Combination of Spatial and Multiscale Markov Random Field Modeling for Motion Estimation

Stephanus Suryadarma Tandjung, Teddy Surya Gunawan, Chong Man Nang

School of Computer Engineering, Nanyang Technological University
Singapore 639798
[steph, teddy]@sentosa.sas.ntu.edu.sg

***Abstract* ¾ Motion estimation can be formulated and solved as a Bayesian estimation problem. Bayesian estimation requires two probability density function models: observation model and motion field model. The optimization process for this method uses sequential approach, such as simulated annealing, iterated conditional mode, mean field annealing, and highest confidence first. In order to increase the speed of computation and to improve the result, we proposed a combination of the spatial and multiscale Markov random field modeling. The proposed framework can be used as an input to the Bayesian motion estimation. Results indicated one of the possible utilization of our strategy for Bayesian estimation method. Moreover, other strategies can be derived from our method.**

***Keywords* ¾ multiscale smoothing, Bayesian motion estimation, Kalman filter, adaptive matching**

1. INTRODUCTION

Motion is a prominent source of temporal variations in image sequences. However, motion estimation suffers from two prominent problems: the occlusion and the aperture problems. Occlusion refers to the covering/uncovering of surface due to 3-D rotation and translation of object which occupies only part of the field of view. While aperture problem points to the fact that the solution to the motion estimation is not unique. Therefore, this paper introduced a strategy to solve those problems.

Motion estimation algorithm can be viewed to have two different stages: sensing stage and regularization stage, as shown in Fig. 1. The result of motion estimation algorithm (motion vectors) is dependent on the optimization of three factors: true motion vectors, communication overhead, and computation time [7]. Moreover, it is difficult to optimize all factors instead there must be a tradeoff among the three factors.

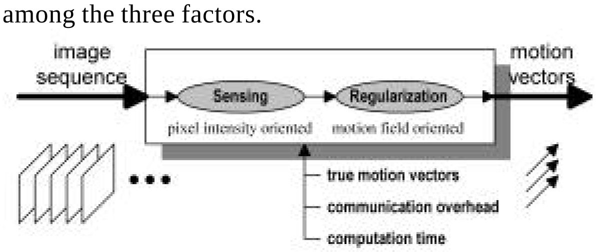

Fig. 1. Motion estimation stages

In Bayesian method, the sensing stage is called observation or likelihood model. While the regularization stage is called prior model. As a sensing stage, the adaptive block size observation model [1] is used. It analyses the unique match from a pixel or a block of a frame with the subsequent frame. Then, the motion vector result is regularized by multiscale regularization stage to achieve smoother motion vectors. The main objectives of this strategy are to increase the speed of optimization processes [3] and to achieve more global result [4] than sequential optimization process.

In our method, the regularization stage is able to receive several types of output that are produced by sensing algorithm, as shown in Fig. 2. Beside the measurement output, the "no measurement" output is also useful for the regularization stage. This will be further elaborated in our experiments.

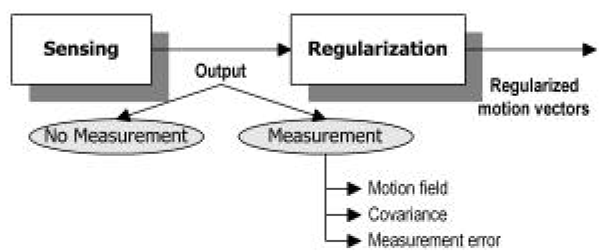

Fig. 2. The type of outputs from sensing stage

This paper describes the combination of spatial and multiscale Markov random field for motion estimation. The overview of adaptive matching algorithm is presented in section 2. The multiscale smoothing algorithm and the integration of matching algorithm-multiscale regularization are explained in section 3 and section 4 respectively. Section 5 includes experimental results, while the last section concludes the paper.

2. ADAPTIVE MATCHING ALGORITHM

An appropriate initialization of motion field can reduce computational cost needed by the stochastic process in Markov Random Field (MRF) based motion estimation. The initialization process is commonly based on the pixel intensity assumption known as the implicit constraint. Therefore, our strategy is to add the implicit constraint into the observation model.

