

# A Two-Stage Algorithm of High Resolution Image Alignment for Mobile Applications

Ren-You Huang<sup>1</sup>, Lan-Rong Dung<sup>2</sup>, Tang-Suan Hong<sup>3</sup>

<sup>1</sup>Institute of Electrical Control Engineering, National Chiao Tung University, Taiwan

<sup>2</sup>Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

<sup>3</sup>Institute of Communications Engineering, National Chiao Tung University, Taiwan

Email: hry76519@gmail.com, lennon@faculty.nctu.edu.tw, tom771006@gmail.com

Received 1 February 2016; accepted 26 March 2016; published 29 March 2016

Copyright © 2016 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Global motion estimation (GME) algorithms are widely applied to computer vision and video processing. In the previous works, the image resolutions are usually low for the real-time requirement (e.g. video stabilization). However, in some mobile devices applications (e.g. image sequence panoramic stitching), the high resolution is necessary to obtain satisfactory quality of panoramic image. However, the computational cost will become too expensive to be suitable for the low power consumption requirement of mobile device. The full search algorithm can obtain the global minimum with extremely computational cost, while the typical fast algorithms may suffer from the local minimum problem. This paper proposed a fast algorithm to deal with  $2560 \times 1920$  high-resolution (HR) image sequences. The proposed method estimates the motion vector by a two-level coarse-to-fine scheme which only exploits sparse reference blocks (25 blocks in this paper) in each level to determine the global motion vector, thus the computational costs are significantly decreased. In order to increase the effective search range and robustness, the predictive motion vector (PMV) technique is used in this work. By the comparisons of computational complexity, the proposed algorithm costs less addition operations than the typical Three-Step Search algorithm (TSS) for estimating the global motion of the HR images without the local minimum problem. The quantitative evaluations show that our method is comparable to the full search algorithm (FSA) which is considered to be the golden baseline.

## Keywords

Global Motion Estimation, Block Matching, High Resolution Image Alignment, Mobile Applications

---

## 1. Introduction

Global motion estimation (GME) had been widely applied to video processing and computer vision in decades.

**How to cite this paper:** Huang, R.-Y., Dung, L.-R. and Hong, T.-S. (2016) A Two-Stage Algorithm of High Resolution Image Alignment for Mobile Applications. *Journal of Computer and Communications*, 4, 36-51.

<http://dx.doi.org/10.4236/jcc.2016.44004>

For example, video stabilization, motion compensation, and the popular image panoramic stitching which can make a photograph with wide field of view. There are two major tasks to stitch a sequence of images: 1) find the global motions of the images with respect to the previous ones and then 2) stitch the sequence to produce a panoramic image. After all the images have been aligned, there are various algorithms to stitch the images. For instance, one may refer to the efficient ways which are to find the optimal seams [1]-[4], or the more complicated ones [5] [6]. This study will focus on the global motion estimation, thus the discussions about the stitching algorithms are beyond the scope of this paper. If the readers are interested in the stitching algorithms, please refer to the related works.

Global motion estimation algorithms can be classified into two categories: direct methods [7]-[18] and feature-based methods [19]-[24]. The direct methods aim to obtain a global motion through global minimization of some cost functions by using the image pixels directly. On the contrary, the feature-based methods first locate sparse set of keypoints in the images and then obtain the global motion parameters by matching the feature correspondences. In this paper, we focus on the direct methods since the feature-based methods are computational expensive in feature description and feature matching, which makes feature-based methods unsuitable for the low power consumption requirement in mobile devices.

The direct methods can be classified into two subgroups: full search kind algorithms [7]-[12] and fast algorithms [13]-[18]. The original full search algorithm compares all the positions in the search windows, which makes the computations extremely expensive. There are some accelerated versions of full search algorithm, e.g. projection-based [9] [10], or skipping some checking points by some criteria of mathematical inequality [11] [12]. Unfortunately, the computations still significantly increase as the search range and block size increase. On the contrary, the fast algorithms such as three-step search (TSS) [13] [14], four-step search [15], special patterns (diamond, hexagon, etc.) search [16] [17], and gradient-descent [18], are well-known for their quick convergence property. However, the fast algorithms usually suffer from the local minimum problem. For the applications of panoramic stitching, especially when the user captures the sequence with large motions, misalignments may produce apparently discontinuities on the seam lines. Therefore, the fast algorithms are not suitable for the panoramic stitching purposes.

Another issue is power consumption for mobile devices. The power consumption will be reduced if 1) the computation costs and 2) the memory access are as low as possible. Therefore, we aimed at developing an algorithm which is fast and low-memory-access while keeping the accuracy as comparable to the full search algorithm as possible.

The remainder of this paper is organized as follows. Section 2 gives the brief review of related works. The overall details of the proposed algorithm are described in Section 3. The comparisons of computational complexity are presented in Section 4. Accuracy verifications are presented in Section 5. Finally, conclusions are summarized in Section 6.

## 2. Related Works

The full search kind algorithms usually consider all the positions in a search window and determine the motion vectors by minimizing some cost functions. The traditional full search algorithm (FSA) computes the sum of absolute differences (SAD) between a reference block and candidate blocks by block matching to determine the motion vector of the reference block. The FSA is able to find a global minimum in search window while the computational cost is extremely high. There are some accelerated versions of FSA: Tu *et al.* [9] proposed the projection method to accelerate the SAD computations of block matching; Puglisi *et al.* [10] proposed an modified version of projection method [9], which is more efficient since only the candidate blocks satisfied the conditions are further used to compute the SAD; the other kind of accelerated version is to skip the computation of SAD by some mathematical inequality [11] [12]. Although these accelerated versions significantly reduced the computational costs in contrast to original FSA, the computational costs still greatly increase as the search range and block size increase with image resolution.

