Dissertation
submitted to the
Combined Faculties for the
Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

put forward by
Dipl.-Phys. Moritz Emanuel Schwarzer-Becker
born in Bruchsal, Germany
Date of oral exam: October 19th, 2016

Crowdsourced Interactive Computer Vision

Referees:
Prof. Dr. rer. nat. Bernd Jähne
Prof. Alessandro Bozzon

Abstract: In this thesis we address supervised algorithms and semi-manual working steps which are used for scenarios where automatic computer vision approaches cannot achieve desired results. In the first part we present a semi-automatic method to acquire depth maps for 2D-3D film conversions. Companies that deal with film conversions often rely on fully-manual working steps to ensure maximum control. As an alternative we discuss an approach which uses computer vision methods to reduce processing time but still provides opportunities to interactively control the outcome. As result we receive detailed, smooth and dense depth maps with sharp edges at discontinuities.

Part II, which presents the major contribution of this work, deals with human annotations used to assist ground truth acquisition for computer vision applications. To optimize this labour-intensive method, we analyse whether annotations created by different online crowds are an adequate alternative to running such projects with experts. For this purpose we propose different methods for improving acquired annotations. We show that appropriate annotation protocols run with laymen can achieve results comparable to those of experts. Since online crowds have much more users than typical expert groups used to run according projects, the presented approach is a viable alternative for large data acquisition projects.

Zusammenfassung: Diese Arbeit beschäftigt sich mit manuellen und computerunterstützten Arbeitsschritten, die in Situationen, in denen ein vollautomatischer Ansatz ungeeignet wäre, die Verwendung von Computer-Vision-Algorithmen ermöglichen oder verbessern. Im ersten Teil der Arbeit wird eine halbautomatische Methode für die 2D-3D Konvertierung von Filmen vorgestellt. Unternehmen, die sich mit Filmkonvertierungen beschäftigen, verwenden oft manuelle Arbeitsschritte um ihre Ergebnisse weitreichend beeinflussen zu können. Als Alternative untersuchen wir eine halbautomatische Methode, die geringere Bearbeitungszeiten anstrebt und dennoch interaktive Eingriffsmöglichkeiten gewährleistet, um wunschgerechte Ergebnise zu erzielen. Als Ergebnis erhalten wir detaillierte und dichte Tiefenkarten mit scharfen Kanten an Tiefen-Diskontinuitäten.

Der zweite, wesentliche Teil dieser Arbeit beschäftigt sich mit manuellen Annotationen, die für die Erstellung von Referenzdatensätzen für Computer-Vision-Anwendungen verwendet werden. Um diese zeitaufwändige Herangehensweise zu optimieren, untersuchen wir, ob wir mithilfe von verschiedenen Online-Benutzergruppen Ergebnisse erzielen können, die eine ernstzunehmende Alternative zu Expertenannotationen darstellen. Für diesen Zweck diskutieren wir verschiedenen Methoden, um die so gesammelten Annotationen zu verbessern. Wir zeigen, dass man mit geeigneten Arbeitsschritten und der Hilfe von Laien Ergebnisse erzielen kann, die in ihrer Qualität mit denen von Experten vergleichbar sind. Da es im Internet sehr viele Benutzergruppen gibt, die für diesen Ansatz geeignet und aufgeschlossen sind, stellt diese Methode eine sinnvolle Möglichkeit dar, große Bilddatensätze zu bearbeiten.

For Greta

Acknowledgements

First I would like to thank my supervisor Prof. Dr. Bernd Jähne for the opportunity and the fruitful conditions to conduct my work on this topic. I am very grateful for an inspiring time at the *Heidelberg Collaboratory for Image Processing* (HCI).

I especially thank Dr. Daniel Kondermann for many nice and constructive discussions. In challenging periods his confidence and inspiration helped me a lot.

Furthermore, I thank all my colleagues, especially Dr. Stephan Meister, Dr. Rahul Nair, Dr. Henrik Schäfer, Katrin Honauer, Burkard Güssefeld and Jens-Malte Gottfried, for their scientific and technical support and for the pleasant atmosphere at the *Heidelberg Collaboratory for Image Processing.*

I thank the *Ministry of Economy Baden-Württemberg* and the companies *Pixomondo*, *Screen-Plane*, *Unexpected*, *Bewegte Bilder* and *Tridelity* for funding parts of my work in the research project *"Entwicklung von Systemen und Methoden zur effektiven Erstellung und Bearbeitung stereoskopischer Inhalte"*. I also give thanks to the *Filmakademie Baden-Württemberg, Institute of Animation* which jointly run this project with the HCI.

My acknowledgement also goes to the *Pallas Ludens GmbH*, which gave me the opportunity to apply my research to crowdsourcing scenarios. Therefore I would like to thank especially Jonas Andrulis, Dr. Daniel Kondermann, Dr. Stephan Meister, Stefano Ligas, Julian Coordts and Kira Monno for their support and for contributing to my experiments as experts. I am looking forward for many more years of pleasant joint work.

Finally I am very grateful for the positive support of my friends and my family. I especially thank my dear Constanze and my newborn daughter Greta for their encouragement, their patience and their emotional support.

Contents

1 | Preliminary Remarks

Computer vision enables technologies which have reached almost any industrial branch and which, presumably, will shape our everyday life increasingly. The automotive industry develops extensive, vision-based driver assistance systems. Once fiction, even autonomous vehicles finally come true [29]. Furthermore, computer vision is used for computer-aided quality inspections which are a standard in many industrial production chains such as the electronic industry, the food industry or the textile industry [39, 11, 51]. Another field of application is medical imaging. Medical imaging has become indispensable for many clinical examinations and medical interventions [14]. In addition to that, recognition methods are used for visual surveillance applications which enable visual access controls, human identification or traffic controls [47, 42]. With regard to the creative industry, there have been great benefits from algorithms that enable visual effects or 3D-movie productions [81]. Other current vision-based trends are augmented and virtual reality applications [4]. Examples are the *Microsoft HoloLens* project[1] or the *Oculus Rift* virtual reality glasses[2].

In such end-use implementations, algorithms usually run automatically. However, there are still applications that either technically require supervision or for which supervision is desired. These approaches can achieve superior results in cases where automatic algorithms perform poorly due to inappropriate image contents. For instance, issues such as motion blur, translucent objects or small particles require a deeper understanding to be adequately addressed. Human perception capabilities can provide the required knowledge.With regard to the important measures **quality**, **scalability** and **costs** however, supervision causes high costs and does not scale well.

In this thesis we address supervised algorithms and semi-manual working steps which are used for scenarios where automatic approaches cannot achieve desired results. The contents are divided into two parts.

In **Part I** we address application scenarios, that often rely on fully-manual working

[1] https://www.microsoft.com/microsoft-hololens

[2] https://www.oculus.com

steps to ensure maximum control. As an alternative we discuss semi-automatic approaches which use computer vision methods to reduce processing time but still provide opportunities to interactively control the outcome.
We discuss the capabilities of this approach by applying it to the application of 2D-3D conversions of films. Many working steps in the creative industry, in particular 2D-3D conversions, are extremely labour-intensive. Finding new ways to reduce user interactions can have a huge impact on costs, efficiency and competitiveness. Initially this approach was proposed in our paper *"Movie Dimensionalization via Sparse User Annotations"* [5] in 2013. Most of the contents of Part I are based on this paper.

Part II presents the major contribution of this thesis. In contrast to Part I, where we discuss how to simplify formerly manual working steps by using interactive computer vision, Part II deals with cases where manual input is consciously used to assist computer vision applications. When the required manual input cannot be reduced any further, there is only one way to improve efficiency: more workers. For instance, in the film industry hundreds of artists work on labour-intensive projects. Reconsidering this, the question arises, whether the complexity of manual interactions can be broken down thus far that everybody can participate. If this was the case and we could rely on even more workers, labour-intensive semi-automatic approaches could be more efficient and provide a better scalability.
Nowadays, in the era of the internet, it is common that vast groups of people contribute to various opportunities. One of these opportunities is called ***crowdsourcing***, which *"is the act of taking a job [...] and outsourcing it to an undefined, generally large group of people [...]"*[3]. Crowdsourcing promises the opportunity to work with many people at once. Thus, we analyse whether its capability is sufficient to further increase the efficiency of supervised computer vision applications.
An important research topic at the *Heidelberg Collaboratory for Image Processing* is the creation of *ground truth* for performance analysis. Since for that purpose human annotations can assist computer vision and measurement methods, we choose this application to evaluate crowd capabilities in Part II.

[3] http://crowdsourcing.typepad.com/

Part I

Interactive Computer Vision

2 | Introduction

In this thesis we address computer vision applications which utilize manual user interactions to succeed. The major drawback of such applications is the required time for manual working steps. The efficiency can be optimized by reducing the required interactions or by choosing workflows which are parallelisable and can be addressed by many people at once. In this first part of the thesis we primarily focus on an interactive approach that aims at reducing the necessary manual effort by relying on semi-automatic algorithms.
A field of application which demands such interactive approaches is given by the creative industry. In this case, the reason for that demand is not only caused by insufficient quality of automatic algorithms but also by the artistic application which requires maximum control about the outcome. Though, there are creative applications which do not need that much artistic freedom but which require plenty of manpower to succeed. One of these applications is 2D-3D film conversion.
To perform a satisfying 2D-3D film conversion, plausibly reconstructed depths are indispensable. So-called depth maps which assign depth to every pixel of an image cannot be produced by fully automatized pipelines for now. The typical issues that let automatic approaches fail are object occlusions, small and filigree particles such as snow and rain as well as moving and semitransparent objects. Though the film industry strongly demands high-quality depth maps. Since film scenes can be arbitrarily complex and typically contain many of such difficult objects, user interaction is needed to reliably convert 2D footage. Beside these technical obstacles to convert 2D footage, artists mostly cannot rely on fully automatized algorithms that preclude further adjustments. As for many applications in the creative industry, most steps require supervision. In this case, depth maps often are changed to alter the depth perception to support a specific dramatic composition. Further processes that mostly are performed manually are *rotoscoping* (creating masks for objects), stereo view generation and accordingly completing images of converted frames at areas which had been occluded in the initial source frame (*inpainting*).

Most companies that work on 2D-3D conversion projects rely on costly and sophisticated workflows. For instance, 400 artists were involved in converting the

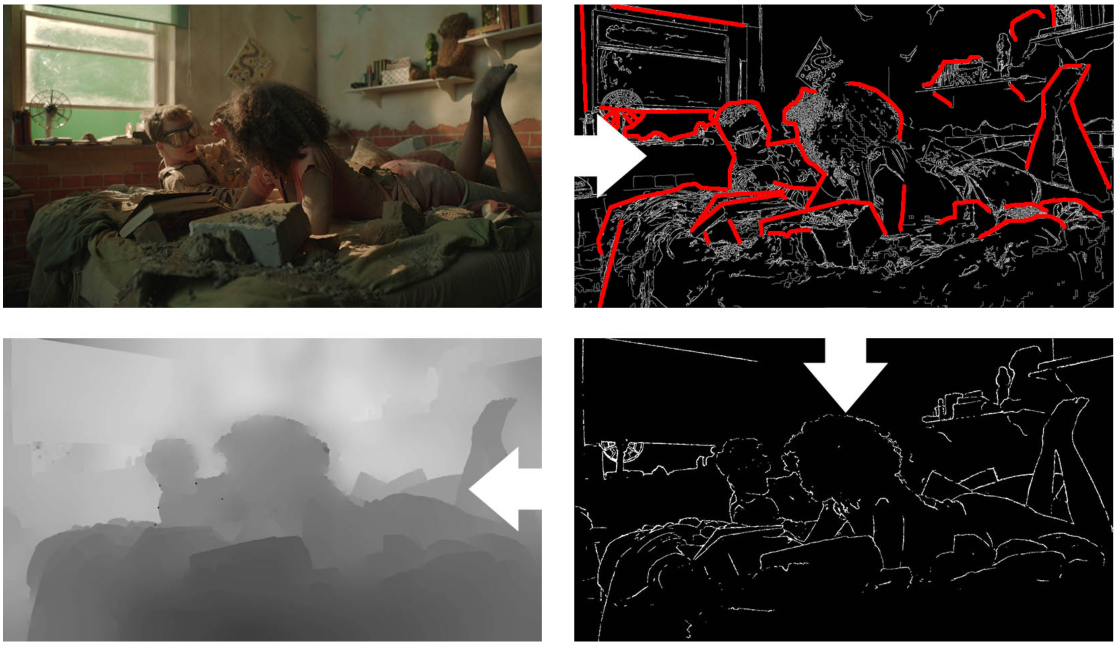


Figure 2.1: *Depth discontinuities are labelled, learned by a random forest and afterwards predicted for several subsequent frames. We use those discontinuity edges as interpolation regularization. Thus, depth maps can be interpolated from sparse depth information. As result, our depth maps are smooth with sharp edges only at discontinuities.*

film *Titanic 3D*[1]. The competitiveness of conversion companies that aim at high quality results is affected by costs and the overall processing time that is needed to convert a film. Thus, conversion tools should aim at reducing processing time while producing comparable quality of results.

In this part we discuss a conversion approach which we initially proposed in Becker et al. [5]. Most of the contents of this part are based on that paper. Our workflow relies on a learning based approach to predict depth discontinuities, structure from motion to automatically retrieve depth information and a variational method for depth inpainting. Unlike more commonly used segmentation based methods, the advantage of our approach is that the outcome of our inpainting process directly provides smooth depth maps with sharp edges *only at artistically relevant* depth discontinuities. We do not have to take care of segment boundaries which lie at continuous depth junctions and lead to depth offsets which require further user interactions.

[1] www.fxguide.com/featured/art-of-stereo-conversion-2d-to-3d-2012/

2.1 Related Work

Our approach is based on annotating and detecting depth discontinuities semi-automatically. Many approaches have been proposed to address this topic fully automatically for example for segmentation[86, 33, 69] optical flow[101], stereo estimation[36], (video or super-pixel), masking[35] and in principle any computer vision approach that aims at producing dense results. A common approach is to rely on total variation regularizers. In this part we focus on a semi-automatic approach to acquire depth edges. The general idea can be applied to all of these previous methods and can increase their accuracy.

An application which strongly demands accurate information about depth discontinuities is film dimensionalization. Converting a film with imprecise depths especially at boundary regions of objects would be inadequate. Several related approaches to address 2D-3D conversions have been proposed.

Sparse reconstruction methods that rely on structure from motion were proposed by tools such as VisualSFM[100] and Phototourism[78]. These approaches rely on a moving camera which capture a scene from different perspectives. As a consequence, this approach cannot succeed on footage captured with a static camera. In addition to that, moving objects cannot be addressed adequately. The outcome of these algorithms are sparse 3D point clouds.
In the case of available sparse motion reconstructions, multiview stereo approaches can be used to densify them (cf. e.g. [30]).

A different approach to reconstruct depth was proposed by Saxena et al. [73]. They rely a fully-automatic learning-based algorithm which assigns depths to single still images only. An similar approach which however addresses videos was proposed by Karsch et al. [44]. To predict depths for a new video, they rely on an existing image database which also keeps depth values for the according images. Then they choose the most appropriate candidates from this database to warp the according depth to the new video. They also perform a motion analysis to address moving objects.

For real films, these fully-automatic conversions mostly do not achieve qualitative results for difficult conditions such as independently moving objects of motion blur. However, automatic approaches can be used, when a scene is highly constrained and carefully set up. An example is the *Trinity* scene in *The Matrix*[2](1999) where multiple cameras and optical flow were used to achieve a slow-motion picture of the protagonist while the viewer's perspective circled around this person.

One of the first authors addressing dimensionalization was Guttmann et al. [34]. They proposed a semi-automatic approach which relies on user scribbles, which is

[2] http://www.matrixeyewear.com/blog/breaking-down-the-special-effects-of-the-matrix

similar to our workflow to acquire depth cues. Another similar tool is *Depth Director* which was presented by Ward et al. [96]. *Depth Director* converts footage by using segmentations and user scribbles. In addition to that, sparse depth cues are used to assign depth to these segments. In contrast to that approach, we focus on depth edge annotations instead of segmentations and use a subsequent variational depth interpolation for depth cues from *both*, structure from motion and user scribbles.

Looking at end-use applications for film conversions, most tools are commercial and closed source. To name some of these, there are low budget applications such as *TriDef 3D*, *Unitypro* or *Movavi* and professional applications as *Nuke*, *Ocular*, *Yuvsoft 2D-to-3D Conversion.* These frameworks typically provide interactive tools for example to create segmentations and to refine results. Depths typically are reconstructed from motion or focus. After creating depth maps, these tools also are used to create stereo views.
Professional Companies that create conversions for big film productions mostly develop their own tools and plugins. Typical working steps are masking, key framing, assigning depths and removing image artefacts. Prominent companies are *Legend 3D*[3] and *Prime Focus*[4].

[3] http://legend3d.com/

[4] http://www.primefocusworld.com/

3 | Interactive 2D-3D Conversion

Our approach is structured as follows. We initially annotate depth edges in keyframes and use this information to train a classification model. Then we use this model to predict depth edges for the complete image sequence. By using this model for predictions, we aim at reducing the working time to process images manually. An outline is given in Figure 3.1. Secondly, depth cues are determined using structure from motion. Alternatively, or when structure from motion is not applicable due to a static scene or many moving objects, depth cues can be acquired by user scribbles. As a last step, depth cues are interpolated to retrieve smooth depths for the complete image. To ensure accurate and pleasing depth discontinuities, the predicted depth edges are used to regularize this interpolation process.

Film dimensionalization requires detailed depth maps for objects lying at focus of attention while minor parts of the footage can be simplified. As mentioned in the introduction, an adequate conversion tool should rely on a minimal amount of user interaction to reduce processing time while the resulting conversion quality should not suffer.
Our approach requires rough user scribbles for depth edge annotations in keyframes. In contrast to a fully-manual conversion where depths are assigned to each pixel by hand, this should be an easier and faster workflow.

In this chapter we discuss the individual working steps or our approach.

Contents of this chapter

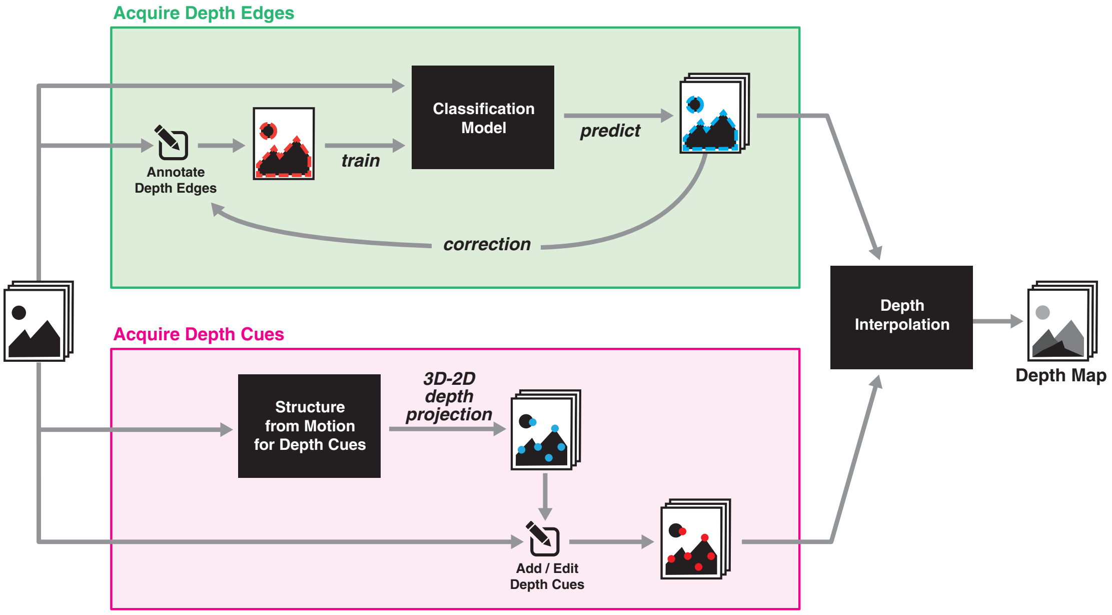


Figure 3.1: *This figure shows an outline for our approach. First, depth edges are acquired by and interactive learning-based approach. In addition to this, depth cues are collected either by a backprojection of structure from motion point clouds or manually. Afterwards, these depth cues are interpolated using the depth edges as regularization.*

3.1 Depth Edges

We aim at plausible depth maps as the overall outcome of our workflow. To adequately perform the interpolation (3.3) of sparse depth cues (3.2), we need to know the regions of depth discontinuities. Depth cues should have sharp transitions at these depth edge regions.

In Figure 3.2 we visualize two types of depth edges. The red circle in 3.2a) shows a C^0 edge, while the same location appears as a C^{-1} edge in 3.2b). We are primarily interested in the later C^{-1} edges which represent depth gaps. Though, our learning-based approach is independent from the actual reason of an edge and could be trained to detect any edge.

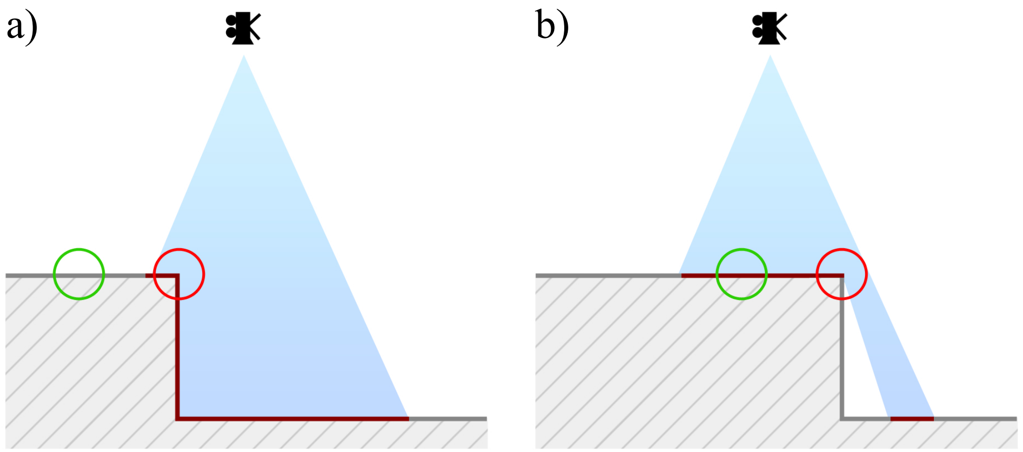


Figure 3.2: *We acquire depth edges which are marked by red circles in this figure. While the kink in Figure a) is a C^0 discontinuity, we basically utilize C^{-1} edges such as in b) which forces a gap between depth values. The green circles mark areas where algorithms can detect texture edges at most.*

3.1.1 Annotating Depth Edges

We use the open source framework *Frapper*, which is provided by the Filmakademie Baden-Württemberg[1], for the annotation process of depth edges. Frapper was already used in movie productions and is constantly improved by the R&D group at the Filmakademie. To ensure a fast and easy annotation process, we use automatically determined canny edges to refine the annotated user scribbles. Roughly annotated edges are optimized by taking the intersection of annotated edge and canny edge. To make sure that annotated areas always contain canny edges, we choose parameters that maximize the amount of adequate canny edges.

Once the annotation step for a single keyframe is completed, we need to propagate these edges to subsequent frames. Since the propagation has to be precise, this process is restricted to reliable methods. Typical methods for motion estimation such as optical flow especially produce imprecise results at these depth discontinuities, as can seen in Figure 3.4. For this reason, we choose a learning-based approach which we introduce in the next section.

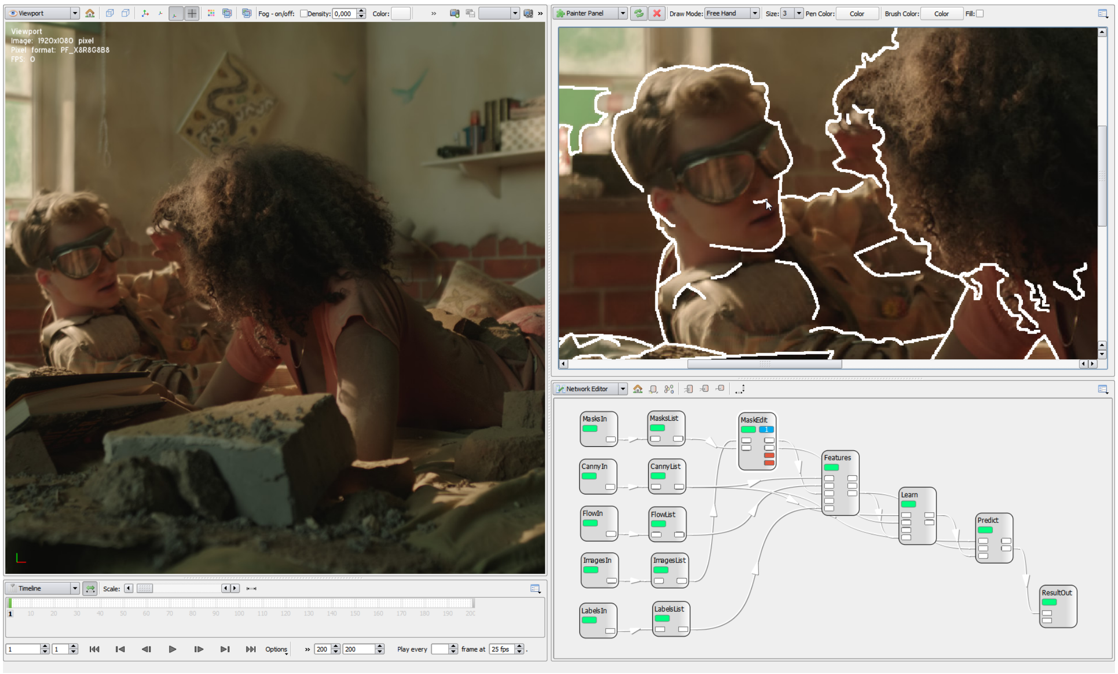


Figure 3.3: *We used an interactive framework which can be supplied with different modules.*

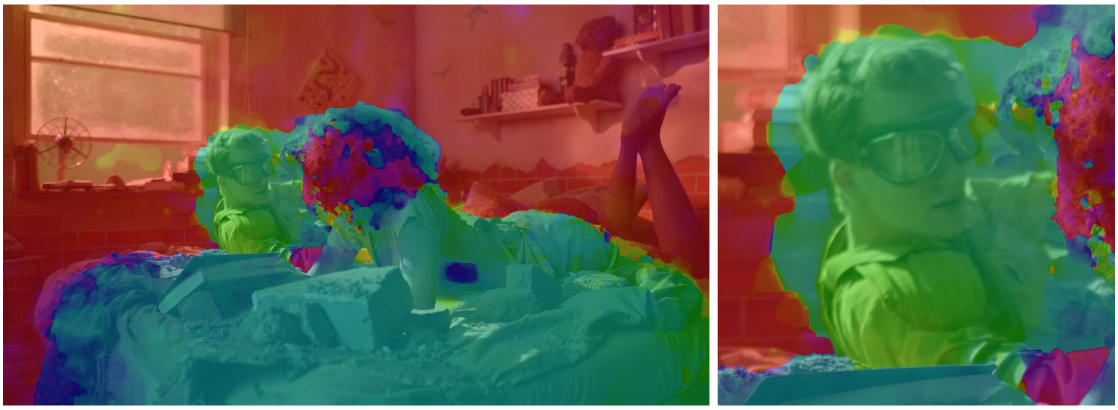


Figure 3.4: *This figure shows a state-of-the-art optical flow result ([83]). Optical flow especially performs poorly at depth discontinuities. The colors in this figure represent the measured flow direction.*

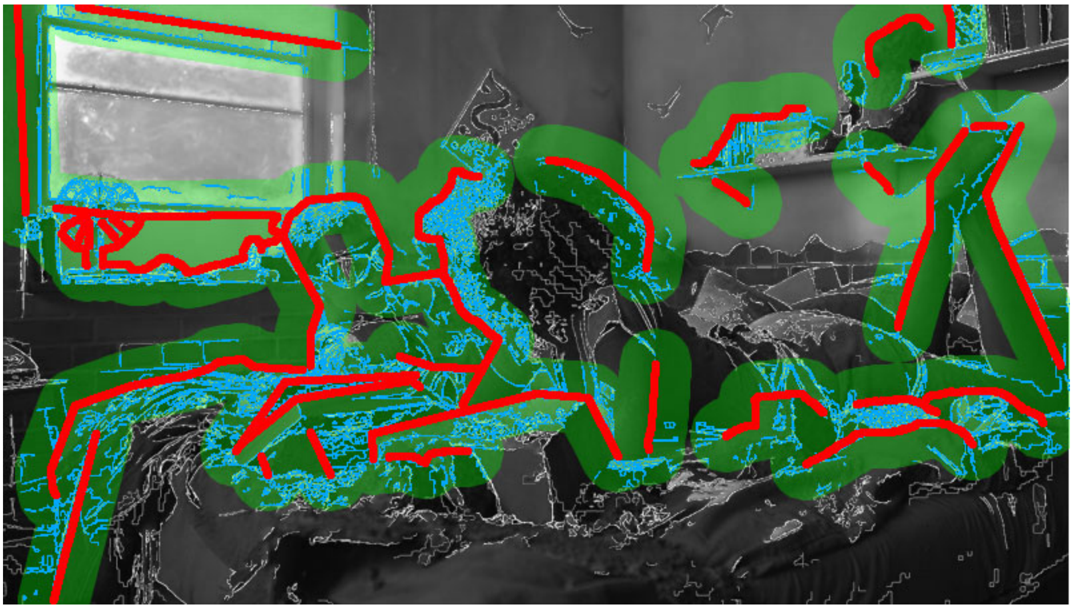


Figure 3.5: *We restrict learning samples to depth edge neighbourhoods which are visualized in green. Pixels within this area are classified as "no edge", "texture edge" (blue) and "depth edge" (red).*

3.1.2 Learning Depth Edges

We use a random forest to learn the annotated depth edges. Therefore we choose sample stratification in each tree to address imbalanced classification samples in the learning set. We classify pixels as *depth edges*, *texture edges* and *no edges*. In addition to that, we restrict the image area from which we choose our samples to the direct neighbourhood of the annotated depth edges. Figure 3.5 visualizes this area in green. As a consequence, we also need to restrict the area which is predicted. Therefore, we use optical flow to roughly estimate this neighbourhood in subsequent frames.

This approach takes advantage of the fact, that subsequent frames are very similar. Instead of learning a general model which can be applied to any image scene, we choose a model that is based specifically on a single shot.

Table 3.1 lists the features used to train the model. We choose color features and edge features. In addition to that, features are mostly calculated for all channels of an HSV image separately and provided for different scale spaces. To address different scale spaces we blur with $sigma = 1, 3, 9$.

Our feature set contains features that are more general and features that are quite

[1] http://research.animationsinstitut.de/frapper/

Edge Cue Descriptions	**Num**
Color cues	**32**
C1. Colors channels	6
C2. Neighbor colors next to putative edge	6
C3. Color Histograms	12
C4. Mean Color of neighbor segments	6
Edge cues	**79**
E1. Color gradient	4
E2. Histogram of gradients	5
E3. Eigen values	4
E4. Histogram of eigen values	20
E5. Ratio between eigen values	4
E6. Eigen values and ratios of neighbor pixels	32
E7. Hessian	3
E8. Diffusion tensor	3
E9. Bilateral filter	3
E10. Gradient of optical flow	1

Table 3.1: *This table lists our features used to train the random forest.*

specific. Especially the color-related features $C1, C2, C3, C4$ are very specific since colors change between scenes. $C1$ keeps the color of the sample itself, while $C2$, $C3$ and $C4$ contain color information of the sample neighbourhood. These neighbourhoods are chosen with respect to eigen vector directions (determined from structure tensor). By doing so, the neighbourhood of edge samples should be determined perpendicular to this edge. Since colors often change at depth edges, the combination of color and edge features seams to be a reasonable approach.

Since the depth edge prediction is performed pixel-wise, resulting depth edges can be incomplete. False positive and negative predictions can be adjusted afterwards to recalculate the model. In this way we can pay particular attention to difficult scenes where predictions will not satisfy our requirements.

3.1.3 Refining Depth Edges

Since the random forest outputs probabilities, we need to threshold these to receive binary values for our depth edge mask. While a low threshold ensures to include as many depth edges as possible, we also detect falsely classified edge pixels. In our experience, accepting false positives over false negatives leads to more appealing results since interpolation step, which is presented in 3.3 compensates single falsely detected depth edge pixels. Missing depth edge pixels on the other hand cause depth interpolation at areas where we expect sharp transitions in depth. We address

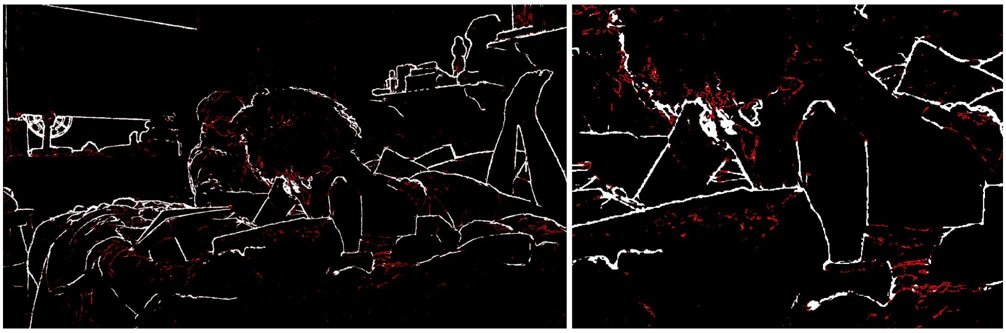


Figure 3.6: *Since predictions may not directly satisfy our demand, we can adjust predicted edges and recalculate the model. The red pixels are removed while white edges were extended as discussed in section 3.1.3.*

these broken edges by running a hysteresis process on the probability distribution. This approach can be compared to the canny algorithm. Pixels with a predefined minimum probability can be enhanced if they have a minimum amount of depth edge and canny edge neighbour pixels.

Since discrete depth edges retrieved by threshold mostly have strengths of more than one pixel, an intersection with canny edge pixels is used to refine edges. As a last step of refinement, depth edges are completed at pixel inclusions. An example of these refinements is illustrated in Figure 3.6.

3.2 Depth Cues from Motion

To automatically determine sparse depth cues, we use the structure from motion tool *VisualSFM*[100]. Its outcome are sparse 3D point clouds and according camera extrinsics. Since the reconstructed points are based on SIFT features, those points mostly represent textured image areas. However, if the scene contains image regions with less texture, we may not receive sufficient points at these areas. For this reason, we additionally use dense optical flow correspondences to reconstruct a dense point cloud. By 3D-2D projection of this point cloud to every view, we retrieve depth values for according 2D coordinates.

If the image sequence is not adequate to be processed by structure from motion algorithms (e.g. due to missing camera motion or moving objects), we need to use user scribbles to assign depth by hand. These scribbles can also be used, when automatic depth cues are not available for specific image areas only. For example, if objects move within a scene, structure from motion can reconstruct the environment but cannot achieve reliable depths for those parts that moved. In this case, user scribbles can be used to fill in missing depths.

3.3 Depth Interpolation

Suppose we have acquired all depth edges which mark depth discontinuities and we have collected depth cues either by structure from motion or by user scribbles. Now we interpolate these sparse depth cues to receive a dense depth map. To interpolate the sparse depths $d(\vec{x}_1), ..., d(\vec{x}_K)$ obtained at the positions $\vec{x}_1, ..., \vec{x}_K$, we use a variational approach, which minimizes an energy functional globally over the whole image range Ω with respect to the sought dense depth map $\hat{d}(\vec{x})$. This energy functional consists of a data term E_d which ensures matching of sparse depths against the dense depth map and a λ-weighted prior term E_p which imposes a smoothness constraint on the resulting depth map.

$$E_d = \sum_{k=1}^{K} p_k \cdot \Phi\left(\left(\hat{d}(\vec{x}_k) - d(\vec{x}_k)\right)^2\right), \tag{3.1}$$

$$E_p = \int_{\Omega} \Phi\left(\|\vec{\nabla}\hat{d}(\vec{x})\|_2^2\right)\, d\vec{x}, \tag{3.2}$$

We choose $\lambda = 400$. Φ denotes a suitable penalty function which ensures sharp borders and smooth areas. We use the Charbonnier penalty function $\Phi(\Delta^2) = \sqrt{\Delta^2 + \varepsilon^2}$, parameterized with $\varepsilon = 0.01$. This represents a differentiable approximation of the l_1 norm [16].

