Progressive Model Refinement Global Motion Estimation Algorithm for Video Coding

Haifeng Wang, Jia Wang, Qingshan Liu, Hanqing Lu
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences (CAS)
(hfwang, wangjia, qsliu, luhq)@nlpr.ia.ac.cn

Abstract: This paper presents a Progressive Model Refinement (PMR) method for Global Motion Estimation (GME) in MPEG-4 video coding. Our contributions consist of two aspects. Firstly, a method of feature point selection is proposed based on the analysis of spatial distribution. It can effectively guarantee the number of feature point won’t become too large and avoid most feature points congregated on a small region. Secondly, a PMR algorithm is proposed to select motion models progressively according to the complexity of the camera motion, which improves the convergence performance of GME and makes the PMR algorithm much more robust and faster than single-model based GME algorithms. Experiments show that the presented algorithm can always select the appropriate model to describe the camera motion.

Key words: progressive model refinement global motion estimation motion model

1. Introduction

The goal of GME is to find the motion of the camera. It is often the first step of video segmentation, sprite coding and other applications, and it is also a powerful tool widely used in video processing and compression as well as in computer vision areas [1]. For video compression, the Global Motion Model (GMM) is described as the motion trajectories of some reference points [1]. The three popular models used in MPEG-4 are translational model, affine model and perspective model, among which 6-parameter affine model is widely used because it can deal with the majority of motion types, such as panning, rotation and zooming, encountered in video coding [2].

The GME method adopted in MPEG-4 is the feature-based fast and robust GME technique (FFRGMET) [3], which performed upon a three-level pyramid [4]. Iterative approaches are used to estimate the model parameters at each level. Although this method significantly improves the performance of the GME method used in MPEG-4 VM, it is still too complex to reach a real-time level. On seeing this, many fast algorithms were proposed to further speed up the GME procedure in recent years. For example, Chan [5] proposed two techniques: motion vector prediction and early termination to fast the parameter calculation. Huang [6] proposed to use the cross-points as feature points (FPs) to calculate the model parameters. Most of these improvements utilize one motion model, usually the affine model and put their effort on the selection of FPs or the reduction of the computation complexity of iteration. However, if the camera motion is beyond the representation scope of affine model, the GME parameters obtained by these methods can not describe the camera motion precisely. In the other hand, if the camera only undergoes a translational motion, using affine model is sure to be too complex and time consuming.

In this paper, we proposed a Progressive Model Refinement (PMR) algorithm to select the motion model progressively according to the complexity of the camera motion. PMR works on the three-level pyramid GME structure in MPEG-4 VM. The models adopted in our PMR method include the translational model, affine model and perspective model. To further reduce the number of FPs without affecting the precision of motion parameters, we proposed a new feature point selection method based on the spatial distribution.

The rest of the paper is organized as follows. In Section 2, the feature point selection method based on spatial distribution is fully discussed. In Section 3, we present the proposed PMR method. Section 4 contains the experiment results and comments. Finally, a conclusion is given in Section 5.

2. Feature Point Selection (FPS)

In video coding, local motion means the displacements of individual objects composing the scene [6]. When pixels undergoing local motion are involved in the GME calculation, the iteration number will increase and the parameters obtained may not describe the camera motion precisely. These pixels are referred to as outliers in GME and should be excluded from the parameter calculation. Next, the outlier detection methods used in FFRGMET is briefly discussed.

2.1. Outlier Detection Methods Used in FFRGMET

FFRGMET algorithm employs two methods to exclude outliers from the GME calculation [6] [7]:

1. On the intermediate level and base level of the pyramid, pixels satisfying the following formula are selected.

$$\left\{(x,y)\left|\left(T_{up}\cdot E\left(\left|I_x\right|+\left|I_y\right|\right)\right)>\left(\left|I_x\right|+\left|I_y\right|\right)>\left(T_{down}\cdot E\left(\left|I_x\right|+\left|I_y\right|\right)\right)\right.\right\}$$

$$AND\left(\left|I_t\right|>T_t\cdot E\left(I_t\right)\right) \qquad (1)$$

I_x and I_y are the spatial components of luminance. I_t is the temporal gradient of luminance. $E(x)$ is the mean value of the set of x .These two conditions can assure that only pixels belong to motion edge are used in the intermediate and base levels of pyramid calculation.

