Temporal Symmetry Constraints in Block Matching

C. Bartels[†]

[†]Eindhoven University of Technology
Den Dolech 2 5600 MB Eindhoven
The Netherlands

G. de Haan[‡]

[‡]Philips Research Laboratories
High Tech Campus 36 5656 AE Eindhoven
The Netherlands

Abstract—Video processing applications such as picture rate conversion depend on accurate and consistent motion estimation. Most motion estimation and optical flow algorithms use some form of spatial regularization to improve consistency, but few perform a temporal regularization based on the symmetry between successive vector fields. Methods that do improve temporal symmetry are often computationally complex. This paper describes a simple implementation of symmetry constraints in a real-time block matching motion estimation algorithm and shows the effect on related occlusion detection methods that are based on forward-backward consistency checks.

I. INTRODUCTION

Accurate motion estimation (ME) is an essential ingredient in many video processing and computer vision algorithms. Consequently, it is the focal point of an active research community. Since Horn and Schunck introduced the gradient optic flow equation [1], strongly regularized optical flow implementations have been developed [2], [3]. In parallel, block correlation based ME methods have become popular in industry use, i.e. in compression [4] and picture rate conversion applications [5]. Block correlation based approaches are more robust in the handling of large motions as they do not depend on linearizations and typically feature simpler optimization mechanisms, geared towards real-time use (in embedded systems). As most of these approaches focus strongly on operations count reduction, they also feature simpler, implicit, regularization mechanisms. For example, the recursive search or spatio-temporal prediction technique [4], [6]–[8] forces vector field smoothness through a limited candidate set mechanism which at the same time reduces the average number of block correlation matches to less than 3 [7]. In this work, we show how this algorithm may profit from additional ‘temporal’ regularization, after introducing why this is relevant for occlusion/inconsistency detection.

Occlusion detection is necessary to correct for areas that the motion estimation algorithm cannot handle directly. Objects with different motions cause a set of pixels to (dis)appear between two consecutive pictures, due to overlap. This means that the occluded pixels ‘exist’ in only one of the two pictures. As most ME algorithms rely on the one-to-one correspondence, or the existence of the same pixels in both pictures, the occlusion areas cause unreliable displacement estimates. Occlusion or inconsistency detection methods are necessary to label these regions for special handling in subsequent algorithms.

A straightforward approach to occlusion detection is based on a comparison of the ‘forward’ vector field with the ‘backward’ vector field: where the two vector fields do not point to each other, the area is labeled as occluded, as illustrated in Figure 1. This approach has been termed left-right consistency check (LRC) for a stereo (disparity estimation) setting [9] and typically suffers from over-classification as the two vector fields are estimated separately and may contain different vectors for the same object. This particularly occurs in areas where the motion is ambiguous due to the aperture-problem or lack of texture. Recently, for optical flow methods, symmetry constraints have been introduced [2], [3], [10] to bias the estimation towards symmetrical solutions in non-occluded areas. In this paper, we look at an implementation of symmetry constraints in a recursive search block matcher, to benefit a real-time system. We further analyze the trade-offs in symmetry and global match-quality for different constraint implementations.

Fig. 1. (a) Two consecutive pictures represented in 1 dimension with 2 different object motions. (b) Symmetric and non-symmetric motion estimates (within an object).

II. SYMMETRIC ESTIMATION

Let us use a bold font to denote vectors, e.g. $\mathbf{x} = (x, y)^T$ and $\mathbf{v} = (v_x, v_y)^T$. Let I_n denote the luminance pixel lattice at the temporal position of picture n , and $\mathbf{D}_{n \rightarrow n+1}$ the vector field that contains the displacement vectors estimated from picture n to $n + 1$. E.g. $I_n(\mathbf{x})$ denotes the luminance value of a pixel at position $\mathbf{x}$, and $\mathbf{D}_{n \rightarrow n+1}(\mathbf{x})$ the forward motion vector. We define $\mathbf{D}_{n+1 \rightarrow n}$ the backward (opposite) motion estimate, the temporal positions and the symmetrical estimation are illustrated in Figure 1. For a motion vector at position $\mathbf{x}$ in picture n the measure of symmetry can be

Fig. 2. The motion vector for the currently processed block is selected from a candidate set. The candidate set includes the spatial (S1, S2) and temporal (T1) prediction vectors. Blocks are processed sequentially, the dotted arrow indicates the current (downward) scanning direction. For the non-symmetric recursive search algorithm, one iteration, consisting of an upward and a downward scan over all blocks, typically suffices.

defined as the Euclidean distance between the vector itself and a motion compensated counterpart in the opposite vector field: $\|D_{n \rightarrow n+1}(\mathbf{x}) - D_{n+1 \rightarrow n}(\mathbf{x} + D_{n \rightarrow n+1}(\mathbf{x}))\|$.

