

快速鲁棒的全局运动估计算法*

贺玉文, 赵黎, 钟玉琢, 杨士强

(清华大学 计算机科学与技术系, 北京 100084)

E-mail: heyw@media.cs.tsinghua.edu.cn

http://www.tsinghua.edu.cn

摘要: 对全局运动估计算法进行研究, 提出一种新的快速且鲁棒性较好的全局运动估计算法. 全局运动估计是 MPEG-4 中的 sprite 编码中的关键技术, 其性能将直接影响编码效率和速度. 所提出的算法比传统的 Levenberg-Marquadet 方法要快, 而且参数估计比较准确. 新算法是基于非线性密度进行估计的, 采用了六参数的仿射模型. 为了提高计算速度, 采用了 3 层金字塔进行多分辨率计算, 而且在每层迭代计算中都抽取重要的特征点进行计算, 同时结合了 Gauss-Newton 优化计算方法. 为了保证计算的准确性, 采用基于直方图和基于块两种方法以有效地去除噪声点, 同时, 在迭代计算时引进鲁棒统计, 减少噪声点的影响. 通过全局运动估计实验比较可以看到新算法在速度上的优越性, 同时, 通过 sprite 图像生成实验可以看出算法参数估计的准确性.

关键词: 参数模型; 全局运动估计; 仿射变换; 金字塔计算

中图法分类号: TP391 **文献标识码:** A

全局运动是指在视频序列帧中占有较大比例的像素运动, 一般主要是由摄像机运动形成的. 在有前景和背景组成的图像中, 背景所占比例较大. 最基本的运动估计是指摄像机固定、求解运动目标相对背景的运动. 如果在拍摄过程中摄像机是运动的, 同时前景物体也有其自身的运动, 那么在视频序列中所表现出来的就是背景、前景物体都有各自的运动: 背景的运动是由摄像机运动造成的, 称为全局运动; 而前景物体表现出来的运动是前景物体相对摄像机的运动. 全局运动估计的目的就是从视频序列中找出背景的运动规律.

全局运动估计可以应用于基于运动的图像分割^[1]、全景图的生成^[2]和 sprite 编码^[3~5]等领域. 在基于运动的图像分割中, 首先要进行全局运动估计, 得到背景的运动, 然后根据背景运动进行运动补偿, 这样就得到前景物体的运动, 根据运动的不同就可以将前景和背景物体分开, 从而完成比较理想的图像分割. 在生成全景图时需要知道相邻帧的运动参数, 进行全局运动估计就可以得到这些运动参数, 根据运动参数可以将相邻帧连接上, 得到全景图. MPEG-4 中 sprite 编码^[3~5]就是利用全景图进行预测和补偿, 从而极大地提高了压缩效率.

Konrad^[7]提出的全局运动估计算法采用 Levenberg-Marquadet (LM) 方法, 用残差直方图方法去除噪声, 算法计算量大, 而且计算结果对噪声敏感, 很难做到实时. Meyer^[8]在运动跟踪中采用了多分辨率仿射模型参数估计, 提高了计算速度, 但算法对噪声敏感, 不能保证计算结果的准确性. 本文针对这两方面对全局运动估计进行研究, 提出新的迭代算法以减少计算量. 另外, 在计算中为了

* 收稿日期: 2000-05-16; 修改日期: 2001-05-25

基金项目: 国家重点基础研究发展规划 973 资助项目 (G1999032704)

作者简介: 贺玉文 (1974-), 男, 江苏丹阳人, 博士生, 主要研究领域为低码率视频编码, 可扩展编码; 赵黎 (1975-), 男, 湖北沙市人, 博士生, 主要研究领域为视频分析与传输, 多媒体数据编码算法, 视频信息基于内容的研究; 钟玉琢 (1938-), 男, 辽宁沈阳人, 教授, 博士生导师, 主要研究领域为多媒体计算机技术; 杨士强 (1952-), 男, 黑龙江哈尔滨人, 教授, 博士生导师, 主要研究领域为多媒体数据压缩技术, 信息广播系统.