We choose a relative probability measure p_k as a weighting factor within the data term. We use p_k to address uncovering and covering regions of the depth map and to obtain temporal consistency. The uncovering and covering of regions implies depth changes, whereas the depth change of a single region can be assumed as smooth over time. Since this factor has to express time-wise consistency, it depends on the previous dense depth map $\hat{d}_{t-1}$ and the current sparse depths d_t.

$$p_k = \exp\left(-\frac{\left(d_{t+1}(\vec{x}_k) - \hat{d}_t(\vec{x}_k)\right)^2}{2\sigma^2}\right), \sigma = 1000 \tag{3.3}$$

To ensure a proper implementation of the prior term denoted by equation (3.2), an adequate discretization of the partial derivatives of the ∇ operator is needed. Therefore we take all pairwise differences between the currently considered central position $\vec{x}$ and all horizontal and vertical neighbour positions $\mathcal{N}(\vec{x})$. Similarly we address image and depth boundaries by excluding them from this neighbourhood.

To optimize the energy term we set up the Euler Lagrange equations $E = E_d + \lambda \cdot E_p$.

4 | Experiments and Results

We divide the evaluation into two sections. First we evaluate depth edges by comparing the random forest predictions to manually annotated edges. Afterwards, we run the complete proposed workflow and create depth maps, which are discussed qualitatively.

4.1 Depth Edges

For a better understanding of the importance of chosen features (Table 3.1) we analyse the variable importance measure. This importance can vary from scene to scene especially for color-based features. The most important features however are edge-based features. According to Figure 4.1, eigen values of the structure tensor (E3) and ratios between first and second eigenvalue (E6) are the most important features with regard to our test footage. While this could have been expected since these values especially describe edges, color-based features are of importance too. Colors of neighbourhood pixels (C2) for example are the fourth most important features.

To evaluate predicted depth edges we compare our results with ground truth annotated by hand. Only pixels of ground truth and prediction which intersected with canny edges were considered for analysis to prevent user induced inaccuracies. Depth edges were annotated *only in frame* 1. Then, frames $\{1, 5, 25\}$ were predicted. Figure 4.2 and 4.3 visualize these comparisons for frame 25 of each scene. Edges in presented Figures were enlarged for visualization purposes. Green edges represent positive detected edges while red false negative pixels could not be detected and blue false positive pixels were wrongly approved to be depth edges. As discussed in section 3.1.3, blue edges do not influence results significantly while missed red pixels will lead to edge bleeding. False negative predicted edges most often occur on low contrast areas such as shadows or motion blur. Since this is a general problem in computer vision which cannot be arbitrarily improved by automated algorithms, additional user annotations be will required for these parts. Of course, significant changes within the footage also lead to a weak detection rates

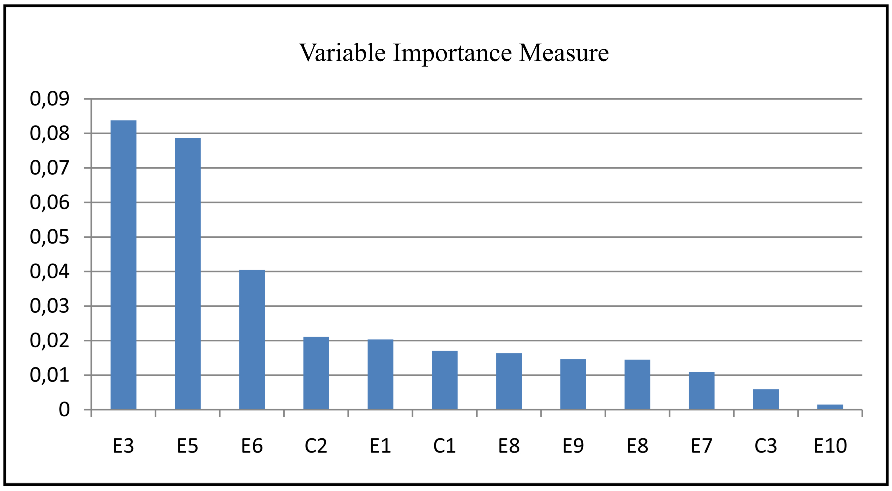


Figure 4.1: *According to the variable importance measure, the most relevant features are eigen values and the ratio between both eigen values.*

#	TotDE	DEp	DEfp	DEfn
Scene 1: Bed				
1	39894	37321 (94%)	3159	2573 (6%)
5	40830	35367 (87%)	8216	5463 (13%)
25	35385	29335 (83%)	10140	6050 (17%)
Scene 2: Frontal				
1	19349	19324 (100%)	1434	25 (0%)
5	20088	18013 (90%)	859	2075 (10%)
25	20134	17255 (86%)	966	2879 (14%)

Table 4.1: *We labeled "ground truth" by hand and evaluated predicted depth edges. Ground truth of frame 1 was taken as RF labels, while predicted edges in frames 1, 5 and 25 where taken for evaluation. The second column (TotDE) contains the number of total ground truth depth edges. (DEp) = positive detected depth edge pixels, (DEfp) = false positive, (DEfn) = false negative. While progressing through the scene, the prediction reliability decreases. Footage of scene 1 can be found in Figure 4.2 while frames of scene 2 are visualized in Figure 4.3.*

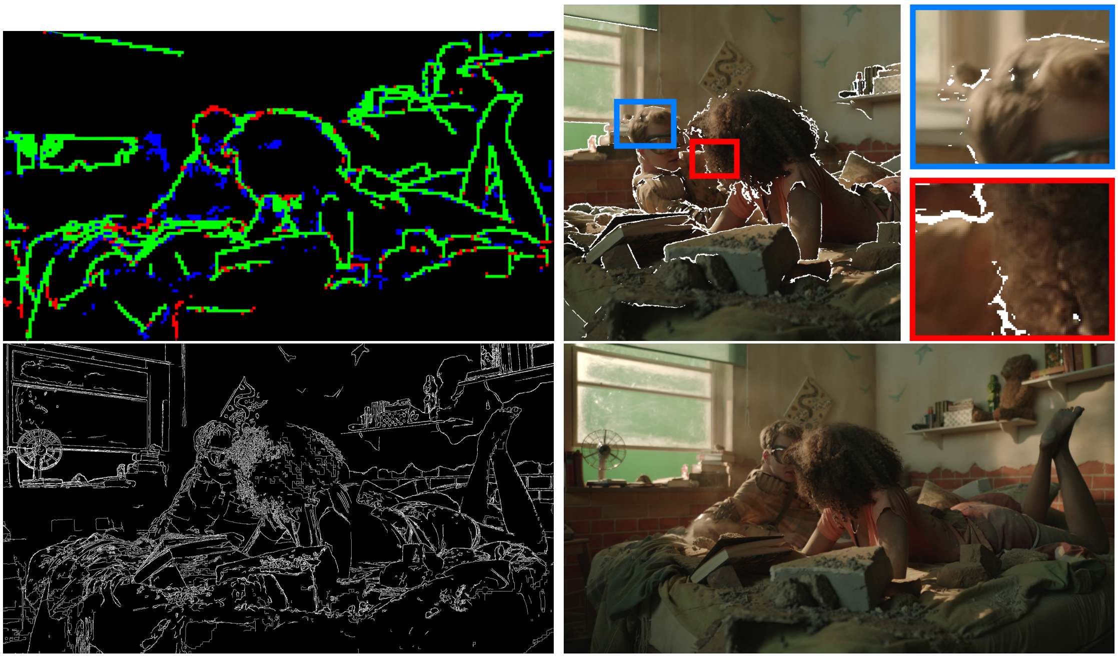


Figure 4.2: *Evaluated Edges of scene 1, frame 25 according to last row of table 4.1. Pixels were enlarged for visualization purposes. Positive detected depth edge pixels are shown in green, false positive in blue and false negative in red. The red false negative pixels are crucial for our conversion quality while blue false positive predicted pixels have only little relevance for this application. The lower loft image shows canny edges which contain all kind of edges. As can be seen, most depth edges are reliably predicted. As could be expected, processing low-contrast pixels is one of the drawbacks which can be seen on the zoomed in lower right image. White edges which should be connected are broken at both head boundaries.*

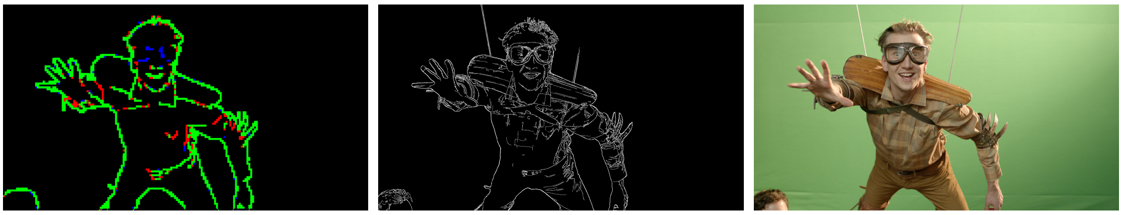


Figure 4.3: *Evaluation of scene 2. While low contrast in Figure 4.2 leads to low prediction rates, this greenscreen scene provides much better conditions. Discontinuities are also detected in the interior of the body while texture edges are widely removed.*

whenever according areas could not been labeled on initial key frames. To counteract this circumstances, additional labeling can be applied to those edges within our user interface to relearn the random forest.

Table 4.1 presents numbers of detected depth edge pixels for frames $\{1, 5, 25\}$. The last row corresponds to Figure 4.2 and 4.3. As discussed, the last column contains false negative values which are most relevant for our conversion application. Detection rates decrease over time. The step size of frames for which additional labeling is necessary depends on the increasing false negative rate. Hence, to generate depth maps in the next section, we process every 25th frame which seemed to be proper choice. Although a thorough analysis of cost savings remains a topic for future research this factor of 25 clearly shows the high potential of our approach to save time without loosing much of the quality.

4.2 Depth Maps

The depth maps are created by depth edge labels for frames $\{1, 25\}$. Since scene 1 contains camera motion, we make use of this additional information. Depth cues are retrieved from structure from motion point clouds. Therefore we use *VisualSFM* [100] for camera tracking. For homogeneous distributed depth cues, we use correspondences converted from state-of-the-art optical flow [83]. Due to moving bodies within this scene, regions of these movements can not be adequately reconstructed by structure from motion. Thus, we use additional manual depth cues for these areas.

The processed frames ($\{1, 5, 10, 15, 20\}$) are shown in Figure 4.4 Proper depth maps are retrieved in the foreground. While the third row of Figure 4.4 shows results based on structure from motion depth cues, the fourth row contains depth maps for which we additionally assigned user scribbles.

Conversion results of scene 2 can be seen in Figure 4.6. Depth maps are created only by user scribbles due to missing camera motion. Undetected depth edges discussed in Figure 4.3 affect our depth map interpolation at the head boundary. Missing pixels lead to color bleeding. False positive pixels on the face (compare blue pixels in Figure 4.3) do not influence the result.

Comparison to other depth contour annotations

In addition to our interpolation step using depth edges, we also compare results retrieved from using alternate regularization masks which are edges from superpixels and canny edges. We use the same depth cues for all of these results. Figure 4.5

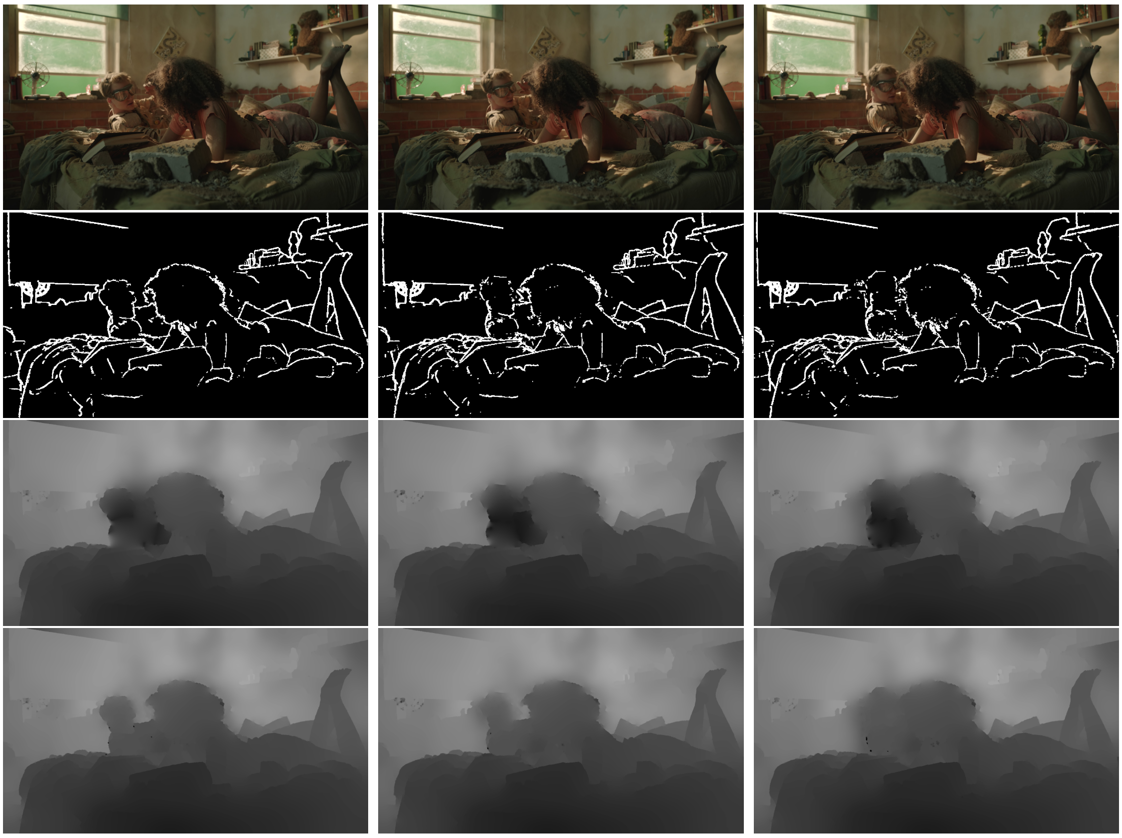


Figure 4.4: *Converted sequence by depth label annotations for frame 1 and 25. Sparse depth cues were taken from structure from motion. The third line shows results received only by automated cues while in the fourth row additional user scribbles were used to correct the depth of the moving person.*

shows the depth map comparison. We highlight falsely generated depth discontinuities with red boxes. Incomplete edges falsely causing smooth depth transitions are highlighted with blue boxes. Since superpixels consist of closed segments, these results do not struggle with incomplete edges. However, due to this fact, smooth depth transitions would only be possible by merging segments which will leads to the loss of details. Canny edges produce incomplete edges leading to color bleeding as well as false depth discontinuities caused by closed edge loops Our approach also contains incomplete depth edges which show bleeding in a few areas. However, it performs really well on smooth depth transitions while adequately addressing depth discontinuities.

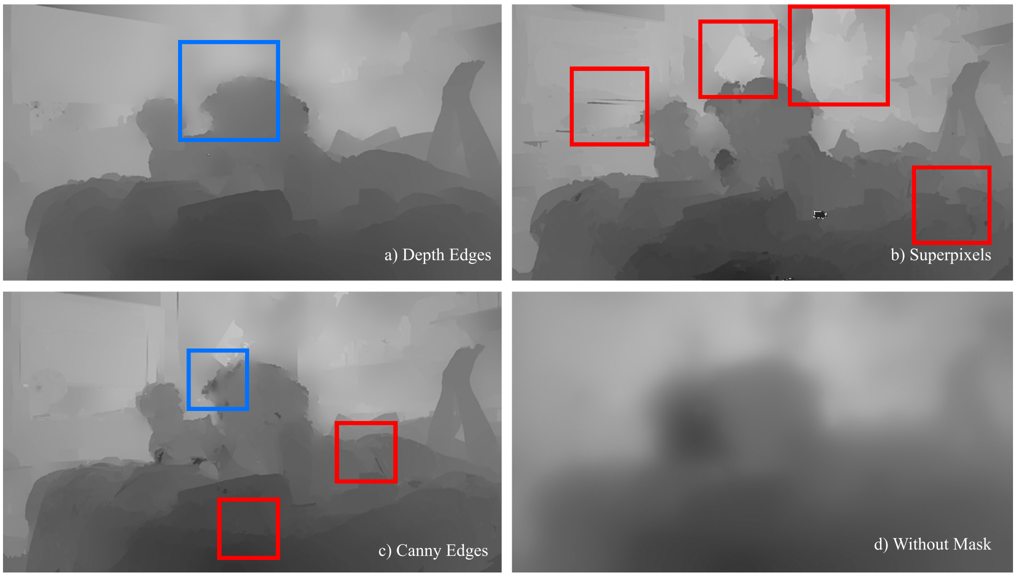


Figure 4.5: *Comparison between different regularization masks. a) Our approach has to deal with partly incomplete edges which leads to color bleeding. b) Results obtained by edges taken from superpixels or segmentation have clean edges but show depth artifacts for single segments. c) Results by canny edges show both drawbacks.*

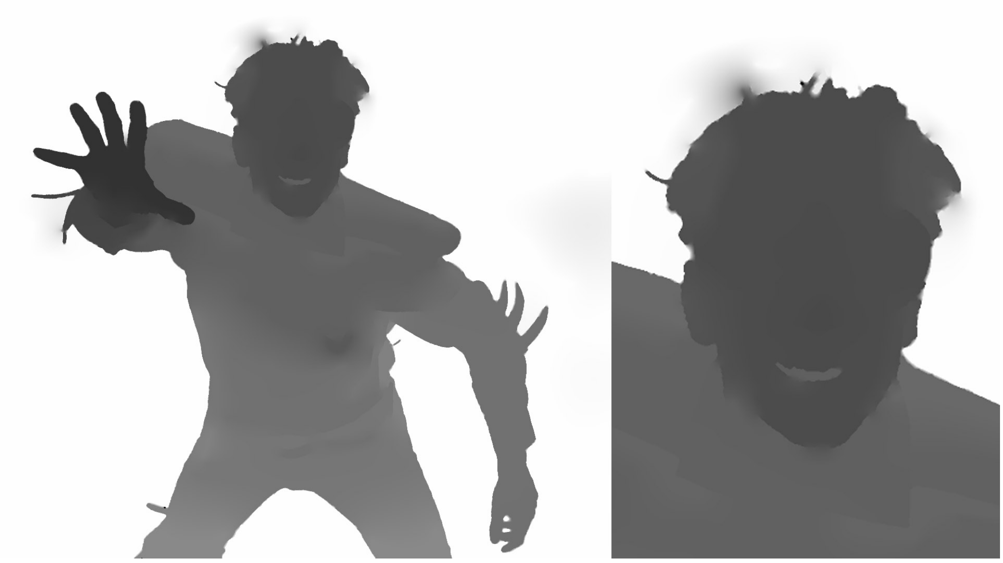


Figure 4.6: *Depth maps results for scene 2. We obtain detailed results using user scribbles. Color bleeding can be seen at a few depth boundaries.*

5 | Conclusion

In this part we discussed an interactive semi-automatic approach to learning edges or more specifically to learning depth edges by the use of random forests. Many low-level computer vision algorithms can profit from depth edges. For instance, depth discontinuities can be used to improve motion or stereo estimation or to benchmark these latter methods. Another obvious application would be masking and cropping image contents.
In our case, we applied the acquired depth edges to film dimensionalization. For this purpose, precise depth maps are of utmost importance, especially at regions of depth discontinuities, to generate an appealing visual impression. Our further working steps were the creation of sparse depth cues which can be performed automatically (via structure from motion) or manually by drawing user scribbles. In the final step, we used a variational approach to interpolate sparse depth cues to generate a dense depth map. To ensure that this interpolation step addresses depth discontinuities adequately, we used the acquired depth edges to regularize the according algorithm. As outcome we received smooth depth maps with sharp edges at these discontinuities. In comparison to interpolation regularizations via segmentations or superpixels, our depth edges did not create stepping artefacts at regions without discontinuity (e.g. the bottom edge of an object touching the ground). Thus, our approach creates an appealing viewing experience while the interactive tool enables artistic freedom to achieve the intended emotional response of the viewer.

One important advantage of our approach is that our classification model is trained for specific sequences only. Thus, we can maximally exploit all of its inherent properties. Adjustments to the model have to be made after around 25 frames. However, depending on how much the scene changes within these frames, adjustments only concern a few image regions. As a consequence, the decreased amount of manual interactions improves efficiency and reduces production costs.

5.1 Outlook

While semi-automatic workflows increase the efficiency of formerly fully-manual tools, especially artistic applications will always require adequate manual tools to control the overall outcome. Thus, further attempts to increase efficiency are limited to ensure artistic freedom. Film productions address these limits by engaging more workers. Current 3D films are created by hundreds of artists[1]. Increasing manpower is an obvious but simple approach that does not scale well. However, in the age of the internet, it has become easy to engage many workers simultaneously. Reconsidering this, one further step to increase manpower could be to provide pieces of manual working steps to anybody on the internet. Once single interactive working steps were easy enough to be processed independently from worker's experience, the internet would be a huge source of an appropriate workforce.
In Part II we analyse this idea by providing annotation micro-tasks to several online crowds. In contrast to the artistic scenario of Part I, we discuss the more scientific use case of acquiring annotations for ground truth datasets in Part II. However, the same approach could be applied to film dimensionalization working steps as well.

[1] www.fxguide.com/featured/art-of-stereo-conversion-2d-to-3d-2012/

Part II

Crowdsourced Computer Vision

In Part I we have shown that a complex workflow such as a 2d-3d conversion project can be simplified by combining computer vision methods with interactive tools. Beside optimizations on algorithms and tools a further option to increase the efficiency of such applications is to involve additional manpower. In the film industry hundreds of artists work on labour-intensive projects. If we think this through the question arises, whether the complexity of interactions needed for algorithms to succeed can be broken down thus far that workflows could be usable by everyone.

In this Part we analyse the capabilities of online crowds with regard to interactive computer vision tasks. We point out recent opportunities of *crowdsourcing* in Chapter 8 and apply this approach to create a reference data set in Chapter 9 and 10.

6 Introduction

During our work on 2D-3D conversions we greatly benefited from computer vision algorithms that for example supplied us with depth information or optical flow results. Computer vision successfully addresses many further low level applications such as detecting corners and edges via image gradients or determining and matching feature descriptors. Such automatic algorithms can rely on features and scale spaces that clearly exceed those perceptible by humans.

However, many computer vision algorithms and applications demand high level knowledge which is not achievable by such methods. As an example, one of our projects at the *Heidelberg Collaboratory for Image Processing* aims at acquiring reference data for automotive image scenes[50]. The goal is to provide scene information such as depth maps, optical flow data and context information.

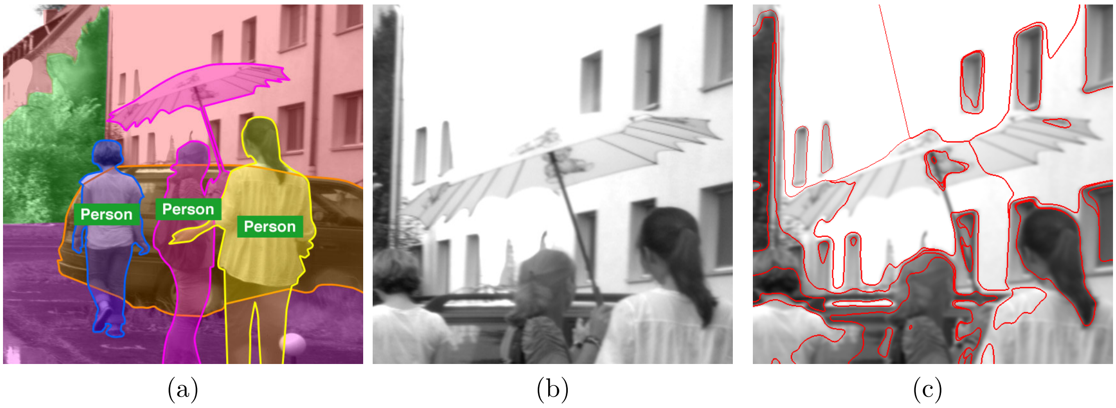


Figure 6.1: *Context information as given in 6.1a cannot be provided by low level algorithms. Edge detectors also fail at object boundaries if image data is poor. The overexposed image of the umbrella in 6.1b does not provide enough data to let algorithms detect the upper boundary.*

Figure 6.1 shows one of these frames captured by a camera system that was placed inside a car. To reliably determine the depths and movements of objects, knowledge about their boundaries would be extremely helpful. For this information though, we need to know the object shape. Figure 6.1b visualizes typical edges achievable by low level detectors. Beside the lack of context information for edges and objects, that image also does not provide sufficient data to let algorithms detect all unassigned edges. The umbrella in 6.1b for example is overexposed and pixel intensities do now allow to detect an edge at the upper boundary.

More sophisticated methods achieved by machine learning, pattern recognition or deep learning are able to generate object segmentations, shapes and classifications. However, these approaches also require initial training data for learning purposes. Thus, the initial data acquisition needs to come from us people.

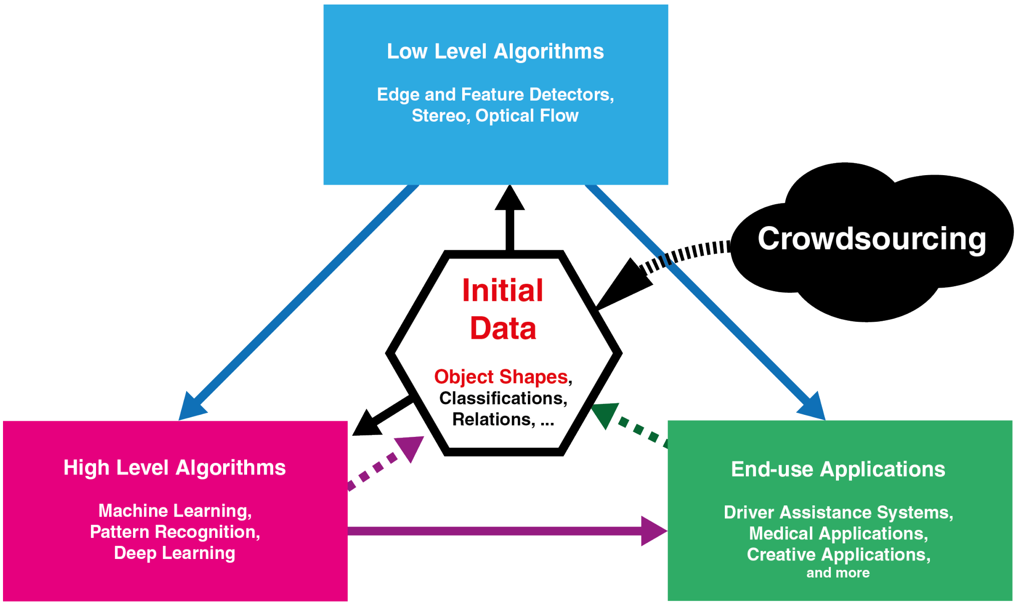

Low Level Algorithms
Edge and Feature Detectors, Stereo, Optical Flow
Crowdsourcing
Initial Data
Object Shapes, Classifications, Relations, ...
High Level Algorithms
Machine Learning, Pattern Recognition, Deep Learning
End-use Applications
Driver Assistance Systems, Medical Applications, Creative Applications, and more

Figure 6.2: *Learning approaches can provide high level knowledge to end-use applications. However, these methods are based on initial information that somehow has to be acquired. Since acquisition can be extremely time-consuming, we analyse if many laymen can succeed on tasks, that formerly have been solved by a few experts.*

When working with thousands of images that need to be assigned with object shapes or semantic labels, data acquisition can be a challenging task. Using the example of our reference data set, we labelled over 70,000 high-precision object boundaries of vehicles and pedestrians. The enormous amount of annotations arose for only 140 seconds of our footage captured with 25 Hz. This example is not a single case. At a time where image data is analysed for many highly diverse applications, it is improbable that the increasing amount of images that is created every day, can be processed by a few experts in future. While in end-use applications image data is mostly processed automatically, the initial data to develop these applications needs to be acquired by the creators.

Opportunities that address initial data acquisition for such large data sets is a general concern. In the film industry where it is common to work with a lot of footage, labour-intensive projects such as the 2D-3D conversion approach presented in Part I, are processed by hundreds of artists at once. To speed up processes that cannot be further simplified or automatized they simply increase the number of workers. If we think this through more thoroughly the question arises, whether the complexity to create initial data can be broken down thus far that everyone can contribute. Opportunities where plenty of people may contribute are common in the internet era. One of these opportunities is called ***crowdsourcing*** which *"is the act of taking a job [...] and outsourcing it to an undefined, generally large group of people [...]"*[1]. Since crowdsourcing promises the opportunity to work with many people at once, we analyse whether its capabilities are worth to be considered to be used for projects like the creation of our ground truth data set.

This Part is structured as follows. We will first introduce the mentioned ground truth data set in Chapter 7 and present acquisition methods for ground truth in general. Apart from common methods we will also discuss crowdsourcing approaches that showed promising results.
Chapter 8 introduces the idea of crowdsourcing and its opportunities. After a discussion about its emergence, we introduce crowd providers (8.2), different annotation protocols (8.4), quality assurance methods (8.5) and finally address scientific applications in Section 8.6.
We conclude that crowdsourcing could be an appropriate opportunity for ground truth acquisition and present our experimental setup to analyse this topic in Chapter 9. Our results on annotation accuracy and the overall efficiency are presented in Chapter 10. Chapter 11 proposes future work and presents first attempts of further annotation protocols. Finally we conclude our analysis in Chapter 12.

[1] http://crowdsourcing.typepad.com/

7 | Reference Data Acquisition

As mentioned in Chapter 6, one of the research topics at the *Heidelberg Collaboratory for Image Processing* is *ground truth* and *reference data* generation for performance analysis of computer vision methods[1][50, 61, 60, 38]. Daniel Kondermann [49] defines three sub-categories for reference data: reference data without ground truth, with weak ground truth and reference data with ground truth. Ground truth in this case means that the data set includes results that are *"at least one order of magnitude more accurate than the quality that can be expected from the vision system at hand"*.

Examples for reference data without ground truth are large data sets with thousands of images for which ground truth creation is infeasible. Meister et al. [61] proposed a huge reference data set which contains 15 Terabyte *"of high-quality stereo image sequences in a driver assistance scenario"*[49]. The footage includes images of strongly varying weather conditions which makes it impossible to create ground truth for it. Using this approach, algorithms can be tested by hand for a wide range of scenarios and intuitively unveils situations where an application could fail. The drawback of this approach is the need of experts that evaluate applications manually. The acquisition time for reference data sets without ground truth can be low. In return, evaluation is labour-intensive and costly.

The opposite approach are reference data sets including ground truth data which can be either synthetic data for which ground truth inherently is given [15, 20] or real data where typical approaches are using LIDAR systems and structured light scanners[60]. The latter approaches achieve very accurate results but are cost-intensive and produce huge amounts of data which needs to be manually processed. Furthermore, dynamic scenes cannot be measured with high accuracy yet. Approaches such as the synthetic SINTEL dataset proposed by Butler et al. [15], use computer-generated graphics with the aim to imitate real data as accurate as possible. The advantage is that properties of rendered objects are well-known and various conditions such as the weather, the time of day, object motion or material properties can be precisely adjusted. The downside is, that such datasets are not well-studied with regard to physical correctness.

The third option for reference data sets discussed by Kondermann is to include weak

[1] http://hci.iwr.uni-heidelberg.de/Benchmarks/

ground truth. Examples are the Middlebury dataset [3] from 2010 or the KITTI dataset [31] presented in 2012. The Middlebury data set was proposed by Baker et al. and provides optical flow ground truth. To achieve optical flow with subpixel accuracy, they used high resolution cameras to capture visible light as well as ultraviolet light. Additionally they applied a beneficial texture to the scene which only was visible in ultraviolet images. Thus they were able to run a blockmatching algorithm only on ultraviolet textured images. The outcome was downsampled to achieve a subpixel accuracy of up to 1/60 of a pixel for low resolution images. Geiger et al. [31] who proposed the KITTI dataset combined an inertial measurement with a high-speed LIDAR system to reconstruct an automotive scene. Due to the LIDAR system, accuracy can be expected to be around 3 pixels which is sufficient for automotive applications.

While the previous ground truth approaches aim at benchmarking low-level vision such as stereo and optical flow methods, there are alternative approaches for high-level computer vision. Machine learning approaches for object recognition and detection often rely on weak ground truth by using **human annotations**. For this purpose annotations mostly do not need to have superior accuracy. Many reasonable sources to test and train high-level methods exist such as the Berkeley segmentation dataset [58], the PASCAL Challenge [26] or the Caltech dataset [27, 32] as well as crowdsourced datasets such as *ImageNet*[22] and *LabelMe*[71] which we discuss in Chapter 8.
However, these rough annotations wouldn't satisfy the high-quality requirements needed to benchmark low-level algorithms. The first attempt to **use human annotations for motion analysis** was proposed by Ce Liu et al. [54] in 2008. They presented a tool capable to annotate layers, objects and point correspondences between subsequent frames. An automatic initial guess for optical flow propagates annotated objects to the next frame. Afterwards this initial guess can be edited with the tool. In 2013, Donath and Kondermann [25] proposed the **first crowdsourcing approach** to annotate optical flow. Therefore they extended the tool of [54] to make it accessible for *Amazon Mechanical Turk*[2] workers. This crowdsourced approach was evaluated with the Middlebury laboratory dataset as well as with the synthetic SINTEL dataset proposed by Butler et al. [15]. Their results were around twice as accurate as data from the KITTI automotive dataset. Hence they conclude that *"MTurk-based motion annotation is a feasible, cost-effective and currently the only method for ground truth generation for large-scale outdoor scenes with dynamically moving objects."* However, real automotive scenes like KITTI include varying weather and lightning conditions and a lot of moving objects. While the accuracy achieved on Middlebury and SINTEL would be appropriate for automotive scenes, they did not prove that crowd workers perform equally on the latter scenario.

[2] A common crowd provider for crowdsourcing projects. We discuss this and other providers in Chapter 8.

Further attempts to use highly accurate human annotations for automotive applications have been made. Examples are the *CamVid Database* [12] from 2009, the *Daimler Urban Segmentation Dataset*[74] from 2013, its successor the *Cityscapes Dataset*[19] proposed in 2015 and the *HCI Stereo Ground Truth Data Set* which is based on the ACCV 2014 Paper *"Stereo Ground Truth with Error Bars"* [50]. The Cityscapes data was not yet available at the time of publication of this work. However they plan to publish 20,000 annotated frames with different weather conditions captured in 50 cities by the beginning of 2016[3]. They suggest to use their dataset for performance analysis of *semantic urban scene understanding* and for research which requires large volumes of annotated images (e.g. deep learning). Every frame will contain at least coarse annotations of object shapes categorized into 25 classes such as different vehicle types, humans and environmental classes. 5000 frames will be annotated with high precision. Additional meta data will be stereo views, depth maps, gps coordinates and motion data from vehicle odometry. The *HCI Stereo Ground Truth Data Set*[4] provides stereo ground truth with error bars. The data was acquired with a method proposed in [50]. Similar to the KITTI dataset, they used a LIDAR system and a stereo camera setup. In contrast to the KITTI acquisition method, first a static scene (all vehicles and moving subjects temporally had been removed) was scanned with the LIDAR system. Afterwards the stereo camera recorded the scene with vehicles and pedestrians on it. With the use of a static LIDAR scan they avoided low density pointclouds and motion artefacts which may arise for moving systems. As a result, the scan only matches the recorded stereo data at image regions which are also static. Depth information for vehicles and pedestrians is not present in the 3D-scan. To address these image regions, the dataset also provides manually-annotated masks for dynamic objects. Beside using masks for dataset creation, these annotations enable especially the evaluation of stereo performance at depth discontinuities. Manual annotations for dynamically moving objects provide geometry-metrics even without knowing stereo and optical flow ground truth. Honauer et al. [38] presented the *HCI Stereo Metrics* which, amongst others, suggest metrics to quantify performance at depth discontinuities.

As we have seen, human annotations are a common approach for different purposes during ground truth acquisition for performance analysis in computer vision. Therefore, the general idea of using crowdsourced acquisition methods such as [25] seems to be a promising idea. We take the related work as an opportunity to analyse whether crowdsourcing is capable to produce annotations that even satisfy high-quality requirements. As an application we choose the *HCI Stereo Ground Truth Data Set* and evaluate representative crowd user groups on contributing masks for dynamic objects.