III. SYMMETRIC RECURSIVE SEARCH IMPLEMENTATION

The symmetric recursive search ME algorithm determines the motion vector for a block based on a set of candidate motion vectors C from the spatio-temporal neighborhood [6]. The evaluation is based on a function that contains a match term E_M (block correlation value), a spatial smoothness constraint E_S (distance of the candidate vector to neighboring block vectors) and a temporal consistency constraint E_T (distance of the candidate vector to the vector it points to in the opposite vector field). The candidate motion vector $\mathbf{v} \in C$ that minimizes this function is assigned to the current block and determines a part of the candidate set for the following blocks. In the following we illustrate an iteration of the algorithm in the estimation of the forward vector field $D_{n \rightarrow n+1}$. The estimation of the backward vector field $D_{n+1 \rightarrow n}$ can be described symmetrically analogous.

For each processed block, the algorithm selects the motion vector from a candidate set. The candidate set is determined from the spatio-temporal neighborhood; it contains, predominantly, predicted motion vectors of neighboring blocks. Let the candidate set for a block $\mathbf{x}$ be denoted by the union of the sets $C_{\text{spat}}(\mathbf{x})$, $C_{\text{temp}}(\mathbf{x})$ and $C_{\text{upd}}(\mathbf{x})$:

$$C(\mathbf{x}) = \left\{ \begin{array}{l} C_{\text{spat}}(\mathbf{x}), \\ C_{\text{temp}}(\mathbf{x}), \\ C_{\text{upd}}(\mathbf{x}) \end{array} \right\} \quad (1)$$

C_{spat} is created by selecting motion vectors from neighboring blocks in the current vector field (e.g. S1, S2 in Fig. 2). As these candidates are only available in blocks where the current scan has passed, an additional ‘temporal’ candidate set C_{temp} is used. C_{temp} consists of motion vector candidates from the previous vector field or previous scans on the current vector field, e.g. T1 in Fig. 2. Update candidates are ‘new’ vectors to be tested, created by the addition of Gaussian noise to the

spatial and temporal candidates:

$$C_{\text{upd}}(\mathbf{x}) = \left\{ \mathbf{x} + \begin{pmatrix} n_x \\ n_y \end{pmatrix} \mid \mathbf{x} \in \{C_{\text{spat}}(\mathbf{x}), C_{\text{temp}}(\mathbf{x})\}, \begin{array}{l} n_x, n_y \sim N(0, \sigma^2) \end{array} \right\} \quad (2)$$

where N denotes the normal distribution with mean 0 and variance σ^2 .

The motion vector that is assigned to the currently processed block is the one that minimizes the following energy term, consisting of the match criterion, the spatial smoothness constraint and the temporal consistency constraint:

$$D_{n \rightarrow n+1}(\mathbf{x}) = \arg \min_{\mathbf{v} \in C(\mathbf{x})} (E_M(\mathbf{v}, \mathbf{x}) + E_S(\mathbf{v}, \mathbf{x}) + E_T(\mathbf{v}, \mathbf{x})) \quad (3)$$

The match criterion is based on the sum of absolute differences (SAD) for all pixels in a block:

$$E_M(\mathbf{v}, \mathbf{x}) = \sum_{\mathbf{p} \in R(\mathbf{x})} |(I_n(\mathbf{p}) - I_{n+1}(\mathbf{p} + \mathbf{v}))| \quad (4)$$

where $R(\mathbf{x})$ denotes the set of pixel positions in a block at location $\mathbf{x}$.

The smoothness constraint is based on the Euclidean distance of the candidate motion vector with the predicted motion vectors of spatially neighboring blocks:

$$E_S(\mathbf{v}, \mathbf{x}) = \lambda \sum_{\mathbf{p} \in N(\mathbf{x})} \|\mathbf{v} - D_{n \rightarrow n+1}(\mathbf{p})\| \quad (5)$$

where $N(\mathbf{x})$ denotes the set of positions of spatially neighboring blocks and λ a weighting term to set the relative strength of the smoothness constraint.

The symmetry term penalizes the difference between a motion vector and its motion compensated counterpart in the opposite vector field:

$$E_T(\mathbf{v}, \mathbf{x}) = \gamma \|\mathbf{v} - D_{n+1 \rightarrow n}(\mathbf{x} + \mathbf{v})\| \quad (6)$$

where γ denotes a weighting factor. For $\gamma = 0$ the system reduces to a non-symmetric recursive search.

