IN PROCEEDINGS OF THE ACIS 4TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ARTIFICIAL INTELLIGENCE, NETWORKING AND PARALLEL/DISTRIBUTED COMPUTING (SNPD'03), LÜBECK, GERMANY, OCTOBER 16–18, 2003, W. DOSCH AND R. Y. LEE, EDS., PP. 358–362.

Estimation of Multiple Motions by Block Matching

Ingo Stuke[1], Til Aach[1], Erhardt Barth[2], and Cicero Mota[1,2]
[1]Institute for Signal Processing
[2]Institute for Neuro- and Bioinformatics
University of Lübeck, Ratzeburger Allee 160, 23538 Lübeck, Germany
{stuke, aach}@isip.uni-luebeck.de, {barth, mota}@inb.uni-luebeck.de

Abstract

This paper deals with the problem of estimating multiple motions at points where these motions are overlaid. We present a new approach that is based on block matching and can deal with both transparent motions and occlusions. We derive a block matching constraint for an arbitrary number of moving layers. Such constraint comes from the theory of motion-based layer separation and can be used for estimating an arbitrary number of overlaid motions. Furthermore, we design a hierarchical algorithm that can distinguish between the occurrence of single, transparent, and occluded motions and can thus select the appropriate local motion model. Performance is demonstrated on image sequences synthesized from natural textures.

1. Introduction

Motion analysis is a key component of applications involving video compression, human and artificial vision, medical image processing and denoising, object tracking, plants growing estimation, weather forecasting etc. Accordingly, many different techniques for single motion estimation have been developed, see [7] for a review. Nevertheless, these methods fail in the case of transparency and occlusion. Transparencies can appear in daily imagery as results of looking at objects trough others, like in X-ray imagery, or as reflections on polished surfaces, for instance glass windows. In such cases we have more than only one motion at the same spatial position. Hence, the estimation of transparent motion can play a important role in the analysis of such imagery data. Different approaches for the estimation of motion vectors for the case of multiple transparent motions have been proposed [9, 3, 4, 12]. The superposition principle of Shizawa and Mase has recently been linearized and thereby solved for an arbitrary number of motions [8]. Such linearization allows the introduction of solutions that include regularization as proposed in [10]. Vernon used a phase-based approach to estimate the motion vectors for the case of only two motions [11]. His approach has also been generalized to an arbitrary number of motions [10]. Based on this generalization, we will here derive a block-matching algorithm for multiple transparent and occluded motions.

2. Theoretical considerations

2.1. The block-matching equation for N motions

In the spatial domain, we model transparent motions as a superposition of N different moving layers:

$$f_k(\boldsymbol{x}) = f(\boldsymbol{x}, k) = g_1(\boldsymbol{x} - k\boldsymbol{v}_1) + g_2(\boldsymbol{x} - k\boldsymbol{v}_2) + \cdots + g_N(\boldsymbol{x} - k\boldsymbol{v}_N). \quad (1)$$

Here, the n-th layer is moving with velocity $\boldsymbol{v}_n$. To derive an equation for block matching, we first transform the above equation to the Fourier domain:

$$F_k(\omega) = \phi_1^k G_1(\omega) + \phi_2^k G_2(\omega) + \cdots + \phi_N^k G_N(\omega) \quad (2)$$

where $\phi_n = e^{-j\omega\cdot\boldsymbol{v}_n}$, $n = 1, \ldots, N$ are the phase shifts and $\omega = (\omega_x, \omega_y)$ are the frequency variables. Uppercase letters denote the Fourier transforms of the respective lower case letters, e.g., F_k is the Fourier transform of f_k.

This relationship has been used for the estimation of only one motion by Jepson and Fleet [5]. Equation (2) has been solved by Vernon [11] for the simplest case of only two motions and in [10] to separate up to N motion layers. Here we will use (2) to obtain a block-matching equation in the spatial domain. We first simplify notation by setting $\Phi_k = (\phi_1^k, \ldots, \phi_N^k)$ and $\boldsymbol{G} = (G_1, \ldots, G_N)$ and obtain the following expression for the above system of equations:

$$F_k = \Phi_k \cdot \boldsymbol{G}. \quad (3)$$

Our goal now is the elimination of the unknown vector $\boldsymbol{G}$ that contains the Fourier-transforms of the motion layers.

