

一种快速全局运动估计算法

郭丽 龚声蓉 崔志明

(苏州大学计算机科学与技术学院, 苏州 215006)

摘要 为了对全局运动进行准确快速的估计,在对全局运动估计算法进行研究的基础上,提出了一种快速全局运动估计算法。该算法基于非线性密度估计,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,将基于块的外点去除算法与特征点提取算法相结合,这样既加快了算法的速度,又提高了计算结果的准确性。全局运动估计的实验结果表明,该算法在速度和准确性上具有一定的优势。

关键词 全局运动估计 参数模型 仿射变换 外点去除 特征点选取

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2007)12-2119-08

A Fast Global Motion Estimation Algorithm

GUO Li, GONG Sheng-rong, CUI Zhi-ming

(Computer Science and Technology School, Soochow University, Suzhou 215006)

Abstract In this paper, the algorithm of global motion estimation is investigated, and a fast global motion estimation algorithm is introduced. The new algorithm is based on nonlinear dense estimation and applies the affine model. In order to reduce the time cost during calculation, three-level pyramid is utilized in the algorithm. It combines outlier detection algorithm based on blocks and feature pixels selection algorithm in the calculation of each pyramid level. The calculation time of global motion estimation is reduced and the accuracy is improved. The experimental results show that the new algorithm is superior to original algorithms in velocity and accuracy.

Keywords global motion estimation, parameter model, affine transform, outlier detection, feature pixels selection

1 引言

像素运动是由场景中实体运动或摄像机运动产生的,全局运动是指在视频序列帧中占有较大比例的像素运动,其主要是由摄像机运动形成的。全局运动估计(global motion estimation, GME)已广泛应用于基于对象的图像分割^[1]、全景图像生成^[2]、视频检索^[3]、视频编码^[1,4]等领域,因此研究全局运动估计是很有意义的。全局运动估计的基本原理是利用密度估计的方法得到最佳的模型参数,一般采用迭代优化的方法来估计,由于图像背景像素数量很

多,而且有前景物体的局部运动干扰,因此在进行全局运动估计时有两个难点:首先是计算量大,其次是由于噪声点多,使估计的准确性难以得到保证。这就限制了全局运动估计方法的广泛应用。一直以来,国内外的众多学者都在致力于研究如何改进全局运动估计中模型参数的求解算法,Yosi Keller等人提出了多种特征点选取的方法^[5,6],以减少计算量;Konrad等人提出了一些去除噪声的方法^[7];贺玉文等人从两方面相结合提出了一种快速鲁棒的全局运动估计算法^[8],但是计算量依然比较大。本文对全局运动估计进行了深入的研究,提出了一种新的快速全局运动估计算法。

基金项目:国家自然科学基金项目(60673092);江苏省自然科学基金项目(BK2003029);铁道部“铁路信息科学与工程”开放实验室基金项目(TDXX0501)

收稿日期:2006-02-17;改回日期:2006-08-22

第一作者简介:郭丽(1980~),女,硕士研究生。主要研究方向为图像处理和视频编码。E-mail: guoli198053@sina.com

2 全局运动估计的原理和方法

2.1 全局运动估计的原理

假设 I_1 和 I_2 是相关联的两幅图像,即图像 I_1 可以经过运动变换到图像 I_2 。视频运动可以用参数模型来表示,假设 (x_i, y_i) 和 (\hat{x}_i, \hat{y}_i) 分别为图像 I_1 和 I_2 中相对应的像素点位置,则这两幅图像之间的运动可以表示为

$$\hat{x}_i = f_1(\mathbf{P}, x_i, y_i) \quad (1)$$

$$\hat{y}_i = f_2(\mathbf{P}, x_i, y_i) \quad (2)$$

其中, \mathbf{P} 是运动模型的参数集, f_1 和 f_2 是定义的函数,即参数模型表达式, i 为对应位置的下标。

基于像素梯度的方法是通过最小化两幅图像的亮度残差来求得参数集 \mathbf{P} ,可用公式表达如下:

$$\mathbf{P} = \arg \min \sum_{(x_i, y_i)} (I_1(x_i, y_i) - I_2(\hat{x}_i, \hat{y}_i))^2 \quad (3)$$