The motion estimator relies on the balanced strength of the match term (E_M) and constraint terms (E_S , E_T) to preserve edges in the output vector fields. If E_S and E_T have a too large

Fig. 3. Occlusion areas and error propagation. (a) Estimation based on a match and/or smoothness term produces unreliable motion vectors in occlusion areas. (b) Too strong symmetry constraints can propagate the errors outside the occlusion areas.

Fig. 4. (Left) M2SE versus total symmetric difference for varying γ and number of iterations. (Right) Threshold value versus inconsistency area (relative to total image size) for different γ values with 2 iterations. Both plots show averages over 40 sequences.

relative influence in the energy minimization (steered with γ and λ) the output vector fields are overly smoothed. If the energy terms have too little influence, they do not achieve the desired filtering effects. In the following section we evaluate this trade-off for varying γ .

The symmetric algorithm further differs from the non-symmetric in the optimization strategy: the symmetry term depends on the availability of a counterpart vector field. For the non-symmetric algorithm, a single spatial optimization of $\mathbf{D}_{n \rightarrow n+1}$ is sufficient. For the symmetric algorithm it is replaced by alternating optimizations of $\mathbf{D}_{n \rightarrow n+1}$ and $\mathbf{D}_{n+1 \rightarrow n}$. This allows for the symmetry term to effect both vector fields and propagate changes.

For completeness, we note that match and symmetry constraints do not hold in occlusion areas, as shown in Figure 3a. Strong symmetry constraints can propagate errors from the occlusion area into the opposite vector field's non-occluded areas, Fig. 3b. It is therefore important to limit the relative strength of the symmetry term. Alternatively, one can use occlusion detection and correction mechanisms, which is outside the scope of this paper.

IV. EVALUATION CRITERIA AND RESULTS

To establish the optimal value for γ we evaluate the vector field quality using a combination of metrics. The M2SE metric [6] is used to establish the correlation quality of the vector field. The SYM metric is used to establish the total symmetric difference between two vector fields. To find an 'optimal' motion estimator we try to minimize the SYM value, without raising the M2SE score (this indicates a lower vector field quality, most likely caused by an overly constrained motion estimation).

The M2SE metric, Eq. 7, tests the correlation quality using the estimated motion between I_n and I_{n+1} , the vector field $\mathbf{D}_{n \rightarrow n+1}$, to create a motion compensated image using I_{n-1} and I_{n+1} at temporal position n . Essentially $\mathbf{D}_{n \rightarrow n-1}$ is created by mirroring $\mathbf{D}_{n \rightarrow n+1}$ and both are used to create the motion

compensated image I_{mc} . The mean square error is determined between the motion compensated image I_{mc} and the original image I_n , this 'temporal extension' tests if the motion vectors align with the true object motion. The M2SE is denoted:

$$\text{M2SE}(\mathbf{D}_{n \rightarrow n+1}) = \frac{1}{w \times h} \sum_{\mathbf{x} \in I} I_n(\mathbf{x}) - I_{mc}(\mathbf{x}) \quad (7)$$

$$I_{mc}(\mathbf{x}) = \frac{1}{2} I_{n-1}(\mathbf{x} - \mathbf{D}_{n \rightarrow n+1}(\mathbf{x})) + \frac{1}{2} I_{n+1}(\mathbf{x} + \mathbf{D}_{n \rightarrow n+1}(\mathbf{x})) \quad (8)$$

where w and h denote the width and height of the pictures.

The total symmetric distance is the sum of all left-right distances:

$$\text{SYM}(\mathbf{D}_{n \rightarrow n+1}, \mathbf{D}_{n+1 \rightarrow n}) = \sum_{\mathbf{x} \in \mathbf{D}} \|\mathbf{D}_{n \rightarrow n+1}(\mathbf{x}) - \mathbf{D}_{n+1 \rightarrow n}(\mathbf{x} + \mathbf{D}_{n \rightarrow n+1}(\mathbf{x}))\| \quad (9)$$

We use a non-symmetric motion estimation pass to initialize both vector fields $\mathbf{D}_{n \rightarrow n+1}$, $\mathbf{D}_{n+1 \rightarrow n}$. After the initialization, multiple iterations of the symmetric motion estimation algorithm are evaluated. The results have been plotted in Figure 4 (left), each point visualizes a particular parameter setting of the motion estimator, where the number of iterations and the symmetry gain factor γ are varied. The gain factor for the smoothness energy is kept constant ($\lambda = 6000$). Values near the lower left corner of the graph represent better results in symmetry and M2SE correlation quality.

