Fast Implementations of Temporal Color Image Filtering

Florian Vogt[1], Dietrich Paulus[1], and Christoph H. Schick[2]

[1] Lehrstuhl für Mustererkennung, Universität Erlangen-Nürnberg
Martensstr. 3, 91058 Erlangen, Germany
{vogt,paulus}@informatik.uni-erlangen.de,
http://www5.informatik.uni-erlangen.de
[2] Chirurgische Universitätsklinik
Krankenhausstr. 12, 91058 Erlangen, Germany
schick@chirurgie-erlangen.de

Abstract The quality of the displayed image during minimal-invasive operations (endoscopy) may be low due to highlights and water vapor in front of the camera lens. In some cases, image processing like temporal color vector median filtering may enhance the visual quality of the images [8]. In this contribution we present a new technique of ordering, filtering and re-ordering of color image pixels that implements temporal filtering by using fast implementations of spatial filters. As an example for a fast spatial filtering library, we use the Intel Image Processing Library (IPL). A temporal color median filter was implemented with the new technique. Computing time and image enhancement were evaluated for this filter.
Keywords: Color Image Processing, Time Filtering, Medical Imaging

1 Introduction

Endoscopic surgery is carried out by the physicians without direct visual contact to the operation area. Instead, the video displayed on the monitor is used for visual feedback. The devices used for display have to pass several tests required to install machines in the operating room. This may be one reason why these devices often are one generation behind with respect to the latest equipment. In the near future, computer support is expected to be available in the operating room as well as in any other clinical process. Much research is done for so-called computer aided surgery that will help education as well as the surgeons itself. Interesting results in this field were presented at the 5^{th} Annual Conference of the International Society for Computer Aided Surgery (ISCAS) 2001 in Berlin which was part of the Computer Assisted Radiology and Surgery (CARS) conference [4].

During surgery, image quality may be low due to highlights and water vapor (fog) in front of the lens. Simple but powerful image processing may in some cases enhance

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) under grant SFB 603/TP B6. Only the authors are responsible for the content.

the visual quality of the endoscopic images displayed on the monitor. Image enhancement and highlight detection for medical imaging has been performed in [5]. In [8], we showed that temporal color median filtering of endoscopic images is well suited to enhance image quality whereas spatial filtering is not.

A prerequisite for temporal filtering of image sequences is a slow change of the camera pose (i.e. of the translation and orientation of the camera) and little motion of the scene. Otherwise, artefacts may be generated by the temporal filtering, e.g. replicated image structures like edges, and with it the sharpness of the image is reduced. Real-time performance is required when such operations will be introduced for image display in the surgery room.

In the last decades, filtering operations were mostly implemented for spatial and not for temporal filtering. Very fast implementations of spatial filtering operations exist, e.g. the Intel Image Processing Library (IPL) [1] which is very fast on processors with Intel's MMX instruction set because the library is capable to use them. In this contribution we examine how those fast implementations of spatial filtering operations can be used for (fast) temporal filtering of images. As an example for a fast spatial filtering library, the IPL is used in the following.

2 Real-time Color Image Operations

State-of-the-art personal computers still lack the required computing power for performing more than trivial operations on color images in real-time, as long as general purpose instructions are used. Such instructions are usually generated by compilers of high-level programming languages.

Intel's MMX instruction set can greatly increase computation times because up to 8 bytes can be processed in parallel; the support of high-level languages, however, is still small. The IPL [1] provides a high-level language interface as it provides a library encapsulating the back-end assembly language code. The OpenCV library (open computer vision library) [2] is — at least at the moment — based on the IPL and provides additional, more complex, image processing functionality. In the following, we consider low-level color image filtering functions that are implemented in the IPL and which are potentially useful for our application. The following enumeration shortly outlines the provided *color image* filtering functions [1]:

Linear filters
- Prewitt, size 3×3, vertical and horizontal
- Sobel, size 3×3, vertical and horizontal
- Laplace, size 3×3,
- Gauss, size 3×3,

