MULTI-RESOLUTION BLOCK MATCHING MOTION ESTIMATION WITH DEFORMATION HANDLING USING GENETIC ALGORITHMS FOR OBJECT TRACKING APPLICATIONS

Harish Bhaskar and Helmut Bez
Loughborough University, U.K.

Keywords: Block motion estimation, affine, deformation handling, genetic algorithms, multi-resolution, object tracking.

Abstract: Motion Estimation is a popular technique for computing the displacement vectors between objects or attributes between images captured at subsequent time stamps. Block matching is a well known technique of motion estimation that has been successfully applied to several applications such as video coding, compression and object tracking. One of the major limitations of the algorithm is its ability to cope with deformation of objects or image attributes within the image. In this paper we present a novel scheme for block matching that combines genetic algorithms with affine transformations to accurate match blocks. The model is adapted into a multi-resolution framework and is applied to object tracking. A detailed analysis of the model alongside critical results illustrating its performance on several synthetic and real-time datasets is presented.

1 INTRODUCTION

Motion estimation techniques aim at deducing displacement vectors for objects or image attributes between two consecutive frames (A. Gyaourova and Cheung, 2003). The main idea behind block matching motion estimation strategies is to divide the image frame into blocks and match blocks between successive frames within a search window using specific search techniques (A.Barjatya, 2005). It is clear that the two distinct phases that make up any block matching method is block partitioning and block searching. The block partitioning scheme is concerned with dividing the original image frame into non-overlapping regions. Block partitioning can be performed using the fixed size or variable size methods (C-C.Chang, 2006) (F.J.Ferri and J.Soret, 1998). The block search mechanism is the process of locating the block in the destination frame that best matches the block in the anchor frame using a specific matching criterion (Turaga and M.Alkanhal, 1998).

Different models have been developed in literature to accomplish robust motion estimation using block based techniques. (C-W.Ting and L-M.Po, 2004) propose the use of different search schemes with fixed and variable block partitioning methods to accomplish robust estimation. In a similar study by (M.Wagner and D.Saupe, 2000), a quad-tree block motion estimation scheme is proposed. Other methods of variable block matching have also been proposed, particularly in the form of polygon approximation, mesh based (Y.Wang and O.Lee, 1996) and binary trees. Another class of block matching methods that have recently used for deformation handling particularly in applications of object tracking is the deformable block matching (O.Lee and Y.Wang, 1995). In a study by (J.H.Velduis and G.W.Brodland, 1999), deformable block matching has been adapted for use in tracking cell particles. A bilinear transformation is used with block matching to handle deformation. In the context of deformable models,triangular or mesh based block decomposition is much popular (Y.Wang and A.Vetro, 1996). The idea behind these schemes is to partition image frames using techniques of finite element analysis, triangulation (M.Yazdi and A.Zaccarin, 1997), mesh grid etc. and employ deformable block matching of vertex points to handle complex motion changes during motion estimation. In the context of mesh based methods, a nodal based scheme for block

Bhaskar H. and Bez H. (2007).
MULTI-RESOLUTION BLOCK MATCHING MOTION ESTIMATION WITH DEFORMATION HANDLING USING GENETIC ALGORITHMS FOR OBJECT TRACKING APPLICATIONS.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 169-174
DOI: 10.5220/0001650501690174

matching is also popular. According to the nodal scheme, mesh is generated such that nodes lies across object boundaries and a simple search of linear motion of these nodal position from the anchor frame to the destination frame will be able to suffice deformation (O.Lee and Y.Wang, 1995). In this paper, we shall highlight a framework that integrates a vector quantization based block partitioning method to an genetic algorithm based search scheme with affine parametrization to accomplish robust, accurate motion estimation with deformation handling. The model is built on a multi-resolution platform with performance feedback.

2 PROPOSED MODEL

The proposed model constitutes of different phases. The first phase is the multi-resolution platform that the framework is based on. The platform combines a scale space representation of data with a multi-resolution level analysis. A multi-resolution model aims at capturing a wide range of levels of detail of an image and can in-turn be used to reconstruct any one of those levels on demand. The distinction between different layers of an image is determined by the resolution. A simple mechanism of tuning the resolution can add finer details to coarser descriptions providing a better approximation of the original image. Mathematically, we can represent the above analysis in the following way. If the resolution is represented using λ, then the initial level is associated with $\lambda = 0$ is 1 and that with any arbitrary resolution λ is $\frac{1}{2^\lambda}$. If f_λ is the image at resolution λ, then at resolution λ+1,