The remaining equation then relates only to the observable Fourier transform of the single images and the phase shifts, i.e., $F_0, \ldots, F_N$ and $\phi_1, \ldots, \phi_N$. We proceed by defining the polynomial

$$p(z) = (z - \phi_1) \cdots (z - \phi_N) = z^N + a_1 z^{N-1} + \cdots + a_N \quad (4)$$

with unknown coefficients $a_1, \ldots, a_N$. The phase terms $\phi_1, \ldots, \phi_N$ are the roots of $p(z)$, i.e., $p(\phi_n) = 0$, for $n = 1, \ldots, N$. Since the components of Φ_k are, by definition, the roots of $p(z)$ to the k-th power, we have:

$$\Phi_N + a_1 \Phi_{N-1} + \cdots + a_N \Phi_0 = (p(\phi_1), \ldots, p(\phi_N)) = \mathbf{0}. \quad (5)$$

Therefore by inserting (5) in (3) we obtain

$$F_N + a_1 F_{N-1} + \cdots + a_N F_0 = (\Phi_N + a_1 \Phi_{N-1} + \cdots + a_N \Phi_0) \cdot \boldsymbol{G} = 0 \quad (6)$$

and consequently

$$F_N = -a_N F_0 - \cdots - a_1 F_{N-1}. \quad (7)$$

Being the coefficients of $p(z)$, the a's are, up to a sign, the symmetric functions of the roots $\phi_1, \ldots, \phi_N$:

$$\begin{aligned} a_1 &= \phi_1 + \phi_2 + \cdots + \phi_N \\ a_2 &= -\sum_{i<l} \phi_i \phi_l \\ a_3 &= \sum_{i<l<k} \phi_i \phi_l \phi_k \\ &\vdots \\ a_N &= (-1)^{N+1} \phi_1 \phi_2 \cdots \phi_N. \end{aligned}$$

Transforming Equation (7) back into the spatial domain leads to

$$f_N(\boldsymbol{x}) = (-1)^N f_0(\boldsymbol{x} - \boldsymbol{v}_1 - \cdots - \boldsymbol{v}_N) + \cdots - \sum_{i<l} f_{N-2}(\boldsymbol{x} - \boldsymbol{v}_i - \boldsymbol{v}_l) + \sum_i f_{N-1}(\boldsymbol{x} - \boldsymbol{v}_i) \quad (8)$$

because the products of phase terms lead to concatenated shifts in the spatial domain. Equation (8) describes how the image at time t_N can be constructed form the N previous images by using the motion vectors. Therefore, this equation can be used as the basis for block-matching methods for a theoretically unlimited number of motions.

2.2. Example for two motions

In case of two motions, using the notation $\boldsymbol{u} = \boldsymbol{v}_1$ and $\boldsymbol{v} = \boldsymbol{v}_2$, Equation (8) reduces to:

$$f_2(\boldsymbol{x}) = -f_0(\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{v}) + f_1(\boldsymbol{x} - \boldsymbol{u}) + f_1(\boldsymbol{x} - \boldsymbol{v}). \quad (9)$$

A block-matching algorithm can be obtained from the above equation by minimizing the following expression, which is the squared sum of differences for a given block:

$$M_2(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{|\boldsymbol{B}|} \sum_{\boldsymbol{x} \in \boldsymbol{B}} \Big(f_2(\boldsymbol{x}) + f_0(\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{v}) - f_1(\boldsymbol{x} - \boldsymbol{u}) - f_1(\boldsymbol{x} - \boldsymbol{v}) \Big)^2. \quad (10)$$

This expression has to be minimized with respect to $\boldsymbol{u}$ and $\boldsymbol{v}$. In the above equation, $\boldsymbol{B}$ is a set that defines the pixels in the block under consideration and $|\boldsymbol{B}|$ is the block size, i.e. number of elements in the set. If there is only one motion inside $\boldsymbol{B}$, i.e. $f_1(\boldsymbol{x}) = f_0(\boldsymbol{x} - \boldsymbol{v})$, the value

$$M_1(\boldsymbol{v}) = \frac{1}{|\boldsymbol{B}|} \sum_{\boldsymbol{x} \in \boldsymbol{B}} (f_1(\boldsymbol{x}) - f_0(\boldsymbol{x} - \boldsymbol{v}))^2 \quad (11)$$

will be small for the correct motion vector $\boldsymbol{v}$. On the other hand, if $\boldsymbol{B}$ includes two motions, the value M_1 will tend to be far from zero for any vector $\boldsymbol{v}$, because one vector cannot compensate for two motions. Accordingly, in case of two transparent motions, $M_2(\boldsymbol{u}, \boldsymbol{v})$ will be small if we insert the correct motion vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

