An efficient and fast global motion estimation algorithm based on motion vector field

Pin Lui*, Jiaying Liu, Zongming Guo
Institute of Computer Science and Technology, Peking University, Beijing, P.R. China

ABSTRACT

Global motion estimation (GME) is widely used in image/video processing and various applications. But the accuracy of estimation results is badly influenced by local motion and noises. Furthermore, the conventional GME algorithms in spatial domain usually need a large number of iteration times, which makes computational complexity extremely higher. In this paper, we propose an efficient and fast GME algorithm based on motion vector field, which adaptively selects input pixels for solving transform models. More characteristics of the image are considered, such as the difference between global motion and local motion, the distribution of motion vectors, and macroblock partition modes. The proposed algorithm includes three steps: First, we obtain several sets of pixels by merging similar bins in the histogram of motion vectors and generate a weight map. Second, we choose the cluster with the minimum distribution variance in the image as the cluster representing the global motion. The pixels with higher weights in this cluster are chosen as the input pixels for solving transform models. Finally, we employ the 6-parameter affine model as the transform model and calculate the parameters. Experimental results show that the proposed algorithm is effective and fast.

Keywords: Global motion estimation (GME), motion vector field, pixel cluster, map fusion

1. INTRODUCTION

Motion estimation and compensation is a core technology of in video processing. Motion in video can be classified into two categories: global motion and local motion. The former refers to the camera motion, while the latter refers to the motion of the objects. Most motion estimation techniques do not distinguish these two kinds of motion, and the motion vectors of the blocks are the mixture of the two. However, separating global and local motion not only results in a more reasonable and compact representation of the motion information, but also leads to many other extended applications besides video compression, such as video mosaicing [1], content based video classification [2], camera motion tracking.

Conventional GME algorithms are mainly classified into the following three categories: feature based methods, frequency domain methods and spatial domain methods. Other methods like spatio-temporal are also developed [7]. Feature based methods rely on extraction and tracking of feature points, which are very difficult especially when dealing with some appearing or disappearing features. Frequency domain methods [8], based on affine theorem of Fourier Transformation and phase correlation techniques, cannot be extended in natural way to higher order motion models. Spatial domain methods are used widely, but most spatial domain methods are iterative and usually involve image warping [3][4][6], which makes it computationally intensive and slow.

Motion models are used to depict global motion no matter what the global motion estimation method is. There are different motion models [5], which can be classified based on the parameters' number. Generally, the model's complexity increases with the number of its parameters. For example, the common practical motion models include 2-parameter translational model, 4-parameter geometric model, 6-parameter affine model, and 8-parameter perspective model. But the ability to describe motion is also increased with the parameter number. The 2-parameter translational model only describes translation, while the 8-parameter perspective model can represent isotropic magnification, translation and rotation. Because of the existence of disappearing pixels and reappearing pixels, even the most complex motion models are not able to represent the actual motion completely. But usually the models are sufficient to the requirements of practically usage. The lower order models are seen as special cases of higher order models. In these models, the 8-parameter perspective model is the most general one, and the 6-parameter affine model provides good tradeoff between generality and ease of estimation.

*lvpin@icst.pku.edu.cn;

Theoretically, only 3 pairs of pixels in reference frame and current frames are needed for solving the 6-parameter affine model, and 4 pairs for 8-parameter perspective model, but these pixels must only participate in global motions. Since we are usually not sure whether a pixel is involved with local motion at the same time, the whole image area is assumed.

Taking account of the chosen model, we calculate the parameters of the model by minimizing the error function via Newton-Raphson method. It is computational costly, thus several speed-up strategies have been suggested in literature [9]-[13]. Szeliski *et al.* used multi-resolution framework [4]. Keller *et al.* proposed selective integration and warp free formulation [9]. These methods reject outliers by using fixed threshold or selecting pixels with greatest gradient. However, the complexity of these algorithms is still high, because the selected pixels do not indicate the essential characteristic of the image.

