ROBUST GLOBAL MOTION COMPENSATION AND ITS APPLICATIONS

By

Seyed Morteza Safdarnejad

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering - Doctor of Philosophy

2017

ABSTRACT

ROBUST GLOBAL MOTION COMPENSATION AND ITS APPLICATIONS

By

Seyed Morteza Safdarnejad

This thesis presents algorithms for robust global motion compensation (GMC). GMC algorithms are used to remove camera motion and transform the video such that in the resultant video, the background appears static and the only motion rises from foreground objects. Many computer vision algorithms are tailored for static cameras, and using GMC as a pre-processing module, it is possible to apply these algorithms on videos from moving cameras. For instance, motion-based video analysis is strongly affected by camera motion. If camera motion is not compensated, it interferes with the motion of interest, such as motion of human, and renders the analysis problem to be more challenging.

Generally, in sequential schemes, GMC estimates the homography transformation between two consecutive frames by matching keypoints on the frames, and maps the second frame to the first frame.Then, by accumulating these transformations, a composite transformation is calculated which maps each frame to the global coordinate. However, existing GMC algorithms are sensitive to existence of foreground motion and fail easily in the case of considerable foreground motion or ambiguous and low texture background.

To address the challenges in GMC, first, we propose a Robust Global Motion Compensation (RGMC) algorithm which explicitly suppresses the foreground effect and utilizes a comprehensive probabilistic verification model to find the best mappings between consecutive frames. Despite the robustness offered by RGMC, we further identify the problem of temporal drift of the estimation, due to accumulation of errors in estimation of mappings between consecutive coordinates. Furthermore, to address the issues of sequential GMC, we propose a Temporally Robust Global Motion

Compensation (TRGMC) algorithm which by *joint* alignment of input frames, estimates accurate and temporally consistent transformations to the global coordinates. Joint alignment not only leads to the temporal consistency of GMC, but also improves GMC stability by using redundancy of the information.

Many applications can benefit from a reliable and accurate GMC algorithm. We first briefly look into these applications. Then, among the many applications, we investigate the problem of sequence alignment, and propose an alignment algorithm for non-overlapping sequences, enabled by performance of TRGMC. Given the transformation to a global coordinate, offered by TRGMC, and the capability of background reconstruction using TRGMC results, we are able to align sequences even if the spatial overlap between the sequences is minimal or nonexistent. To this end, we first spatially align the sequences such that extrapolated backgrounds are aligned well and trajectories of moving objects are spatially smooth in the global coordinate. Next, we temporally align the sequences based on the smoothness of spatio-temporal trajectory of moving objects across the fields of view of different cameras.

This thesis is dedicated to Samira, my beautiful considerate wife, whom without her help and encouragement I would have achieved far less in my life.

ACKNOWLEDGMENTS

Firstly, I would like to express my gratitude to my thesis advisor, Dr. Xiaoming Liu whose knowledge, understanding and dedication added considerably to my Ph.D. experience. Throughout my Ph.D., all the time he has spent on one to one dedicated discussions, thoughtful and helpful comments, and careful review of papers has prepared me better for this dissertation and a successful career in the future.

Also, I am extremely thankful of my advisor Dr. Lalita Udpa for her support and help during every single step of my Ph.D., and all what I had the chance to learn from her. Without her support, my graduate studies path would have been much more difficult. I am very grateful that I had the opportunity to learn from two advisors, by closely working with Dr. Liu and Dr. Udpa. Dr. Liu and Dr. Udpa definitely provided me with the skills that I needed to successfully complete my thesis. I would also like to thank the remainder of my committee members, Dr. Satish Udpa and Dr. Arun Ross for their valuable comments and contributions along the way.

I thank my fellow labmates both in computer vision lab, and non-destructive evaluation lab, for all the fun we had during the past years, and every thing I learned from them during discussions and meetings.

Last but not the least, I would like to thank my family: my parents and my brothers and sister for supporting me spiritually throughout writing this thesis and my life in general. I do not know how to thank my awesome wife, Samira, whom without her nothing in my life would have been as exciting and meaningful. Thanks for always being there for me.

