Optimal Separable Interpolation of Color Images with Bayer array format

Hanno Scharr

Interdisciplinary Center for Scientific Computing
Ruprecht-Karls-Universität Heidelberg
Im Neuenheimer Feld 368
69120 Heidelberg, Germany

Abstract. In this paper new separable interpolation schemes for the channels of a single chip color camera are presented. The pixel grid of the camera is called "Bayer" pattern (compare [6] and fig. 1). As different colors are on different grids interpolation of all channels onto one common grid has to be performed. An analysis of the sampling scheme of this camera chip restricts the interpolation methods to filters on inter pixel positions in the middle of the processed color channel grids. Additionally this analysis defines the optimal resolution of the interpolated data which is of high interest concerning data compression. For this kind of interpolation optimal filters are presented that eliminate any possibility of phase errors. Thereby pixel values are supposed to be point samples of an intensity signal convolved by the average pixel area of the chip.

Keywords: Optimal interpolation, data reduction, Bayer array format, phase error, sampling, registration

1 Introduction

If we use single chip color cameras for scientific applications we are confronted with the problem to store all channels on the same pixel grid. Further if phase sensitive algorithms as e.g. motion estimation shall be applied we need interpolation schemes with minimal phase error. For the example of camera chips with a pixel pattern called "Bayer" pattern with 2×2 unit cell (see fig. 1 or [6]) optimal interpolation schemes that eliminate any possibility of phase errors are presented. As these interpolation filters shall be implemented in field programmable gate array (FPGA) hardware the support of the filters has to be rather small.

…	…	…	…	…	…
…	G	R	G	R	…
…	B	G	B	G	…
…	G	R	G	R	…
…	B	G	B	G	…
…	…	…	…	…	…

Fig. 1. Pixel sequence of a camera chip with "Bayer" pattern. R: red, G: green, B: blue.

1.1 Paper organization

In an analysis of the Bayer array format (sec. 2) we separate the interpolation task into two independent steps. First the interpolation of grid data on inter pixel positions is done and second data is reduced by subsampling. Thus section 3 addresses optimal interpolation, followed by a discussion on aspects of an eventually trivial subsampling in section 4. A summary concludes the paper (sec. 5).

2 Analysis of the interpolation task

All pixel of the camera chip are supposed to be identical except for their color selectivity produced by a Bayer color mosaic filter. If we ignore this selectivity we get a usual grau value camera chip whose well known properties (compare e.g. citeBJ97) shall be analysed briefly before we study the Bayer pattern.

Pixel values we get by this chip are samples of a continuous intensity signal. This continuous signal has been convolved by the camera optics and by the light sensitivity of the real average pixel. The succeeding sampling at the central point of each pixel can be interpreted as a multiplication by a δ-comb. This is a convolution of the signal in the wave number domain (Fourier domain) by the transfer function of a δ-comb which also is a δ-comb. This convolution is the reason for the well known aliasing errors by sampling. The maximum wave number is called ”Nykvist” wave number. A resolution given by this pixel grid is called ”full” resolution in the remainder of this paper.

The three channels of the Bayer pattern can be divided into two types of grids. First the grids of the red or blue pixels. Data supplied on these grids are identical to data acquired on full resolution as described above, succeeded by separable subsampling by considering every second pixel in each coordinate direction. The grid of the green pixels corresponds to a subsampling as it is usual for a $\sqrt{2}$-pyramid. These observations already lead to the conclusion that the camera optics are not supposed to yield perfectly sharp images. Otherwise the subsampling without presmoothing of the data produces undesirable aliasing errors. As the coarsest grid defines the maximum sharpness of the system, we may suppose that the function we want to interpolate is restricted to a wave number interval [-0.5, 0.5] (units normalized to Nykvist wave number). The maximum resolution we can achieve therefore is the resolution of the coarsest grid. Higher resolutions as the one of the green image lead to redundant data sets.[1] Thus the interpolation is reduced to generation of values at uniform pixel positions on a grid like the blue or green one in all three color channels. This usually is called ”registration”.