Only a few pixels are needed to solve models, which means most of the pixels with pure global motion are redundant for solving models and only increase the computational cost. On the other hand, outliers have great influence in the estimation process, and the accuracy of the result does not necessarily increase with the amount of input pixels for solving models. Though various methods are used to reject outliers, most of them deal with it in the iterative process of solving models, thus increase the computational cost. It is advantageous to reject outliers before solving models rather than in the process of solving models. To sum up, it is not only important but also necessary to carefully select the input pixels for solving models. The carefully selected pixels should verily represent the global motion, which demands synthetically consideration in characteristics of the motion and the image.

The rest of the paper is organized as follows. Section 2 reviews the conventional GME algorithm on spatial. Section 3 introduces the proposed novel algorithm based on motion vector field, which adaptively selects input pixels. Experimental results are given in Section 4. Finally, concluding remarks and discussion are given in Section 5.

2. CONVENTIONAL GME ALGORITHM ON SPATIAL

Given the motion model, global motion estimation on spatial becomes an optimization problem to minimize the error function, which is the sum of squared differences (SSD) between the current frame I and the motion compensated previous frame $\tilde{I}$,

$$E = \sum_{i=1}^{n} e_i^2, \quad with \ e_i = \tilde{I}(\tilde{x}_i, \tilde{y}_i) - I(x_i, y_i), \tag{1}$$

where $I(x_i, y_i)$ denotes the spatial coordinates of pixel i in the current frame, and $\tilde{I}(\tilde{x}_i, \tilde{y}_i)$ denotes the coordinates of the corresponding pixel in the motion compensated previous frame. n is the total number of pixels.

Conventional GME algorithm rejects outliers mainly in iterative model-solving process. One of the most popular algorithm, which is called M-estimators [14], removes the influence of outliers by minimizing

$$E = \sum_{i=1}^{n} \rho_i^2, \tag{2}$$

where ρ is a symmetric positive-definite function with a unique minimum at $e_i = 0$. Instead of the original quadratic error function

$$\rho(e_i) = e_i^2 \tag{3}$$

A truncated quadratic is used.

$$\rho(e_i) = \begin{cases} e_i^2, & |e_i| \le t \\ 0, & |e_i| > t \end{cases}, \tag{4}$$

where t is the threshold. The algorithm is based on the following reason: the error function (see Equation (1)) takes the assumption that the error terms are assumed to be independent and identically distributed zero-mean Gaussian random variable in both the horizontal and the vertical directions, so minimization of the error function is equivalent to

maximum-likelihood estimation with independent Gaussian noises with a constant standard deviation. But it is not applicable in practical because of many outliers. So a binary mask is used in M-estimator algorithm.

3. GME ALGORITHM WITH ADAPTIVELY SELECTING INPUT PIXELS

In this section, we present a GME algorithm, which adaptively selects input pixels considering the motion and image characteristics synthetically. The algorithm includes the following 3 stages. First, we get several sets of candidate pixels which are called pixel clusters, and generate some weight maps which are fused into one later. Second, we decide which cluster is more probable to represent the global motion. Finally, we employ the 6-parameter affine model as the transform model, calculate the parameters of the model and make global motion compensation to the reference frame. Figure 1 shows the block diagram of our algorithm.

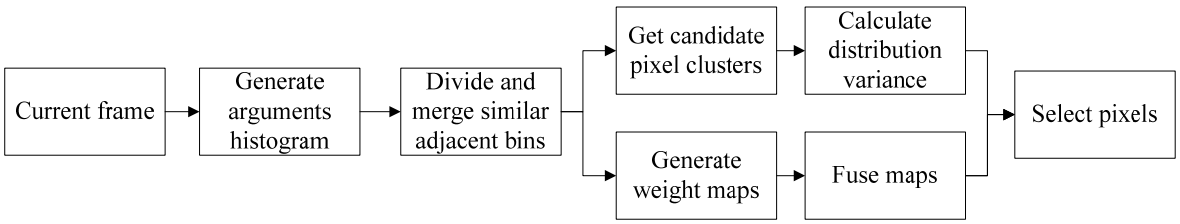


Fig. 1. Block diagram of proposed algorithm.