$$f_{\lambda+1} = f_\lambda + \Gamma_\lambda \tag{1}$$

where Γ_λ is the details at resolution λ. In contrast, the scale space representation of data deals with representing images in such a way that the spatial-frequency localizations are simultaneously preserved. This is achieved by decomposing images into a set of spatial-frequency component images. Scale space theory, therefore, deals with handling image structures at different scale such that the original image can be embedded into a one-parameter family of derived component images thereby allowing fine-scale structures to be successively suppressed.Mathematically, to accomplish the above, a simple operation of convolution can be used. However, it is important to note that the overhead of using the convolution operator is kept low. For any given image $I(x, y)$, its linear scale space representation is composed of components $L_\vartheta(x, y)$ defined as a convolution operator of the image $I(x, y)$ and a Gaussian kernel of the form:

$$G_\vartheta(x, y) = \frac{1}{2\pi\vartheta} e^{-\frac{x^2+y^2}{2\vartheta}} \tag{2}$$

, such that

$$L_\vartheta(x, y) = G_\vartheta(x, y) * I(x, y) \tag{3}$$

where $\vartheta = \sigma^2$ is the variance of the Gaussian. Performance based feedback automates the selection of relevant resolution and scale for any particular frame pair. A brief algorithm describing the process is as follows.

- Initialize the resolutions $\lambda_{[1:q]}$ to $[0, 1, 2, ..., q]$ and scales $\vartheta_{[1:q]}$ to $[1, 2, 3, ..., q + 1]$ for any value of q (4 chosen for this experiment).
- Select the median of resolutions as the initial starting resolution and scale. The median is 2 in our experiments and the chosen values of (λ, ϑ) are $(2, 3)$
- Input at any time instant t, two successive frame pairs of a video sequence, (f_t, f_{t+1}).
- Re-sample the images f_t and f_{t+1} into the selected resolution using bi-cubic interpolation
- Convolve the image at selected scale (in matching positions with the resolution) with a Gaussian kernel to obtain a filtered output $(G_\vartheta * f_t, G_\vartheta * f_{t+1})$
- Perform Motion Estimation of these input images at this scale-resolution using the motion estimation algorithm specified in the subsection below and reconstruct the target frame using the estimated motion parameters.
- Evaluate the performance of the model using the metrics: PSNR, Entropy and Time as in (H.Bhaskar and S.Singh, 2006)
- If the frame pair processed is (f_t, f_{t+1}) at $t = 1$ then automatically slide up to a higher resolution and repeat process by incrementing t. Otherwise, if $t > 1$ then if $PSNR_t > PSNR_{t-1}$ then slide down to lower resolution - scale otherwise slide up to higher resolution - scale combination.
- Repeat the process for all frame pairs

The second phase of the algorithm deals with motion estimation. For the purpose of motion estimation we extend the technique of deformable block matching that combines the process of block partitioning, block search and motion modeling. A vector quantization based block partitioning scheme is combined with a genetic algorithm based search method for robust motion estimation (H.Bhaskar and S.Singh, 2006). We extend the basic model in such a way that block deformation is handled using a combined genetic algorithm affine motion model.

The block partitioning phase remains unchanged while the genetic algorithm based block search scheme is altered to include the affine transformations. In the subsection below, a detailed algorithm of the modified block search scheme based on genetic algorithm and affine transforms is presented.

2.1 Vector Quantization Based Block Partitioning

The vector quantization scheme for block partitioning illustrated in (H.Bhaskar and S.Singh, 2006) has been used in the proposed deformable block matching. It is important to realize that the image frames f_t and f_{t+1} that is input to this stage of the algorithm refers to the filtered output of the previous stage. According to the vector quantization scheme, image frames are partitioned based on the information content present within them. The model separates regions of interest based on clustering and places a boundary separating these regions. For this, the vector quantization mechanism uses the gray level feature attributes for separating different image regions and the center of mass of different intersection configurations is employed to deduce the best partition suitable for the image frames.

2.2 Affine-Genetic Algorithm Motion Model

The idea behind the genetic algorithm affine motion model combination is to use the affine transformation equation on every block during fitness function evaluation. The algorithm for the block search scheme is as follows.

The genetic algorithm based block matching algorithm described below is used to match the centroid of any block from the partitioned structure of frame f_t to its successive frame f_{t+1} at different angles theta and parameters shear and scale. The inputs to the genetic algorithm are the block b_t and the centroid (x_c, y_c) of the block.

