第49卷第3期 中国科学技术大学学报 Vol. 49, No. 3
2019年3月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Mar. 2019

文章编号:0253-2778(2019)03-0237-07

基于多路径优化的视频稳像方法

干文杰,张举勇,邓建松

(中国科学技术大学数学科学学院,安徽合肥 230026)

摘要:为了对日常生活中手持设备拍摄的视频进行稳像处理,提出了一种基于多路径优化的稳像方法.首先,进行运动估计,将输入视频的每一帧划分成均匀的网格,通过估计网格顶点的运动矢量得到帧间的运动;接着,进行运动平滑,将上一步的运动累加起来得到相机的运动轨迹,然后利用设计的多路径优化模型进行平滑处理;最后,进行运动补偿,对视频的每一帧进行补偿,得到稳像后的视频.将该方法和最近一些较好的方法进行处理时间和稳像结果的对比实验,结果表明,该方法耗时少,能取得令人满意的稳像效果.

关键词:视频稳像;相机运动轨迹;网格;多路径

中图分类号:TP391.41 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2019.03.009

引用格式:干文杰,张举勇,邓建松. 基于多路径优化的视频稳像方法[J]. 中国科学技术大学学报,2019,49(3):237-243.

GAN Wenjie, ZHANG Juyong, DENG Jiansong. A multi-path optimization based video stabilization method[J]. Journal of University of Science and Technology of China, 2019,49(3):237-243.

A multi-path optimization based video stabilization method

GAN Wenjie, ZHANG Juyong, DENG Jiansong

(*School of Mathematical Sciences, University of Science and Technology of China, Hefei* 230026, *China*)

Abstract: A multi-path optimization based video stabilization method was introduced, which is used to stabilize videos shot by hand-held devices in daily life. First, motions between two adjacent frames were estimated. Each frame of the input video was divided into a uniform mesh, and the motion of each mesh was estimated. Then the camera path was smoothed. The camera path was calculated by accumulating the motions estimated in the previous step, and a multi-path optimization based path smoothing model was designed to smooth the camera path. Finally, the stabilized video was calculated by warping each frame of the input video. The proposed method has been compared with some popular methods on running time and stabilization results. The comparison shows that the proposed method is efficient and can produce satisfactory results.

Key words: video stabilization; camera path; mesh; multi-path

收稿日期:2017-12-21;**修回日期**:2018-05-02

基金项目:国家重点研发计划(2016YFC0800501),国家自然科学基金(61672481)资助.

作者简介:干文杰,男,1992年生,硕士.研究方向:图像和视频处理. E-mail:ganwj@mail.ustc.edu.cn

通讯作者:张举勇,博士/副教授. E-mail:juyong@ustc.edu.cn

0 引言

随着数码设备的流行，我们能很方便地使用各种手持设备(手机，数码相机等)拍摄视频，但拍摄的视频往往存在视觉上的抖动，影响观看. 目前国内外防抖技术的研究方法包括了机械、光学、机电以及数字防抖等. 相比而言数字防抖技术成本低，且具有很好的效果，因此，数字视频稳像技术已经成为视频和图像处理领域的研究热点之一[1-5].

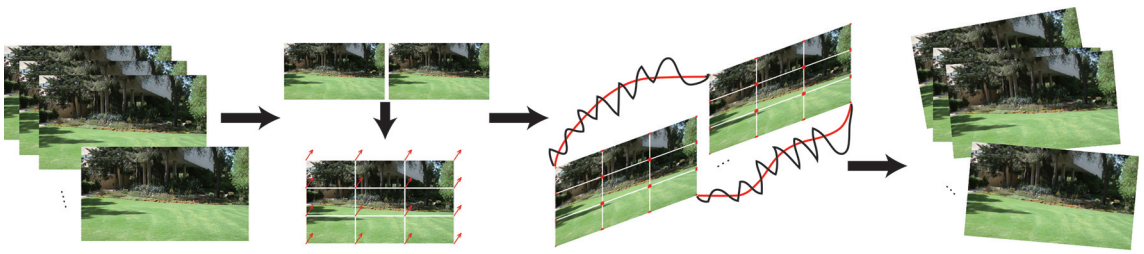
视频稳像的目的是去除视频中的抖动部分，得到视觉上稳定的结果. 数字视频稳像方法一般分成三个部分：①运动估计，估计出相机的运动轨迹；②运动平滑，对估计出来的相机运动轨迹进行平滑处理，得到平滑的运动轨迹；③运动补偿，利用平滑的运动轨迹和原来的运动轨迹之间的差，求出每一帧的运动补偿量，然后对每一帧进行运动补偿.