[3] http://www.cityscapes-dataset.net/

[4] http://hci.iwr.uni-heidelberg.de/Benchmarks/document/StereoErrorBars

8 | Crowdsourcing

As presented in Chapter 7, there are several applications for human annotations for computer vision problems. Human input is used to acquire optical flow [54, 25], semantic and rough object annotations [22, 71] or to assist accurate ground truth acquisition [19, 50]. Since most of these applications address large datasets, the number of workers is the bottleneck of manual data acquisition. In this chapter we discuss *crowdsourcing* as an approach to procure labour. We present its idea in general and analyse proposed opportunities for scientific use cases.

According to the International Telecommunication Union (ITU) [1] about 3.2 billion people had access to the internet in 2015. 2 billion of these people are living in developing countries. Looking at households, 81.3 percent of households in developed contries and 34.1 percent of households in developing countries have internet access. From 2000 to 2015 the internet usage has grown about 806 percent[2] worldwide. The reasons for that immense growth are diverse. Prices for hardware decreased, the number of internet connections rose due to infrastructure deployment and thus making connection speeds getting steadily faster. Along with this growth many phenomena have arisen not only changing our daily life but also enabling a vast number of new opportunities and technologies. Now, the work of many people involves the internet for different purposes. Some websites offer micro-tasks to anybody. This placement service is also called *crowdsourcing.* The idea of crowdsourcing involves mechanisms and processes that have existed long before the internet, but are now pushed by the global network and exceeds the opportunities of the offline age.

The term *crowdsourcing* was coined by Jeff Howe and appears in an article of the Wired magazine[41]. Howe says: *"Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call."*[3] According to Unterberg[85], Howe relates the naming crowdsourcing to the book *The Wisdom of Crowds*[84], written by James Surowiecki. Surowiecki presents several case studies especially in the field of economics and psychology. He argues that

[1] http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf

[2] http://www.internetworldstats.com/stats.htm

[3] http://crowdsourcing.typepad.com/

decisions made in groups often perform better against those of single individuals of that group. However in many situations crowds made weak decisions due to their cognitive abilities or failing cooperation when they are influenced by the ideas and mind of others. Nevertheless, good judgements were made when there were various and diverse ideas and opinions, a wide range of experience and knowledge within the crowd, and when their decisions are made independently from each other to be merged into one collective decision. While Surowiecki's *The Wisdom of Crowds* argues rather generally about crowd decisions, the age of internet enabled a vast number of online platforms and communication opportunities trying to utilize the judgement of crowds. Maier-Hein et al. [56] names *Amazon Mechanical Turk*[4] to be the *"earliest microtask-based crowdsourcing system"*. It was invented by Peter Cohen and initially used to find duplicate descriptions on Amazon's web pages. At the same time several computer vision approaches that used crowdsourcing were published by Luis von Ahn [89, 90, 91], Catherine Wah [95] and Russell et al. [71].

Our further discussion addresses terminology and variants of crowdsourcing in Section 8.1. A list of crowd providers that offer different opportunities for crowdsourcing is given in 8.2. Incentives that let people contribute to crowdsourcing opportunities are presented in 8.3. Common approaches to design crowdsourcing micro-tasks and methods to assure quality are presented in Section 8.4 and 8.5. Finally we discuss further computer vision approaches that use crowdsourcing in 8.6.

Contents of this chapter

[4] https://www.mturk.com

8.1 Terminology and Variants

According to Leimeister[52] one can categorize the usage of crowdsourcing into three classes: *crowd funding*, *crowd creation* and *crowd voting*. The latter two types play an important part for applications which require human knowledge or opinions to succeed. *Crowd voting* primarily is used to create ratings for products. For example *Amazon* or app stores intensely use user ratings to give recommendations and there exist many rating portals which base their business models on crowd voting.

Crowd creation applications normally require people to contribute more than just votes. Companies for example ask their customers to submit ideas or to produce concepts and designs to improve products. Crowd creation also appears on several online platforms whereas *Amazon's Mechanical Turk*[13] is one of the most prominent ones. Those platforms provide requesters to run micro tasks which are processed by the crowd. Micro tasks can vary immensely in their requirements and can reach from classification tasks for provided images to physical tasks such as the request of taking a photo of a specific location. Also to resolve a CAPTCHA[5] could be counted to crowd creation. For example reCAPTCHA not only provides tasks to prevent automatized scripts to access online services but also utilize the given answers to digitize printed material[92].

Contrary to Leimeister and his classification of crowdsourcing, the company *Crowdsourcing LLC*, which provides news and articles about crowdsourcing divides crowdsourcing into five categories on their website[6]. Namely those are *crowd funding*, *cloud labor*, *crowd creativity*, *distributed knowledge* and *open innovation*. *Cloud labor* means that crowdsourcing can comply labour demands whereby workers perform simple to specialized tasks. *Crowd creativity* addresses contributions of creative communities such as product development, photography, advertising, graphic design and brand concepts. *Distributed knowledge* involves crowd driven knowledge systems which may be news, blogs and journalism. *Open innovation* means that an entity not only relies on their internal developments and ideas but also includes external knowledge and innovations to stay competitive.

Quinn et al. [63] introduce a taxonomy for the term *human computation* which was coined by the same-titled dissertation of Luis von Ahn[88]. They associate *human computation* with crowdsourcing, *social computing* and *collective intelligence*. According to Quinn et al. [63], human computation *"problems fit the general paradigm of computation, and as such might someday be solvable by computers"*. While crowdsourcing generally includes all applications which involve a crowd into its workflows, this is not the case for human computation. The latter term only addresses workflows where *"human participation is directed by the computational*

[5] Completely Automated Public Turing test to tell Computers and Humans Apart

[6] www.crowdsourcing.org

system or process"[63].

Our intent is to involve worker's contributions into computer vision processes. According to previous definitions our approach could be categorized as *human computation*, *crowd creation* and *cloud labor*. However, in this work we stick to the general term crowdsourcing due to its more common expression.

8.2 Crowd Providers

To benefit from the workforce of crowd workers we somehow need to access a crowd. The straight forward approach is to rely on commercial crowd providers that have specialized on crowdsourcing. One of the most popular and as mentioned, one of the earliest crowd providers is *Amazon Mechanical Turk*[7] (MTurk). MTurk acts as a mediator between their workers (MTurkers) and MTurk requesters. MTurk requesters need to provide a U.S. billing address in order to use the platform. They can place their own HTML framework on MTurk and connect this system with MTurk using the MTurk application programming interface (API). Requesters have to pay a Mechanical Turk fee which is twenty percent of the workers reward[8]. MTurkers are paid per task. The payment is specified by the requester and needs to be at least $0.01. According to Ipeirotis [43] *"25 percent of the HITs created on Mechanical Turk have a price tag of just $0.01, 70 percent have a reward of $0.05 or less, and 90 percent pay less than $0.10"*. Ipeirotis estimated the hourly wage to be approximately $5 while Horton et al. [40] measured an hourly median wage of $1.38. According to Mason and Watts [59], increasing the payment for tasks also increases the quantity but not the quality. However, within their analysis, *"workers who were paid more also perceived the value of their work to be greater, and thus were no more motivated than workers paid less"*[59].
Micro-tasks, which are also called HITs (human intelligence tasks) can be limited to certain MTurkers. For example a task can be rolled out only to MTurkers which successfully contributed a specific number of results to a previous project. Also MTurkers can be assigned quality levels which enables requesters to categorize user groups with different competences.
The MTurk crowd is quite diverse regarding their nationality, gender, and age. According to Kittur et al. [48] *"MTurk participants were more demographically diverse than standard Internet samples and significantly more diverse than typical American college samples"*. This especially matters for research in psychology and social sciences.

CrowdFlower is another crowdsourcing service. Contrary to MTurk, they rely on workers which are not directly registered at CrowdFlower. Crowdflower uses various

[7] www.mturk.com

[8] As of December 2015

worker channels from MTurk, *Samasource*[9], *clixsense*[10] and more. Additionally to MTurk they also provide management and analytic tools regarding the workers and tasks and a gold standard validation. Finin et al. analyse and compare both, CrowdFlower and MTurk [28].

Samasource and *CloudFactory* are commercial services which mediate crowd sourced jobs to workers living in developing countries. They manage those workers and educate them for specific tasks. Tasks are paid according to working time.

Providers such as *clickworker*[11] address to manage e-commerce via their crowd. *Crowdsource* [12] provide several writing related solutions. There are also commercial crowds such as *rapidworkers*[13], jobboy [14], *microworkers* [15] and *crowdtap*[16], which are specialized on marketing and brand related tasks. Workers are asked to rate videos or follow somebody on social media. Since the focus of these crowds deviate from our intent, we primarily aim at using MTurk for our purposes since MTurk has shown to be capable for computer vision tasks in general.

In contrast to the mentioned crowds there are also companies which offer a service to fully manage the whole crowdsourcing workflow. Requesters only submit their data while the service providers take over data management, crowd management, quality assurance, the actual crowdsourcing software and the payment of workers. The Chinese company *Datatang*[17] is one example. Another company is *Pallas Ludens*[18] which is a spin-off of the Heidelberg University. *Pallas Ludens* specializes in crowdsourcing data for computer vision research and applications. The company arose during research on crowdsourcing at the *Heidelberg Collaboratory for Image Processing.* Results presented in Chapter 10 were accomplished in close collaboration with *Pallas Ludens.*

8.3 Incentives for Crowd Workers

The incentives of people who contribute to crowdsourcing projects are an important attribute with regard to contribution quality, reliability and efficiency. Furthermore the worker's motivation constrains the maximum viable task difficulty. Weak incentives result in less attentiveness, worse quality or a bad user participation. This

[9] www.samasource.org
[10] www.clixsense.com
[11] www.clickworker.com
[12] www.crowdsource.com
[13] www.rapidworkers.com
[14] www.jobboy.com
[15] www.microworkers.com
[16] www.crowdtap.com
[17] factory.datatang.com
[18] www.pallas-ludens.com

is a far-reaching interdisciplinary topic which also involves subjects like psychology or user experience design. We discuss the most obvious opportunities but do not claim to present an extensive analysis in this Section.

Payment

One of the most obvious incentives is **pay**. Opportunities to get paid by money for contributing to crowdsourcing projects were discussed in the previous Section. Besides real money, payments can also be transacted with virtual currencies. For instance players of online games are often able to watch optional video advertisings to be rewarded with in-game goods. Instead of video ads, content providers could also mediate crowdsourcing jobs. Another interest group could be customers of a online movie provider who could optionally solve micro-tasks to get rewarded with free movies in return. For now, most content providers offer paid content or use advertising as a monetization model. To mediate crowdsourcing jobs would be another option. While there exist only a few commercial crowds whose workers are paid by real money, there are plenty of content providers which could optionally mediate such micro-tasks and whose number of users exceeds that of commercial providers by far. Offering micro-tasks to crowds whose initial intent is different might create new problems. Users do not necessarily assume to get confronted with a task apart from the game. While many of them are familiar with passively watching ads to earn virtual goods, they need to play an active role for micro-tasks. Though the task may be fairly refunded, it means further effort to convince users to agree to such offers. Another issue could be a weak attentiveness and receptivity compared to commercial crowd workers due to the initial situation.

In Chapter 9 and 10 we analyse tasks that we offered to players of an online game provider.

Altruism

Projects which appear to be of interest for people induce another potentially strong incentive: **altruism**. When people are confident that their contributions are for a good cause, their commitment can be pretty strong, though they are not paid for it. Citizen science projects are good examples [71, 18]. Volunteers for instance localize and track wildlife in *Gorongosa National Park* in Mozambique. By doing so they help to document the recovery of the park whose population was decimated by decades of war[19]. The latter project is part of the *Zooniverse* platform[20] introduced by Simson at al. [76]. They provide a web-based citizen science framework which

[19] www.wildcamgorongosa.org/

[20] www.zooniverse.org

has been first used for the *GalaxyZoo* project [65] in 2007. 165,000 volunteers were engaged in the classification of galaxy images. Until the beginning of 2014 the platform counted 900,000 registered volunteers overall.

Having volunteers solving crowdsourced tasks can be a powerful approach. However people need to be interested in the overall project or agree with its purpose. There are a vast number of projects which could benefit from crowdsourcing but many of them may not meet the requirements to benefit from altruism.

Games and Gamification

To make contributors stay motivated, the workflow design can be optimized to make the most fun of it. The appeal of crowdsourcing tasks can be improved by ***gamification*** elements[23]. Gamification means that basic functions are enhanced by elements that are typical for games. The user experience and motivation can be increased although the actual application of a task may be unappealing or too complex. Such gamifying elements can be experience points, a highscore list to create a competitive environment or a progress bar to track one's progress. Also contributions could be rated leading to different rankings and rewards. Using gamification and reward systems to keep people motivated is nothing new. These principles are even known to be used for modern school education. Research within this field claims gamified applications to show significant improvements regarding motivation, learning and result quality [89, 77, 91, 37].

A popular example from 2010 is the scientific discovery game *FoldIt* which was presented by Cooper et al. [18]. Actually their research on protein structure prediction is extremely computationally expensive. Even smaller proteins do have over 1,000 degrees of freedom[17]. To address this problem they exploit strategies and approaches of *FoldIt* players. Players can directly manipulate 3D protein models and use simplified versions of the *Rosetta* structure prediction algorithms[68] to optimize an automatically calculated energy. By using this combination of human strategies with computational algorithms, Khatib et al.[46] were able to retrieve qualitative models of an AIDS protein that kept researchers on the run for fifteen years. Using the crowd they succeeded within three weeks. For research on AIDS the revelation of this protein is only a small piece. However capabilities of this approach are a remarkable example for the potential of crowd assisted science. Besides typical game elements like colorful achievement points popping up, *FoldIt* provides high score lists for different categories and periods.

There are a bunch of further crowdsourced computer vision applications packaged as a game. Luis von Ahn was one of the very first adopters who used crowdsourcing for computer vision applications. His first approach introduced in 2004 was *the ESP game*[89], which generates labels for image contents as outcome. Two players get to see the same image. Both try to guess the same describing word.Once they found a match they could proceed with the next image. In 2006 von Ahn et al. proposed an extension for 2 to 5 players called *Phetch* [90] which collects semantically richer descriptions. In the same year, von Ahn et al. presented *Peekaboom* [91], a game whose outcome not only are labels for image content but also the localization of that content. In 2009 Ho et al. [37] published *KissKissBan* which is similar to *the ESP game*[89] but brings along some extensions which lead to a more diverse set of annotations. Besides the two collaborative players there is also a competitive player. The latter one tries to avoid the others to reach a consensus. Another approach presented by Di Salvo et al.[24] in 2013 asks players to take the best possible photos of fish from a video sequence. Technically they have to adjust a region of interest by moving a crosshair. After collecting multiple results for each frame, Di Salvo et al. run a cluster algorithm to remove noise. The gathered data is used to run segmentation algorithms.

Crowdsourcing games can evoke strong motivations making people to stubbornly solve tasks. However it takes a lot of effort to build a game and not every crowdsourced application can be mapped onto a game. Also the quality and quantity of contributions closely depend on how well a game is implemented. A big issue here is *user retention* which means that project operators need to maintain the user basis. Relying on a game for crowdsourcing applications requires maintenance and product updates to keep players interested. This might be an unnecessary drawback for many projects and in particular for smaller and temporally short ones.

Implicit Work

Quinn et al. [63] names **implicit work** as an additional incentive. That means crowdsourced tasks can be integrated into another workflow which existed anyway. As an example they mention the *ReCAPTCHA* system presented by von Ahn et al.[92] in 2008, which replaces computer generated CAPTCHAs[21] that are solved by users to identify them as human. *ReCAPTCHA* provides images of words scanned from old books or newspapers which could not be recognized by optical character recognition. CAPTCHAs are a great way to get human computation tasks solved as long as their complexity is reasonable.

8.4 Designing Crowdsourcing Tasks

Our primary goal in this work is to collect high level knowledge by using crowdsourcing approaches. In previous chapters we have presented several opportunities

[21] Completely Automated Public Turing test to tell Computers and Humans Apart

to engage crowds. Furthermore we have discussed incentives that could induce people to contribute to crowdsourcing projects. Suppose we had access to one of these crowds, we now have to decide how to set up the actual micro-tasks that are assigned to crowd workers. Besides the previously mentioned incentives there are several issues and properties which could be considered to run a successful crowdsourced project. For a proper task design, appropriate quality assurance mechanisms need to be worked out. Basically there are three issues that need to be addressed:

1. Which is the most **suitable task workflow and difficulty** for a crowd?
2. How to address **occasional errors**?
3. How to deal with workers that wilfully contribute **wrong results**?

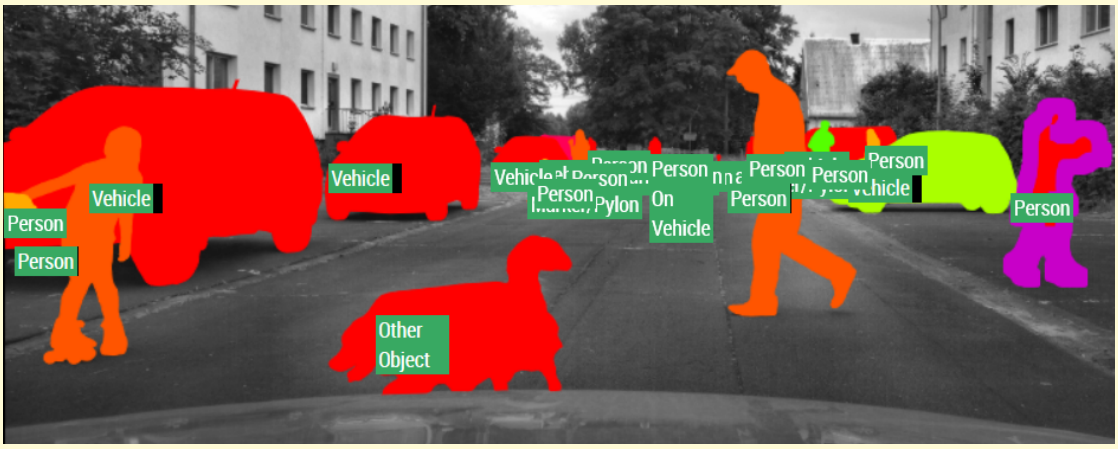
One of the most limiting factors for crowdsourced micro-tasks is the competence and perceptive faculty of crowd workers. These properties have to be carefully considered to design suitable tasks. A crowd which shows a large variety of these properties means an even bigger challenge. In the latter case one could try to find common ground with regard to task difficulty to guarantee reasonable results from every worker. To rely on more sophisticated tasks, another option would be to select only those workers that show the minimum required skill to succeed on these tasks. In 2012 Su et al. [82] presented a crowdsourcing approach which relies on different tasks and user groups. Their goal was to acquire bounding boxes for different objects in images. They defined three separate tasks: drawing bounding boxes, rating the quality of bounding boxes which previously had been drawn and answering the question whether the shown image contains additional non-annotated objects. To be granted access to each task, workers had to pass separate gold standard tests [87].

In 2008, Sorokin and Forsyth [80] analysed how *Amazon Mechanical Turk* workers perform on such different annotation protocols. They test tasks such as binary questions about the affinity of automatically selected and uniformly distributed points, drawing contours around objects of interest and placing landmarks on human bodies for pose estimation. In 2010, Whiteman [99] discussed further task types with regard to motivation and competition. In the same year, Little et al.[53] presented *Turkit* which is a toolkit to test and evaluate approaches directly on Mechanical Turk.

Since the competence of crowd workers can strongly vary, it is also a question of when to schedule a task. For this purpose, Rajan et al.[66] presented an *"online learning approach for optimal task scheduling"* in 2013. They show *"that algorithms that schedule jobs by adaptively learning current crowd performance can significantly outperform other algorithms that do not learn"*.

The following example presents reasonable crowdsourcing workflows with regard to our goal of annotating object contours for reference data sets:

Example: Different Task Workflows



Micro-tasks for computer vision applications can be implemented in several ways. Suppose images of an automotive scene have to be labelled. In particular, contours of every object which affects the driver's perception need to be annotated. This demand can be mapped to different tasks.

1. A frame could be processed by a **single worker** who **annotates all objects**. This would be a time-consuming and advanced task. Annotations may have a frame-wise worker-specific systematic error. Worker's annotation quality may decrease during long tasks. Though, using this approach for versatile experts could be reasonable to save the overhead which had arisen from splitting the task in further pieces.

2. A frame could be mapped to several micro-tasks. Several workers could be assigned to specific **parts of the image** or they are asked to **label a limited number of annotations only**. Workers could see previously added annotations (which they may improve if needed) and then address some of the remaining objects.

3. Though we broke down a full-frame annotation task into several micro-tasks, it is still an advanced task and the process could be further simplified. One of the **easiest tasks is a binary question**. In our case a segmentation algorithm could highlight different parts of a frame and workers have to classify whether the shown areas belong to an object of interest.

8.5 Quality Assurance Methods

We have proposed different options for designing and assigning micro-tasks. A remaining major topic which needs to be addressed to successfully run crowdsourcing projects is quality assurance. Especially in our case where results are utilized for reference data sets, it is inevitable to ensure a proper result accuracy.

A common practice to evaluate and process results is to determine the **majority vote** of multiple user contributions. This approach pursues the proposition of Surowiecki's book *"The Wisdom of Crowds"*[84] which implied that good results can be produced when independent decisions of crowds are merged into one collective decision. Majority votes can increase process robustness by addressing the natural variability of human work and enables the detection of outliers. However, the need for additional results will increase the project's costs and its running time.
In 2014, Maier-Hein et al. [57] showed that results gathered from laymen via majority votes compete with those of experts. In their study about processing laparoscopic images, they analysed segmentations of medical tools annotated by laymen and by experts. As a result "*the performance of an endoscopic object classifier was not statistically different when trained with crowd data compared to expert data*". This performance of laymen compared to experts is proven by many other papers (Welinder et al.[97] (2010), Snow et al. [79] (2008), David and Skene [1] (1979)).

Majority votings of crowdsourced tasks is an approach that can lead to reasonable task results [79, 67]. In 2008, Sheng et al. [75] analysed the effectiveness of repeated labelling and majority voting. They conclude that "*Repeated-labeling is a tool that should be considered whenever labeling might be noisy, but can be repeated.*" They basically focus on theoretical cases where results are noisy but have a similar quality. However, when quality differs too much, majority voting was not the best solution to maximize quality. In this case "*it is preferable to use the single highest-quality labeler*". Since majority votes are a common and promising approach to achieve reasonable results, we attempt to use these methods in our analysis in Chapter 9. In contrast to previously discussed proposals we primarily use this approach to evaluate contour annotations.

Majority votes primarily are used to merge appropriate results into a single, usually better result. If the number of outliers is little, the approach can also be used to detect and reject these. However, as its name implies, majority votes can only succeed if the majority of candidates is reasonable. In an unsupervised crowdsourcing project, almost certainly some contributors will permanently submit wrong results. Reasons are twofold. Either people's skill and its perceptive faculty was too little to solve according micro-tasks, or contributors submitted wrong results wilfully. The latter persons are also named *spammers.* Those people either try to achieve task rewards with minimal effort or aim at manipulating the process in general. In any case, contributors with generally wrong results need to be rejected to guarantee a reasonable outcome. A common approach to detect spammers and to rate user contributions are **gold standard** tests[87, 80, 82, 98]. This means the best achievable results for a subset of tasks are already available. These results are used to rate the reliability and quality of user contributions. In our own approach we also name such tasks *calibration tasks*. Calibration tasks enable us to classify users by accuracy and efficiency. To use gold standard tests, initial test data is needed. Such data could either be computer-simulated images with known ground truth or a small sub set of images annotated by project owners. Results of users who passed the gold standard test can be used to supplement the gold standard test data. Westphal et al.[98] successfully utilized gold standard tests for their research on interstellar dust. They found over 27,000 volunteers that passed their tests on localizing putative interstellar impacts. Westphal et al. chose laboratory simulations and images which had been already examined to rate users and their results. Gold standard tasks can also be used to train users. Since the correct answers are known users can be instructed about their performance. Hence, common mistakes which would be repeated can be prevented. For example the previously mentioned approach of Su et al. [82] uses confident results to train new participants. Users receive specific feedback for inaccurately drawn bounding boxes, for annotations which address wrong objects or for falsely answered questions.

8.6 Scientific Crowdsourcing

The previous sections basically presented different opportunities and approaches about implementations of crowdsourcing projects. Scientific applications usually run their crowdsourcing tasks either on the commercial crowd platform *Amazon Mechanical Turk* or as separate *citizen science* projects. User's incentives to contribute to these applications are either pay or altruism whereas especially projects that rely on altruism mostly use gamification methods to keep participants motivated. The most common approaches to assure qualitative results are majority votes and gold standard tests.

We now discuss approaches whose application correspond the our analysis of acquiring object annotations for large image data sets.

Crowdsourced scientific acquisition tasks vary strongly. They reach from simple tasks such as answering binary questions about image contents to sophisticated tasks for complex projects which require to work on videos[93] or processing protein structures[18]. One of the biggest crowdsourced image data bases was presented by Deng et al. [22] in 2009. It is named *ImageNet* and contains more than 14 million images whereas over 1 million of them are labelled with bounding boxes[22].

[22] http://image-net.org/about-stats

The database can be used as a training resource or benchmark dataset for *"visual recognition applications such as object recognition, image classification and object localization"* [22]. To obtain image labels, Deng et al. asked users of *Amazon Mechanical Turk*, if images contain objects of a specific class. Users had to answer binary questions with 'yes' or 'no'.
Before the release of *ImageNet*, Russell et al. [71] presented a similar dataset project in 2008. They proposed a database and an online annotation tool called *LabelMe* which has been used to collect annotations for over 180,000 objects in more than 12,000 images. The number of images deceeds that of *ImageNet* by far. However *LabelMe* not only provides classifications but also segmentations for every object. Contrary to the commercial Mechanical Turk crowd which was used for *ImageNet*, *LabelMe* is accessible for everyone and relies on volunteers. The common tasks given to these users is way more complex than a binary question. People have to draw contours around each object and afterwards assign a label to it. Independently from their database, Russell et al. also provide the LabelMe annotation tool source code as well as an mobile app for free. The tool can be used to run crowdsourcing projects with an individual crowd and can also be deployed for MTurk[23].

In 2010, Branson et al. [10] presented a crowd-driven semi-automatic classification algorithm to classify birds. In contrast to the previously mentioned approaches, they integrated algorithms directly into the crowdsourcing pipeline. Their goal was to design tasks that formerly needed advanced knowledge, to be solvable by anybody. Similar to the *20 questions game*[24], users had to answer automatically determined questions to classify birds. After each iteration their multi-class object recognition algorithm determines the most informative next question. By doing so the needed number of answers given by untrained Mechanical Turk users to determine a bird species decreased from 11.11 to 6.43. Wah et al. [94] and Branson et al. [9] also proposed an extension of this approach in 2011 and 2014. MTurkers again were asked binary or multiple choice questions. Additionally they had to click on specific image areas such as the breast, belly or beak of a bird. Branson et al. [9] exploit the human expertise in detecting, localizing and roughly classifying objects as well as the advantages of computers in computing probabilities, associating categories and attributes and handling complex taxonomies such as that of birds. By the use of dynamically determined questions, people succeeded more than 3 times faster on specifying bird species.
A similar approach to utilize user feedback was proposed by Rupprecht et al. [70] in 2015. They presented an iterative crowdsourced segmentation workflow. An algorithm determines the most reasonable coordinate for the next segmentation seed which then is classified by the crowd.

The previously presented approaches all address single independent images. However, for our analysis we aim at processing frames of a video sequence. Several stud-

[23] http://labelme2.csail.mit.edu

[24] http://20q.net/

ies address interactive object recognition and labeling of videos[7, 6, 45, 62]. Vondrick et al.[93] analyse how to efficiently annotate video contents with a crowd with regard to costs, quality and performance. Based on a user study from 2013, they claim that typical crowdsourced micro-tasks are not suitable to annotate videos. Further they present an optimized video annotation framework for skilled MTurk users and automatic interpolation algorithms to optimize the performance. According to Vondrick one cannot solely rely on low-wage crowds to address massive video data sets. Intelligent annotation protocols were needed to achieve economical and qualitative results.

While Vondrick couldn't benefit from inexperienced workers, several projects have shown that receptive people can even succeed on complex tasks that concern scientific data. In neuroscience for example humans can help to map connectomes[25] by recognizing retinal neurons of mice in microscopical images. Sebastian Seung developed a game called *EyeWire* which addresses exactly that purpose. Formerly developed at the Massachusetts Institute for Technology and meanwhile in collaboration with the Seung Lab at University of Princeton the game has 200,000 players from 145 countries[26]. They have developed an artificial intelligence algorithm to help users to process gamified tasks. Since the algorithm still cannot perform fully automatically, users are asked to segment neurons inside given cubes measuring 4.5 micro meters.

[25] https://en.wikipedia.org/wiki/Connectome, *"A connectome is a comprehensive map of neural connections in the brain [...]"*

[26] http://blog.eyewire.org/about/

9 | Experimental Setup

We have discussed crowdsourcing for scientific use cases in Chapter 8. The most common crowd-based approach in science is to let people annotate feature points, bounding boxes or contours as well as descriptive labels and classifications. To gather such annotations, different types of annotation tools and protocols were presented. The complexity of micro-tasks reach from answering simple questions to sophisticated requests such as to segment neurons or to work on 3D protein models.
In this part of the thesis we analyse if different crowd user groups can achieve **highly accurate** object annotations that are **usable for automotive ground truth datasets**. Donath and Kondermann [25] were able to acquire motion annotations whose accuracy was sufficient to be used in automotive scenarios. However, they only evaluated their results with laboratory reference data and did not prove whether this approach can be applied on real automotive data. While Donath and Kondermann addressed optical flow, we will focus on object annotations. The most common annotations for this scenario are bounding boxes and contours. We primarily focus on the more sophisticated contour annotations which are needed to perform dense full scene labelling.

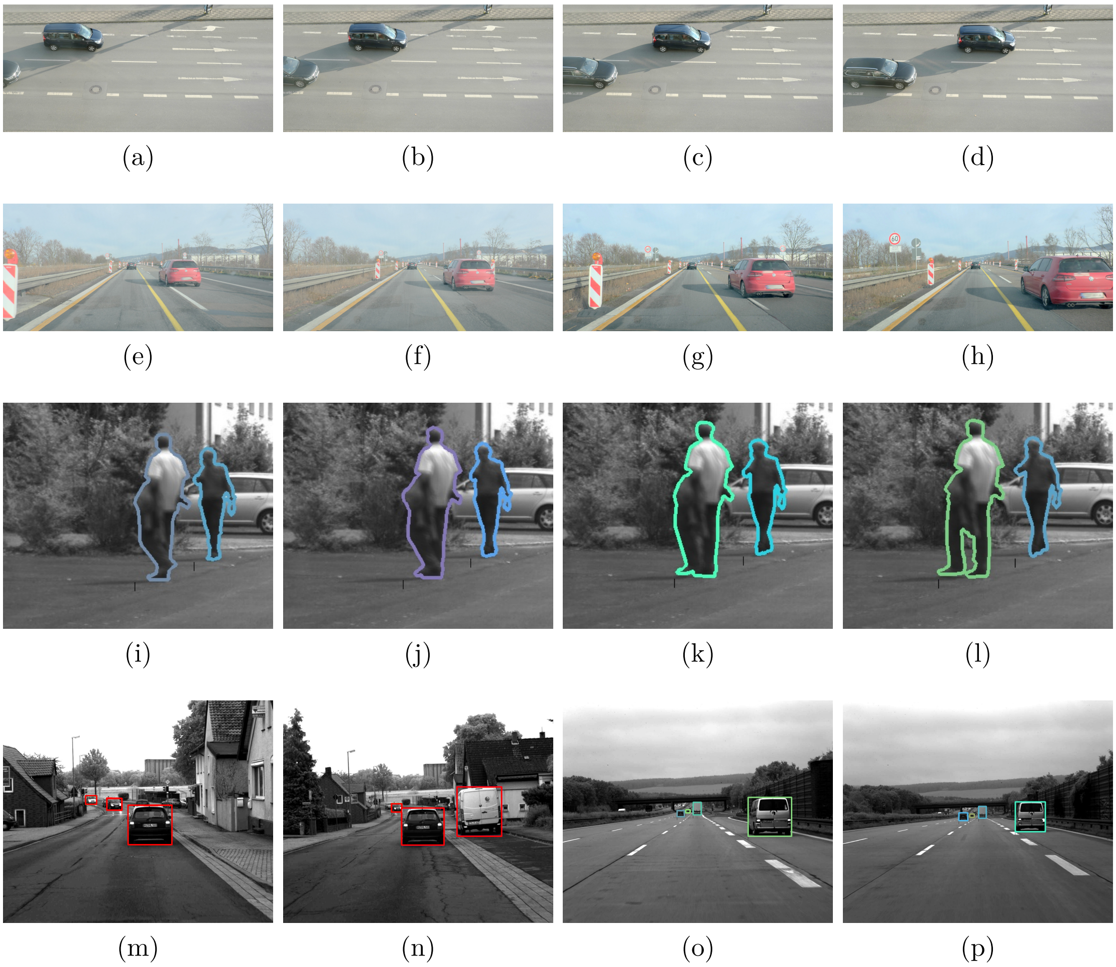
This chapter is structured as follows. After a quick motivation of using crowdsourcing for ground truth acquisition in 9.1 we present the evaluated test data in 9.2 and discuss expected annotation errors in 9.3. Section 9.4 presents performance indicators used to evaluate annotations. In 9.5 we introduce the user groups which have been evaluated. Finally we give a quick overview of our web-framework and user interface in 9.6. Our experiments and their results are presented in Chapter 9.

9.1 Crowdsourcing Ground Truth Acquisition

We have discussed ground truth datasets that rely on human annotations in Chapter 7 and crowd-based approaches to acquire annotations in Chapter 8. An extensive ground truth dataset for automotive applications can provide semantic labellings, object boundary annotations, information about reflections and surface conditions, depth information, stereo views and a motion analysis. To supply large datasets with annotations, we aim at gathering high-quality object boundaries by the use of crowdsourced micro-tasks. Annotations can be used for performance analysis of scene understanding and for research which requires large volumes of annotated images. Annotated object boundaries can also be used for further ground truth acquisition steps such as aligning 2D images to 3D scans.

Our main performance indicators to analyse crowds are the annotation quality and the time needed to process a data set. We compare the according results with those of experts. The motivation to collect annotations via crowds instead of experts is the scalability. The number of experts is strongly limited and thus, the time needed to process a huge data set could be unacceptably long. The opportunity to work with thousands of people at once sounds like a good idea. Well-chosen crowds could enable to process huge data volumes unthinkable to do with a limited number of experts. Furthermore, several publications have shown that results gathered by laymen can compete with those of experts [57, 97, 79, 1].

If the outcome did not satisfy our high-quality requirements, crowdsourcing could be taken as preliminary stage whose outcome is finalized by experts. This may be a feasible approach for complex tasks such as annotating object contours which is quite time consuming. Moderate contours gathered by crowds could be a good starting point for an experienced annotator reducing the overall processing time for the whole project. Furthermore, from an economical point of view it could be a huge advantage to obtain poorer data within hours instead of perfect data within weeks or months. If one relied on a group of a few experts, the working time per task is essential since the data set basically is processed serially. However, this does not apply for large crowds of laymen. When evaluating the processing time it is not only of interest to optimize working times for single micro-tasks but also the overall time needed to process a dataset. Even slowly processed tasks could be feasible if many of these tasks can be run parallel. For the same reason, it could be viable to break down a complex expert task into many simple tasks.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure 9.1: *Our **test data** consists of three sequences.*

9.2 Reference Data

For our evaluation we use differently challenging image sequences containing vehicles and pedestrians that need to be annotated. That data visualized in Figure 9.1 is similar or equal to data captured by real automotive systems in which we also had insight. Hence the chosen sequences are representative especially for such applications.

Sequence 1. Footage similar to typical image sequences captured by automotive or traffic surveillance systems (20 frames overall, Figure 9.1a-9.1h). This sequence contains color images which can be advantageous for human perception.

Sequence 2. A subsequence of footage of the *HCI Stereo Ground Truth Data Set*[1]

[1] http://hci.iwr.uni-heidelberg.de/Benchmarks/document/StereoErrorBars

(10 frames overall, 9.1i-9.1l). These frames basically contain pedestrians. Contrary to sequence 1 this data is captured as greyscale images. Thus distinguishing human bodies from background can be very difficult at some areas which might have been easier if images had color channels.

Sequence 3. A subsequence of similar footage containing vehicles in city traffic and on motorways (53 frames overfall, 9.1m-9.1p).