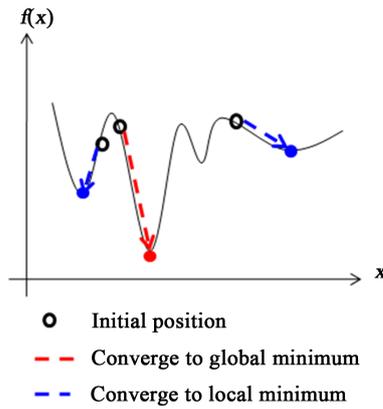
On the other hand, the fast algorithms assume the energy functions (e.g. SAD) are unimodal and the optimal solution is quickly converge to the minimum using different optimization algorithms. The typical fast algorithms, for example, three-step search (TSS) algorithms [13] [14], four-step search (4SS) [15], special patterns (e.g., diamond, hexagon) search [16] [17], and gradient-descent [18], are well-known for their quick convergence property. The TSS algorithm determines motion vector for one block in three iterations, and the computation of

SAD are significantly less than FSA. The 4SS algorithm is a modified version of TSS which is more robust and accurate than TSS. The special pattern search algorithms are also well-known for their fast convergence and compatible peak signal-to-noise ratio (PSNR) quality in video coding. However, the real world scenes are complex and the SAD is not unimodal in general, hence there might be local minimum problem. **Figure 1** shows the illustration of local minimum problem for 1-D case. The optimal solution may converge to local minimum instead of global minimum if the initial position is close to the local minimum.

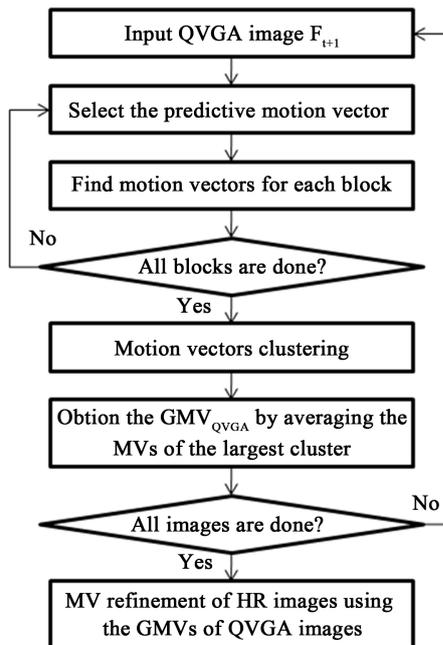
### 3. The Proposed Algorithm

The proposed algorithm is a two-level scheme which processes the QVGA versions of the original HR images followed by a motion vector refinement in HR images. Although the down-sampled image will lose some detail informations, the major structures and edges in the image are usually preserved. The QVGA global motion multiplied by 8 will approach to the HR global motion, hence we only need to search for small range in HR domain to refine the HR global motion vector. This significantly reduce the computation costs in HR full search.

The overall flow chart is shown in **Figure 2**. We first down-sample all the HR images to obtained the QVGA ones, and only select  $N_{QVGA}$  reference blocks uniformly distributed in the center of image in order to reduce the



**Figure 1.** The illustration of local minimum problem for 1-D case.



**Figure 2.** The overview of the proposed algorithm.

computational costs. Every block is predicted by the global motion vector of previous image or the local motion vector of the processed neighbor block, in order to extend the effective search range. Once the motion vectors of all blocks are obtained, we perform the simplest clustering algorithm to reject the “outliers” of motion vectors and obtain the global motion vector which is equal to the center of the largest cluster. Finally, we apply the global motion vectors of QVGA images to predict the motion of HR images and refine the motion vectors of  $N_{HR}$  blocks in a small search range followed by the same motion vectors clustering to obtain the final HR global motion vectors of all images.

In the following subsections, we will describe all the details of the proposed algorithm. Section 3.1 first briefly reviews the traditional full search and then describes the global motion estimation in the QVGA resolution, including the details of the predictive motion estimation and clustering algorithm. Section 3.2 describes the HR global motion refinement.

### 3.1. QVGA (320 × 240) Global Motion Estimation

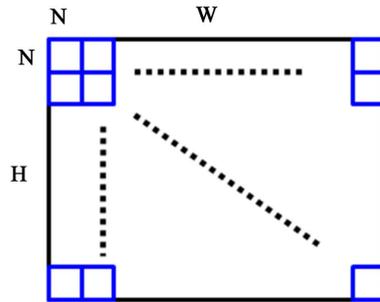
#### 3.1.1. Reference Blocks Reduction

The block-based motion estimation algorithms usually divide the image of size  $H \times W$  into  $N \times N$  pixels sub-blocks, as shown in **Figure 3**. The full search algorithm computes the sum of absolute differences (SAD) between reference image  $I_{t+1}$  and all the candidate blocks within a search window in the previous image  $I_t$ , the local motion of one reference block is determined by the criteria of minimizing the SAD:

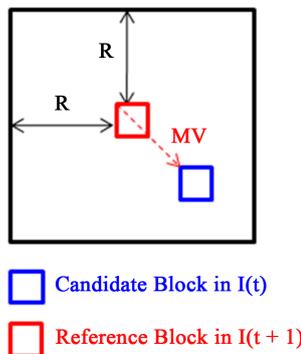
$$MV = \arg \min_{(u,v)} \left\{ \sum_{i,j} \left( |I_{t+1}(i,j) - I_t(i+u, j+v)| \right) \right\} \quad (1)$$

After all the local motion vectors of reference blocks are estimated, the global motion vector is estimated by averaging all the local motion vectors or by some statistical method, e.g. histogram, least-square, etc.

**Figure 4** shows the full search algorithm within a  $(2R+1) \times (2R+1)$  search window. Assume the total number of blocks is  $N_B$ , the number of addition operations for a global motion estimation is



**Figure 3.** The traditional block based motion estimation algorithms divide the image into subblocks with size of  $N \times N$ .



**Figure 4.** The full search algorithm compares all the positions in the search window to determine the motion vector.

$$N_B (2R+1)^2 2N^2 \quad (2)$$

For a QVGA( $320 \times 240$ ) image, suppose  $N = 16$ , then there are  $N_B = 300$  reference blocks for global motion estimation. In fact, the reference blocks in neighbor usually have similar motion vectors, hence only sparse reference blocks are needed. The reference blocks should have the following properties:

- **Repeatability:** The reference blocks in  $I_{t+1}$  are supposed to find their true correspondences in  $I_t$ . This means that the blocks near the four boundaries of image are not suitable.
- **Independence:** The reference blocks in image  $I_{t+1}$  should be as uncorrelated to each other as possible, hence the estimated global motion will approach to the true camera motion rather than the motions of moving objects.