2.3. Behavior at occlusions

In case of occluded motions Equation (9) is no longer valid and we will now show how it fails. We model the occlusion of the layer g_2 by the occluding layer g_1 by

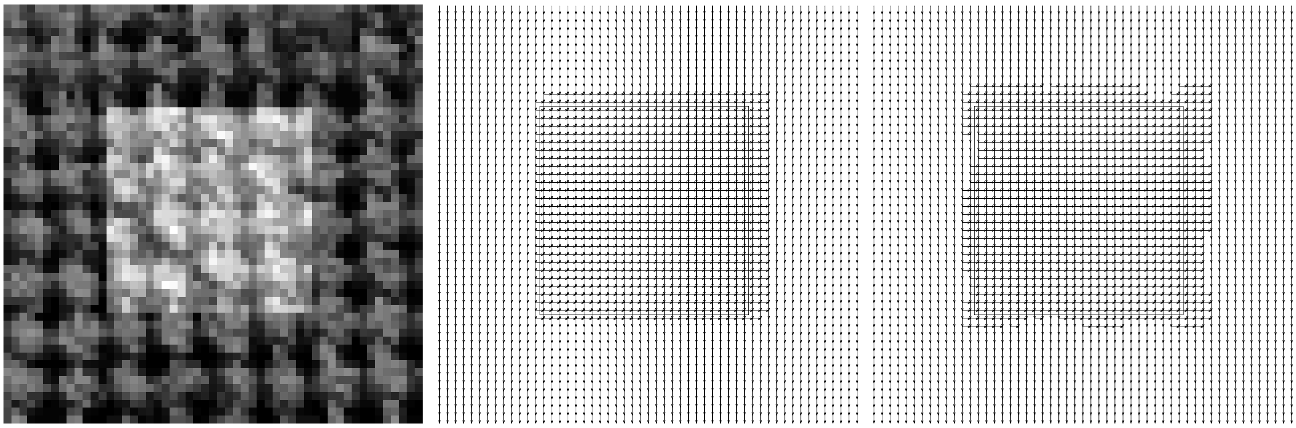
$$f_k(\boldsymbol{x}) = \chi(\boldsymbol{x} - k\boldsymbol{u}) g_1(\boldsymbol{x} - k\boldsymbol{u}) + (1 - \chi(\boldsymbol{x} - k\boldsymbol{u})) g_2(\boldsymbol{x} - k\boldsymbol{v}). \quad (12)$$

$\chi = 1$ where g_1 occludes g_2 and $\chi = 0$ otherwise [6]. By evaluating the expression in the parenthesis of Equation (10) for the above occlusion model we obtain

$$\begin{aligned} f_2(\boldsymbol{x}) + f_0(\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{v}) - f_1(\boldsymbol{x} - \boldsymbol{v}) - f_1(\boldsymbol{x} - \boldsymbol{u}) \\ = \big(\chi(\boldsymbol{x} - 2\boldsymbol{u}) - \chi(\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{v})\big) \\ \big(g_2(\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{v}) - g_2(\boldsymbol{x} - 2\boldsymbol{u})\big). \quad (13) \end{aligned}$$

Inside the block we have a region near the occluding boundary where the values are non-zero. This leads to a high value of M_2. The size of the region near the occluding boundary depends only on the difference of the velocities. In fact, by replacing $\boldsymbol{y} = \boldsymbol{x} - 2\boldsymbol{u}$ in the right-hand side of the above equation we find

$$\begin{aligned} f_2(\boldsymbol{x}) + f_0(\boldsymbol{x} - \boldsymbol{u} - \boldsymbol{v}) - f_1(\boldsymbol{x} - \boldsymbol{v}) - f_1(\boldsymbol{x} - \boldsymbol{u}) \\ = \big(\chi(\boldsymbol{y}) - \chi(\boldsymbol{y} + \boldsymbol{u} - \boldsymbol{v})\big) \\ \big(g_2(\boldsymbol{y} + \boldsymbol{u} - \boldsymbol{v}) - g_2(\boldsymbol{y})\big), \quad (14) \end{aligned}$$



(a) (b) (c)

Figure 1. Results for transparent motions. See text for details.

which means that the distortion is located on a strip, which is at most $|\boldsymbol{u} - \boldsymbol{v}|$ wide. For the simplest case of a straight-line border, the strip is $|\boldsymbol{N} \cdot (\boldsymbol{u} - \boldsymbol{v})|$ wide, where $\boldsymbol{N}$ is the unit vector normal to the border. Due to this distortion it is not guaranteed that the minimum of M_2 yields the correct motion vectors. A more formal treatment of motions at the occluding boundary is given in [2, 1].

