Completeness Proof for Three Atoms. *Lecture Notes in Computer Science*, 1450:185–193, 1998.

[42] G. Cimmino. Calcolo approssimato per soluzioni dei sistemi di equazioni lineari. *La Ricerca Scientifica XVI*, Series II, Anno IX 1:326–333, 1938.

[43] D. Colton, A.K. Louis, and H.W. Engl. *Surveys on Solution Methods for Inverse Problems*. Springer, Austria, 2000.

[44] Thomas H. Cormen, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill, Inc., New York, NY, USA, 1990.

[45] George B. Dantzig. Linear Programming. *Operations Research*, 50(1):42–47, 2002.

[46] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B,*, 39(1):1–38, 1977.

[47] Zdeněk Dostál and David Horák. Theoretically Supported Scalable FETI for Numerical Solution of Variational Inequalities. *SIAM J. on Num. Ana.*, 45:500–513, 2007.

[48] Zdeněk Dostál, David Horák, and Dan Stefanica. Quadratic Programming and Scalable Algorithms for Variational Inequalities. In A.B. de Castro, D. Gomez, P. Quintela, and P. Salgado, editors, *Numerical Mathematics and Advanced Applications - ENUMATH 2005*, pages 61–76. Springer Verlag, New York, 2006.

[49] Zdeněk Dostál and Joachim Schöberl. Minimizing quadratic functions over the non-negative cone with rate of convergence and finite termination. *SIAM J. Opt.*, 2002.

[50] P. Fishburn, P. Schwander, L. Shepp, and R. Vanderbei. The discrete radon transform and its approximate inversion via linear programming. *Discr. Appl. Math.*, 75:39–61, 1997.

[51] D. Gale. A theorem on flows in networks. *Pacific J. Math.*, 7:1073–1082, 1957.

[52] R. J. Gardner, P. Gritzmann, and D. Prangenberg. On the computational complexity of reconstructing lattice sets from their x-rays. *Discrete Math.*, 202(1-3):45–71, 1999.

[53] R. J. Gardner, P. Gritzmann, and D. Prangenberg. On the comptational complexity of determining polyatomic structures by X-rays. *Theor. Comput. Sci.*, 233(1-2):91–106, 2000.

[54] D.M. Gavrila. Multi-feature hierarchical template matching using distancetransforms. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 4399–444, 1998.

[55] D.M. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Proc. IEEE Conf. Computer Vision*, volume 1, pages 87–93, 1999.

[56] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(6):721–741, 1984.

[57] F. Giannessi and F. Niccolucci. Connections between nonlinear and integer programming problems. *Symposia Mathematica*, pages 161–176, 1976. Istituto Nazionale di Alta Mathematica.

[58] G. H. Golub and C. F. VanLoan. *Matrix Computations*. Johns Hopkins Univ. Pr., Baltimore, Md., 3 edition, 1997.

[59] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Addison Wesley, Reading, MA, 2 edition, 1992.

[60] D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. *J. Royal Statistical Soc.*, 51(2):271–279, 1989. Series B.

[61] P. Gritzmann, S. de Vries, and M. Wiegelmann. Approximating binary images from discrete X-rays. *SIAM J. Optimization*, 11(2):522–546, 2000.

[62] P. Gritzmann, D. Prangenberg, S. de Vries, and M. Wiegelmann. Success and failure of certain reconstruction and uniqueness algorithms in discrete tomography. *Int. J. Imag. Syst. Technol.*, 9:101–109, 1998.

[63] Chris Guy and Dominic Ffytche. *An Introduction to the Principles of Medical Imaging*. Imperial College Press, 2005.

[64] Jacques Hadamard. *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, Connecticut, 1923.

[65] W.K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57:97–109, 1970.

[66] G. T. Herman and A. Kuba, editors. *Discrete Tomography: Foundations, Algorithms, and Applications*. Birkhäuser Boston, 1999.

[67] Gabor T. Herman and Attila Kuba. *Advances in Discrete Tomography and Its Applications (Applied and Numerical Harmonic Analysis)*. Birkhauser, 2007.