Further this translation of the sampling is desired to have the least possible phase errors. Phase errors are produced by the imaginary part of the transfer function of a filter. As the imaginary part of symmetric filters in equal to zero they are the best choice here. By inspecting the Bayer pattern we can observe, that interpolation on any integer pixel position at full resolution can be done by symmetric filters, whereas inter pixel positions determine nonsymmetric filters.

Such a symmetric interpolation reconstructs the ideally processable part of the intensity signal convoluted by the point spread function of the optics and the light sensitivity of an average pixel. Interpolations suitable for this kind of reconstruction will be presented in the next section.

We can also construct signals as they would have been created by pixels that cover the whole chip area in *every* color channel. Towards this end a further operation is necessary. It is discussed in section 4.

[1] If the spectrum of the image projected on the chip (degraded by the camera optics) was not restricted to $[\ -0.5, 0.5\ ]^2$, we had to smooth the green channel before subsampling to the resolution of the other channels. But then severe aliasing errors occur in these coarse channels, which is not acceptable for scientific application.

3 Interpolation

In this section three interpolations will be briefly presented, one of them has been optimized for the zero padding approach. As shown in [4] interpolation filters of this type are separable. Consequently all derivations will be in one dimension.
Interpolation methods for signals with equidistant sampling points are well known in numerics. They are based on e.g. polynomial, rational or trigonometric functions (compare e.g. [7]).
Trigonometric basis functions usually are used for periodic signals. In our case only few points of the total signal shall be processed at once. That is why this kind of interpolation seems not to be well suited here.
If we use rational functions as a basis, the problem of so called "unreachable" points occurs. In other words reliable interpolation results are not guaranteed in all cases, even on equidistant sample points.
The remaining polynomial basis functions are unproblematic and commonly used in image processing. The two most common interpolation schemes will be presented next as we want to compare them with the optimized results. They will be described in the following two sections. For further studies we refer to [4]. A more complete overview on B-splines can be found in [1].
For the optimization of interpolation filters the zero padding concept of interpolation in Fourier domain will be reviewed subsequently.

3.1 Polynomial interpolation

The difference between polynomial and B-spline interpolation presented in the next section is the kind of junction conditions required. Smoothness plays the most important role for B-splines (see below). Whereas for polynomial interpolation we are restricted to the assumption, that the interpolated continuous function exactly hits the signal sampling points. Let these points be (x_i, y_i) where $i \in \{0, \ldots, N-1\}$. For $N = 3$ we get the polynomial

$$P_3(x) = (\frac{1}{2}\,y_0 - y_1 + \frac{1}{2}\,y_3)\,x^2 + (-\frac{3}{2}\,y_0 + 2\,y_1 - \frac{1}{2}\,y_3)\,x + y_0\,, \tag{1}$$

for $N = 4$

$$\begin{aligned} P_4(x) = {} & (\tfrac{1}{6}\,y_0 + \tfrac{1}{2}\,y_1 - \tfrac{1}{2}\,y_2 + \tfrac{1}{6}\,y_4)\,x^3 + (y_0 - \tfrac{5}{2}\,y_1 + 2\,y_2 - \tfrac{1}{2}\,y_4)\,x^2 \\ & + (-\tfrac{11}{6}\,y_0 + 3\,y_1 - \tfrac{3}{2}\,y_2 + \tfrac{1}{3}\,y_4)\,x + y_0\,. \end{aligned} \tag{2}$$

If we set in a fixed position for x we get an interpolation result of the form

$$\sum_{i}^{N-1} m_i\,y_i\,, \tag{3}$$

where m_i can be interpreted as filter coefficients[2] f_i where $f_i = m_{N-1-i}$. If we use nearest neighbors for the construction of a polynomial and piecewise compound the interpolation function, we get a continuous function. This function is differentiable in the inner of piece intervals but generally nondifferentiable at interval boundaries. If the interpolation function additionally has to be differentiable at interval boundaries, we arrive at B-splines described in the next section.

[2] In the definition of signal convolution, the convolving kernel is mirrored.

