

## 基于单应轨迹的视频实时稳像算法\*

董晶, 杨夏, 于起峰

(国防科技大学 航天科学与工程学院, 湖南 长沙 410073)

**摘要:** 针对视频稳像中实时任务的需求特点, 提出一种基于单应轨迹的视频稳像算法。估计序列图像帧间的单应变换, 并将该变换作用于图像窗口的4个角点, 从而对每帧图像产生4个短的单应轨迹以代表视频短时间内的运动。利用关联卡尔曼滤波器以一种连续方式对不同帧的单应轨迹进行平滑。对图像合成采用包括性和相似约束以提高结果视频的可视性质量。该算法以在线方式工作, 消除了缓存输入图像帧导致的延迟, 具备不依赖于复杂的3D重建和长距离特征跟踪的优点, 并有效避免了单应模型表达视频运动模型的误差积累问题。实验表明该算法能够有效对包含2D和较复杂3D场景的视频进行稳像, 并且能够达到实时处理速度。

**关键词:** 实时算法; 视频稳像; 卡尔曼滤波; 视频运动分析; 单应变换

**中图分类号:** TP316 **文献标志码:** A **文章编号:** 1001-2486(2014)02-0099-06

## A real-time video stabilization algorithm based on homography trajectory

DONG Jing, YANG Xia, YU Qifeng

(College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Motivated by the demands of real-time video stabilization, a real-time video stabilization algorithm based on homography trajectory is proposed. For each input frame, our approach regenerates four short homography trajectories by applying inter-frame homography transformations to the four corners of image rectangle. An associate Kalman filter is then performed to smooth these transformational trajectories. Finally, at the stage of image composition, constraints of inclusion and similarity are considered for selecting a visually plausible stabilized video. The proposed method can offer real-time video stabilization and it can remove the delays for caching coming images. In addition, our method does not rely on sophisticated 3D reconstruction or long-range feature tracking and it can effectively relieve the errors introduced by using homography to model the video motion. Experiments show that our approach can offer real-time stabilizing for videos with 2D scenes or the 3D scenes with moderate depth variation.

**Key words:** real-time algorithm; video stabilization; Kalman filter; video motion analysis; homography

随着视频采集设备易用化和小型化发展, 视频采集技术被广泛应用于移动设备(手机、平板电脑等)和各种小型载具上(无人机、机器人等)。由于平台的不稳定运动, 采集视频往往呈现出抖动现象。这会降低视频质量而难于观看。为了提高视频质量和观看效果, 可以通过机械方式消除设备抖动或是通过光学方式改变光路进行补偿。这两种方式往往能够提供较高质量的稳定视频, 但是它们存在价格昂贵或是笨重而不便操作的缺点。另一方面, 基于软件的稳像技术仅仅依赖于视频中的图像信息, 不需要额外的辅助设备或是采集设备的相关知识。因此, 基于软件的稳像算法为各种视觉任务, 如无人机探索<sup>[1]</sup>, 机器人导

航<sup>[2]</sup>和视频目标重定位<sup>[3]</sup>等, 提供了一种便利有效的解决方案。

通常, 基于软件的稳像算法可以分为三个步骤: 运动模型估计、运动平滑和图像合成。按照运动模型分类, 现存算法大致可以分为利用长距离特征轨迹<sup>[4-6]</sup>, 三维重建(单目)<sup>[7-8]</sup>和单应变换矩阵积累<sup>[1,9-10]</sup>来表示视频运动的方法。

其中基于特征轨迹的方法存在鲁棒性和计算量方面的缺点, 因为当场景快速变换时, 由于视场限制, 无法获得足够长的特征轨迹, 并且特征轨迹的提取容易受到图像噪声和物体遮挡干扰。此外模型估计需要足够多的特征轨迹, 而可靠跟踪大量特征需要大量计算并且极具挑战。

\* 收稿日期: 2013-08-20

基金项目: 国家自然科学基金资助项目(11002156); 国家部委资助项目

作者简介: 董晶(1984—), 男, 湖南郴州人, 博士研究生, E-mail: dongjing1756@163.com;