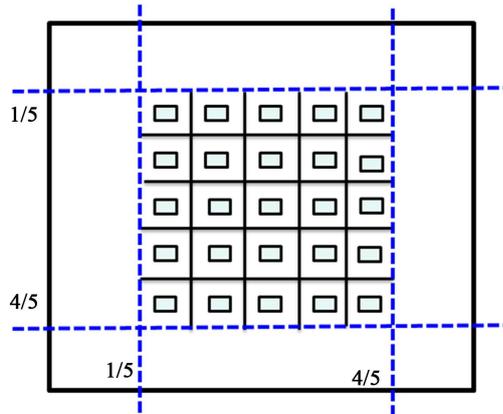
Taking the above properties into considerations, we uniformly take  $N_{QVGA} = 25$  sample blocks in the center of image, as shown in **Figure 5**. We only select the samples from the center of image with size  $(0.6H) \times (0.6W)$ . The central part of image is uniformly divided into  $5 \times 5$  large blocks, then the reference blocks of size  $N \times N$  are sampled from the center of these large blocks. This sampling scheme significantly reduces the number of reference blocks. The comparisons of the accuracy between the proposed method and the traditional full search algorithm are shown in Section 4.

Although the number of reference blocks  $N_B$  is reduced to 25, the computational costs significantly increase as the search range  $R$  and block size  $N$  increase. The block size  $N$  can not be small since the reference blocks have to contain representative and enough informations. In this paper, we set  $N = 16$  for QVGA global motion estimation, which is a common size in many related works. For the sequence panoramic stitching purposes, users may move the camera in large motions between two successive images, hence the search range  $R$  should be as large as possible. However, according to (2), if  $R$  increases by a factor of  $M$ , the total number of addition operations will increase by a factor of  $M^2$ . Hence we need a solution to increase the search range while keeping the computation costs unchanged. This paper utilized the idea of predictive motion vector (PMV) which is widely applied to video coding [25].

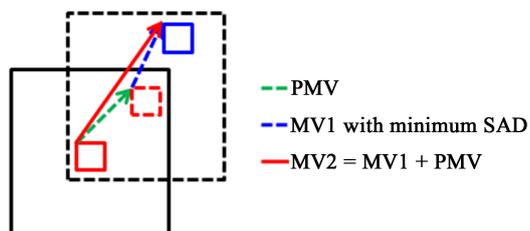
### 3.1.2. Predictive Motion Estimation

Intra/inter frame motion prediction has been widely used in video coding for local motion vector predictions. The intra-frame prediction provides possible motion to adjacent reference blocks hence the blocks can find the global minimum beyond the search range  $R$ , *i.e.* the search range is effectively “extended” without any additional computation. **Figure 6** shows the illustration of intra-frame motion prediction. The inter-frame prediction assumes the motions of camera between two successive frames (or photoshots) are similar, hence the motion of previous frame can be exploited to predict the possible motion of current frame. **Figure 7** shows the inter-frame motion prediction.

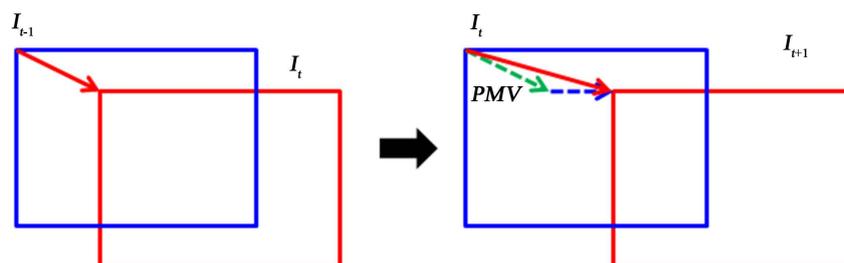
**Figure 8** gives an example for a pair of images in an outdoor sequence. In this example, the global motion vector is greater than the search range in horizontal direction. It is clear that the global motion vector estimated with prediction is more reliable since there is exactly a dominant motion in the histogram.



**Figure 5.** The proposed reference blocks selection scheme.



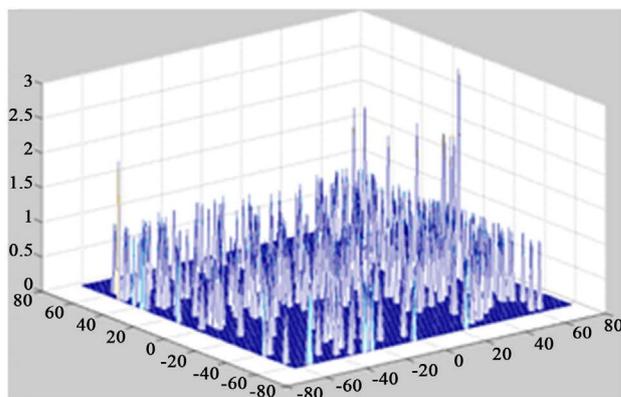
**Figure 6.** Intra-frame motion prediction.



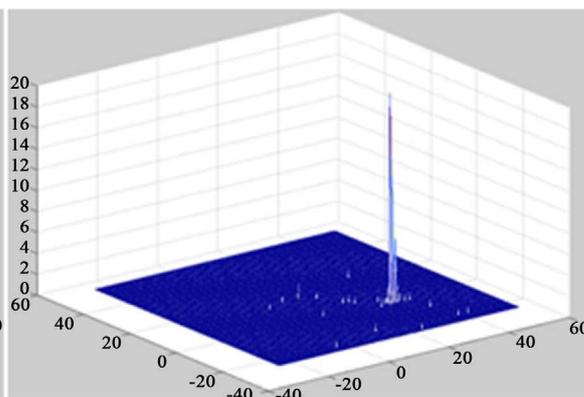
**Figure 7.** Inter-frame motion prediction.



(a)



(b)



(c)

**Figure 8.** An example for a pair of outdoor images. (a) The original images; (b) The histogram of 300 motion vectors computed by full search without prediction; (c) The histogram of 300 motion vectors computed by full search with prediction.