目前，已经有很多数字视频稳像的方法. 早期的一些方法主要使用全局的仿射变换矩阵或者单应变换矩阵作为相机的运动，然后使用一些简单的相机运动轨迹平滑方法，如使用低通高斯滤波[6-7]，卡尔曼滤波[8]和最小二乘拟合[9]等方法. 相比早些年的方法，近几年的运动估计或平滑方法相对比较复杂. Grundmann 等[10]利用影视学原理设计了 l_1 优化处理的运动平滑方法，他们的优化模型由三部分组成，分别是运动轨迹的一阶、二阶和三阶微分. Liu 等[11]将视频的每一帧划分成均匀的网格，然后估计每一个网格子块的运动轨迹，通过设计优化问题，求解得到平滑的运动轨迹. 他们的优化问题不仅考虑每个网格子块的运动轨迹的光滑性，也考虑到了子块间运动的一致性. Liu 等[12]改进原来的光流得到稳流，然后将稳流作为帧间的运动，通过利用类似文献[11]的方法光滑稳流得到稳定的视频. 现有的这些方法中，有一些[11-12]能得到比较好的结果，但是过于复杂，导致比较耗时；有一些[6-8]虽然效率高，但是得到的结果仍存在残余的抖动.

本文提出了一种新的视频稳像方法，也是由运动估计、运动平滑和运动补偿 3 个部分组成. 本文方法的主要贡献在于提出了一种新的运动估计方法，并应用文献[11]关于路径平滑的策略对本文的路径进行平滑处理. 实验结果表明，提出的方法速度快且稳像效果好.

1 方法概述

本文的方法主要分成 3 个部分，分别是运动估计、运动平滑和运动补偿，如图 1 所示.



黑色曲线表示原来的相机运动轨迹，红色曲线表示平滑处理后的运动轨迹.

图 1 本文方法流程图

Fig. 1 Overview of our method

①对输入视频进行运动估计. 对于输入视频中相邻的两帧，首先分别检测出它们的特征点. 然后将特征点进行匹配，利用匹配的特征点估计出帧间的仿射变换矩阵. 最后将前一帧划分成 $m \times n$ 的均匀网格，把仿射变换矩阵作用在网格上，然后就得到了每个网格顶点的运动矢量. 帧间的运动是由所有网格顶点的运动矢量组成的，如图 1(b)所示.

②对输入视频进行运动平滑. 我们将对应位置网格顶点的运动矢量累加起来，就得到了一条路径. 相机的运动轨迹是由所有网格顶点的路径组成的，如图 1(c)所示. 通过设计相应的平滑算法并求解，得到平滑的运动轨迹.

③对输入视频进行运动补偿. 利用平滑的运动轨迹和原来的运动轨迹之间的差异，计算出每一帧的运动补偿量，并对每一帧进行补偿，最后就得到了稳定的视频.

2 基于多路径的视频稳像模型

下面分别介绍本文方法的运动估计、多路径运动平滑以及运动补偿部分.

2.1　帧间运动估计

首先，估计帧间的全局运动. 在二维图像的变换模型[13]中，相邻视频帧序列间的全局运动主要表现为平移、缩放、旋转、错切和翻转等基本变换. 很多更复杂的变换由这些基本变换组合而成，例如刚性变换、相似变换和仿射变换. 考虑到稳像算法的普适性，本文采用仿射变换作为帧间的全局运动. 仿射变换的数学模型如下：

$$p'=\begin{pmatrix}x'\\y'\end{pmatrix}=\begin{pmatrix}a & b\\c & d\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}t_x\\t_y\end{pmatrix}=\boldsymbol{D}p+\boldsymbol{T} \quad (1)$$

记仿射变换 $\boldsymbol{A}=[\boldsymbol{D};\ \boldsymbol{T}]$，其中，$p'$ 和 p 分别是当前帧和前一帧中对应的像素坐标，$\boldsymbol{D}$ 表示缩放和尺度变换部分，$\boldsymbol{T}$ 表示平移量，t_x 和 t_y 表示水平和垂直方向的平移量. 可以看到，仿射变换的数学模型中包含 6 个自由度，它包含了刚性变换、相似变换以及错切变换和翻转变换. 仿射变换一个很重要的性质是它能保持平行性，也就是说，变换前的一对平行线，变换之后，它们仍然是平行线.

