MOTION ESTIMATION IN FROTH IMAGE SEQUENCES

J.J. Francis, G. de Jager (Digital Image Processing Group,
Department of Electrical Engineering,
University of Cape Town)
and D. Bradshaw (Minerals Processing Research Unit,
Department of Chemical Engineering,
University of Cape Town)
Email for correspondence: {jfrancis, gdj}@dip.ee.uct.ac.za
db@chemeng.uct.ac.za

Abstract

The problem of estimating a dense motion field for froth image sequences is explored. Four algorithms are quantitatively examined. The quantitative data set include a froth image sequence and the same sequence pre-warped. In all cases the performance of the algorithm is measured by, reconstructing the warping field, and estimating the error between the image and an estimated version generated from the computed motion field.

Introduction

The problem of estimating motion in froth image sequences has gained importance with the appearance of several systems designed to measure froth visual parameters. For some examples of previous systems see, Botha et al. (1999), Bezuidenhout et al. (1997), Sadr-kazemi and Cilliers (1997), Kordek and Kulig (1997) and Symonds and De Jager (1992). From a process control perspective, the motion of a froth close to the weir, with froth level, provides an estimate of the mass-pull of the cell. The stability of the froth is a useful parameter. Froth stability is a function of pulp density and reagent addition rates. In principle, stability can be measured by finding discontinuities in the motion field. The discontinuities will be caused by bubbles bursting and revealing new bubbles, or by bubbles merging. The bubble merge and burst events cause gross changes in the intensity profile of the image. These changes will result in a bubble tracking algorithm generating large errors. These errors will then present themselves in a motion field as a discontinuous cluster of motion vectors. This implies that the first step in measuring these discontiniuities is to correctly estimate a dense motion field.

This paper will examine the need for preprocessing froth images before motion estimation. A motion estimation algorithm will be proposed. To provide a quantitative analysis of performance this proposed algorithm will be compared to some algorithms described in the open literature as being applied to motion analysis of froth sequences. Further quantification of performance will be performed by measuring how well the proposed algorithm estimates one image, given the motion field and the next image in the sequence.

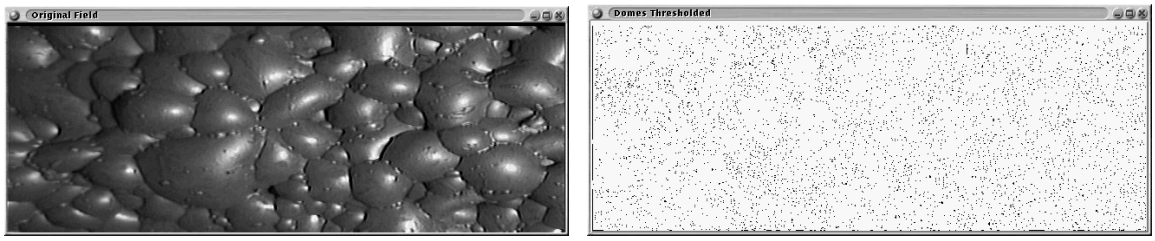
The paper proceeds as follows, first the need for denoising is examined and a denoising algorithm examined. The background for some motion estimation algorithms is presented. Finally the algorithms are experimentally compared.

Denoising

It is expected that the images obtained will be corrupted by additive noise. The need for denoising will be shown by the following example.

- Consider the image shown in figure 1(a).
- Consider the positions of the maxima of the example image as shown in figure 1(b).

Clearly there are more maxima than correspond to visually significant features. Ideally, the maxima should correspond to the bubble highlights, or regions of maximum intensity on each bubble. Ideally the effect of noise on subsequent processing should be minimised. A variety of denoising algorithms exist on the literature, these range from linear filters (mean filters) (see, for example, Jahne (1993)) to non-linear approaches (see, for example, Perona and Malik (1990), Schaefer (1987), Chen et al. (1999), Chin and Yeh (1983) and Wang (1994)). A reasonable requirement of any denoising technique is that the edges of the bubbles be preserved. This, in theory at least, allows for the extent of the bubble to be determined via some segmentation process (for example watershed (see Vincent and Soille (1991))). Of the possible techniques, anisotropic diffusion is selected for its edge preservation property (see Perona and Malik (1990)) and capacity to smooth across various scales. Subsequently, anisotropic diffusion is described in some detail.



(a) An example image.

(b) The maxima of the example image. 4 485 maxima detected.

Figure 1: The need for denoising.

Anisotropic Diffusion

The anisotropic diffusion algorithm used is Perona and Malik (1990). A number of extensions and refinements to Perona and Malik's algorithms have been suggested (see, for example, Alvarez and Mazorra (1994); Catte et al. (1992); Whitaker and Pizer (1993); You et al. (1996). The algorithm implements a spatially variant form of the heat diffusion equation. This allows for edge preservation and, occasionally, for edge enhancement. The form of the diffusion equation is as follows:

$$f_t = \nabla \cdot (g(\|\nabla f\|)\nabla f) \tag{1}$$

Perona and Malik suggest the following for $g()$:

$$g(\|\nabla f\|) = e^{(-(\|\nabla f\|/K)^2)} \qquad \text{or,} \tag{2}$$

$$g(\|\nabla f\|) = \frac{1}{1 + (\frac{\|\nabla f\|}{K})^2} \tag{3}$$

and show that a time step of,

$$0 \leq \Delta t \leq 0.25 \tag{4}$$

is needed for stability.

K may be set by hand, or by Canny's noise estimator (see Canny (1986)) ∇ represents the grad operator, or in this case $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$. The Euclidean norm is represented by $\| \cdot \|$.

Motion estimation algorithms

A wide variety of motion estimation algorithms exist in the literature (see any of the proceedings for the IEEE Conference on Image Processing). Without being too precise, the algorithms can be divided into three classes block matching techniques, gradient based optical flow and segmentation based techniques. The following algorithms are selected to give an indication of the relative performance of the various classes.

Some motion estimation algorithms are described. Block matching, Gradient based optical flow, Pixel tracing and Watershed based motion estimation are described. All the motion estimation algorithms implicitly assume that the intensities of pixels is conserved during motion. A further assumption is that the motion is linear between frames. These assumptions are not too unrealistic, considering that the frames are $\frac{1}{24}s$ apart.

Block Matching

Block matching has been used extensively in compressing image sequences, notably in the H.261 and MPEG-2 standards (see Horne et al. (1996)). The block motion estimation algorithm used is the *Successive Elimination Algorithm* developed by Li and Salari (1995). This is an optimised full search algorithm. The essence of a block matching is to divide images into a series of overlapping blocks. The motion of the pixel centred within a block, is the vector between the pixel position and the centre pixel of the best match block in the next image. The match is found either by the Mean Square Error (MSE), or Mean of the Absolute Differences (MAD). The search space can be made to vary from a small neighbourhood around the block of interest, to the entire image.