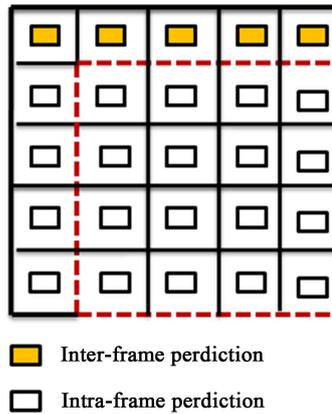
In the proposed algorithm, the the reference blocks of the first row apply the inter-frame prediction and the rest of rows apply the intra-frame prediction, as shown in **Figure 9**. In this paper, we proposed a PMV selection scheme for the intra-frame prediction, as shown in **Figure 10**. We only consider three processed blocks adjacent to the current block since the neighborhood blocks give reliable prediction. In order to increase robustness, the PMV is estimated by choosing the median of the neighbor MV:

$$\begin{cases} PMV_x = \text{median} \{MV1_x, MV2_x, MV3_x\} \\ PMV_y = \text{median} \{MV1_y, MV2_y, MV3_y\} \end{cases} \quad (3)$$

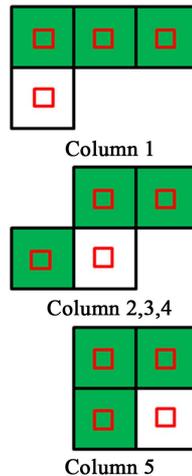
The motion vectors in the first row blocks and first column blocks are less precision since the blocks select PMV from GMV of previous frame or the motion vectors of blocks which are not adjacent to them. Considering this problem, the motion vectors of blocks in the first row and first column are only for intra-frame prediction purposes, hence only the rest of 16 motion vectors are selected for further processing.

The simplest way to estimate the GMV is to averaging all the motion vectors of reference blocks, *i.e.*,  $GMV = \frac{1}{N_B} \sum MV_i$ . However, averaged motion vector suffered from the “outliers” problem since all motion

vectors have equivalent weights  $\left(\frac{1}{N_B}\right)$ . The outliers values will significantly influence the averaged value



**Figure 9.** PMV selection for all the blocks. The reference blocks in the 1st row are predicted by the inter-frame prediction. The blocks in the rest of rows are predicted by the intra-frame prediction.

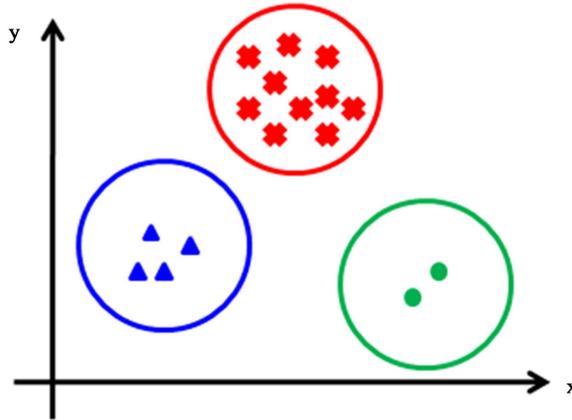


**Figure 10.** The proposed intra-frame PMV selection scheme. The motion vectors of green blocks are exploited to predict the motion of white block.

especially when the number of blocks is small. To alleviate this problem, we applied a simple clustering algorithm to automatically cluster the motion vectors into several (at least one) clusters, and the final global motion vector of a QVGA image is obtained by averaging all the motion vectors in the largest cluster.

### 3.1.3. Threshold-Order-Dependent Clustering

The threshold-order-dependent (TOD) clustering [26] is the simplest algorithm in data clustering. The clustering result depends on the order of input data and the distance threshold  $\theta$  (*i.e.* radius of a cluster). **Figure 11** shows an example of 2D data clustering. The only one parameter is the distance threshold, which controls the final number of clusters. If the threshold is small, there might be a lot of isolated clusters; on the contrary, if the threshold is large, the largest cluster probably contains the “outliers” which decrease the precision in our global motion estimation. In this paper, we set  $\theta = 5$  by experience. The details of TOD algorithm are shown in **Algorithm 1**.



**Figure 11.** An example of clustering.

---

#### Algorithm 1 TOD Clustering Algorithm

---

**Require:**  $\mathbf{x}$ : input data;  $N$ : number of data;  $\theta$ : distance threshold;

**Ensure:**  $\mathbf{c}$ : centers of clusters;  $N_c$ : number of clusters;  $\mathbf{m}$ : data numbers of clusters;

```

1: initial  $c_1 = x_1$ ,  $N_c = 1$ ,  $m_1 = 1$ ;
2: for  $i = 2$  to  $N$  do
3:    $D_{min} = \infty$ ;
4:   //Find the minimum distance between  $x_i$  and  $c_k$ 
5:   for  $k = 1$  to  $N_c$  do
6:     compute  $D(i, k) = \|x_i - c_k\|_2$ ;
7:     if  $D(i, k) < D_{min}$  then
8:        $k_0 = k$ ;
9:        $D_{min} = D(i, k)$ ;
10:    end if
11:  end for
12:  if  $D_{min} < \theta$  then
13:    //Update  $k_0$ -th cluster
14:     $c_{k_0} = (c_{k_0} * m_{k_0} + x_i) / (m_{k_0} + 1)$ ;
15:     $m_{k_0} = m_{k_0} + 1$ ;
16:  else
17:    //Create a new cluster centered by  $x_i$ 
18:     $N_c = N_c + 1$ ;
19:     $c_{N_c} = x_i$ ;
20:     $m_{N_c} = 1$ ;
21:  end if
22: end for

```

---

After motion clustering, the global motion vector is equal to the center of the largest cluster, for example, the red cluster shown in **Figure 11**. By motion vectors clustering, the outliers problem is avoided. Note that the computational complexity of TOD is  $O(N^2)$ , however, the number of data (motion vectors) in this paper is only 16, hence the computation cost of TOD is negligible (compared with the computation cost of motion estimation).

### 3.2. Global Motion Refinement for HR(2560 × 1920) Images

After the GMV of QVGA image is obtained, we only need to refine the global motion of HR image by exploiting the fact that the image contents are similar to the QVGA version. The division of image is the same as the QVGA version. We multiply the GMV of QVGA by 8 as the PMV of HR image, and simply perform the traditional full search in a small search range. Note that the search range in HR domain is reduced to  $R = 16$ , and the size  $N$  of reference blocks is multiplied by 8, *i.e.*  $N = 128$ . We only compute the motion vectors of  $N_{HR} = 16$  reference blocks corresponding to the 16 blocks in QVGA image, *i.e.*, we ignored the unreliable motion vectors of the first row blocks and first column blocks. Finally, the refined global motion vector is determined by TOD clustering of the motion vectors of the 16 blocks. **Figure 12** shows the HR global motion vector refinement. The proposed algorithm is completely described in **Algorithm 2**.