于起峰(通信作者), 男, 教授, 博士, 博士生导师, 中国科学院院士, E-mail: yuqifeng1756@163.com

基于三维重建的方法也同样存在计算量过大的缺点。此外单目三维重建方法依赖于视频中的景深变化。但是视频可能不存在足够的景深信息,比如对于平面场景或是当相机移动较小的时候。因此基于三维重建的稳像方法还存在适用性不足的缺点。

基于单应变换矩阵积累的方法计算代价小,并且相对于前两种方法,序列图像间的单应估计是一种已经成熟的技术,能提供更加可靠、鲁棒的结果。但是理论上单应模型只能适用于 2D 场景图像之间的变换或是相机没有位移时(相机光心的空间位置没有改变)拍摄图像之间的变换<sup>[20]</sup>,而无法表示相机在 3D 场景运动导致的景深的变化。因此当相机在一个 3D 场景运动,如果使用单应变换矩阵积累的方式来表达视频的运动,随着时间的增长,单应表达的运动和视频真实运动之间的误差会不断增加,最终会使得稳像算法失败。

本文算法使用单应轨迹来代表视频运动,因此本质上是基于单应变换模型的方法。但是,如果使用极其短的单应轨迹(由 3 个点组成)来代表视频在短时间内(3 帧之间)的运动,就可以使得单应表达的运动和视频真实的运动之间的误差很小。这是因为当时间足够短时,可以近似地认为相机没有位移,这符合使用单应表达运动的第二种情况。此外,本文算法使用的单应轨迹会根据视频的最新图像不断更新防止轨迹长度增加,从而减少了误差积累。实验证明本文提出的稳像算法对 2D 场景有效,并能够使用于 3D 场景下存在一定景深变化的视频。尽管本算法的模型对 3D 场景存在小的误差,但实验表明这种小的误差在稳像应用中是可以容忍的,毕竟稳像的最终目的是改善视频的可观质量而不是精确恢复视频运动。这种改进使得本文提出的稳像算法可以应用于更加多类型的视频,并且保持了基于单应模型算法的优点:快速和稳定。这两个优点对于实时应用是非常有益的。

另一方面,现存的稳像算法大多是以批处理<sup>[11-12]</sup>或延迟处理<sup>[3-4]</sup>方式工作。批处理方式的稳像算法致力于提高图像合成质量<sup>[11]</sup>,比如增加稳定视频的有效视场,降低运动模糊。但是计算量过大以及可能需要重复遍历视频<sup>[12]</sup>,使得这类方法无法适用于实时应用。另一方面,部分延迟处理方式的稳像算法能够达到实时处理速度,但是输入图像帧不能被立刻处理后输出,而是先缓存用于路径规划。因此,输出图像帧对输入图