We preprocessed test images by enhancing the local contrast as shown in Figure 9.2. This method reveals many details that otherwise would have been hidden. However real world footage never is perfectly illuminated, contrasty and sharp. Figure 9.3a visualizes an image region where human perception is stretched to it's limits. The shaded area below the car is hardly distinguishable from the car's final bezel. Figures 9.3b and 9.3c show an optimized high-resolution ground truth version of this image area. This example points out that the expectable quality of human annotations is not only limited by the performance of our annotators but also by our image data. Cases like the example above are common and we discuss some of these in our further analysis.

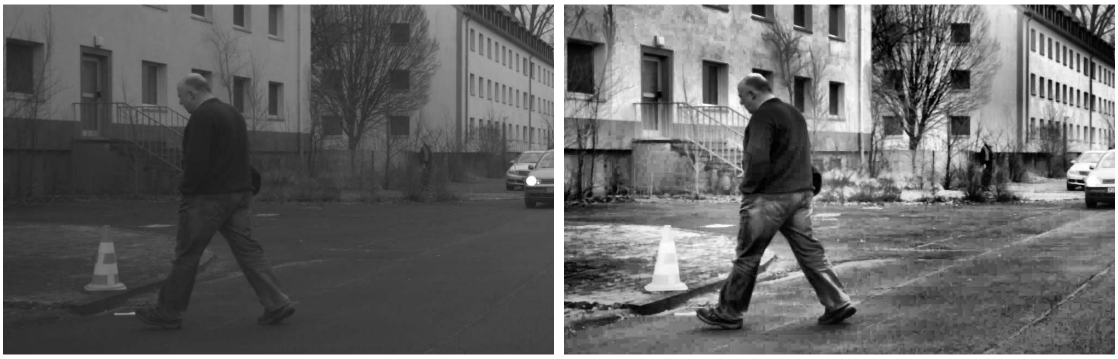


Figure 9.2: *Preprocessing image data can strongly enhance crowdsourced recognition tasks. In contrast to low-level vision algorithms, humans can easily interpret noisy contrast-enhanced images.*

9.3 Ground Truth and Error Analysis

For our analysis, we want to evaluate annotations with regard to the best results achievable. Initially, we annotated the data ourselves. For this purpose and in contrast to the workflows to be evaluated, we could access additional tools and access images with higher resolutions to ensure most accurate results possible. While annotators participating in our evaluation were provided with a single frame only, we could access the whole sequence at once. Knowing previous and subsequent frames appeared to be an important information. However the annotated reference data is

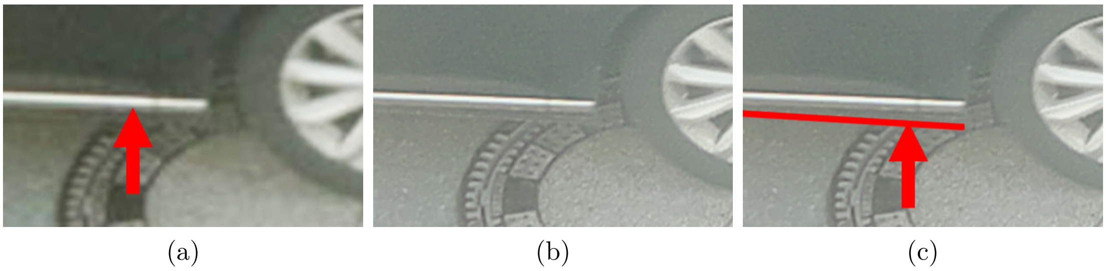


Figure 9.3: *In some cases, pixel-accurate annotations are not possible since image data does not supply sufficient information to guess the correct object shape. Figure 9.3a visualizes such a case where at first sight the bottom end of a car seems to be at the bright horizontal bezel. Figure 9.3b is the high resolution version used for ground truth generation only. The bottom end appears, albeit hardy distinguishable, to lie beneath the formerly assumed shape.*

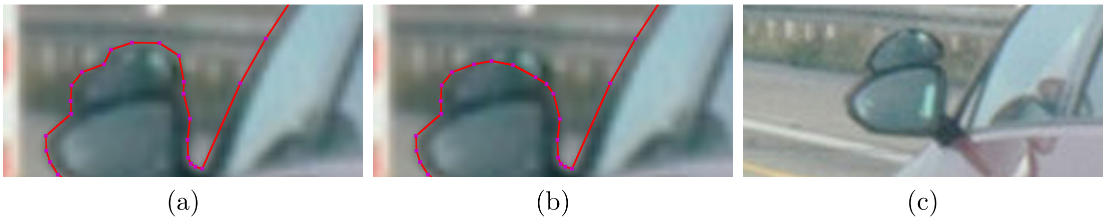


Figure 9.4: *Figures 9.4a and 9.4b show two possible annotations including a side mirror in front of guardrail. Annotation 9.4a assumes the upper part to belong to the mirror while annotation 9.4b excludes that part assuming the vertical lines to be guardrail posts. The subsequent frame in 9.4c resolves this ambiguity providing a better resolution and a clearer contrast between foreground and background.*

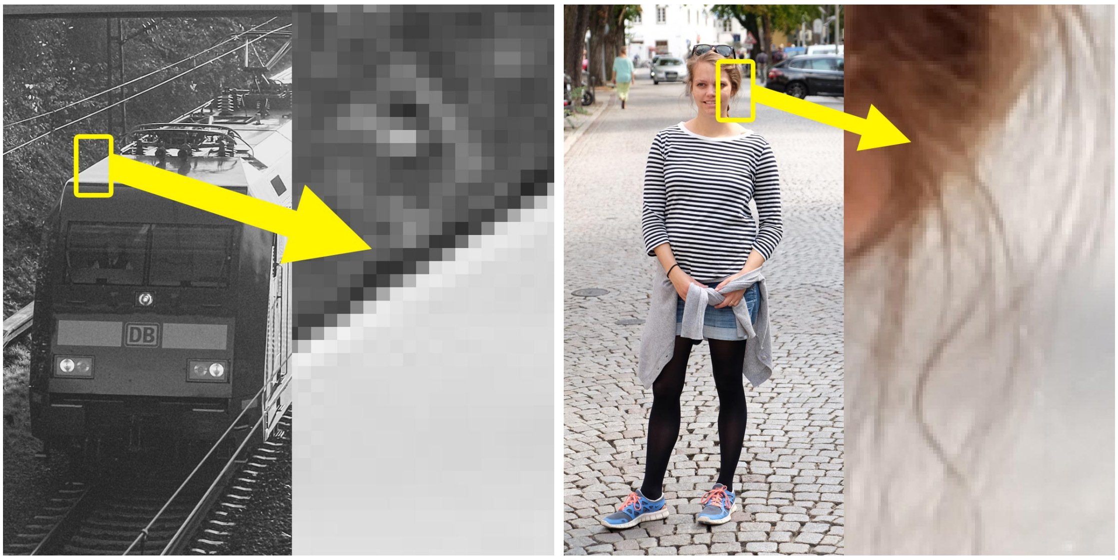
faced with the same error types as annotations collected from our crowds. Errors which have to be taken into consideration when evaluating annotations are:

1. **Systematic errors**
   Systematic errors occur at image areas like that of figure 9.3 where the image information induces wrong perceptions of object shapes. During data inspection on subpixel level, we became aware of many of such ambiguous and hard to comprehend object parts for which information of single images may not be sufficient for confident annotations. An example is shown in figure 9.4. Subfigures 9.4a and 9.4b show two possible annotations including a side mirror in front of a guardrail. Annotation 9.4a assumes the upper part to belong to the mirror while annotation 9.4b excludes that part assuming the vertical lines to be guardrail posts. Figure 9.4c shows that image region for the subsequent frame which resolves this ambiguity providing a better resolution and a better contrast between foreground and background.

Images do not show accurate object borders in general. Borders can cover multiple pixels whereas our annotations have an infinitesimal width. In such cases, our perception for object boundaries differs and there are several valid solutions for an annotation. Ambiguities especially occur for blurred borders due to out of focus shots or motion blur. Further reasons are image artefacts caused by compression or chromatic aberration. We do not consider these deviations as an error, but we have to take them into account for our evaluation.

2. **Random errors**
   Annotations also have random errors which occur by imprecise user inputs. For example the precision of annotations is limited by the way our annotation tool is used. The variance of annotations given by experts can be quit small (less than 1 pixel for unambiguous borders). On the other hand, random errors caused by laymen which never used such tools before are much taller.

3. **Inaccuracies related to commitment**
   This is not an issue for reference data generation since our goal is to work as precise as possible. However, the major reason for inaccurate annotations contributed by participants with weak incentives is their little commitment. In these cases, commitment related errors weight much more than any other error source.

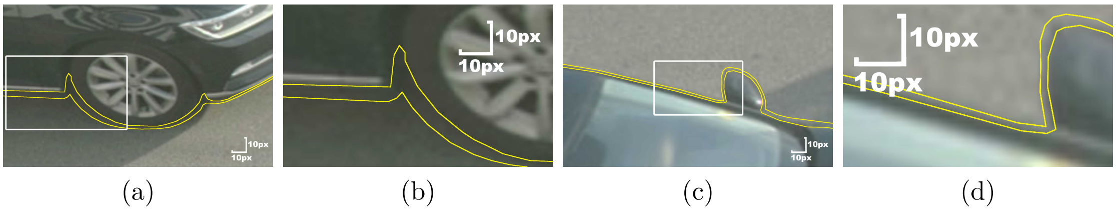


(a) Unambiguous boundary

(b) Ambiguous boundary

Figure 9.5: *The definedness of object boundaries strongly varies. A good example are hairs where an explicit contour mostly cannot be defined.*

Considering these error sources, it would not be suitable to compare annotations with unique reference annotations. By annotating the same objects several times, our average annotation precision appeared to be less than 0.3 pixels. However these values are related to our perception of object boundaries and do not take into account that other persons find different valid solutions. Since we want to analyse how crowds perform in comparison to the best results possible, we somehow have to address deviations caused by perception. Therefore we define an inner and outer object boundary which represents the tolerance for correct annotations (see figure 9.6 and 9.5). The mean width of that defined area is 1.7 ± 0.6 pixels (first quartile: 1.2px, third quartile: 2.1px). During careful inspection of our data, we assume to have excluded any systematic error from these annotations. For our further evaluation we do not consider the random error of the inner and outer contour. In consideration of the result quality collected from our crowds, our own random error carries no weight.



(a) (b) (c) (d)

Figure 9.6: *We define an inner and outer object boundary since it turns out that object boundaries are ambiguous. In our further evaluation we consider all annotated pixels inside these boundaries as correct annotations.*

The distance between outer and inner object boundary is related to image quality and complexity of a scene. Image artefacts, blurred objects, weak contrasts or translucent objects are reasons for ambiguous object borders. Our reference data can be considered to have an intermediate complexity concerning these ambiguities. Besides our evaluation we also worked with lots of other representative data sets. Some of them consist of more complex image data caused by artefacts and low resolution files. On the other hand and in accordance with all data sets we have seen, we do not expect automotive scenes to be significantly easier to annotate as the presented images. The determined average uncertainty of 1.7 ± 0.6 pixels for manually annotated object boundaries seems to be a representative value for such image scenes.

9.4 Performance Indicators

During our evaluation we focus on four different performance indicators to rate and compare our crowds. These are:

1. We evaluate the **shape consistency** of the annotated filled shape A and the

filled ground truth shape GT via the *Jaccard distance*:

$$\text{Jaccard} = 1 - \frac{\mid A \cap GT_i \mid}{\mid (A \cup GT_o) - (GT_{o-i}) \mid} \tag{9.1}$$

For its calculation we do not consider the area GT_{o-i} between outer and inner object boundary GT_o and GT_i. The *Jaccard distance* is a quick and computationally performant indicator whether two annotations match at all. Though, it reduces the correlation between shapes to a single value and does not provide detailed information about the deviation from reference data. A uniform shape error can map to the same *Jaccard distance* as a single but much taller shape error, though it would make a huge difference regarding the usefulness of an annotation.

2. Complementary to the Jaccard distance we also analyse the **absolute distances** between annotation pixels and reference pixels. For this purpose we take the pixel-wise *Hadamard Product* between the mask of annotated pixels C and the distance map of GT_{o-i}. The resulting image D contains the distances for all pixels of C with regard to GT_{o-i}.

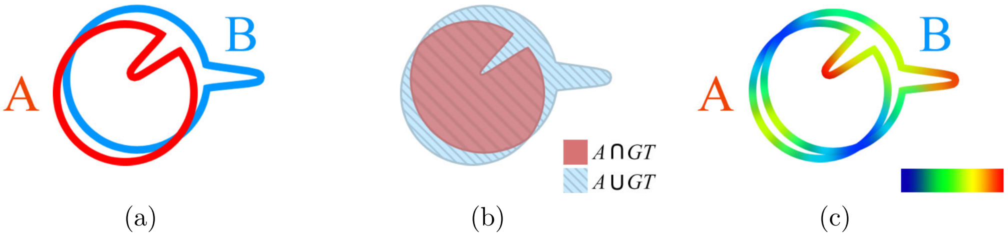
$$D_{C,GT} = C \circ \text{DistanceMap}(GT_{o-i}) \tag{9.2}$$

This approach has its weakness for concave objects. Figure 9.7c visualizes the distances of two concave shapes. If one calculates distances of contour A with regard to B in one direction only, partially concave shapes of B may not be taken into account. An approximation would be to determine $D_{A,B}$ and $D_{B,A}$ as well. However, comparisons between one-directionally and bi-directionally calculated distances show that the difference is negligible for our set of collected annotations. Additionally, concave shapes would also be observed by the Jaccard distance. Thus, for reasons of simplifications we restrict the calculation of absolute distances to equation 9.2.
We natively store annotation coordinates as float values. Before rendering those values to binary masks, we scale up these coordinates to ensure subpixel accurate results though working with integer coordinates (figure 9.8). The scaling factor is chosen such that the systematic error caused by this mapping is negligible for our error discussion.
In Chapter 9 we primarily evaluate absolute distances by determining the percentage of contour points lying within a **2 pixel** and **5 pixel threshold** compared to ground truth. These are typical values requested by many reference data projects. According to Geiger et al. [31], an accuracy of 3 pixels is sufficiency for automotive applications.

3. We determine the number of **false positive (FP)** and **false negative (FN)** annotations. Due to a vague instruction, annotators may had different interpretations which objects to annotate. This mainly was caused by different

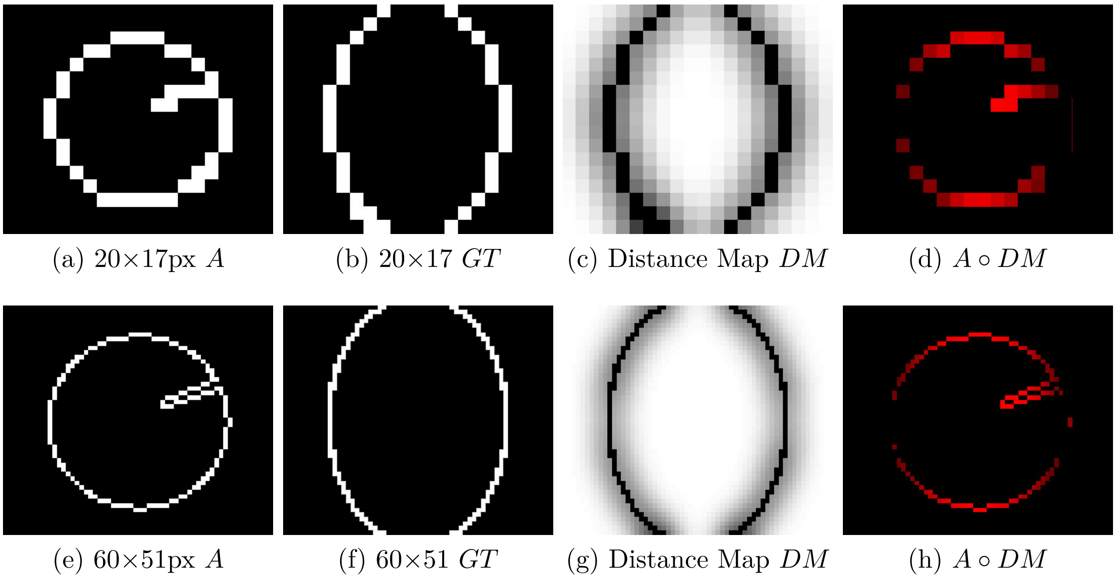
requirements of former projects. Thus we determine **FN** for ambiguous and unambiguous objects as well. Ambiguous objects in this case are tiny vehicles (width or height less than 17 pixels) and vehicles that are cut by image borders.

4. We analyse the **processing time** per annotation which allows us to discuss the efficiency of different crowds compared to experts.



(a) (b) (c)

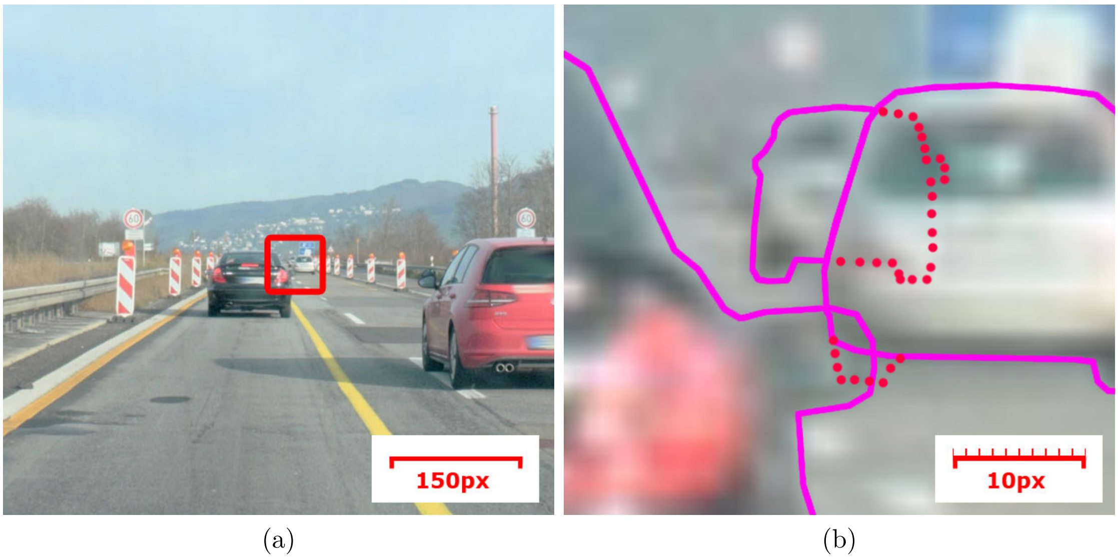
Figure 9.7: *The Jaccard similarity coefficient $\frac{A \cap GT}{A \cup GT}$ describes the overall similarity between two shapes but does not describe error characteristics. The absolute distances between two contours depend on the direction of distance calculation. Figure 9.7c visualizes the distances calculated for $A \to B$ and $B \to A$.*



(a) 20×17px A (b) 20×17 GT (c) Distance Map DM (d) $A \circ DM$

(e) 60×51px A (f) 60×51 GT (g) Distance Map DM (h) $A \circ DM$

Figure 9.8: *These figures illustrate the comparison between two contours. Contours which natively are saved as polygons are rendered to binary masks. The second mask is transformed to a distance map. The intersection of first mask and distance map contains the distances for every contour pixel. Since working with rendered integer coordinates does not fulfil subpixel-accurate annotations, we scale up floating point contour coordinates before rendering them to binary masks.*



(a)

(b)

Figure 9.9: *Requiring unique annotations for occluded objects is inappropriate for our analysis since some workers try to estimate the actual shape while others annotate visible edges only. Thus, our inner and outer annotation tolerance addresses occluded parts also.*

9.5 Choosing Appropriate Crowds

While working with crowdsourced computer vision applications we gained experience with several characteristic crowds. Thus the spectrum of participants for our following analysis reach from experts which were trained for specific tasks to laymen that are unexpectedly confronted which such tasks. We evaluate results gathered from the following crowds:

1. **Experienced and trained users**
   For evaluation purposes, our crowdsourcing tasks are processed by a small group of experienced and trained experts. We expect experts to solve our tasks fast and with highest precision achievable. Results should be superior compared to other participants.

2. **Amazon Mechanical Turk**
   We worked with the *Amazon Mechanical Turk* crowd which we introduced in Section 8.2. This is a common approach in this field of research. Results gathered by this crowd are not adequate in general. However, Amazon Mechanical Turk workers (*MTurkers*) can be assigned a *quality level* attribute. By use of that attribute we sort out superb participants by gold standard tests to ensure acceptable results. We evaluate crowdsourcing workflows for the filtered (*superb*) and unfiltered (*all*) MTurk crowd.

3. **A gaming crowd**
   We access the crowd of an online game provider. Players of these online games can optionally earn in-game achievements by solving external tasks. A common way to do so is to watch video ads. Alternatively the content provider also places our computer vision tasks. As a quid pro quo the content provider is paid for each task request. The challenge to work with such crowds is to gain the attention of participants. The attention and receptivity of players that expect a video ad when requesting a task is low. While players have a passive role when watching videos, we need them to become active for our purposes. Also their incentives to earn in-game achievements may be too weak for advanced tasks. Many users of gaming crowds only spend the minimum required effort to receive their in-game reward. The majority can only be expected to solve simple tasks. Though, there are a vast number of gaming crowds with billions of players. If only a small percentage of those users performed well, this could be a great chance for huge projects with hundreds of thousands of images that need to be processed.
   If it should emerge that laymen can achieve reasonable results, crowdsourced tasks can be a serious alternative to video ads for content providers. Video ads can only be shown limited times to a single user. Afterwards the owner of that advertising won't pay for further hits. Thus, the opportunities for content providers to earn money via online ads are limited by the number of users and the number of different available ads. Additionally, ads mostly are limited to specific regions and countries. Concerning this matter, to rely on crowdsourcing tasks instead of video ads can be a huge benefit for content providers. Contrary to online ads, the number of crowdsourcing tasks per user basically is limited by the number of overall tasks and hence by the size of the related project. These tasks do not necessarily need to be restricted to specific regions and languages and can be provided worldwide.

It has to be noted that the presented results from MTurkers and gaming crowd participants are random samples only. The quality can vary for different day times or even the time of the year can make a difference. A further parameter that can influence the results is the nationality of a user when it comes to questions about local traffic regulations. Results gathered from the mentioned crowds partly could be different when running the experiments on other conditions. Though, the presented results conform to what we experienced over two years of research.

9.6 Web-Framework and User Interface

We developed a web-based responsive user interface which enables us to provide different micro-tasks via a website. Thus we were able to embed our micro-tasks into other websites such as *Amazon Mechanical Turk* and the gaming platform mentioned above.

For this evaluation we basically used our contour creation workflow. Users were asked to add contour annotations to all vehicles or pedestrians of one frame. First we gave a quick introduction to the next task. Afterwards we provided the following toolchain to add contours.

Basic annotation steps for contour creation:

1. **Contour Creation** (Figure 9.11a)

    a) *Click Approach*: Draw a contour by subsequent mouse clicks creating turning points.

    b) *Draw Approach*: Press and hold the mouse button to enable freehand drawing.

    The contour is completed once the starting point is clicked again or alternatively once the freehand drawing cursor moves over the starting point position.

2. **Contour Correction** (Figure 9.11b)

    After completing the creation step, the turning points are visualized. Now there are several options for fine adjustments.

    - Turning points can be moved or deleted.
    - The contour can be moved as a whole.

    This correction step was optional. Contributors could proceed without correcting their contours.

We made huge efforts to create an effective annotation tool. However, as for many user interfaces there are many possibilities for further optimizations. Even small changes of our user experience design could mean a reasonable impact to the results. While this topic is not addressed in our evaluation, we keep it in mind for future work.

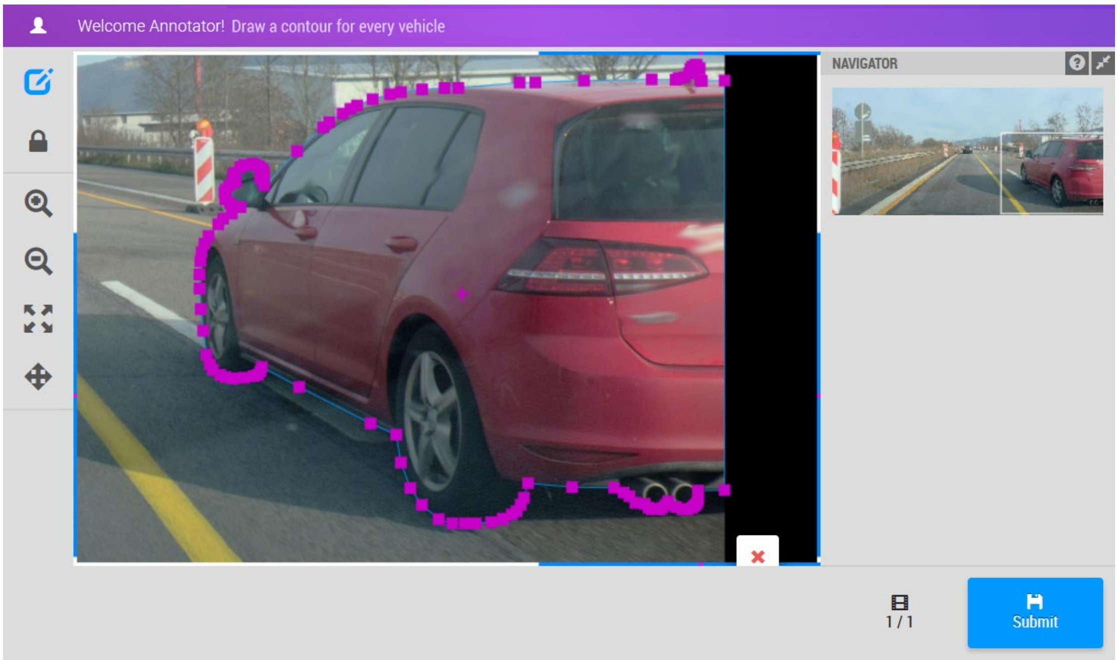


Figure 9.10: *We developed a web application which provides multiple annotation tools and workflows. This figure shows our contour tool.*

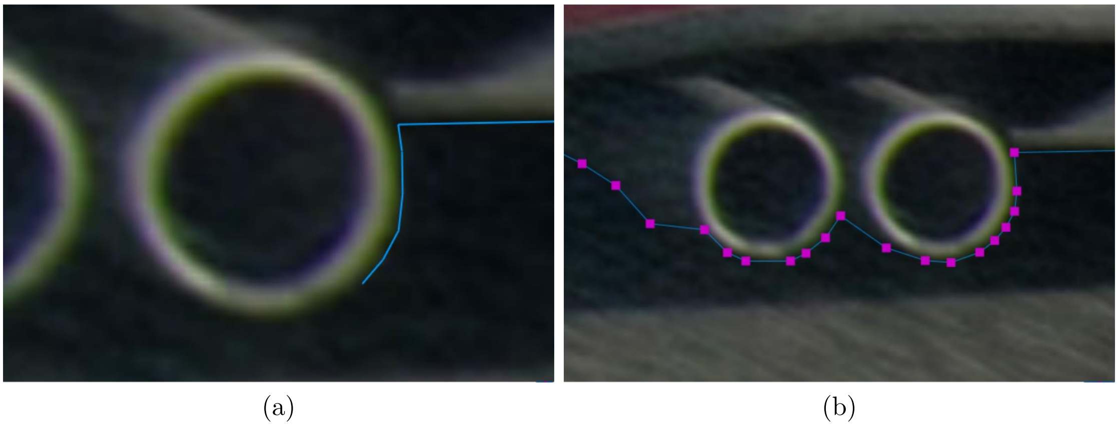


Figure 9.11: *Contour Creation: A contour can be created by freehand drawing or by placing single turning points (9.11a). After closing the loop, every single turning point can be edited again (9.11a).*

10 | Experiments and Results

In the previous chapter we discuss our experimental setup, our reference data and the performance indicators for our analysis. In this chapter we present the experiments which we assembled for this analysis on crowdsourced human annotations. We discuss four different annotation protocols, compare their required annotation times and exploit multiple annotations per object to determine confidence measures.

First we evaluate single annotations without further postprocessing (10.1). Afterwards we analyse several advanced annotation protocols. In 10.2 we present results achieved with majority votes. For this purpose we process multiple results per frame. Alternatively we analyse an iterative annotation protocol, where several users directly contribute to the same result (10.3). In 10.4 we present an extension to the iterative approach by using propagations from previous frames as an initial guess. After qualitative evaluations of these approaches, we compare working and processing times in Section 10.5. Finally we give a discussion about confidence measures for annotations in 10.2.3.

Contents of this chapter

10.1 Evaluation of Single Annotations

As a first analysis, we evaluate single contour annotations. Participators are asked to **add contours to all visible vehicles**. Every task addresses one frame. We run these tests on *sequence 1* which keeps 20 frames overall. Thus we set multiple independent tasks for each frame to collect a reasonable amount of annotations for our evaluation.

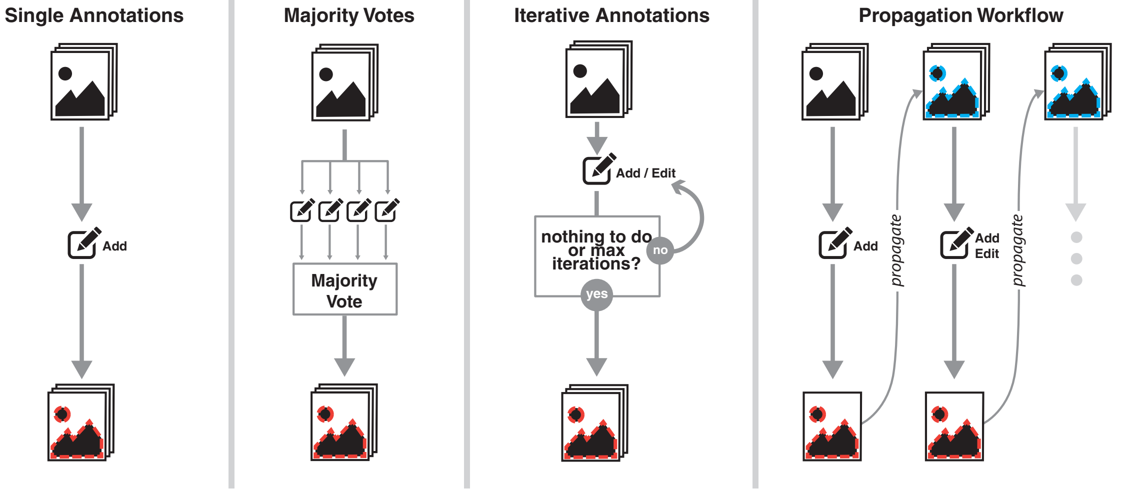


Figure 10.1: *Overview for different annotation protocols: We start our analysis with an evaluation of single annotations. In the following sections we discuss majority votes, where several contributions per frame are merged into one result. In addition to this, we discuss an iterative approach were annotators have the opportunity to adjust annotations of their predecessors. As an extension to this approach, we use propagation algorithms to create initial annotations for subsequent frames.*

We collected results from six experts which are familiar with automotive scenes but had not seen our test images in particular. Experts were asked to submit at most five contributions each. Every frame was only processed once, which is a reasonable approach when working with experts due to labour costs and an adequate expected annotation quality.

For the evaluation of MTurkers we ran two projects. The first was accessible for all MTurkers. The second MTurk project was restricted to selected (superb) MTurkers only. Payments were based on Ipeirotis et al.[43] who analysed typical MTurk wages. The superb MTurk subgroup was assembled by workers that did well in previous projects. Initially that were about 60 persons. However the nature of the MTurk crowd is that although there are thousands of registered MTurkers[1] only a

[1] Over 500,000 registered users in 2012, https://forums.aws.amazon.com/thread.jspa?threadID=58891

few of them will work on task types provided by our requests. Effectively, only 10 MTurkers of this subgroup contributed to our first evaluation. It was difficult to achieve a homogeneous distribution of participants since once tasks are spawned, these tasks are processed by single users immediately. Unfortunately it was not possible to limit the number of tasks per user on MTurk. To overcome results strongly biased by single users, we activated our projects at different daytimes and partly discarded results of single users when their number of annotations differed tremendously from the average value.

The results of our gaming crowd were only collected from annotators that succeeded on a former gold standard test. Therefore we evaluated the *symmetric difference* $A\Delta GT$ between annotation masks A and ground truth masks GT. Annotations had to satisfy the following condition:

$$\frac{\mid A\Delta GT \mid}{\mid GT \mid} \leq 0.1 \tag{10.1}$$

We approximated GT to be the outer ground truth shape GT_o. All participants from our gaming crowd whose annotations satisfied equation 10.1 were granted access to the actual evaluation tasks.

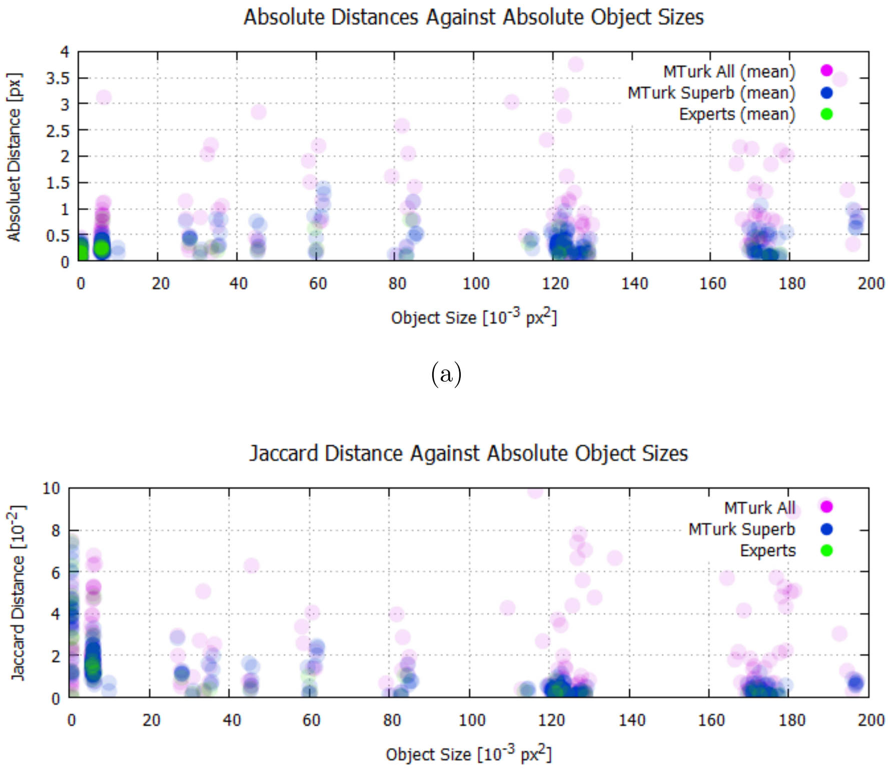
10.1.1 Error Discussion

Table 10.1 lists the resulting performance indicators for each crowd. The Jaccard distance values represent the average amount calculated for all collected annotations. Absolute errors of contour points are kept in D. We primarily discuss its distribution and percentiles since mean values are not that informative for a distance measure with heavy outliers. Furthermore we provide the percentage of inliers in respect to a **2 pixel** and **5 pixel threshold** compared to ground truth. These are typical values requested by many reference data projects. The given false positive and false negative values are related to entire objects and specify the number of missed annotations and the number of falsely annotated objects.