3.1 Pixel Clustering

A large number of experimental observations indicate that different objects usually have different motion (Figure 2). The 2-D histogram of motion vectors on both horizontal and vertical direction shows this clearly. When employing the 1-D histogram, either of the motion direction or of the magnitude of the motion vector, the difference shown is obvious enough to separate different kinds of pixels. We choose 1-D histogram instead of 2-D histogram, because the former is less complex while is good enough for separating pixels. As to the problem of choosing the 1-D feature, though the amplitude and the argument of a vector take different and irreplaceable characteristics respectively, the histogram of motion direction is more suitable. Because the motion vectors are depicted in Cartesian coordinates, and the argument is not too sensitive to the spatial coordinates of the pixel compared with the amplitude.

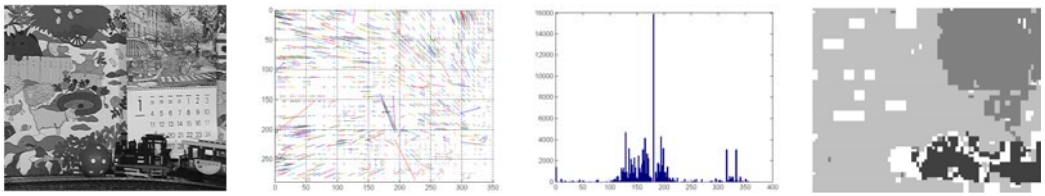


Fig. 2. MOBILE sequence, 116th frame. (a) The intensity image. (b) Motion vectors. The lengths are 4 times of original amplitude. Different colors mean different arguments. (c) Argument density histogram. (d) Corresponding segmented image. Different gray scales represent different clusters.

We segment a histogram to separate different pixels into clusters. Histogram segmentation is widely used in image analysis and data analysis. It is one subject of clustering in pattern classification. Several classical algorithms have been developed in literature, such as k-means algorithm, hierarchical algorithm, Gauss Mixture Models. These methods are different in many ways. But the difference is not that important in our algorithm since the pixel clustering in our algorithm is not a standard clustering problem. In fact it is much easier than clustering problem. This is because the goal of pixel clustering in our algorithm is not to cluster each pixel; we just need to select representative pixels for different motions. According to observations, though the amount of pixels with local motion may be larger than pixels only with global motion, yet clusters representing global motion are usually more significant in the histogram. This is because different object usually have different local motion, and the density is dispersed. If it is not under extreme circumstances, the pixels only involved in global motion always make up of clusters, which are obvious and significant in the histogram.

So the first significant 4~6 clusters in the histogram is enough for finding clusters representing global motion. The differences between clustering in our algorithm and in pattern classification include the following four aspects.

1. In our algorithm, it is unnecessary to cluster all pixels. We employ pixel clustering in our algorithm in order to get some representative pixels, so it does not matter if some pixels do not belong to any clusters. While for clustering problems in pattern classification, any pixel should belong to a certain cluster.

2. In our algorithm, it is unnecessary to get all clusters. Only significant clusters are needed. Compared with clustering in pattern classification, this condition is very loose.

3. Though exact segmentation which has no over and under segmentations is very good for our algorithm, slight over segmentation is also tolerable, while under segmentation may bring into some drawbacks and should be avoided.

4. In our algorithm, if two bins are not adjacent, they must not belong to the same cluster, while histogram segmentation in pattern classification does not have this restrict. In fact, the same global motion may indeed have different motion direction, for example, the global motion of zoom in/out has motion directions covering degrees from 0 to 360. But as mentioned above, over segmentation is tolerable while under segmentation should be avoided, thus the restriction is reasonable and guarantees that different motions are differentiated.