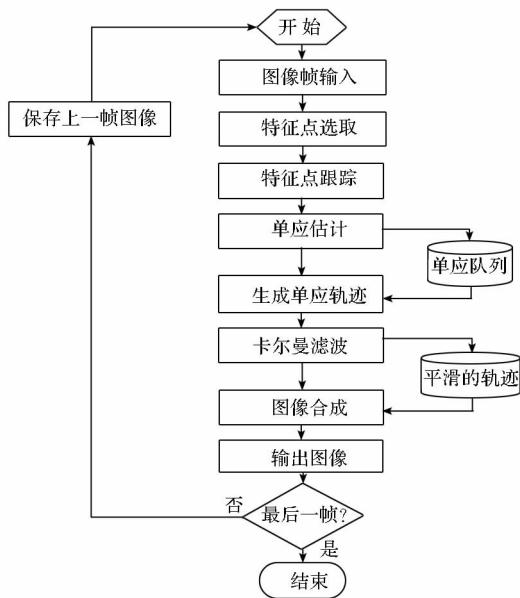


图 1 算法流程

Fig. 1 Flow chart of the proposed algorithms

像帧存在数帧的滞后。这种延迟对于一些反馈控制应用是无法接受的。

为了提供完全实时的稳像,本文使用卡尔曼滤波器代替需要帧缓存的路径规划方法。卡尔曼滤波器是一种标准的实时滤波器,能立刻修正输入数据。然而传统的卡尔曼滤波器需要较长的收敛过程,无法对过短的轨迹进行滤波。文献[6]将传统的卡尔曼滤波器用于平滑特征轨迹,并认为轨迹长度应该至少大于 20 才能得到较好的结果。这远远大于本文提出的单应轨迹长度。为了解决该问题,本文提出了一种关联卡尔曼滤波器,通过对滤波器的修正状态(corrected state)进行变换来关联不同帧之间的单应轨迹,这使得滤波器能够以一种连续的方式对不同帧的单应轨迹进行滤波,从而得到平滑的视频运动。最后算法根据平滑后的单应轨迹,在包括性和相似性约束下对输入图像进行变换,得到合成图像组成的稳定视频。图 1 给出了本文算法的流程。

## 1 运动估计

### 1.1 单应估计

图像间的单应估计存在多种算法<sup>[13-14]</sup>,它们可以用于我们的稳像算法中,但是往往存在计算量较大的问题。本文设计了一种针对序列图像间的单应估计算法,具有高效、计算代价可根据参数调整的特点。具体实现分为三步:特征点选取,特征点跟踪,单应解算。

#### 1.1.1 特征点选取

为了高效计算序列图像间的单应,特征点选

取应该具备以下几个特点:

1)选取的特征点应该广泛而均匀地分布于图像。如果特征点集中于局部的话,解算单应只代表了局部的变换,这会使得全局变换的误差较大,会造成晃动的假象。

2)相对背景越突出的特征点越有利于准确跟踪,因此应该被优先选取。

3)特征点选取的数量应该根据计算代价和单应估计的可靠性进行折中。本算法中将特征点数量设置为参数,有利于根据应用需要调整。

根据以上特点,特征选取算法设计如下:

1)使用 FAST 角点<sup>[15]</sup>检测算法对图像进行特征点检测,并计算特征点的响应(响应越大则越突出)。

2)按给定参数将图像均匀分为多个子区域(网格格式),在每个区域中选择响应最大的特征点。

3)若某个子区域中未检测出特征点,则选取子区域的中间点作为特征点。

### 1.1.2 特征点跟踪

特征点跟踪可以使用 KLT 算法<sup>[16]</sup>,模板相关<sup>[17]</sup>,和 SIFT - FLOW 算法<sup>[18]</sup>。其中 SIFT - FLOW 算法计算代价最大,难用于实时特征点跟踪。KLT 算法的计算代价小,可以有效跟踪突出的特征点,但是容易受到光照变化,运动模糊影响。此外对不突出的特征点,KLT 较容易跟踪错误。使用归一化的模板相关算法(NCC)可以弥补 KLT 的不足。首先它对光照变化适应更好,其次通过增加模板大小,模板相关能一定程度防止在运动模糊和弱特征的情况下跟踪出错。

本文将 KLT 和 NCC 算法结合进行特征跟踪。对每个选取的特征点按照响应值大小进行划分,响应小于给定阈值的点使用 KLT 跟踪,响应大于给定阈值的点使用 NCC 跟踪。NCC 的计算代价可以使用积分图像方法优化归一化计算过程和使用频域乘积优化相关(卷积)计算过程<sup>[17]</sup>,并使用金字塔策略进一步减少计算代价。由于序列图像间的位移通常较小,相关搜索被限定了范围,以提高效率。图 2 给出了本文特征点跟踪算法和完全使用 KLT 的特征点跟踪算法之间的比较。两幅图像为同一序列中的相邻两帧,其中黑线为 KLT 算法的跟踪结果,白线为 NCC 算法的跟踪结果。该图像拍摄时,相机正向左上角旋转(运动量较小),从右图可以看到本文方法给出了更多正确的特征点跟踪结果。