Non-linear, rank-ordering filters
- Minimum filter, arbitrary window size
- Maximum filter, arbitrary window size
- Median filter, arbitrary window size

Morphological operations
- Erode, arbitrary number of iterations within 8-neighborhood

– Dilate, arbitrary number of iterations within 8-neighborhood
– Open, arbitrary number of iterations within 8-neighborhood
– Close, arbitrary number of iterations within 8-neighborhood

Threshold threshold operation for each of $1, \ldots, 4$ channels of the image

All IPL color image filters operate on single color channels. We restrict our further investigations to RGB color images, as the IPL provides functions to convert images between commonly used color spaces. For linear filters, there is no difference between single channel filtering and color pixel filtering. For the rank-ordering filters one has to define an ordering for color pixel vectors, e.g. the vectors could be ordered according to their Euclidean norm (this is further referred as color vector filtering as proposed e.g. in [7]). The color median filter of the IPL operates on the single color channels, i.e. the R, G and B channel are separately median filtered. This means, the IPL color median filter is not a color *vector* median filter.

We filtered a randomly generated color image and a (real) endoscopic image with a color vector median filter and with the IPL color median filter. The results are shown in Figure 2 and Figure 6. The results of the random image look very different, whereas the results of of the endoscopic image look surprisingly similar. A more formal explanation of this impression follows.

Figure 1 shows the absolute difference between the resulting images. Again, the large difference of the filter results of the random image in contrast to the very small difference of the filter results of the endoscopic image can be seen very clearly. Let a color image $\boldsymbol{f}$ with N rows and M columns be defined as

$$\boldsymbol{f} = \left[\boldsymbol{f}_{ij}\right]_{i=0,\ldots,N-1,j=0,\ldots,M-1} \tag{1}$$

where

$$\boldsymbol{f}_{ij} = (r_{ij}, g_{ij}, b_{ij})^{\mathrm{T}} \ . \tag{2}$$

We define the mean pixel difference $\overline{m_d}(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)})$, of two images $\boldsymbol{f}^{(1)}$ and $\boldsymbol{f}^{(2)}$ as

$$\overline{m_d}(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}) = \frac{1}{3MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (|r_{ij}^{(1)} - r_{ij}^{(2)}| + |g_{ij}^{(1)} - g_{ij}^{(2)}| + |b_{ij}^{(1)} - b_{ij}^{(2)}|). \tag{3}$$

The mean pixel difference between the two filter results of the random image was 57.38 pixel, the mean pixel difference between the two filter results of the endoscopic image was 1.48 pixel. This reflects very well what can be seen when regarding the filter results and the difference images: there is large difference between the color vector median filter and the IPL color median filter for random images, but the difference between the two filters is very small (regarding the mean difference) for the endoscopic image.

Table 1 shows the execution times of the listed IPL filters. In our case, real-time image processing means 25 images per second, i.e. 40 milliseconds per image. Only the morphological operations and the median filter for sizes > 5 could not be applied in real-time. But as computers get faster, it will also be possible to apply these filters in real-time. The large execution times for large median filters sizes lead to the assumption

Table 1. Execution times measured on an RGB color image of size 720×576 with a computer with a 800 MHz Pentium III processor. Mean values of 100 subsequent filtering operations are shown $\pm$ standard deviation, the time values are given in milliseconds. For the morphological operations (erode, dilate, open, close), the size value defines the number of iterations, for the rank-ordering filters (min, max, median), the size value defines the side length of the window, and for the threshold operation, the size value was used as threshold.