其中, $I_1(x_i, y_i)$ 和 $I_2(\hat{x}_i, \hat{y}_i)$ 分别代表图像 I_1 和 I_2 中的对应像素点 (x_i, y_i) 和 (\hat{x}_i, \hat{y}_i) 的亮度值。

2.2 基于仿射运动模型的梯度运动估计法

假设一个位置在 (x, y) 的像素点经过 Δt 时间运动到了位置 $(x + \Delta x, y + \Delta y)$,则由光流方程的假设条件可以知道,沿着运动轨迹,物体的亮度保持不变,即

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t)$$

$I(x, y, t)$ 表示在点 (x, y) 处物体的亮度值。

将 $I(x + dt, y + dt, t + dt)$ 在 $I(x, y, t)$ 处用泰勒展开式展开,即

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + I_x \times u + I_y \times v + I_t + \text{高次项} \quad (4)$$

其中,

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y} \quad I_t = \frac{\partial I}{\partial t} \quad u = \frac{\partial x}{\partial t} \quad v = \frac{\partial y}{\partial t}$$

(u, v) 代表瞬时的像素运动速度矢量,也被称为光流矢量; I_x, I_y 分别代表 x 方向和 y 方向的梯度, I_t 代表时间偏导。

由于 2 维运动矢量估计问题具有“不适定”的特性,因此要进行运动估计还需要根据 2 维运动场模型结构附加约束条件。2 维运动场模型可根据其对应的 3 维运动场数学模型的不同而不同,一个由 3 维刚性物体运动产生的运动场,在正交投影下生成的 2 维运动场,既可以用仿射运动模型描述,也可以用非线性透视运动模型描述。此外,还有更复杂

的二次曲面映射模型等。在这些模型中,由于正交投影的仿射模型是较简单的线性模型,因此采用仿射模型可以减少图像背景各像素点运动速度场的估计计算量,同时仿射模型可以描述平移、旋转和缩放等运动,最适合本文研究的视频图像中场景的情况,因此本文采用六参数模型公式来进行 2 维运动矢量估计。

六参数仿射模型可以表示成

$$u = a_1 x + a_2 y + b_1 \quad (5)$$

$$v = a_3 x + a_4 y + b_2 \quad (6)$$

对于小的运动,由于高次项可以忽略,因此可以得到

$$I_x \times u + I_y \times v + I_t = 0 \quad (7)$$

将六参数模型(式(5)和式(6))代入式(7),得

$$I_x \times x \times a_1 + I_x \times y \times a_2 + I_x \times b_1 + I_y \times x \times a_3 + I_y \times y \times a_4 + I_y \times b_2 = -I_t \quad (8)$$

由于该方程有 6 个未知数,无法求解,考虑到对于图像中的每一个像素点都满足式(8),因此对每一个像素点列出式(8),假设有 K 个像素点,于是得到

$$\mathbf{F} \times \mathbf{a} = \mathbf{b} \quad (9)$$

其中,

$$\mathbf{F} = \begin{bmatrix} I_{x_1} \times x_1 & I_{x_1} \times y_1 & I_{x_1} & I_{y_1} \times x_1 & I_{y_1} \times y_1 & I_{y_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_{x_k} \times x_k & I_{x_k} \times y_k & I_{x_k} & I_{y_k} \times x_k & I_{y_k} \times y_k & I_{y_k} \end{bmatrix}$$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ a_3 \\ a_4 \\ b_2 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -I_{t,1} \\ \vdots \\ -I_{t,k} \end{bmatrix}$$

2.3 参数求解

由以上的推导可得到如下的表达式:

$$\mathbf{F} \times \mathbf{a} = \mathbf{b}$$

其中, \mathbf{F} 的大小为 $k \times 6$, \mathbf{a} 的大小为 6×1 , \mathbf{b} 的大小为 $k \times 1$ 。