Optical flow estimation

Here the motion is modelled as follows:

$$f_x u + f_y v + f_t = 0 \tag{5}$$

f_x, f_y and f_t represent the derivative of the image with respect to x, y and time, respectively. u and v represent the x and y coordinates of the motion field at some point. See Horn and Schunk (1981) for the seminal work on gradient based optical flow estimation. Equation (5) provides on constraint on the optical flow constraint. In order to generate the (u, v) values it is necessary to add a second constraint. Horn and Schunk suggest minimising,

$$E = \iint_{x,y} dx\ dy\ (f_x u + f_y v + f_t)^2 + (u_x^2 + u_y^2 + v_x^2 + v_y^2)^2 \tag{6}$$

The second constraint can be seen to impose smoothness on the motion vector field.

Pixel Tracing

Pixel tracing, developed by Nguyen and Holtam (1997), involves matching a block taken from the centre of an image with blocks at various positions in the subsequent image. In a sense, this is block matching with one large block and a subsampled search space.

The algorithm proceeds as follows:

1. A square region of around 100 pixels wide is selected from the centre of a frame at time t.
2. In the frame at time $t + \Delta t$ starting from the centre of the eight directions are searched: North, South, East, West, North-East, North-West, South-East and South-West. The number of pixels searched along each direction is chosen from twenty to thirty.
3. The position of the best match is found by taking the position of minimum sum of squared error.
4. The motion vector is then the difference between the two positions; start and best match.

Watershed based motion estimation

Watershed segmentation is an example of region growing segmentation techniques (see Dobrin et al. (1994); Vincent and Soille (1991) and Beucher and Meyer (1993) for example). A seed region is found (these are referred to as markers) and these are then grown, taking the profile of the image into account, until the regions run into each other. The difficulty lies in the initial choice of seed regions.

To smooth the motion field at bubble boundaries the motion field is smoothed between bubble centres. After all, usually two neighbouring bubbles have similar motion.

Watershed based motion estimation is an example of region matching in some feature space. Several examples of region based motion estimation algorithms exist in the literature, for example, Kottke and Sun (1994) and De Smet and De Vleeschauwer (1997).

The algorithm proceeds as follows:

- After Watershed segmentation a number of features are calculated on each Watershed region.
    1. The grey level mean.
    2. The grey level standard deviation.
    3. The standard deviation of the x coordinates of the pixels.
    4. The standard deviation of the y coordinates of the pixels.
    5. The number of pixels.
- The features (from both frames) are normalised to unit variance and zero mean.
- The the best match region in frame $t + \Delta t$ is found by minimising:

  $ERROR = \arg\min_q (R_t^p - R_{t+\Delta t}^q)^2$

  Where p represents the search region in frame t, q the reference region(s) in frame $t + \Delta t$ and R is the feature vector.

Results

Results are generated to illustrate the denoising algorithm. Quantitative and qualitative results are generated for the motion estimation algorithms.

The effect of denoising can be seen in the following figure 2. Perona and Malik's algorithm is used. The time step used was 0.2, and the number of iterations 4, Equation (2) was chosen for $g()$. Note that the equalised error image has a rough (random) texture, in other words there is not much correlation between adjacent pixels. This indicates that the discarded information is more likely to be noise than image data.

A quantitative analysis of the motion estimation algorithms is needed to evaluate performance quantitatively. This is done by estimating the image at time t from the motion field and the image at time $t + \Delta t$ (here $\Delta t \approx \frac{1}{24} s$). The error between the estimated and actual images is a measure of the performance of the algorithm. The data set used is 224 froth images (or about 9 seconds worth) taken from the Merensky rougher bank at Amandelbult. Only the even field of each interlaced frame is used.

Table 1 shows the comparison of the proposed algorithm to some other motion estimation algorithms. The average time given is the average time needed to process a pair of images. Some care should be exercised in comparing the average running time between algorithms since it is impossible to guarantee that the conditions during each run were identical. At best the average running time allows a relative ranking between the algorithms. Table 2 shows the same for the pre-warped date. Here each image is warped. The motion field is then estimated between the image and its warped version. This enables the algorithms to be compared for large motion. Figure 3 shows the original images used for the qualitative comparison. Figures 4, 5, 6 and 7 and show the qualitative comparison between the different algorithms.

All the images were denoised using Perona and Malik's anisotropic diffusion algorithm. The time step used was 0.2, and the number of iterations 4, Equation (2) was chosen for $g()$. The algorithms were implemented in Java1.2 running on a dual processor Celeron PC. The operating system used was FreeBSD 4.1 Stable. The algorithms used were initialised as follows:

Pixel Tracing The dimensions of the central block used for pixel tracing is 101×101, and the number of pixels searched along each search direction is 30.

Block Matching The size of each block is 9×9 and the search window is 5×5 pixels.

Optical Flow The number of iterations (or refinement steps) used were 16.

Watershed The radius of the search area is limited to 50 pixels.

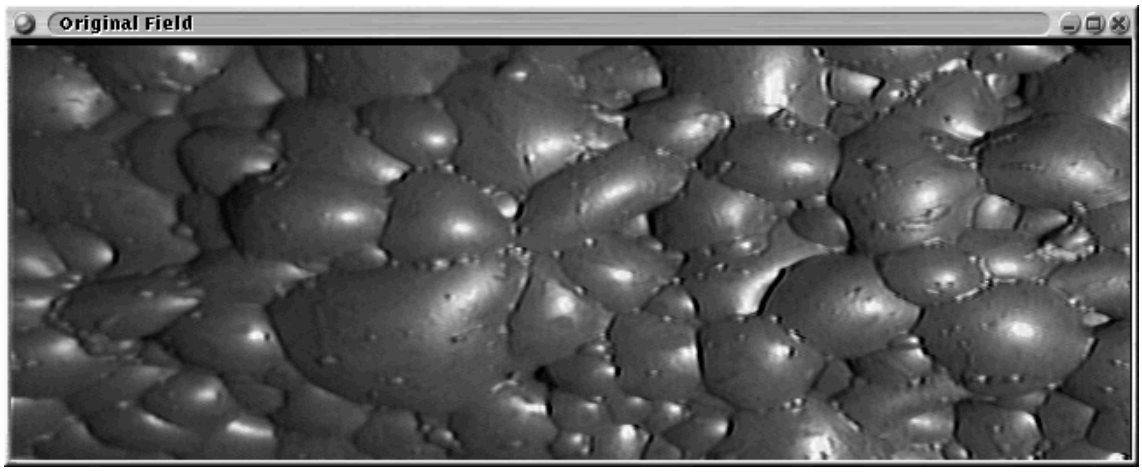
Table 1: Comparison between different motion estimation algorithms.