2. Residual block based outlier detection method is used at the base level. The pixels belonging to the foreground appear to show large residuals and concentrate to a compact region. Removing pixels in this kind of region can help to locate the outliers more accurately.

In order to further reduce the number of FPs, we proposed a FPS method based on the spatial distribution.

2.2. Spatial Distribution Based FPS (SDBFPS)

A similar approach based on the spatial distribution of FPs was proposed in [8], in which the whole image is divided into 100 sub-regions and the top 10% in each region are chosen. Our SDBFPS method improves this idea. It divides FPs obtained by applying formula 1 into 4 groups according to their spatial location, as shown in figure 1. Different regions are treated in different ways.

Region 1 is split into 4*4 sized blocks and region 2 and 3 are split into 8*8 sized blocks. For region 1 and region 2, one FP with the largest spatial gradient in each block is kept. For region 3, two FPs are remained. Experiments show that FPs in region 4 are most likely out of the range of the image, so all the FPs in region 4 are removed.

This SDBFPS algorithm can be used in any traditional single-model based GME method. As for the three-level GME in MEPG-4 VM, it is preformed on the bottom level. The FPs obtained is then down-sampled with the raw Y-component data and used as the input for the top two levels.

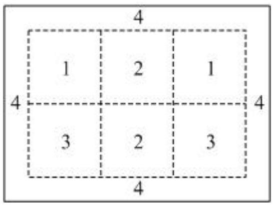


Figure 1 Spatial distribution

Table 1 shows the number of FPs of basketball sequences with $(n-2)^{th}$ frame as the reference frame. We can see that by using the SDBFPS algorithm, the number of FPs will reduce by 75 percent with PSNR drops about 0.03.

Table 1 Figure points ($\times 1000$)

Frame	5	10	15	135	140	145
A	27.9	29.5	30.9	39.9	43.2	44.9
B	6.6	7.1	7.6	8.1	9.0	9.3

A: Affine Model; B: Affine Model with SDBFPS

3. Progressive Model Refinement (PMR)

Nowadays, most GME algorithms use the affine model, including FFRGMET. There are two disadvantages for using only one model for all sequences. Firstly, if the camera undergoes simple motions, like translational motion, taking affine model may seem much more computation complicated. Secondly, affine model can not describe the motion precisely if the camera motion is too complex and beyond the description scope of the affine model. So it becomes obvious that there should be an adaptive model selection method.

The proposed algorithm works on the three-level GME structure. Model refinement (MR) takes place between levels. Three motion models, that is, translational model, affine model and perspective model are used in our algorithm. The top level always utilizes the translational model. Whether the motion models used in the intermediate and the bottom levels will remain the same or upgrade to the affine model or the perspective model depends on the GMC results. The progressive here means MR will follow an order from translational model to affine model and to perspective model. The motion model won’t jump from the translational model directly to the perspective model.

The MR procedure between the top level and the intermediate level is as follows:

1. Project the motion parameters calculated in the top level to the intermediate level.
2. Perform GMC to get the residual between the current frame and the reconstructed frame.
3. Split the residual into 4*4 sized blocks and get the sum of each block. The total block number is denoted as S_{total}
4. Calculate the number of outlier blocks detected by the residual block based outlier detection method mentioned in Section 2.1 and denote it as $S_{outlier}$
5. Use $S_{background}$ denote the number of blocks that FPs mainly locate in:
$$S_{background} = S_{total} - S_{outlier}$$
6. Calculate the number of blocks belonging to $S_{background}$ whose value is smaller than a predefined threshold T_{block} and denote it as S_{match}.
7. Give threshold T_{match}. If
$$S_{match} < T_{match} \cdot S_{background},$$
it means the current motion model can not represent the camera motion precisely and should be transferred to some more complicate model, that is, affine model for translational model and perspective model for affine model. Otherwise, the motion model remains unchanged for the next level.