3.2 B-spline interpolation

In order to construct B-splines two assumptions have to be met. First, the continuous interpolated signal has to be differentiable $N-2$ times in every point, especially at piece interval boundaries. Second, constant functions have to be interpolated correctly. This is sufficient to be able to represent B-splines by piecewise defined basis functions $B_i(x)$ with the property

$$B_n(s) = B_0(s-n)$$

(translation invariant). The correct interpolation of constant functions is guaranteed by

$$\sum_n B_n(x) = 1\,.$$

Interpolation is done by multiplying basis function B_n (corresponding to position x_n) with the signal value y_n. The sum of these weighted basis functions is the continuous signal.
The basis function $B_0(x)$ with a support of length $N = 3$ is

$$B_0(x) = \begin{cases} \frac{1}{2}(x+\frac{3}{2})^2 & \text{for } -1.5 \le x < -0.5 \\ \frac{3}{4} - x^2 & \text{for } -0.5 \le x < 0.5 \\ \frac{1}{2}(x-\frac{3}{2})^2 & \text{for } 0.5 \le x < 1.5 \\ 0 & \text{else.} \end{cases} \tag{4}$$

For $N = 4$ we get

$$B_0(x) = \begin{cases} \frac{1}{6}(x+2)^3 & \text{for } -2 \le x < -1 \\ -\frac{1}{2}(x+2)^3 + 2(x+2)^2 - 2(x+2) - \frac{2}{3} & \text{for } -1 \le x < 0 \\ -\frac{1}{2}(2-x)^3 + 2(2-x)^2 - 2(2-x) - \frac{2}{3} & \text{for } 0 \le x < 1 \\ \frac{1}{6}(2-x)^3 & \text{for } 1 \le x < 2 \\ 0 & \text{else.} \end{cases} \tag{5}$$

Values of basis functions $B_n(x)$ which are not zero at a sampling position x can be interpreted as filter coefficients in the same way as above.

3.3 Interpolation by zero padding

For this interpolation method one supposes that signal frequencies are not changed in the first Brillouin zone, i.e. the spectrum as we usually compute it by Fourier transformation. Only frequencies outside this zone are set to zero. By doing so we get a continuous function which can be resampled. Discrete interpolation therefore can be done in three steps:

1. Reconstruction of the continuous, periodic image by multiplication of the first Brillouin zone with one, all higher zones by zero.
2. Sampling of the image on a new pixel grid. For fixed convolution masks the pixel distance can not be changed, only translations of the pixel grid can be performed. In Fourier domain this corresponds with convolutions by δ-combs $\hat{\delta}_\alpha(\tilde{k})$ (see below) with an arbitrary but fixed phase α and fixed distances between the single δ-peaks of the comb.
3. Addition of the new grids, if higher resolutions are desired.

The original grid can be described by $\delta_0(x)$ where

$$\delta_0(x) = \sum_{r=-\infty}^{\infty} \delta(x - r\triangle x)\,.$$

The new grid then is given by $\delta_\alpha(x)$ where

$$\delta_\alpha(x) = \sum_{r=-\infty}^{\infty} \delta(x - (r+\alpha)\triangle x)\,.$$

As the continuous image in spatial domain is multiplied by the pixel grid, in the Fourier domain its spectrum is convolved by

$$\hat{\delta}_\alpha(\tilde{k}) = \sum_{r=-\infty}^{\infty} \delta(\tilde{k} - r\triangle\tilde{k})\mathrm{e}^{-\mathrm{i}\alpha\frac{\tilde{k}}{\triangle\tilde{k}}} \tag{6}$$

$$= \sum_{r=-\infty}^{\infty} \delta(\tilde{k} - r\triangle\tilde{k})\mathrm{e}^{-\mathrm{i}\alpha r}$$

where $\triangle\tilde{k} = 1/\triangle x$. The replications of the first Brillouin zone are multiplied by a phase $\mathrm{e}^{-\mathrm{i}\alpha r}$ where r is the distance between the particular mid points of the zones and the origin. Addition of the grids in spatial domain corresponds with an addition of the spectra. In the first Brillouin zone all spectra are identical. In the translated Brillouin zones spectra are shifted by a phase. For an interpolation on inter pixel positions for double resolution this phase factor is -1. Thus addition of the grids causes complete erasement of the data on every second zone. Double resolution in spatial domain corresponds with an enlargement of the first Brillouin zone to double diameter. The new area thus covers only erased frequencies.