Our model yields motions vector that has been tested as the best candidate for a given pixel position. The observation model is shown on the left term of Equation 1 while the block model on the right term. Both models can have a minimum energy value according to d, the displacement vector and h, the block size. λ are weighting functions, g is the image, and l is the line field.

$$\min_{d,h}\left\{\lambda_g U_g\left(g_t \,|\, d, l, h, g_{t-1}\right) + \lambda_h U_h\left(h \,|\, g_{t-1}\right)\right\} \tag{1}$$

Equation 2 below is the functions of the model.

$$\min_{d,h}\left\{\lambda_g \sum_{x=1}^{h} \frac{\left[g_t(x) - g_{t-1}(x + d(x))\right]^2}{2\sigma^2} + \lambda_g \Theta(h)\right\} \tag{2}$$

where h denotes orders of MRF neighborhood system and σ^2 denotes the variance of pixel intensities and x is a pixel position. The auxiliary variable for "indiscriminate texture" variable is

$$\Theta = \begin{cases} \Theta + 1; & U_g\left(g_t \,|\, \hat{d}_{(i,j,k)}, l, h, g_{t-1}\right) - U_g\left(g_t \,|\, \hat{d}_{(i,j,k-1)}, l, h, g_{t-1}\right) = 0 \\ 0; & \end{cases} \tag{3}$$

This adaptive observation model is used for local motion estimation, such that the implicit information on the beginning of optimization process is exploited. The MRF based regularization technique is applied to the output of sensing stage that used matching based motion estimation. In this technique, the explicit information, as the prior knowledge, is exploited to minimize the matching error on only one motion vector. As a result, the aperture problem is solved and the occlusion location can be detected.

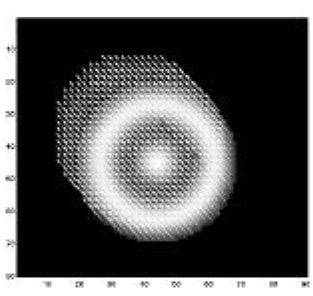

Fig. 3. The motion vector result of the adaptive block size observation model

As an example, Fig. 3 shows the motion vector results and Fig. 4 shows the line field. From those figures, several locations in the middle of moving object that has different motion vectors can be identified. Fig. 5 is motion vector results from a shifted block of a frame of calendar-train sequence. This result shows the result than cannot be achieved by a fixed block size [3].

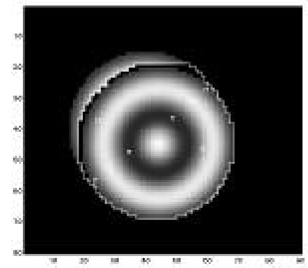

Fig. 4. The line field for the occlusion locations

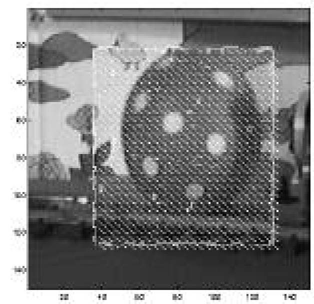

Fig. 5. The motion vector results for synthetic motion from real image

3. MULTISCALE SMOOTHING ALGORITHM

The motion fields on the finest resolution can be smoothed with multiscale smoothing algorithm [6]. For block matching approach, the finest resolution is not motion vector per pixel but motion vector per block. Nevertheless, the motion vector per block resolution can be processed also by this multiscale-smoothing algorithm. Contrary to the fact that the motion vector result can not be smoothed anymore, because of the block-matching algorithm which use implicit smoothing constraint for each pixel.

The multiscale-smoothing algorithm produces the best estimate if given all available measurements. The measurement can be different measurement error or no measurement. In fact, no measurement information can be used to reduce the speed of measurement if it can be detected earlier. For example, the matching process can be done only around the object boundary and high texture object, while the rest pixels can be assign no measurement information. This is can be explained mathematically, consider the multiscale process:

$$x(s) = A(s)x(s\bar{\gamma}) + B(s)w(s) \tag{4}$$

is the process from finest scale s and propagate to higher scale $s\bar{\gamma}$ in quadtree until the lowest scale is reached. $w(s)$ is a zero-mean unit-variance white noise process that is addes to the model with a gain of β. Then the measurement equation:

$$y(s) = C(s)x(s) + v(s) \tag{5}$$

where $v(s)$ is a zero-mean white noise process with covariance $R(s)$. This covariance can be related to the matching algorithm which value is the displacement error or the block size. Moreover, further research is needed to find the most appropriate value.