Filter	Size 3	Size 5	Size 7	Size 9	Size 11
Prewitt 3×3 H	5 ± 0.045				
Prewitt 3×3 V	5 ± 0.043				
Sobel 3×3 H	6 ± 1.503				
Sobel 3×3 V	5 ± 0.025				
Dilate	47 ± 0.187	53 ± 0.742	68 ± 0.903	71 ± 1.180	78 ± 2
Erode	47 ± 4	53 ± 0.201	67 ± 0.783	71 ± 1.227	77 ± 2
Open	59 ± 0.413	72 ± 0.856	99 ± 0.954	105 ± 1.118	116 ± 0.761
Close	59 ± 0.518	72 ± 1.038	99 ± 1.077	105 ± 1.060	116 ± 1.448
Min	12 ± 2	14 ± 0.069	18 ± 0.820	20 ± 0.086	23 ± 0.093
Max	12 ± 0.063	14 ± 0.077	18 ± 0.341	20 ± 0.154	23 ± 0.093
Median	10 ± 0.068	39 ± 0.441	338 ± 4	392 ± 1.277	451 ± 2
Threshold	8 ± 3	8 ± 0.141	8 ± 2	8 ± 1.551	8 ± 0.889

that the implementation is not scalable very well. There exists an implementation of the median filter [6] where the complexity of the algorithm is $O(n)$ (for a window of size n x n).

3 Temporal and 3-D Filtering

The current IPL interface operates on single images. Several filters in space can easily be extended to operate in the temporal domain, i.e. to filter image sequences.

Using a new and relatively simple idea of re-ordering temporal data into a spatial data structure, temporal filtering can be implemented using the IPL or *any other* fast spatial image processing library. After completion of the operation, re-ordering has to be performed. In the experiments (see Section 4) we evaluate the whole process for the color median filter.

There are two possibilities for the process of fusion (ordering), filtering and extraction (re-ordering). Figure 3 displays the first possibility, Figure 4 displays the second. Both Figures show the case of a time filter of size 3.

In the first case (Figure 3), all three color images $\boldsymbol{f}^{(k-1)}$, $\boldsymbol{f}^{(k)}$, and $\boldsymbol{f}^{(k+1)}$ are fused into one large image $\boldsymbol{g}^{(k)}$ with $3 \cdot N$ rows:

$$\boldsymbol{g}_r^{(k)} = \begin{cases} \boldsymbol{f}_{\lfloor r/3 \rfloor}^{(k-1)}, & \text{if } r \bmod 3 = 0 \\ \boldsymbol{f}_{\lfloor r/3 \rfloor}^{(k)}, & \text{if } r \bmod 3 = 1 \\ \boldsymbol{f}_{\lfloor r/3 \rfloor}^{(k+1)}, & \text{if } r \bmod 3 = 2 \end{cases} \qquad \text{for } r = 0, \ldots, N-1 \tag{4}$$

where $\boldsymbol{f}_r = \left[\boldsymbol{f}_{r,j}\right]_{j=0\ldots M-1}$. This means the first row of $\boldsymbol{f}^{(k-1)}$, followed by the first row of $\boldsymbol{f}^{(k)}$, followed by the first row of $\boldsymbol{f}^{(k+1)}$, followed by the second row of $\boldsymbol{f}^{(k-1)}$, and so on, are fused into image $\boldsymbol{g}^{(k)}$. Now, the $(1,3)$ spatial filter (where 1 column and 3 rows are used as mask) is applied to this large image. Afterwards the rows $1, 4, 7, \ldots, 3 \cdot N - 2$ are extracted from the large image, leading to the time filtered image. This fusion-filtering-extraction process can easily be extended to larger sizes of the time filter. The only restriction is that the time filter size is odd, a restriction that is very common.

In the second case (Figure 4), the rows $\boldsymbol{f}_l^{(k-1)}$, $\boldsymbol{f}_l^{(k)}$ and $\boldsymbol{f}_l^{(k+1)}$ are fused into N single images with 3 rows and M columns, for $l = 0, \cdots, N-1$. Each of these images is filtered separately by the (1,3) spatial filter. Afterwards the middle row of each image (here the second row) is extracted, leading to the temporally filtered image. Again, the only restriction is that the filter size is odd.

Both described methods can very easily be extended to a space *and* time filter of size (x, y, t) for $y = 1$. All that has to be done is to apply a filter of size (x, t) instead of $(1, t)$.