图2 特征点跟踪比较

Fig.2 Comparison of feature tracking

### 1.1.3 单应解算

对每帧图像选取特征点,并跟踪其在下一帧的位置,从而得到运动向量。根据特征跟踪得到的运动向量,可以解算得到帧间的单应矩阵。对于稳像算法,解算的鲁棒性应该优先于精确性。因为恢复的运动(参数或轨迹)通常需要经过平滑处理,从而使小的误差在稳定视频中难以察觉,但是错误的解算会使视频出现剧烈晃动。本文算法使用鲁棒性较强的(RANSAC)算法<sup>[19]</sup>解算单应矩阵,拒绝阈值设置为 1 像素。

### 1.2 单应轨迹

单应轨迹由一组二维的点组成  $\{P_i = |F_s \leq i \leq F_e\}$ ,其中  $F_s$  是轨迹起始帧的编号, $F_e$  是轨迹结束帧的编号。得到相邻图像帧间的单应变换后,单应轨迹可以按照式(1)迭代计算得到,其中  $H_{i-1}$  是图像帧  $i$  和  $i-1$  之间的单应变换。

$$P_i = H_{i-1}P_{i-1} \quad (1)$$

对于实时稳像算法,轨迹的结束帧就是当前输入帧。对每帧输入图像轨迹都会重新产生,并以图像窗口的 4 个角作为起始点。图 3 给出了一个例子:一条单应轨迹在每一帧输入时从图像窗口的左上角产生,其中虚线是上一帧输入时的单应轨迹,实线是当前帧输入时产生的单应轨迹。 $F$  是上帧轨迹中第一个点和第二个点之间的单应变换,当前帧输入时, $F$  将被删除,余下的单应变换将和新得到的变换  $N$  (上一帧和当前帧之间的变换)重新从左上角产生轨迹。

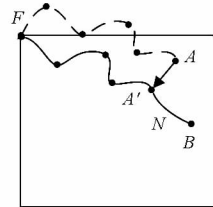


图3 单应轨迹

Fig.3 Homography trajectory

4 个图像角的单应轨迹将共同表示视频最近一小段的运动。由于轨迹很短且在不断更新,这避免了误差积累。

## 2 运动平滑

从图 3 可以看到当前帧的轨迹将重复上一帧轨迹从第二点开始的运动,并且增加一次新的运动。但是因为删除了单应变换  $F$ , 两条轨迹间存在跳变 ( $A$  到  $A'$ )。传统的卡尔曼滤波器 (Traditional Kalman Filter, TKF) 可以独立平滑两条轨迹,或是把跳变误认作观测数据的噪声,进行连续平滑。显然两种方式都难以取得好的效果。为此,本文提出了一种关联卡尔曼滤波 (Associate Kalman Filter, AKF),通过变换修正后状态避免跳变引起的噪声,并能够对单应轨迹进行连续平滑。AKF 的实现分为状态初始和修正后状态变换两个步骤。

### 2.1 状态初始

式(2)给出了 AKF 的状态模型,其中参数  $\hat{x}$  和  $\hat{y}$  表示轨迹的位移,参数  $\hat{x}^v$  和  $\hat{y}^v$  表示轨迹的速度,变量  $t$  为帧编号,代表时间的变换,  $N(\ast)$  是高斯噪声。

$$\begin{pmatrix} \hat{x} \\ \hat{x}^v \\ \hat{y} \\ \hat{y}^v \end{pmatrix}^t = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \hat{x} \\ \hat{x}^v \\ \hat{y} \\ \hat{y}^v \end{pmatrix}^{t-1} + \begin{pmatrix} 0 \\ N(0, \sigma_x) \\ 0 \\ N(0, \sigma_y) \end{pmatrix} \quad (2)$$

式(3)给出了 AKF 的观测模型,在观测模型中,只存在位移变量与相应的观测噪声,速度变量  $\hat{x}^v$  和  $\hat{y}^v$  是无法观测到的。