Since the differences between different clustering algorithms are not that important for our application, we simply choose the k-means method to segment 1-D histogram. The k-means algorithm works like this: each cluster in the partition is defined by its member objects and by its centroid, or center. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is minimized. The result is a set of clusters that are as compact and well-separated as possible.

$$d_{k,i} = \left\| x_k - \hat{\mu}_i \right\|^2 \tag{5}$$

We use squared Euclidean distances as similarity/dissimilarity evaluation. The Euclidean distances are calculated according to Equation (5), where x_k is a sample point belonging to cluster i and $\hat{\mu}_i$ is the centroid of cluster i. Each centroid is the mean of the points in that cluster. For more details, please refer to [15].

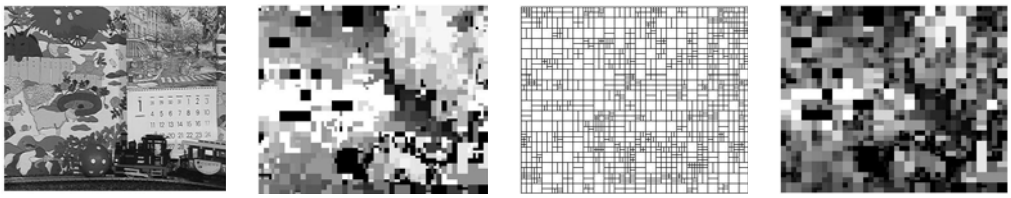
To generate the 1-D histogram, arguments of motion vectors are normalized into the degree range of [0, 360]. The density histogram of arguments is generated with an interval of 10 degrees; therefore there are 36 bins in the histogram. The sample data includes the bins' number and the number of data in the bin. The k-means algorithm demands that the cluster number should be given before. As mentioned above, the pixel clustering in our algorithm is not a standard clustering problem, so we simply set the cluster number to 4. Notice the clustering results cannot be used directly. In our algorithm only adjacent bins may belong to the same cluster, so we re-cluster the result of clustering algorithm, simply let bins which are not adjacent while belonging to the same cluster belong to different clusters. Thus the number of clusters we actually get is often more than 4. And since we only need the first few significant clusters, we select the first 4~6 significant clusters that have higher peaks and larger densities.

The complexity of k-means algorithm is $O(ndcT)$, where d is the number of features, which is also called the dimension of samples. c is the number of clusters. T is the iteration times, which is usually less than the samples' number. Because both the samples' number and dimension are very small, the algorithm works very fast. An example of corresponding segmented image example is shown in Figure 2 (d).

In the above process, we do not consider that there is no global motion in the image. But it should not be neglected. We also generate a cluster representing no motion. Pixels of this cluster do not have motion vectors.

3.2 Map Generation

Weight map is useful for treating pixels based on their levels of importance. Pixels have different levels of importance in global motion estimation. On one hand, there may be some pixels, which do not belong to any cluster. It means that the pixels belong to unstructured areas and should be regarded as outliers. On the other hand, a cluster covers certain range of arguments, and data have different distances from the centroid, which means pixels have different levels of similarity within the cluster. In addition, pixels also vary in other aspects. For example, two pixels with the same motion vector belong to the same cluster while one's surrounding area is smooth while the other's is delicate, and the former is more probably to be a background pixel. Usually motion vectors of the edge pixels are regarded as more reliable than the inside pixels. These factors should be taken into consideration.



(a) (b) (c) (d)

Fig. 3. MOBILE sequence, 116th frame. (a) The intensity image. (b) Similarity map. Different gray scales represent different weights. (c) Partition of the frame. (d) Fused Map. Different gray scales represent different weights.

We generate a similarity map (Figure 3 (b)) for treating pixels of the same cluster differently. If a pixel belongs to a certain cluster, its weight is calculated based on Equation (6), where $x_{k,i}$ represents the motion vector argument of pixel *k* in cluster *i*, and $\hat{\theta}_i$ is the argument centroid of cluster *i*. Other weight is set to 0.

$$\omega_{k,i} = \left| x_{k,i} - \hat{\theta}_i \right| \tag{6}$$

The texture information of an image is helpful in analyzing the image content. The partition modes map (Figure 3 (c)) of an image contains some simple information of image texture. The partition mode of a block reflects the characteristic of the area. For example, background tends to be smooth area with less details and it is more probable to be partitioned into larger blocks, such as 16x16, 16x8, and 8x16, while objects with details are tend to be partitioned into smaller blocks, such as 8x4, 4x4. Thus we generate a map based on the partition mode of the block, with smaller weight for smaller blocks. Pixels of larger blocks are preferred to have greater weights.