对于一幅图像,由于 k 往往大于 6,即所列出的方程式将远远大于未知数的个数,因此可应用最小二乘原理来解决这个问题,也就是最小化公式

$$\|\mathbf{F} \times \mathbf{a} - \mathbf{b}\|^2$$

进一步化简得

$$(\mathbf{F}^T \mathbf{F}) \times \mathbf{a} = (\mathbf{F}^T \mathbf{b}) \quad (10)$$

$$6 \times 6 \quad 6 \times 1 \quad 6 \times 1$$

可简单表示成

$$A \times a = B \quad (11)$$

其中, $A = F^T F$ $B = F^T b$

2.4 改善精度

在上面的公式推导过程中,对于公式 $I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + I_x \times u + I_y \times v + I_t +$ 高次项,由于假设其对应的运动为小的运动,因此可忽略高次项,但是实际情况是,由于这种假设往往是不精确的,因此无法忽略高次项的存在,这样问题将转化到多项式求根的问题上,其可以用牛顿迭代算法或者用 LM (Levenberg-Marquadet) 算法来求解这个问题。

2.5 全局运动参数模型的迭代求解算法

全局运动估计共分为 3 层金字塔计算,而金字塔则用 $[1/4, 1/2, 1/4]$ 滤波器产生。

MPEG-4 校验模型^[9]采用了 Konrad 提出的全局运动估计算法^[7],算法描述如下:

(1) 对金字塔第 3 层图像进行全局运动估计计算。

先利用三步搜索方法估计平移参数(式(5)、式(6)中的 b_1 和 b_2),这主要是针对大平移运动;然后根据计算结果用残差直方图去除 10% 的残差较大的点,最后再用 LM 方法对剩余像素点进行参数估计计算;

(2) 将计算结果映射到第 2 层金字塔图像(b_1 和 b_2 扩大 1 倍),并根据计算结果用残差直方图去除 10% 残差大的点,然后用剩余像素点对金字塔第 2 层用 LM 方法进行参数估计计算;

(3) 将计算结果映射到第 1 层金字塔图像,再用步骤(2)中相同的方法计算参数,即得到最终的参数值。

3 快速全局运动估计算法

由于目前的主要的全局运动估计算法都是基于密度估计的,而这些方法主要面临以下两个困难:计算代价高和噪声高引起计算结果不准确。如何解决计算量和估计精度两者间的矛盾是当前的研究重点。

3.1 噪声点去除

在全局运动估计中,局部运动(前景物体的运动)点的运动会干扰全局运动的计算,可称这些局部运动点为噪声点(外点)。文献[7]提出了一种直方图去除外点法,由于直方图只考虑了统计特性,因此去除的外点比较分散。局部运动往往是由于前景

物体的运动造成的,因此那些局部运动点具有聚集成块的特性,贺玉文等提出了基于块的去局部运动点法^[8],但其也只是在金字塔图像的最底层使用了该方法,金字塔的上面两层同样使用直方图法。由于前景物体局部运动点的聚集性并不随图像分辨率的降低而改变,因此本文对其进行了进一步修改,即在金字塔的每一层图像上均使用基于块的外点去除法。

对于金字塔的每一层图像,先经过运动补偿形成残差图像,然后将残差图像分成 $M \times N$ 大小的块,最后计算每个块的 SAD(sum of absolute difference) 值。SAD 定义如下:

$$SAD = \sum_{i=0}^M \sum_{j=0}^N |\gamma_{i,j}| \quad (12)$$

其中, $\gamma_{i,j}$ 是点 (i, j) 的残差。

先将所有块的 SAD 值进行排序,并将 SAD 值大的前 30% 的块作为预选去除块,再按照下面的方法确定最后的去除块:

(1) 如图 1 所示,设 B_0 是当前预选去除块,如果它的 8 邻域块中有多于 4 个预选去除块,那么这个预选去除块将被去除,并标志为去除块。

| | | |
|-------|-------|-------|
| B_1 | B_2 | B_3 |
| B_4 | B_0 | B_5 |
| B_6 | B_7 | B_8 |

图 1 基于块的外点去除

Fig. 1 Outlier rejection based on block

(2) 第 1 步完成后,对于剩余预选去除块,如果它的 8 邻域块中有去除块,那么这个预选去除块也将被去除。

所划分块的大小对计算量和去除外点的性能有很大影响,如果划分得过小,虽然能够提高去除前景区域的有效性,但却增加了计算量,一般对于去除的前景区域并不要求很精确;而划分得过大,则去除前景区域就会有扩大,这样会去除许多背景区域。经过实验发现,对于分辨率为 352×288 的图像来说, 16×16 大小的块比较合适。由 Gaussian 图像构造法构造的 3 层图像金字塔,由于每一层的分辨率均为下一层的 1/4,因此所选择的块的大小也为下层块大小的 1/4。

3.2 特征点选取

由于全局运动估计是基于多点密度估计的,在去

除了 30% 的局部运动点后,剩下 70% 的背景点,计算量依然非常大,因此如何在这些背景点中选择可靠的计算点,就成为加速全局运动估计算法的关键。

大家知道,在全局运动估计中都涉及到梯度运算,由于梯度运算会受到噪声点的影响,而且梯度值越小的话,该点越容易受到噪声影响,如果让这些点参与计算,不仅会使计算量增大,而且会使计算结果不准确,因此本文提出如下的选取特征点的公式:

$$C = \sqrt{\nabla_x^2 + \nabla_y^2} \quad (13)$$

其中, ∇_x 和 ∇_y 分别代表该点的亮度值 $I(x, y)$ 在 x 方向和 y 方向的梯度,通过设定相应的阈值,即可将 C 大于阈值的点作为特征点。经过实验,选取的阈值为所有剩余点的 C 的平均值的 2.75 倍。

3.3 快速全局运动估计算法描述

(1) 用 Gaussian 图像构造法构造图像金字塔

(2) 对金字塔顶层图像进行全局运动估计计算

① 用高斯滤波器进行滤波,滤除图像的随机噪声;

② 利用改进的三步搜索法计算整帧图像的平移参数;

③ 根据第②步求得的平移参数对图像进行运动补偿,求得残差图像;

④ 将残差图像分成相应大小的块,根据噪声点去除算法去除局部运动点;

⑤ 根据特征点选取算法选取特征点;

⑥ 用 LM 方法对剩余像素点进行参数估计计算,求得相应的参数集。

(3) 将金字塔顶层求得的参数集映射到金字塔的下一层,并对该层进行全局运动估计

① 用高斯滤波器进行滤波,滤除图像的随机噪声;

② 根据映射的参数集对当前图像进行运动补偿,求得残差图像;

③ 将残差图像分成相应大小的块,再根据噪声点去除算法去除局部运动点;

④ 根据特征点选取算法选取特征点;

⑤ 用 LM 方法对剩余像素点进行参数估计计算,求得相应的参数集。

(4) 将金字塔中间层的参数集映射到金字塔的底层,用第(3)步中的方法求得该层的参数集,即最终求得的参数集。

4 实验结果

为验证本文算法的效果,用本文提出的改进算法(简称 GMC_3 算法)对一些标准测试序列进行了测试实验,同时与 Konrad 提出的 LM 算法^[7](简称 GMC_1 算法)以及贺玉文等提出的算法^[8](简称 GMC_2 算法)进行了比较。

实验数据为标准测试序列,包括:(1) CIF 格式 352×288 大小的 Bus 序列,针对摄像头左右平移运动;(2) QCIF 格式 176×144 大小的 Foreman 序列,针对摄像头旋转运动;(3) QCIF 格式 176×144 大小的 Coastguard 序列,针对摄像头上下平移运动。