The Jaccard distance turns out not to be a good measure when comparing differently-sized objects. The reason is that annotation errors are quite independent of related object sizes which can be seen in figure 10.2a. Here the absolute distances are plotted against the related object size. Values are quite similar for all object sizes. However, having size-independent absolute errors makes the relative Jaccard distance measure to be size-dependent. Figure 10.2b visualizes Jaccard distances against the related object size. The distribution of Jaccard distances for small objects is widespread and values are big compared with those of taller objects. As an example, figure 10.3 shows two objects whereas 10.3a shows significant absolute errors while the annotation in 10.3b only slightly deviates from ground truth. Though the

	Crowd 1	**Crowd 2**		**Crowd 3**
	Experts	*MTurk Superb*	*MTurk All*	*Gaming Crowd*
Received Contours	48	305	476	864
Annotators	6	10	61	182
Jaccard Distance $[10^{-2}]$	1.4 ± 1.8	1.2 ± 1.4	2.8 ± 3.7	8.7 ± 5.2
D_{mean} $[px]$	0.3 ± 0.6	0.4 ± 0.6	1.8 ± 1.9	6.7 ± 5.7
D_{median} $[px]$	0	0.1	1.2	5.5
$D_{percentile(5)}$ $[px]$	0	0	0	0
$D_{percentile(25)}$ $[px]$	0	0	0	1.0
$D_{percentile(75)}$ $[px]$	0.3	0.7	2.3	12.1
$D_{percentile(95)}$ $[px]$	4.2	4.0	22.0	33.0
Inliers ($\leq$2px)	96.8%	96.1%	78.5%	27.7%
Inliers ($\leq$5px)	99.5%	99.5%	89.6%	50.6%
FP	0	0	0.5%	0.9%
FN_{all}	5.9%	33.9%	45.5%	25.8%
$\text{FN}_{>50px,uncut}$	0	0.3%	9.5%	1.2%
$\text{FN}_{17px<size\leq 50px}$	0%	11.6%	13.7%	3.6%
FN_{cut}	2.0%	10.6%	11.7%	18.8%
$\text{FN}_{size\leq 17px}$	3.9%	11.3%	10.7%	2.3%

Table 10.1: *This table lists performance indicators of our crowds calculated for contour annotation tasks. As could be expected, best results were achieved by experts. Superb MTurkers also achieved highly-precise annotations but missed more objects to annotate. The precision of the unfiltered MTurk crowd was significantly lower and almost every second object was missed. Annotations of the gaming crowd were pretty rough while their false negative rate even was better than that of superb MTurkers.*



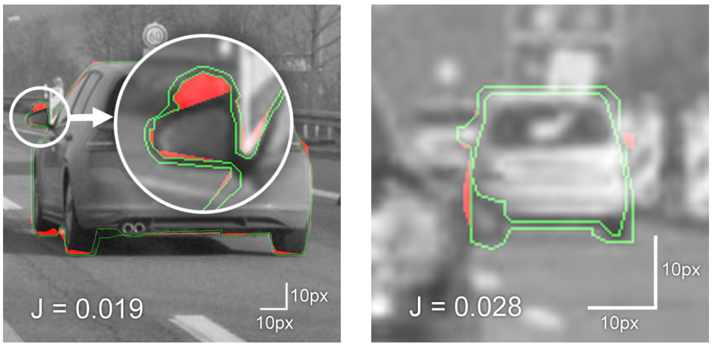
(a)

(b)

Figure 10.2: *The Jaccard distance compares shapes relatively. However, annotation errors are widely independent from object sizes (10.2a). The distribution of Jaccard distances with regard to object size is visualized in figure 10.2b. Since Jaccard distances are inappropriate to compare differently-sized annotations, we mainly evaluate absolute distance measures.*

Jaccard distance of 10.3b is 50 percent taller than that of 10.3a. With regard to the size dependency of Jaccard distances, this measure is inappropriate to compare annotations of differently-sized objects.
Furthermore, using the average Jaccard distance to compare different crowds is questionable. A crowd which leaves small objects unannotated can achieve a better overall Jaccard distance than a crowd annotating all objects even if the latter crowd achieves smaller absolute errors. Hence complete results would be devalued against incomplete results given that all results are erroneous.

For our further evaluation we focus on absolute distances and detection rates. An error distribution of these distances is visualized in Figure 10.4a. Figure 10.4b shows the percentage of correct annotations for different error thresholds.



(a)

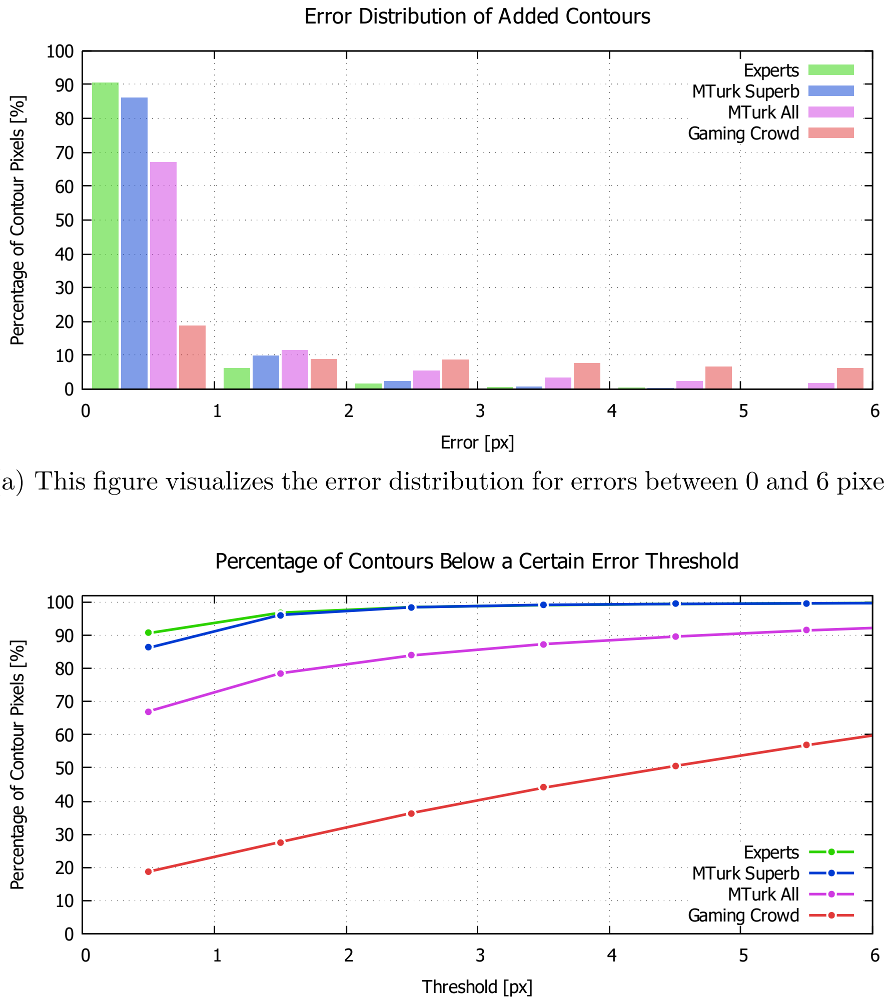
(b)

Figure 10.3: *The annotation in 10.3b has a Jaccard distance which is 50% taller than the one in 10.3a though 10.3a has much more absolute errors. Thus, Jaccard distances for objects with different sizes cannot be compared.*

10.1.2 Annotation Precision

When working with reference data, many data set creators require an accuracy of 2 to 5 pixels with at most 5% outliers. Within these thresholds the annotation precision of results from experts and those from superb MTurkers is nearly the same. 96.8% of annotations from experts and 96.1% from superb MTurkers lie within the 2 pixel tolerance. Regarding the 5 pixel threshold both groups achieve an amount of 99.5%. The chosen group of experts was very familiar with our tool and with the type of image data. We can assume that these results are the best that can be expected from single annotations.
In comparison, the unfiltered MTurk crowd performs worse with 78.5% less than 2 pixel and 89.6% less than 5px deviation from ground truth. Thus, without further ado the results of an unfiltered MTurk crowd do not satisfy the annotation accuracy requirements mentioned above. The gaming crowd annotations are way too imprecise to be used for high-quality reference data sets. Only 50.6% of annotated pixels lie within the 5 pixel threshold.

To get a better impression of our results, Figure 10.5 provides characteristic annotation examples for each crowd. In accordance with the values of table 10.1, contours annotated by experts or superb MTurkers deviate at most a few pixels from ground truth (10.5b). The results of the unfiltered MTurk crowd clearly show some heavily deviating annotations (10.5c). The majority however matches with the actual shape. The number of strongly deviating annotations is much higher for the gaming crowd results (10.5d). Some of them can be counted as heavy outliers. These annotations can at most deliver information about the rough object position but not about the shape. The superposition of all annotations however indicates the actual shape.

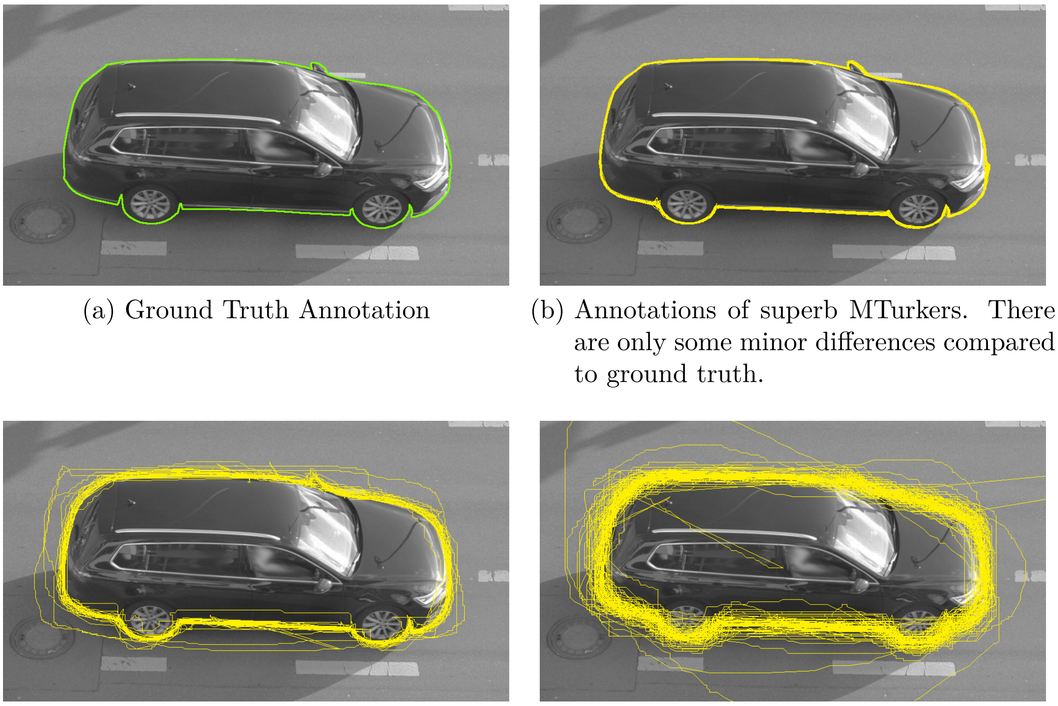


(a) This figure visualizes the error distribution for errors between 0 and 6 pixels.

(b) This figure shows the percentage of correct results in respect to increasing thresholds. While 99.5% of results from experts and superb MTurkers deviate less than 5 pixels from our reference data, this is only true for 50% of the gaming crowd results. Regarding the unfiltered MTurk crowd, 89.6% of results lie within the 5 pixel threshold.

Figure 10.4

When creating a reference data set, several demands can be stated. With regard to the images of Figure 10.5 the data should be free from extreme outliers. For most applications a homogeneous deviation would be better than a partly smaller deviation coupled with heavy outliers. If we worked only with one result per image we couldn't trust unfiltered MTurk or gaming crowd results since though the majority represents the correct shape, we could not accept its outliers. Hence, to succeed with these crowds we would need to exclude annotations with heavy errors. We discuss some methods addressing this issue in the following Section 10.2.



(a) Ground Truth Annotation

(b) Annotations of superb MTurkers. There are only some minor differences compared to ground truth.

(c) Annotations of all Mturkers. Some results extremely deviate from ground truth but the majority fits the actual shape quite well.

(d) Annotations of the gaming crowd. Results are imprecise and contain some heavy outliers. Though the sum of all annotations clearly matches with the actual shape.

Figure 10.5

10.1.3 False Negatives Rates

A big difference between results from experts and those from all other participants is the number of missing annotations $\mathbf{FN}_{all}$. Besides the overall percentage of false negatives FN_{all}, we categorized missing annotations in various characteristic subgroups. We determined the percentages for missing objects that are cut at image borders (FN_{cut}), objects that are smaller than 17 pixels ($\text{FN}_{size \leq 17px}$), objects that are taller than 17 pixels but smaller than 50 pixels ($\text{FN}_{17px < size \leq 50px}$) and finally the percentage of all other false negatives ($\text{FN}_{size > 50px, uncut}$).

While experts missed 5.9% of all subjects, superb MTurkers performed quite worse and didn't annotate every third object. Unfiltered MTurkers missed even every second object, the gaming crowd every fourth. Obviously such an amount of false negatives wouldn't be acceptable for a reference data set. Most of the unannotated vehicles are either cut at image borders or they are very small and occluded by other objects. Our participants may have been biased from previous project instructions. In previous projects, we only asked for fully visible vehicle rears. It is hard to tell whether they really overlooked these objects or whether they ignored them due to old instructions. A further plausible and important reason for false negatives is a weak incentive in regard to task completion. It seems that many participants adapt their effort to the minimum amount which is needed to create acceptable results. A natural way to do so is to annotate the obvious objects only, since we didn't require a minimum number of annotations. Technically, participants could have even submitted empty contributions. However, the used platforms are known to exclude participants with unacceptable results which might be the reason for nearly no empty contribution. The percentage $\text{FN}_{size>50px,uncut}$ represents missing annotations for objects that are taller than 50 pixels in width or height and which are not cut by image borders. We exclude that these values arise from misleading instructions and misinterpretations. Hence, even if FN_{cut}, $\text{FN}_{size\leq 17px}$ and $\text{FN}_{17px<size\leq 50px}$ would have been caused by misunderstandings and we assumed such errors to be avoidable in future, $\text{FN}_{size>50px,uncut}$ would still be the minimum expectable amount of false negatives. Experts did not produce wrong results regarding these measures. Superb MTurkers missed 11.6% of such unambiguous objects. The worst results were produced by the unfiltered MTurk crowd with $\text{FN}_{size>50px,uncut} = 9.5\%$. The percentage of missed annotations of unambiguous objects was pretty low for the gaming crowd. While they performed worse with regard to annotation precision, only 1.2% of unambiguous objects were not annotated.

10.1.4 Summary

The overall performance of the presented crowds in regard to annotation precision and false negatives does not satisfy our requirements for a reference data set. The mentioned demand of 2 to 5 pixels annotation accuracy and at most 5% outliers seems to be quite tough and roughly corresponds to the performance which can be expected from experts. Even experts achieved a false negative rate of 5.9 percent. However, we only received 16 contributions containing 48 annotations from experts. These results let us estimate the quality achievable by experts but the number of contributions is too low for a reasonable estimate of false negative rates.
Superb MTurkers can achieve a similar annotation precision as experts but fail at the demanded maximum false negative rate. The unfiltered MTurk crowd cannot satisfy the required precision and additionally produces a very high false negative rate. False negative rates of the gaming crowd are astonishing since they are quite low while their annotation precision is far away from acceptable.

The overall outcome emphasizes not to rely on single results at all. While the annotation precision of experts and superb MTurkers seems acceptable, both could benefit from advanced annotation protocols with regard to false negative rates. Single results from unfiltered MTurkers and gaming crowd participants are not usable for our purpose. In accordance with the idea of the *"wisdom of the crowds"*[84], we analyse how combining multiple contributions per frame can ensure error robustness. We discuss such methods in the next section.

10.2 Majority Votes

In the previous section we compared each user contribution separately. According to results discussed above, it makes little sense to rely on one contribution per frame since heavy outliers and tall false negative rates make results unusable for our application. In the next sections we analyse more sophisticated methods that rely on multiple contributions. We start with an evaluation of majority votes in this section.

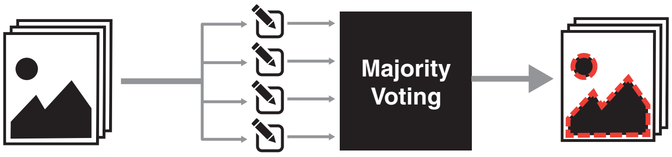


Figure 10.6: *Majority votes merge independent contributions to a single result.*

To calculate majority votes we need to collect multiple annotations per object, which we already did in the previous experiment. Afterwards, all annotation candidates for a specific object have to be determined which means we need to solve a cluster problem. Typical clustering methods for points clouds are the *DBSCAN* algorithm [72] or the *k-means* algorithm [55]. However, since we are working with shape annotations, we can use simpler approaches to determine majority vote candidates. Our test sequence has many clearly separable objects.To match contours, we use the *Jaccard Distance* (equation 9.1) and accept two compared annotations as a match if their distance comes below a given threshold. This threshold can be set to a relatively high value for sequences without dense or strongly overlapping objects but has to be kept low for crowded scenes to avoid mismatches. Alternatively and based on former measures we could also match contours by comparing its distance maps. For reasons of simplification we stick to the Jaccard Distance approach.

Once we found all presumable annotation candidates for an object, we sum up the according distance maps. Afterwards, annotations are rated by its intersection with the overall distance map. The resulting cost w_j is given by:

$$w_j = n_j \sum_{x,y} (C_j \circ \sum_i \text{DistanceMap}(C_i)) \tag{10.2}$$

whereas the normalization $n_j = 1/\sum_{x,y} C_j$ is used to address different contour lengths. The majority vote C_{mv} is given by its minimum cost $w_{mv} = \min_j(w_j)$.

In addition to this **annotation-based majority vote** where a complete annotation is extracted from all candidates, we discuss also the results of a **point-based**

	Minimum Number of Candidates	Maximal Jaccard Distance
MV_1	2	0.1
PWMV_1	2	0.1
MV_2	2	0.25
PWMV_2	2	0.25
MV_3	3	0.25
PWMV_3	3	0.25

Table 10.2: *We run our majority votes with different parameter sets. MV represents annotation-based majority votes and PWMV represents point-wise majority votes.*

extension. Our preliminary findings show that annotations of advanced users may have only little errors while the majority of annotated points are good. The annotation shown in Figure 10.3a on page 74 for example has explicit errors but the majority of all points is quite good. Hence, a majority vote determined by comparing entire contours can still have parts that are worse than those of other candidates. To address this issue, we extend annotation-based majority votes with a point-wise method. Therefore we iterate over all contour points $p_{mv,i}$ of the annotation-based majority vote C_{mv}. For each point $p_{mv,i}$ we collect a set P_i of the nearest neighbour points of all candidates. Then, the former point $p_{mv,i}$ is replaced by that point which minimizes the sum of distances of P_i. A weakness of our point-wise majority votes approach is given by the distance measure between points which we discussed in Section 9.4. Distances are calculated with regard to the former annotation-based result. If other results contain concave variations, reasonable points at this concave areas won't be considered as nearest neighbours.

We run the **majority vote** algorithms with several parameter sets and discuss those presented in table 10.2. A parameter set holds values for the **maximum allowed Jaccard distance** and values for the **minimum allowed number of candidate matches** to be considered for majority voting. The two parameters have an inverse effect. A higher number of required matches leads to more false negatives but prevents false positives. A taller Jaccard distance on the other hand accepts smaller intersections between candidates and potentially produces more matches. However if annotation precision is low and image content crowded, a high Jaccard distance can produce mismatches. For reasonable results we need to balance that threshold in accordance with expectable annotation precision, the density of objects and the diversity in annotation quality. The first parameter set requires at least two annotations to match within a Jaccard distance smaller than

0.1 to consider them as majority vote candidates. The according results are denoted as MV_1 for the annotation-based result and $PWMV_1$ for the point-wise majority vote. The second parameter set requires maximum Jaccard distances of 0.25. These results are given by MV_2 and $PWMV_2$ for the unfiltered MTurk crowd and for the gaming crowd. The third parameter set also requires a maximum distance of 0.25 and increases the minimum required number of matches to 3.

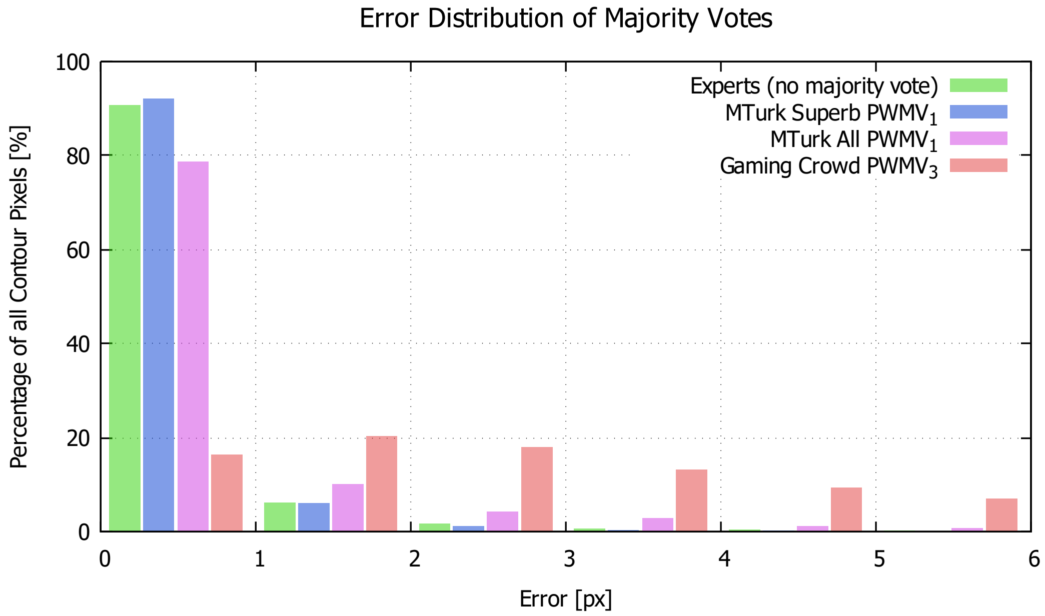
10.2.1 Evaluating Majority Votes

For a reasonable comparison of unfiltered and superb MTurkers we limit the maximum number of contributions for each majority vote to 5 which is reasonable from an economic point of view. Since the previous annotations from the gaming crowd could not satisfy high quality requirements, we do not choose a restriction to majority vote candidates for this crowd, meaning that up to 100 candidates are used for majority voting in this case. While the comparison between gaming crowd and MTurk seems inappropriate under these conditions, it can be justified from an economic point of view. The resulting financial costs were similar for both settings.
However, we will analyse majority votes for different numbers of candidates in 10.2.2.

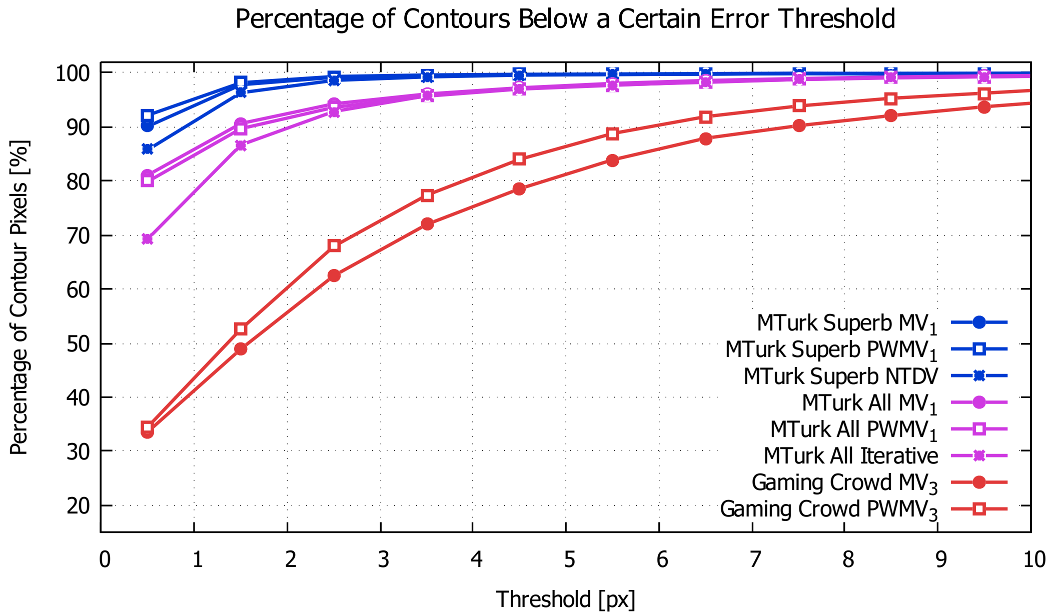
In regard to annotation precision, all presented majority votes perform better than single task results. The error distribution and the percentages of inliers in regard to the 2 pixel and 5 pixel threshold are given in Figure 10.7. Performance indicators for majority votes are listed in tables 10.3, 10.4 and 10.5 More details can be found in tables tables A1, A2 and A3 in the appendix.

Superb MTurkers

The annotation precision of **superb MTurkers** already turned out to be comparable to those of experts for single contributions. Applying a **majority vote** increases the percentage of acceptable contour points with regard to the 2 pixel tolerance from formerly 96.1% to 97.8% (MV_1) and 98.5% ($PWMV_1$). In other words, the former amount of 3.9% of annotated pixels lying outside the 2 pixel tolerance was reduced by 44% and 62%. This precision is better than that of unprocessed annotations of experts. Inliers in regard to the 5 pixel threshold were pushed from 99.5% to 99.7%. A comparison between unprocessed and processed results can be found in Figure 10.8a.
According to table 10.3, majority votes improved false negative rates significantly. The false negative rate of annotations taller than 17 pixels in width or height dropped from formerly 11.9% to 3.2%. None of unambiguous objects taller than 50 pixels were missed. Most of objects that were missed are vehicles at image borders and vehicles smaller than 17 pixels.

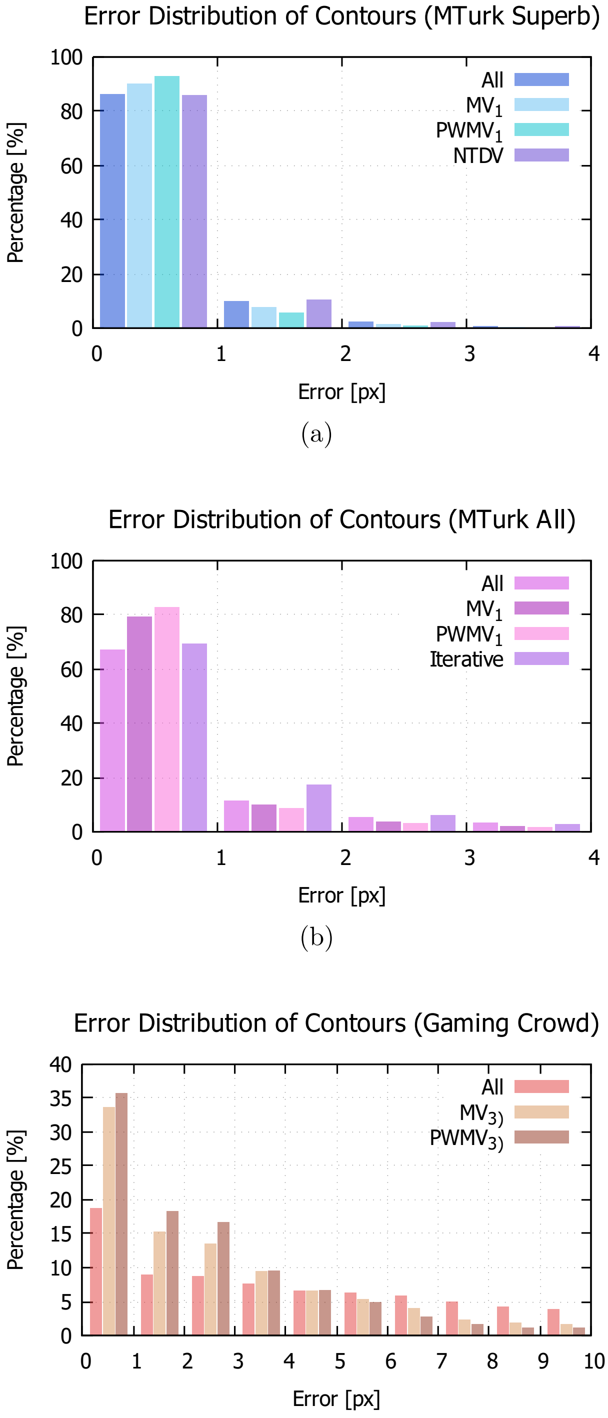


(a) All crowds benefit from majority votes. Processed results from superb MTurkers are even slightly better than those of experts.



(b) The difference between annotation-based and point-wise majority vote is little for Mturkers. Gaming crowd results do clearly profit from point-wise majority votes. Nothing-to-do-votes turn out to perform worse than majority votes in regard to annotation precision below a 2 pixel threshold.

Figure 10.7



(a)

(b)

(c)

Figure 10.8: *This figure visualizes the error distribution for different post process methods. Results clearly benefit from majority votes.*

	Experts	**MTurk Superb**			
		All	MV_1	$PWMV_1$	NTDV
Inliers ($\leq$2px)	96.8%	96.1%	97.8%	98.5%	96.3%
Inliers ($\leq$5px)	99.5%	99.5%	99.7%	99.7%	99.6%
FN_{all}	5.9%	33.9%	19.4%	19.4%	14.5%
$FN_{>50px,uncut}$	0%	0.3%	0%	0%	0%

Table 10.3: *Majority vote results for superb MTurkers. More details are given in table A1 in the appendix.*

	MTurk All							
	All	MV_1	$PWMV_1$	MV_2	$PWMV_2$	MV_3	$PWMV_3$	Iter.
Inliers ($\leq$2px)	78.5%	90.6%	91.6%	89.0%	89.7%	89.3%	89.6%	86.7%
Inliers ($\leq$5px)	89.6%	97.2%	97.6%	96.2%	96.3%	96.7%	97.1%	97.2%
FN_{all}	45.5%	25.8%	25.8%	24.2%	24.2%	38.7%	38.7%	12.9%
$FN_{>50px,uncut}$	9.5%	1.6%	1.6%	0%	0%	3.2%	3.2%	0%

Table 10.4: *Majority vote results for all MTurkers. More details are given in Table A2 in the appendix.*

Thus, applying majority votes on annotations of superb MTurkers leads to annotations whose accuracy exceeds that of experts. The false negative rate is still too high to make results be considered for ground truth data sets.

	Gaming Crowd						
	All	MV_1	$PWMV_1$	MV_2	$PWMV_2$	MV_3	$PWMV_3$
Inliers ($\leq$2px)	27.7%	38.2%	42.7%	49.6%	51.3%	49.3%	54.0%
Inliers ($\leq$5px)	50.6%	63.3%	68.9%	78.4%	81.5%	79.5%	86.9%
FN_{all}	25.8%	22.7%	22.7%	20.5%	20.5%	25.0%	25.0%
$FN_{>50px,uncut}$	1.2%	2.3%	2.3%	0%	0%	2.3%	2.3%

Table 10.5: *Majority vote results for gaming crowd annotations. More details are given in Table A3 in the appendix.*

Unfiltered MTurkers

Using majority votes for **unfiltered MTurkers** increases their annotation precision by over 10% in regard to the 2 pixel threshold and by about 7% in regard to the 5 pixel threshold. Especially heavy outliers can be removed. The 95% percentile dropped from 22 pixels to about 8 to 9 pixels. The error distributions can be found in Figure 10.8b, an excerpt of according values is given in Table 10.4. Looking at the results for different parameter sets, our algorithm performs best for $PWMV_1$. Using this parameter setting we can reduce outliers by 61% in regard to the 2 pixel threshold and by 44% in regard to the 5 pixel threshold.
The difference to other parameter settings lies between 1 and 3%. The reason might be the higher tolerance for Jaccard distances defined for MV_2 and MV_3, which is 0.25 in contrast to 0.1 for MV_1. However we measured an average Jaccard distance of 0.029 ± 0.037 for unprocessed annotations in regard to ground truth. Hence a maximum allowed Jaccard distance of 0.1 (MV_1) is much more reasonable as a value of 0.25 to address the majority but still avoid outlier candidates.
False positives (formerly 0.5%) could be avoided no matter which settings we chose. The false negative rates also drop significantly. The rate for unambiguous objects taller than 50 pixels changes from 9.5% to 1.6% (MV_1), 0% (MV_2) and 3.2% (MV_3). Best results for false negative rates were achieved by MV_2 which was expectable from the most tolerant parameter setting. The number of false negatives for small objects ($\leq$ 17px) increases. As we discussed in 10.1, the Jaccard distance used for matching is sensitive in regard to object sizes because of the relatively constant annotation errors. While this is an issue in our matching process in general, it was not the reason for the increase. The increase was caused since there weren't enough annotation candidates to build a majority vote.
Majority votes led to a significant increase of annotation accuracy. If one aims at acquiring contours whose turning points deviate at most 5 pixels from ground truth, the percentage of 97.6% of points achieving this accuracy might be sufficient for that purpose. However, false negative rates are still too high to use these results without further improvements.

Gaming Crowd

The best results for gaming crowd contributions were achieved for parameter set $PMWV_3$. The average Jaccard distance in regard to ground truth for unprocessed annotations is 0.087 ± 0.052. Thus, majority votes based on a maximum Jaccard distance of 0.25 achieved better results here (Figure 10.8c, table 10.5). Precision is pushed from 27.7% to 38.2% (MV_1), 49.6% (MV_2) and 49.3% (MV_3) in regard to the 2 pixel threshold. Point-wise majority votes have noticeable higher precision as the annotation-based approach. The point-wise version achieves 42.7% ($PWMV_1$), 51.3% ($PWMV_1$) and 54.0% ($PWMV_1$). A taller difference between point-wise

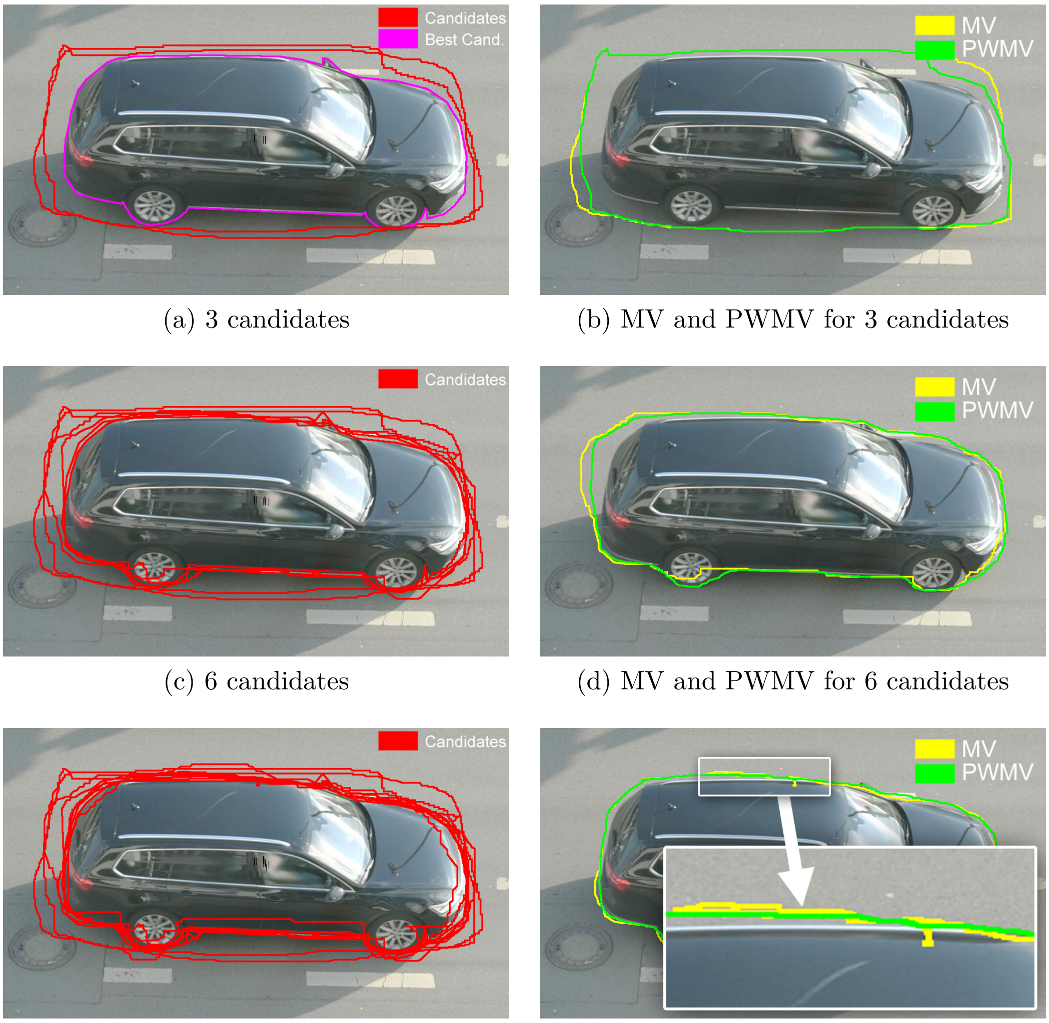
and annotation-based results arises in regard to 5 pixel thresholds. The point-wise approach $PWMV_3$ with 86.9% outperforms the annotation-based approach MV_3 with 79.5% by 7.4%. $PWMV_3$ reduces outliers in regard to a 2 pixel threshold by 36% and in regard to a 5 pixel threshold by 74%.
Overall false negative rates improve but the value for objects taller than 50 pixels increases for MV_1 and MV_3 caused by an insufficient number of candidates.
Although majority votes impressively improved annotation accuracy, these results still are not usable for high-quality ground truth datasets. At least 13.1% of contour points deviate more than 5 pixels from ground truth whereas 25% of all objects have not been annotated.