---

**Algorithm 2** The proposed HR GME algorithm

---

**Require:**  $I$ : HR image sequence;  $N$ : number of images;

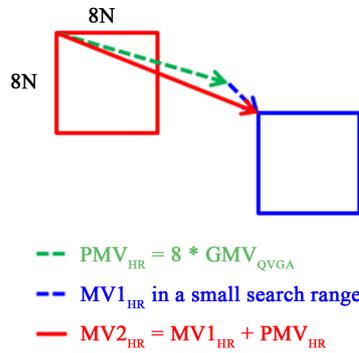
**Ensure:**  $gmv$ : HR global motion vectors

```

1: initial  $gmv_1 = (0, 0)$ ,  $R_1 = 32$ ,  $R_2 = 16$ ,  $N_1 = 16$ ,  $N_2 = 128$ ;
2: //QVGA Global motion estimation
3: for  $i = 2$  to  $N$  do
4:   Downsample  $I_i$  by a factor of 8 to obtain  $F_i$ ;
5:    $j = 1$ ;
6:   for  $r = 1$  to 5 do
7:     for  $c = 1$  to 5 do
8:       if  $r == 1$  then
9:          $pmv = gmv_{i-1}$ ;
10:      else
11:        Select  $pmv$  according to Fig. 10 and (3);
12:      end if
13:      Shift the search window in  $F_{i-1}$  by  $pmv$ ;
14:      //The search range is  $R_1$ 
15:      Find the  $mv_j$  for  $N_1 \times N_1$  block  $B_1(r, c)$  of  $F_i$ ;
16:       $mv_j = mv_j + pmv$ ;
17:       $j = j + 1$ ;
18:    end for
19:  end for
20:  Perform TOD clustering for the 16 MVs;
21:   $gmv_i =$  center of the largest cluster;
22: end for
23: //HR Global motion refinement
24: for  $i = 2$  to  $N$  do
25:    $pmv = 8 * gmv_i$ ;
26:    $j = 1$ ;
27:   for  $r = 2$  to 5 do
28:     for  $c = 2$  to 5 do
29:       Shift the search window in  $I_{i-1}$  by  $pmv$ ;
30:       //The search range is  $R_2$ 
31:       Find the  $mv_j$  for  $N_2 \times N_2$  block  $B_2(r, c)$  of  $I_i$ ;
32:        $mv_j = mv_j + pmv$ ;
33:        $j = j + 1$ ;
34:     end for
35:   end for
36:   Perform TOD clustering for the 16 MVs;
37:    $gmv_i =$  center of the largest cluster;
38: end for

```

---



**Figure 12.** Illustration of the motion estimation in high resolution domain.

The proposed global motion estimation algorithm applied the idea of predictive motion vectors, hence the search range is effectively enlarged without additional computations. The comparisons of computational complexity are given in next section.

#### 4. Computational Complexity

In this section, the proposed algorithm is compared with four algorithms, which are full search algorithm (FSA), three-step search (TSS) algorithm [13], and projection-based method [9] [10], respectively. The factor  $\bar{F}_N$  is set to 89.74 in [10]. We only compare the addition (subtraction) operations in the block matching since other computations are rather minor.

**Table 1** lists the computational costs of motion estimation for one block. The proposed algorithm is a modified version of FSA, hence the computational complexity for one block is the same as FSA. The proposed method only computes 25 blocks in QVGA and 16 blocks in HR while the others computes all the blocks to determine the global motion vector. We first compare the computation costs under QVGA resolution with block size  $N = 16$  and search range  $R = 32$ . **Table 2** shows that our method is much faster than FSA since we only consider 25 blocks, but is still slower than others especially the TSS algorithm. Consider the HR motion vectors, our method provide the effective search range:

$$R_E = 8R_1 + R_2 \quad (4)$$

Hence, in our case, the effective search range of HR motion vectors is  $R_E = 272$ . With the block size  $N = 128$ , **Table 3** shows that our method is much faster than the algorithms proposed in [9] [10] and even faster than the fast algorithm TSS [13]. **Table 4** lists the computational speed ratio with respect to FSA for HR global motion estimation. The proposed method is 4671.58 times faster than FSA and 18.05 times faster than the projection-based method [10].

Moreover, the effective search range in our method is more than 272 since we applied the PMVs to extend the QVGA search range. The effective search range in QVGA is  $R_{1E} = 2R_1$ . Note that we only consider  $R_{1E}$  as twice the  $R_1$  for conservative estimation. In fact,  $R_{1E}$  could be more than  $2R_1$  since we select the PMV from adjacent blocks which were predicted by their neighbors. Consider  $R_{1E} = 2R_1$ , the overall effective search range is  $R_E = 528$ , which is usually enough for ordinary usage of HR panoramic stitching. The computational costs are shown in **Table 5**, and the **Table 6** lists the speed ratio with respect to FSA in HR motion estimation with  $R_E = 528$ . There is no additional computations since we applied the PMVs to extend the overall search range.

The proposed algorithm can be further accelerated for sequence panoramic stitching purposes since the users usually capture an image sequence in horizontal camera motions with only little vertical jitters. In such kind of applications, we can reduce the Y-direction search range to accelerate the proposed algorithm.

#### 5. Accuracy Verifications

The proposed algorithm is a fast version of FSA which determines the global motion vector with only 25 reference blocks, hence we need to verify the accuracy of our method with the original FSA, which is considered as the golden baseline. In this section, we apply two different comparisons to verify the accuracy of our

**Table 1.** Computation costs for each block.

Methods	No. Additions/Block
FSA	$(2R+1)^2 2N^2$
TSS [13]	$9(\log_2(R/2)+1)2N^2$
[9]	$(2R+1)^2 (2N-1) + 2N^2$
[10]	$(2R+1)^2 (N-2) + \bar{F}_x (2(2N-3)) + 8N^2$
Our method	$(2R+1)^2 2N^2$

**Table 2.** Computation costs for whole QVGA image with  $R = 32$ ,  $N = 16$ .

Methods	No. Add./Block	No. Blocks	Total No. Add.
FSA	2,163,200	300	648,960,000
TSS [13]	23,040	300	6,912,000
[9]	131,487	300	39,446,100
[10]	66,403	300	19,920,900
Our method	2,163,200	25	54,080,000