减少噪声影响,引进了两种去除噪声的方法.一种是基于残差直方图方法^[7],另一种是基于残差块的方法,其中基于残差块的方法是本文所提出来的.这两种方法的结合使用可以较好地去除前景运动带来的噪声.

本文第1节讲述全局运动估计原理和方法,同时详细描述全局运动估计新算法.第2节讲述该算法和LM算法的实验比较.最后是讨论以及对今后研究工作的展望.

1 全局运动估计原理和方法

1.1 全局运动估计原理

首先建立数学模型,用 $I(\mathbf{r}, t)$ 表示 t 时刻的图像, \mathbf{r} 表示位置矢量.采用六参数模型来表示两个时刻图像之间像素的对应关系:

$$\begin{cases} x' = ax + by + c \\ y' = dx + ey + f \end{cases} \quad (1)$$

在该模型下定义参数矢量 $\mathbf{P} = (a, b, c, d, e, f)^T$,其中分量 c, f 与平移运动有关,分量 a, b, c, d 与放缩、旋转运动有关.这样,在六参数模型下就可以由参数矢量 \mathbf{P} 用 $I(\mathbf{r}, t)$ 预测 $I^*(\mathbf{r}, t + \Delta t, \mathbf{P})$,其中 $I(\mathbf{r}, t)$ 和 $I^*(\mathbf{r}, t + \Delta t, \mathbf{P})$ 中的位置对应关系满足六参数模型,即

$$\begin{cases} x(t) = ax(t + \Delta t) + by(t + \Delta t) + c \\ y(t) = dx(t + \Delta t) + ey(t + \Delta t) + f \end{cases} \quad (2)$$

$$I^*(\mathbf{r}(t + \Delta t), t + \Delta t, \mathbf{P}) = I(\mathbf{r}(t), t). \quad (3)$$

其中 $\mathbf{r}(t + \Delta t) = (x(t + \Delta t), y(t + \Delta t))^T$, $\mathbf{r}(t) = (x(t), y(t))^T$.全局运动估计问题就成为求解最优的运动模型参数 \mathbf{P} 的问题,使得模型参数在 \mathbf{P} 时预测图像 $I^*(\mathbf{r}, t + \Delta t, \mathbf{P})$ 和待预测图像 $I(\mathbf{r}, t + \Delta t)$ 在最小二乘法准则下最相似.

1.2 改进的迭代算法

这是本文提出的全局运动估计的一种新的非线性迭代方法.一般地,在处理优化问题时通常使用 Gauss-Newton(GN)方法或者 LM 方法^[9],这些方法虽然收敛性比较好,但计算量比较大,针对计算量大这个问题,本文对 Gauss-Newton 方法进行了研究,提出的新迭代方法可以有效地减少计算量.用 $[X(t), Y(t)]^T$ 表示某像素点在 t 时刻图像中的位置,用 $[X(t+1), Y(t+1)]^T$ 表示该像素点在 $(t+1)$ 时刻图像中的对应位置.它们的对应关系用六参数模型表示:

$$\begin{cases} X(t) = aX(t+1) + bY(t+1) + c \\ Y(t) = dX(t+1) + eY(t+1) + f \end{cases} \quad (4)$$

同时,得到对偶表示:

$$\begin{bmatrix} X(t+1) \\ Y(t+1) \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix}^{-1} \left(\begin{bmatrix} X(t) \\ Y(t) \end{bmatrix} - \begin{bmatrix} c \\ f \end{bmatrix} \right) = \begin{bmatrix} a' & b' \\ d' & e' \end{bmatrix} \begin{bmatrix} X(t) \\ Y(t) \end{bmatrix} + \begin{bmatrix} c' \\ f' \end{bmatrix}. \quad (5)$$

其中

$$\begin{bmatrix} a' & b' \\ d' & e' \end{bmatrix} = \begin{bmatrix} a & b \\ d & e \end{bmatrix}^{-1}, \quad (5-1)$$

$$\begin{bmatrix} c' \\ f' \end{bmatrix} = - \begin{bmatrix} a & b \\ d & e \end{bmatrix}^{-1} \begin{bmatrix} c \\ f \end{bmatrix}. \quad (5-2)$$