Algorithm	Error		
	MAD	MSE	Average Time (ms)
Block Matching	2.022	24.36	483 351.1
Optical Flow	3.079	105.6	12 038.9
Pixel Tracing	4.569	150.3	1 130.7
Watershed	7.460	303.4	395 066.2

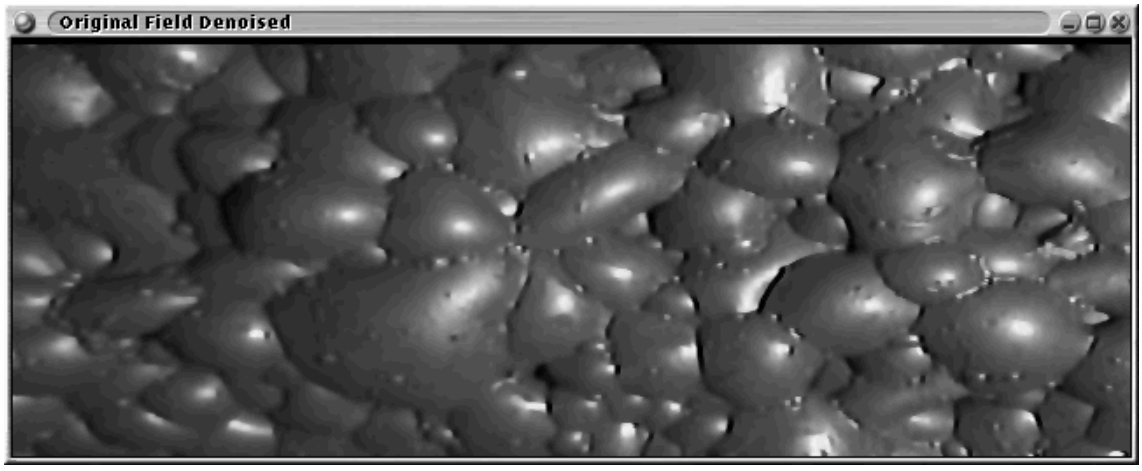
Pixel tracing provides a single motion vector for the centre pixel of the image. This vector has been replicated over the region of interest of the image. This explains the consistent motion field across figure 4(a). The smooth motion field explains the lack of discontinuities in the registered image.

The motion field due to block matching shows a fairly uniform motion field even without a smoothness constraint on the motion field. This can be explained by the large block sizes used 15×15 and the small search area (a block of width 5).

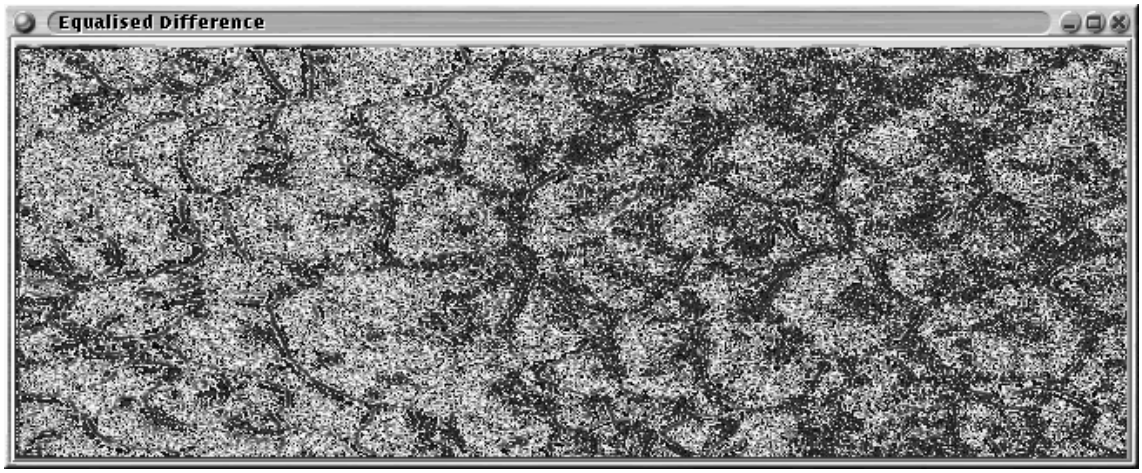
The motion field due to Horn and Schunk's algorithm is fairly smooth. A large number of iterations is necessary to ensure that the motion estimates at corners and edges of the bubbles diffuse into the interior of the bubbles.



(a) Original image.



(b) Denoised image.

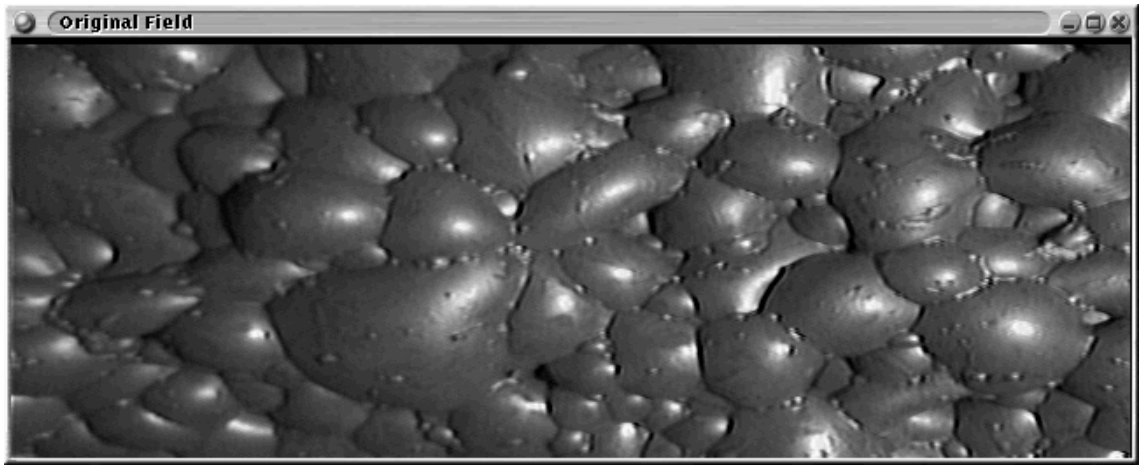


(c) Equalised difference image.