**Table 3.** Computation costs for whole QVGA image with  $R = 32$ ,  $N = 16$ .

Methods	No. Add./Block	No. Blocks	Total No. Add.
FSA	9,732,915,200	300	2,919,874,560,000
TSS [13]	2,385,101	300	715,530,300
[9]	75,774,143	300	22,732,242,900
[10]	376,01,630	300	11,280,489,000
Our method	(QVGA) 2,163,200 (HR) 35,684,352	(QVGA) 25 (HR) 16	625,029,632

**Table 4.** Speed ratio of  $R_E = 272$  with respect to FSA for HR global motion estimation.

Methods	Total No. Add.	Speed Ratio
FSA	2,919,874,560,000	1.00
TSS [13]	715,530,300	4080.71
[9]	22,732,242,900	128.45
[10]	11,280,489,000	258.84
Our method	625,029,632	4671.58

**Table 5.** Computation costs for whole HR image with  $R_E = 528$ ,  $N = 128$ .

Methods	No. Add./Block	No. Blocks	Total No. Add.
FSA	36,610,015,232	300	10,983,004,569,600
TSS [13]	2,667,300	300	800,190,000
[9]	284,931,263	300	85,479,378,900
[10]	140,949,854	300	42,284,956,200
Our method	(QVGA) 2,163,200 (HR) 35,684,352	(QVGA) 25 (HR) 16	625,029,632

**Table 6.** Speed ratio of  $R_g = 528$  with respect to FSA for HR global motion estimation.

Methods	Total No. Add.	Speed Ratio
FSA	10,983,004,569,600	1.00
TSS [13]	800,190,000	13725.50
[9]	85,479,378,900	128.49
[10]	42,284,956,200	259.74
Our method	625,029,632	17571.97

method. Section 5.1 gives the comparisons of global motion vectors obtained by FSA and our method. Another comparison is PSNR in the overlapped region of every two successive images. For panoramic stitching applications, the overlapped regions of two successive images should be as similar as possible, such that there will be less visual discontinuities. In Section 5.2, we apply the PSNR value as the quantitative index to show that the global motion vector errors do not affect the stitching quality since the PSNR differences are below 0.5 dB, which is usually treated as noise-level difference and negligible.

### 5.1. Estimation Errors

If we disregard the expensive computations of FSA, the GMV obtained by FSA can be the ground truth since FSA takes all the candidate blocks in search windows into considerations. Therefore, we should compare the GMV obtained by our method with the GMV obtained by FSA, the GMV error is defined as follows:

$$E_p = |GMV_{p1} - GMV_{p2}|, p \in \{X, Y\} \quad (5)$$

We tested our method with six real-world panoramic sequences, including three indoor scenes and three outdoor scenes. There are at least 19 images in each sequence, and there are total 148 pairs of successive images used to estimate the global motion vectors. All the sequences were captured by hand-held camera and the major motion was horizontal. **Figure 13** shows the sample images of the six sequences.

**Figure 14** shows the histograms of GMV error calculate by (5). The first column represents X-direction error histograms and the second column represents Y-direction error histograms. The horizontal axis of histograms are the GMV error (in pixels) and vertical axis are the number of images. We can observe that the number of images with great GMV errors in the indoor sequences are more than outdoor ones since there are more homogeneous regions (e.g. ceiling, walls, etc) which will degrade the accuracy of motion estimation and large disparities which made local motion vectors in large variance. The outdoor scenes usually contain more textures which are helpful to the block matching and the disparities are usually small, hence there are small number of images with large GMV errors in contrast to the indoor scenes. **Figure 14** shows that the accuracy of our method is comparable to FSA. Even in the indoor sequences with large motions, which are challenging for global motion estimation, our method performed a satisfactory accuracy (GMV errors within 5 pixels) in most of images.

### 5.2. Quantitative Evaluation

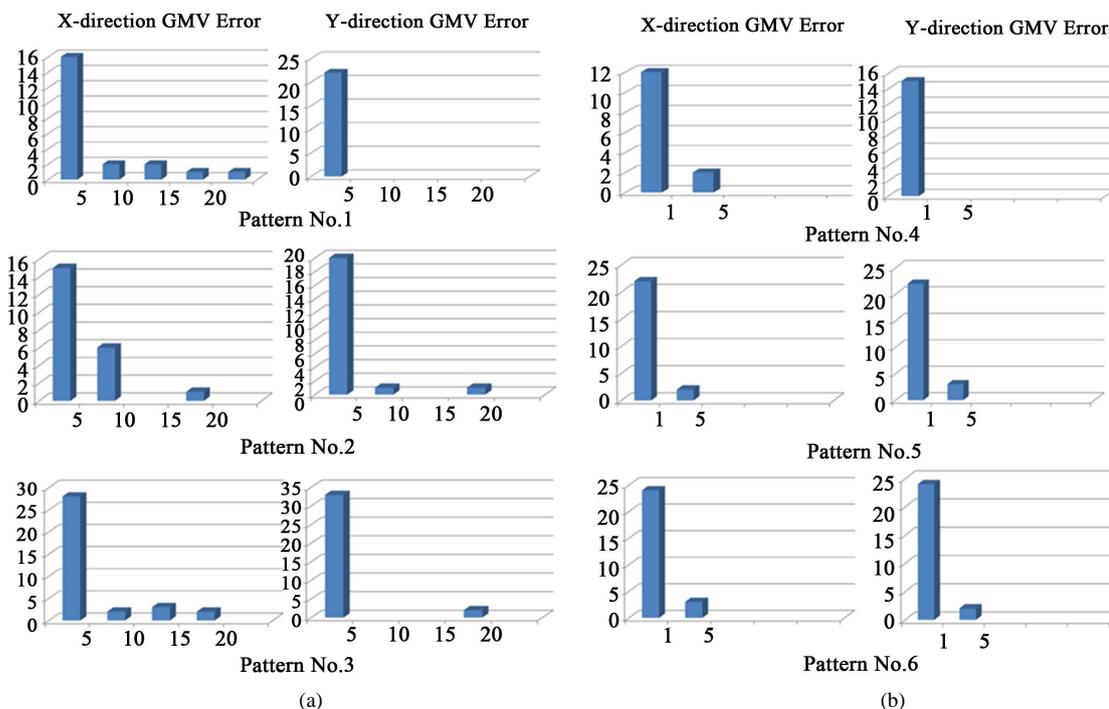
The GMV errors is not enough to describe “how accurate our method is”. Another way to evaluate the accuracy of the GMV is to measure the “similarity” of the overlapped region of two images, *i.e.*, the more accurate GMV, the less difference between the two images in the overlapped region. A well-known index to evaluate the “similarity” of two images is peak signal-to-noise ratio (PSNR) [27], which is defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{N_{\Omega}} \sum_{(i,j) \in \Omega} |I_t(i,j) - I_{t+1}(i,j)|^2} \quad (6)$$

where  $\Omega$  is the overlapped region as shown in **Figure 15** and  $N_{\Omega}$  is the number of pixels in the overlapped region. The PSNR value computed in the overlapped region of two images using the GMV obtained by FSA