3.4 Optimal interpolation filters

In this optimization an ansatz function is compared to a reference function by a weighted norm. The reference function is constructed by the zero padding assumption.

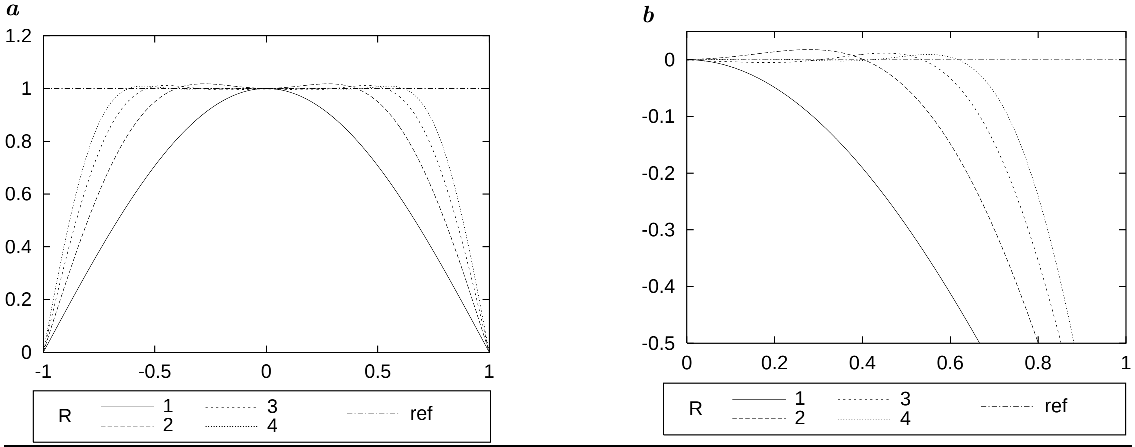
Reference function As the optimization is only performed in the first Brillouin zone, the reference function $\hat{I}_{\mathrm{ref}}$ is identical 1.

Ansatz function The transfer function of a discrete filter $[p_0,\ p_1,\ \dots,\ p_R]$ with $R+1$ coefficients is

$$\hat{I}_{r_0}(\tilde{k}, \boldsymbol{p}) = \sum_{r=0}^{R} p_r \exp(-\mathrm{i}\pi(r - r_0)\tilde{k}) = \exp(-\mathrm{i}\pi r_0\tilde{k})\sum_{r=0}^{R} p_r \exp(-\mathrm{i}\pi r\tilde{k}) \tag{7}$$

The value computed be the mask is stored at a grid point corresponding to r_0. E.g. for a common symmetric interpolation mask of length 4 we set $r_0 = 1.5$. By this concept phase shifts in higher Brillouin zones are fixed to the desired value. (compare eq. 6). The ansatz function in eq. 7 yields no exact interpolation for a constant signal, i.e. for $\tilde{k} = 0$. Therefore we add the constraint

$$p_0 = 1 - \sum_{r=1}^{R-1} p_i\,. \tag{8}$$



R	filter	r_0	error
2	[128, 128]/256	0.5	8.63e-02
4	[-21.91, 149.91, 149.91, -21.91]/256	1.5	1.77e-02
6	[7.02, -34.20, 155.18, 155.18, -34.20, 7.02]/256	2.5	5.77e-03
8	[-2.87, 13.35, -40.40, 157.93, 157.93, -40.40, 13.35, -2.87]/256	3.5	2.39e-03
2	[128, 128]/256	0.5	8.63e-02
4	[-22, 150, 150, -22]/256	1.5	1.77e0-2
6	[7, -34, 155, 155, -34, 7]/256	2.5	5.81e-03
8	[-3, 14, -42, 160, 156, -39, 13, -3]/256	3.5	2.63e-03

Table 1. Real part (**a**) and difference to the reference (**b**) of the transfer functions of the interpolation filters optimized by eq. 7 in floating point precision. The imaginary part is constant 0. Filter coefficients in floating point and integer precision are given in the table. Please note that optimal integer results are not always symmetric. In this case the mirrored filter yields the same error.