- Parameter Initialization: The variable parameters of the genetic algorithm will be the genes in the chromosomes. In our experiments they will be the the pixel displacement value in x and y directions, the angle theta of the input block, the shear factor s and scale (r_x, r_y) are encoded as the chromosome $(T_x, T_y, \theta, s, r_x, r_y)$. The translation, rotation and scale parameters of the model are initialized using the phase correlation and log-polar transforms. This speeds up the genetic algorithm search scheme and also increases the accuracy of estimation.

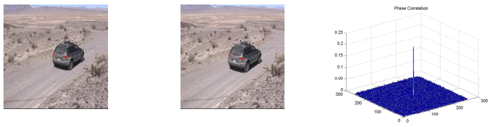

Figure 1: Phase Correlation.

  - Translation parameters using phase correlation: The phase correlation technique is a frequency domain approach to determine the translative movement between two consecutive images. A simple algorithm illustrating the process of determining an approximate translative motion characteristics between two images is as follows.
    - ∗ Consider the input block b_t and its corresponding block at the successive frame b_{t+1}
    - ∗ Apply a window function to remove edge effects from the block images
    - ∗ Apply a 2D Fourier transform to the images and produce $F_t = \Psi(b_t)$ and $F_{b+1} = \Psi(b_{t+1})$; where ψ is the Fourier operator.
    - ∗ Compute the complex conjugate of F_{t+1}, multiply the Fourier transforms element-wise and normalize to produce a normalized cross power spectrum NPS using

$$NPS = \frac{F_t F^*_{t+1}}{|F_t F^*_{t+1}|} \tag{4}$$

    - ∗ Apply inverse Fourier transform on the normalized power spectrum to obtain $PS = \psi^{-1}(NPS)$; where ψ^{-1} is the inverse Fourier operator.
    - ∗ Determine the peak as the the translative coordinates using

$$(\Delta x, \Delta y) = argmax(PS) \tag{5}$$

    - ∗ An illustration describing the process of phase correlation using a sample image is as shown in Figure 1.
  - Rotation and Scale using Log-Polar Transforms: The log-polar transform is a conformal mapping of points on cartesian plane to points on the log-polar plane. The transformation can accommodate an arbitrary rotations and a range of scale changes. If an block image in the cartesian plan is represented using $b(x, y)$, then the log polar transform of the block image with origin O at location (x_o, y_o) is

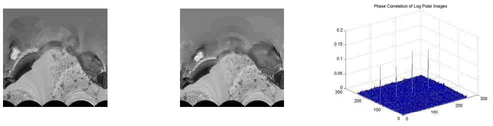

Figure 2: Log Polar Transform.

$$b^*(\psi,\phi) = b\xi(x,y) \tag{6}$$

where,
$\psi = Mlog(r+\alpha)$, α is any constant
$r = \sqrt{(x-x_o)^2+(y-y_o)^2}$ and
$\phi = \tan^{-1}\frac{y-y_o}{x-x_o}$

In order to determine the approximate values of rotation and scale using the log-polar transforms, we convert the image frames into the log polar domain and then use phase correlation between the log-polar images to identify the rotation and scale parameters as in Figure 2.

- Population Initialization: A population P of these n chromosomes representing $(T_x, T_y, \theta, s, r_x, r_y)$ is generated from uniformly distributed random numbers where,
  - $1 \leq n \leq limit$ and $limit$ (100) is the maximum size of the population that is user defined.
  - The values of pre-initialized parameters such as translational, rotational and scale are generated within a small range of their initial value.
- To evaluate the fitness $E(n)$ for every chromosome n:
  - Extract the pixels locations corresponding to the block from frame f_t using the centroid (x_c, y_c) and block size information
  - Affine Transforming these pixels using the translation parameters (T_x, T_y), rotation angle θ, shear factor s and scale r_x, r_y using,
    $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x & 0 & 0 \\ 0 & r_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
    $$\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
  - If b_t represents the original block under consideration, b^*_{t+1} represents the block identified at the destination frame after transformation and (h, w) the dimensions of the block, then the fitness E can be measured as the mean absolute difference (MAD).

$$MAD = \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w}\left|b_t(i,j) - b^*_{t+1}(i,j)\right| \tag{7}$$

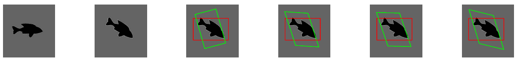

Figure 3: 2D Deformable Block Matching.