对于输入视频中相邻的两帧，分别设为第 $t-1$ 帧和第 t 帧，检测出它们的 SIFT（scale invariant feature transform）特征点[14]. SIFT 特征不仅具有尺度不变性，而且，即使改变旋转角度、图像亮度或拍摄视角，仍然能够得到好的检测效果. 接着，对检测出来的 SIFT 特征点进行匹配，并选出那些好的匹配，去掉那些差的匹配. 最后，根据匹配的 SIFT 特征点，利用 RANSAC 算法[15]估计出帧间的仿射变换矩阵. 如图 2(a)所示，第 $t-1$ 帧和第 t 帧相邻两帧间估计出来的仿射变换矩阵为

$$\boldsymbol{A}_{t-1}=\begin{bmatrix}0.944821 & -0.009939 & 29.377473\\0.003717 & 0.9455320 & 23.288247\end{bmatrix} \quad (2)$$

第 t-1 帧　　第 t 帧　　帧间的运动

(a)　　(b)

圆圈表示检测出来匹配的 SIFT 特征点

图 2　视频中第 $t-1$ 帧和第 t 帧(a)以及相邻两帧间估计的运动(b)

Fig. 2　Two adjacent frames of the input video(a) and the motion between two adjacent frames(b)

利用 $\boldsymbol{A}_{t-1}$，可以将前一帧的图像 I_{t-1} 近似地变换成后一帧图像 I_t.

在得到 $\boldsymbol{A}_{t-1}$ 之后，我们估计帧间的相机运动. 首先，我们将第 $t-1$ 帧划分 $m\times n$ 的均匀网格 M^{t-1}，然后将 $\boldsymbol{A}_{t-1}$ 作用在网格 M^{t-1} 上，就能得到网格 M^{t-1} 所有顶点的运动矢量. 具体地，设网格 M^{t-1} 中第 i 个顶点的位置为 p_i^{t-1}，则该顶点的运动矢量 $\boldsymbol{v}_i^{t-1}$ 可以表示为

$$\boldsymbol{v}_i^{t-1}=(\boldsymbol{D}_{t-1}p_i^{t-1}+\boldsymbol{T}_{t-1})-p_i^{t-1} \quad (3)$$

式中，$\boldsymbol{A}_{t-1}=[\boldsymbol{D}_{t-1};\boldsymbol{T}_{t-1}]$，$p_i^{t-1}$ 是第 i 个网格顶点所在位置的像素坐标. 帧间的运动由所有网格顶点的运动矢量组成，如图 2(b)所示.

2.2　基于多路径的运动平滑

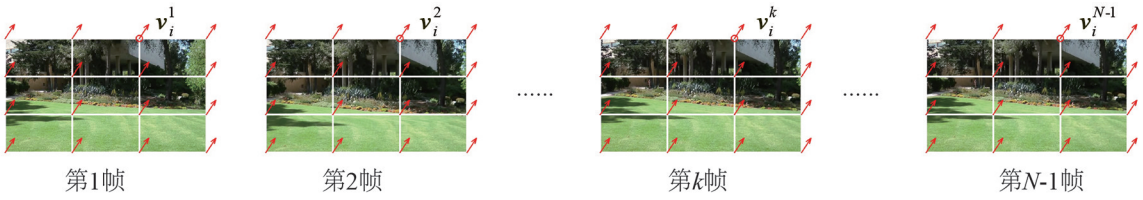
设输入的视频有 N 帧，我们将视频中的每一帧都划分成 $m\times n$ 的均匀网格. 通过上一步，得到了所有相邻帧间的运动，如图 3 所示. 接下来，我们计算相机的运动轨迹. 首先，选定每一帧网格中的第 i 个顶点，如图 3 中红色圆圈指出的顶点，该点在第 k 帧有运动矢量 $\boldsymbol{v}_i^k$，我们累加所有网格的第 i 个顶点的运动矢量，就得到了一条路径 $\boldsymbol{C}_i=\{C_i^k\}_{k=1}^N$，其中，