Error functional As the ansatz function generally is complex, real and imaginary part are calculated. The absolute error $d(\tilde{k}, \boldsymbol{p})$ then becomes

$$\begin{aligned} d(\tilde{k}, \boldsymbol{p}) &= \sqrt{(\Re(I_{r_0}(\tilde{k}, \boldsymbol{p})) - \Re(I_{\mathrm{ref}}(\tilde{k})))^2 + (\Im(I_{r_0}(\tilde{k}, \boldsymbol{p})) - \Im(I_{\mathrm{ref}}(\tilde{k})))^2} \\ &= \sqrt{(\Re(I_{r_0}(\tilde{k}, \boldsymbol{p})) - 1)^2 + \Im(I_{r_0}(\tilde{k}, \boldsymbol{p}))^2}\,. \end{aligned}$$

This error functional is nonlinear.

Results The filter coefficients presented in table 1 are optimized in L_2 norm with a weighting function $\cos(\pi/2\,\tilde{k})^4$. Table 1 lists filters that store their values in the middle between the old grid points (i.e., $r_0 = (N-1)/2$). They can be used to double resolution.

3.5 Comparison of the interpolations

The following table shows error values of the particular interpolation methods, here N is the number of pixels used (support of the filter).

...	...	...	...	...	...
...	1	R	1	R	...
...	2		2		...
...	1	R	1	R	...
...	2		2		...
...	...	...	...	...	...

Fig. 2. Interpolation steps for the red grid. The results of row convolutions are stored at position "1", for column convolutions at "2".

N	B-Spline	Polynom	optimiert
2	0.089	0.089	–
3	0.11	0.056	0.038
4	0.089	0.030	0.018
6	–	0.014	0.0058
8	–	0.12	0.0024

The filter for $N = 2$ is [0.5,0.5], i.e. linear interpolation. The nonoptimal filters and polynomials for $N = 6$ and $N = 8$ shall not be given here, as they (as the optimal ones, too) result in large total interpolation schemes on Bayer pattern cameras. For the polynomial interpolation we can observe the well known instable behaviour for large N by the inclination of the error for $N = 8$.

For filters up to length $N = 4$ the optimization reduces the error about a factor 1.5 to 1.7 with respect to the corresponding polynomial interpolation. Compared to linear interpolation ($N = 2$) errors are reduced about a factor 5. If we use larger masks, the error difference between the best common interpolation an the best optimal one is about a factor 5.6 . Compared to linear interpolation we get a factor 35.

3.6 Calculation of registered channels

We will now see, how a registration to the grid of the blue (or red) pixel can be done by interpolation. As the blue pixels are already on the right positions, this channel needs no interpolation. The red channel can be interpolated seperably as usual by first applying interpolation kernels to rows (storage at "1" in fig. 2) and then doing so for columns (storage at "2" in fig. 2).

The green channel is processed similar, but instead of rows and columns for the diagonals. If we suppose that the camera optics produces a signal that can be represented on the coarser grids of the other channels, we could interpolate the green channel by smaller convolution kernels. Because then the wave number spectrum of the green channel does not use its total first Brillouin zone, but only the interval $[-\sqrt{2}/2, \sqrt{2}/2]$ (separable along diagonals!). Therefore the huge errors at high wave numbers are suppressed automatically.

Alternatively a registration onto green pixel can be done. To do so we have to choose a subgrid "a" or "b" as shown in fig. 3. Choosing subgrid "a" red pixel are interpolated only in horizontal direction, blue ones only in vertical direction. For subgrid "b" it is the other way round. There are two advantages in this approach. On the one hand all interpolations are done in the same way, thus interpolation errors are equivalent. On the other hand only one dimensional interpolations have to be done, which yields increasing performance.