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} + \begin{pmatrix} N(0, \tilde{\sigma}_x) \\ N(0, \tilde{\sigma}_y) \end{pmatrix} \quad (3)$$

观测噪声可以根据估计的视频晃动量设置。过程噪声设定得越小,则滤波器估计的运动越平滑,但是对变化的反应越慢。通常过程噪声的方差可以设为(0.1~0.5)。

### 2.2 修正后状态的状态变换

以图 3 为例,AKF 需要根据最新的观测数据  $B$  来估计运动,但是此时 AKF 的状态还是基于上一帧的轨迹。因此需要将 AKF 的状态变换为基于当前轨迹的状态。由于两条轨迹的运动是重复的,只是发生了错位。因此,可以认为状态中的速度量( $\hat{x}^v$  和  $\hat{y}^v$ )是不变的,只是位移量( $\hat{x}$  和  $\hat{y}$ )发生改变。

如前描述,每一帧都产生 4 条单应轨迹,相应地将会有 4 个错位向量(如图 3 中,  $A$  到  $A'$  描述)。使用 4 个错位向量可以估计一个单应变换  $H$ ,并按照式(4)对修正状态中的位移量进行变换。

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix}^{t+1} = H \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix}^t \quad (4)$$

变换修正状态之后,AKF 再根据新的观测数据  $B$  估计当前帧的运动。从而避免了将错位误当作噪声,或是对每帧的轨迹进行独立的平滑。

## 3 图像合成

图像合成的原理是根据滤波器估计的运动和观测运动之间的差量对输入图像进行变换,从而使输出视频的运动更加平滑。然而为了提高视频质量,本文算法对图像合成进行了以下约束。

### 3.1 包括性约束

为了补偿晃动,图像变换会造成黑边,对此,通常的做法是裁剪输出图像。然而对于较大的晃动,补偿可能会造成很大的黑边,裁剪后的图像内仍然存在较大的黑边,严重影响视频质量。

本文算法通过检测裁剪窗口是否被变换图像完全包括来判断是否存在黑边。若存在,则减少图像变换的补偿量,直到裁剪图像中不存在黑边。但是,包括性约束会破坏滤波器估计运动的连续性,从而增加结果的不稳定性。图 4 中的例子,采取包括性限制的结果(右图)中黑边被消除。

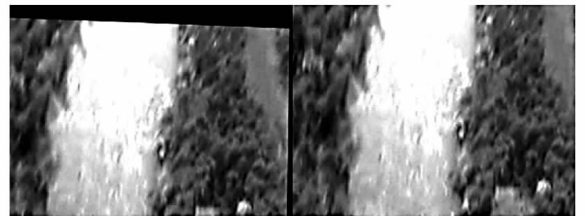


图 4 包括性约束

Fig. 4 Inclusion constraint

### 3.2 相似性约束

由于图像变换采用的是单应变换,会使输出图像的变换产生透视和剪切效应,从而产生变形。对此可以使用相似变换代替单应,消除透视和剪切效应。图 5 中的例子(原视频由文献[20]提供),采取相似性限制的结果(右图)中变形被消除。

相似变换只表示缩放、旋转和位移三种运动。相对于单应变换,相似变换能补偿的运动复杂度较低,因此相似性限制也会增加结果的不稳定性。因为透视和剪切效应对于远景视频往往难于察觉,而对于近景视频则容易察觉(影响视频质量),本算法将相似性约束作为可选约束。

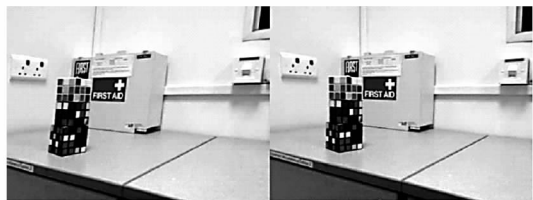


图5 相似性约束

Fig.5 Similarity constraint

## 4 实验结果

### 4.1 算法比较

我们使用一组在室内拍摄的视频(3D 场景,图8中第4组)和一组由无人机拍摄的视频(2D 场景,图8中第2组)对本文算法(采用 AKF)和文献[9]的算法(采用 SKF)进行了比较。为了量化视频的稳定性,采用高斯高通滤波对两种算法的输出视频中的特征轨迹进行滤波,用得到的高频能量作为视频晃动的度量。越不稳定的视频,其特征轨迹越不平滑,得到高频能量就越大,反之越小。