[68] G.T. Herman and A. Kuba. Discrete tomography in medical imaging. *Proceedings of the IEEE*, 91(10):1612– 1626, Oct. 2003.

[69] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, 3 edition, 1996.

[70] G.N. Hounsfield. A method of and apparatus for examination of a body by radiation such as X-ray or gamma radiation. *Patent Specification 1283915, The Patent Office*, 1972.

[71] D. Huttenlocher, D. Klanderman, and A. Rucklige. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[72] H.M. Iyer and K. Hirahara. *Seismic Tomography: Theory and Practice*. Kluwer Academic Press, 1993.

[73] S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Acad. Pol. Sci. Lett. A35*, pages 355–357, 1937.

[74] Avinash C. Kak and Malcom Slaney. *Principles of Computerized Tomographic Imaging*. Applied Mathematics, SIAM, 2001.

[75] Jens Keuchel. *Image Partitioning based on Semidefinite Programming*. PhD thesis, University of Mannheim, Germany, 2004.

[76] Jens Keuchel. Multiclass Image Labeling with Semidefinite Programming. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, volume 3952 of *LNCS*, pages 454–467. Springer, 2006.

[77] S. Kim and M. Kojima. Second Order Cone Programming Relaxation of Nonconvex Quadratic Optimization Problems. *Optimization Methods and Software*, 15(3–4):201–224, 2001.

[78] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[79] C. Kisielowski, P. Schwander, F. Baumann, M. Seibt, Y. Kim, and A. Ourmazd. An approach to quantitative high-resolution transmission electron microscopy of crystalline materials. *Ultramicroscopy*, 58:131–155, 1995.

[80] J. M. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *IEEE Symp. Foundations of Comp. Science*, pages 14–23, 1999.

[81] Vladimir Kolmogorov and Carsten Rother. Minimizing Nonsubmodular Functions with Graph Cuts-A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1274–1279, 2007.

[82] Vladimir Kolmogorov and Ramin Zabih. What Energy Functions can be minimized via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

[83] Vladimir Kolmogorov and Ramin Zabih. Multi-camera Scene Reconstruction via Graph Cuts. In *In European Conference on Computer Vision (ECCV)*, May 2002.

[84] S. Krimmel, J. Stephan, J. Baumann, A. Nagy, A. Kuba, and Z. Kiss. Discrete Tomography for the Reconstruction of Single Material Objects in Non-Destructive Testing - A Simulation and Experimental Study. *Electr. Notes in Discr. Math.*, 20:455–474, 2005.

[85] A. Kuba, G.T. Herman, S. Matej, and A. Todd-Pokropek. Medical applications of discrete tomography. *Discrete Mathematical Problems with Medical Applications, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS*, 55:195–208, 2000.

[86] A. Kuba, L. Rodek, Z. Kiss, L. Ruskó, A. Nagy, and M. Balaskó. Discrete tomography in neutron radiography. *Nuclear Instruments and Methods in Physics Research A*, 542:376–382, April 2005.

[87] A. Kuba, L. Ruskó, L. Rodek, and Z. Kiss. Preliminary sudies of discrete tomography in neutron imaging. *IEEE Trans. Nucl. Sci.* **NS-52**, pages 380–385, 2005.

[88] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving Markov Random Fields using Second Order Cone Programming. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[89] L. Lauterbur. *Principles of Magnetic Resonance Imaging*. Wiley-IEEE Press, 1999.

[90] P.K. Lewin, J.E. Trogadis, and J.K. Stevens. Three-dimensional reconstruction from serial X-ray Tomography of an Egyptian Mummified Head. *Clin. Anat.*, 3:215–218, 1990.

[91] H. Y. Liao and G. T. Herman. Automated estimation of the parameters of Gibbs priors to be used in binary tomography. *Discrete Appl. Math.*, 139(1-3):149–170, 2004.

[92] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications, Special Issue on Linear Algebra in Control, Signals and Image Processing*, 284:193–228, November 1998.