Whereas all listed rank-ordering filters can be directly extended to a space *and* time filter (3-D filter) of size (x, y, t) with the property $t \cdot y \bmod 2 = 1$, there exists no such extension for the other filters. The process of fusion, filtering, and extraction is shown in Figure 5. Instead of fusing one line out of each source image to a new image, y lines are used, leading to N images with $y \cdot t$ rows and M columns. These images are filtered and the middle row is extracted, leading to the time and space filtered result image. This process has a very large overhead (leading to larger time values for processing, see Table 2) because each line of a source image has to be copied into y images.

For all linear 2-D filters (like Gauss or box filter), it is possible to extend them to 3-D filters by filtering the image with the spatial version of the filter, followed by a filtering operation with the temporal version, i.e. size $(1, 1, t)$ (as described above).

4 Application in Medical Imaging

We applied the color median time filter to an endoscopic image sequence. Since we did not find significant computing time differences between the two possibilities of fusion, filtering and extraction (see Section 3 and Figures 3, 4), we used the second method (see Figure 4). The evaluation of the filtered image sequence was done by five experienced physicians who had to judge the image quality.

The first idea was to judge single images. Looking at the images, we saw filtering artefacts like replicated image structures. But these artefacts were not seen looking at the whole images sequence. This is due to the fact that the artefacts mostly occur only for one frame and if the image sequence is displayed with 25 frames per second, an artefact that only occurs in one frame is not visible for humans. Therefore, we decided that the physicians should judge the whole image sequence rather than evaluating single images. The filtered image sequence (170 frames) was judged by the physicians as follows: the image disturbance by small water vapor particles and highlights was reduced

while the image quality of not disturbed areas did not change.

One result of the implemented time color median (size 3) is shown in Figure 7,[1] where the difference image was transformed by

$$c'_{ij} = \begin{cases} 255, & \text{if } c_{ij} > 50 \\ c_{ij} \cdot \frac{255}{50}, & \text{if } c_{ij} \leq 50 \end{cases} \text{ for } i = 0 \ldots N-1, j = 0 \ldots M-1 \qquad (5)$$

for all three color channels $c \in \{r, g, b\}$.

Table 2 shows the computation time of our implementation (using IPL) of the temporal color median filter for different sizes. The $(1, 1, 3)$, $(1, 1, 5)$, and $(3, 1, 3)$ median filter can be applied in real-time. The large copy overhead for the (x, y, t) filter for $y > 1$ can be seen very clearly.

Table 2. Computation times of the implemented color median filter measured on 100 random color images of size 720×576 with a computer with a 800 MHz Pentium III processor. The computation times includes the process of fusing, filtering and extraction. Mean values $\pm$ standard deviations are shown in milliseconds. The (x, y, t) filter for sizes > 3 was only evaluated on one single image because the time needed for those filter was larger than 21 seconds and the information about the mean value and the standard deviation is not important in this case.

$(1,1,t)$	
$t=3$	20 ± 3
$t=5$	40 ± 6
$t=7$	613 ± 115
$t=9$	$1,077 \pm 307$

$(x,1,t)$	
$x=t=3$	32 ± 3
$x=t=5$	201 ± 2
$x=t=7$	$2,262 \pm 9$
$x=t=9$	$3,412 \pm 57$

(x,y,t)	
$x=y=t=3$	$3,125 \pm 232$
$x=y=t=5$	$21,825$
$x=y=t=7$	$79,565$
$x=y=t=9$	$274,706$

5 Conclusions

With the presented method of fusion, filtering and extraction, it is possible to implement temporal filter which use a spatial version of the filter. Using the IPL we showed that the color median filter for sizes $(1, 1, 3)$, $(1, 1, 5)$ and $(3, 1, 3)$ can be applied in real-time.

The evaluation by the physicians who had to judge the image enhancement of the temporal color median filtered endoscopic image sequence showed that color median filtering improves (in this case) image quality.