- Optimization: Determine the chromosome with minimum error $n_{emin} = n$ where E is minimum. As this represents a pixel in the block, determine all the neighbors (NH_k) of the pixel, where $1 \leq k \leq 8$.
  - For all k, determine the error of matching as in Fitness evaluation.
  - If $E(NH_k) < E(n_{emin})$, then $n_{emin} = NH_k$
- Selection: Define selection probabilities to select chromosomes for mutation or cloning.
- Cross-Over: All chromosomes n_{cr} that are chosen for cross-over are taken into the next generation after swapping one or more random genes between every successive chromosome.
- Mutation: All Chromosomes n_{mu} chosen for mutation are replaced with uniformly distributed random values for centroid, angle, shear, scale and squeeze.
- Termination: Three termination criterion are specified in the proposed model. Check if any condition is satisfied, otherwise iterate until termination.
  - Zero Error: If a chromosome returned an error value zero through fitness evaluation, Or
  - Maximum Generations: If the number of generations (i.e. process loops) exceeds a predefined threshold, Or
  - Stall Generations: If the number of stall generations (i.e. process loops where there is no change in the fitness values) exceeds a predefined threshold.

3 RESULTS AND ANALYSIS

Detailed results and analysis of the proposed model is presented in this section of the paper. On the second part of this section we demonstrate how the motion estimation scheme is adapted to object tracking applications.

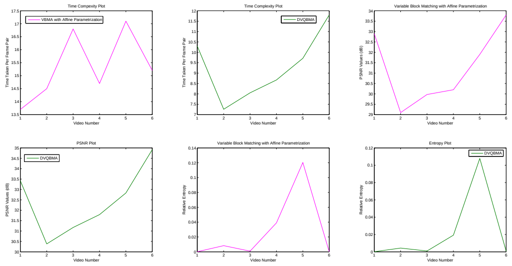

Figure 4: Performance Comparison of Proposed Model to Baselines.

3.1 Performance Evaluation of Motion Estimation

In Figure 3 we illustrate the stages of the proposed block matching scheme. The first and the second images illustrate the original frame and the transformed frame of a sample synthetic image. Through the other images we illustrate how the genetic algorithm is used to identify the optimal motion parameters. Different configurations are evolved through increasing generations getting the solution closer to optimal. The red block represents the objects original position in the anchor frame and the green boundary specifies the location of the block during block searching using genetic algorithm at different generations. To further affirm the performance of the model on different real time datasets, we perform experiments of the model on 6 different video data each containing around 40 frames. The averaged performances on each videos are measured using time, relative entropy and PSNR metrics and compared to the baseline model in Figure 4. The baseline model uses affine parametrization with other search schemes on a variable block partitioned data. We also compare the proposed model against the original block matching model that does not handle deformation and a rotation invariant model.

It is very evident that the averaged time complexity of the proposed motion estimation mechanism that handles deformation still does not match the requirements of real-time. However, with a multi-resolution optimization approach it might well be possible to improve the time efficiency. The results compare well with the quad-tree block matching mechanism with affine parametrization. There is a clear advantage in using the proposed strategy for deformation handling than an extension to any other variable block partitioning scheme with sub-optimal search. The quality of motion estimation is recorded and compared in the graphs. It can be observed that there is clear improvement in the quality of motion estimation when deformation of objects is handled during motion estimation. In comparison to the baseline model, there is clear increase of about 2dB in the PSNR values. A clear increase in the PSNR values can be noted during the progressive improvements in the model from the basic framework to the rotation invariant model and finally to the deformation handling model. This clearly indicates how useful deformation handling is during motion estimation. A very similar trend can also be visualized between different models when compared against the performance metric of relative entropy. The reconstructions made from the deformation handling model match closer to the expected outcome of the image frame. This highlights the accuracy and robustness of the strategy in accomplishing motion estimation. In comparison to the baseline model there is a clear improvement in the values of relative entropy.

3.2 Object Tracking Applications

In this section we describe how the motion estimation mechanism above can be adapted to object tracking applications and also analyze how the efficiency of motion estimation influences the quality of object tracking. We have extended the model for application in object tracking through simple clustering of features characteristics including motion information. To use the proposed model into object tracking motion vectors are clustered such that the moving group of blocks possessing similar motion and feature characteristics will form the object of interest. Trajectories are plotted using the center of mass location of the blocks that constitute the objects. We have tested the approach on a number of different datasets. We have displayed the results of the model on some of them. Figure 5 illustrates the motion trajectory (represented using red dots) of a single/multiple object tracked over different time stamps. As the model does not perform object segmentation, produces a number of small unwanted trajectories that have been removed through manually entered semantic information. The semantic information can be of the form of velocity information of the object in motion, color of the moving objects etc. We have in our experiments displayed the motion trajectory of the group of blocks that have been tracked longest on the image sequences. Generally in any scene this information corresponds to the object of interest. The first two images are the trajectories of the proposed model and a polygon shape feature based nearest neighbor tracking scheme proposed in (H.Bhaskar and S.Singh, 2005). As it can be clearly observed the trajectory of the baseline model