3. Hierarchical algorithm

In order to deal with the above mentioned cases of single, transparent and occluded motions we design an hierarchical algorithm described below and summarized in Algorithm 1. An extension to more than two motions is straightforward but not given here.

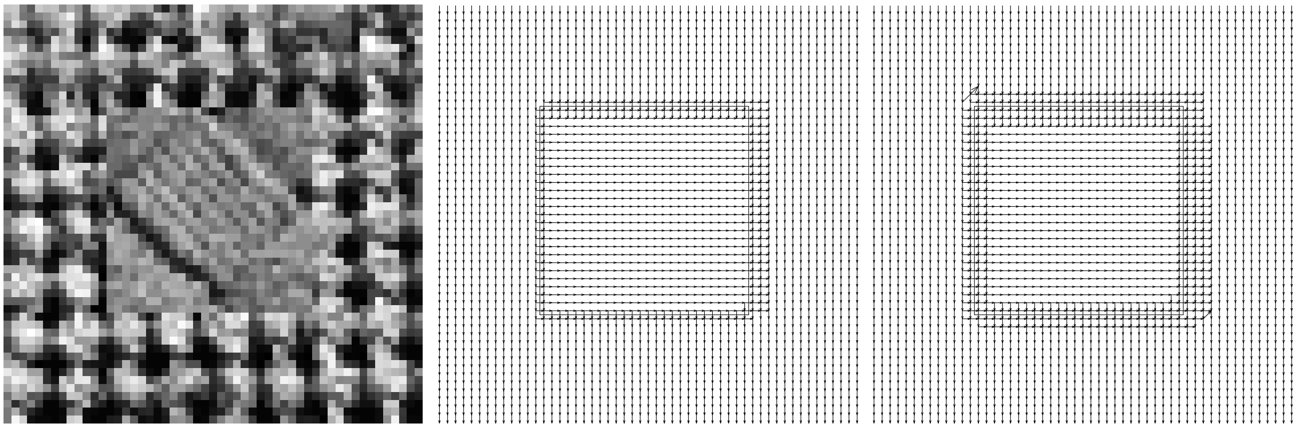
First, we estimate one motion if M_1 is smaller than a given threshold T_1. Second, we estimate two motions if M_1 is larger than the threshold T_1 and M_2 is smaller than a second threshold T_2. If at a certain position both values M_1 and M_2 exceed their thresholds, the movements do neither comply with the assumption of one or two transparent motions. In such case, this position is marked as occluded. In the second phase we determine motion vectors for the marked pixels only. We iterate the algorithm at the marked pixels L times and increase the size of the block at each iteration. The estimation of the motion vectors for the marked pixels is based on non-marked pixels only, because the marked pixels violate the assumption of one or two motions and would thus not allow to minimize either expression M_1 or M_2. The iteration can be repeated until motion vectors are found for all marked pixels or a maximum number of iterations is reached. This two-phase approach enables us to compute two motions at the occluding boundary by avoiding the terms in the right side of Equation (14).

Algorithm 1 Hierarchical algorithm

```
1:  for all pixels do
2:    Compute minimum value of M_1 and the corresponding motion vector.
3:    if M_1 ≤ T_1 then
4:      Choose single-motion model
5:    else
6:      Compute the minimum value of M_2 and the two motion vectors
7:      if M_2 ≤ T_2 then
8:        Choose model for two transparent motions
9:      else
10:       Mark pixel
11:     end if
12:   end if
13:   Increase window sizes and repeat lines 2 to 12 for all marked pixels. Ignore marked pixels inside the current window.
14: end for
```

4. Results

Image (a) of Figure 1 shows the first frame of a sequence consisting of two image layers: a square moving with velocity $\boldsymbol{u} = (1, 0)$ and a background moving with velocity $\boldsymbol{v} = (0, 1)$ pixels per frame. Both layers are textured and overlaid additively. The textures are natural, taken from the MIT VisTex database. In (b) we show the motion field estimated from up to three consecutive



(a) (b) (c)

Figure 2. Results for occluded motions. See text for details.