(1) Bus 序列的第 0 帧、第 1 帧如图 2 所示,由于公共汽车在运动,因此为了拍摄公共汽车,摄像机要跟着公共汽车向左运动,由于摄像机的运动,致使静止背景中的小汽车、栏杆等出现了全局运动。实验时,用 3 种算法分别对背景的全局运动参数进行计算,计算结果如表 1 所示。

从表 1 可以看出,本文所提出的算法计算时间最短,大约为 GMC_1 算法的 73%, GMC_2 算法的 75%,图 3 是根据全局运动参数进行运动补偿后,用



(a) 第 0 帧



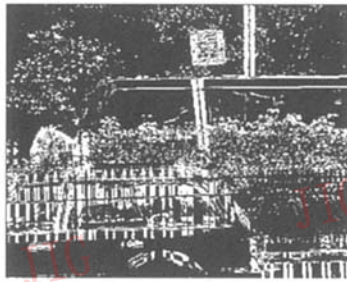
(b) 第 1 帧

图 2 Bus 序列图像

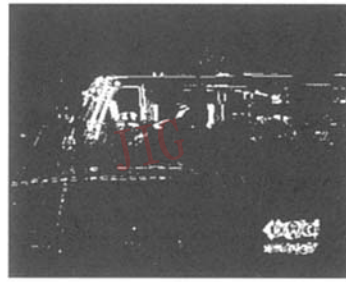
Fig. 2 Images of Bus sequence

表 1 Bus 序列全局运动估计计算结果
Tab.1 GME result of Bus sequence

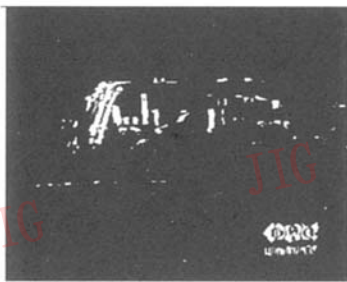
| 计算方法 | 参数 a_1 | 参数 a_2 | 参数 b_1 | 参数 a_3 | 参数 a_4 | 参数 b_2 | 所用时间 (ms) |
|-------|-----------|----------|----------|----------|-----------|----------|-----------|
| GMC_1 | -0.009 97 | 0.000 66 | 8.768 04 | 0.000 18 | -0.009 71 | 1.374 06 | 660 |
| GMC_2 | -0.009 79 | 0.000 74 | 8.633 08 | 0.000 19 | -0.010 11 | 1.360 14 | 641 |
| GMC_3 | -0.010 07 | 0.000 98 | 8.655 36 | 0.000 28 | -0.008 83 | 1.135 13 | 481 |



(a) 没有运动补偿的差值图



(b) GMC_1 算法运动补偿后的差值图



(c) GMC_2 算法运动补偿后的差值图



(d) GMC_3 算法运动补偿后的差值图

图 3 用 3 种不同算法进行运动补偿后的差值图经相同阈值分割得到的二值图

Fig.3 Binary image of absolute difference with motion compensation of three different algorithm

相同阈值分割得到的二值图。从没有运动补偿的差值图(图 3(a))可以看到,前景和背景都有,而进行全局运动估计后,再根据估计参数进行运动补偿,则差值图(图 3(b)~图 3(d))中主要都是前景公共汽车,背景基本上都被去除了,这说明全局运动估计是有效的。由图 3(b)中可以看到,还有一些背景点(栏杆)没有被有效去除,从图 3(c)、图 3(d)可以看出,GMC_2、GMC_3 算法去除背景点的

效果要好得多。图 3(c)、图 3(d)的效果差不多,但图 3(d)采用的算法(GMC_3)其时间要少于图 3(c)的算法(GMC_2)。

(2) Foreman 序列的第 80 帧、第 83 帧如图 4 所示,拍摄该图时,由于摄像机存在旋转运动,使得静止的背景出现全局运动,而前景人物也存在自身的局部运动。实验时,用 3 种算法分别对背景的全局运动参数进行计算,计算结果如表 2 所示。