References

1. http://support.intel.com/support/performancetools/libraries/ipl/index.htm, Revision 2.0.
2. http://www.intel.com/research/mrl/research/opencv/.

[1] If the image sequence is displayed, or if the two images are shown sequentially on a projection, the enhancement is more obvious.

3. K.-H. Franke, editor. *5. Workshop Farbbildverarbeitung*. Schriftenreihe des Zentrums für Bild- und Signalverarbeitung e.V. Ilmenau, Ilmenau, 1999.
4. H. U. Lempke, M. W. Vannier, K. Inamura, A.G. Farman, and K. Doi, editors. *Computer Assisted Radiology and Surgery, 15$^{\text{th}}$ International Congress and Exhibition*. Elsevier Science B.V., 2001.
5. C. Palm, T. Lehmann, and K. Spitzer. Bestimmung der Lichtquellenfarbe bei der Endoskopie makrotexturierter Oberflächen des Kehlkopfs. In Franke [3], pages 3–10.
6. D. Paulus. Aktives Bildverstehen, 2000. Habilitationsschrift in der Praktischen Informatik, Universität Erlangen-Nürnberg.
7. B. Smolka, M. Szezepansik, and K. Wojciechowski. Random walk approach to the problem of impulse noise reduction. In Franke [3], pages 43–50.
8. F. Vogt, C. Klimowicz, D. Paulus, W. Hohenberger, H. Niemann, and C. H. Schick. Bildverarbeitung in der Endoskopie des Bauchraums. In H. Pöppl, editor, *5. Workshop Bildverarbeitung für die Medizin*, pages 320–324, Heidelberg, 2001. Springer.

Figure 1. Absolute differences between the color vector median filter and the IPL color median filter, on the left side the difference image of the random image, on the right side the difference image of the endoscopic image.

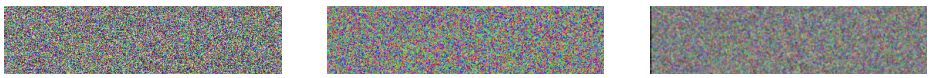

Figure 2. Original random image (left), color vector median filtered random image (middle) and IPL color median filtered random image (right)

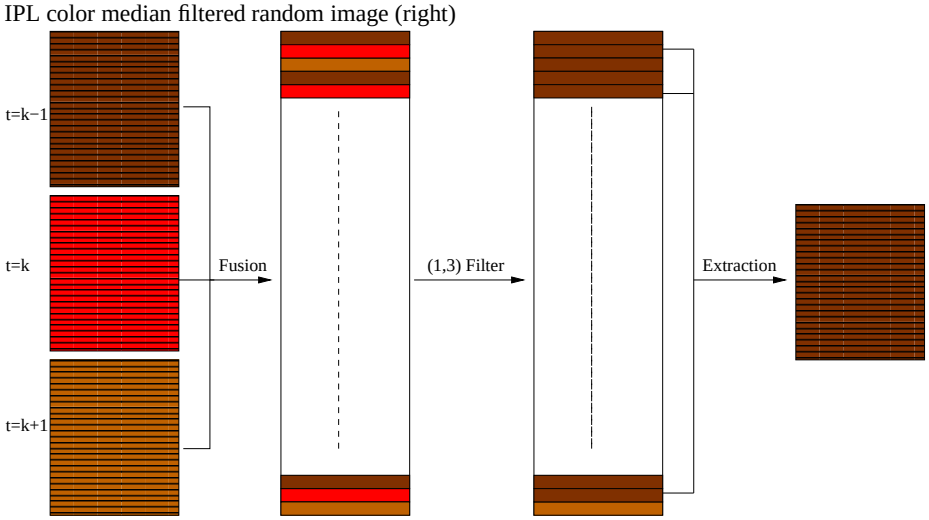

Figure 3. The process of fusion, filtering and extraction to obtain a temporal filter with the help of the spatial version of this filter is shown (size 3 of the temporal filter). The source (color) images are fused into one large image that is filtered with the spatial version of the filter.