假设光照条件不变,由于亮度相同,所以得到这样的一系列的方程(设图像中像素总个数为 N):

$$I(X_i(t), Y_i(t), t) = I(X_i(t+1), Y_i(t+1), t+1), \quad i=0, 1, \dots, N-1.$$

其中 i 表示第 i 个点, 这是一个超约束方程组, 未知数的个数为参数总个数 6, 而方程总个数为 N (例如 176×144), N 一般远大于 6, 所以在一般情况下该方程组没有精确解, 用最小二乘法方法可以求解最优解. 为了简单表示, 可以用矩阵来表示图像, I^* 表示 $(t+1)$ 时刻的图像矩阵, $I(P)$ 表示参数为 P 时的预测图像矩阵, 矩阵表示为

$$I^* = [I(X_0(t+1), Y_0(t+1), t+1), I(X_1(t+1), Y_1(t+1), t+1), \dots, I(X_{N-1}(t+1), Y_{N-1}(t+1), t+1)]^T,$$

$$I(P) = [I(X_0(t), Y_0(t), t), I(X_1(t), Y_1(t), t), \dots, I(X_{N-1}(t), Y_{N-1}(t), t)]^T.$$

将 t 时刻的预测值 $I(X(t), Y(t), t)$ 在 $(t+1)$ 时刻的观测值 $I(X(t+1), Y(t+1), t+1)$ 处展开:

$$(X(t), Y(t), t) = I(X(t+1), Y(t+1), t+1) + \frac{\partial I}{\partial X(t+1)}(X'(t+1) - X(t+1)) - \frac{\partial I}{\partial Y(t+1)}(Y'(t+1) - Y(t+1)). \quad (6)$$

其中 $(X'(t+1), Y'(t+1))$ 是 t 时刻图像 $(X(t), Y(t))$ 点在 $t+1$ 时刻图像中对应的位置. 根据式 (5), 将式 (6) 中的 $(X'(t+1), Y'(t+1))$ 用 $(X(t+1), Y(t+1))$ 表示, 得到

$$I_x X(t+1)(a' - 1) + I_x Y(t+1)b' + I_x c' + I_y X(t+1)d' + I_y Y(t+1)(e' - 1) + I_y f' = \Delta I. \quad (7)$$

其中

$$I_x = \frac{\partial I}{\partial X(t+1)}, \quad I_y = \frac{\partial I}{\partial Y(t+1)}, \quad \Delta I = I(X(t+1), Y(t+1), t+1) - I(X(t), Y(t), t).$$

由于每个像素都有一个方程式 (7) 作约束, 这样得到的约束方程组也是超约束方程, 无法直接解得待定参数, 用矩阵来表示该方程, 令

$$P^* = [a' - 1, b', c', d', e' - 1, f']^T = P' - [1, 0, 0, 0, 1, 0]^T, \quad (7-1)$$

$$\Delta I = I^* - I(P),$$

$$F = \begin{bmatrix} I_{X_1} X_1 & I_{Y_1} Y_1 & I_{X_1} & I_{Y_1} X_1 & I_{Y_1} Y_1 & I_{Y_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_{X_N} X_N & I_{Y_N} Y_N & I_{X_N} & I_{Y_N} X_N & I_{Y_N} Y_N & I_{Y_N} \end{bmatrix}.$$

由式 (7) 得到

$$FP^* = \Delta I;$$

定义能量函数:

$$E; E = (\Delta I - FP^*)^T W (\Delta I - FP^*),$$

其中 W 为权值矩阵. W 为对角矩阵, 其对角元素 $W_{i,i}$ 表示第 i 个像素的加权值.