图6给出了量化结果,其中竖直坐标表示高频能量。两种算法比较的结果显示两种算法对于2D 场景视频的稳像效果接近,都能消除视频中的大部分晃动。但是对于3D 场景的视频,本文的算法结果要远远优于文献[9]的算法。事实上,文献[9]的算法无法用于3D 场景,量化结果显示其输出视频的晃动量甚至高于原来的视频。

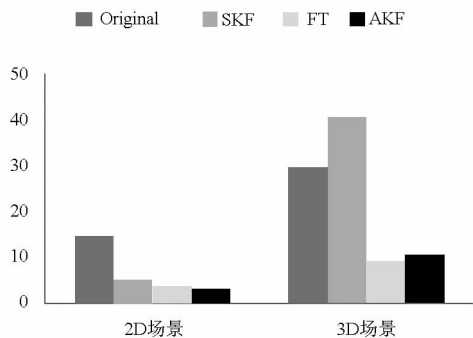


图6 算法比较

Fig.6 Comparison of algorithms

文献[4]提出了一种基于特征轨迹(Feature Track, FT)的稳像算法可以用于3D 场景的视频。该算法使用了延迟处理模式,以获得更好的路径规划,并且计算量较大(主要开销为特征轨迹的获取),因此无法用于实时。本文算法以实时处理方式与其进行了比较,图6中的量化结果显示,文献[4]的方法对3D 场景下视频得到了更加稳定的结果。然而文献[4]的方法对一组存在运动模糊的

视频和较大前景运动目标的视频处理失败(图7给出了两个视频的图片示例,原视频由文献[4]提供)。第一个视频(图7左)处理失败的原因是运动模糊中断了特征轨迹的提取。第二个视频(图7右)处理失败的原因是运动目标阻断了过多特征轨迹,而使图像变换时依赖的控制点过少,从而导致图像畸变和晃动。本文算法对这两组视频都给出了较稳定的结果,因为单应估计是一种全局变换估计,对运动模糊和局部遮挡的适应性更好。



图7 视频图例

Fig.7 Samples from video

### 4.2 算法性能

本文算法对各种场景的视频进行了广泛的测试,部分结果可以从文献[21]下载观看。图8给出了稳像结果的图片示例,其中黑线是特征轨迹,



图8 测试结果

Fig.8 Testing result

反映了视频的运动。从图中可见,稳定后视频的运动(右列)要比原来的视频更加平滑(左列)。

本文算法在一台笔记本上,使用单核 2.5GHz 处理器进行了测试。算法由 Visual Studio 2010 编译,编译优化为 o3,部分模块使用 opencv 实现,没有使用 GPU 或是其他图像处理硬件加速。表 1 给出了一些测试视频的信息和算法对视频的处理速度。结果表明算法在实验机上,对分辨率为  $640 \times 480$  和  $320 \times 240$  的图像均达到了实时处理要求。

表 1 视频信息和算法处理速度

Tab. 1 Videos' information and processing speed of proposed algorithm

序号 (图 8 由 上自下)	分辨率 (像素)	视频时间 /s	处理速度 /fps
1	$320 \times 240$	66	104.3
2	$640 \times 480$	38	66.4
3	$320 \times 240$	16	97.4
4	$640 \times 480$	32	40.1
5	$640 \times 480$	35	45.2

## 5 总结

针对现有稳像算法大多以批处理或延迟处理的问题,提出了一种新的实时处理的稳像算法。该算法使用单应轨迹表示视频运动,并用改进的卡尔曼滤波器对轨迹进行滤波。实验表明算法可广泛适用于各种视频,并且比较传统的算法,本文算法在可靠性和计算速度上有较大改进。