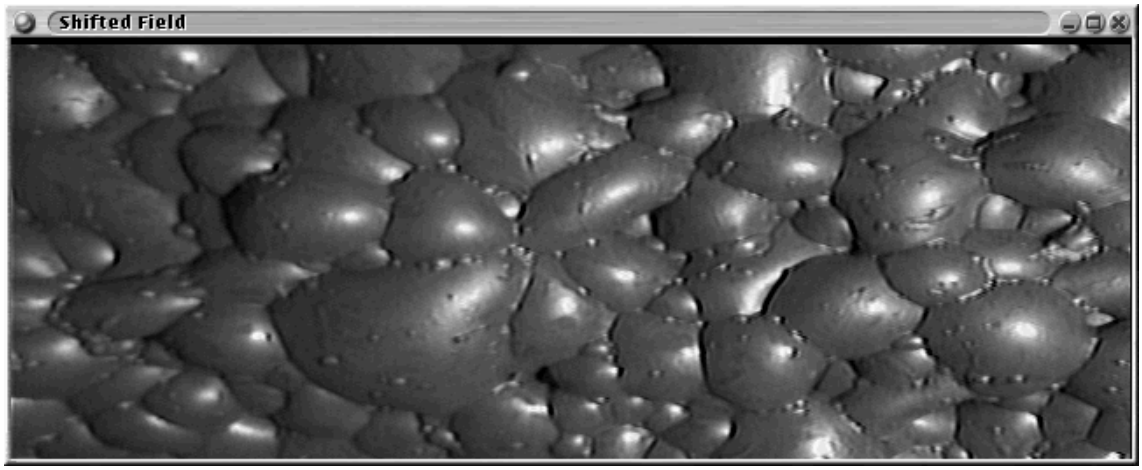
Figure 2: The effect of denoising on froth images.

Table 2: Comparison between different motion estimation algorithms for pre-warped data.

Algorithm	Field Error		Register Error	
	MAD $(\times 10^{-3})$	MSE $(\times 10^{-3})$	MAD $(\times 10^{-3})$	MSE $(\times 10^{-3})$
Block Matching	11.3	74.26	6.790	392.6
Optical Flow	25.91	220.1	61.31	3 174.0
Pixel Tracing	32.26	198.9	48.15	1 626.0
Watershed	9.940	74.48	17.96	768.6

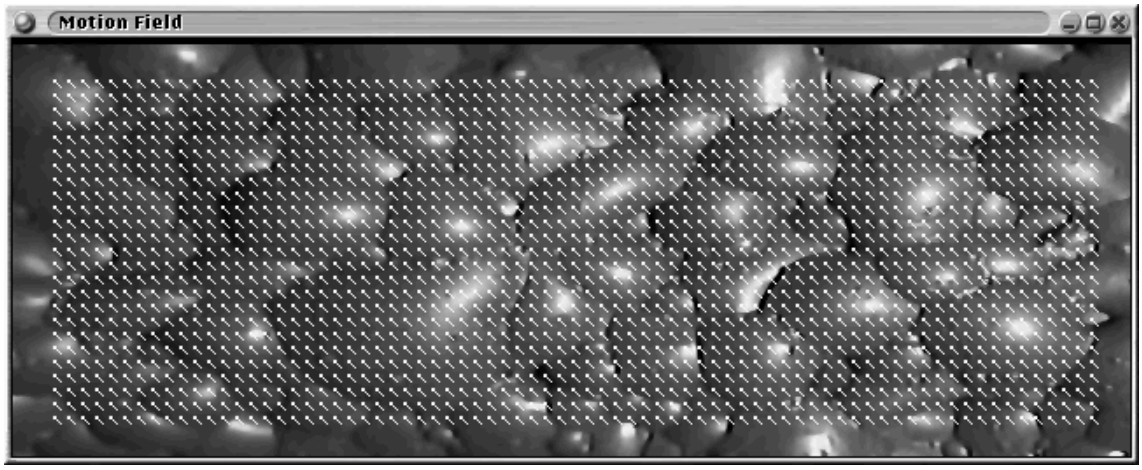


(a) Even field at time t.

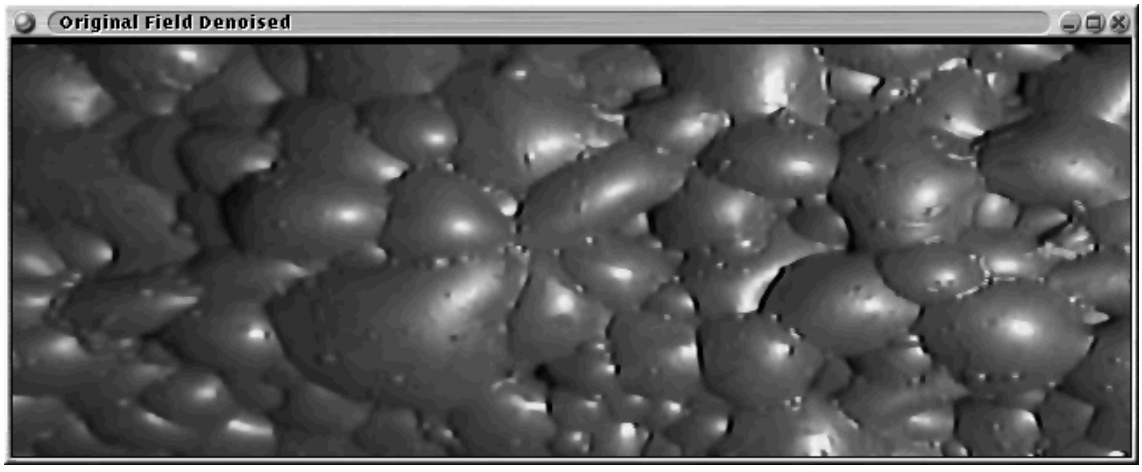


(b) Even field at time $t + \Delta t$. $\Delta t = \frac{1}{24}$ ms.

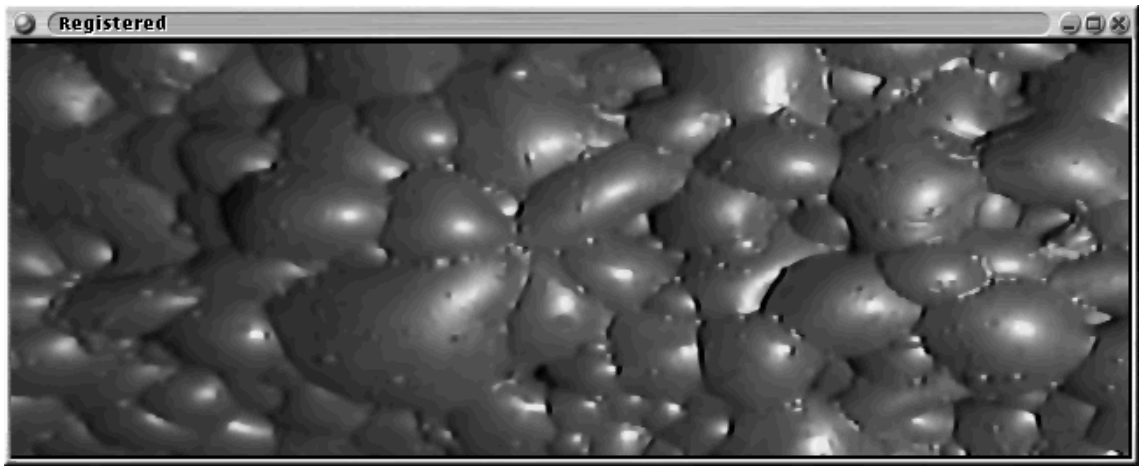
Figure 3: The input images.