TABLE OF CONTENTS

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xi**

LIST OF ALGORITHMS . . . . . **xv**

Chapter 1 Introduction and Contributions **1**
1.1 Organization 5
1.2 Contributions 6

Chapter 2 Background **8**
2.1 Homography transformation 8
2.2 RANSAC 10
2.3 Congealing 11

Chapter 3 Robust Global Motion Compensation **13**
3.1 Previous Work 14
3.2 Proposed Method 16
3.2.1 Foreground Suppression 17
3.2.2 Homography Verification Model 19
3.3 Experimental Results 25
3.3.1 Dataset 25
3.3.2 Parameters 26
3.3.3 Evaluation metric 27
3.3.4 Accuracy assessment 27
3.3.5 Computational cost 28
3.3.6 Qualitative evaluation 28
3.4 Conclusions 31

Chapter 4 Temporally Robust Global Motion Compensation **32**
4.1 Previous Work 35
4.2 Proposed TRGMC Algorithm 39
4.2.1 Formulation of keypoint-based congealing 40
4.2.2 Optimization solution 42
4.2.3 Weight assignment 43
4.2.4 Initialization 44
4.2.5 Outlier handling 45
4.2.6 Alignment of non-keyframes 46
4.3 Experimental Results 47
4.3.1 Baselines and details 47
4.3.2 Datasets and metric 48

4.3.3 Quantitative evaluation . . . 49
4.3.4 Qualitative evaluation . . . 51
4.3.5 Computational efficiency . . . 52
4.3.6 Accuracy vs. efficiency trade-off . . . 52
4.4 Conclusions . . . 52

Chapter 5 Global Motion Compensation Applications . . . 54
5.1 Motion panorama . . . 54
5.2 Raster scan of scenes/Image mosaic . . . 55
5.3 Background reconstruction . . . 55
5.4 Foreground segmentation . . . 56
5.5 Human action recognition . . . 56
5.6 Multi-object tracking (MOT) . . . 58
5.7 Spatio-temporal alignment of non-overlapping sequences . . . 58

Chapter 6 Spatio-Temporal Alignment of Non-Overlapping Sequences from Independently Panning Cameras . . . 60
6.1 Introduction . . . 60
6.2 Previous Work . . . 63
6.2.1 Jointly moving cameras . . . 63
6.2.2 Cameras following similar trajectories . . . 63
6.2.3 Stationary cameras at different views . . . 64
6.2.4 Time synchronization . . . 64
6.3 Proposed Method . . . 65
6.3.1 Notations . . . 66
6.3.2 Spatial alignment . . . 66
6.3.2.1 Global motion compensation . . . 67
6.3.2.2 Spatial alignment of overlapping backgrounds . . . 69
6.3.2.3 Spatial alignment of non-overlapping sequences . . . 69
6.3.3 Temporal alignment . . . 74
6.3.3.1 Estimation of temporal offset . . . 75
6.3.3.2 Motion trajectory filtering . . . 76
6.4 Experimental results . . . 77
6.4.1 Dataset . . . 77
6.4.2 Qualitative results . . . 78
6.4.3 Quantitative results . . . 82
6.4.4 Computational cost . . . 83
6.4.5 Limitations . . . 83
6.5 Conclusions . . . 84

Chapter 7 Conclusion and Future Work . . . 85
7.1 Conclusions and discussions . . . 85
7.2 Future work . . . 88
7.2.1 Speed-up TRGMC . . . 88
7.2.2 Joint alignment and outlier rejection . . . 90

APPENDICES **91**
Appendix A Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis . . . 92
Appendix B Publications 110

BIBLIOGRAPHY **112**

LIST OF TABLES

Table 3.1: Impact of different settings on average BRE for each algorithm. DT and LT denote default ($\tau_s = 1000$) and lowered ($\tau_s = 100$) detection threshold in SURF algorithm, respectively. For RGMC $\tau_s = 100$ is used and 3 different setting of (T_C, T_M) are reported. D-M and D-E denote default setting of $(T_C, T_M) = (50, 100)$ with motion history and error handling turned off, respectively. 27

Table 4.1: Comparison of GMC algorithms on quantitative dataset (*GT: Ground truth, BF: Backward-Forward, B: Backward). 49

Table 4.2: Comparison of GMC algorithms on qualitative dataset. 52

Table 6.1: Temporal and spatial alignment error in seconds and pixels, respectively, for real (R) and synthetic (S) sequences. 77

Table A.1: Comparison of multiple datasets for action recognition (AR), scene understanding (SU), and genre categorization (GC). 98

Table A.2: Performances (genre categorization accuracy) of different baseline algorithms on SVW. 105

Table A.3: Performance of different combinations of trajectory descriptors on SVW. . 106

LIST OF FIGURES

Figure 3.1: RGMC algorithm flowchart: (a) color indicates various motion vector clusters, (b) the merged cluster of background, (c) the motion history, and (d) the motion compensated video. 17

Figure 3.2: (a) Motion history $\mathbf{M}_t$, (b) Mask $\mathbf{M} = \mathbb{I}(\mathbf{M}_t > \tau)$, (c) edge matching for an accurate θ_t that matches the background, (d) edge matching for an inaccurate θ_t that matches the foreground. 21

Figure 3.3: (a-b) Two consecutive frames and the matched quadruplet by the labeler, (c) the absolute difference of two frames matched via the quadruplet in (a,b), (d) manually labeled foreground mask. 24