由 $\frac{\partial E}{\partial P^*} = 0$ 解得

$$P^* = (F^T W F)^{-1} (F^T W (\Delta I)). \quad (8)$$

根据式 (7-1) 得到

$$P' = P^* + [1, 0, 0, 0, 1, 0]^T;$$

由式 (5-1) 和式 (5-2) 可以解得 P_1 :

$$\begin{bmatrix} a_1 & b_1 \\ d_1 & e_1 \end{bmatrix} = \begin{bmatrix} a' & b' \\ d' & e' \end{bmatrix}^{-1}; \quad \begin{bmatrix} c_1 \\ f_1 \end{bmatrix} = - \begin{bmatrix} a' & b' \\ d' & e' \end{bmatrix}^{-1} \begin{bmatrix} c' \\ f' \end{bmatrix}.$$

由 P_1 和 P 可以解得第 1 次迭代得到的 $t+1$ 时刻和 t 时刻图像间的仿射模型参数 P^1 :

$$\begin{aligned} \begin{bmatrix} X(t) \\ Y(t) \end{bmatrix} &= \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ d_1 & e_1 \end{bmatrix} \begin{bmatrix} X(t+1) \\ Y(t+1) \end{bmatrix} + \begin{bmatrix} c_1 \\ f_1 \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix} \\ &= \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} a_1 & b_1 \\ d_1 & e_1 \end{bmatrix} \begin{bmatrix} X(t+1) \\ Y(t+1) \end{bmatrix} + \begin{bmatrix} a & b \\ d & e \end{bmatrix} \begin{bmatrix} c_1 \\ f_1 \end{bmatrix} + \begin{bmatrix} c \\ f \end{bmatrix}. \end{aligned}$$

由上式可以得到 P^1 的各个分量:

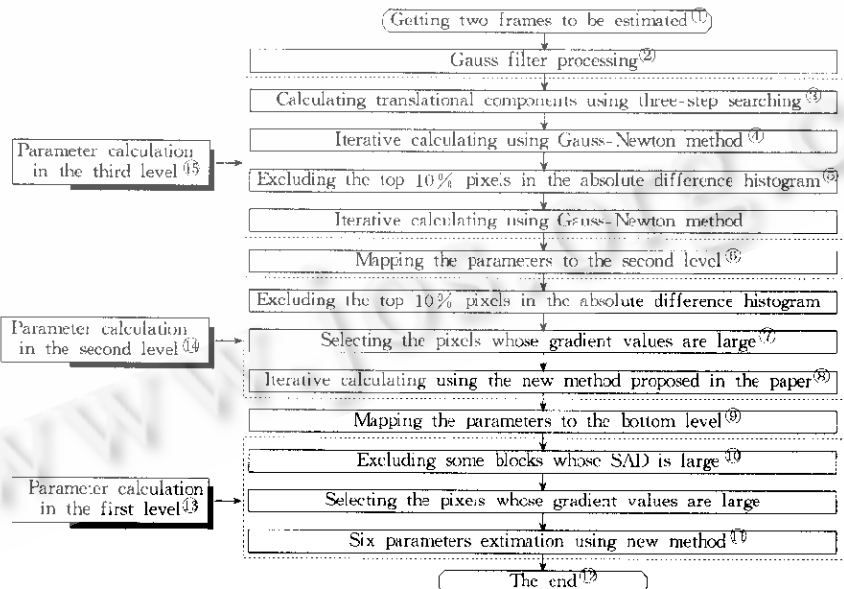
$$a' = aa_1 + bd_1; b' = ab_1 + be_1; c' = ac_1 + bf_1 + c; d' = da_1 + ed_1; e' = db_1 + ee_1; f' = dc_1 + ef_1 + f.$$

利用式(8)可以反复迭代,直到满足结束条件为止,这样就得到 t 时刻和 $t+1$ 时刻的仿射模型参数 P .

本方法与 GN 方法、LM 方法相比,在相同计算条件下,在一次迭代过程中的计算量最少,这是因为 GN 和 LM 方法在每次迭代过程中都要计算式(8)中的 F 矩阵,而本方法中只要计算一次就可以了.