$$C_i^k=C_i^1+\sum_{j=1}^{k-1}\boldsymbol{v}_i^j,\ C_i^1=0 \quad (4)$$

C_i^k 表示第 i 条路径在 k 时刻的位置. 设相机的运动轨迹为 $\{\boldsymbol{C}_i\}_{i=1,\cdots,m\times n}$. 可以看到，相机的运动轨迹是由所有的网格顶点的路径组成的，如图 1(c)所示. 接下来，我们要对它进行平滑处理. 设平滑处理后的相机轨迹为 $\{\boldsymbol{P}_i\}_{i=1,\cdots,m\times n}$，同样的，有 $\{\boldsymbol{P}_i\}=\{P_i^k\}_{k=1}^N$.

首先，考虑对单条路径的平滑. 对于给定的一条路径 $\boldsymbol{C}$，我们想要得到平滑处理后的路径 $\boldsymbol{P}$，其中，$\boldsymbol{C}=\{C^k\}_{k=1}^N$，$\boldsymbol{P}=\{P^k\}_{k=1}^N$. $\boldsymbol{P}$ 可以通过极小化一个能量函数得到，该能量函数如下：

$$f(\boldsymbol{P})=\sum_{k=1}^{N}\left(\| P^k-C^k \|^2+\lambda\sum_{r\in\Omega_k}\| P^k-P^r \|^2\right) \tag{5}$$



红色箭头表示每个顶点估计出来的帧间运动矢量

图 3 每帧视频划分为均匀网格示意图

Fig. 3 Each frame of the input video being divided into an uniform mesh

式中,λ 是自定义的参数,Ω_k 表示第 k 帧的相邻帧. 式(5)中,数据项 $\| P^k-C^k \|^2$ 使得平滑处理后的路径尽可能地接近原始路径,这能保证稳定后的视频不会出现过度裁剪和很大的扭曲. $\| P^k-P^r \|^2$ 为平滑项,能去掉原始路径中的抖动部分.

Liu 等[11]提出了一种多路径平滑处理的视频稳像方法,他们将视频的每一帧网格化处理,计算每个网格顶点的路径,然后联合所有路径进行平滑处理. 他们指出,如果对单条路径都分别平滑处理,那么,路径之间的空间关系将不能得到保留,最后可能导致稳像后的视频中有扭曲的现象. 因此,我们采用文献[11]的策略,在对相机运动轨迹平滑的同时,也保留所有网格顶点路径之间的空间关系. 最后的相机运动轨迹的平滑处理是通过极小化下面的能量函数来完成的:

$$\sum_{i=1}^{m\times n}f(\{\boldsymbol{P}_i\})+\alpha\sum_{k=1}^{N}\sum_{j\in N(i)}\| P_i^k-P_j^k \|^2 \tag{6}$$

式中,α 是自定义的参数,$N(i)$ 表示第 i 个网格顶点的 1 邻域. 新引入的能量项 $\sum_{j\in N(i)}\| P_i^k-P_j^k \|^2$ 能够使得平滑处理后的路径具有一致性,也就是每一个网格顶点的路径和其邻域顶点的路径尽量接近,这样就能有效地减小扭曲.

通过求解(6)就可以得到平滑处理后的相机运动轨迹 $\{\boldsymbol{P}_i\}_{i=1,\cdots,m\times n}$. 由于优化问题(6)是二次的,因此,可以转换成求解如下稀疏线性方程组:

$$\left.\begin{aligned}&P_i^k+2\lambda\sum_{r\in\Omega_t}(P_i^k-P_i^r)+2\alpha\sum_{j\in N(i)}(P_i^k-P_j^k)=C_i^k,\\&i=1,2,\cdots,m\times n,\ k=1,2,\cdots,N\end{aligned}\right\} \tag{7}$$

求解式(7)可以使用任何一种稀疏线性系统求解器,可以采用 Eigen 库、MKL(Math Kernel Library) 库和 Matlab 等求解. 本文采用基于 Jacobi 迭代[17]的方法,具体的迭代形式如下:

$$P_i^{k,(t+1)}=\frac{1}{\gamma}(C_i^k+2\lambda\sum_{r\in\Omega_k}P_i^{r,(t+1)}+2\alpha\sum_{j\in N(i)}P_j^{k,(t)}), \tag{8}$$

式中,$\gamma=2\sum_{r\in\Omega_t}\lambda+2\sum_{j\in N(i)}\alpha+1$,$P_i^{k,(t+1)}$ 表示第 $t+1$ 次迭代的结果,取迭代的初始值 $P_i^{k,(0)}=C_i^k$.