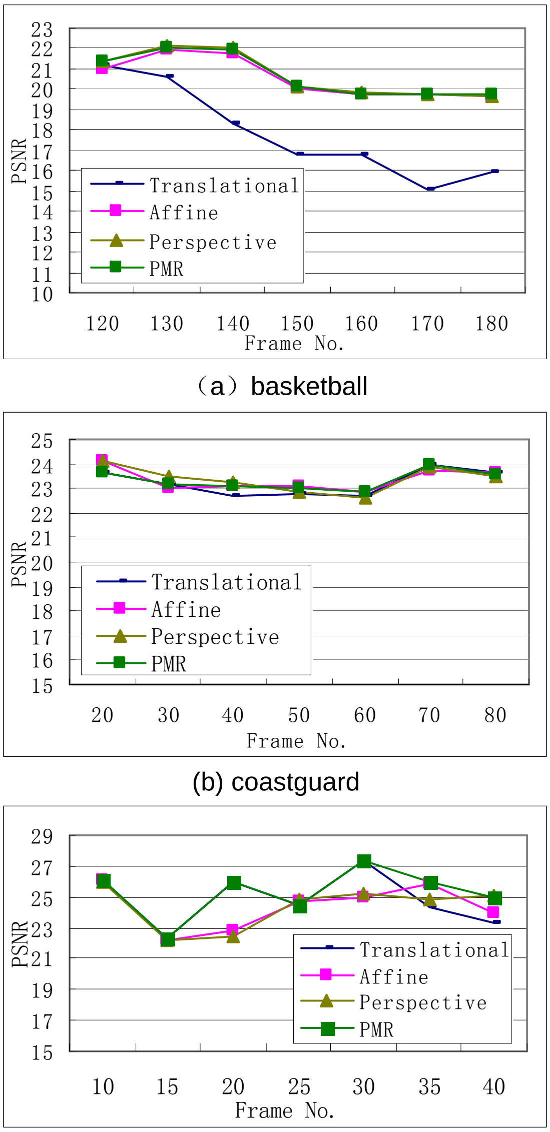
For the MR procedure between the intermediate level and the bottom level, the GMC procedure still takes place at the intermediate level. This is because the intermediate level contains more information than the top level and is much simpler than the bottom level.

4. Experiment Results and Analysis

In order to compare the performance of our PMR algorithm with the traditional single-model based GME method, we built a GME experiment environment. The program is written in C++ and realized in Visual C++. The advantage of this experiment method over transplanting algorithms to VM is that we can get more intermediate results. Experiments were carried out using the standard test sequences with global and object motion: basketball, coastguard, and foreman.

GME methods based on the translational model, affine model and perspective model are realized to compare their performance with our PMR algorithm. Here, we use the peak signal-noise ratio (PSNR) between the current frame and the frame reconstructed using the estimated global motion parameters as an objective measure of the GME precision.

Figure 2 shows the PSNR performance of the four algorithms. The part selected in basketball sequence is of complex global motion and foreground motion. Figure 2 (a) shows that the camera motion is totally beyond the scope of the translational model and our PMR algorithm outperforms affine model in this case.



（a）basketball

(b) coastguard

(c) foreman

Figure 2 PSNR performance comparison

The camera motion in the first 80 frames of coastguard is mainly composed of camera tracking. In this case, translational model based GME is precisely enough for representing the global motion. Figure 2 (b) shows that the performance of our PMR algorithm is comparable with the other three methods.

Figure 2 (c) shows the performance of the four algorithms on foreman sequence. The difference between foreman and the first two sequences is that the size of the foreground is quite large with respect to that of the background. As a result, the FPs selected are composed of large number of foreground points, which complicates and may bias the parameter calculation procedure. As a consequence, the affine model and the perspective model based GME algorithms may not come into convergence before the 32-times iteration limit reaches. This is why the translational model based GME algorithm and our PMR algorithm outperform affine model and perspective model based algorithms at frame 20 and 30. This experiment also shows that our PMR algorithm is less dependent on the FPs as compared with single-model based GME algorithm.