**Figure 13.** Test sequence for accuracy verification. (1)-(3) are indoor sequences. (4)-(6) are outdoor sequences. All the sequences are captured by hand-held camera with mainly horizontal motion. The image resolution is  $2560 \times 1920$  pixels.



**Figure 14.** The GMV error statistic hitogram. (a) The indoor sequences GMV errors (in pixels); (b) The outdoor sequences GMV errors (in pixels). The horizontal axis are GMV errors (in pixels) and the vertical axis are number of images.

should be the highest one. The only thing we need to prove is that the difference between the PSNR of two images using GMV of FSA and the PSNR of two images using the GMV of our method is small enough. Therefore, we used the same sequences in Section 5.1 to compare the PSNR differences, which is defined as follows:

$$\Delta PSNR = PSNR_{FSA} - PSNR_{proposed} \tag{7}$$

Figure 16(a) shows the histograms of PSNR differences of indoor sequences and Figure 16(b) shows the

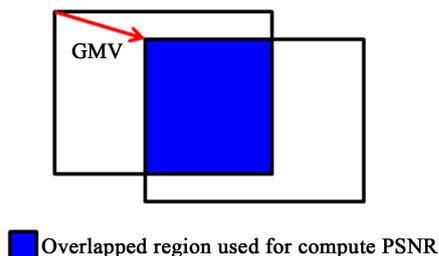


Figure 15. Only the overlapped regions in two successive images are used to compute the PSNR value.

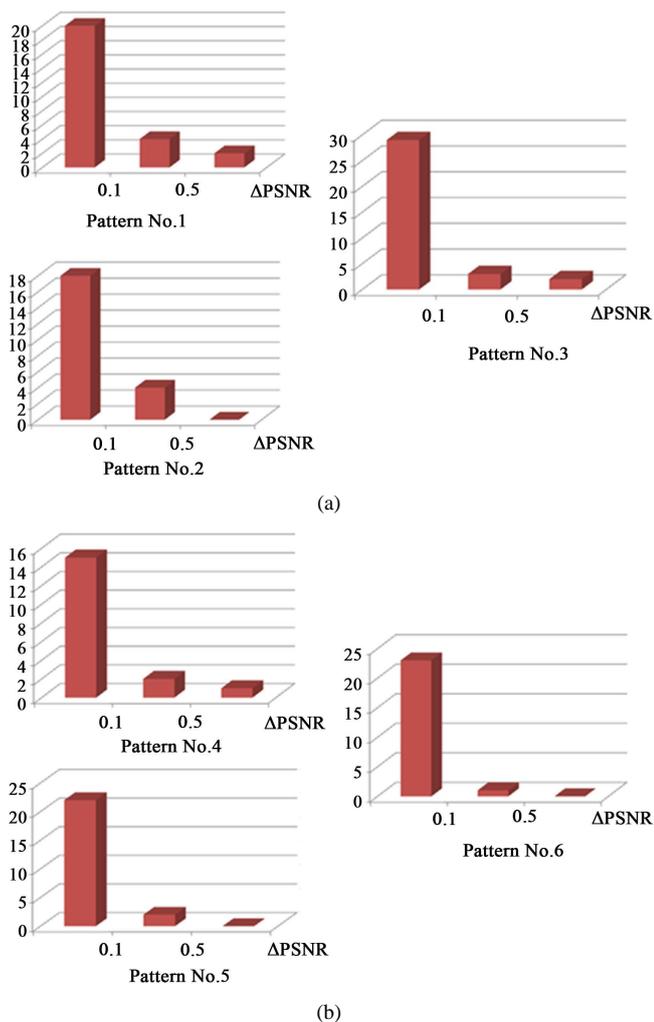


Figure 16. The histograms of PSNR differences. (a) The PSNR differences of indoor sequences; (b) The PSNR differences of outdoor sequences.

histograms of PSNR differences of outdoor sequences. The horizontal axis are  $\Delta PSNR$  (in dB) and the vertical axis are the number of image pairs. Note that most of the  $\Delta PSNR$  values are less than 0.5 dB, which is usually considered as the noise-level difference, *i.e.*, the similarity of the overlapped region in two images based on our GMV is comparable to the one based on GMV of FSA.

## 6. Conclusions

This paper proposed a fast global motion estimation algorithm for HR ( $2560 \times 1920$ ) image alignment of mobile applications. The proposed method is a modified version of full search algorithm which only considers 25 reference blocks uniformly distributed in the center of image. By applying the predictive motion vector scheme, our method is able to deal with the large camera motions and even faster than the typical three-step search (TSS) algorithm. The local minimum problem is avoided since the proposed method is a kind of full search algorithm. Six real-world sequences with total 148 pairs of successive images are used to verify our method by comparing the GMV errors and similarity with FSA. The first comparison shows that the GMV differences between our method and FSA are less than 5 pixels in both X and Y directions for most cases. The second comparison shows that the similarity of the overlapped region in two images using our GMVs is comparable with the one using GMVs of FSA.

In the future, we will focus on solving the challenging problems which make the block matching task difficult. For example, illumination differences between two successive images, large disparity in the scenes. These problems exist in real world image sequences especially in indoor scenes. We will aim to improve the algorithm to alleviate the problems and increase the accuracy.

## Acknowledgements

The authors would like to thank the Editor and the referee for their comments. This work was supported in part by the National Science Council, Taiwan, under Grant No. 98-2221-E-009-138.