2.3 运动补偿

在得到了光滑的相机运动轨迹后,我们根据原来相机运动轨迹和求得的新的相机运动轨迹之间的关系,可以得到每一帧网格在新相机运动轨迹下的新网格,然后根据新网格和原网格之间的位置差异,求得每一帧的运动补偿量,最后对每一帧进行运动补偿,从而得到稳定的视频. 具体地,对于视频中的每一帧,我们已经将其划分成 $m\times n$ 的均匀网格,原来第 k 帧的网格为 M^k,设对应的新网格为 $\overline{M}^k$,其中,M^k 已知,$\overline{M}^k$ 未知,因此,我们需要确定 $\overline{M}^k$ 中每个顶点的位置. 设网格 M^k 的第 i 个顶点为 v_i^k,其位置为 p_i^k,相应的,设网格 $\overline{M}^k$ 的第 i 个顶点为 $\overline{v}_i^k$,其位置为 $\overline{p}_i^k$. $\overline{p}_i^k$ 可以通过下式求得:

$$\overline{p}_i^k=P_i^k-C_i^k+p_i^k \tag{9}$$

对式(9)的求解能求得 $\overline{M}^k$ 中所有顶点的位置. 在得到了 $\overline{M}^k$ 后,然后我们根据 M^k 和 $\overline{M}^k$ 中对应网格子块的位置(如图 4 所示,M^k 中由 v_i^k,v_j^k,v_m^k 和 v_n^k 构成的一个网格子块和 $\overline{M}^k$ 中由 $\overline{v}_i^k$,$\overline{v}_j^k$,$\overline{v}_m^k$ 和 $\overline{v}_n^k$ 构成的网格子块相对应),可以求出子块间的单应变换矩阵[13],而所有子块间的单应变换矩阵就是前面提到的运动补偿量. 然后利用每个子块对应的单应变换矩阵,对每个子块的图像进行几何变换,最后就得到了第 k 帧运动补偿后的图像,如图 5 所示. 通过对视频中所有帧进行运动补偿,就得到了稳定的视频.

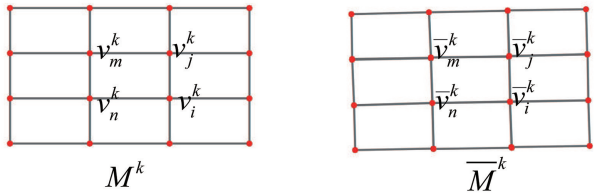


图 4　原网格和运动平滑后网格的比较

Fig. 4　Comparison of the original mesh with the smoothed mesh

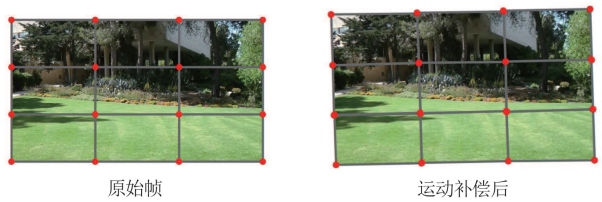


图 5　视频中第 k 帧运动补偿前后的比较

Fig. 5　Comparison of the frame with and without motion compensation

3　对比实验

我们选择最近 4 种较好的方法和本文方法进行对比实验，它们分别是 Wang 等[18]的时空优化方法(STO)、Liu 等[11]的 BCP 方法、YouTube 视频网站提供的在线视频稳定器以及 Adobe 公司的 Adobe After Effect CC 2015(AE)软件所带的视频稳定器. 所有对比实验的视频都是来自于文献[11]所提供的公共数据集.

我们用 C++实现本文的方法，并在一台 CPU 为 Intel 酷睿 i7 6900K，主频 3.2GHz，内存容量为 4GB 的台式电脑上进行对比实验. 在运动估计阶段，我们参考文献[11,19]的方法，把视频中的每一帧都划分成 16×16 的均匀网格，然后估计网格顶点的运动矢量. 在运动平滑阶段，我们通过大量实验发现，取 $\lambda=15$，$\alpha=5$ 总能得到较好的结果.