[93] G. MacLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 1996.

[94] S. Matej, G. T. Herman, and A. Vardi. Binary tomography on the hexagonal grid using Gibbs priors. *Int. J. Imag. Syst. Technol.*, 9:126–131, 1998.

[95] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation

of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 21(6):1087–1092, 1953.

[96] C. Michelot. A Finite Algorithm for Finding the Projection of a Point onto the Canonical Simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.

[97] M. Muramatsu and T. Suzuki. A new second-order cone programming relaxation for max-cut problems. *Technical Report CS-01-01, UEC*, 2001.

[98] Antal Nagy and Attila Kuba. Reconstruction of binary matrices from fan-beam projections. *Acta Cybern.*, 17(2):359–383, 2005.

[99] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction*. SIAM, Philadelphia, 2001.

[100] Frank Natterer. Numerical Methods in Tomography. *Acta Numerica*, 1999.

[101] Mila Nikolova. Markovian Reconstruction Using a GNC Approach. *IEEE Transactions on Image Processing*, 8(9):1204–1220, September 1999.

[102] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2000.

[103] D. G. W. Onnasch and G. P. M. Prause. Heart Chamber Reconstruction from Biplane Angiography. In G. T. Herman and A. Kuba, editors, *Discrete Tomography: Foundations, Algorithms, and Applications*, pages 385–401. Birkhäuser, 1999.

[104] C. Pellot, A. Herment, M. Sigelle, P. Horain, H. Maître, and P. Peronneau. A 3D reconstruction of vascular structures from two X-ray angiograms using an adapted simulated annealing algorithm. *IEEE Trans. Med. Imag.*, 13:48–60, 1994.

[105] T. Pham Dinh and S. Elbernoussi. Duality in d.c. (difference of convex functions) optimization subgradient methods. In *Trends in Mathematical Optimization, Int. Series of Numer. Math.*, volume 84, pages 277–293. Birkhäuser Verlag, Basel, 1988.

[106] T. Pham Dinh and L. T. Hoai An. A d.c. optimization algorithm for solving the trust-region subproblem. *SIAM J. Optim.*, 8(2):476–505, 1998.

[107] M. Pincus. A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems. *Oper. Res.*, 18:1225–1228, 1970.

[108] G.P.M. Prause. *Binäre Rekonstruktion der dreidimensionalen dynamischen Ventrikelgeometrie aus biplanen angiokardiographischen Bildserien*. PhD thesis, Mathematisch-Naturwissenschaftliche Fakultät der Christian-Albrechts-Universität Kiel, 1994.

[109] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Berichte Sächsische Akademie der Wissenschaften, Math.-Phys. Kl., Leipzig*, 69:262–267, 1917.

[110] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In *International Conference on Computer Vision (ICCV'05)*, 2005.

[111] R. T. Rockafellar. *Convex analysis*. Princeton Univ. Press, Princeton, NJ, 2 edition, 1972.

[112] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems, 1998.

[113] Carsten Rother, Sanjiv Kumar, Vladimir Kolmogorov, and Andrew Blake. Digital Tapestry. In *CVPR'05*, 2005.

[114] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D*, 60:259–268, 1992.

[115] H.J. Ryser. Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.*, 9:371–377, 1957.

[116] Pablo M. Salzberg, Pedro I. Rivera-Vega, and Ariel Rodríguez. Network flow model for binary tomography on lattices. *International Journal of Imaging Systems and Technology*, 9(2-3):147–154, 1998.

[117] Walter J. Schempp. *Magnetic Resonance Imaging: Mathematical Foundations and Applications*. Wiley-Liss, September 1998.

[118] C. Schnörr, T. Schüle, and S. Weber. Variational Reconstruction with DC-Programming. In G. T. Herman and A. Kuba, editors, *Advances in Discrete Tomography and Its Applications*. Birkhäuser, Boston, 2006. To appear.