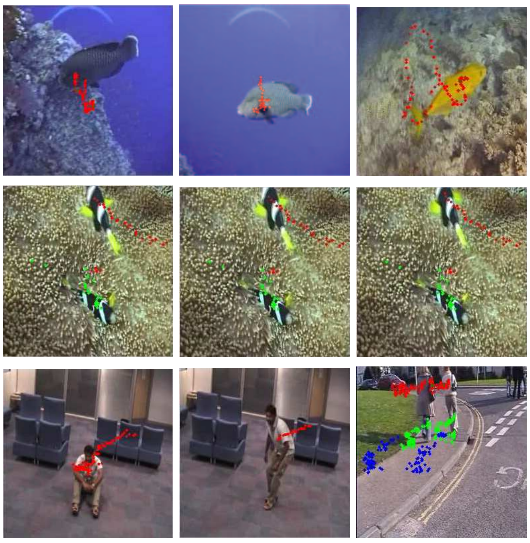

Figure 5: Object Trajectory of Sample Video Sequences.

disintegrates once the object has complex deformation whereas the proposed scheme continues to handle deformation reliably for the entire video. The main reason to this is that the model relies on image segmentation and polygon based shape approximation. The second group of 3 images illustrate the tracking of multiple objects in an image sequence. This sequence is also an example of a very noisy sequence with most of its background moving as well. Finally, examples of human tracking are also illustrated. In the first, we have extracted the trajectory of the body of the person moving and displayed it. The model actually produces different trajectories for moving parts of hands, legs, face etc. as in the next image. The second image is the output of the shape feature based baseline model. Again the technique fails to track objects immediately after a complex deformation is noticed.

4 CONCLUSION

In this paper we presented a novel deformation handling mechanism for block matching based on genetic algorithm that can be extended for use in object tracking applications. The model combines the vector quantization based variable block partitioning and applies an affine based genetic algorithm matching scheme for block matching. We have also presented results on several real time datasets to illustrate the proof of concept. Analysis of the results on the model has proved that the model is robust and reliable for tracking deformational changes in objects in video sequences.

REFERENCES

A. Gyaourova, C. K. and Cheung, S.-C. (2003). Block matching for object tracking. Technical report, Lawrence Livermore Nation Laboratory.

A.Barjatya (2005). Block matching algorithms for motion estimation. Technical report.

C-C.Chang, L-L.Chen, T.-S. (2006). Multi-resolution based motion estimation for object tracking using genetic algorithm. In *VIE Conference*.

C-W.Ting, W.-H. and L-M.Po (2004). Fast block matching motion estimation by recent-biased search for multiple reference frames. In *International Conference on Image Processing (ICIP)*.

F.J.Ferri, J. J. and J.Soret (1998). Variable-size block matching algorithm for motion estimation using a perceptual-based splitting criterion. In *International Conference on Pattern Recognition (ICPR)*, page 286.

H.Bhaskar and S.Singh (2005). Multiple particle tracking in live cell imaging using green fluorescent protein (gfp) tagged videos. In *International Conference on Advances in Patter Recognition*, pages 792–803.

H.Bhaskar, R. and S.Singh (2006). A novel vector quantization based block matching strategy using genetic algorithm search. Submitted to Pattern Recognition Letters.

J.H.Velduis and G.W.Brodland (1999). A deformable block-matching algorithm for tracking epithelial cells. In *Image and Vision Computing*, pages 905–911.

M.Wagner and D.Saupe (2000). Video coding with quadtrees and adaptive vector quantization. In *Proceedings EUSIPCO*.

M.Yazdi and A.Zaccarin (1997). Interframe coding using deformable triangles of variable size. In *International Conference on Image Processing*, page 456.

O.Lee and Y.Wang (1995). Motion compensated prediction using nodal based deformable block matching. In *Visual Communications and Image Representation*, pages 26–34.

Turaga, D. and M.Alkanhal (1998). Search algorithms for block-matching in motion estimation.

Y.Wang and O.Lee (1996). Use of 2d deformable mesh structures for video compression, part i - the synthesis problem: Mesh based function approximation and mapping. In *IEEE Trans. Circuits and Systems on Video Technology*, pages 636–646.

Y.Wang, O. and A.Vetro (1996). Use of 2d deformable mesh structures for video compression, part ii - the analysis problem and a region-based coder employing an active mesh representation. In *IBID*, pages 647–659.