Figure 3.4: Empirical and fitted distributions for (a) $E \sim N(0.52, 0.04)$, (b) $\Delta s \sim N(0, 2 \times 10^{-5})$, (c) $\Delta\alpha \sim N(0, 2 \times 10^{-3})$, (d) $t_x \sim Laplace(0, 1.50)$, (e) $t_y \sim Laplace(0, 0.95)$, and (f) $H \sim N(2.1, 0.25)$ 25

Figure 3.5: Sample frames of the test videos in (a) SVW, (b) HMDB51, and (c) Hollywood2 datasets. 26

Figure 3.6: Each row shows GMC results of two consecutive frames from video ID S17, S19, and S9 by (a) manual labeling, (b) MLESAC, (c) HEASK, and (d) RGMC. In (a), colorful pixels show the pixels that are different between overlaid frames. In (b-d), the pixel brightness indicates the difference. 29

Figure 3.7: Per-video BRE using the optimal setting for each algorithm compared with ground truth (GT) matching BRE. 30

Figure 3.8: A 40-frame sequence of gymnastics backflips in textureless background stitched using (a) MLESAC, (b) HEASK, and (c) RGMC. Consistency of the background shows the superiority of RGMC. For HEASK, stitching up to frame #10 is shown, after which the stitching drastically fails. 30

Figure 4.1: Schematic diagrams of proposed TRGMC and existing sequential GMC algorithms, and resultant motion panorama for a video shot by panning the camera up and down. Background continuity breaks easily in the case of the sequential GMC [88]. 33

Figure 4.2: Flowchart of the TRGMC algorithm. 39

Figure 4.3: The notation used in TRGMC. 41

Figure 4.4: Comparison of the ratios of **background**-**foreground** matches for (a) sequential GMC and (b) TRGMC. 44

Figure 4.5: (a) The input frame, (b) the reliability map, with the red color showing higher reliability. 45

Figure 4.6: Average BRE of frame pairs versus the time difference between the two frames. 50

Figure 4.7: Top view of the frames and links (a) before and (b) after TRGMC. The parallel links in (b) show successful *spatial* alignment of keypoints. Average of frames (c) before and (d) after TRGMC. For better visibility, we show up to 15 links emanated per frame. 50

Figure 4.8: Composite image formed by overlaying the frame n on frame 1 for several videos after TRGMC. Left to right, top to bottom, n is equal to 144, 489, 912, 93, respectively. In the overlap region the difference between the frames is shown. 51

Figure 4.9: Error and efficiency vs. the keyframe selection step, Δf. 53

Figure 5.1: Temporal overlay of frames from different videos processed by TRGMC. Trajectory of the center of image plane over time is overlaid on each plot to show the camera motion pattern, where color changes from blue to red with progression of time. 55

Figure 5.2: Background reconstruction results. 56

Figure 5.3: Foreground segmentation: (a) Input frame, (b) reconstructed background, (c) difference of (a,b) on (a). 57

Figure 5.4: Dense trajectories of the (a) original video, and (b) TRGMC-processed video. 57

Figure 5.5: Multi-player tracking using [3] for a football video with camera panning to the right, before (top) and after processing by TRGMC (bottom). 59

Figure 6.1: (a) Top view of spatio-temporal FOV of two moving cameras capturing sequences S_1 and S_2; Non-overlapping sequences (NOS) may not even cover some common spatial region over the progression of time, i.e, no overall spatial overlap will exist. (b) Spatio-temporal alignment of NOS results in displaying sequences from multiple freely panning cameras in a common coordinate and at the correct time shift. 61

Figure 6.2: Various scenarios in spatio-temporal alignment of sequences: (a) jointly moving cameras, (b) independently moving cameras at different times following similar trajectories, (c) stationary cameras with different viewpoints, (d) the proposed independently panning cameras with non-overlapping sequences. 64

Figure 6.3: Flowchart of our spatio-temporal alignment algorithm. First, spatial alignment is performed by background reconstruction for each sequence (a) and aligning the backgrounds (b). Second, given the spatial alignment parameters, keypoint trajectories (c) are mapped to the world coordinate and the best temporal alignment in terms of continuity of moving object trajectories is found (d). Finally, spatio-temporal alignment parameters are used for displaying the sequence in a world coordinate system and at the correct time shift (e). 65

Figure 6.4: Spatial alignment of non-overlapping sequences using background extrapolation and smoothness of object trajectories. 70

Figure 6.5: Trajectories, tracks, and fitted space-time curve to the tracks from 3 videos. 74

Figure 6.6: Spatial alignment of non-overlapping sequences. Top to bottom: reconstructed backgrounds of two sequences with negligible overlap, extrapolated backgrounds, and aligned background with trajectory of moving objects overlaid on the background. 78