1.3 全局运动估计改进算法

算法流程图如图 1 所示.因为一幅图像的像素点很多,并且包含前景点,这些前景点并不满足背景点的运动参数模型,因此对于全局运动估计来说,这些前景点就是噪声点,在参数估计中,这些点被称为外点.由于以上原因,全局运动估计具有计算量大和外点多两个特点,在实时应用中就要减少其计算量.另一方面,为了准确估计,就要采取有效措施去除其中的外点,本文中所要介绍的算法就是针对这两方面,并采取以下有效措施,使其不但计算准确,而且能够提高计算速度.



①开始读入待估计的两帧图像,②Gauss滤波去除噪声,③三步法计算第3层金字塔的平移参数,④用 Gauss-Newton 算法进行六参数估计,⑤用直方图统计去除 10% 的外点,⑥将第 3 层金字塔的参数映射到第 2 层金字塔,⑦抽取梯度值大的特征点,⑧用新迭代算法进行六参数估计,⑨将第 2 层金字塔的参数映射到第 1 层金字塔,⑩去除部分残差值较大的块,⑪用基于新迭代算法进行六参数估计,⑫结束,⑬第 1 层金字塔的参数计算,⑭第 2 层金字塔的参数计算,⑮第 3 层金字塔的参数计算.

Fig. 1 Flow char: of GME algorithm

图 1 算法流程图

① 由于有噪声存在,而差分运算容易受噪声的影响,所以最初要对图像进行低通滤波去除

噪声.

② 在第 3 层金字塔首先用三步搜索^[7]的方法计算平移参数. 这主要是针对有大平移运动的情况, 防止该层的 GN 迭代陷入局部极小点, 导致参数估计失败. 当有大平移运动时, 初始参数值和所求的参数值相差较大, 非线性迭代方法很容易提前进入某一个局部极小点, 从而使迭代提前终止.

③ 在第 3 层金字塔当三步搜索后采用 GN 法作初始估计. 因为在三步搜索中所比较的是整幅图像的残差的绝对值之和(SAD), 当前景的灰度值较大并且纹理较细, 而背景灰度值较小或者灰度值较大但纹理很粗时, 整幅图像的残差的绝对值之和就由前景决定, 这时, 三步搜索显然无法正确估计背景的平移参数. 在这种情况下, GN 法虽然不能准确地估计运动参数, 但却能粗略地估计出背景的运动参数趋势.

④ 为了使参数估计准确, 所以在每层金字塔迭代计算之前去掉外点, 在改进算法中采用基于直方图和基于块的两种去除外点的方法. 在金字塔的最上面两层用基于直方图的方法去除外点, 这是因为在这两层的参数还不够准确, 此时残差大的点比较分散, 并不都是前景点; 在金字塔的最下层由于已经有前两层的计算, 所以残差大的点基本上都是前景点, 前景区域的块残差都很大, 这很适合采用基于块的方法去除外点.

⑤ 为了节省计算时间, 需要减少每次迭代计算的点数. 在改进算法中, 在背景区域内抽取部分特征像素点进行计算. 根据迭代计算公式可知, 对计算结果影响较大的是那些梯度大的点. 本文用阈值法抽取那些梯度较大的点, 这样就大大减少了计算量, 而且由于抽取特征点是在去除外点的基础上, 所以这样做既能减少计算量, 又能保证计算的准确性.

⑥ 在金字塔下两层的迭代计算中, 采用了本文提出的新的迭代方法, 这主要是为了减少计算时间.

上述基于残差直方图去除外点是指根据当前运动参数进行补偿后得到残差图像, 然后根据残差图像直方图中 90% 处得到阈值, 最后就可以去除残差大于阈值的点. 直方图的特点是统计信息, 它丢失了空间信息, 所以, 去除的点在空间是分散的. 基于残差块去除外点是指将补偿后得到的残差图像分成一定大小块(本算法中为 16×16), 然后计算每块的残差绝对值之和, 根据预先设定好的阈值选出比阈值大的块, 如果该块周围块中(4 邻域)也有比阈值大的块, 那么该块就是要删除的块, 否则保留该块. 这样做的前提是假设前景是一个比较集中的区域, 这样就可以很好地去除前景区域.