Using majority votes to increase annotation accuracy and to decrease false negative rates is an adequate approach for crowdsourced contour annotations. Before we conclude these results and compare them to an alternative iterative approach, we discuss annotation-based and point-wise majority votes by some characteristic examples.

10.2.2 Annotation-based vs. Point-wise Majority Votes

On an average, point-wise majority votes perform better than annotation-based votes for all crowds. At a first glance this seems to be expectable since the annotation-based approach determines one of multiple contour candidates. This contour is in the best case the most sensible choice in comparison the the other candidates. However, parts of this contour can still be worse in comparison to others. A point-wise majority vote can combine contour points from all candidates. In the best case, the annotation-based contour is improved at those parts which are worse in comparison to other candidates.

However, when analysing single results, several characteristics occur. Point-wise majority votes are not better in general. It depends on the precision and the accuracy whether a point-wise vote performs better. Furthermore and in accordance with Sheng et al. [75] who analysed majority votes for simpler user contributions, a majority vote does not maximize the quality when the quality of candidates differs too much. As can be seen in Figure 10.9, majority votes improve for an increasing number of candidates. The left column visualizes the candidates and on the right column the outcome of annotation-based and point-wise approaches is shown. The actually best annotation candidate (shown as purple annotation in 10.9a) is determined for neither of the candidate sets and approaches. In cases like these we would benefit from choosing only this particular annotation.



(a) 3 candidates

(b) MV and PWMV for 3 candidates

(c) 6 candidates

(d) MV and PWMV for 6 candidates

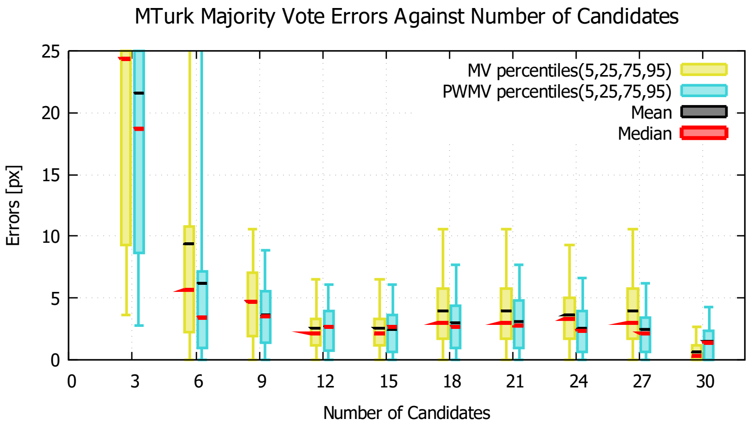
(e) 9 candidates

(f) MV and PWMV for 9 candidates

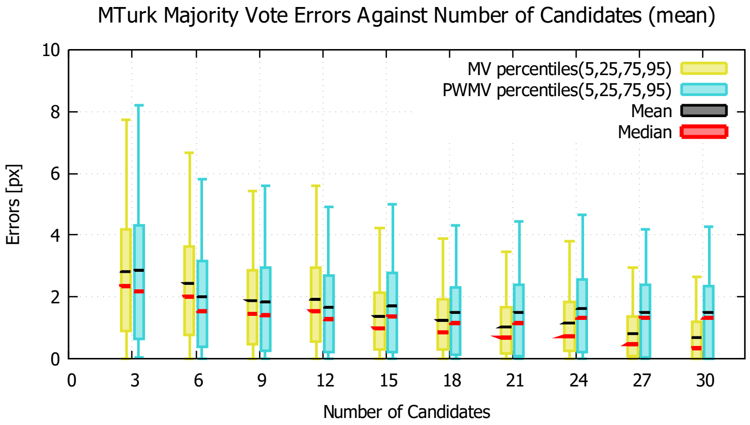
Figure 10.9: *Comparison of annotation-based majority votes (MV) and point-wise majority votes (PWMV). Annotations were taken from MTurk results. The candidates used for majority voting are visualized in red. Increasing the number of candidates improves the majority votes. The actually best annotation (which is visible in 10.9a) however cannot be detected for neither of the candidate sets.*

Majority Votes of Annotations with Strongly Varying Quality

Apart from the fact that majority votes do not guarantee to maximize the quality, there are cases where annotation-based votes perform better than point-wise votes. The majority votes shown in Figure 10.9 are based on a candidate set with overall 30 annotationso of unfiltered MTurkers. To calculate these majority votes, we choose a maximum allowed Jaccard distance of 0.25 to collect these candidates. As



(a) This figure shows the quality of annotation-based and point-wise majority votes in regard to the number of candidates used for calculation. In this example we use annotations of a single object for which we collected over 30 annotations. In contrast to figure 10.10b we chronologically add candidates for majority vote calculation.



(b) In contrast to 10.10a, the candidates of this figure are not added chronologically. For each number of candidates we determine the average errors of several random subsamples from the whole candidate set. Suppose the quality of all candidates was typical for this crowd, the distribution above could motivate to rely on annotation-based majority votes.

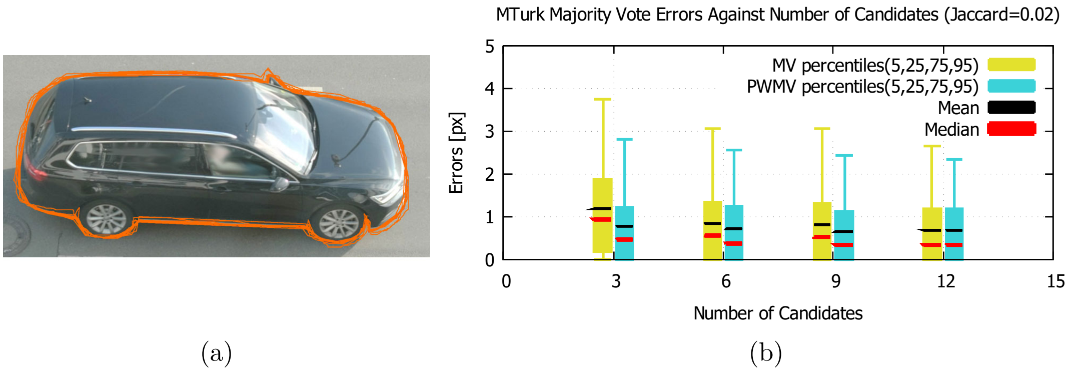
Figure 10.10

stated in 10.2.1, this value is relatively high and inappropriate with regard to the overall quality of MTurkers. Though, choosing this value enables us to analyse the error progression for a large candidate set.
While 10.9 shows the outcome for the first three candidate subsets, the complete error progression of up to 30 *chronologically* added candidates is visualized in Figure 10.10. While for most candidate sets the point-wise result is better than the annotation-based result, this is not the case for the sets with 12, 15 and 30 candi-

dates. These results depend on the chronological order of added candidates. For a more general error progression without chronological dependency, we also evaluate the candidate set by determining average errors for multiple random subsamples of different subset sizes. The results are given in Figure 10.10b. For this specific set of candidates, an annotation-based majority vote benefits more from additional annotation candidates as point-wise votes do. For candidate subset with more than 12 annotations, annotation-based results are better than point-wise results.

Majority Votes of Precise Annotations

The previous results are based on an imprecise set of candidates with a maximum allowed Jaccard distance of 0.25. Furthermore the accuracy of single annotations strongly differs. To compare these results to a more favourable settings with regard to point-wise majority votes, we determine another candidate set with a maximum Jaccard distance of 0.02 for the same object. By restricting candidates by such a low Jaccard distance automatically increases the precision and thus assures similar accuracies for all candidates. Using this setting we determined 11 matching annotations overall. Figure 10.11 shows the error progression for increasing numbers of candidates. Obviously point-wise majority votes perform better for all sizes of candidate sets. The resulting 95-percentile is significantly lower for point-wise results. The quality of annotation-based majority votes improves when increasing the number of candidates from 3 to 6 (or higher). The quality of point-wise results only changes slightly. Relying on only 3 candidates for a point-wise majority vote can be a reasonable decision with regard to economical reasons.



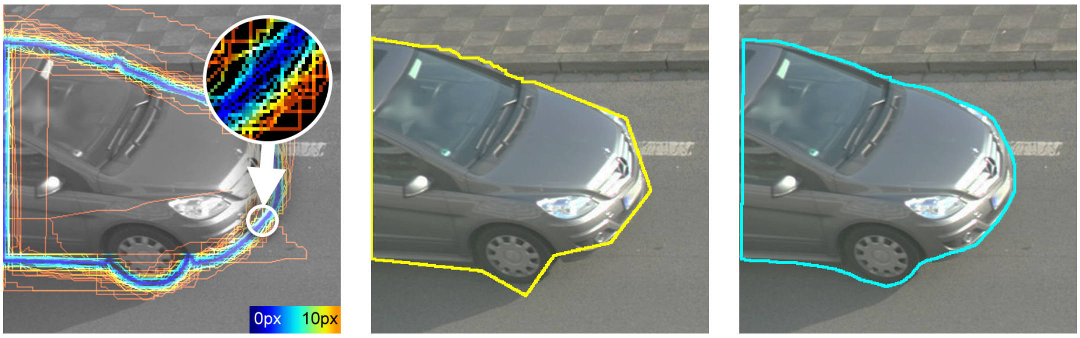
(a) (b)

Figure 10.11: *Setting a low Jaccard distance to find majority vote candidates implicates similar quality of these annotations. Under these conditions, point-wise majority votes perform better an require less candidates for adequate results.*

Majority Votes of Imprecise Annotations with Comparable Accuracy

Point-wise majority votes are useful especially when accuracies of candidates are comparable. Appropriate samples are given by gaming crowd annotations. These annotations have low precision but the accuracy is comparable for most of them. Figure 10.12 shows an example. We used again a tall Jaccard distance of 0.25. In contrast to the previous MTurk example, this is a good choice since the majority of resulting candidates have a comparable accuracy.

In 10.12a, the deviation from ground truth is visualized with a color map. It has to be noted that the blue contour represents the superposition of multiple contours and does not show a single result. The annotation-based majority vote given in 10.12b is a good estimate with regard to all other candidates. Though it contains too less turning points for an adequate representation of the subject, the deviation from ground truth is low in comparison to other candidates. The point-wise result in 10.12c however clearly improves the annotation-based result.



(a) Annotations of laymen such as the presented gaming crowd mostly are rough. However the majority oscillates about the actual shape.

(b) The annotation-based majority vote is a good pick in comparison to most other annotations.

(c) Point-wise majority votes deliver smoother and more adequate contours.

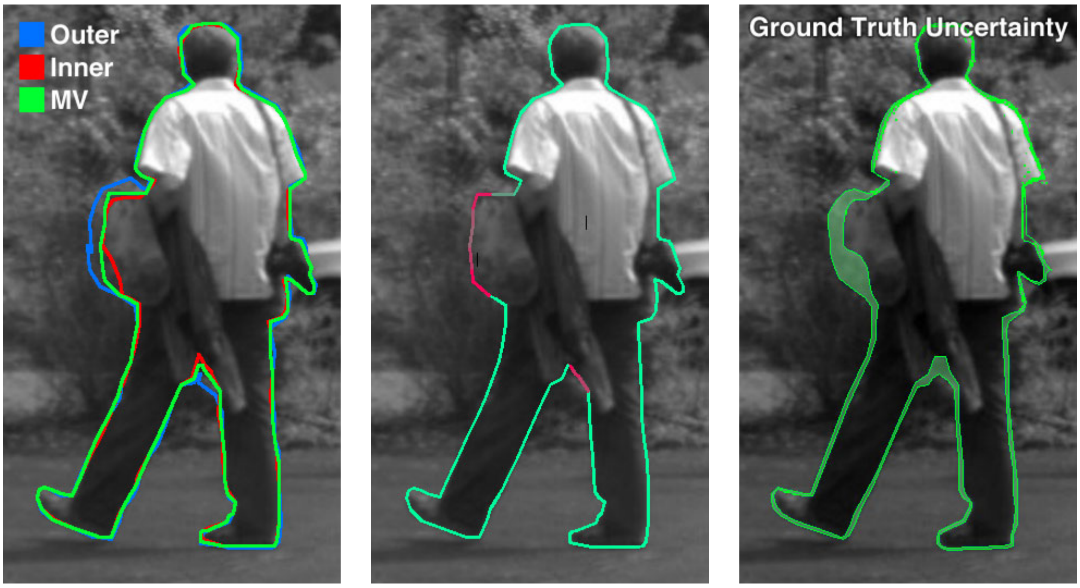
Figure 10.12: *Majority votes for gaming crowd annotations.*

10.2.3 Detecting Ambiguous Edges by Annotation Uncertainties

Since we collect several annotations per object, we can evaluate variances of point-wise distances. This enables different options. With regard to the evaluation above, we can utilize point-wise variances to determine whether collecting further annotations for specific objects can improve majority votes.

Secondly, we can utilize the candidate sets to define an **inner and outer annotation** which indicates the annotation precision. These annotations are comparable to the inner and outer ground truth annotations which we use to address ambiguous edges.

A third option to utilize point-wise variances is to **detect ambiguous edges**. For this case, the average deviation of nearest neighbour points has to be smaller than the size of putative ambiguities. Then, point-wise significantly increased variances can indicate edge ambiguities.



(a) Inner, outer and majority vote annotations of superb MTurkers.

(b) Annotation with colourized point-wise variances.

(c) Inner and outer ground truth annotation.

Figure 10.13

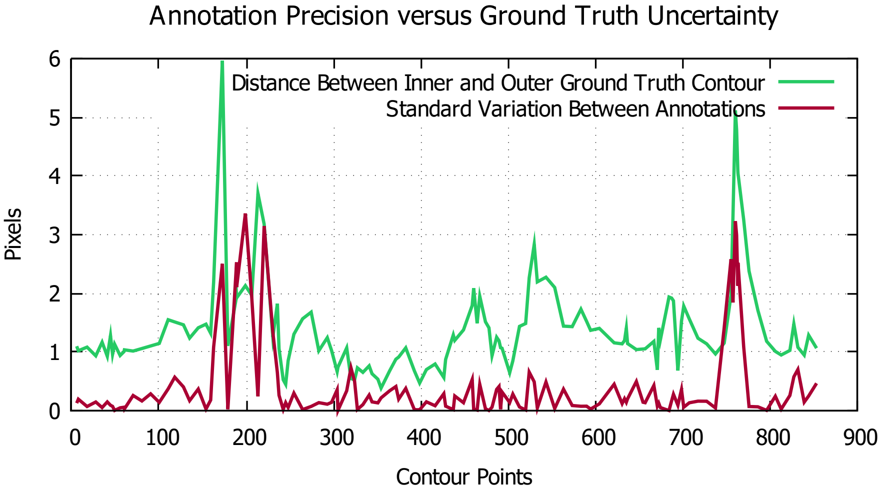


Figure 10.14

To emphasize these options, we use a different image sequence that contains significant ambiguities. In particular these are grey-scale images of pedestrians within an automotive scenario. We ask crowd users to annotate these pedestrians. Due to the colourless image type, these subjects often are hardly distinguishable from background. At several contour parts the boundaries between pedestrians and background can only be estimated.
Figure 10.14 shows an example for hardly distinguishable object boundaries annotated by superb MTurkers. Beside the majority vote, Figure 10.13a visualizes the inner and outer annotation which represent the annotation precision. To determine these annotations, we calculate point-wise distances between the majority vote and neighbouring annotations. Due to our calculation of point-wise majority votes we already know these distances. Then, to create an inner and outer annotation we choose that points, which correspond to the third quartile of distances with regard to the inside and outside of the majority vote annotation.
While annotations only deviate slightly for most contour parts, participators could not reach a consensus at ambiguous regions. Obviously, the confidence of annotations can strongly differ for various parts of an object. As can be seen by comparing Figure 10.13a with 10.13c, the point-wise variance of annotations correlates with the ground truth uncertainty. This behaviour is emphasized by Figure 10.14 which plots point-wise variances and ground truth uncertainties.

Using point-wise variances to detect edge ambiguities only makes sense when the average variance is small in comparison to the size of ambiguities. The results indicated by Figure 10.14 and 10.13 can only be achieved for precise annotations of superb MTurkers. Annotations of unfiltered MTurkers for instance, could not be used to detect ambiguities of that size. However, inner and outer annotations can be determined for every annotation set to estimate its precision.

10.2.4 Summary

Majority votes are an adequate approach to improve annotated contours, though it cannot guarantee to achieve the maximum quality. Both presented variants mean a significant improvement over single annotated contours. The biggest increase of accuracy was achieved for gaming crowd results were outliers with regard to a 5 pixel accuracy could be reduced by 74%.
To receive good results, the two parameters (for the minimum number of candidates and the maximum allowed Jaccard distance for an annotation to be considered as candidate) have to be chosen carefully. As we have seen in the previous examples, choosing an inappropriate tall maximum Jaccard distance can result in candidates with strongly varying quality. In this case, annotation-based majority votes can perform better than point-wise majority votes. However, for rough annotations such as the gaming crowd contributions, taller maximum Jaccard distances are needed to receive adequate candidates for majority voting. Since in this case, the

accuracy of most annotations is comparable, combining several annotations via point-wise majority votes is favourable.
Another benefit of majority votes is the possibility to estimate the annotation precision as discussed in 10.2.3. When the average precision is high, as for most of the annotations of superb MTurkers, point-wise deviations from this average precision can indicate edge ambiguities. This knowledge about uncertainties of edges can be an important information for performance analysis.

One of the major disadvantages arising for majority-vote-based annotation protocols is the required working time. For every annotation candidate we need one additional contribution. As a consequence, the working time increases with a factor that is equal to the number of desired annotation candidates. While the increase in working time could be addressed with big crowds, the increasing costs can be a reason to avoid majority voids. If working time and costs play no role, majority votes can be highly recommended.

Future Improvements

Further improvements can be made in the case of strongly-varying quality of annotation candidates. To address this issue, we could take the maximum allowed Jaccard distance as a variable value which decreases when enough candidates satisfy a smaller value.
Another approach could be to weight annotations by the annotator's reliability. Therefore we could evaluate all previous annotations of the according annotators and assign trust levels to these annotators. In case that majority vote candidates differ too much (Figure 10.9a), we could choose the most reliable annotation with regard to these trust levels.
In addition to this, a majority vote algorithm could also involve image information. With regard to object boundaries, image gradients could be indicators for a reasonable annotation. Annotation candidates that correspond to edges calculated from image data could be preferred.

10.3 Iterative Annotations

In addition to majority votes we analyse the performance of an iterative annotation protocol according to the principle *"Two heads are better than one"*. Therefore we return previous contributions to other participators and let them improve these initial annotations. Missing annotations can still be added in these tasks as well as the deletion of false positives. We restrict the number of such iterations to 5. Hence the number of iterative contributions roughly corresponds to the number of contributions which we used for the majority vote approach. Additional to the

	MTurk Superb			**MTurk All**		
	Single	PWMV$_1$	NTDV	Single	PWMV$_1$	NTDV
Inliers ($\leq$2px)	96.1%	98.5%	96.3%	78.5%	91.6%	86.7%
Inliers ($\leq$5px)	99.5%	99.7%	99.6%	89.6%	97.6%	97.2%
FN$_{all}$	33.9%	19.4%	14.5%	45.5%	25.8%	12.4%

Table 10.6: *Results of this iterative approach (NTDV) in comparison to previous results of single annotations and majority votes.*

fixed maximum number of iterations we assume a contribution as final, when two subsequent participators reached a consensus. We call each contribution which does not change its predecessor a *nothing-to-do-vote* (NTDV). We run this NTDV approach only with the MTurk crowds, since according to former results we cannot assume our gaming crowd to manage these extended methods well.

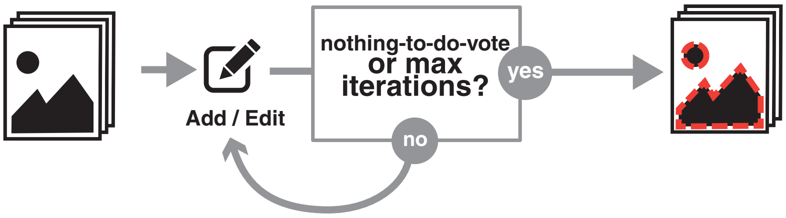


Figure 10.15: *We evaluate an iterative approach which we also named nothing-to-do-vote approach. Here the iteration of adjustment tasks has a break condition which takes effect once several contributors reached a consensus.*

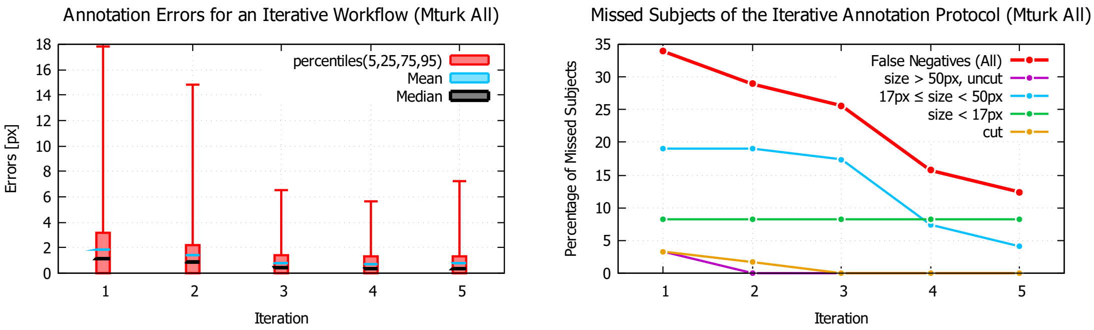
In Table 10.6, results of this approach are compared with previous majority votes and single annotations. More values are listed in tables A1 and A2 in the appendix. The previous Figure 10.8 shows the according error distribution in comparison to single added contours and majority votes.

Superb MTurkers

The nothing-to-do-vote workflow run with superb MTurkers achieves a slightly better annotation precision in comparison to normal add tasks. Majority votes still perform significantly better in regard to the 2 pixel tolerance which is visualized by the previous error distribution given in Figure 10.8a. Nothing-to-do-votes on the other hand perform better on false negative rates (FN$_{cut,NTDV}$ = 0, FN$_{cut,MV}$ = 4.5%).

Unfiltered MTurkers

The iterative approach run with unfiltered MTurkers shows improvements over annotations created by single annotators. Both, the annotation accuracy and false negative rates improve. In regard to the 5 pixel tolerance, annotation accuracy is equal to majority vote results (see Figure 10.8b). Though, majority votes perform still better when we require a 2 pixel accuracy.
A huge benefit from the iterative approach is the reduction of false negatives. While majority votes reduce the formerly false negative rate of 45.5% to 25.8%, the iterative approach achieves a value of 12.4%. The iterative reduction of false negatives is visualized in Figure 10.16b.



(a) Annotation precision for subsequent iterations of add/edit cycles (Mturk, all).

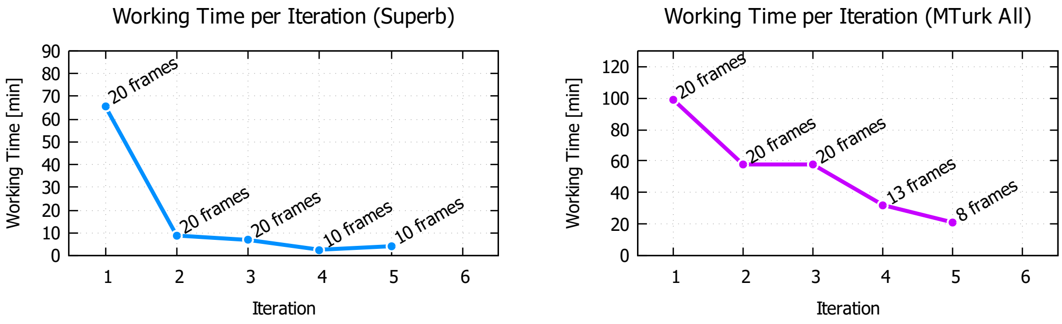
(b) False negative rates for subsequent iterations of add/edit cycles (Mturk, all).

Figure 10.16

In contrast to the superb subgroup of MTurkers, the unfiltered user group clearly benefit with regard to the accuracy. The reason might be their handling of our annotation tools. As shown in 9.6, a sensible contour creation consists of a drawing step and a correction step. The correction step is optional. Its intent is to correct bad turning points that might have arisen during the drawing step. While good annotators can achieve nearly errorless annotations only by drawing, a weak annotator cannot. Especially for weak annotators, the correction step provides the opportunity to clearly improve the previous contour. However, annotators tend to skip this correction step during contour creation. However, in progressed steps of an iterative workflow, most of annotations already exist which let workers focus on correcting them.

Working Time

Beside good detection rates of iterative tasks the major difference to majority votes is that tasks can benefit from former results. This is the case if annotations given



(a)

(b)

Figure 10.17: *This figure shows the overall working times for each iteration. The number of frames processed in each iteration is given above each data point. We require two nothing-to-do-votes to define a frame as final. These votes can be submitted in iteration 2 and 3 at the earliest. Thus, only iteration 4 and 5 have less frames that need to be processed.*

from former results are adequate for time-saving adjustments or even good enough to be accepted as they are. Hence the incremental character of iterative workflows can mean significant time savings compared to majority votes. In comparison to a single annotation task, we obtain an average working time increase of factor 2.7 for unfiltered MTurkers and of factor 1.3 for superb MTurkers. In contrast, the overall working time for majority vote approaches depends on the number of used single annotations. In case of 5 majority vote candidates, the working time also increases by factor 5.

10.3.1 Summary

Iterative workflows can significantly reduce false negative rates and require less time as majority votes. However the process of editing an existing annotation is advanced in contrast to adding new annotations. It would have been inappropriate to run the iterative approach with gaming crowd participators. User interface optimizations and new annotation tools could address this issue in future.
A further disadvantage of iterative approaches is the destructibility of existing annotations. Existing results of advanced users could get worsened by subsequent annotators. To address this issue, we could use user trust levels similar to the suggested approach for strongly varying quality of majority votes candidates (10.2.4). By doing so, an unreliable annotator may not significantly change annotations of trustworthy annotators.

Future Improvements

If crowds are capable to successfully participate on iterative tasks, a combination of iterative submissions and majority votes could make the best of both approaches. An additional task design to recognize false negatives also seems to be reasonable. It could be sufficient to let all relevant objects be roughly marked in a preceding task. Afterwards an annotation task could be restricted to the marked areas and require the formerly determined number of objects to be annotated.

10.4 Propagation Methods

Section 10.3 has shown that the iterative annotation protocol can increase quality and decrease false negative rates. Since this method achieved more precise results as single contributions and processing times are little once an initial annotation is given, it seems reasonable to utilize final annotations as initial guesses for subsequent frames. Thus we analyse how our participators perform on editing annotations that are propagated from previous frames. We aim at shorter processing times compared to annotations from scratch while expecting a quality similar to the presented iterative approach. We have already analysed propagation methods for temporal dense image contents in part one. There we used a learning-based approach. For reasons of simplification and to ensure quick processing times, we rely on simple chamfer matchings [8] in this approach. Nevertheless, a more advanced algorithm like that in Part I could bring further improvements since chamfer matching simply matches edges by varying positions and scales.

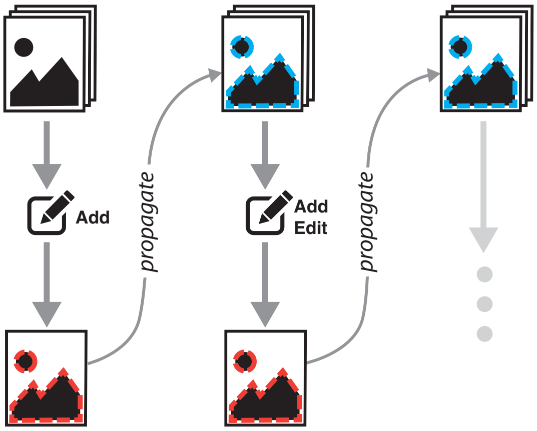
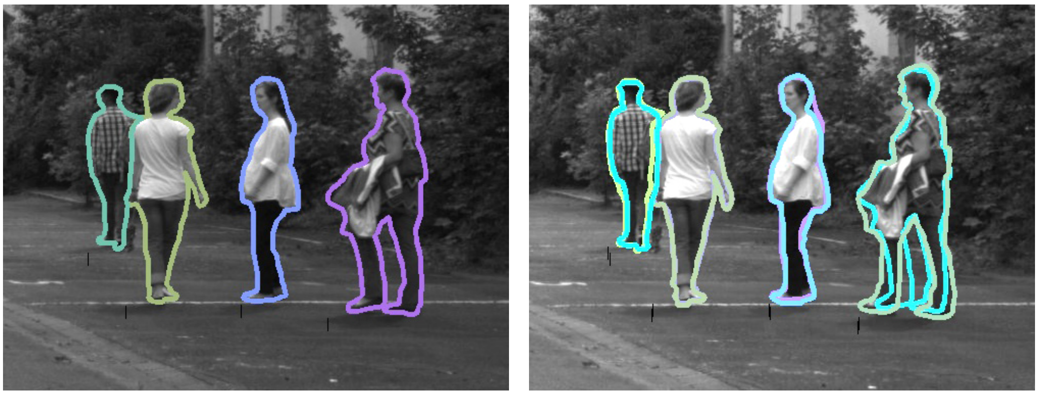


Figure 10.18: *Propagations of annotations are used as initial guess for subsequent frames.*

The usage of propagation methods is applicable for temporally dense image data. Reference data sets however do not necessarily aim at providing dense data. For learning purposes it might be more appropriate to ensure a wide variance of image contents. Even when the raw image data is temporal dense, economical reasons may lead to the decision to process data with a tall temporal step width. In these cases the utilized propagation method is be applicable. Learning based approaches still might be able to provide reasonable initial guesses to make use of the general idea of propagations.



(a) Pedestrian contours

(b) Overlapping of actual shape and propagations from previous frame.

Figure 10.19: *Figure 10.19b shows the overlapping of propagations and actual shapes. Obviously a primitive matching method such as the used chamfer matching cannot address human motions and requires annotators to adjust nearly every annotation.*

10.4.1 Propagation of Contour Annotations

We analyse this approach for experts, superb MTurkers and for the unfiltered MTurk crowd. We are asking for pedestrian contours in grey-scale images. In contrast to our vehicle data set, pedestrian contours have several concave areas. The measure of absolute distances may be erroneous at these parts (see Section 9.4). In contrast, Jaccard distances seem to be a better performance indicator for this test data. All objects annotated in this analysis have a similar size and thus the sensibility of Jaccard distances with regard to absolute object sizes is mostly negligible. Thus, Jaccard distances are a viable performance indicator in this analysis.

Experts

The annotation quality and processing time of experts for adjusting propagated annotations turn out to be very similar in comparison to a normal add task as can be

	Experts		
	Add	Edited Prop.	Automatic Prop.
Jaccard [10^{-2}]	3.5 ± 1.8	3.3 ± 1.1	13.8 ± 3.7
Inliers (≤2px)	94.7%	94.1%	57.0%
Inliers (≤5px)	98.8%	99.2%	91.9%
Time$_{mean}$ [s]	241	196	-
Time$_{3-quartile}$ [s]	228	202	-
Time$_{max}$ [s]	806	229	-

Table 10.7: *Comparison between added annotations and edited annotations based on propagations. More values can be found in table A4 in the appendix.*

	Superb MTurkers			All MTurkers		
	Add	Edited Prop.	Automatic Prop.	Add	Add MV	Edited Prop.
Jaccard [10^{-2}]	3.5 ± 2.0	3.6 ± 3.5	12.1 ± 0.5	10.9 ± 7.4	6.0 ± 4.9	6.9 ± 4.1
Inliers (≤2px)	94.5%	93.4%	62.5%	71.1%	87.5%	84.0%
Inliers (≤5px)	98.2%	98.7%	93.5%	91.0%	97.5%	97.0%
Time$_{mean}$ [s]	112	306	-	155	-	181
Time$_{3-quartile}$ [s]	107	308	-	208	-	238
Time$_{max}$ [s]	686	596	-	503	-	662

Table 10.8: *Comparison between added annotations and edited annotations based on propagations. More values can be found in table A4 in the appendix.*

seen in table 10.7. Our simple propagation method is limited to the adjustments of position and scale of an annotation. Hence the propagation cannot address moving body parts. As a result, nearly every turning point of the propagated initial guess needs to be corrected, which is a time-consuming task. The propagation quality is indicated by values in column *Automatic Propagation* in table 10.7. By the use of such rough propagations, this approach does not lead to desired improvements for the expert user group.

Superb MTurkers

Superb Annotators also do not profit from the presented propagation approach. While the annotation quality is comparable to normal add tasks, the processing time increases by about factor three. If we neglect the advantage of lower false negative rates in iterative tasks as measured in 10.2, this approach is not appropriate to be run with superb MTurkers. An interesting fact is the processing time of normal add tasks. participants needed half the time of experts to produce comparable results. On the other side, times for adjustment tasks were fifty percent taller than those of experts. Hence, superb MTurkers are significantly more efficient by using the add tool than using edit methods. This is another indication for an inefficient edit mode of our user interface and should be addressed in future work.

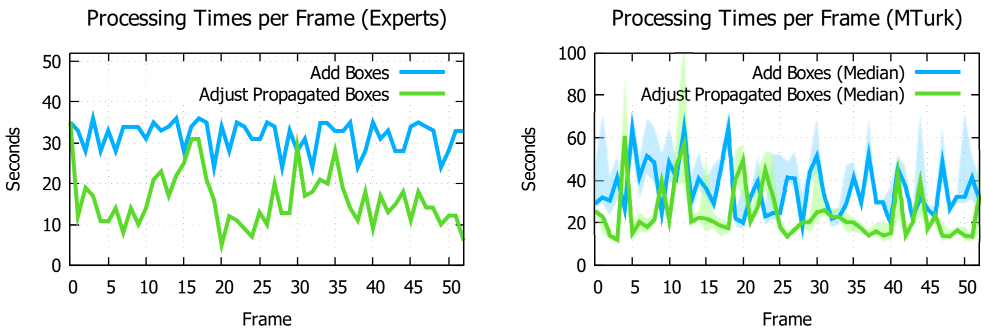
Unfiltered MTurkers

The unfiltered MTurk crowd clearly benefit from propagation adjustment tasks. The annotation precision is better and annotation times are similar for both, add and edit tasks. The propagation was based on majority votes of normally added annotations. In comparison to normal add tasks, these majority votes improved results comparable to our observations of Section 10.2. The resulting quality of adjustment tasks (Jaccard = 6.9 ± 4.1) based on propagated majority votes is more comparable to the initial majority vote (J = 6.0 ± 4.9) than to single added annotations (J = 10.9 ± 7.4).
This difference might be cause be the same reasons that make unfiltered MTurkers perform more efficient on iterative tasks, due to their tool handling. We already discussed this issue in Section ref{sec:itermturkall. A normal add task provides several choices to create a contour which also enables users to use inappropriate techniques. Users are not required to correct misplaced contour points, once a contour is created. This is a common issue for unfiltered MTurkers which mostly submit directly after the creation step. However it is hard to create an errorless contour without further corrections. In the case of propagation adjustment tasks, participants are asked to especially focus on correcting misplaced turning points without bothering about the actual creation step.Thus, the superior results of unfiltered MTurkers on propagation tasks are caused by requiring them to work more carefully and not by a more general advantage of the presented propagation method.

Summary

With regard to annotation quality, unfiltered MTurkers are the only user group which benefits from propagation tasks. The processing time could be improved for neither of the test groups. The main reason is that propagations are rough and nearly every contour turning point needs to be adjusted if done correctly. Adjusting a contour with our user interface however requires at least a similar effort as creating a new contour. Hence, to address this issue we need to either optimize the adjustment tools or improve the propagation quality. The less adjustments need to be done to correct a propagation, the more this workflow might be beneficial in regard to processing times. On the other side, this means that annotation types that can be propagated more confidently and whose adjustments mean less effort should profit all the more from this approach. During our work with reference data sets we also intensely collected bounding box annotations. A bounding box seems to be a great minimal viable product to test propagation workflows quite independently from propagation quality and user interface dependencies. Template matching produce good matches on a temporal dense scene and correcting a box only requires to adjust four points at most. For this reason, we analyse the same annotation workflow for box annotations in the next section.

