EFFICIENT GLOBAL MOTION ESTIMATION USING FIXED AND RANDOM SUBSAMPLING PATTERNS

H. Alzoubi, W. D. Pan

Dept. of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, AL 35899, USA
Email: <alzoubh, dwpan>@ece.uah.edu

ABSTRACT

Global motion generally describes the motion of the camera, although it may comprise large object motion. The region of support for global motion representation consists of the entire image frame. Therefore, estimating global motion parameters tends to be computationally costly due to the involvement of all the pixels in the calculation. Efficient global motion estimation (GME) techniques are sought after in many applications such as video coding, image stabilization and super-resolution. In this paper, we propose to select only a small subset of the pixels in estimating the global motion parameters, based on a combination of fixed and random subsampling patterns. Simulation results demonstrate that the proposed method was able to speed up the conventional all-pixel GME approach by up to 7 times, without significant loss in the estimation accuracy. The combined subsampling patterns were also found to provide better motion estimation accuracy/complexity tradeoffs than those achievable by using either fixed or random patterns alone.

Index Terms— Global motion estimation (GME), Levenberg-Marquardt algorithm, subsampling, perspective model, computational complexity

1. INTRODUCTION

Global motion estimation has been widely used in many applications, e.g., video coding, image stabilization, and super-resolution. Tools based on global motion estimation (GME) have been adopted by the MPEG-4 standard (advanced simple profile) [1]. If real time MPEG-4 coding/decoding is required, then accurate and very fast GME is of great importance. Another example is image stabilization techniques widely used in consumer camera and camcorder products to remove undesirable shakes or jiggles, and to provide a much less irritating viewing experience. A general image stabilization algorithm is composed of a global motion estimation module, a motion compensation (or correction) module (GMC), and an image composition (IC) module. GME estimates the global motion between images, and sends the motion parameters to MC, which computes the global transformation necessary to stabilize the current frame. IC then warps the current frame according to that transformation, generating the stabilized image sequence. Accurate and fast global motion estimation is thus important since the motion estimation accuracy directly affects the motion correction performance of the stabilization system in a real-time system. GME has also found applications in resolution enhancement of images. Super-resolution reconstruction is defined as the process of combining multiple low-resolution images to form a higher resolution image. Most of the super resolution image reconstruction methods proposed in the literature consist of three stages: registration, blur estimation, and refinement [2]. Registration is the process of computing and compensating for image motion, where global motion estimation plays a key role. However, as we will see in the next section, GME is intrinsically expensive computationally.

2. GLOBAL MOTION ESTIMATION

In GME involving two image frames I_k and I_{k+1}, one seeks to minimize the following sum of squared differences between I_{k+1} and its predicted version $\acute{I}_{k+1}$, which is obtained after transforming all the pixels in I_k.

$$E = \sum_{all\ i} \sum_{all\ j} e^2(i,j), \tag{1}$$

where $e(i,j)$ is the luminance difference between a pixel with coordinate (i,j) in I_{k+1} and the corresponding pixel in the predicted frame $\acute{I}_{k+1}$:

$$\begin{aligned} e(i,j) &= I_{k+1}[i,j] - \acute{I}_{k+1}[i,j] \\ &= I_{k+1}[i,j] - I_k[x(i,j), y(i,j)]. \end{aligned} \tag{2}$$

The transform mapping functions $x(i,j)$ and $y(i,j)$ should be so chosen that E in (1) is minimized. For example, if the perspective motion model is employed, as in MPEG-4 [1], for estimating the global motion in a video sequence, then the mapping function consists of eight motion parameters, m_1 through m_8, as described below.