2 实验比较和算法评价

我们用本文提出的改进算法对一些图像序列做了测试, 并与传统的 LM 方法进行了比较. 实验① Foreman 是针对旋转运动的, 实验② Coastguard 是针对大偏移运动的, Foreman 序列如图 2 所示.



(a) Frame 15
(a) 第 15 帧



(b) Frame 17
(b) 第 17 帧

Fig. 2
图 2

(1) 对于 Foreman 序列中的第 15 帧和第 17 帧进行测试,计算结果见表 1.

Table 1 GME results of Foreman sequence

表 1 Foreman 序列计算结果

Calculating method ^①	Value of six parameters (c, f are translational components along X and Y axis) ^②						Time consuming ratio ^③	PSNR ^④
	a	b	c	d	e	f		
LM algorithm ^⑤	1.001 8	-0.000 8	0.100 1	-0.000 1	0.998 9	0.733 3	1	25.78
Proposed algorithm ^⑥	1.000 4	0.000 2	0.204 2	-0.000 5	1.000 0	0.675 5	0.16	25.80

①计算方法,②估计的六参数值(c, f 分别为 X 和 Y 轴平移分量),
③计算时间比值,④峰值信噪比,⑤传统 LM 法,⑥本文的算法.

从上表可以看出,本文的算法所用时间只有传统 LM 算法的 16%,而计算结果并不差.下面是根据运动参数进行运动补偿后,用相同阈值分割得到如图 3 所示的二值图.



(a) Binary image of absolute difference without motion compensation

(b) Binary image of absolute difference with motion compensation of LM algorithm

(c) Binary image of absolute difference with motion compensation of proposed algorithm

(a) 没有运动补偿的差值图

(b) LM 算法运动补偿后的差值图

(c) 本文的算法运动补偿后的差值图

Fig. 3
图 3

从上面的运动补偿前的差值图可以看到,前景和背景都有,而进行全局运动估计后,根据估计参数进行运动补偿,差值图中主要都是前景,而背景都被去除了,这说明全局运动估计是有效的.

图 4 和图 5 是本文的算法在进行前两层金字塔计算后进行基于块的去除外点和选取计算点的图示.



Fig. 4 Picture of pixel selection in GME calculation

图 4 选取计算点图



Fig. 5 Coarse segmentation picture

图 5 去除前景图

在图 4 中,灰色表示将要去除的外点区域(前景),白色表示选取的计算点(梯度值大的点),黑色表示剩余点.从图 4 和原图的比较可以看到,去除的外点大部分都是前景点,而选取计算点主要是背景中的边缘点,这说明本文的去除外点和选取计算点的方法是很有效的.

图 5 是全局运动估计后根据运动参数进行基于块的分割图,白色表示前景区域,黑色表示背景区域.从该图中可以得到前景区域和背景区域,从背景区域中就比较得到前景的轮廓.

(2) 对于 Coastguard 序列中的第 195 帧和第 205 帧进行测试(大平移运动),如图 6 所示.

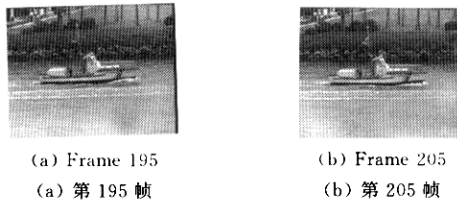


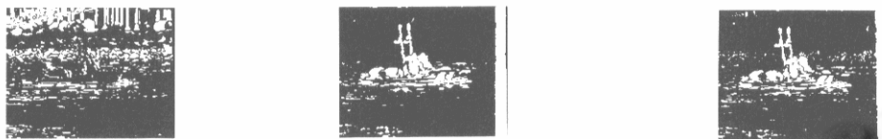
Fig. 6
图 6

Table 2 GME results of Coastguard sequence
表 2 Coastguard 序列计算结果

Calculating method ^①	Value of six parameters (c, f are translational components along X and Y axis) ^②						Time consuming ratio ^③	PSNR ^④
	a	b	c	d	e	f		
LM algorithm ^⑤	0.996 0	-0.026 7	10.359 4	-0.002 1	0.998 0	0.025 8	1	19.87
Proposed algorithm ^⑥	1.000 6	-0.004 0	9.848 9	-0.000 6	0.998 4	-0.032 8	0.28	18.82

①计算方法, ②估计的六参数值(c, f 分别为 X 和 Y 轴平移分量),
③计算时间比值, ④峰值信噪比, ⑤传统LM法, ⑥本文的算法.