10.4.2 Propagation of Bounding Box Annotations



(a) (b)

Figure 10.20: *Experts clearly benefit from initial guesses. The required time per box (*2.6*s) is half of that needed to add boxes from scratch (*5.2*s). Mturkers profit as well from propagation tasks. The average annotation time per box drops from* 15.9 *seconds to* 9.0 *seconds.*

We analyse propagation workflows for bounding box annotations of vehicles. The test data is taken from a similar ground truth data set as was used for the previous pedestrian contours. Instructions on this test were to annotate all vehicles which are not occluded or cut at image borders. We evaluated results of experts and unfiltered MTurkers.

Both user groups benefit from propagation tasks. The average annotation time per box drops from 5.2 seconds to 2.6 seconds for experts and from 15.9 seconds to 9.0 seconds for MTurkers. Figure 10.20 shows the according annotation times per frame. The peaks of 10.20a appear at frames where only little or none (frame 0) of the bounding boxes could be propagated. This shows, that the creation process of boxes requires more time as correcting them.

The results of MTurkers shown in figure 10.21 indicate, that task instructions again have been imprecise. Some of the participants assumed side mirrors to be included into annotations while others excluded them. Especially when aiming at post process methods such as majority votes, ambiguities like these need to be avoided to achieve optimal results. Assuming both solutions as valid, annotation quality is comparable to that of normal add tasks.

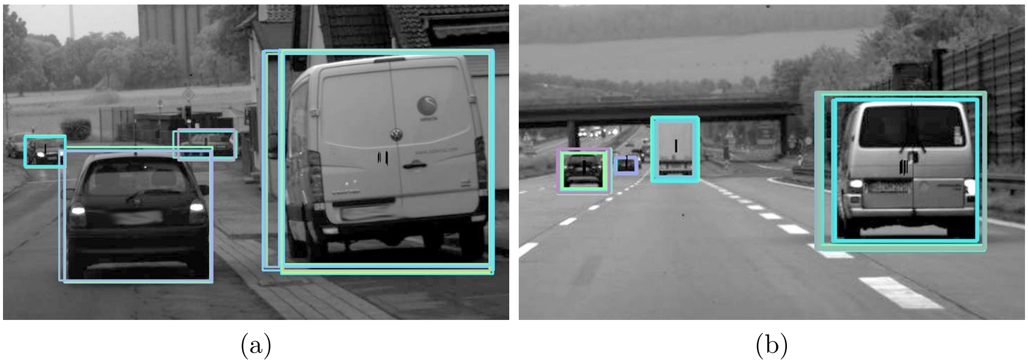


Figure 10.21: *We tested propagation workflows on bounding box projects to test this approach on a more favourable setting. Our instructions again turned out to be vague. Some of the participants included side mirros into their annotations, some did not. If one relies on algorithms such as majority votes, such ambiguities have to be avoided to achieve optimal results.*

Summary

The results for bounding boxes prove our assumption that the applicability of propagations is related to propagation quality and simplicity of adjustment tools. In the case of boxes which only need to be slightly adjusted, our propagation approach reduced the working time by 50% for experts and by 43% for MTurkers. As a consequence, relying on propagations to speed up annotation acquisition is a viable approach as long as adequate propagations and adjustment tools can be provided. Using advanced propagation methods for contours also could lead to shorter processing times. In contrast to bounding boxes, contour annotations require a time-consuming creation step. A precise contour propagation which does not need a lot of adjustment could increase efficiency far more than propagations do for bounding box projects. Advanced contour propagations that address dynamic objects would be a major improvement for this workflow. Using approaches such as the learning-based algorithm presented in Part I might be a considerable upgrade.
Since several results indicated that our contour adjustment tool is inefficient in comparison to the creation tool, the optimization of these tools is another obvious option to achieve better and more efficient results.

Efficiency of Crowds				
	Time per Annotation [seconds]	Annotations per Hour per User	Annotations per Hour (Majority V.)*	Annotations per Hour (Iterative)
Experts	157	22.9	-	-
Mturk Superb	163	22.1	7.4*	17
Mturk All	206	17.5	3.5	6.5
Gaming Crowd	33	109.1	10.9**	-

Table 10.9: *This table lists annotation times for different annotation protocols.* **) We assume 3 candidates for majority votes of superb MTurkers, 5 candidates for unfiltered MTurkers and 10 candidates for majority votes of the gaming crowd.*

10.5 Processing Times

The previous Sections primarily discussed crowdsourced annotation approaches in a qualitative manner. To analyse the applicability of crowdsourcing for high-quality annotations, we also need to consider according processing times. In this Section we roughly discuss throughputs that can be established with the presented approaches. However, it needs to be mentioned that the following measures are only estimations to get an idea about the performance of crowds. Quality and performance strongly depend on the number and competence of participators, their motivation and perception. These conditions can strongly vary depending on day time or even the season of year.

We aim at roughly estimating the temporal value of crowdsourcing large high-quality datasets. Therefore we compare processing times for contour annotations of vehicles. The values given in Table 10.9 represent the average processing time determined by the overall processing time for a complete dataset and by the number of annotations acquired.

Experts

According to Table 10.9, our experts took between two and three minutes to annotate a vehicle contour. As an estimation for the maximum capability of an expert we assume 7 hours of work per day. We neglect breaks, loading times between tasks and fluctuating concentration during 7 hours of annotating. We assume that experts process a frame only once, although our previous results show that for instance the iterative approach could also achieve better results for experts. According to these favourable assumptions, an expert could achieve about 160 high-quality contour annotations per day. This amount scales with the number of experts.

Superb MTurkers

The processing time per annotation of superb MTurkers is comparable to that of experts. However, although the quality of single annotations was quite good, experts still performed better. The major difference between contributions from experts and superb MTurkers is the false negative rate. We expect a large amount of false negatives to be caused by vague instructions. Though, it would be inadequate to rely on single contributions of MTurkers. The presented majority votes and iterative tasks are valid considerations to improve results.
With regard to majority votes, the final annotation time rise by a factor equally to the number of used contributions. While we used five contributions per majority vote for our analysis, Section 10.2.2 indicates that three precise majority vote candidates can already be enough to achieve a proper quality that can compete or even exceed those of experts. In this case, annotation times would be three times taller than those of experts.
While the annotation quality of superb MTurkers might already be good enough to skip time-consuming majority votes, at least iterative annotations should be considered to address erroneous contributions. According to our measures in Section 10.3, iterative annotations increase the overall processing time by about 30%.

Superb MTurkers can compete with experts with regard to annotation quality but they require longer annotation times if one considers reliable results. For the latter case we estimate a final processing time per annotation of three and a half minutes for iterative annotations and eight minutes when relying on majority votes of three annotation candidates.

The number of superb MTurk participators is little. We initially found about 60 persons who contributed to previous projects and whose results satisfied our superb quality requirements. However, once we started our test project for superb MTurkers, only 10 of these MTurkers participated in our tests. In our experience, superb MTurkers can achieve adequate results but it is difficult to find enough workers to run large projects exclusively on Amazon Mechanical Turk. Superb MTurkers can be beneficial if experts are unavailable or if manpower shall be extended.

Unfiltered MTurkers

The working time to annotate a vehicle required by unfiltered MTurkers was about three and a half minutes. We collected about one thousand annotations within 24 hours for testing purposes. Suppose we rely on majority votes with five candidates per vote, we could achieve 200 annotations per day. However we have to consider that these results do not satisfy our high-quality requirements. 91.6% of majority vote annotations were more precise than 2 pixels but the overall false negative rate was still 25.8%. Under these conditions and without further ado, relying on unfiltered MTurkers is not appropriate for a high-quality project.

Gaming Crowd

Users of the gaming crowd only spent half a minute per annotation which isn't surprising with regard to the annotation quality. While we used 100 candidates per majority vote in our previous gaming crowd results, a comparable result quality could be achieved by only 10 to 15 candidates. This would result in an annotation time of about 7 minutes. During our tests we collected about 2000 annotations within 24 hours. The use of 10 candidates per majority vote would result in 200 annotations per day which is pretty low for the according quality. The resulting annotations do not satisfy our high-quality requirements.
However, if these results were desirable (e.g. for learning-based applications), we could achieve a multiple of the amount above by using additional crowds. Since our participators are laymen, every crowd on the internet may achieve comparable results. The gaming provider which we used for these tests claims to have over 100 million registered users. While no details are given about the rate of active users or multiple registrations per user, we assume the lower bound at 10,000 - 50,000 active users at the very least. Since this is only one of many content providers that rely on the according monetarization model, we could benefit from many more users of other crowds. Due to this fact, research on further approaches to rely on laymen-crowds for labour-intensive computer vision applications is considerable.

10.5.1 Summary

While many annotations can be acquired from crowds such as the used gaming crowd, their results do not satisfy our quality requirements for now. However, low annotation times and plenty of comparable crowds encourages to run further attempts on that field.

Superb MTurkers can achieve adequate efficiency, but the little number of workers limits the scalability of Amazon Mechanical Turk. Additionally to Amazon Mechanical Turk we also worked with other commercial crowds. Results of these crowds were qualitatively comparable to the results of superb MTurkers. The presented methods to improve results are applicable to these crowds as well. By using an iterative workflow, there were no issues with missing annotations. Workers required 233 seconds on average to annotate various contours including vehicles and pedestrians. The typical size of such crowds reach from about thirty to a few hundred workers. Since these workers annotate about 7 hours per day as well, the according crowds are a serious and efficient alternative. One hundred of these workers could achieve 10,790 annotations per day. For the same amount of annotations, we would need 67 experts. As in our country of residence a student assistant who could act as expert costs around $20 per hour (including overhead), crowdsourced acquisition projects can also be reasonable from an economical point of view, since usually crowd workers costs less than half of that.

11 | Future Work

We have analysed different approaches to acquire contour annotations in Section 10. In comparison to experts, all crowds show deficits and produce results with high false negative rates. This latter issue could be addressed with additional annotation steps that aim at detecting missing annotations as proposed in [82]. Overall we presented three advanced annotations protocols: majority votes, iterative annotations and propagated iterations. Actually these are complementary approaches and could be merged into one big annotation protocol. Majority votes perform best with regard to annotation quality, iterative annotations achieve low false negative rates and its initial annotation which requires most of the processing time can be replaced with propagated annotations as long as appropriate propagation methods are used.

The analysis of **majority votes** shows that majority votes cannot reliably detect the best annotations when annotation candidates have strongly varying quality. In this case, the best result would be achieved by choosing the annotation of the most reliable annotator. For this purpose we could evaluate all previous annotations of the according annotators to assign trust levels. A higher trust level could be assigned to annotators whose previous annotations were close to the final majority votes. Using trust levels could be a good idea in general. They could be utilized whenever decisions between annotations have to be made. Suppose a trustworthy annotator created annotations for an initial iterative annotation step. Afterwards, these annotations are strongly modified by an unreliable or unweighted user. In this case we could preserve the initial annotation by assessing the trust level.
A further approach to improve majority votes and contour annotations in general could be to analyse local image data. Annotated contours that correspond to local image gradients could be preferred for majority voting.

Furthermore, one option to improve results is to run a **supervised quality assurance**. When working with ground truth datasets that might be used for security-relevant applications, quality assurance is necessary in any case. For future projects we aim at efficient quality assurance opportunities. We present some of these attempts in Section 11.1.

While results from Amazon Mechanical Turk were quite accurate after applying majority votes, gaming crowd results could not satisfy our high-quality requirements at all. Beside further usability optimizations regarding our user interface, we could also aim at more **simplified annotation types** to achieve reasonable results from such crowds. In Section 11.2 we present first attempts of using **simple questions** instead of drawing tools, to acquire object annotations.

11.1 Quality Assurance

To accomplish high precision reference data sets, we aim at the highest annotation quality for contours that is achievable. According to previous results and without further effort or supervision, superb MTurkers were the only participants able to compete with our experts so far. To achieve better results in future, we could optimize annotation protocols, post-process algorithms or the usability of our user interface. Beside that, we can also perform a supervised quality assurance. Thus, a few experts evaluate the received annotations and return those micro-tasks, that have been unsatisfying.
Most of the mechanisms presented to improve user contributions can also be used to assist quality assurance methods. For example, suppose we received several annotations to perform majority votes. If many annotations did not reach a consensus with other majority vote candidates for a specific frame, this frame is predestined to contain inappropriate annotations. As a consequence, we could pay particular attention to such suspicious frames. An alternative indicator could be defined for iterative annotation protocols (the presented nothing-to-do-votes). If workers did not reach a consensus after the maximum allowed number of iterations, frames could be flagged as suspicious.

Running a supervised quality assurance for large data sets is a time consuming task. The proposed ***sanity checks*** can help to speed up that measures by unveiling frames and annotations that are suspicious. By the use of reliable sanity checks, suspicious frames can be treated with caution while others could be processed with less attention. First tests show significant time savings when quality controllers can focus on frames that presumably contain a specific error instead of performing a general inspection.
Many further sanity checks are conceivable. The presented propagation methods for example could be used to unveil missing annotations. If subsequent frames do not contain annotations at propagated positions, this might be an indicator for missing subjects.

When working with object correspondences between subsequent frames, sanity checks can even unveil semantic errors. For example, we successfully used a sanity check which detected falsely allocated temporal correspondences of vehicles driving on the opposite lane (according to the viewing direction of the recording camera).

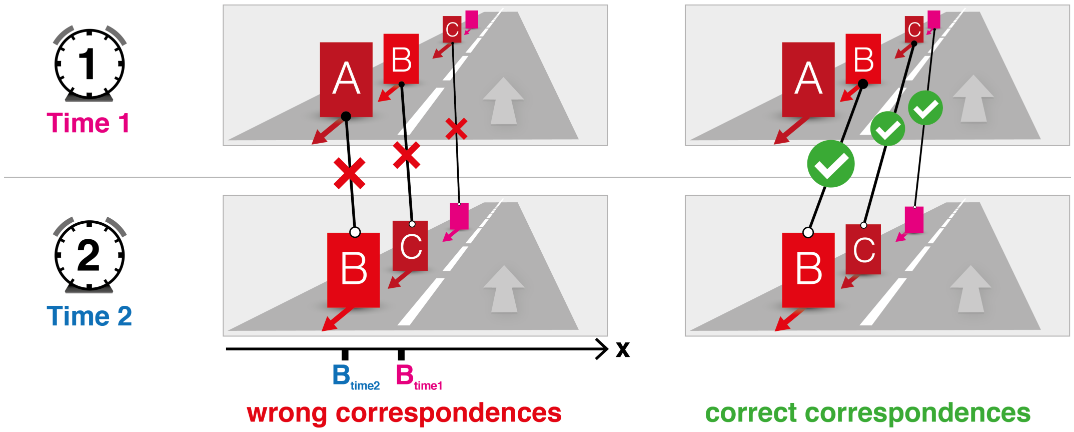
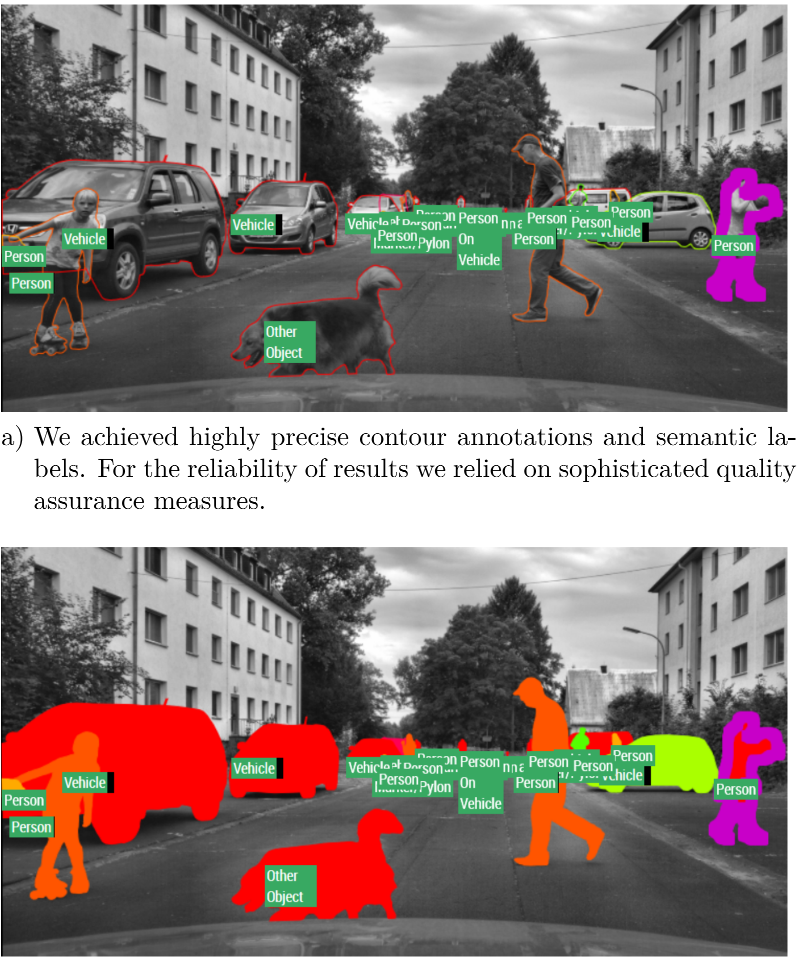


Figure 11.1: ***Detection of false temporal correspondences:*** *We detected falsely assigned correspondences of vehicles by comparing their lateral position p_x. Since vehicles drove on the opposite left lane, their lateral position had to change to the left over time: $p_{x,time(n)} > p_{x,time(n+1)}$. However, since vehicles looked similar, many of them were assigned to false counterparts that did not satisfy the former condition.*

Figure 11.1 illustrates this example. Users had to assign correspondences to objects of subsequent frames. For this purpose, both frames were visualized side by side. Usually, the lateral position of vehicles driving on the opposite left lane should change to the left for chronologically subsequent frames. However, we detected many correspondences that mapped these vehicles to similar looking counterparts whose position was closer to the image center. Obviously, the similar look of vehicles driving in a row misled people to assign false correspondences. Since the image sequence was captured on a straight lane, this sanity check could be applied easily to the according vehicles driving on the opposite lane. As outcome, we detected many false correspondences which were hardly visible to the naked eye.
Further project-specific sanity checks could be related to annotation sizes, aspect ratios, their positions or the number of annotations per frame. It could also be possible to detect inconsistencies by evaluating meta data. We could capture the mouse cursor trajectory while users work on micro-tasks. Parameters such as movement speed, cursor position, and the number of mouse clicks might let us predict annotation accuracy and credibility. Such information could also help to identify weak interface designs and reasons for misuse.

Applying Crowdsourcing to Real Projects

By the use of quality assurance measures and crowds that achieve results comparable to superb MTurkers, we collected contour annotations and semantic labels for several thousand frames. Figure 11.2 shows an example of these annotations that are a part of the *HCI Stereo Ground Truth Dataset*. We experienced that while working with partial hardly distinguishable image contents, a reliable high-precision result is only achievable with appropriate quality assurance methods. While we always had an expert in the loop to ensure data reliability, it is thinkable to redesign and replace some of the mentioned supervised assurance methods by automatic sanity checks returning suspicious annotations to the task queue.



(a) We achieved highly precise contour annotations and semantic labels. For the reliability of results we relied on sophisticated quality assurance measures.

(b) The annotations are a part of the ground truth data set proposed by Kondermann et al. [50], which also includes optical flow data and depth imformation.

Figure 11.2

11.2 Simplifying Annotation Tasks to Binary Questions

Our evaluation of contour annotation tasks in Chapter 10 has shown that even laymen such as our participants from a gaming crowd can create reasonable amounts

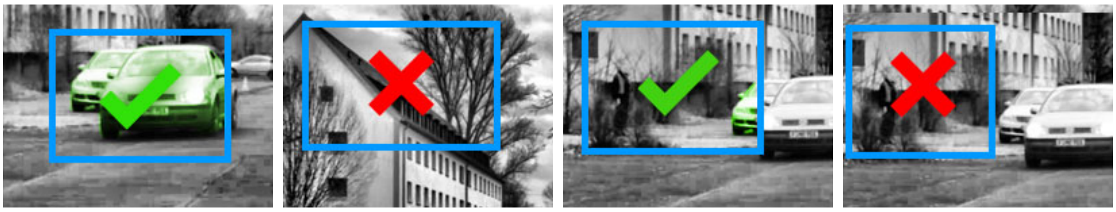


Figure 11.3: *Users of a gaming crowd where asked to answer binary questions about specific image areas. Our first test was to ask them whether a box contains cars or not. Instructions primarily were given visually by the images above.*

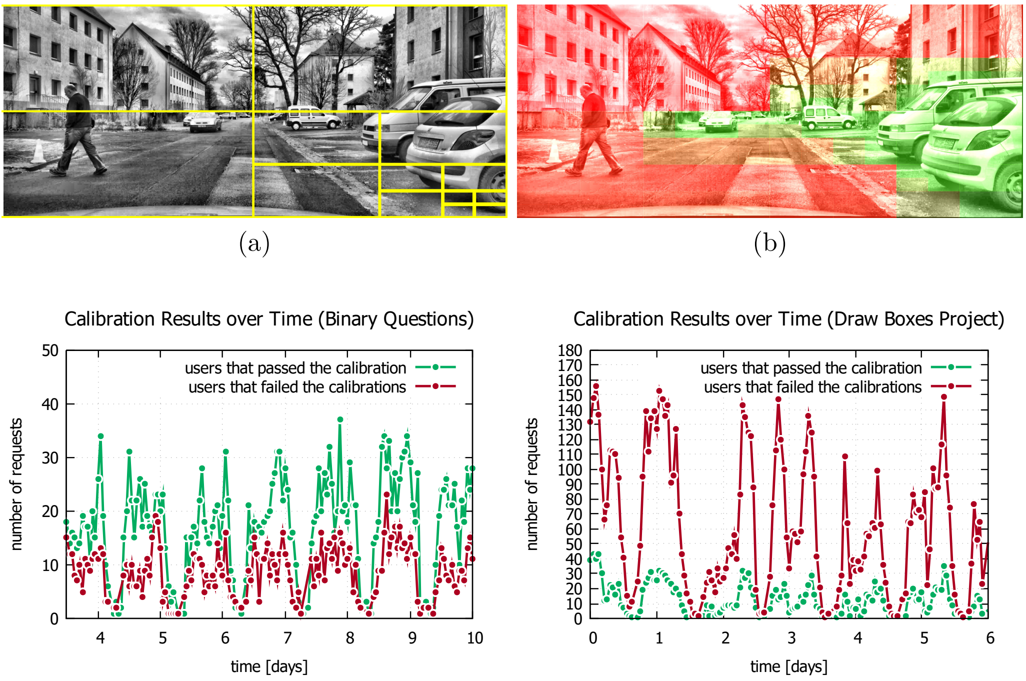
of contour annotations. While majority votes could significantly improve these contributions, the remaining quality still did not satisfy our high-quality requirements. Instead of further optimizing the presented approach to be usable for as many persons as possible, we can also approach the other way around. Thus the question arises which is the easiest user input that can be used to annotate objects. The simplest task that we can set is a binary question. Hence, if it was possible to break down our advanced tasks into simple binary questions, might be achievable by laymen with weak incentives and less motivation.

With regard to this idea, Su et al. [82] proposed a combination of drawing bounding boxes and verifying them with binary questions. Furthermore, several techniques to aggregate binary answers from crowds have been proposed. Their goal is to collect correct values for questions only by evaluating the user's answers. Evaluations of such aggregation methods were presented by Hung et al. [64] and Dalvi et al [21]. Ruprecht et al. [70] present a crowdsourced segmentation approach using a binary question workflow.

Binary questions could be a reasonable mechanism to acquire object annotations with minimal manual effort. With regard to the immense number of questions that could be answered by online crowds, this might be an interesting field of research. For this reason, we rudimentarily tested the applicability of binary questions answered by the gaming crowd. However, the following first experiments and results shall only present principle ideas and opportunities.

The goal was to localize vehicles in automotive images sequences. In contrast to advanced techniques [64, 21, 70] we chose a simple minimal viable setup. Participants where asked, **whether highlighted image areas contain vehicles or not**. To achieve a minimum of required questions to process an image, we chose an iterative approach. Each image was uniformly divided into four region of interests which were labelled. Afterwards we determined the majority vote for ten answers per region. If vehicles were detected, this region was again divided into four subregions. Figure 11.4a shows the superposition of answers for four iterations.

We used a gold standard test to accept only answers of participants with reasonable



(a) (b)

(c) (d)

Figure 11.4: *People were asked to classify image regions as 'contains vehicle(s)' or 'does not contain vehicle(s)'. A previous gold standard test determined much more accepted participants as we achieved for bounding box annotation tests.*

results. The success rate to pass our gold standard filter was high. Figure 11.4c visualizes successful and unsuccessful requests. In comparison, the success rates of gaming crowd participators which passed a gold standard test for bounding box annotations were much lower as can be seen in 11.4d.

All of the resulting majority votes were correct (Figure 11.4b). However, this approach obviously has its drawbacks. The precision of approximated objects depends on the size of labelled image areas and thus on the number of questions. For an annotation accuracy of 5 pixels we would need to set questions about equally-sized regions. Ignoring the reduced number of questions caused by our iterative approach, this would lead to nearly 150,000 questions for an image with 2560 × 1440 pixels. Another issue which is unresolved are the boundaries between vehicles. Our current question only addresses boundaries between vehicles and background. To address this issue we set another question asking **whether a region contains only one or multiple vehicles**. Figure 11.5 illustrates these results. Using this additional question enables us to find also boundaries between vehicles. Furthermore, both questions could be combined providing three answers: "no vehicle", "one vehicle", "multiple vehicles". Though, the issue of numerous required tasks to achieve a high resolution remains.

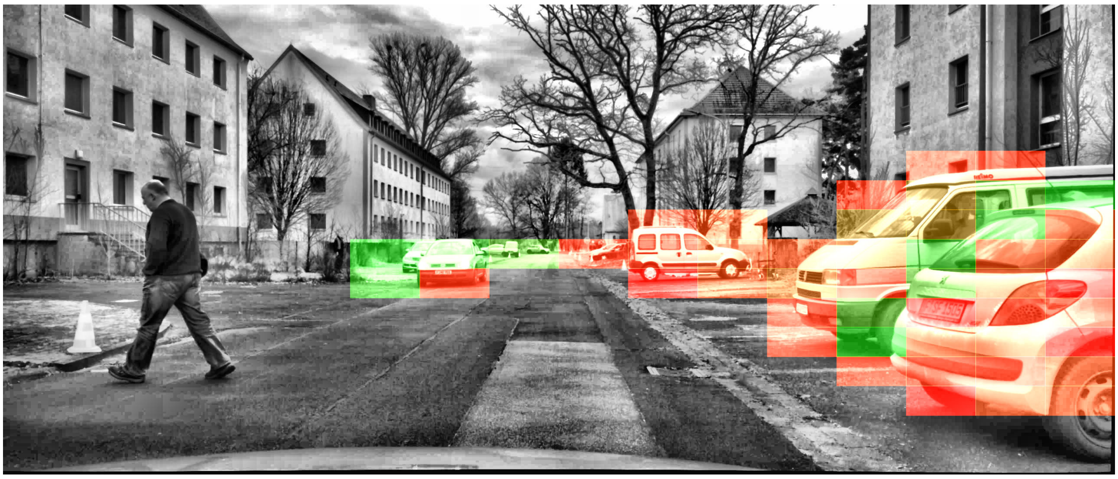
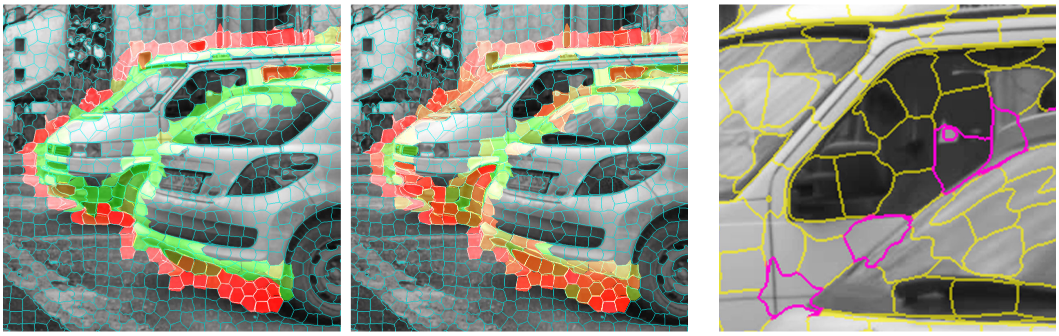


Figure 11.5: *To determine boundaries between objects, we asked whether a box contains more than one vehicle. The green boxes on the lower right corner clearly divide both vehicles.*



(a) Answers for the questions *"Does the segment contain a vehicle?"* and *"Does the segment contain more than one vehicle?"*.

(b) Tattered segments.

Figure 11.6: *Thousands of square boxes would be needed if we wanted to achieve a high-precision result. Thus we try to compensate this weakness by using superpixels. Those segments mostly match object shapes without the need if immense size reductions as needed for square boxes. However, the success rate of our participants to label segments correctly was much lower.*

Instead of showing square regions, we can also utilize segmentation and super pixel algorithms. In the best case, boundaries of super pixels match with boundaries of vehicles. We repeated the two previous questions and set them for segments of the SLIC Superpixel algorithm presented by Achanta et al. [2]. However, any other segmenting algorithm might have also served that purpose. Results weren't as good as in our previous two attempts. A visualization is given in Figure 11.6. We received over 7% wrong majority votes (11% wrong answers overall) for the question whether a segment contains a vehicle and 23% wrong majority votes (34% wrong answers overall) for the question whether a segment contains more than one vehicle.

According to our own experience it is more difficult to interpret the content of segments due to its tattered shape. Figure 11.6b highlights segments that slightly contain pixels of two vehicles. Users need to pay more attention to successfully label these segments. We adjusted the SLIC algorithm to return segments as compact as possible while still describing object shapes. Though, many segments have to be quite tattered to reasonably describe object boundaries.

Outlook

Our first attempt to use binary questions for object annotations was promising. Questions about square regions could successfully be answered. The required number of questions to achieve a reasonable precision can be very high and shows a major drawback. Advanced approaches such as presented by Rupprecht at al. [70] could be an appropriate solution concerning this matter. Answers about tattered segments generated by a superpixel algorithm indicated limits of our approach. However, these were our first attempts on using binary questions. Our user interface could be immensely improved to be used for simple questions. According to our experience with other crowdsourced tasks and with regard to results such as those presented by Hung et al. [64], we expect the general idea to be quite capable. Future attempts could exploit knowledge about previous results to weight user's answers. Furthermore, a combined approach of drawing annotations and answering binary questions as presented by Su et al. [82] could be a reasonable extension.

12 | Conclusion

In this work we investigated manual and semi-automatic approaches that assist computer vision applications. We discussed a semi-automatic approach to acquire depth edges in Part I and presented different approaches to use online crowds for ground truth acquisition in Part II. Since we have already drawn a conclusion for Part I on page 29, this chapter focuses on Part II.

12.1 Summary

In Chapter 6 and 7 we discussed the idea of using crowdsourcing to acquire labour-intensive annotations which are used to either assist the creation of ground truth datasets or which can directly be used for performance analysis. Due to related crowdsourced approaches that achieved reasonable results for low level vision applications such as motion analysis, we decided to analyse the capabilities of crowdsourcing with regard to highly-precise contour annotations. While we primarily focused on high-quality results that are usable for the purpose of ground truth creation, the acquired annotations could of course also be used to train learning-based applications.

In Chapter 8 we introduced the general principle of crowdsourcing, presented typical crowds that can be engaged and we discussed previous approaches that use crowds for general computer vision applications. While we discussed promising approaches for a variety of applications, to the best of our knowledge none of these contributions addressed high-quality contour annotations (on a pixel-level) for large ground truth datasets.

In Chapter 9 we presented our experimental setup to acquire and evaluate crowdsourced contour annotations. We chose test sequences with an automotive context, since automotive computer vision algorithms are a current field of research that strongly benefits from precise annotations to train and benchmark driver assistance algorithms.
We introduced four user groups whose annotations were evaluated. Besides ex-

perts, these were Amazon Mechanical Turk workers, a subgroup of experienced (superb) Amazon Mechanical Turk workers and users of an online gaming crowd. While Amazon Mechanical Turk represents typical crowds that can be engaged for crowdsourced micro-tasks, the gaming crowd usually is only familiar with tasks such as watching video ads to gain in-game rewards. However, in contrast to dedicated crowds, gaming crowds consists of considerably more users that may solve our tasks.
With regard to the acquisition process of our ground truth data which was used to benchmark user annotations, we emphasized that images normally do not show sharp edges for object boundaries. The size of object boundaries can vary within several pixels. As a consequence, we provided inner and outer ground truth annotations to address ambiguous edges, the average width of this edge area was 1.7 ± 0.6 pixels.
Since the comparison of contours is not trivial, we presented various performance indicators such as the Jaccard distance which performs well with regard to concave shapes but is inappropriate to compare differently-sized contours. As further indicators we used distance maps to determine absolute deviations between contours, false negative (and positive) rates and the processing time.

Our experiments and results were discussed in Chapter 10. We basically presented four different approaches to acquire object annotations for vehicles.
First we analysed the quality of single annotations created with our web-based annotation tool by comparing them to ground truth annotations as well as to annotations acquired by experts. We primarily discussed the amount of annotated contour points lying within a 2 pixel and 5 pixel tolerance regarding the ground truth contour. These are typical values requested for various types of ground truth datasets. In addition to that, we aimed to receive at most 5% outliers. Best results were achieved by experienced Amazon Mechanical Turk workers. 99.5% of annotated points lied within the 5 pixel tolerance which was equal to the value of experts. Regarding the 2 pixel tolerance, experienced MTurkers achieved a value of 96.1% which is slightly lower than the 96.8% achieved by experts. While these values were highly satisfying, experienced MTurkers missed to annotate 33.9% of all subjects. The reason for this high false negative rate may have been inappropriate task instructions. The false negative rate for missing annotations that could not be caused by vague instructions was 0.3%. The results of random MTurkers and gaming crowd users were worse. The amount of pixels lying within the 5 pixel tolerance was 89.5% for random MTurkers and only 50.6% for annotations of gaming crowd participants. Random MTurkers missed to annotate 46% of all objects and 10% of unambiguous objects regarding vague instructions. Gaming crowd participants missed 26% objects whereas 1.2% of these objects were unambiguous. We concluded that these first results are promising for further approaches but do not satisfy the stated high-quality requirements.

Our first approach to improve annotation quality was the usage of **majority votes**.

We proposed an *annotation-based* majority vote whereas one unique annotation was selected from a set of annotations which all addressed the same object. Furthermore we proposed point-wise majority votes. In comparison to annotation-based majority votes, this approach combined several annotations by choosing that contour points which minimize the sum of distances with regard to their nearest neighbour points from other annotations. Both majority votes improved annotations significantly whereas the point-wise method achieved the better results. Former outliers could be reduced by 62% regarding the 2 pixel tolerance. Thus, the resulting annotation quality of experienced annotators even exceeded the quality of single expert annotations ($\text{Inliers}_{<2px,superbMturkers}$=98.5%, $\text{Inliers}_{<5px,superbMturkers}$=99.7%). False negative rates decreased from 33.9% to 19.4% which would still be an inadequate value. However, none of the unambiguous objects were missed. The impact of majority votes for random MTurk annotations and gaming crowd annotations was comparable. Outliers decreased by 61% (MTurk) and 36% (gaming crowd) regarding the 2 pixel tolerance and by 44% (MTurk) and 74% (gaming crowd) regarding the 5 pixel tolerance. False negative rates decreased from 46% to 24% (1.6% for unambiguous objects) for random MTurkers but did not change significantly for gaming crowd results.
Summarizing, majority votes significantly improved annotation accuracy and are highly recommendable once the increased processing time and related costs caused by additional required annotations are acceptable. Especially the huge improvements for initially roughly annotated contours of gaming crowd participants (which in general are unfamiliar with annotation tasks) are a noteworthy result.
A further observed characteristic of majority votes was that the annotation quality could only be maximized (choosing the best annotation from all candidates or determining a point-wise majority vote that is at least as good as the best single annotation), when the accuracy of majority vote candidates was similar. If this was not the case, it would have been preferable to choose the single best annotation of the most reliable annotator. Though, all majority votes were better than the sum of single annotations.
Since we collected multiple annotations per object to perform majority votes, we could also use this additional information to define a confidence measure. We introduced the possibility to determine inner and outer contour tolerances with regard to according majority vote contours. The deviation of these annotations represent the precision and reliability of resulting majority votes. A special case was discussed for generally precise annotations of objects with partial ambiguous edges. When the average annotation deviation is low in contrast to the size of ambiguities, a partial stronger deviation of annotation precision can be an indicator for ambiguous edges. With that kind of information, resulting ground truth datasets can even address ambiguous object edges for which single annotations would be inappropriate. Knowledge about ambiguities could be important when using annotations as training sets or for evaluations of algorithms.