$$x(i,j) = \frac{m_1 i + m_2 j + m_3}{m_7 i + m_8 j + 1}, \tag{3}$$

$$y(i,j) = \frac{m_4 i + m_5 j + m_6}{m_7 i + m_8 j + 1}. \tag{4}$$

The well-known Levenberg-Marquardt algorithm (LMA) [3] can be used to iteratively estimate the vector $\boldsymbol{m} = [m_1, m_2, ..., m_8]$ that minimizes E in (1). In LMA, each iteration is given by

$$\boldsymbol{m}^{(n+1)} = \boldsymbol{m}^{(n)} + \boldsymbol{s}^{(n)}, \tag{5}$$

where $\boldsymbol{s}^{(n)}$ is an update (during the n-th iteration) that can be found by solving the linear equation

$$\left[\boldsymbol{J}^T(\boldsymbol{m}^{(n)})\boldsymbol{J}(\boldsymbol{m}^{(n)}) + \mu^{(n)}\boldsymbol{I}\right] \boldsymbol{s}^{(n)} = -\boldsymbol{J}^T(\boldsymbol{m}^{(n)})\boldsymbol{r}(\boldsymbol{m}^{(n)}), \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix, and μ is a nonnegative scalar parameter. $\boldsymbol{r}(\boldsymbol{m})$ is a column vector given in (7).

$$\boldsymbol{r}(\boldsymbol{m}) = \left[\ e(1,1),\ e(1,2),\ ...,\ e(2,1),\ e(2,2),\ ...,\ all\, i,\ j\ \right]^T. \tag{7}$$

$\boldsymbol{J}(\boldsymbol{m})$, as given in (8), is the Jacobian matrix of $\boldsymbol{r}(\boldsymbol{m})$.

$$\boldsymbol{J}(\boldsymbol{m}) = \begin{bmatrix} \frac{\partial e(1,1)}{\partial m_1} & \frac{\partial e(1,1)}{\partial m_2} & ... & \frac{\partial e(1,1)}{\partial m_8} \\ \frac{\partial e(1,2)}{\partial m_1} & \frac{\partial e(1,2)}{\partial m_2} & ... & \frac{\partial e(1,2)}{\partial m_8} \\ \frac{\partial e(1,3)}{\partial m_1} & \frac{\partial e(1,3)}{\partial m_2} & ... & \frac{\partial e(1,3)}{\partial m_8} \\ ... & ... & ... & all\, i,\ j \end{bmatrix} \tag{8}$$

Each entry in $\boldsymbol{J}(\boldsymbol{m})$ can be evaluated as follows.

$$\left[\frac{\partial e(i,j)}{\partial m_k}\right] = \left[\frac{\partial e(i,j)}{\partial x}\right] \cdot \frac{\partial x}{\partial m_k}, \tag{9}$$

$$\left[\frac{\partial e(i,j)}{\partial m_k}\right] = \left[\frac{\partial e(i,j)}{\partial y}\right] \cdot \frac{\partial y}{\partial m_k}, \tag{10}$$

where $k = 1, 2, \ldots, 8$. In turn, $\frac{\partial x}{\partial m_k}$ and $\frac{\partial y}{\partial m_k}$ can be evaluated from (3) and (4), respectively. For example, it can be readily shown that

$$\begin{aligned} &\frac{\partial x}{\partial m_1} = \frac{i}{D}, \frac{\partial x}{\partial m_2} = \frac{j}{D}, \frac{\partial x}{\partial m_3} = \frac{1}{D}, \\ &\frac{\partial y}{\partial m_4} = \frac{i}{D}, \frac{\partial y}{\partial m_6} = \frac{1}{D}, \frac{\partial y}{\partial m_8} = -\frac{yj}{D}, \end{aligned} \tag{11}$$

where

$$D = m_7 i + m_8 j + 1. \tag{12}$$

From the expressions of $\boldsymbol{r}(\boldsymbol{m})$ and $\boldsymbol{J}(\boldsymbol{m})$ in (7) and (8), we can see that the LMA operates on all the pixels within an image frame. For each pixel, equations (9)-(11) must be calculated, thereby making the global motion estimation a very computationally intensive process.

3. SUBSAMPLING PATTERNS

The complexity of the GME can be reduced significantly if only a small subset of pixels is used in estimating the motion parameters. However, using too few pixels in the calculation may cause severe degradation in the accuracy of motion estimation. Ideally, for a given size of the subset, one should select the subset of pixels that best represent the global motion to be estimated. Nonetheless, the search for such a good subset may incur additional computational complexity, which may defeat the very purpose of reducing the computational complexity of the GME.