(a) An example motion field due to Pixel Tracing. Motion field scaled by 4. Average motion $(1, -1)$.



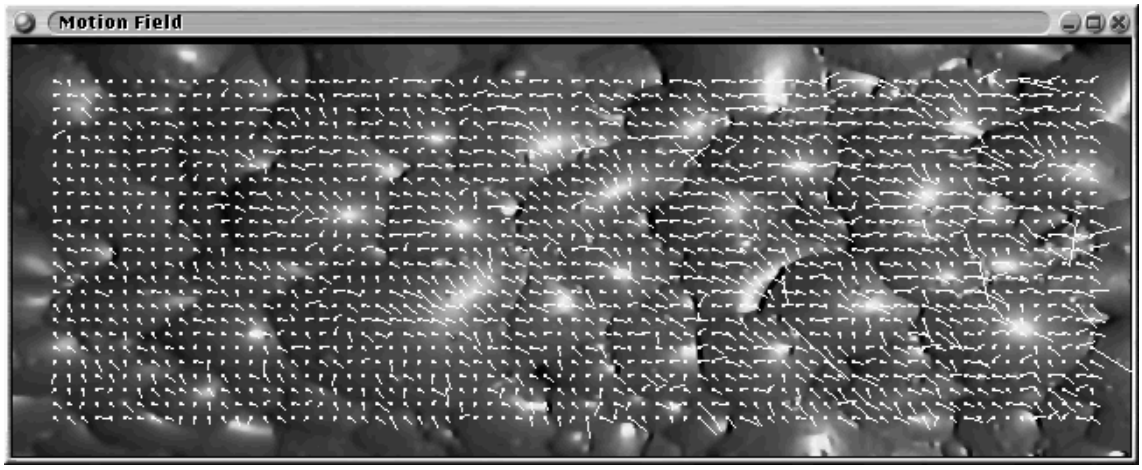
(b) Original image, denoised.



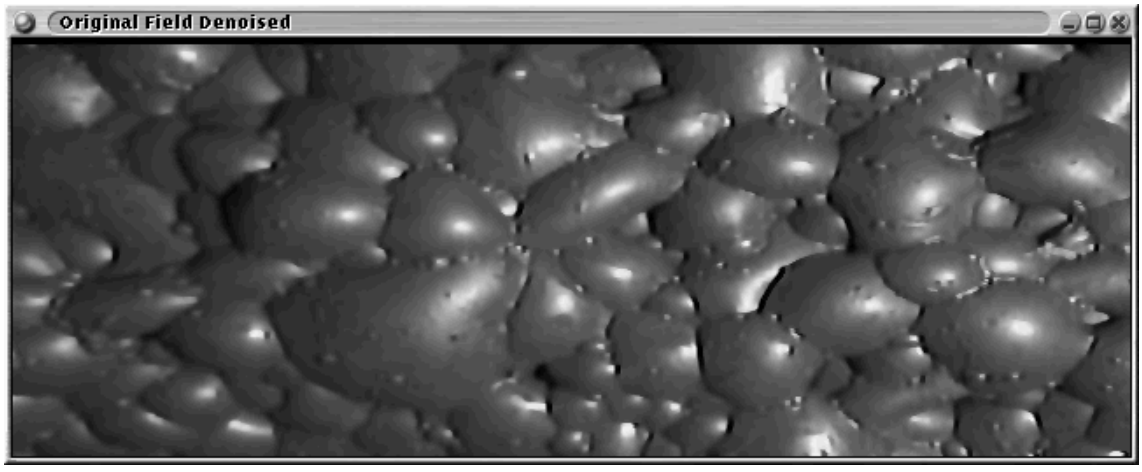
(c) Registered image.

(d) Registered error, stretched. Energy in image is 6 326 305.

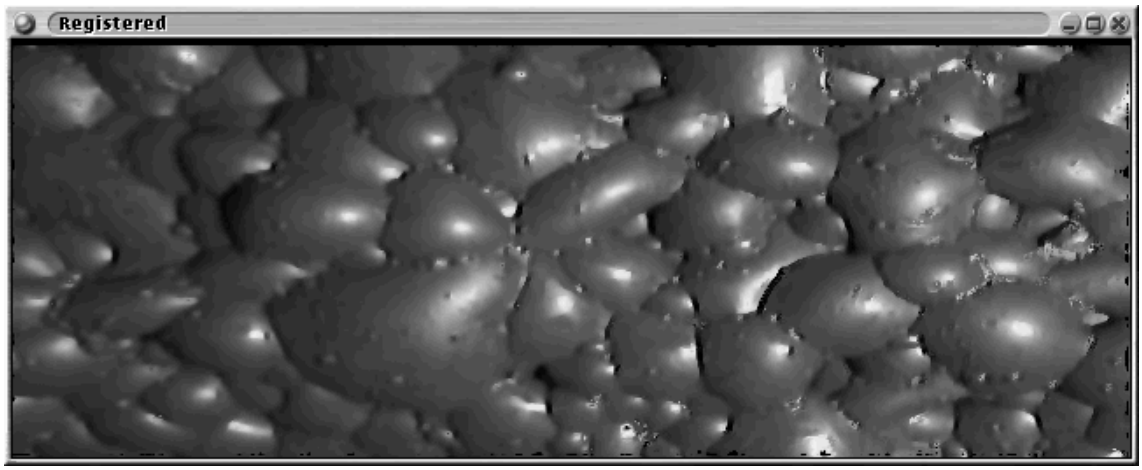
Figure 4: Motion estimation via Pixel Tracing.