In order to use this algorithm, the states $x(s)$ must be given at the root node: zero mean with covariance $P(0)$. Lyapunov equation can be used to compute the prior covariances of all states at each node as follows:

$$P(s) = A(s)P(s\bar{\gamma})A^T(s) + B(s)B^T(s) \qquad (6)$$

The core of multiscale algorithm consists of an upward estimation sweep and a downward smoothing sweep. Define the following quantities:

- $Y(s) = \{y(\sigma) | \sigma \text{ is a descendant of } s\}$ is the collection of measurements at all nodes below *s* but not including *s*.
- $\hat{x}(\sigma|s) = E[x(\sigma|\sigma \in Y_s \cup y(s))]$ is the best estimate of $x(s)$ given measurement at node s and all nodes below *s*.
- $\hat{x}(\sigma|s+) = E[x(\sigma|\sigma \in Y_s)]$ is the best estimate of $x(s)$ given measurement at all nodes below *s*.
- $\tilde{P}(\sigma|s) = Cov[x(\sigma) - \hat{x}(\sigma|s)]$
- $\tilde{P}(\sigma|s+) = Cov[x(\sigma) - \hat{x}(\sigma|s+)]$

The upward sweep initializes from the finest level of the prior covariances: $\hat{x}(\sigma|s+) = 0, P(s|s+) = P(s)$ and the upward model can be expressed as

$$x(s\bar{\gamma}) = F(s)x(s) + \bar{w}(s) \qquad (7)$$

and

$$y(s) = C(s)x(s) + v(s) \qquad (8)$$

where

$$F(s) = P(s\bar{\gamma})A^T P(s)^{-1}, E\left[\bar{w}(s)\bar{w}(s)^T\right] = P(s\bar{\gamma}) - F(s)A(s)P(s\bar{\gamma}) = Q(s)$$

The upward sweep computes the best estimate of the state at a node given all measurements at or below that node. The computation takes three steps at each scale :

- *update step*

$$x(s|s) = \hat{x}(s|s+) + K(s)[y(s) - C(s)\hat{x}(s|s+)] \qquad (9)$$

$$P(s|s) = [I - K(s)C(s)]P(s|s+) \qquad (10)$$

$$K(s) = P(s|s+)C^T(s)\left[C(s)P(s|s+)C^T(s) + R(s)\right]^{-1} \qquad (11)$$

The updated estimate is the best estimate of $x(s)$ given all measurements at or below *s*.

- *prediction step*

$$\hat{x}(s|s\alpha_i) = F(s\alpha_i)\hat{x}(s\alpha_i|s\alpha_i) \qquad (12)$$

$$P(s|s\alpha_i) = F(s\alpha_i)P(s\alpha_i|s\alpha_i)F^T(s\alpha_i) + Q(s\alpha_i) \qquad (13)$$

The predicted estimate is the best estimate of $x(s)$ given all measurement at node $s\alpha_i\,(i = 1 \ldots q)$ or below.

- *merge step*

$$\hat{x}(s|s+) = P(s|s+)\sum_{i=1}^{q} P^{-1}(s|s\alpha_i)\hat{x}(s|s\alpha_i) \qquad (14)$$

$$P(s|s+) = \left[(1-q)P(s)^{-1} + \sum_{i=1}^{q} P^{-1}(s|s\alpha_i)\right]^{-1} \qquad (15)$$

The merge step combines the predicted estimate of $x(s)$ given each of its child subtrees. The merged estimate is the best estimate of $x(s)$ given all measurement below node *s*.

The downward sweep computes the best estimate of the states at a node given all available measurement everywhere on the tree:

$$\hat{x}(s|0) = \hat{x}(s|s) + J(s)[\hat{x}(s\bar{\gamma}|0) - \hat{x}(s\bar{\gamma}|s)] \qquad (16)$$

$$P(s|0) = P(s|s) + J(s)[P(s\bar{\gamma}|0) - P(s\bar{\gamma}|s)J(s)] \qquad (17)$$

$$J(s) = P(s|s)F^T P^{-1}(s\bar{\gamma}|s) \qquad (17)$$

Equation 16 is the final result (motion vectors) of the proposed motion estimation algorithm.

4. MATCHING ALGORITHM-MULTISCALE REGULARIZATION INTEGRATION

Motion estimation can be divided into two main stages, sensing stage and regularization stage, as explained in Section 1. The sensing stage can be optical flow algorithm, block matching algorithm or adaptive matching algorithm. We emphasize on adaptive matching algorithm, because of its feature that has variable block size. The block size is recursively computed at some extent until a motion vector with a unique measurement error value is achieved. This approach can be thought as a variable aperture size for the sensing algorithm. It is also possible to input the result of different resolution on quadtree or multiscale smoothing algorithm. At this moment, we put the measurement result at the finest resolution.

After adaptive matching algorithm has been executed as a sensing stage, the most possible matching for the motion fields based on intensity can be achieved. The smoothing algorithm can be further applied to achieve smoother motion vectors. Moreover, we used multiscale-smoothing algorithm. This will ensure that the motion vector results can be treated as Markov random field's modeling in scale.

If fixed block matching is used as a parameter to the multiscale smoothing, the motion vector at the finest resolution can not be computed using the fixed block size. However, the motion vector at coarser levels can be computed using the fixed block size as a input parameter. This is due to the implicit smoothness that has been applied in the development of motion field.