3.1　运行时间的比较

首先对 5 种方法的处理时间进行比较. 如表 1 所示，我们统计了 5 种方法处理不同分辨率的一帧图像所需的时间. 对于分辨率为 1280×720 像素的视频，STO 方法的处理时间为 450ms/帧，BCP 方法的处理时间为 392ms/ 帧，而 Youtube 方法和本文的方法分别只需要 40ms 和 45ms 处理一帧. 对于分辨率为 640×360 像素的视频，AE 方法的处理时间为 250ms/帧. 从表 1 可以看出，Youtube 方法和本文的方法在处理时间上要明显快于其他 3 种方法，而 Youtube 方法稍快于本文的方法.

表 1　各种方法的时间比较

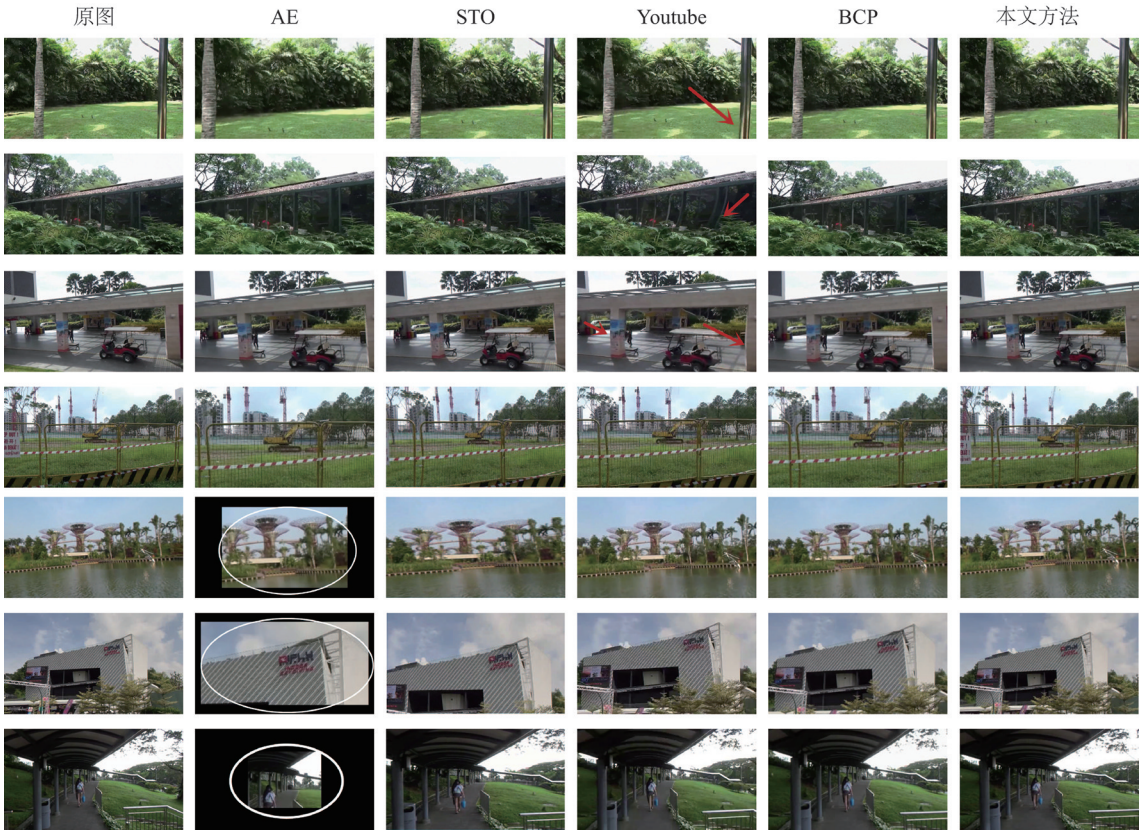
Tab. 1　The comparison of running time

视频稳像方法	分辨率/像素	t/ms
AE	640×360	250
STO	1280×720	450
Youtube	1280×720	40
BCP	1280×720	392
本文方法	1280×720	45