从表 2 可以看出, 本文算法所用时间只有传统 LM 算法的 28%, 而从如图 7 所示的运动补偿的差值图可以看出, 本文的算法的计算结果比传统 LM 算法要准确. 下面是根据运动参数进行运动补偿后, 用相同阈值分割得到的二值图.



(a) Binary image of absolute difference without motion compensation
(a) 没有运动补偿的差值图
(b) Binary image of absolute difference with motion compensation of LM algorithm
(b) LM 算法运动补偿后的差值图
(c) Binary image of absolute difference with motion compensation of proposed algorithm
(c) 本文的算法运动补偿后的差值图

Fig. 7
图 7

在上面两幅运动补偿的差值图中, 右边有一条白带, 这是因为原图中右边的黑边造成的, 这与算法无关. 两种算法的差值图相比较, LM 算法的差值图中还有许多背景点没有去除, 这说明全局运动估计的参数值不够准确.

图 8 和图 9 是本文的算法在进行前两层金字塔计算后进行基于块的去前景和选取计算点的图示.

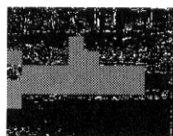


Fig. 8 Picture of pixel selection in GME calculation
图 8 选取计算点图

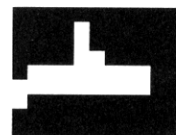


Fig. 9 Coarse segmentation picture
图 9 去除前景图

图 8 中去除的外点包含了绝大多数前景点, 这就能保证全局运动估计的准确性. 图 9 中基于块分割所得到的前景区域是有效的. 这说明本文所提出的算法对于大平移运动情形, 不仅比传统 LM

算法更快,而且更为准确.

(3) 利用本文的算法对 MPEG-4 标准测试序列 stefan 序列(352×240)进行全局运动估计后,根据仿射参数生成全景图^[3],由此可以看出算法所估计参数的准确性,这个序列共 240 帧.图 10 是全景图的生成流程,首先对当前帧进行全局运动估计,得到该帧的仿射变换参数;然后根据此参数对当前帧进行仿射变换,得到它的变形图;最后将变形图和帧存中的全景图进行合成,就可以得到新的全景图,重复此过程就可以生成该视频序列的整个全景图,如图 11 所示.

从生成的全景图来看,本文所提出的全局运动算法所估计的仿射模型参数是比较准确的.