表 2 Foreman 序列全局运动估计计算结果
Tab.2 GME result of Foreman sequence

| 计算方法 | 参数 a_1 | 参数 a_2 | 参数 b_1 | 参数 a_3 | 参数 a_4 | 参数 b_2 | 所用时间 (ms) |
|-------|-----------|----------|-----------|----------|-----------|------------|-----------|
| GMC_1 | -0.000 27 | 0.001 83 | 0.035 66 | 0.006 91 | -0.000 79 | -0.129 739 | 131 |
| GMC_2 | -0.002 01 | 0.003 12 | -0.431 00 | 0.006 72 | 0.001 92 | 0.218 957 | 120 |
| GMC_3 | -0.006 69 | 0.001 95 | -0.192 30 | 0.014 02 | 0.001 37 | -0.424 124 | 70 |

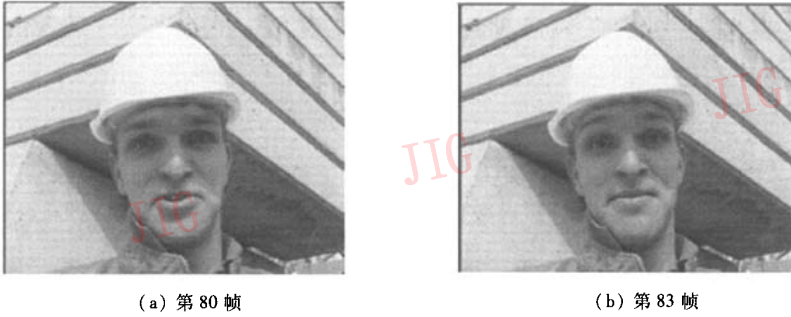


图 4 Foreman 序列图像

Fig.4 Images of Foreman sequence

从表 2 可以看出,本文提出的算法的计算时间只有 GMC_1 算法的 54%,GMC_2 算法的 58%,图 5 为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,从没有运动补偿的差值图(图 5(a))可以看到,前景和背景都有,而进行全局运动估计后,再根据估计参数进行运动补偿,差值图(图 5(b)~图 5(d))中主要都是前景人物头部,背景基本上都被去除了,这说明全局运动估计是有效的。由图 5(b)中可以看到,还有一些背景点(墙)没有被有效去除,从图 5(c)、图 5(d)可以看出,本文算法与 GMC_2 算法去除背景点的效果要好得多,而本文

提出的算法的计算结果要更准确,去除的背景点更有效,比较图 5(c)和图 5(d)可以看到,图 5(d)中的人物左肩膀后的背景被有效去除了。

(3) Coastguard 序列第 70 帧、第 74 帧如图 6 所示,由图 6 可见,由于水中的两艘船自身存在局部运动,同时摄像机也在运动,因此使得静止不动的岸边物体存在全局运动。实验时,用 GMC_1、GMC_2、GMC_3 3 种算法分别对背景的全局运动参数进行计算,计算结果如表 3 所示。