3.2　稳像结果的比较

我们选取了 100 多个公共的视频测试集来进行 5 种方法的稳像效果对比实验. 图 6 展示了 8 个比较结果，其中，第一列是原始视频中的某一帧，后面每一列都是不同方法产生的稳定视频中的一帧. 图中，红色箭头指出了扭曲部分，白色的圆圈指出该帧过度裁剪. 从红色箭头可以看到，Youtube 提供的在线视频稳定器处理后的视频中，容易产生较大扭曲. 白色圆圈可以看到，Adobe After Effect CC 2015 (AE) 软件所带的视频稳定器处理某些复杂的视频容易产生过度裁剪，也就是会有大量的信息损失. STO 方法、BCP 方法和本文方法的结果中基本没有很大的扭曲，在裁剪方面，本文方法的结果控制得更好，也就是说，本文方法的结果有更少的内容损失.

此外，我们选择 6 类视频邀请用户进行主观评价，6 类视频分别是 Simple 类、Quick Rotation 类、Running 类、Parallax 类、Zooming 类和 Crowd 类. 一共邀请了 50 名用户，他们的年龄在 18 到 50 岁之间不等，且职业背景不尽相同，有学生、教师、工人等. 在评价过程中，分别让用户将我们的结果和其他 4 种方法的结果进行对比. 每个用户一共需要接受 4 组对比实验，每组实验分别是评价本文的方法和其他 4 种方法中的 1 种. 例如，其中的一组实验，要求用户评价本文的方法和 STO 方法. 首先，我们从每一类视频中抽取 5 个视频，也就是一共 30 个视频，然后将 30 个原始视频、本文的结果和 STO 方法的结果展示给用户，让用户进行评价. 图 7 是用户调查的主观评价结果. 从用户主观评价结果来看，本文的方法在 6 类视频中，基本都要好于 Youtube 稳定器和 AE 稳定器. 对于 STO 方法和 BCP 方法，本文方法在 Quick Rotation 类和 Zooming 类上的结果要明显好于 STO 方法，在 Zooming 类上要明显好于 BCP 方法.



红色箭头指出扭曲的部分,白色圆圈表示过度裁剪.