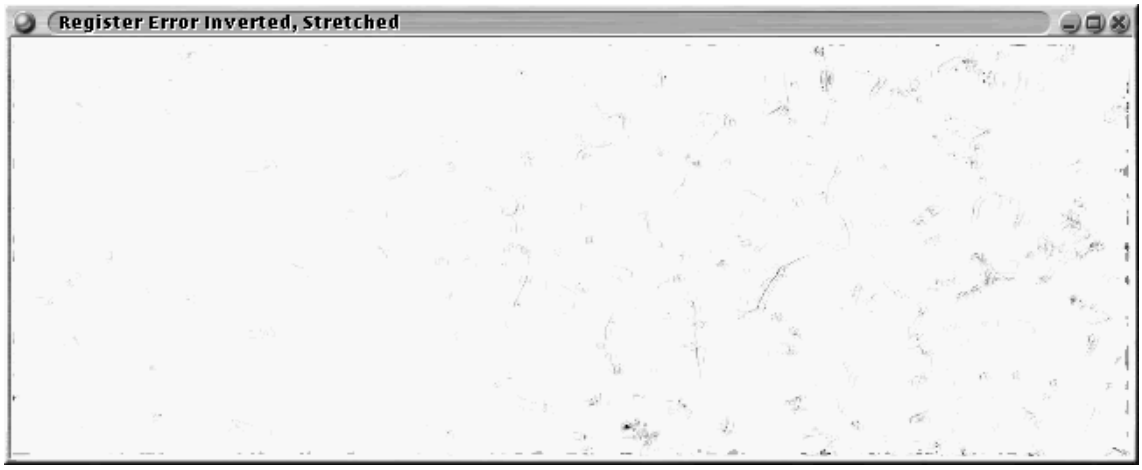
(a) An example motion field due to Horn and Schunk. Motion field scaled by 4. Average motion $(0.94, -0.55)$.



(b) Original image, denoised.

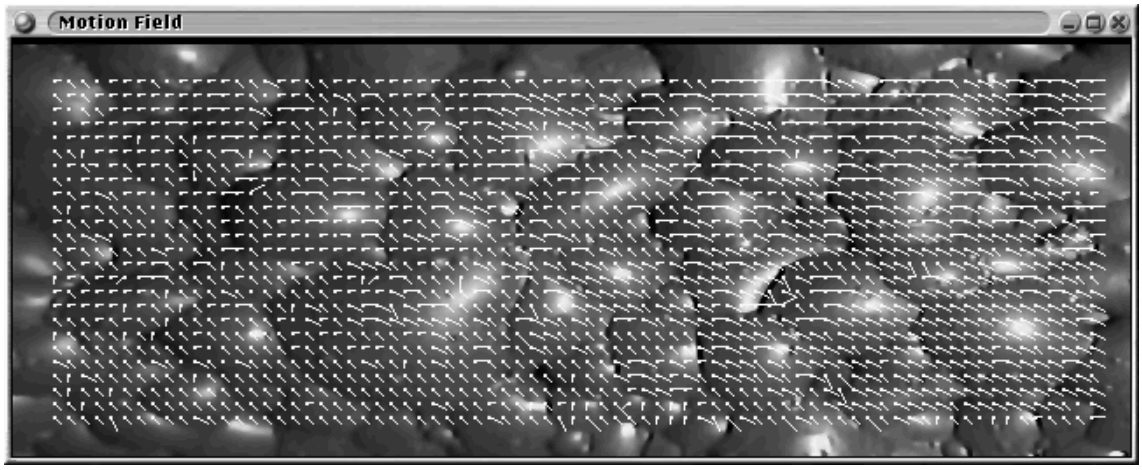


(c) Registered image.

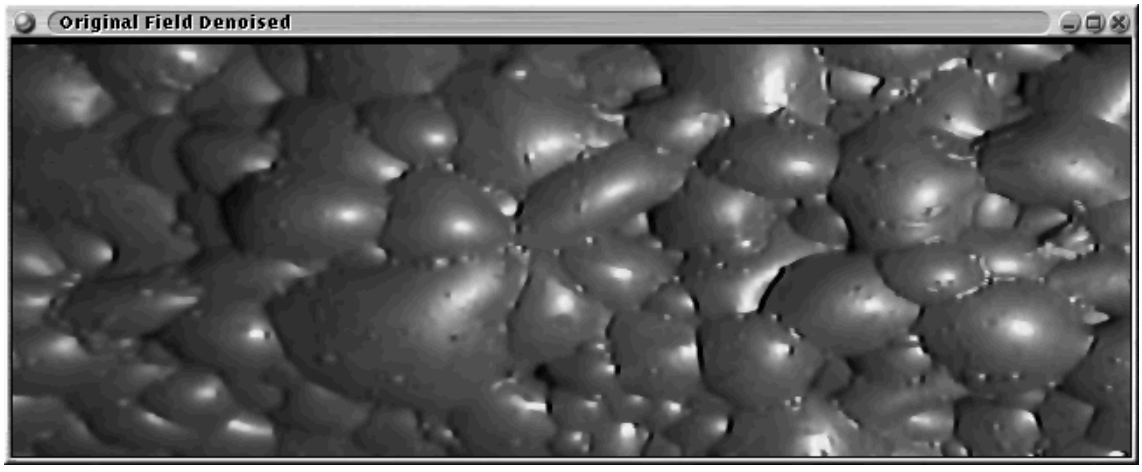


(d) Registered error, stretched. Energy in image is 3 331 470.

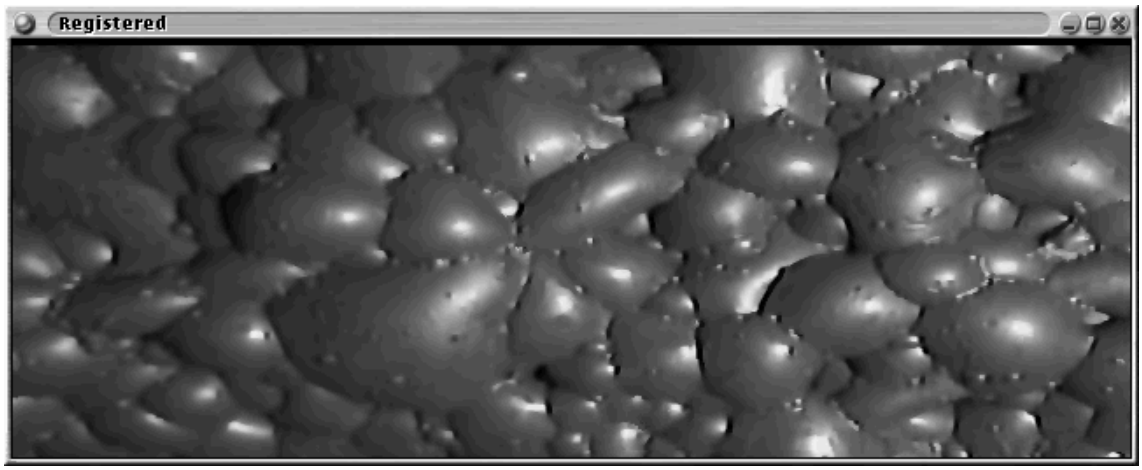
Figure 5: Motion estimation via Horn and Schunk.