## References

- [1] Milgram, D.L. (1975) Computer Methods for Creating Photomosaics. *IEEE Transactions on Computers*, **11**, 1113-1119. <http://dx.doi.org/10.1109/t-c.1975.224142>
- [2] Davis, J. (1998) Mosaics of Scenes with Moving Objects. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, 23-25 Jun 1998, 354-360. <http://dx.doi.org/10.1109/cvpr.1998.698630>
- [3] Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D. and Cohen, M. (2004) Interactive Digital Photomontage. *ACM Transactions on Graphics (TOG)*, **23**, 294-302. <http://dx.doi.org/10.1145/1186562.1015718>
- [4] Xiong, Y. and Pulli, K. (2010) Fast Panorama Stitching for High-Quality Panoramic Images on Mobile Phones. *IEEE Transactions on Consumer Electronics*, **56**, 298-306. <http://dx.doi.org/10.1109/tce.2010.5505931>
- [5] Zomet, A., Levin, A., Peleg, S. and Weiss, Y. (2006) Seamless Image Stitching by Minimizing False Edges. *IEEE Transactions on Image Processing*, **15**, 969-977. <http://dx.doi.org/10.1109/tip.2005.863958>
- [6] Jia, J. and Tang, C.-K. (2005) Eliminating Structure and Intensity Misalignment in Image Stitching. *Tenth IEEE International Conference on Computer Vision (ICCV)*, **2**, 1651-1658. <http://dx.doi.org/10.1109/iccv.2005.87>
- [7] Dufaux, F. and Konrad, J. (2000) Efficient, Robust, and Fast Global Motion Estimation for Video Coding. *IEEE Transactions on Image Processing*, **9**, 497-501. <http://dx.doi.org/10.1109/83.826785>
- [8] Su, Y., Sun, M.-T. and Hsu, V. (2005) Global Motion Estimation from Coarsely Sampled Motion Vector Field and the Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, **15**, 232-242. <http://dx.doi.org/10.1109/TCSVT.2004.841656>
- [9] Tu, C., Tran, T.D., Prince, J.L. and Topiwala, P.N. (2000) Projection-Based Block-Matching Motion Estimation. *International Symposium on Optical Science and Technology*, 374-383.
- [10] Puglisi, G. and Battiato, S. (2011) A Robust Image Alignment Algorithm for Video Stabilization Purposes. *IEEE Transactions on Circuits and Systems for Video Technology*, **21**, 1390-1400. <http://dx.doi.org/10.1109/tcsvt.2011.2162689>
- [11] Battiato, S., Bruna, A.R. and Puglisi, G. (2010) A Robust Block Based Image/Video Registration Approach for Mobile Image Devices. *IEEE Transactions on Multimedia*, **12**, 622-635. <http://dx.doi.org/10.1109/tmm.2010.2060474>

- [12] Zhu, C. and Qi, W.-S. and Ser, W. (2005) Predictive Fine Granularity Successive Elimination for Fast Optimal Block-Matching Motion Estimation. *IEEE Transactions on Image Processing*, **14**, 213-221. <http://dx.doi.org/10.1109/TIP.2004.840702>
- [13] Koga, T. (1981) Motion-Compensated Interframe Coding for Video Conferencing. *Proceedings of National Telecommunication Conference*, New Orleans, 29 November-3 December 1981, G5.3.1-G5.3.5.
- [14] Li, R., Zeng, B. and Liou, M.L. (1994) A New Three-Step Search Algorithm for Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **4**, 438-442. <http://dx.doi.org/10.1109/76.313138>
- [15] Po, L.-M. and Ma, W.-C. (1996) A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **6**, 313-317. <http://dx.doi.org/10.1109/76.499840>
- [16] Zhu, S. and Ma, K.-K. (2000) A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. *IEEE Transactions on Image Processing*, **9**, 287-290. <http://dx.doi.org/10.1109/tip.2000.826791>
- [17] Zhu, C., Lin, X. and Chau, L.-P. (2002) Hexagon-Based Search Pattern for Fast Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **12**, 349-355. <http://dx.doi.org/10.1109/TCSVT.2002.1003474>
- [18] Liu, L.-K. and Feig, E. (1996) A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, **6**, 419-422. <http://dx.doi.org/10.1109/76.510936>
- [19] Battiato, S., Gallo, G., Puglisi, G., and Scellato, S. (2007) SIFT Features Tracking for Video Stabilization. *14th International Conference on Image Analysis and Processing (ICIAP)*, Modena, 10-14 September 2007, 825-830. <http://dx.doi.org/10.1109/iciap.2007.4362878>
- [20] Yang, J., Schonfeld, D. and Mohamed, M. (2009) Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion. *IEEE Transactions on Circuits and Systems for Video Technology*, **19**, 945-954. <http://dx.doi.org/10.1109/TCSVT.2009.2020252>
- [21] Farin, D. and Peter H.N. de W. (2005) Evaluation of a Feature-Based Global-Motion Estimation System. *Visual Communications and Image Processing*, 12 July 2005 SPIE—The International Society for Optical Engineering, Beijing, 59603X-59603X-12.
- [22] Bosco, A., Bruna, A., Battiato, S., Bella, G. and Puglisi, G. (2008) Digital Video Stabilization through Curve Warping Techniques. *IEEE Transactions on Consumer Electronics*, **54**, 220-224. <http://dx.doi.org/10.1109/tce.2008.4560078>
- [23] Fang, X., Luo, B., Zhao, H., Tang, J. and Zhai, S. (2010) New Multi-Resolution Image Stitching with Local and Global Alignment. *IET Computer Vision*, **4**, 231-246. <http://dx.doi.org/10.1049/iet-cvi.2009.0025>
- [24] Bay, H., Ess, A., Tuytelaars, T. and Van Gool, L. (2008) Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, **110**, 346-359. <http://dx.doi.org/10.1016/j.cviu.2007.09.014>
- [25] Srinivasan, R. and Rao, K.R. (1985) Predictive Coding Based on Efficient Motion Estimation. *IEEE Transactions on Communications*, **33**, 888-896. <http://dx.doi.org/10.1109/tcom.1985.1096398>
- [26] Kandel, A. (1999) Introduction to Pattern Recognition: Statistical, Structural, Neural, and Fuzzy Logic Approaches. World Scientific, Singapore.
- [27] Gonzalez, R.C. and Woods, R.E. (2002) Digital Image Processing. 2nd Edition, Prentice-Hall, Inc., Upper Saddle River.