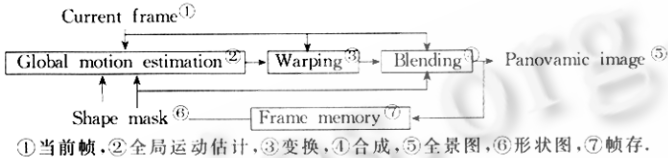


Fig. 10 Panovamic image generation algorithm
图10 全景图的生成流程

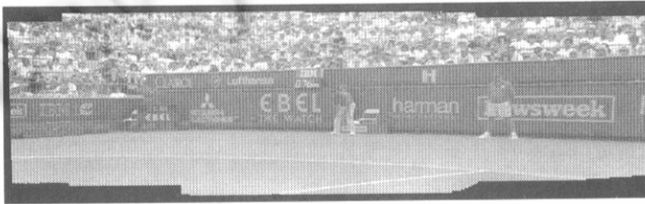


Fig. 11 Panovamic image of stefan sequence
图 11 由 stefan 序列生成的全景图

3 结论和展望

本文提出一种求解模型参数的新方法和快速的全局运动估计算法,并将这种全局运动估计算法和已有的 Levenberg-Marquadet 算法进行了实验比较.从实验结果来看,本文所提出的算法在大的平移运动和一般噪声时都有效,基本满足实时应用要求,而且速度比 Levenberg-Marquadet 算法快 5 倍以上.迭代算法成功的关键是减少噪声的影响,外点对于非线性迭代算法影响很大.所以,只有在准确去除外点之后才能想办法提高速度,否则虽然速度快,但是很难保证计算的准确性.经过全局运动估计后能得到前景区域,对该区域再进行运动估计和补偿后就得到前景轮廓.由于提高了速度,所以该算法可以用在实时 sprite 软件编码中.另一方面,对于有多个物体运动的视频,可以首先利用全局运动估计找到运动物体区域,此时可以实现运动分割或者对多个运动物体进行跟踪^[8].

References:

- [1] Smolic, A., Sikora, T., Ohm, J. R. Long-Term global motion estimation and its application for sprite coding, content description, and segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1999,9(8):1227~1242.
- [2] Giaccone, P. R., Greenhill, D., Jones, G. A. Creating virtual scenes using active contours and global motion estimation. *Pattern Recognition*, 1998,2(Proceedings of the 14th International Conference):1505~1507.
- [3] ISO/IEC WG11 MPEG Video Group. MPEG-4 video verification model version 12.1, ISO/IEC/JTC1/SC29/WG11, No. MPEG98/N2552, Roma, 1998.
- [4] ISO/IEC WG11 MPEG Video Group. MPEG-4 video verification model version 8.0. ISO/IEC/JTC1/SC29/WG11, MPEG-97/N1796, San Jose, 1998.

- [5] Kim, Eung Tae, Kim, Hyung-Myung. Fast and robust parameter estimation method for global motion compensation in the video coder. *IEEE Transactions on Consumer Electronics*, 1999,45(1):76~83.
- [6] Wu, Feng, Gao, Wen, Xiang, Yang-Zhao, *et al.* On-Line sprite encoding with large global motion estimation. In: *Proceedings of the Data Compression Conference'98*. 1998. 546.
- [7] Konrad, Janusz, Dufaux, Frederic. Digital equipment corporation, improved global motion estimation for N3. Proposal of ISO/IEC/SC29/WG11, No. MPEG97/M3096, San Jose, 1998.
- [8] Meyer, F. G., Bouthemy, P. Region-Based tracking using affine motion models in long image sequences. *CVGIP: Image Understanding*, 1994,60(2):119~140.
- [9] Yuan, Ya-xiang. *Numeric Method of Nonlinear Programming*. Shanghai: Shanghai Science and Technology Press, 1993 (in Chinese).

附中文参考文献:

- [9] 袁亚湘. 非线性规划数值方法. 上海:上海科学技术出版社,1993.

A Fast and Robust Algorithm of Global Motion Estimation*

HE Yu-wen, ZHAO Li, ZHONG Yu-zhuo, YANG Shi-qiang

(*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*)

E-mail: heyw@media.cs.tsinghua.edu.cn

<http://www.tsinghua.edu.cn>

Abstract: In this paper, the algorithm of global motion estimation is investigated, and a new algorithm for fast global motion estimation is introduced. Global motion estimation is the key problem in sprite coding, which is adopted in MPEG-4 compression standard. The proposed new algorithm is much faster than the traditional Levenberg-Marquadet algorithm, but the estimated result of the new algorithm is more accurate. The new algorithm is based on nonlinear dense estimation and applies the affine model. In order to accelerate the algorithm, three-level pyramid is utilized in the calculation. It uses those important feature pixels in the calculation of every pyramid level, and combines Gauss-Newton algorithm. In order to guarantee the accuracy of the result, noise is effectively eliminated by histogram-based and block-based methods. Robust statistics is used in the calculation to reduce the influence of noise. The experimental results show that the new algorithm is superior to Levenberg-Marquadet algorithm.

Key words: parameter model; global motion estimation; affine transform; pyramid calculation

* Received May 16, 2000; accepted May 25, 2001

Supported by the National Grand Fundamental Research 973 Program of China under Grant No. G1999032704