(a) An example motion field due to Block Motion estimation. Motion field scaled by 4. Average motion $(1.2, -0.70)$.



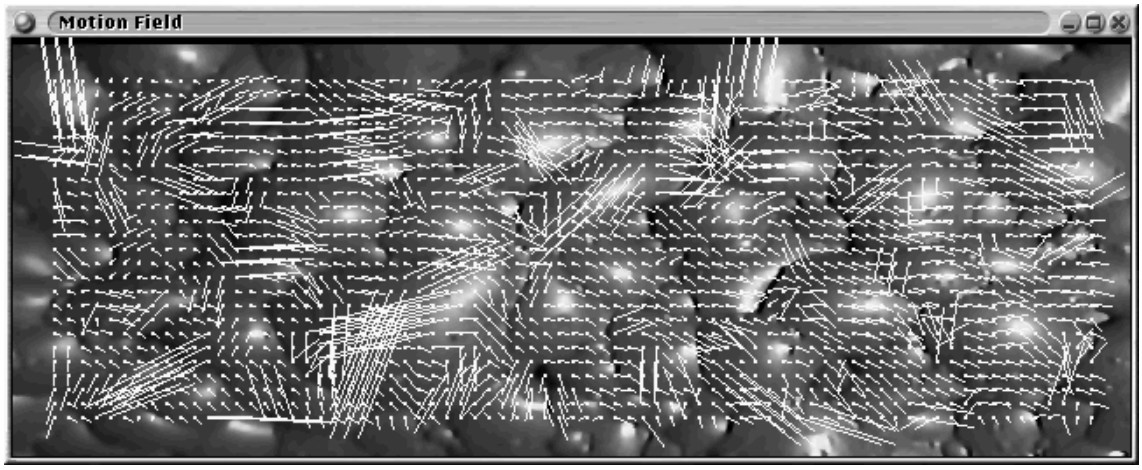
(b) Original image, denoised.



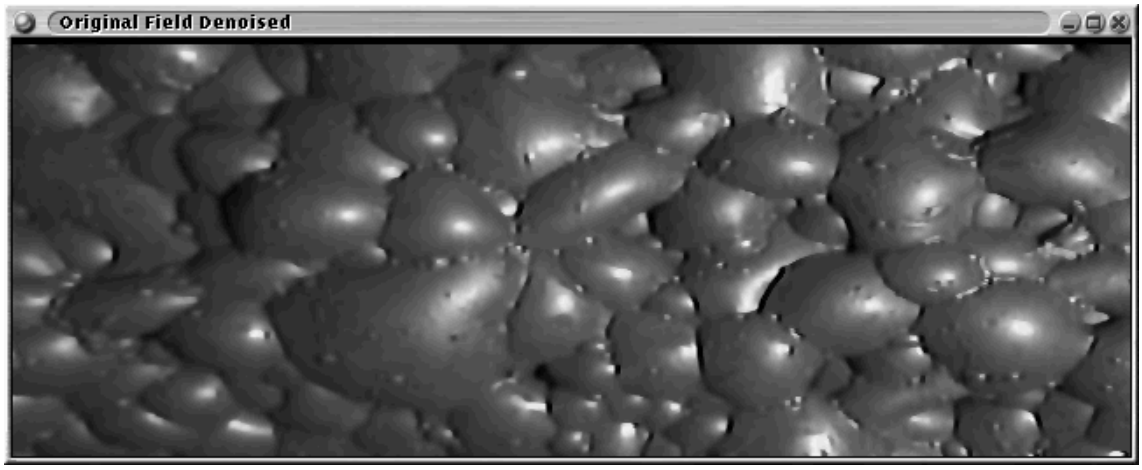
(c) Registered image.

(d) Registered error, stretched. Energy in image is 2 148 359

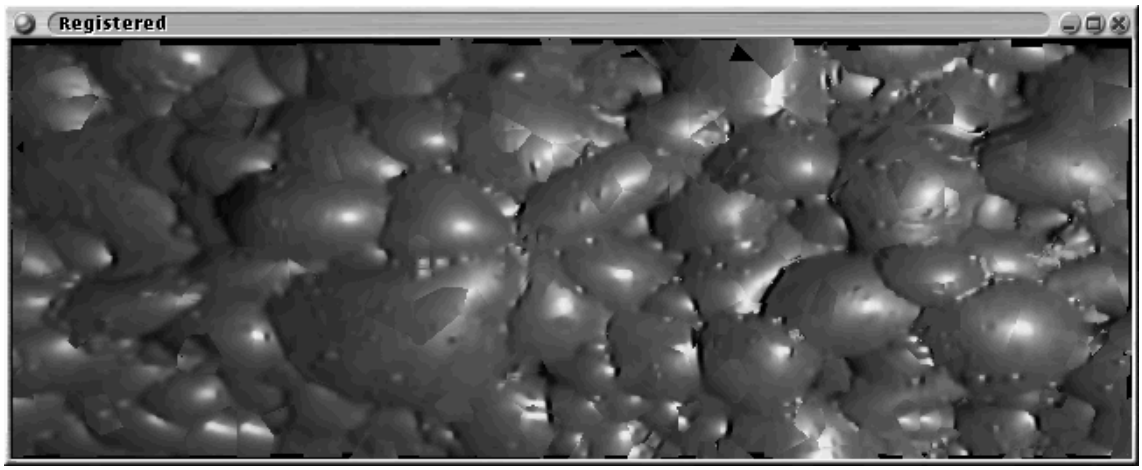
Figure 6: Motion estimation via Block Motion Estimation.



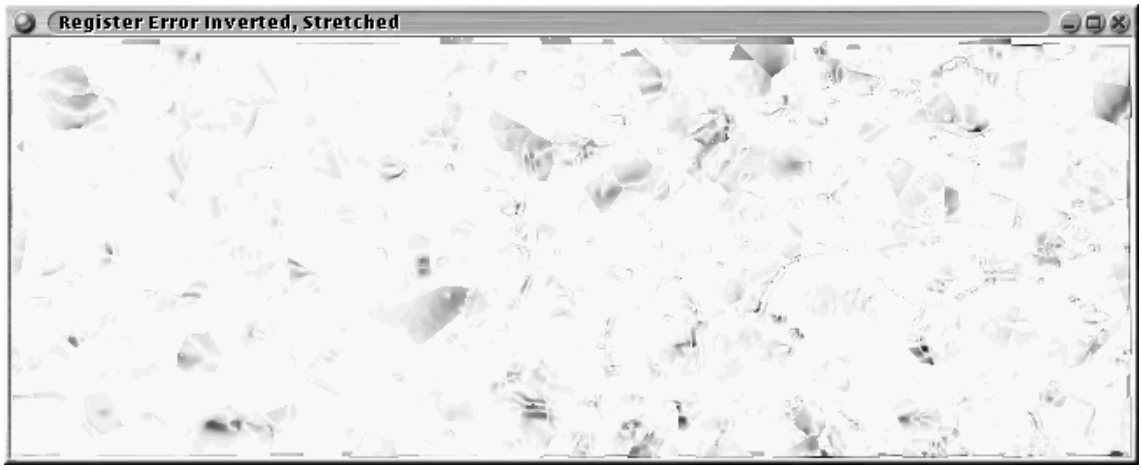
(a) An example motion field due to watershed based motion estimation. Motion field scaled by 4. Average motion $(1.1, -0.63)$.