Figure 6.7: Each row shows spatio-temporal alignment results on a set of real NOS, with *some* overall spatial overlap. For each sequence, input frames at the estimated time shift and trajectories of moving objects in the world coordinate are shown. The input frames are transformed to the world coordinate to make a composite image via alpha blending. 79

Figure 6.8: Results for two synthetic NOS from an accident footage (S2). (a) Trajectories of moving objects, (b) aligned input frames, (c) original frame where the synthetic frames are cropped. 80

Figure 6.9: Each row shows spatio-temporal alignment results on a set of real NOS, with *no* overall spatial overlap. For each sequence, input frames at the estimated time shift and trajectories of moving objects in the world coordinate are shown. The input frames are transformed to the world coordinate to make a composite image via alpha blending. 81

Figure 7.1: Comparison of the links made by (a) TRGMC with (b) the links which might be made via the proposed speed-up scheme. Chosen links by minimum spanning tree are shown in red and links between adjacent frames are shown in blue. 89

Figure A.1: Sample frames from all 30 sports categories of SVW. 94

Figure A.2: SVW challenges: (a) Related equipment does not exist, (b) Background is cluttered and uncorrelated with the sport, (c) Uncommon camera angles increase the intra-class variations, (d) Multiple sports co-exist (1: Hurdling, 2: Long jump, 3: Cycling). 100

Figure A.3: Annotated actions categories ([343, 359, Forearm], [380, 400, Set], [438, 454, Spike]) within a video from *V*olleyball genre category. Since distinct actions from the same sport genre may share a common field, visual appearance alone is not enough for action recognition in SVW. 101

Figure A.4: Distribution of (a) number of participents in videos, (b) aspects of the action field and (c) camera views angles, in 30 categories. *I*rrelevant field is a field that from its appearance, the sports category cannot be deduced (e.g., practicing in the backyard). *S*hared field refers to the condition in which from just field appearance, more than one sports category might be inferred (e.g., track and field sports). *U*nique field is the one that just from field context, the corresponding sports category can be conjectured (e.g., Bowling tracks). 103

Figure A.5: Confusion matrices of (a) context-based and (b) motion-based categorization algorithms. 106

LIST OF ALGORITHMS

Algorithm 1: Robust Global Motion Compensation 18

Algorithm 2: TRGMC Algorithm 48

Chapter 1

Introduction and Contributions

Due to the boom of smartphones and the ever increasing amount of videos, video analysis has received much attention in computer vision. A variety of problems is defined for video analysis including activity recognition [45, 51, 52, 71, 81, 122], event/action detection [25, 47, 125], video categorization [29,73,118,126,129,130], video saliency detection [21,56,84,91,94,123,131,132], etc. Effective motion analysis is the gist of many vision problems, e.g., action recognition, video annotation and video surveillance. On the other hand, as the video analysis research is maturing, era of designing algorithms based on staged videos has passed and datasets of *unconstrained real-world* videos are emerging.

However, unconstrained videos bring in new challenges in video analysis. For instance, motion-based video analysis is highly affected by camera motion. Thus, global motion compensation algorithms (GMC) are used to remove *intentional* (due to camera pan/tilt/zoom) and *unwanted* (e.g., due to hand shaking) camera motion. GMC is utilized in applications such as video stitching, or as pre-processing for motion-based video analysis. Due to its importance, this dissertation focuses on GMC and how it might be used in different applications. The term “global motion compensation" is also used in video coding literature, where background motion is estimated roughly to enhance the video compression performance [40, 98], in some compression formats such as MPEG-4.

Normally, GMC estimates the homography transformation between two consecutive frames by matching keypoints on the frames, and maps the second frame to a global coordinate. To remedy outliers in keypoint matches, robust techniques are proposed for homography estimation, e.g.,

RANSAC [35], by assuming the number of outliers to the correct homography is less than inliers. However, in the presence of *predominant foreground*, i.e., moving objects and people, a larger proportion of the putative matches are mismatches. Predominant foreground may result from a higher percentage of coverage by foreground pixels, or occlusion, textureless and non-informative background, blurred background (e.g., camera following the foreground motion), or a combination of these reasons. In presence of predominant foreground, the common variations of RANSAC have little chance of selecting a minimal set of background keypoints by random sub-sampling in a limited number of iterations. Despite its importance, the predominant foreground problem has been overlooked in both video stabilization and GMC algorithms. Since GMC estimates homography between consecutive frames and then uses a cascade of homographies to map the current frame to the global motion-compensated coordinate, failure in GMC at a single frame affects all the subsequent frames. This renders the predominant foreground problem very common and significant. Thus, GMC robustness is highly desirable. GMC problem is also aggravated as speed of foreground motion increases, e.g., in sports videos.