By the given interpolation schemes we produce color images on subgrids with a pixel size (i.e. the real size of the chip pixels) corresponding to double resolution. Thus we

...	...	...	...	...	...
...	a		a		...
...		b		b	...
...	a		a		...
...		b		b	...
...	...	...	...	...	...

Fig. 3. Green subgrid a and b.

red

1/16		1/16
3/8	•	3/8
1/16		1/16

green

		1/32		
	1/8		1/8	
1/32		3/8 •		1/32
	1/8		1/8	
		1/32		

blue

1/16		3/8		1/16
		•		
1/16		3/8		1/16

Fig. 4. Total filter for linear interpolation and subsampling with additional separable smoothing by [1,2,1]/4. Values are stored at the pixel with a •.

might want to double the virtual pixel size for example to simulate a 3-chip-camera. To do so, a modified approach is needed. We therefore first study subsampling and then present the resulting approach.

4 Subsampling

The signal constructed so far corresponds with a sampling at the full resolution of the chip and a subsequent subsampling. In order to preserve the local light intensity the signal has to be convolved separably by [1,2,1]/4 before subsampling, i.e. simulation of pixels with doubled diameter. As aliasing errors, which are usually suppressed be this presmoothing, are already in the signal further smoothing is not recommended. [3] A suitable presmoothing therefore has to and can only be done by the camera optics. By this knowledge we now can construct a filter that interpolates to full resolution, smoothes by [1,2,1]/4 to simulate larger pixels and subsamples in one step. For the most common cases of linear and interpolation by kernels of size 4 the resulting total filters are given in figures 4 and 5 for registration on the green subgrid "a".

5 Summary

In this paper optimal interpolation filters for inter pixel positions in the middle between original grid points have been presented. In an application they are used to register color channels of a single chip camera with Bayer pattern. This registration is free of phase errors. Simulation of a camera with larger pixels can be done by applying a smoothing kernel [1,2,1]/4 before subsampling. Total filters for this task are presented as well.

Acknowledgement. Parts of this work have been funded by the DFG research unit "image sequence processing for the study of dynamic processes" (Ja395/6).

[3] As mentioned above (see sec. 2) the spectrum of the image projected on the chip is supposed to be restricted to [-0.5, 0.5 $]^2$. If this is not entirely the case, we have to smooth the green channel before subsampling. This is sufficiently done by [1,2,1]/4, even in cases where scientific exact data is not required.

red

$\frac{1}{8}p^2$		$\frac{1}{16}p + \frac{1}{8}pq$		$\frac{1}{16}p + \frac{1}{8}pq$		$\frac{1}{8}p^2$
$\frac{1}{8}(p^2 + pq)$		$\frac{1}{16}(p + q) + \frac{1}{8}(pq + q^2)$		$\frac{1}{16}(p + q) + \frac{1}{8}(pq + q^2)$		$\frac{1}{8}(p^2 + pq)$
$\frac{1}{4}(pq + p)$		$\frac{1}{8} + \frac{3}{8}q + \frac{1}{4}q^2$	●	$\frac{1}{8} + \frac{3}{8}q + \frac{1}{4}q^2$		$\frac{1}{4}(pq + p)$
$\frac{1}{8}(p^2 + pq)$		$\frac{1}{16}(p + q) + \frac{1}{8}(pq + q^2)$		$\frac{1}{16}(p + q) + \frac{1}{8}(pq + q^2)$		$\frac{1}{8}(p^2 + pq)$
$\frac{1}{8}p^2$		$\frac{1}{16}p + \frac{1}{8}pq$		$\frac{1}{16}p + \frac{1}{8}pq$		$\frac{1}{8}p^2$