(b) Original image, denoised.



(c) Registered image.



(d) Registered error, stretched. Energy in image is 22 740 619

Figure 7: Motion estimation via Watershed based Motion Estimation.

The watershed based motion estimation algorithm shows a discontinuous motion field. The problem with this algorithm is that there is no overall smoothness constraint on the motion field. The bubbles in a group of neighbouring bubbles, taken as a whole, do not move together. It should be noted that the average time needed to watershed is $9\,532$ ms.

Conclusions

Qualitatively, the Horn and Schunk algorithm provides a smooth motion field which gives some idea of the irregularities in the motion between two fields. Quantitatively, the low energy in the difference between the registered and original image, indicated that the motion field captures more of the changes than the other algorithms.

The performance enhancement of Pixel Tracing comes at the price of accuracy. For example, Pixel Tracing cannot accurately determine motion along $(1, 2)$.

The watershed based motion estimation algorithm does not do as well as the other algorithms for small motion, but estimates large motion well. The algorithm is considerably slower than the Horn and Schunk or Pixel Tracing algorithms. Running time is similar to the Block Matching algorithm.

In short, for estimating small motion optical flow is preferred if a dense motion field is required. For large motion either block matching or watershed based motion estimation provide the desired performance.

Acknowledgements

- NRF and AMPLATS for financial support.
- AMPLATS for froth image sequences.

References

Alvarez, Luis and Mazorra, Luis. Signal and image restoration using shock filters and anisotropic diffusion. *Siam Journal of Numerical Analysis*, 31(2):590 – 605, April 1994.

Beucher, S. and Meyer, F. The morphological approach to segmentation: The watershed transformation. In Dougherty, E. R., editor, *Mathematical Morphology in Image Processing*, chapter 12, pages 433 – 481. Marcel Decker, 1993.

Bezuidenhout, M., Van Deventer, J. S. J., and Moolman, D. W. The identification of perturbations in a base metal flotation plant using computer vision of the froth surface. *Minerals Engineering*, 10(10): 1057–1073, 1997.

Botha, C. P., Weber, D. M., Van Olst, M., and Moolman, D. W. A practical system for realtime on-plant flotation froth visual parameter extraction. In *Proceedings of IEEE Africon 1999*, September 1999.

Canny, John. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–8(6):679–698, November 1986.

Catte, Francine, Lions, Pierre-Louis, Morel, Jean-Michel, and Coll, Tomeu. Image selective smoothing and edge detection by non-linear diffusion. *Siam Journal of Numerical Analysis*, 29(1):182 – 193, February 1992.

Chen, Tao, Ma, Kai-Kuang, and li Hui Chen. Tri-state median filter for image denoising. *ITIP*, 8(12):1834 – 1838, December 1999.

Chin, Roland T. and Yeh, Chia-Lung. Quantitative evaluation of some edge-preserving noise-smoothing techniques. *Computer Vision Graphics and Image Processing*, 23:67 – 91, 1983.

De Smet, P. and De Vleeschauwer, D. Motion-based segmentation using a thresholded merging strategy on watershed segments. In *Proceedings of the ICIP*, 1997.

Dobrin, Bogdan P., Viero, Timo, and Gabbouj, Moncef. Fast watershed algorithms: analysis and extensions. In *Nonlinear Image Processing V*, volume 2180, pages 209 – 220. SPIE, 1994.

Horn, Berthold K. P. and Schunk, Brian G. Determining optical flow. *Artificial Intelligence*, 17:185 – 203, 1981.

Horne, Caspar, Naveen, T., and Tabatabai, Ali. Study of the characteristics of the mpeg2 4:2:2 profile - application of the mpeg2 in studio environment. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):251 – 272, June 1996.

Jahne, Bernd. *Digital Image Processing : Concepts, Algorithms and Scientific Applications*. Springer-Verlag, Berlin Heidelberg, 2 edition, 1993.

Kordek, Jacek and Kulig, Jaroslaw. The analysis of diffraction patterns of flotation froth as the basis of estimation the content of a useful component. In *Proceedings of the XX IMPC, 21–26 September*, Aachen, 1997.

Kottke, D. P. and Sun, Ying. Motion estimation via cluster matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1128 – 1132, November 1994.

Li, W. and Salari, E. Successive elimination algorithm for motion estimation. *IEEE Transactions On Image Processing*, 4(1):105 – 107, January 1995.

Nguyen, Khoi Ke and Holtam, Peter. The application of pixel tracing techniques in the flotation process. In *Proceedings of the first joint Australian and New Zealand biennial conference on Digital Imaging and Vision Computing and Applications*, pages 207 – 212, 1997.

Perona, P. and Malik, J. Scale-space filtering and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(7):629 – 639, July 1990.

Sadr-kazemi, N. and Cilliers, J. J. An image processing algorithm for measurement of flotation froth bubble size and shape distributions. *Minerals Engineering*, 10(10):??? – ???, 1997.

Schaefer, P. Maragos R. W. Morphological filters - part ii: Their relations to median, order statistic, and stack filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(8):1170 – 1184, August 1987.

Symonds, P. J. and De Jager, G. A technique for automatically segmenting images of the surface froth structures that are prevalent in industrial flotation cells. In *Proceedings of the 1992 South African Symposium on Communications and Signal Processing*, pages 111–115, University of Cape Town, Rondebosch South Africa, September 1992.

Vincent, L. and Soille, P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583 – 598, 1991.

Wang, Xin. Optimal edge-preserving hybrid filters. *IEEE Transactions on Image Processing*, 3(6):862 – 865, November 1994.

Whitaker, Ross T. and Pizer, Stephen M. A multi-scale approach to nonuniform diffusion. *Computer Vision Graphics and Image Processing: Image Understanding*, 57(1):99 – 110, January 1993.

You, Yu-Li, Tannenbaum, Allen, and Kaveh, Mostafa. Behavioral analysis of anisotropic diffusion in image processing. *IEEE Transactions on Image Processing*, 5(11):1539 – 1553, November 1996.