To address the predominant foreground problem, we propose a robust GMC (RGMC) method for suppressing foreground keypoint matches and mismatches, enabling a reliable homography estimation in presence of predominant foreground and/or textureless background. Also, we propose a novel and efficient probabilistic model for homography verification that considers keypoint matching error and consistency of the image edges after warping, and benefits from motion history gleaned from prior matched frames. We demonstrate the superiority of RGMC on challenging videos from three video datasets, when compared with state-of-the-art methods.

Our further investigations reveal that the sequential processing scheme causes frequent GMC failures for multiple reasons: 1) Sequential GMC is only as strong as the *weakest* pair of consecutive frames. A single frame with high blur or dominant foreground motion can cause the rest of the

video to fail. 2) Generally, multiple planes exist in the scene. The common assumption of a single homography will accumulate residual errors into remarkable errors. 3) Even if the error of consecutive frames is in a sub-pixel scale, due to the *multiplication* of several homography matrices, the error can be significant over time [74]. These problems are especially severe when processing long videos and/or when the camera motion becomes more complicated. For instance, when the camera pans to left and right repeatedly, or severe camera vibration exists, the GMC error is obvious by exhibiting discontinuity on the background.Although RGMC introduces robustness to the failures, it still suffers from accumulation of error.

To address the issues of sequential GMC, we propose a temporally robust global motion compensation (TRGMC) algorithm which by *joint* alignment of input frames, estimates accurate and temporally consistent transformations to the global motion compensated coordinate. TRGMC densely connects pairs of frames, by matching local keypoints. Joint alignment (a.k.a. congealing) of these frames is formulated as an optimization problem where the transformation of each frame is updated iteratively, such that for each *link* interconnecting a keypoint pair, the spatial coordinates of two end points are identical. This novel *keypoint-based congealing*, built upon succinct keypoint coordinates instead of high-dimensional appearance features, is the core of TRGMC. Joint alignment not only leads to the temporal consistency of GMC, but also improves GMC stability by using redundancy of the information. The improved stability is crucial for GMC, especially in the presence of considerable foreground motion, motion blur, non-rigid motion like water, or low-texture background. The joint alignment scheme also provides capabilities such as coarse-to-fine alignment, i.e., alignment of the keyframes followed by non-keyframes, and appropriate weighting of keypoints matches, which cannot be naturally integrated in sequential GMC. Our quantitative experiments reveal that TRGMC pushes the alignment error close to human performance.

Many applications may benefit from an accurate and robust global motion compensation algo-

rithm. We briefly review these applications, namely human action recognition, motion panorama creation, multi-object tracking for moving camera and when visual cues are insufficient for reliable tracking, and spatio-temporal alignment of video sequences.

Furthermore, among many potential applications, we deeply investigate the problem of spatio-temporal alignment of multiple video sequences, captured by freely panning handheld cameras. We identify and tackle a novel scenario of this problem referred to as Non-Overlapping Sequences (NOS). NOS are captured by multiple freely panning handheld cameras whose field of views might even have no direct spatial overlap. However, over the progression of time, there are nearby regions in the scene that are observed by the cameras independently and probably at distinct time instants. This assumption is less restrictive than common region being observed by field of view of different cameras over progression of time, and obviously much less restrictive than the common requirement of direct spatial overlap between frames from different cameras. With the popularity of mobile sensors, NOS rise when multiple cooperative users capture a public event to create a panoramic video, or when consolidating multiple footages of an incident or crime scene into a single video. This enables reconstruction of events or crime scenes captured by amateur users.

To tackle this novel scenario, we first spatially align the sequences by reconstructing the background of each sequence using TRGMC algorithm and then registering these backgrounds, even if the backgrounds are not overlapping. To do this, first, reconstructed background images are extrapolated. Then, a cost function is defined and minimized such that while extrapolated backgrounds are aligned well, trajectory of moving objects leaving field of view of one camera and entering field of view of another camera are spatially smooth. Given the spatial alignment, we temporally synchronize the sequences, such that the trajectories of moving objects (e.g., cars or pedestrians) across sequences are consistent with the prediction of when a moving object leaving the field of view of a camera, would appear in the field of view of another camera.

Finally, to develop algorithms for analyzing user-generated videos, unconstrained and representative datasets are of great significance. For this purpose, we collected a dataset of *Sports Videos in the Wild (SVW)*, consisting of videos captured by users of a leading sports training smartphone app (Coach's Eye®) while practicing a sport or watching a game. The dataset contains 4000+ videos selected by reviewing $\sim 85,000$ videos and consists of 30 sports categories and 44 actions. Videos of sports practice, which frequently happens outside the typical sports field, have huge intra-class variations due to background clutter, unrepresentative environment, existence of different training equipment and most importantly, imperfect actions. On the other hand, using smartphones for video capturing by ordinary people, in comparison to videos captured by professional crew for broadcasting, leads to challenges due to camera vibration and motion, occlusion, view point variation, and poor illumination. Given various manual labels, this dataset can be used for a wide range of computer vision applications, such as action recognition, action detection, genre categorization, and spatio-temporal alignment. On the sport genre categorization problem, we also design the evaluation protocol and evaluate three different methods to provide baselines for future works.