5. EXPERIMENTAL RESULTS

Experiments were performed on the synthetic images, as shown in Fig. 6. The synthetic image is a moving black object in the white background. We designed the synthetic image such that the black object is shifted several pixels apart and has no texture.

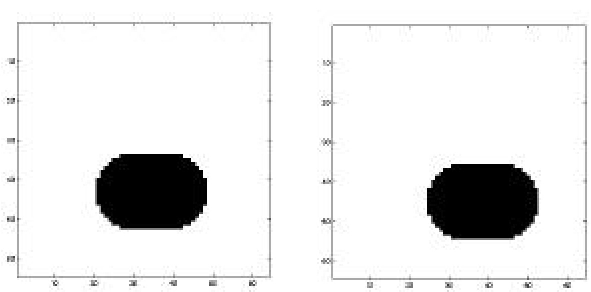

Fig. 6. The synthetic image

We applied the synthetic image to the adaptive matching algorithm. Fig. 7 (a) shows the difference between successive frames. And Fig 7 (b) shows the motion vector result of the algorithm. It can be noticed from this result that the algorithm can produce motion vector only where the difference exists. However, the human perception can not agree with such result, because we know that a whole object is moving, not only where the difference exists. In order to handle this problem, we added segmentation information to the multiscale regularization and gave "no measurement" information to the pixel locations that have zero motion vector but actually belong to the moving object.

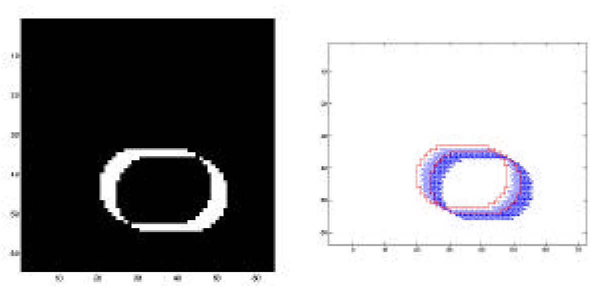

Fig. 7. The difference image and the adaptive matching motion vector results

The output from the sensing stage with "no measurement" information is applied to the regularization stage. Furthermore, the output from the efficient multiscale regularization algorithm is applied to Bayesian motion estimation to achieve true motion vectors. Therefore, Bayesian motion estimation is combined with observation model, displacement model for smoothness and line model for discontinuity. Fig. 8 shows the final result of our proposed scheme.

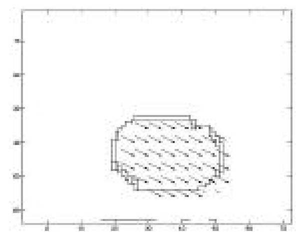

Fig. 8. The motion vector results of our proposed scheme

6. CONCLUSIONS

We have presented the theoretical scheme and the implementation for the combination of spatial and multiscale Markov Random Field modeling for motion estimation. The synthetic image was used to verify our scheme. Results indicated that our scheme could handle two prominent motion estimation problems, the occlusion and the aperture problems. Moreover, more strategies can be derived from our proposed scheme to further improve the motion vector results, such as intelligence interpretation and other higher level algorithms.

REFERENCES

[1] Stephanus S. T., M. N. Chong, and K. K. Wong, "Adaptive blocksize observation model for MRF modeling of motion field," *IEEE Int. Conf. On Information, Communications and Signal Processing*, vol. CD ROM ICICS'99, December 1999.

[2] K. C. Chou, A. S. Wilsky, and A. Benveniste, "Multiscale recursive estimation, data fusion, and regularization," *IEEE Trans. On Automatic Control*, vol. 39, pp. 464-478, March 1994.

[3] Stephanus S. T., T. S. Gunawan, and M. N. Chong, "Motion estimation using adaptive matching and multiscale method," *SPIE Conf. On Visual Communication and Image Processing*, June 2000.

[4] Stephanus S. T., T. S. Gunawan, and M. N. Chong, "Multiscale motion estimation," *IECI Conf. On Multimedia and Network*, pp. 260-264, March 2000.

[5] E. Dubois and J. Konrad, "Bayesian estimation of motion vector fields," *IEEE Trans. on Pattern Analysis and Machine Intelligent*, vol. 14, pp. 910-927, September 1992.

[6] T. T.-J. Jo, *Multiscale Modeling and Estimation of Large-Scale Dynamic Systems*, Ph.D. thesis, Massachusetts Institute of Technology, September 1998.

[7] F. Dufaux, F. Moscheni, "Motion estimation techniques for digital TV : a review and a new contribution," in *IEEE Proceeding*, Vol. 83, pp. 858-876, June 1995.