[119] T. Schoenemann and D. Cremers. Near Real-time Motion Segmentation using Graph Cuts. In *Pattern Recognition (Proc. DAGM)*, volume 4174 of *LNCS*, pages 455–464, Berlin, Germany, September 2006. Springer.

[120] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, June 1998.

[121] T. Schüle. Medical 3D–Reconstruction with LP-based Discrete Tomography. Master's thesis, University of Mannheim, Computer Vision, Graphics, and Pattern Recognition Group, February 2003.

[122] T. Schüle, C. Schnörr, S. Weber, and J. Hornegger. Discrete Tomography by Convex-Concave Regularization and D.C. Programming. *Discrete Applied Mathematics*, 151:229–243, 2005.

[123] T. Schüle, S. Weber, and C. Schnörr:. Adaptive Reconstruction of Discrete-Valued Objects from few Projections. *Electr. Notes in Discr. Math.*, 20:365–384, 2005.

[124] Thomas Schüle. *Large Scale Nonconvex Optimization in Discrete Tomography with Applications to Medical Imaging and Industrial Testing*. PhD thesis, University of Mannheim, 2008.

[125] P. Schwander, C. Kisielowski, F. Baumann, Y. Kim, and A. Ourmazd. Mapping projected potential, interfacial roughness, and composition in general crystalline solids by quantitative transmission electron microscopy. *Phys. Rev. Lett.*, 71:4150–4153, 1993.

[126] C.H. Slump and J.J. Gerbrands. A network flow approach to reconstruction of the left ventricle from two projections. *Comp. Graph. Im. Proc.*, 18:18–36, 1982.

[127] K. Tanabe. Projection method for solving a singular system. *Numer. Math.*, 17:203–214, 1971.

[128] A.N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math Dokl*, 4:1035–1038, 1963.

[129] Peter Toft. *The Radon Transform - Theory and Implementation*. PhD thesis, Dept. of Mathematical Modelling, Technical University of Denmark, 1996.

[130] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[131] Hans J.H. Tuenter. The Minimum $\mathcal{L}_2$-Distance Projection onto the Canonical Simplex: A Simple Algorithm. *Algo. Res. Quart.*, 4:53–55, 2001.

[132] S. Weber, A. Nagy, T. Schüle, C. Schnörr, and A. Kuba. A Benchmark Evaluation of Large-Scale Optimization Approaches to Binary Tomography. In *13th International Conference, DGCI 2006, LNCS, Szeged, Hungary*, volume 4245, pages 146–156, October 2006.

[133] S. Weber, C. Schnörr, and J. Hornegger. A linear programming relaxation for binary tomography with smoothness priors. In *Proc. Int. Workshop on Combinatorial Image Analysis (IWCIA'03)*, 2003. Palermo, Italy, May 14-16/2003.

[134] S. Weber, T. Schüle, and C. Schnörr. Prior Learning and Convex-Concave Regularization of Binary Tomography. *Electr. Notes in Discr. Math.*, 20:313–327, 2005.

[135] S. Weber, T. Schüle, C. Schnörr, and J. Hornegger. A Linear Programming Approach to Limited Angle 3D Reconstruction from DSA Projections. In T. Wittenberg et al., editor, *Bildverarbeitung für die Medizin 2003*, pages 41–45. Springer Verlag, March 2003.

[136] S. Weber, T. Schüle, C. Schnörr, and J. Hornegger. A Linear Programming Approach to Limited Angle 3D Reconstruction from DSA Projections. *Special Issue of Methods of Information in Medicine*, 4:320–326, 2004.

[137] H. Wolkowicz, Romesh Saigal, and Lieven Vandenberghe, editors. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, 2000.

[138] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1996.

[139] T. Yasuda, Y. Shigeki, and H. Ohshita. 3-D visualisation of an ancient Egyptian mummy. *IEEE Comp. Graph. App.*, 1992.

[140] E. Alper Yildirim and Stephen J. Wright. Warm-Start Strategies in Interior-Point Methods for Linear Programming. *SIAM J. on Optimization*, 12(3):782–810, 2002.

Index