In our PMR algorithm, the motion model used on the top level is always the translational model. As shown in figure 3, the number of iteration on the top level (level 2) of PMR is always the same as that of the translational model. When it is determined to use the affine model in the intermediate model, the x- and y- offsets estimated on the top level are projected to the intermediate level and used as the initial value of the affine model. This kind of parameter projection between models helps to improve the convergence performance of the affine model and the same situation happens between the affine model and the perspective model. So there is a great reduction of the number of iteration, as shown in figure 3. The corresponding PSNR performance is shown in figure 2 (a). It should be pointed out that the numbers of FPs used in the four algorithms are almost the same. So the experiments show that our PMR algorithm can achieve high GME precision with low computation complexity.

5. Comment

In this paper, we proposed a progressive model refinement algorithm. This algorithm works on the three-level GME structure used in MPEG-4 VM. The model refinement procedure takes place when the GME procedure goes from one level to another level. This progressive refinement method improves the convergence performance of L-M algorithm and makes our algorithm much more robust and faster than traditional single-model based algorithms. Experiments show that this model selection method can efficiently select motion models adaptively according to the complexity of the camera motion.

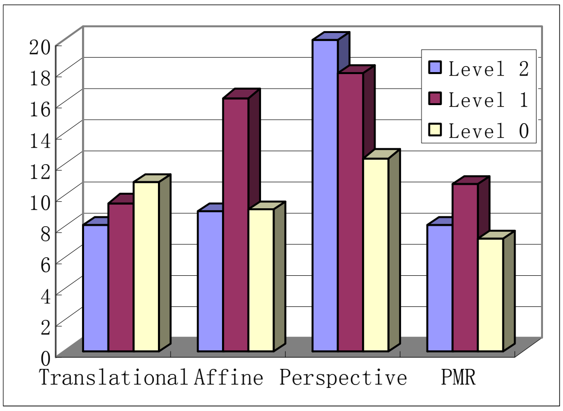


Figure 3 Iteration times comparison (basketball)

6. References

[1] Ye.Ping. Su, Ming-Ting. Sun, and Vincent. Hsu, “Global Motion Estimation From Coarsely Sampled Motion Vector Field and the Applications”, IEEE Transactions on Circuits and Systems for Video Technology, Feb. 2005, pp.232-242.

[2] Bin. Qi, and Aishy. Amer, “Robust and Fast Global Motion Estimation Oriented to Video Object Segmentation”, IEEE International Conference on Image Processing, 2005, Sept. 2005, pp.153-156.

[3] Y.W. He, W. Qi, S.Q. Yang, and Y.Z. Zhong, “Feature-based Fast and Robust Global Motion Estimation Technique for Sprite Coding”, ISO/IEC JTC1/SC29/WG11, MPEG00/M6226, Beijing, July 2000.

[4] MPEG, “MPEG-4 Video Verification Model version 18.0”, ISO/IEC JTC1/SC29/WG11/N3908, Pisa, January 2001.

[5] Wing.Cheong. Chan, Oscar C. Au, and Ming Fai Fu, “Improved Global Motion Estimation Using Prediction and Early Terminaiton”, IEEE International Conference on Image Processing, 2002, pp. 285-288.

[6] Jen-Chi Huang, and Wen-Shyong Hsieh, “Automatic Feature-based Global Motion Estimation in Video Sequences”, IEEE Transactions on Consumer Electronics, Aug. 2004, pp.911 – 915.

[7] MPEG, “Text of 14496-7 PDTR”, ISO/IEC JTC1/SC29/WG11/N4344, Sydney, July 2001.

[8] Y. Keller, and A. Averbuch, “Fast Global Motion Estimation For MPEG-4 Video Compression” International Conference on Image Processing (ICIP) 2002, Rochester, USA, September 2002.