从表 3 可以看出,本文提出的算法的计算时间为 GMC_1 算法的 67%,GMC_2 算法的 75%,图 7 是根

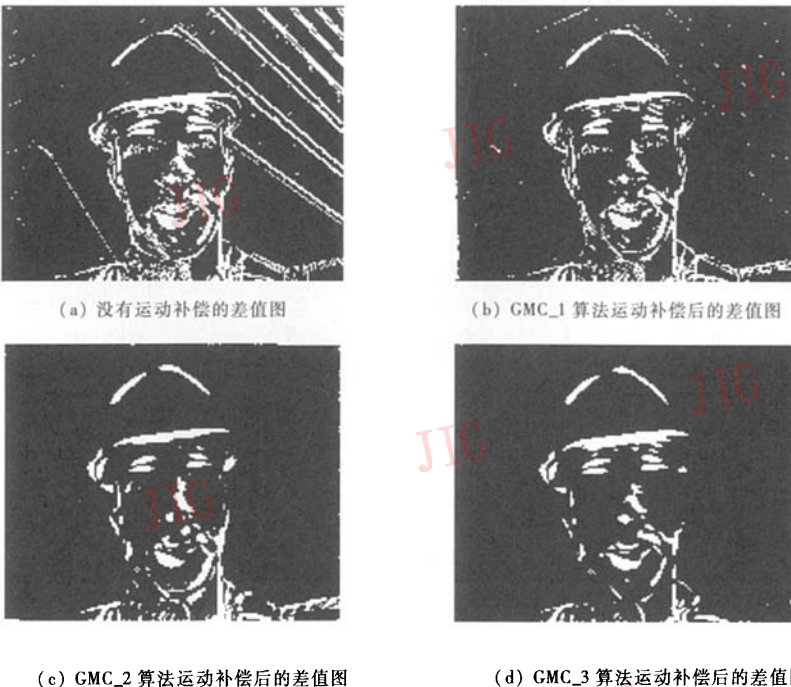
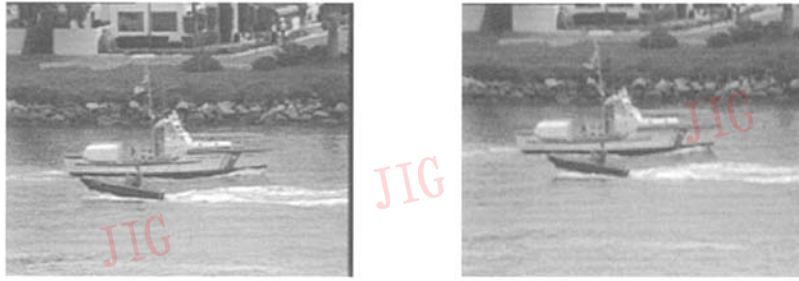


图 5 用 3 种不同算法进行运动补偿后的差值图经相同阈值分割得到的二值图

Fig.5 Binary image of absolute difference with motion compensation of three different algorithm



(a) 第 70 帧

(b) 第 74 帧

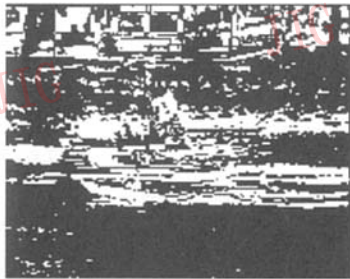
图 6 Coastguard 序列图像

Fig.6 Images of Coastguard sequence

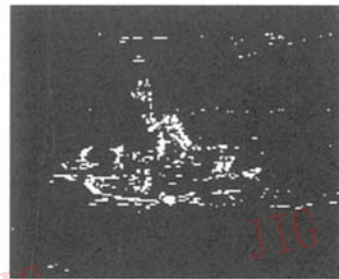
表 3 Coastguard 序列全局运动估计计算结果

Tab.3 GME result of Coastguard sequence

| 计算方法 | 参数 a_1 | 参数 a_2 | 参数 b_1 | 参数 a_3 | 参数 a_4 | 参数 b_2 | 所用时间 (ms) |
|-------|----------|----------|-----------|--------------|-----------|----------|-----------|
| GMC_1 | 0.002 33 | 0.005 08 | -0.870 95 | -0.000 381 | -0.003 93 | 13.215 6 | 180 |
| GMC_2 | 0.002 04 | 0.008 11 | -1.257 89 | -0.000 708 | -0.006 04 | 13.309 9 | 160 |
| GMC_3 | 0.003 62 | 0.005 40 | -1.607 02 | 9.150 4e-005 | -0.005 98 | 13.221 5 | 120 |