1.1 Organization

The remainder of this thesis is outlined as follows. In Chapter 2, we present the related background and theory for global motion compensation. Chapters 3 presents robust global motion compensation (RGMC). The further refinements to GMC by TRGMC algorithm are discussed in Chapter 4. In Chapter 5, we briefly review potential applications of GMC, and in Chapter 6, we focus on one of these applications and propose an algorithms for spatio-temporal alignment of non-overlapping sequences. In Chapter 7 we present the conclusion and the proposed future work. Details on the

collected amateur sports videos dataset, SVW, is presented in Appendix A.

1.2 Contributions

In this thesis, the challenging problem of global motion compensation for real world videos is addressed. There are many factors rendering this problem challenging. The following contributions are made in consideration of these factors.

- To address the important challenge of foreground occlusion in global motion compensation, a novel sequential global motion compensation algorithm is proposed. Namely, Robust Global Motion Compensation (RGMC) explicitly suppresses the foreground effect on estimation of the homography between the consecutive frames. Further, to evaluate each candidate homography, a novel probabilistic verification model is proposed which integrates motion history, edge matching, and point matching scores for homography evaluation.

- A novel joint alignment algorithm named Temporally Robust Global Motion Compensation (TRGMC) is proposed. Benefiting from the joint alignment, TRGMC further avoids the temporal drift problem. Also, further robustness is achieved as unlike sequential schemes, failure in alignment of a single pair of consecutive frames will not affect all the upcoming frames.

- Among many potential applications of RGMC and TRGMC, we further investigate the problem of sequence alignment. Based on the capability of transferring the frames to a global coordinate and also background reconstruction, we identify a novel scenario in sequence alignment and propose an algorithm for it. Namely, the proposed algorithm is capable of performing spatio-temporal alignment of non-overlapping sequences from freely panning

cameras.

- A dataset of *Sports Videos in the Wild (SVW)* is collected, which consists of videos captured by users of a leading sports training smartphone app while practicing a sport or watching a game. SVW is more unconstrained than existing human action datasets, especially sports dataset. On the sport genre categorization problem, we also design the evaluation protocol and evaluate three different methods to provide baselines for future works.

Chapter 2

Background

In this chapter, an overview of related theory and mathematics for global motion compensation is presented. Our proposed algorithms rely on estimating homography transformation. So, we first cover this transformation. A robust algorithm for estimation of homography transformation from noisy keypoint matches is called random sample consensus (RANSAC) [35]. We also review RANSAC algorithm in this section. Finally, congealing as a technique for alignment of a stack of images is closely related to our proposed temporally robust GMC. Basics of congealing are also presented in this section.

2.1 Homography transformation

Homography estimation is a key step in many computer vision applications. Assuming a pinhole camera model, two images taken from different viewpoints from the same planar scene are related by a homography transformation. Under homography H, the 3D point coordinates of camera 1, i.e. $\mathbf{X}_1$, are related to the 3D point coordinates of camera 2, i.e. $\mathbf{X}_2$ via,

$$\mathbf{X}_2 = H\mathbf{X}_1. \tag{2.1}$$

Similarly, if a camera has pure rotation around its optical center, the images are related to each other through homography transformation.

In the homogeneous image coordinates $\mathbf{x}_1$ and $\mathbf{x}_2$, there is scale ambiguity, and thus we have,

$$\mathbf{x}_2 \sim H\mathbf{x}_1. \tag{2.2}$$

There are a wide range of techniques for homography estimation. A good survey on these techniques can be found in [1]. H is a 3×3 matrix, but as it is defined only up to a scale, the total number of degrees of freedom is 8. As each point correspondence between two images provides two constraints, four point correspondences are enough to find the homography describing the transformation between two given images.

A simple homography estimation algorithm is Direct Linear Transform (DLT). First, the relationship between the two corresponding points is written as

$$c \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.3}$$

where c is a non-zero constant, $H = \begin{pmatrix} h_1, h_2, h_3 \\ h_4, h_5, h_6 \\ h_7, h_8, h_9 \end{pmatrix}$, and $\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$ and $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ represent $\mathbf{x}_2$ and $\mathbf{x}_1$, respectively.