green

				$\frac{1}{8}p^2$				
			$\frac{1}{8}p^2 + \frac{1}{8}pq$		$\frac{1}{8}p^2 + \frac{1}{8}pq$			
		$\frac{1}{4}pq$		$\frac{1}{8}(p + q)^2$		$\frac{1}{4}pq$		
	$\frac{1}{8}p^2 + \frac{1}{8}pq$		$\frac{1}{16} + \frac{1}{4}q^2 + \frac{1}{4}pq$		$\frac{1}{16} + \frac{1}{4}q^2 + \frac{1}{4}pq$		$\frac{1}{8}p^2 + \frac{1}{8}pq$	
$\frac{1}{8}p^2$		$\frac{1}{8}(p + q)^2$		$\frac{1}{4} + \frac{1}{2}q^2$ ●		$\frac{1}{8}(p + q)^2$		$\frac{1}{8}p^2$
	$\frac{1}{8}p^2 + \frac{1}{8}pq$		$\frac{1}{16} + \frac{1}{4}q^2 + \frac{1}{4}pq$		$\frac{1}{16} + \frac{1}{4}q^2 + \frac{1}{4}pq$		$\frac{1}{8}p^2 + \frac{1}{8}pq$	
		$\frac{1}{4}pq$		$\frac{1}{8}(p + q)^2$		$\frac{1}{4}pq$		
			$\frac{1}{8}p^2 + \frac{1}{8}pq$		$\frac{1}{8}p^2 + \frac{1}{8}pq$			
				$\frac{1}{8}p^2$				

blue

$\frac{1}{8}p^2$		$\frac{1}{8}(p^2 + pq)$		$\frac{1}{4}(pq + p)$		$\frac{1}{8}(p^2 + pq)$		$\frac{1}{8}p^2$
$\frac{1}{16}p + \frac{1}{8}pq$		$\frac{1}{16}(p + q) + \frac{1}{8}(pq + q^2)$		$\frac{1}{8} + \frac{3}{8}q + \frac{1}{4}q^2$		$\frac{1}{16}(p + q) + \frac{1}{8}(q^2 + pq)$		$\frac{1}{16}p + \frac{1}{8}pq$
				●				
$\frac{1}{16}p + \frac{1}{8}pq$		$\frac{1}{16}(p + q) + \frac{1}{8}(pq + q^2)$		$\frac{1}{8} + \frac{3}{8}q + \frac{1}{4}q^2$		$\frac{1}{16}(p + q) + \frac{1}{8}(q^2 + pq)$		$\frac{1}{16}p + \frac{1}{8}pq$
$\frac{1}{8}p^2$		$\frac{1}{8}(p^2 + pq)$		$\frac{1}{4}(pq + p)$		$\frac{1}{8}(p^2 + pq)$		$\frac{1}{8}p^2$

Fig. 5. Total filter for interpolation by [p,q,q,p] and subsampling with additional separable smoothing by [1,2,1]/4. Values are stored at the pixel with a ●.

References

1. Blake, A. und Isard, M. *Active Contours*, Springer, 1998.
2. Jähne, B, *Digital Image Processing*, Springer, 4 edition, 1997.
3. B. Jähne, H. Scharr, S. Körkel, *Principles of filter design*, In B. Jähne, H. Haußecker, P. Geißler (Eds.), Handbook on Computer Vision and Applications, Vol. 2: Signal Processing and Pattern Recognition, Academic Press, San Diego, 125–152, 1999.
4. Jähne, B., *Interpolation*, In Jähne, B., Haußecker, H., und Geißler, P. (Eds.), Handbook of Computer Vision and Applications. Academic Press. Applications, Vol. 2: Signal Processing and Pattern Recognition, Academic Press, San Diego, 1999.
5. H. Scharr, *Optimale Operatoren in der Digitalen Bildverarbeitung*, Dissertation, Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany, 2000.
6. Seitz, P., *Solid-State Image Sensing*, In Jähne, B., Haußecker, H., und Geißler, P. (Eds.), Handbook of Computer Vision and Applications. Academic Press. Applications, Vol. 1: Sensors and Imaging, Academic Press, San Diego, 1999.
7. Stoer, J. (1994), *Numerische Mathematik 1*, Springer, 7 Ausgabe, 1994.