In [4], a subset selection criterion based on gradient magnitudes was proposed. In the selection process, the image is divided into small regions, where the top 10% pixels with the largest gradient magnitudes will be selected. While this method is effective in reducing the overall complexity, its overhead cannot be ignored. For example, the gradient of each pixel needs to be calculated, followed by an expensive sorting operation required to reveal those top 10% pixels. In addition, a comparison has to be made for each pixel to see if it lies within the top 10% or not. On the other extreme, a random subset selection method was proposed in [5] for GME in fast image-based tracking. This is a rather simple method, with the extra overhead for subset selection as low as the cost of generating a random bitmap, which is of the same size of an image frame. However, since the positions of the selected pixels are random, numerical instabilities might result.

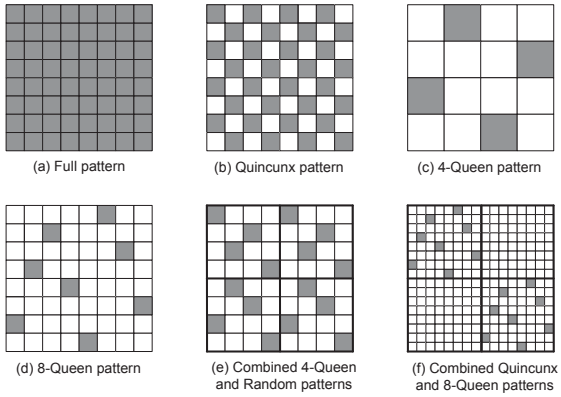

Fig. 1. Various subsampling patterns for choosing a subset of pixels for GME. (a) All the pixels are chosen. (b) The Quincunx pattern with a subsampling ratio of $1/2$. (c) 4Q pattern with a subsampling ratio of $1/4$. (d) 8Q pattern with a subsampling ratio of $1/8$. (e) RD-4Q is a combination of random and fixed patterns. (f) Quin-8Q is a combination of two fixed patterns.

Pixel selection can also follow certain fixed subsampling pattern. Since neighboring pixels are very likely to experience the same kind of motion, we can choose only one pixel as the representative of a group of pixels in calculating the motion parameters. Fixed subsampling patterns were found to be effective for complexity reduction of both the local [6] and global motion estimation [7]. However, the accuracy of motion estimation would suffer if a large subsampling ratio is used [7]. Recently, an N-Queen decimation lattice has been used to select a subset of pixels, which resulted in much faster local motion estimation [8]. The N-queen decimation lattice (see Fig. 1) can improve the representation by holding only one pixel from each row, thus the spatial information is represented in all directions [8]. The N-queen pattern has the following advantages. First, the selected pixels are with regular distances away from each other, and all the pixels are uniformly distributed. Thus the image can be better reconstructed by using the selected pixels, resulting in a good motion estimation accuracy. Second, the pattern can be applied hierarchically. Third, the pattern can be easily combined with other patterns for further complexity reduction and more accurate motion estimation. For example, in Fig. 1(e), a combination of the random and 4-Queen patterns is shown, while (f) shows a hierarchical application of a Quincunx pattern followed by an 8-Queen pattern. In the RD-4Q pattern (Fig. 1(e)), four pixels are randomly selected for each 4×4 block, albeit with the constraint that no more than one selected pixel occupies the same row or the same column of the block. Obviously, the overhead for this subset selection method based on combined patterns is very low. In this paper, we apply the subsampling method that combines random and fixed subsampling patterns to global motion estimation. Experiment results showed that the combined subsampling patterns could provide significantly improved tradeoffs between motion estimation accuracy and complexity than those achievable by using either fixed or random patterns alone.

4. GME BASED ON PIXEL SUBSAMPLES

Operating on a subset of pixels that are selected based on a certain subsampling pattern, the proposed GME algorithm can be summarized by the following steps:

1. Obtain a coarse estimate of the translational components of the motion model by using a fast 3-step search method [9].
2. A threshold T is initialized to a large number (e.g., 255), to be used for outlier rejection. Similar to policy adopted in [9], 10% of the chosen pixels with the largest errors are excluded.
3. Select pixels according to one of the subsampling patterns shown in Fig. 1. For each selected pixel at location (i, j), compute its corresponding position (x, y), using (3) and (4).
4. Compute the error $e(i, j)$, using (2). If $e < T$, include this pixel in the calculation of $\boldsymbol{r}(\boldsymbol{m})$ and $\boldsymbol{J}(\boldsymbol{m})$, given in (7) and (8). If this is the first iteration, a histogram of $|e(i, j)|$ is constructed.
5. Solve the linear equation (6) and update the motion parameters using (5).
6. In the first iteration only, T is re-calculated to exclude the top 10% of the histogram.
7. Steps 3, 4, and 5 are repeated for a maximum of 32 iterations. The process stops earlier if the update term $\boldsymbol{s}^{(n)}$ in (5) is smaller than a threshold of 0.001 for the translational component of the motion parameters, and 0.00001 for the other parameters.

5. SIMULATION RESULTS

We tested the GME methods on 11 video sequences, including "Carphone", "Claire", "Containership", "Foreman", "Miss America", "Mobile and Calendar", "Mother Daughter", "Salesman", "Silent", "Tempete", and "Tennis", each of which contains 250 frames, except for "Miss America" (150 frames). These video sequences can be found at [10]. GME is applied on selected pixels obtained by using those subsampling patterns depicted in (c), (d), (e), and (f) of Fig. 1 (abbreviated as 4Q, 8Q, RD-4Q, and Quin-8Q, respectively). The Peak Signal-to-Noise Ratio (PSNR) as a measure of the accuracy of motion estimation, and the computation time were adopted as the performance metrics. Comparisons were made against other three GME methods, including the conventional GME using all the pixels (abbreviated as FS, or full size), GME based on subsampling using gradient magnitudes as the criteria (denoted by GR) [4], as well as GME based on a random subsampling pattern (denoted by RD). Simulations were conducted on a PC with 3.0 GHz Pentium IV processor, 512 MB RAM, and an MS Windows XP OS. The source codes were written in MATLAB.

Simulation results are summarized in Table 1 and Table 2. Note that differences in Table 2 between the computation times for the sequences "Mobile and Calendar", "Tempete", and "Tennis", which are in the CIF format, and those for the remaining eight sequences, which are in the QCIF format. Table 1 shows that, as expected, all GME methods based on pixel subsampling cause varying degrees of degradation in the accuracy of motion estimation, as opposed to the conventional full-data GME method. However, the losses in accuracy are very small. From Table 1, we can see that GME based on a subset of pixels chosen according to fixed patterns comes very close to the all-pixel GME in terms of PSNR.

For a fair comparison between these partial-data GME methods, we calculated the speedup / accuracy degradation ratios (SADR) obtained by dividing the entries in the bottom row of Table 2 with the corresponding entries in the bottom row of Table 1. We can see the GR method can ensure a good accuracy, but its speedup is too low (1.3 times) due to its high overhead, even with a high subsampling factor (0.04). Therefore, its overall performance (as measured by the SADR) is very poor compared with those N-queen patterns. The RD method also has a similarly poor overall performance, mainly due to its severe accuracy loss. The RD-4Q method ranks the highest in SADR, with very high accuracy (0.02 dB below the FS method) and a modest speedup ratio (3.6). The advantage of a hierarchical combination of the N-queen with a random pattern becomes obvious when we compare the 4Q method with the RD-4Q method, after noting that both methods use the same subsampling ratio. The 4-queen and 8-queen patterns can achieve fairly impressive speedup factors with a reasonable prediction accuaracy (the average degradations in PSNR are -0.03 dB and -0.06 dB respectively). Although the Quin-8D method is the fastest (with a speedup ratio of 7.1), its overall performance is lower than that of the 8Q method. This can be explained by the subsampling pattern in Fig. 1(f), where the selected pixels are not as evenly distributed as those in Fig. 1(d).

6. CONCLUSION

This paper demonstrated that global motion estimation (GME) could be substantially accelerated by using the pixel-subsampling approach, based on a combination of fixed and random patterns. With sufficiently high accuracy, these fast GME methods would be suitable for many real-time motion estimation applications.

7. REFERENCES

[1] "MPEG-4 video verification model version 18.0," in: ISO/IEC JTC1/SC29/WG11 N3908, Pisa, Italy, 2001.