After some manipulation, we get the following equations,

$$-h_1x - h_2y - h_3 + (h_7x + h_8y + h_9)u = 0, \tag{2.4}$$

$$-h_4x - h_5y - h_6 + (h_7x + h_8y + h_9)v = 0. \tag{2.5}$$

Rewriting in the matrix form, we have,

$$A\mathbf{h} = 0, \tag{2.6}$$

where $A = \begin{pmatrix} -x, -y, -1, 0, 0, 0, ux, uy, u \\ 0, 0, 0, -x, -y, -1, vx, vy, v \end{pmatrix}$ and $\mathbf{h} = \begin{pmatrix} h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9 \end{pmatrix}^T$. Thus, by solving the equation 2.6, DLT algorithm finds the homography. Each point correspondence makes up for two rows in A, so if there are at least four corresponding points available, the resultant 8×9 matrix A may be used and the $1D$ null space of A is the solution space for $\mathbf{h}$.

2.2 RANSAC

The Random Sample Consensus (RANSAC), which first was introduced by Fischler and Bolles [35], is an algorithm for fitting a mathematical model to experimental data. Specifically, when data contains outliers, RANSAC fits the model by detecting the outliers and fitting to the inliers. A data item is considered as an outlier if it does not fit to the true model reflecting the true set of parameters. Interestingly, the percentage of outliers for which RANSAC can find a proper model can be larger than 50%, which is the breakdown point for many other techniques. In the context of homography estimation from keypoint correspondences, RANSAC is very suitable as false matches due to appearance ambiguities in keypoint matching are outliers to the homography model describing the relationship between the points from the two images.

RANSAC is an iterative algorithm with two steps:

1. Hypothesize: A sample subset containing minimal data items, e.g. 4 keypoint correspondences in the case of homography estimation, is randomly selected from the input dataset. Using only this subset, the model is estimated. This is in contrary to methods such as least square robust estimators which use all the available data, possibly with different weights, to

estimate the model.

2. Test: RANSAC identifies the elements in the input dataset which are consistent with the model, as the inliers to the model, or the consensus set.

The iterative procedure is repeated until the probability of finding a better consensus set drops beyond a certain threshold.

Over time, many different variations of RANSAC have been proposed. RANSAC can be sensitive to the choice of the correct noise threshold that defines which data points fit a model instantiated with a certain set of parameters. If the threshold is too large, then all the hypotheses may be ranked equally good. In contract, if the noise threshold is too small, the estimated parameters tend to be unstable, i.e. by adding or removing a single data item to the set of inliers, the estimate of the parameters may change considerably. For instance, to partially compensate for this undesirable effect, Torr et al. proposed MLESAC (Maximum Likelihood Estimation SAmple and Consensus) [107]. Instead of ranking each consensus set based on its cardinality, MLESAC evaluates quality of the consensus set by calculating its likelihood.

2.3 Congealing

Congealing refers to the problem of unsupervised alignment of an ensemble of images. Generally, the parametric nature of misalignment (translation, similarity, affine, etc.) should be known in advance and images should have similar content and appearance. The seminal work of Learned-Miller [53] utilizes a sum of entropy of ensemble of images as the cost function. To mitigate the sensitivity issue of this method, Cox et al. [18] propose a SSD (sum of squared differences) cost function, optimized via Gauss-Newton optimization method. The misalignment function ξ is

defined over a stack of N images,

$$\arg\min_{\Phi} \xi(\Phi) \tag{2.7}$$

where $\Phi = \{\theta_1, \theta_2, ..., \theta_{N-1}\}$ is the set of $N-1$ warp parameters vectors corresponding to the images in the stack. Parametric warp function for the pixel coordinate $\mathbf{x}$ is denoted by $\mathscr{W}(\mathbf{x};\theta)$. In the least squared congealing method of Cox et al. [18], the misalignment of image i, I_i, relative to the rest of the images in the stack is defined as,

$$\xi_i(\theta) = \sum_{j=1;j\neq i}^{N} [I_j - I_i(\theta)]^2. \tag{2.8}$$

The nonlinear Eqn. 2.8 is difficult to minimize, so, it is linearized by taking the first order Taylor expansion series around $I_i(\theta)$, and the increment $\Delta\theta$ is estimated using,

$$\arg\min_{\Delta\theta} \sum_{j=1;j\neq i} \left[I_j + \frac{\partial I_j(\theta)^T}{\partial \theta} \Delta\theta - I_i \right]^2 \tag{2.9}$$

where $\frac{\partial I_j(\theta)^T}{\partial \theta}$ are the steepest descent images calculated by $\frac{\partial I_j(\theta)^T}{\partial \theta} = \frac{\partial \mathscr{W}}{\partial \theta} \nabla I_j(\theta)$. The solution to Eqn. 2.9 is given by,

$$\Delta\theta = H^{-1} \left[\sum_{j=1;j\neq i}^{N} \frac{\partial I_j(\theta)}{\partial \theta} \big(I_j(\theta) - Ii \big) \right] \tag{2.10}$$

where

$$H = \frac{\partial I_j(\theta)}{\partial \theta} \frac{\partial I_j(\theta)^T}{\partial \theta} \tag{2.11}$$

is referred to as pseudo-Hessian. So, iteratively solving for $\Delta\theta$ and updating θ until convergence is obtained, will lead to the set of aligning warp parameters.