frames. Note that both the transparent motions in the center and the single motion of the background are correctly estimated and that the border between the two regions is sharp. For better visualization, the boundary of the square is marked by a rectangle corresponding to the position of the boundary in the first frame. In this example we used a block size of 3×3 pixels. The thresholds T_1 and T_2 were set to one. The motion field obtained after adding spatio-temporal Gaussian noise to the image sequence, at a signal-to-noise ratio of 35 dB, is shown in (c). We used a larger window of 5×5 pixels for the first phase and 9×9 pixels for the second phase of the algorithm. For both phases, $T_1 = 11$ and $T_2 = 17$. The larger window size increases both the robustness to noise and the smearing of the motion field at object boundaries.

Figure 2 shows results for the case where the moving square occludes the moving background according to Equation (12). In (a) the first frame of the sequence is depicted. Note in (b), that we obtain the correct motion vectors at the occluding boundary. Again, filter size was 3×3 pixels for the first phase and 5×5 pixels for the second phase and $T_1 = T_2 = 1$. In (c) results have been obtained for a noisy sequence (35 dB). Block size was 5×5 pixels for the first and 9×9 for the second phase, $T_1 = 11$, and $T_2 = 17$. Note the increased smear and the two outliers at the edges, which are both due to the noise. For all results presented in both figures we used the same Algorithm 1 and only one iteration in the second phase, i.e. $L = 1$.

5. Summary and conclusion

We derived a block-matching method for estimating an arbitrary number of transparent motions and we have also shown how to estimate multiple motions at occluding boundaries. To estimate N motions at the same spatial position, $N + 1$ successive image frames are needed. Moreover, we derived a hierarchical decision rule for selecting the best-fitting local-motion model. The performance of the algorithm is demonstrated on noise free and noisy sequences. The hierarchical decision requires threshold parameters, which we have so far chosen empirically. We currently develop a statistical framework that will allow to determine the thresholds by significance tests.

6. Acknowledgment

Work is supported by the *Deutsche Forschungsgemeinschaft* under Ba 1176/7-2.

7. References

[1] E. Barth, I. Stuke, T. Aach, and C. Mota. Spatio-temporal motion estimation for transparency and occlusion. In *Proc. IEEE Int. Conf. Image Processing*, volume III, pages 69–72, Barcelona, Spain, Sept. 14–17, 2003. IEEE Signal Processing Soc.

[2] E. Barth, I. Stuke, and C. Mota. Analysis of motion and curvature in image sequences. In *Proc. IEEE Southwest Symp. Image Analysis and Interpretation*, pages 206–10, Santa Fe, NM, Apr. 7–9, 2002. IEEE Computer Press.

[3] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan. 1996.

[4] T. Darrell and E. Simoncelli. Nulling filters and the separation of transparent motions. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 738–9, New York, June 14–17, 1993. IEEE Computer Press.

[5] D. J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[6] D. J. Fleet and K. Langley. Computational analysis of non-Fourier motion. *Vision Research*, 34(22):3057–79, Nov. 1994.

[7] H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook of Computer Vision and Applications*, volume 2, pages 309–96. Academic Press, 1999.

[8] C. Mota, I. Stuke, and E. Barth. Analytic solutions for multiple motions. In *Proc. IEEE Int. Conf. Image Processing*, volume II, pages 917–20, Thessaloniki, Greece, Oct. 7–10, 2001. IEEE Signal Processing Soc.

[9] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 274–8, Atlantic City, NJ, June 1990. IEEE Computer Press.

[10] I. Stuke, T. Aach, C. Mota, and E. Barth. Estimation of multiple motions: regularization and performance evaluation. In B. Vasudev, T. R. Hsing, A. G. Tescher, and T. Ebrahimi, editors, *Image and Video Communications and Processing 2003*, volume 5022 of *Proceedings of SPIE*, pages 75–86, May 2003.

[11] D. Vernon. Decoupling Fourier components of dynamic image sequences: a theory of signal separation, image segmentation and optical flow estimation. In H. Burkhardt and B. Neumann, editors, *Computer Vision - ECCV'98*, volume 1407/II of *LNCS*, pages 68–85. Springer Verlag, Jan. 1998.

[12] W. Yu, K. Daniilidis, S. Beauchemin, and G. Sommer. Detection and characterization of multiple motion points. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 171–7, Fort Collins, CO, June 23–25, 1999. IEEE Computer Press.