After obtaining these maps, we fuse them into one simply by adding. The weight of the pixel means its representativeness of the cluster it belongs to. The weight map will be used in later stage.

3.3 Candidate Cluster Selection

In the above process, different clusters are treated equally, because the information used above to obtain clusters and generate weight maps does not indicate whether the motion is global or local. But if we take some prior knowledge and observation facts into consideration, generally the two kinds of motion have different characteristics, which can be used to separate them. For example, the local motion is often restricted in a small area, while the global motion spreads around the whole image. To certain extent, this difference is one of the reasons why motions are seen as global or local. It is not always right to distinguish global and local motion by the amount of pixels. And distinguishing them by their spatial distribution is more reasonable. Uniform distribution means wide spread in the image.

To evaluate the spatial distribution difference between the two kinds of motion, we calculate the spatial distribution variance for each cluster. First, we divide the image into several circle-like areas corresponding to the clusters. Then we calculate the density function and variance for each cluster. Suppose the circle-like areas are Ω_1, Ω_2, ..., Ω_m from the image center to margin. The spatial distribution density of cluster *i* in area *k* is calculated as this:

$$p_{i,k} = M_{i,k} / N_k, \quad 1 \le k \le m, \tag{7}$$

where N_k denotes the pixel number in the circle-like area *k*, while $M_{i,k}$ is the number of pixels belonging to cluster *i* in area *k*. And the spatial distribution variance of cluster *i* in the whole image is calculated as,

$$v_i = \sum_{k=1}^{m} \sigma_k p_{i,k}^2, \tag{8}$$

where σ_k is the weight of circle-like areas *k* which satisfies the following condition:

$$\sigma_k \le \sigma_t, \quad if \quad s < t. \tag{9}$$

We set different weights for these areas, based on the prior knowledge that outer areas are more probably to be background and mainly involve in global motion while inner areas are more probably to be foreground and involve both

local and global motion. So the density functions of outer areas are more important in evaluating the spatial distribution of a cluster and should be assigned larger weights.

The distribution variance of a cluster reflects its distribution uniformity in the whole image of the cluster. We choose the cluster with the minimum variance as the cluster representing the global motion. If there are two or more clusters with the same minimum variance, which is a rare situation, we choose the one having more pixels.

Pixels with larger weights in this cluster are more convincing than the others. We sort the pixels of the cluster in descending order according to their weights and choose the first few pixels. In our algorithm the number is given as this.

$$q = \min(|\Lambda|, 2048), \tag{10}$$

where Λ is pixel set of the cluster. q is not larger than 2048. For a CIF format (352x288) sequence, the input pixels for solving models are only 2% of the original size. Thus the computational complexity is greatly decreased.

4. EXPERIMENTAL RESULTS

The proposed algorithm was implemented with Matlab in our experiments, compared with the conventional methods without adaptively selecting input pixels for solving transform models. All the experiments were done on the following standard test sequences in CIF format: VECTRAN, COASTGUARD, MOBILE, FOREMAN, STEFAN. The resolution of these sequences is 352 by 288. The H.264/AVC reference software JM86 is used to encode these sequences. The configuration is listed in Table 1.

Table. 1. H.264/AVC encoder configuration

Video Format	YUV 4:2:0, 8 bits
Profile / Level	66 (baseline profile) / 30
Number of Reference Frames	10
Hadamard in Sub-pel Search	Yes
Search Range	16
Sub-pel ME	Enabled
Backward Search	No
Inter Search 8x4, 4x8, 4x4	Enabled
Joint Estimation	No
Weighted Prediction	No
GOP Structure	IPPP
QP	28

For each sequence, we pseudo-randomly selected 10 P-frame. Both subjective quality and objective quality are compared. The PSNR of the current frame to compensated reference frame using proposed algorithm and conventional algorithm is calculated respectively. And the PSNR of the current frame to reference frame without global motion compensation is also calculated as a contrast. Table 2 shows the results.