图 6 本文方法与 AE,STO,Youtube 及 BCP 方法的比较

Fig. 6 The comparison of our method with AE, STO, Youtube stabilizer and BCP

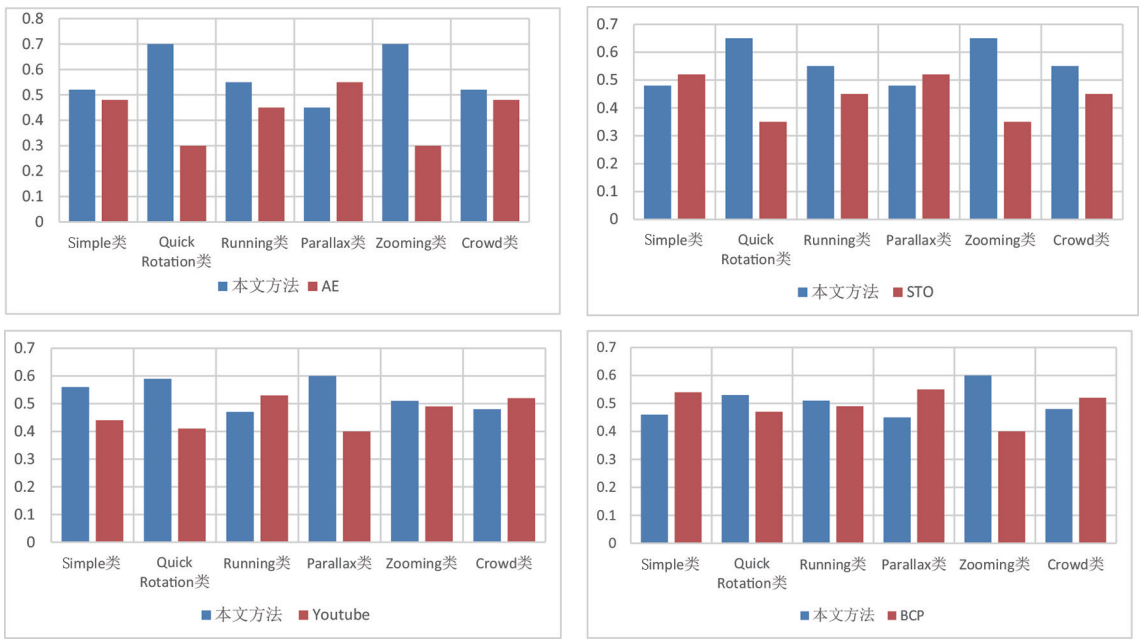


图 7 用户评价结果示意图

Fig. 7 The result of user study

4 结论

本文提出了一种基于多路径优化的稳像方法. 在运动估计的时候,我们将每一帧划分成均匀的网格,然后估计网格顶点的运动矢量. 在运动平滑的时候,考虑了裁剪、扭曲和稳定性,设计了多路径优化问题,并使用基于 Jacobi 迭代的方法求解. 实验结果表明,本文方法耗时少,且稳像结果令人满意.

由于本文的方法是基于特征点检测、匹配和帧间仿射变换的,因此,对于某些特征点较少的视频,会导致帧间的仿射变换估计得不准确,可能会导致最后的稳像结果不是很令人满意. 另一方面,本文在运动平滑的时候,参数 λ 和 α 的选取并没有采用自适应的方法,因此,经验性的参数选择很可能会对最后的稳像结果有一些影响. 未来的工作是改进目前的基于特征点检测的运动估计方法,在运动平滑处理时,设计自适应的方法.

参考文献(References)

[1] ZHANG L, CHEN X Q, KONG X Y, et al. Geodesic video stabilization in transformation space[J]. IEEE Transactions on Image Processing, 2017, 26(5): 2219-2229.

[2] LIU S, XU B, DENG C, et al. A hybrid approach for near-range video stabilization[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 27(9): 1922-1933.

[3] WANG Z Q, ZHANG L, HUANG H. Multiplane video stabilization[J]. Computer Graphics Forum, 2013, 32(7):265-273.

[4] LIU S, LI M, ZHU S, et al. CodingFlow: Enable video coding for video stabilization[J]. IEEE Transactions on Image Processing, 2017, 26(7): 3291-3302.

[5] LING Q, DENG S, LI F, et al. A feedback-based robust video stabilization method for traffic videos[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28(3): 561-572.

[6] LEE K Y, CHUANG Y Y, CHEN B Y, et al. Video stabilization using robust feature trajectories[C]// 2009 IEEE 12th International Conference on Computer Vision (ICCV). IEEE, 2009:1397-1404.

[7] MATSUSHITA Y, OFEK E, GE W, et al. Full-frame video stabilization with motion inpainting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(7):1150-1163.

[8] CHEN B Y, LEE K Y, HUANG W T, et al. Capturing intention based full-frame video stabilization[J]. Computer Graphics Forum, 2008, 27(7): 1805-1814.

[9] CHANG H C, LAI S H, LU K R. A robust and efficient video stabilization algorithm[C]// 2004 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2004: 29-32.

[10] GRUNDMANN M, KWATRA V, ESSA I. Auto-directed video stabilization with robust L1 optimal camera paths[C]// CVPR 2011. IEEE, 2011: 225-232.

[11] LIU S, YUAN L, TAN P, et al. Bundled camera paths for video stabilization[J]. ACM Transactions on Graphics, 2013, 32(4): 1-10.

[12] LIU S, YUAN L, TAN P, et al. Steadyflow: Spatially smooth optical flow for video stabilization[C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014: 4209-4216.

[13] HARTLEY R, ZISSERMAN A. Multiple View Geometry in Computer Vision[M]. Cambridge: Cambridge University Press, 2000:1865-1872.

[14] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[15] FISCHLER M A, BOLLES R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography[J]. Readings in Computer Vision, 1987: 726-740.

[16] NOCEDAL J, WRIGHT S J. Numerical Optimization[M]. New York: Springer, 2006.

[17] BRONSHTEIN I N, SEMENDYAYEV K A, MUSIOL G, et al. Handbook of Mathematics[M]. New York: Van Nostrand Reinhold Co, 1985.

[18] WANG Y S, LIU F, HSU P S, et al. Spatially and temporally optimized video stabilization[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(8):1354-1361.

[19] LIU S, TAN P, YUAN L, et al. MeshFlow: Minimum latency online video stabilization[C]// Computer Vision: ECCV 2016. Cham, Switzerland: Springer, 2016: 800-815.