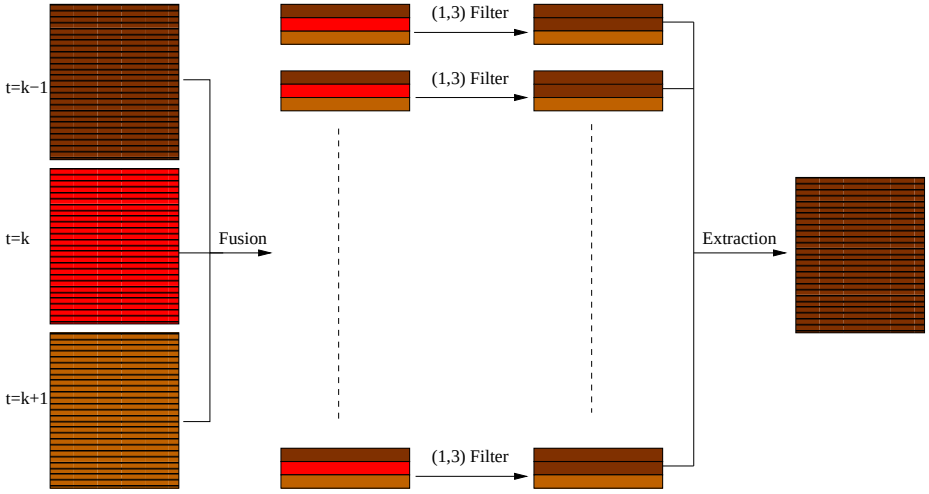

Figure 4. The process of fusion, filtering and extraction to obtain a temporal filter with the help of the spatial version of this filter is shown (size 3 of the temporal filter). The source (color) images are fused into N images (N = number of rows of the source images) that are filtered with the spatial version of the filter.

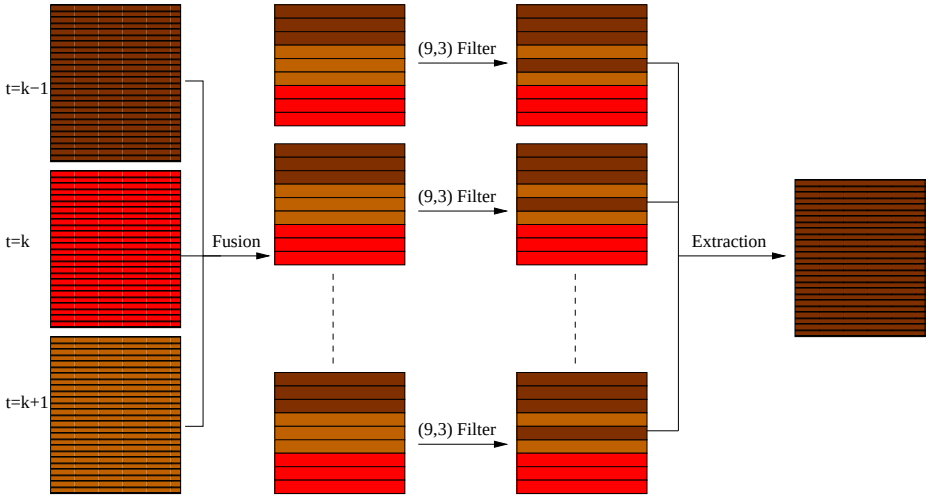

Figure 5. The process of fusion, filtering and extraction to obtain a temporal and spatial filter (3-D filter) is shown. The source (color) images are fused into N images (N = number of rows of the source images) that are filtered with the spatial version of the filter. The case of a $(3, 3, 3)$ filter is shown.

Figure 6. Original endoscopic image (left), color vector median filtered endoscopic image (middle) and IPL color median filtered endoscopic image (right)

Figure 7. Original image (left), filtered image (middle), difference image (right), where the pixel values were transformed by Equation 5.