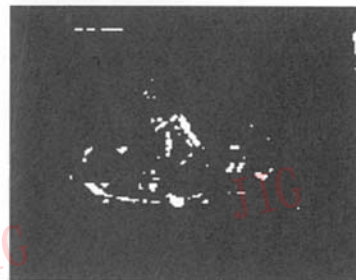
(a) 没有运动补偿的差值图



(b) GMC_1 算法运动补偿后的差值图



(c) GMC_2 算法运动补偿后的差值图



(d) GMC_3 算法运动补偿后的差值图

图 7 用 3 种不同算法进行运动补偿后的差值图经相同阈值分割得到的二值图

Fig.7 Binary image of absolute difference with motion compensation of three different algorithm

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。从没有运动补偿的差值图(图 7(a))可以看到,前景和背景都有,而进行全局运动估计后,

再根据估计参数进行运动补偿,图 7(b)~图 7(d)中主要都是前景两艘船,背景基本上都被去除了,这说明全局运动估计是有效的。由图 7(b)可以看到,

还有一些背景点(右上方)没有被有效去除,从图 7(c)、图 7(d)可以看出,GMC_2 算法和本文算法去除背景点的效果要好。图 7(c)、图 7(d)的效果差不多,但采用本文算法计算补偿的差值图(图 7(d))的运算时间要少于用 GMC_2 算法计算图 7(c)的运算时间。

5 结 论

本文提出了一种基于梯度的全局运动估计快速算法,并将这种全局运动估计算法和已有的算法进行了实验比较。从实验结果来看,无论从时间,还是从准确性上,改进的算法较之前的算法性能都有所提高。迭代算法成功的关键是减少噪声的影响,由于外点对于非线性迭代算法影响很大,所以本文从去除外点着手,同时结合特征点的选取,使得在加快计算速度的同时,计算准确性也得到了保证。MPEG-4 标准引入了基于模型的编码方法,可以极大地提高压缩比,这也是下一代编码方法发展的趋势,为此下一步工作就是将本文提出的全局运动估计算法应用到全局运动补偿编码中,以实现基于模型的编码,进一步改进现有的编码器的性能。

参考文献 (References)

- Smolic A, Sikora T, Ohm J R. Long-term global motion estimation and its application for sprite coding, content description, and segmentation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1999, 9(8): 1227 ~ 1242.
- Giaccone P R, Greenhill D, Jones G A. Creating virtual scenes using active contours and global motion estimation [A]. In: Proceedings Fourteenth International Conference on Pattern Recognition [C], Brisbane, Qld, Australia, 1998, 2: 1505 ~ 1507.
- Yu Tian-li, Zhang Yu-jin. Video retrieval based on the global motion information [J]. Acta Electronic Sinica, 2001, 29(12): 1794 ~ 1798. [俞天力, 章毓晋. 基于全局运动信息的视频检索技术 [J]. 电子学报, 2001, 29(12): 1794 ~ 1798.]
- Lu Yan, Gao Wen, Wu Feng. Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(5): 394 ~ 405.
- Yosi Keller, Amir Averbuch. Fast gradient methods based on global motion estimation for video compression [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(4): 300 ~ 309.
- Dufaux F, Konrad J. Efficient, robust and fast global motion estimation for video coding [J]. IEEE Transactions on Image Processing, 2000, 9(3): 497 ~ 501.
- Konrad J, Dufaux F. Improved global motion estimation for N3 [A]. In: Meeting of ISO/IEC/SC29/WG11, MPEG97/M3096 [C], San Jose, CA, USA, 1998.
- He Yu-wen, Zhao Li, Zhong Yu-zuo, et al. A fast and robust algorithm of global motion estimation [J]. Journal of Software, 2001, 12(8): 1220 ~ 1228. [贺玉文, 赵黎, 钟玉琢等. 快速鲁棒的全局运动估计算法 [J]. 软件学报, 2001, 12(8): 1220 ~ 1228.]
- ISO/IEC WG11 MPEG Video Group. MPEG-4 video verification model version 15.0 [A]. In: Meeting of ISO/IEC/JTC1/SC29/WG11, MPEG98/N3093 [C], Hawaii, America, 1999.