We notice that the objective quality may not be consistent with the subjective quality, especially in global motion estimation and compensation. The local motion will not be compensated. So if there are many moving objects in an image, the PSNR of the current frame to compensated reference frame may be worse than that of the current frame to reference frame without compensation even if global motion estimation and compensation is explicit. Large structured areas in local motion are excluded from the calculation of PSNR for impartiality in our experiment. Removing unstructured areas needs manual work and these areas are reserved in our experiment. In addition, after being compensated, the reference frame often moves away from its original place, leaving vacancies which we know nothing about and have to pad with value like 0 or 1. This often happens in the margins of the image. Vacancy areas should also be excluded from the calculation of PSNR for impartiality.

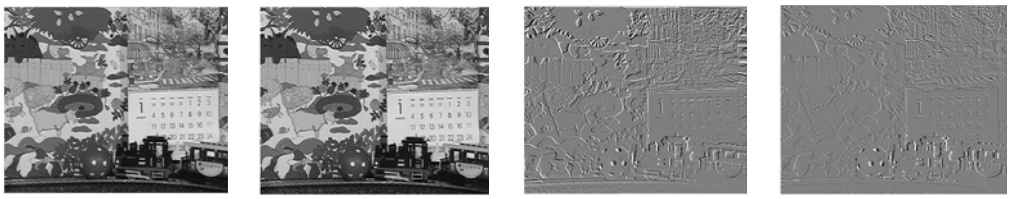
Table. 2. The average PSNR of the frame with global motion compensation to the current frame.

Sequence	Average PSNR (without compensation)	Average PSNR (without selection)	Average PSNR (proposed algorithm)	PSNR gain
COASTGUARD	27.08	30.32	31.54	1.22
VECTRA	20.00	22.76	24.39	1.63
MOBILE	21.00	18.47	19.85	1.38
STEFAN	19.50	19.09	19.05	-0.04
FOREMAN	24.79	27.09	27.04	-0.05

It is seen from the table that our proposed algorithm is effective in general. The average PSNR gain of the result reaches more than 1 dB for some sequences such as COASTGUARD, VECTRA and MOBILE. But for STEFAN and FORMAN sequences our algorithm shows less superiority in objective quality. Because in STEFAN there are large areas of unstructured but detailed background which tend to lower the PSNR. This can be seen from the low PSNR of the current frame to reference frame without global motion compensation. And for FOREMAN, the reason is probably that there are too many homogeneous areas especially in background, and the camera motion is slight in many frames. Thus our algorithm is less accurate in selecting pixel cluster representing global motion. Nevertheless, the PSNR results of these two sequences only have a slight decrease.

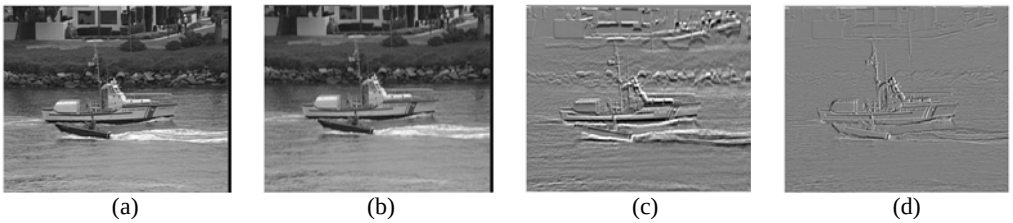
(a) (b) (c) (d)

Fig. 4. VECTRA sequence. (a) Current frame (71th frame). (b) Reference frame. (c) Difference image between (a) and compensated reference frame without selecting pixels. (d) Difference image between (a) and compensated reference frame with proposed algorithm.



(a) (b) (c) (d)

Fig. 5. MOBILE sequence. (a) Current frame (116th frame). (b) Reference frame. (c) Difference image between (a) and compensated reference frame without selecting pixels. (d) Difference image between (a) and compensated reference frame with proposed algorithm.