本文算法基于单应估计,因此潜在的假设是视频运动是连续和刚性的。对于过大遮挡(破坏了连续性)和弹性运动(如汹涌的海面),单应模型完全无法描述这类运动,这时本文算法会失败。将来我们将考虑结合其他的运动模型改进算法的性能。

## 参考文献 (References)

- [1] Wang C T, Kim J H, Byun K Y, et al. Robust digital image stabilization using the Kalman filter [J]. Transactions on Consumer Electronics, 2009, 55(1):6-14.
- [2] Kurazume R, Hirose S. Development of image stabilization system for remote operation of walking robots[C]//Proceedings of IEEE International Conference on Robotics and Automation, 2000, 2:1856-1861.
- [3] Grundmann M, Kwatra V, Essa I. Auto-directed video stabilization with robust L1 optimal camera paths [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011:225-232.
- [4] Liu F, Gleicher M, Wang J, et al. Subspace video stabilization[J]. ACM Transactions on Graphics, 2011, 30(1):4.
- [5] Lee K Y, Chuang Y Y, Chen B Y, et al. Video stabilization using robust feature trajectories [C]//Proceedings of 12th International Conference on Computer Vision, 2009:1397-1404.
- [6] Ryu Y G, Roh H C, Chung M J, et al. Long-time video stabilization using point-feature trajectory smoothing [C]//Proceeding of IEEE International Conference on Consumer Electronics, 2011:189-190.
- [7] Zhang G F, Hua W, Qin X Y, et al. Video stabilization based on a 3D perspective camera model[J]. The Visual Computer, 2009, 25(11):997-1008.
- [8] Liu F, Gleicher M, Jin H L, et al. Content-preserving warps for 3D video stabilization [J]. Transactions on Graphics, 2009, 28(3):44.
- [9] Litvin A, Konrad J, Karl W C, et al. Probabilistic video stabilization using Kalman filtering and mosaicing [C]//Proceedings of SPIE Conference on Image and Video Communications and Processing, 2003:663-674.
- [10] Morimoto C, Chellappa R. Evaluation of image stabilization algorithms [C]//Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, 1998, 5:2789-2792.
- [11] Matsushita Y, Ofek E, Tang X O, et al. Full-frame video stabilization [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005:50-57.
- [12] Pan P, Minagawa A, Sun J, et al. A dual pass video stabilization system using iterative motion estimation and adaptive motion smoothing [C]//Proceedings of 20th International Conference on Pattern Recognition, 2010:2298-2301.
- [13] Kumar S, Azartash H, Biswas M, et al. Real-time affine global motion estimation using phase correlation and its application for digital image stabilization [J]. IEEE Transactions on Image Processing, 2011, 20(12):3406-3418.
- [14] Pang D, Chen H, Halawa S. Efficient video stabilization with dual-tree complex wavelet transform [R]. San Francisco: Stanford University Press, 2010.
- [15] Rosten E, Porter R, Drummond T. Faster and better: A machine learning approach to corner detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(1):105-119.
- [16] Lucas B D, Kanade T. An iterative image registration technique with an application to stereo vision [C]//Proceedings of International Joint Conference on Artificial Intelligence, 1981, 2:674-679.
- [17] Hii A J H, Hann C E, Chase J G, et al. Fast normalized cross correlation for motion tracking using basis functions [J]. Computer Methods and Programs in Biomedicine, 2006, 82(2):144-156.
- [18] Liu C, Yuen J, Torralba A. Sift flow: Dense correspondence across scenes and its applications [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(5):978-994.
- [19] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography [J]. Communications of the ACM, 1981, 24(6):381-395.
- [20] Liu S C, Wang Y T, Yuan L, et al. Video stabilization with a depth camera [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2012:89-95.
- [21] <https://www.dropbox.com/s/fhohq26cgxoym2m/videostab>