[2] A. Bovik, *Handbook of Image & Video Processing,* Academic Press, pp. 259-267, 2000.

[3] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing,* Second Edition. Cambridge University Press, pp. 656-706, 2002.

[4] Y. Keller and A. Averbuch, "Fast gradient methods based on global motion estimation for video compression," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 13, no. 4, pp. 300-309, April 2003.

[5] F. Dellaert and R. Collins, "Fast image-based tracking by selective pixel integration," *ICCV Workshop on Frame-Rate Vision,* Corfu, Greece, September 1999.

[6] Y.-L. Chan and W.-C. Siu, "New adaptive pixel decimation for block motion vector estimation," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6, no. 1, pp. 113-118, February 1996.

[7] H. Alzoubi and W. D. Pan, "Very fast global motion estimation using partial data," in Proc. of *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1189-1192, Honolulu, HI, April 2007.

[8] C.-N. Wang, S.-W. Yang, C.-M. Liu, and T. Chiang, "A hierarchical N-Queen decimation lattice and hardware architecture for motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 14, no. 4, pp. 429-440, April 2004.

[9] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Trans. Image Processing,* vol. 9, no. 3, pp. 497-501, March 2000.

[10] Collection of test video sequences: http://media.xiph.org/video/derf.

Table 1. Average PSNR (in dB). Except for the FS (full-data) method, only the PSNR degradations with respect to the FS method are shown for other methods, for ease of comparison.

Sequence	FS	GR	RD	4Q	8Q	RD-4Q	Quin-8Q
Carphone	32.64	-0.07	-0.89	-0.03	-0.04	-0.01	-0.14
Claire	43.65	-0.01	-0.55	-0.03	-0.12	-0.04	-0.32
Containership	44.57	-0.11	-0.21	-0.01	-0.06	-0.02	-0.09
Foreman	29.69	-0.05	-0.43	-0.04	-0.07	-0.03	-0.12
Miss America	43.19	-0.12	-0.47	-0.10	-0.24	-0.01	-0.06
Mobile and Calendar	26.08	-0.01	-0.04	-0.01	-0.02	-0.01	-0.06
Mother and Daughter	41.16	-0.09	-0.36	-0.02	-0.05	-0.03	-0.07
Salesman	39.27	-0.06	-0.09	-0.01	-0.02	-0.01	-0.04
Silent	32.91	-0.06	-0.19	-0.02	-0.03	-0.01	-0.06
Tempete	27.87	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
Tennis	27.40	-0.04	-0.22	-0.03	-0.04	-0.04	-0.06
Average degradation	0.00	-0.06	-0.32	-0.03	-0.06	-0.02	-0.09

Table 2. Computation times (in seconds) of the GME methods and their average speedup ratios over the FS method.

Sequence	FS	GR	RD	4Q	8Q	RD-4Q	Quin-8Q
Carphone	212.1	192.7	34.4	63.8	46.4	65.0	36.4
Claire	211.1	195.3	34.5	64.9	46.2	63.3	37.1
Containership	214.8	193.4	34.5	65.0	45.9	63.1	37.1
Foreman	211.5	194.8	34.6	64.9	45.5	64.2	37.2
Miss America	136.2	118.1	20.8	39.8	27.8	38.8	22.1
Mobile and Calendar	1571.3	915.4	145.8	424.6	202.9	359.5	154.0
Mother and Daughter	213.2	194.3	35.4	64.1	45.1	65.0	37.2
Salesman	212.1	194.2	34.3	63.7	45.4	64.9	35.8
Silent	210.3	192.6	34.2	64.6	44.7	63.6	36.8
Tempete	1588.9	937.5	148.9	412.4	202.7	356.8	149.6
Tennis	1560.3	962.1	148.2	411.8	202.1	350.8	150.4
Average Speedup	1.0	1.3	7.4	3.4	5.5	3.6	7.1

Table 3. Speedup / accuracy degradation ratios (SADR) of the GME methods.

	GR	RD	4Q	8Q	RD-4Q	Quin-8Q
Subsampling Ratio	1/25	1/25	1/4	1/8	1/4	1/16
SADR	21.67	23.12	113.33	91.66	180	78.89