For increasing gain factors the figures show a decrease in SYM value up to a point where the symmetry constraint becomes too dominant: the SYM value does not improve further and the M2SE correlation value sharply increases. The estimation is 'optimal' around $\gamma = 5000$. A larger number of iterations improves the estimation result by a limited margin. However, for most implementations the limited gain will not outweigh the cost of additional iterations.

To evaluate the effect on the detected inconsistency we

create a map of the thresholded symmetric difference between the vector fields, the forward-backward-consistency (FBC). The set of forward inconsistent/occluded pixels for a picture n is defined as:

$$O_{n \rightarrow n+1} = \{\mathbf{x} \mid \|\mathbf{D}_{n \rightarrow n+1}(\mathbf{x}) - \mathbf{D}_{n+1 \rightarrow n}(\mathbf{x} + \mathbf{D}_{n \rightarrow n+1}(\mathbf{x}))\| > \delta\} \quad (10)$$

Where δ denotes a threshold value. The optimum threshold value depends on the motion estimation algorithm used: if the vector fields are more symmetric, a lower threshold value can be used to create the binary map. We therefore evaluate the threshold values for different symmetry gain factors γ . The relative detected FBC occlusion area versus the threshold values (averaged over a large set of sequences) is plotted in Figure 4 (right). Example occlusion maps are illustrated in Figures 5 and 6. As visible in the binary occlusion maps, the selection of a threshold value is difficult. For low threshold values many small variations between the vector fields are incorrectly classified as occluded. For high threshold values one may lose real occluded objects with small motion differences. The symmetric constraints have a clear effect: the curves in Figure 4 get a stronger corner definition for increasing symmetry gain values γ , which helps in the selection of a threshold value. The plot further shows a reduction of approximately 50% in detected area, for a system with low gain factor $\gamma = 4000$ which does not negatively affect the M2SE performance. The reduction in detected area versus the threshold value is also clearly visible in the example maps, Fig. 6.

V. CONCLUSION

We have shown that symmetry constraints can be applied in a recursive search block matching scheme at relative low-cost. The developed algorithm performs well with respect to the symmetrically inconsistent area (an approximate 50% reduction), without compromising the overall vector field quality. The addition of a symmetry energy function requires that the recursive spatial optimization of $\mathbf{D}_{n \rightarrow n+1}$ is extended to alternating optimizations of $\mathbf{D}_{n \rightarrow n+1}$ and $\mathbf{D}_{n+1 \rightarrow n}$. In general, multiple iterations can make the algorithm expensive. However, our results show that two iterations suffice in practice.

REFERENCES

- [1] B. Horn and B. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [2] L. Alvarez, R. Deriche, T. Papadopoulos, and J. Sánchez, "Symmetrical Dense Optical Flow Estimation with Occlusions Detection," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 371–385, 2007.
- [3] S. Ince and J. Konrad, "Occlusion-aware optical flow estimation," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1443–1451, 2008.
- [4] G. Lee, M. Wang, H. Lin, D. Su, and B. Lin, "Algorithm/Architecture Co-Design of 3-D Spatio-Temporal Motion Estimation for Video Coding," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 455–465, 2007.
- [5] E. B. Bellers, J. G. W. M. Janssen, and M. Penners, "Motion compensated frame rate conversion for motion blur reduction," *SID Digest of Technical Papers*, vol. 38, no. 1, pp. 1454–1457, 2007.

Fig. 5. Forward occlusion detection on skatgirl sequence without symmetry constraints for various threshold values δ .

Fig. 6. Forward occlusion detection on skatgirl sequence with symmetry constraints (2 iterations, $\gamma = 4000$) for various threshold values δ .

- [6] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, "True-motion estimation with 3-d recursive search block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368–379, 1993.
- [7] N. Atzpadin, P. Kauff, and O. Schreer, "Stereo analysis by hybrid recursive matching for real-time immersive video conferencing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 3, pp. 321–334, 2004.
- [8] A. Tourapis, O. Au, and M. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 934–947, 2002.
- [9] G. Egnal and R. Wildes, "Detecting binocular half-occlusions: empirical comparisons of five approaches," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1127–1133, 2002.
- [10] L. Alvarez, R. Deriche, T. Papadopoulos, and J. Sánchez, "Symmetrical Dense Optical Flow Estimation with Occlusions Detection," *Proceedings European Conference on Computer Vision*, pp. 721–735, 2002.