Our second approach to improve single annotations was to use an **iterative an-**

notation protocol. Initial annotations were returned to further annotators that could improve and complete the former contribution. While this approach could not achieve the accuracy of majority votes, it performed better with regard to false negative values. According values decreased from 34% to 15% for experienced Mturkers and from 46% to 12.9% for random MTurkers. Annotators did not miss unambiguous objects.
A further advantage of this iterative approach over majority votes was the moderate increase of processing time. While the usage of majority votes increase the processing time by a factor equally to the number of majority vote candidates, the working time on iterative tasks decreased steadily for subsequent iterations. Times increased by factor 1.3 for experienced MTurkers and by 2.7 for random MTurkers.

As a consequence, we analysed the **effectiveness of propagating previous annotations** to subsequent frames to use them as initial annotation within an iterative annotation protocol. For reasons of simplification we used simple but performant chamfer matchings to propagate contours. We asked participants to annotate contours of pedestrians within an automotive scenario. In contrast to previous annotations of vehicles, human contours had concave shapes with more details which resulted in a longer annotation time. Our simple propagation method could not achieve an adequate quality as experts and experienced annotators needed at least the same time to correct these propagations as to add annotations from scratch. Results of inexperienced annotators benefited from this approach which however was a result of according inappropriate annotations workflows. The reason that annotators could not benefit from propagated initial guesses was the inaccurate propagation quality (which required annotators to adjust nearly every single contour point) and the laborious adjustment tool itself.
Since we expected this approach to achieve more desirable results when using qualitatively better propagations and more efficient adjustment tools, we applied this idea to a more favourable setup. We asked participants to annotate bounding boxes of vehicles which can be propagated more reliably and are easier to adjust afterwards. Under these conditions experts and Mechanical Turk workers achieved significantly lower annotation times while quality was comparable. The processing time decreased by 50% for experts and by 43% for MTurkers. Due to these results, we assume an even larger benefit for time-consuming contour annotations given that the propagation quality improves notably and adjustment tools for contours are easier to use. A promising approach could be to use contour propagations as proposed in Part I and [5].

Finally we discussed **processing times** for all observed user groups. Experienced Mechanical Turk workers were the only user group that could compete with experts with regard to both, quality and working time as well. However we could not achieve a reasonable amount of participants to deal with large datasets by using Amazon Mechanical Turk only. Though, apart from the presented user groups, we also have successfully run crowdsourced ground truth acquisitions with crowds whose skill

was comparable to experienced MTurkers and who performed efficiently by using an iterative annotation protocol. These crowds can provide up to a few hundreds of workers which is adequate for acquisition projects with several thousands of images.
Our participants from the gaming crowd could not create adequate annotations usable for high-quality datasets, though the improvements made by majority votes were promising and motivating. The accuracy of 87% of annotated contour points deviating less than 5 pixels from ground truth can be considerable for *weak ground truth* which can be used for high level vision such as learning approaches. These annotators achieved low processing times in comparison to all other user groups. Keeping in mind that plenty of such crowds exist, it seems highly promising to run further attempts to benefit from large crowds of laymen.

12.2 Outlook

We presented possible improvements for most of the presented approaches in the respective sections. Further approaches and annotation protocols that could bring improvements with regard to annotation quality and efficiency were discussed in Chapter 11.
From a more general point of view, we think that crowdsourcing can bring many other opportunities in future that benefit from the vast number of users that could supply intuitiveness and perceptive capabilities. A major challenge to successfully involve crowdsourcing is to provide adequate micro-tasks. While single-user applications often provide sophisticated tools with many options to address a variety of purposes, we have to rethink this approach when relying on crowdsourced tasks. In the latter case, the efficiency should strongly increase when breaking down formerly extensive applications into separated micro-tasks which are easy and intuitively to solve. A first attempt to strongly simplify task complexity by asking binary questions was presented in 11.2. While this approach needs to be further improved to run efficiently, results were promising with regard to the large number of questions we could address with crowds such as the evaluated gaming user group. Furthermore, these simple micro-tasks may be an appropriate annotation type to be run on mobile devices, which could be another source of large crowds.
In 8.5 we presented several quality assurance methods and named approaches which we utilized successfully. While we apply supervised quality assurance methods for real ground truth acquisition projects for now, this working step could be largely replaced by automatic sanity checks which return suspicious annotations to the task queue.

Summarizing, crowdsourcing can enable new opportunities with regard to computer vision applications that may have been infeasible due to labour-intensity otherwise. As an example, we have shown that using crowds to acquire high-quality annotations can be an adequate approach. While several computer vision applications successfully utilized crowdsourcing capabilities for now, we expect that there are yet many more to come.

Part III

Appendix

A | Supplemental Tables

The following tables contain additional values for our analysis of majority votes, interative annotation protocols and the according working times.

	MTurk Superb			
	All	MV_1	$PWMV_1$	NTDV
Jaccard [10^{-2}]	1.2 ± 1.4	1.2 ± 1.4	0.9 ± 1.1	1.5 ± 2.0
D_{mean} $[px]$	0.4 ± 0.6	0.7 ± 0.5	0.2 ± 0.4	0.4 ± 0.6
D_{median} $[px]$	0	0	0	0
$D_{percentile(5)}$ $[px]$	0	0	0	0
$D_{percentile(25)}$ $[px]$	0	0	0	0
$D_{percentile(75)}$ $[px]$	0.7	0.3	0.3	0.7
$D_{percentile(95)}$ $[px]$	4.0	2.6	2.3	3.6
Inliers (≤2px)	96.1%	97.8%	98.5%	96.3%
Inliers (≤5px)	99.5%	99.7%	99.7%	99.6%
FP	0	0%	0%	0%
FN_{all}	33.9%	19.4%	19.4%	14.5%
$FN_{>50px,uncut}$	0.3%	0%	0%	0%
$FN_{17px<size\leq50px}$	11.6%	3.2%	3.2%	3.2%
FN_{cut}	10.6%	4.8%	4.8%	0%
$FN_{size\leq17px}$	11.3%	11.3%	11.3%	11.3%

Table A1: *Results of superb MTurkers processing advanced annotation protocols. All false negative percentages are calculated in regard to the total number of ground truth annotations.*

	MTurk All							
	All	MV_1	$PWMV_1$	MV_2	$PWMV_2$	MV_3	$PWMV_3$	Iter.
Jaccard $[10^{-2}]$	2.8 ± 3.7	1.4 ± 1.5	1.2 ± 1.2	1.6 ± 2.0	1.6 ± 1.4	1.7 ± 1.3	1.3 ± 1.3	0.8 ± 0.8
D_{mean} $[px]$	1.8 ± 1.9	0.7 ± 0.9	0.6 ± 0.9	0.8 ± 1.1	0.7 ± 1.0	0.7 ± 1.0	0.7 ± 1.0	1.0 ± 1.4
D_{median} $[px]$	1.2	0.4	0.2	0.4	0.4	0.6	0.3	0.5
$D_{percentile(5)}$ $[px]$	0	0	0	0	0	0	0	0
$D_{percentile(25)}$ $[px]$	0	0	0	0	0	0	0	0
$D_{percentile(75)}$ $[px]$	2.3	0.9	0.7	1.0	0.6	1.0	0.9	1.4
$D_{percentile(95)}$ $[px]$	22.0	7.7	7.5	9.0	8.6	7.9	7.9	9.2
Inliers (≤2px)	78.5%	90.6%	91.6%	89.0%	89.7%	89.3%	89.6%	86.7%
Inliers (≤5px)	89.6%	97.2%	97.6%	96.2%	96.3%	96.7%	97.1%	97.2%
FP	0.5%	0%	0%	0%	0%	0%	0%	0%
FN_{all}	45.5%	25.8%	25.8%	24.2%	24.2%	38.7%	38.7%	12.4%
$FN_{>50px,uncut}$	9.5%	1.6%	1.6%	0%	0%	3.2%	3.2%	0%
$FN_{17px<size\leq50px}$	13.7%	8.1%	8.1%	8.1%	8.1%	14.5%	14.5%	4.1%
FN_{cut}	11.7%	4.8%	4.8%	4.8%	4.8%	9.7%	9.7%	0%
$FN_{size\leq17px}$	10.7%	11.3%	11.3%	11.3%	11.3%	11.3%	11.3%	8.3%

Table A2: *Results of all MTurkers processing advanced annotation protocols. All false negative percentages are calculated in regard to the total number of ground truth annotations.*

	Gaming Crowd						
	All	MV_1	$PWMV_1$	MV_2	$PWMV_2$	MV_3	$PWMV_3$
Jaccard [10^{-2}]	8.7 ± 5.2	8.8 ± 7.4	8.3 ± 6.4	6.2 ± 6.7	5.7 ± 5.8	5.1 ± 4.1	4.7 ± 3.3
D_{mean} [px]	6.7 ± 5.7	5.2 ± 4.1	5.1 ± 4.3	3.2 ± 2.9	2.7 ± 2.4	2.7 ± 2.7	2.4 ± 2.4
D_{median} [px]	5.5	4.4	4.2	2.4	2.0	2.2	1.8
$D_{percentile(5)}$ [px]	0	0	0	0	0	0	0
$D_{percentile(25)}$ [px]	1.0	0.3	0.3	0	0	0	0
$D_{percentile(75)}$ [px]	12.1	9.7	8.7	5.8	4.9	5.7	4.1
$D_{percentile(95)}$ [px]	33.0	30.0	34.9	17.5	15.0	14.9	14.3
Inliers (≤2px)	27.7%	38.2%	42.7%	49.6%	51.3%	49.3%	54.0%
Inliers (≤5px)	50.6%	63.3%	68.9%	78.4%	81.5%	79.5%	86.9%
FP	0.9%	0%	0%	0%	0%	0%	0%
FN_{all}	25.8%	22.7%	22.7%	20.5%	20.5%	25.0%	25.0%
$FN_{>50px,uncut}$	1.2%	2.3%	2.3%	0%	0%	2.3%	2.3%
$FN_{17px<size\leq50px}$	3.6%	11.4%	11.4%	11.4%	11.4%	13.6%	13.6%
FN_{cut}	18.8%	2.3%	2.3%	2.3%	2.3%	2.3%	2.3%
$FN_{size\leq17px}$	2.3%	6.8%	6.8%	6.8%	6.8%	6.8%	6.8%

Table A3: *Results of the gaming crowd processing advanced annotation protocols. All false negative percentages are calculated in regard to the total number of ground truth annotations.*

	Experts			**Superb MTurkers**		**All MTurkers**		
	Add	Automatic Propagation	Edited Prop.	Add	Edited Prop.	Add	Add MV	Edited Prop.
Jaccard [10^{-2}]	3.5 ± 1.8	13.8 ± 3.7	3.3 ± 1.1	3.5 ± 2.0	3.6 ± 3.5	10.9 ± 7.4	6.0 ± 4.9	6.9 ± 4.1
$Distance_{mean}$ $[px]$	0.5 ± 1.0	2.1 ± 1.9	0.5 ± 0.9	0.6 ± 1.1	0.5 ± 0.7	1.8 ± 2.1	0.8 ± 1.0	1.0 ± 1.3
D_{median} $[px]$	0.1	1.7	0.2	0.1	0.1	1.2	0.5	0.6
$D_{percentile(5)}$ $[px]$	0	0	0	0	0	0	0	0
$D_{percentile(25)}$ $[px]$	0	0.3	0	0	0		0	0
$D_{percentile(75)}$ $[px]$	0.7	3.6	0.7	0.7	0.7	3.0	1.0	1.7
$D_{percentile(95)}$ $[px]$	5.2	8.5	3.7	11.3	4.7	15.2	6.4	7.7
Inliers ($\leq$2px)	94.7%	57.0%	94.1%	94.5%	93.4%	71.1%	87.5%	84.0%
Inliers ($\leq$5px)	98.8%	91.9%	99.2%	98.2%	98.7%	91.0%	97.5%	97.0%
Time$_{mean}$ [s]	241 ± 208		196 ± 19	112 ± 116	306 ± 95	155 ± 122		181 ± 164
Time$_{min}$ [s]	78		164	55	125	19		6
Time$_{1-quartile}$ [s]	127		186	69	266	62		63
Time$_{median}$ [s]	207		196	87	287	111		112
Time$_{3-quartile}$ [s]	228		202	107	308	208		238
Time$_{max}$ [s]	806		229	686	596	503		662

Table A4: *Comparison between simply added annotations and edited annotations based on propagations.*

B | Lists

B.1 List of Figures

B.2 List of Tables

C | Bibliography

[1] A. M. S. A. P. Dawid. "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 20–28. ISSN: 00359254, 14679876. URL: http://www.jstor.org/stable/2346806.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282. DOI: 10.1109/tpami.2012.120.

[3] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. "A Database and Evaluation Methodology for Optical Flow". In: *International Journal of Computer Vision* 92.1 (2010), pp. 1–31. ISSN: 1573-1405. DOI: 10.1007/s11263-010-0390-2.

[4] W. Barfield, ed. *Fundamentals of Wearable Computers and Augmented Reality, Second Edition.* Informa UK Limited, 2015. DOI: 10.1201/b18703.

[5] M. Becker, M. Baron, D. Kondermann, M. Bussler, and V. Helzle. "Movie dimensionalization via sparse user annotations". In: *3DTV-Conference: The True Vision-Capture, Transmission and Dispaly of 3D Video (3DTV-CON), 2013.* 2013, pp. 1–4. DOI: 10.1109/3DTV.2013.6676633.

[6] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini. "An interactive tool for manual, semi-automatic and automatic video annotation". In: *Computer Vision and Image Understanding* 131 (2015), pp. 88–99. DOI: 10.1016/j.cviu.2014.06.015.

[7] S. Bianco, G. Ciocca, P. Napoletano, R. Schettini, R. Margherita, G. Marini, and G. Pantaleo. "Cooking action recognition with ivat: an interactive video annotation tool". In: *Image Analysis and Processing–ICIAP 2013.* Springer, 2013, pp. 631–641. DOI: 10.1007/978-3-642-41184-7_64.

[8] G. Borgefors. "Hierarchical chamfer matching: a parametric edge matching algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.6 (1988), pp. 849–865. DOI: 10.1109/34.9107.

[9] S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie. “The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization”. In: *International Journal of Computer Vision* 108.1-2 (2014), pp. 3–29. URL: http://dx.doi.org/10.1007/s11263-014-0698-4.

[10] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. “Visual recognition with humans in the loop”. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 438–451. URL: http://dx.doi.org/10.1007/978-3-642-15561-1_32.

[11] T. Brosnan and D.-W. Sun. “Improving quality inspection of food products by computer vision—a review”. In: *Journal of Food Engineering* 61.1 (2004). Applications of computer vision in the food industry, pp. 3 –16. ISSN: 0260-8774. DOI: http://dx.doi.org/10.1016/S0260-8774(03)00183-3. URL: http://www.sciencedirect.com/science/article/pii/S0260877403001833.

[12] G. J. Brostow, J. Fauqueur, and R. Cipolla. “Semantic object classes in video: A high-definition ground truth database”. English. In: *Pattern Recognition Letters* 30.2 (2009), pp. 88–97. DOI: 10.1016/j.patrec.2008.04.005.

[13] M. Buhrmester, T. Kwang, and S. D. Gosling. “Amazon’s Mechanical Turk a new source of inexpensive, yet high-quality, data?” In: *Perspectives on psychological science* 6.1 (2011), pp. 3–5. DOI: 10.1177/1745691610393980.

[14] J. T. Bushberg and J. M. Boone. *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011. ISBN: 978-0781780575.

[15] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. “Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI”. In: ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. Chap. A Naturalistic Open Source Movie for Optical Flow Evaluation, pp. 611–625. ISBN: 978-3-642-33783-3. DOI: 10.1007/978-3-642-33783-3_44.

[16] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. “Two Deterministic Half-Quadratic Regularization Algorithms for Computed Imaging”. In: *ICIP (2)*. 1994, pp. 168–172. DOI: 10.1109/ICIP.1994.413553.

[17] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, et al. “Predicting protein structures with a multiplayer online game”. In: *Nature* 466.7307 (2010), pp. 756–760. DOI: 10.1038/nature09304.

[18] S. Cooper, A. Treuille, J. Barbero, A. Leaver-Fay, K. Tuite, F. Khatib, A. C. Snyder, M. Beenen, D. Salesin, D. Baker, et al. "The challenge of designing scientific discovery games". In: *Proceedings of the Fifth international Conference on the Foundations of Digital Games*. ACM. 2010, pp. 40–47. DOI: 10.1145/1822348.1822354. URL: http://doi.acm.org/10.1145/1822348.1822354.

[19] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The Cityscapes Dataset". In: *CVPR Workshop on The Future of Datasets in Vision*. 2015.

[20] C. Creusot and N. Courty. "Ground truth for pedestrian analysis and application to camera calibration". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE. 2013, pp. 712–718. DOI: 10.1109/CVPRW.2013.108.

[21] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. "Aggregating Crowdsourced Binary Ratings". In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: International World Wide Web Conferences Steering Committee, 2013, pp. 285–294. ISBN: 978-1-4503-2035-1. URL: http://dl.acm.org/citation.cfm?id=2488388.2488414.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255. URL: http://dx.doi.org/10.1109/cvpr.2009.5206848.

[23] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. "From game design elements to gamefulness: defining gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. ACM. 2011, pp. 9–15. DOI: 10.1145/2181037.2181040.

[24] R Di Salvo, D Giordano, and I Kavasidis. "A crowdsourcing approach to support video annotation". In: *Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications*. ACM. 2013, p. 8. DOI: 10.1145/2501105.2501113.

[25] A. Donath and D. Kondermann. "Computer Vision Systems: 9th International Conference, ICVS 2013, St. Petersburg, Russia, July 16-18, 2013. Proceedings". In: ed. by M. Chen, B. Leibe, and B. Neumann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Is Crowdsourcing for Optical Flow Ground Truth Generation Feasible?, pp. 193–202. ISBN: 978-3-642-39402-7. DOI: 10.1007/978-3-642-39402-7_20.

[26] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2 (2010), pp. 303–338. DOI: 10.1007/s11263-009-0275-4.

[27] L. Fei-Fei, R. Fergus, and P. Perona. "One-shot learning of object categories". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28.4 (2006), pp. 594–611. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2006.79.

[28] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. "Annotating Named Entities in Twitter Data with Crowdsourcing". In: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. CSLDAMT '10. Los Angeles, California: Association for Computational Linguistics, 2010, pp. 80–88. URL: http://dl.acm.org/citation.cfm?id=1866696.1866709.

[29] U. Franke, D. Pfeiffer, C. Rabe, C. Knoeppel, M. Enzweiler, F. Stein, and R. G. Herrtwich. "Making Bertha See". In: *Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops*. ICCVW '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 214–221. ISBN: 978-1-4799-3022-7. DOI: 10.1109/ICCVW.2013.36.

[30] Y. Furukawa and J. Ponce. "Accurate, Dense, and Robust Multi-View Stereopsis". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32.8 (2010), pp. 1362–1376. DOI: 10.1109/TPAMI.2009.161.

[31] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[32] G. Griffin, A. Holub, and P. Perona. "Caltech-256 object category dataset". In: (2007). URL: http://www.vision.caltech.edu/Image_Datasets/Caltech101/.

[33] M. Grundmann, V. Kwatra, M. Han, and I. Essa. "Efficient hierarchical graph-based video segmentation". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 2141–2148. DOI: 10.1109/CVPR.2010.5539893.

[34] M. Guttmann, L. Wolf, and D. Cohen-Or. "Semi-automatic stereo extraction from video footage". In: *Computer Vision, 2009 IEEE 12th International Conference on*. 2009, pp. 136–142. DOI: 10.1109/ICCV.2009.5459158.

[35] K. He, J. Sun, and X. Tang. "Fast matting using large kernel matting laplacian matrices". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 2165–2172. DOI: 10.1109/CVPR.2010.5539896.

[36] H. Hirschmuller. "Accurate and efficient stereo processing by semi-global matching and mutual information". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 807–814. DOI: 10.1109/CVPR.2005.56.

[37] C.-J. Ho, T.-H. Chang, J.-C. Lee, J. Y.-j. Hsu, and K.-T. Chen. "KissKiss-Ban: a competitive human computation game for image annotation". In: *Proceedings of the acm sigkdd workshop on human computation.* ACM. 2009, pp. 11–14. DOI: 10.1145/1600150.1600153.

[38] K. Honauer, L. Maier-Hein, and D. Kondermann. "The HCI Stereo Metrics: Geometry-Aware Performance Analysis of Stereo Algorithms". In: *The IEEE International Conference on Computer Vision (ICCV).* 2015. URL: hciweb.iwr.uni-heidelberg.de/stereometrics.

[39] D. Hong, H. Lee, M. Y. Kim, H. Cho, and J. I. Moon. "Sensor fusion of phase measuring profilometry and stereo vision for three-dimensional inspection of electronic components assembled on printed circuit boards". In: *Appl. Opt.* 48.21 (2009), pp. 4158–4169. DOI: 10.1364/AO.48.004158. URL: http://ao.osa.org/abstract.cfm?URI=ao-48-21-4158.

[40] J. J. Horton and L. B. Chilton. "The labor economics of paid crowdsourcing". In: *Proceedings of the 11th ACM conference on Electronic commerce.* ACM. 2010, pp. 209–218.

[41] J. Howe. "The rise of crowdsourcing". In: *Wired magazine* 14.6 (2006), pp. 1–4. URL: http://www.wired.com/2006/06/crowds/.

[42] W. Hu, T. Tan, L. Wang, and S. Maybank. "A survey on visual surveillance of object motion and behaviors". In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 34.3 (2004), pp. 334–352. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2004.829274.

[43] P. G. Ipeirotis. "Analyzing the Amazon Mechanical Turk Marketplace". In: *XRDS* 17.2 (Dec. 2010), pp. 16–21. ISSN: 1528-4972. DOI: 10.1145/1869086.1869094.

[44] K. Karsch, C. Liu, and S. B. Kang. "Depth extraction from video using non-parametric sampling". In: *Computer Vision–ECCV 2012.* Springer, 2012, pp. 775–788. DOI: 10.1109/TPAMI.2014.2316835.

[45] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato. "An innovative web-based collaborative platform for video annotation". In: *Multimedia Tools and Applications* 70.1 (2014), pp. 413–432. ISSN: 1573-7721. DOI: 10.1007/s11042-013-1419-7.

[46] F. Khatib, F. DiMaio, S. Cooper, M. Kazmierczyk, M. Gilski, S. Krzywda, H. Zabranska, I. Pichova, J. Thompson, Z. Popović, et al. "Crystal structure of a monomeric retroviral protease solved by protein folding game players". In: *Nature structural & molecular biology* 18.10 (2011), pp. 1175–1177. DOI: 10.1038/nsmb.2119.

[47] I. S. Kim, H. S. Choi, K. M. Yi, J. Y. Choi, and S. G. Kong. "Intelligent visual surveillance — A survey". In: *International Journal of Control, Automation and Systems* 8.5 (2010), pp. 926–939. ISSN: 2005-4092. DOI: 10.1007/s12555-010-0501-4.

[48] A. Kittur, E. H. Chi, and B. Suh. “Crowdsourcing user studies with Mechanical Turk”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2008, pp. 453–456. DOI: 10.1145/1357054.1357127.

[49] D. Kondermann. “Ground Truth Design Principles: An Overview”. In: *Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications*. VIGTA ’13. St. Petersburg, Russia: ACM, 2013, 5:1–5:4. ISBN: 978-1-4503-2169-3. DOI: 10.1145/2501105.2501114.

[50] D. Kondermann, R. Nair, S. Meister, W. Mischler, B. Güssefeld, K. Honauer, S. Hofmann, C. Brenner, and B. Jähne. “Stereo Ground Truth with Error Bars”. In: *Computer Vision–ACCV 2014*. Springer, 2015, pp. 595–610. URL: http://dx.doi.org/10.1007/978-3-319-16814-2_39.

[51] A. Kumar. “Computer-Vision-Based Fabric Defect Detection: A Survey”. In: *Industrial Electronics, IEEE Transactions on* 55.1 (2008), pp. 348–363. ISSN: 0278-0046. DOI: 10.1109/TIE.1930.896476.

[52] J. M. Leimeister. “Crowdsourcing”. In: *Controlling & Management* 56 (2012), pp. 388–392. DOI: 10.1365/s12176-012-0662-5.

[53] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. “Turkit: human computation algorithms on mechanical turk”. In: *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. ACM. 2010, pp. 57–66. DOI: 10.1145/1866029.1866040.

[54] C. Liu, W. Freeman, E. Adelson, and Y. Weiss. “Human-assisted motion annotation”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587845.

[55] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003, 284—292. DOI: 10.5860/CHOICE.41-5949.

[56] L. Maier-Hein, D. Kondermann, T. Roß, S. Mersmann, E. Heim, S. Bodenstedt, H. G. Kenngott, A. Sanchez, M. Wagner, A. Preukschas, A.-L. Wekerle, S. Helfert, K. März, A. Mehrabi, S. Speidel, and C. Stock. “Crowdtruth validation: a new paradigm for validating algorithms that rely on image correspondences”. In: *International Journal of Computer Assisted Radiology and Surgery* 10.8 (2015), pp. 1201–1212. ISSN: 1861-6429. DOI: 10.1007/s11548-015-1168-3.

[57] L. Maier-Hein, S. Mersmann, D. Kondermann, S. Bodenstedt, A. Sanchez, C. Stock, H. Kenngott, M. Eisenmann, and S. Speidel. “Can Masses of Non-Experts Train Highly Accurate Image Classifiers?” English. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*. Ed. by P. Golland, N. Hata, C. Barillot, J. Hornegger, and R. Howe. Vol. 8674. Lecture Notes in Computer Science. Springer International Publishing, 2014,

pp. 438–445. ISBN: 978-3-319-10469-0. DOI: 10.1007/978-3-319-10470-6_55.

[58] D. Martin, C. Fowlkes, D. Tal, and J. Malik. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on.* Vol. 2. 2001, 416–423 vol.2. DOI: 10.1109/ICCV.2001.937655.

[59] W. Mason and D. J. Watts. "Financial Incentives and the "Performance of Crowds"". In: *Proceedings of the ACM SIGKDD Workshop on Human Computation.* HCOMP '09. Paris, France: ACM, 2009, pp. 77–85. ISBN: 978-1-60558-672-4. DOI: 10.1145/1600150.1600175.

[60] S. Meister, S. Izadi, P. Kohli, M. Hämmerle, C. Rother, and D. Kondermann. "When Can We Use KinectFusion for Ground Truth Acquisition?" In: *Workshop on Color-Depth Camera Fusion in Robotics, IEEE International Conference on Intelligent Robots and Systems.* 1. 2012.

[61] S. Meister, B. Jähne, and D. Kondermann. "Outdoor stereo camera system for the generation of real-world benchmark data sets". In: *Optical Engineering* 51.2 (2012), pp. 021107–1–021107–6. DOI: 10.1117/1.OE.51.2.021107.

[62] D.-I. B. Meixner. "Annotated Interactive Non-linear Video". In: (2014).

[63] A. J. Quinn and B. B. Bederson. "Human computation: a survey and taxonomy of a growing field". In: *Proceedings of the SIGCHI conference on human factors in computing systems.* ACM. 2011, pp. 1403–1412. URL: http://dx.doi.org/10.1145/1978942.1979148.

[64] N. Quoc Viet Hung, N. T. Tam, L. N. Tran, and K. Aberer. "Web Information Systems Engineering – WISE 2013: 14th International Conference, Nanjing, China, October 13-15, 2013, Proceedings, Part II". In: ed. by X. Lin, Y. Manolopoulos, D. Srivastava, and G. Huang. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. An Evaluation of Aggregation Techniques in Crowdsourcing, pp. 1–15. ISBN: 978-3-642-41154-0. DOI: 10.1007/978-3-642-41154-0_1.

[65] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, P. Murray, K. Schawinski, A. S. Szalay, and J. Vandenberg. "Galaxy Zoo: Exploring the Motivations of Citizen Science Volunteers". In: *Astronomy Education Review* 9.1 (2010), p. 010103. DOI: 10.3847/AER2009036. arXiv: 0909.2925 [astro-ph.IM].

[66] V. Rajan, S. Bhattacharya, L. E. Celis, D. Chander, K. Dasgupta, and S. Karanam. "CrowdControl: An online learning approach for optimal task scheduling in a dynamic crowd platform". In: *proceedings of ICML Workshop: Machine Learning Meets Crowdsourcing, Atlanta, Georgia, USA.* 2013.

[67] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. "Learning From Crowds". In: *J. Mach. Learn. Res.* 11 (Aug. 2010), pp. 1297–1322. ISSN: 1532-4435. URL: `http://dl.acm.org/citation.cfm?id=1756006.1859894`.

[68] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker. "Protein structure prediction using Rosetta". In: *Methods in enzymology* 383 (2004), pp. 66–93. DOI: `10.1016/S0076-6879(04)83004-0`.

[69] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts". In: *ACM Transactions on Graphics (TOG)* 23.3 (2004), pp. 309–314. DOI: `10.1145/1015706.1015720`.

[70] C. Rupprecht, L. Peter, and N. Navab. "Image Segmentation in Twenty Questions". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). URL: `http://dx.doi.org/10.1109/cvpr.2015.7298952`.

[71] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. "LabelMe: a database and web-based tool for image annotation". In: *International journal of computer vision* 77.1-3 (2008), pp. 157–173. DOI: `10.1007/s11263-007-0090-8`.

[72] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu. "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". In: *Data Mining and Knowledge Discovery* 2.2 (), pp. 169–194. ISSN: 1573-756X. DOI: `10.1023/A:1009745219419`.

[73] A. Saxena, M. Sun, and A. Ng. "Make3d: Learning 3d scene structure from a single still image". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.5 (2009), pp. 824–840. DOI: `10.1109/TPAMI.2008.132`.

[74] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth. "Pattern Recognition: 35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings". In: ed. by J. Weickert, M. Hein, and B. Schiele. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Chap. Efficient Multi-cue Scene Segmentation, pp. 435–445. ISBN: 978-3-642-40602-7. DOI: `10.1007/978-3-642-40602-7_46`.

[75] V. S. Sheng, F. Provost, and P. G. Ipeirotis. "Get another label? improving data quality and data mining using multiple, noisy labelers". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2008, pp. 614–622. DOI: `10.1145/1401890.1401965`.

[76] R. Simpson, K. R. Page, and D. De Roure. "Zooniverse: Observing the World's Largest Citizen Science Platform". In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW '14 Companion. Seoul, Korea: International World Wide Web Conferences Steering Committee, 2014, pp. 1049–1054. ISBN: 978-1-4503-2745-9. DOI: `10.1145/2567948.2579215`.

[77] R. Smith. “The future of work is play: Global shifts suggest rise in productivity games”. In: *Games Innovation Conference (IGIC), 2011 IEEE International*. IEEE. 2011, pp. 40–43. DOI: 10.1109/IGIC.2011.6115127.

[78] N. Snavely, S. M. Seitz, and R. Szeliski. “Photo tourism: Exploring photo collections in 3D”. In: *SIGGRAPH Conference Proceedings*. New York, NY, USA: ACM Press, 2006, pp. 835–846. ISBN: 1-59593-364-6. DOI: 10.1145/1179352.1141964.

[79] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng. “Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks”. In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2008, pp. 254–263. URL: http://dl.acm.org/citation.cfm?id=1613715.1613751.

[80] A. Sorokin and D. Forsyth. “Utility data annotation with amazon mechanical turk”. In: *Urbana* 51.61 (2008), p. 820.

[81] S. Spielmann, V. Helzle, and R. Nair. “On-set Depth Capturing for VFX Productions Using Time of Flight”. In: *ACM SIGGRAPH 2013 Talks*. SIGGRAPH ’13. Anaheim, California: ACM, 2013, 13:1–13:1. ISBN: 978-1-4503-2344-4. DOI: 10.1145/2504459.2504475.

[82] H. Su, J. Deng, and L. Fei-Fei. *Crowdsourcing Annotations for Visual Object Detection*. 2012. URL: http://www.aaai.org/ocs/index.php/WS/AAAIW12/paper/view/5350.

[83] D. Sun, S. Roth, and M. J. Black. *Secrets of Optical Flow Estimation and Their Principles*. 2010. DOI: 10.1109/CVPR.2010.5539939.

[84] J. Surowiecki. *The wisdom of crowds*. Anchor, 2005. ISBN: 978-0-385-50386-0. URL: https://www.worldcat.org/oclc/61254310.

[85] B. Unterberg. “Crowdsourcing (Jeff Howe)”. In: *Social-Media-Handbuch : Theorien, Methoden, Modelle* (2010), pp. 121–135. URL: https://www.econbiz.de/Record/crowdsourcing-jeff-howe-unterberg-bastian/10008771533.

[86] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. “Multiple hypothesis video segmentation from superpixel flows”. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 268–281. URL: http://dl.acm.org/citation.cfm?id=1888150.1888172.

[87] E. Versi. “"Gold standard" is an appropriate term.” In: *BMJ* 305.6846 (1992), pp. 187–187. ISSN: 0959-8138. DOI: 10.1136/bmj.305.6846.187-b. eprint: http://www.bmj.com/content/305/6846/187.1.full.pdf. URL: http://www.bmj.com/content/305/6846/187.1.

[88] L. Von Ahn. “Human computation”. In: *Design Automation Conference, 2009. DAC’09. 46th ACM/IEEE*. IEEE. 2009, pp. 418–419. URL: http://dx.doi.org/10.1145/1629911.1630023.

[89] L. Von Ahn and L. Dabbish. “Labeling images with a computer game”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2004, pp. 319–326. URL: http://dx.doi.org/10.1145/985692.985733.

[90] L. Von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum. “Improving accessibility of the web with a computer game”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM. 2006, pp. 79–82. URL: http://dx.doi.org/10.1145/1124772.1124785.

[91] L. Von Ahn, R. Liu, and M. Blum. “Peekaboom: a game for locating objects in images”. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM. 2006, pp. 55–64. URL: http://dx.doi.org/10.1145/1124772.1124782.

[92] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. “recaptcha: Human-based character recognition via web security measures”. In: *Science* 321.5895 (2008), pp. 1465–1468. URL: http://dx.doi.org/10.1126/science.1160379.

[93] C. Vondrick, D. Patterson, and D. Ramanan. “Efficiently scaling up crowdsourced video annotation”. In: *International Journal of Computer Vision* 101.1 (2013), pp. 184–204. DOI: 10.1007/s11263-012-0564-1.

[94] C. Wah, S. Branson, P. Perona, and S. Belongie. “Multiclass recognition and part localization with humans in the loop”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. 2011, pp. 2524–2531. DOI: 10.1109/ICCV.2011.6126539.

[95] C. Wah. “Crowdsourcing and its applications in computer vision”. In: *University of California, San Diego* (2006).

[96] B. Ward, S. B. Kang, and E. P. Bennett. “Depth Director: A System for Adding Depth to Movies”. In: *IEEE Computer Graphics and Applications* 31.1 (2011), pp. 36–48. ISSN: 0272-1716. DOI: http://doi.ieeecomputersociety.org/10.1109/MCG.2010.103.

[97] P. Welinder, S. Branson, P. Perona, and S. J. Belongie. “The multidimensional wisdom of crowds”. In: *Advances in neural information processing systems*. 2010, pp. 2424–2432.

[98] A. J. Westphal, A Allbrink, C Allen, S Bajt, R Bastien, H Bechtel, P Bleuet, J Borg, S Bowker, F Brenker, et al. “Non-destructive search for interstellar dust using synchrotron microprobes”. In: *X-RAY OPTICS AND MICROANALYSIS: Proceedings of the 20th International Congress*. Vol. 1221. 1. AIP Publishing. 2010, pp. 131–138. DOI: 10.1063/1.3399239.

[99] D. Wightman. “Crowdsourcing human-based computation”. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. ACM. 2010, pp. 551–560. DOI: 10.1145/1868914.1868976.

[100] C. Wu. *VisualSFM: A Visual Structure from Motion System.* 2011.

[101] L. Xu, J. Jia, and Y. Matsushita. “Motion detail preserving optical flow estimation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.9 (2012), pp. 1744–1757. DOI: `10.1109/TPAMI.2011.236`.