(a) (b) (c) (d)

Fig. 6. COASTGUARD sequence. (a) Current frame (73th frame). (b) Reference frame. (c) Difference image between (a) and compensated reference frame without selecting pixels. (d) Difference image between (a) and compensated reference frame with proposed algorithm.

As to subjective quality, the difference images between current frame and reference frame with or without global motion compensation are shown in Figure 4 and Figure 5. Besides the pseudo-randomly selected frames, we also tested some special frames. For example, the camera strongly shakes in COASTGUARD from frame 65 to frame 75. The typical global motion is well estimated. The result is shown in Figure 6.

5. CONCLUSION AND FUTURE WORK

In this paper, we propose an efficient and fast global motion estimation algorithm based on motion vector field. By considering the distribution characteristics of motion vectors and the area smoothness, the algorithm selects pixels mainly involved in global motion. Thus the accuracy of the estimation gets improved, and the computational complexity is lowered. The effectiveness of the method is proved by experiments on standard test sequences. The average PSNR gain of the result reaches more than 1 dB for some sequences, while the proposed method only uses a small subset of the original pixels, which greatly decreases the computational cost. Future work will focus on getting more information from the image and extend them to other applications such as tracking of regions of interest.

ACKNOWLEDGEMENTS

This work is supported by National Basic Research Program (973 Program) of China under contract No.2009CB320907.

REFERENCES

[1] R. Szeliski, “Image mosaicing for tele-reality,” Proc. IEEE Workshop on Applications of Computer Vision, 44-53 (1994).

[2] E. Ardizzone, M. La Casica, D. Molinelli, “Motion and color-based video indexing and retrieval,” Proc. ICPR, 135-139 (1996).

[3] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, “Hierarchical model-based motion estimation,” Proc. ECCV, 237-252 (1992).

[4] R. Szeliski and J. Coughlan, “Hierarchical splinebased image registration,” Proc. CVPR, 194-201(1994).

[5] Glasbey, C. A. andMardia, K. V., “A review of image warping methods,” Journal of Applied Statistics. Papers 3, 155-171 (1998).

[6] F. Dufaux and J. Konrad, “Efficient, robust, and fast global motion estimation for video coding,” IEEE Trans. Image Processing. Papers 9, 497-501 (2000).

[7] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang, “Motion analysis and segmentation through spatio-temporal slices Processing,” IEEE Trans. Image Processing. Papers 12, 341-355 (2003).

[8] S. Kumar, M. Biswas, T. Q Nguyen, “Global motion estimation in frequency and spatial domain,” Proc. of IEEE ICASSP. Papers 3, 333-336 (2004).

[9] Y. Keller and A. Averbuch, “Fast gradient methods based on global motion estimation for video compression,” IEEE Trans. CSVT. Papers 13, 300-309 (2003).

[10] A. Smolic and J.-R. Ohm, “Robust global motion estimation using a simplified m-estimator approach,” Proc. of IEEE Int. Conf. Image Processing. Papers 1, 868-871 (2000).

[11] Y. Su, M.-T. Sun, and V. Hsu, “Global motion estimation from coarsely sampled motion vector field and the applications,” Proc. of IEEE Int. Symp. Circuits and Systems. Papers 2, 628-631(2003).

[12] F. Moscheni, F. Dufaux, and M. Kunt, “A new two-stage global/local motion estimation based on a background/foreground segmentation,” Proc. of IEEE ICASSP, 2261-2264(1995).

[13] Bin Qi, Mohammed Ghazal, Aishy Amer, “Robust global motion estimation oriented to video object segmentation,” IEEE Trans. Image Processing. Papers 17, 958-967 (2008).

[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling. B.P. Flannary, [Numerical Recipes in C], Cambridge University Press, (1992).

[15] Richard O. Duda, Peter E. Hart, David G. Stork, [Pattern Classification (2nd Edition)], John Wiley & Sons, Inc. New York, (2001).