Chapter 3

Robust Global Motion Compensation

The objective of global motion compensation (GMC) is to remove *intentional* (due to camera pan/tilt/zoom) and *unwanted* (e.g., due to hand shaking) camera motion. GMC is utilized in applications such as video stitching, or as pre-processing for motion-based video analysis. Effective motion analysis is the gist of many vision problems, e.g., action recognition, video annotation and video surveillance. For instance, in action recognition as an important computer vision problem, motion analysis via dense trajectories has shown superior performance [87, 114, 116]. However, the moving camera often interferes with the motion of human, thus it is desired to compensate for camera motion. Note that a related problem is video stabilization, which aims to remove *unwanted* camera motion, while GMC removes both *intentional* and *unwanted* camera motion [24].

Normally, GMC estimates the homography transformation between two consecutive frames by matching keypoints on the frames, and maps the second frame to a global coordinate. To remedy outliers in keypoint matches, robust techniques are proposed for homography estimation, e.g., RANSAC [35], by assuming the number of outliers to the correct homography is much less than inliers. However, in the presence of *p*redominant foreground, i.e., moving objects and people, a larger proportion of the putative matches are mismatches.

Predominant foreground may result from a higher percentage of coverage by foreground pixels, or occlusion, textureless and non-informative background, blurred background (e.g., camera following the foreground motion), or a combination of these reasons. In presence of predominant foreground, the common variations of RANSAC have little chance of selecting a minimal

set of background keypoints by random sub-sampling in a limited number of iterations. Despite its importance, the predominant foreground problem has been overlooked in both video stabilization and GMC algorithms. Even for algorithms designed explicitly for robustness to foreground motion [24, 30, 63], predominant foreground is reported to cause failure. Since GMC estimates homography between consecutive frames and then uses a cascade of homographies to map the current frame to the global motion-compensated coordinate, failure in GMC at a single frame affects all the subsequent frames. This renders the predominant foreground problem very common and significant. Thus, GMC robustness is highly desirable. GMC problem is also aggravated as speed of foreground motion increases, e.g., in sports videos. We qualitatively investigate 500 videos from Sports Videos in Wild (SVW) dataset [89], and observe 35% failure, i.e., background instability, by the baseline method of MLESAC [107], in contrast to 5.1% failure for the proposed method. This demonstrates that the robustness problem is very common and severe for real-world videos.

The main contribution of this chapter is a robust GMC (RGMC) method for suppressing foreground keypoint matches and mismatches, enabling a reliable homography estimation in presence of predominant foreground and textureless background. Also, we propose a novel and efficient probabilistic model for homography verification that considers keypoint matching error and consistency of the image edges after warping, and benefits from motion history gleaned from prior matched frames. We demonstrate the superiority of RGMC on challenging videos from three video datasets, when compared with state-of-the-art methods.

3.1 Previous Work

Due to existence of outliers, robust techniques are widely used for homography estimation, e.g., RANSAC [35] and its variants such as Locally-Optimized RANSAC [17], MLESAC [107] and

Guided-MLESAC [106]. While RANSAC aims to maximize the number of inliers, MLESAC searches the best hypothesis that maximizes the likelihood via RANSAC, assuming that the inliers are Gaussian distributed and outliers are distributed randomly. To handle the same outlier issue, [57] directly rejects unreliable keypoint matches. However, in case of predominant foreground, problematic matches from the foreground are not unreliable in terms of appearance. Recent works focus on estimating the best or multiple homographies in case of multi-plane background [6, 69, 105, 109, 133]. For instance, Uemura et al. [109] segment each frame using color MeanShift algorithm to multiple regions denoting different *planes* in the background and find the dominant plane for homography estimation. Using RANSAC and based on the number of inliers for estimated homography for each region, the dominant background planes is found and used for final homography estimation. In contrast, we segment the frame to *foreground* and *background* regions by analyzing motion vector clusters, and remove foreground for robust GMC.

Many works concentrate on rejecting mismatches from point correspondence, by relying on the assumption that similarity of the mismatched key-points is not enough. For instance, in [57], keypoint mapping functions from frame I to I' and reverse, denoted as f and f' are learned. If a point mapped according to f and then f' is actually mapped back to its original coordinates, then the associated key-point is considered a good match. However, in case of moving foreground, matches can be reliable in terms of appearance similarity, but considered mismatch due to inconsistency with background transformation.

Yan et al. [124] propose a probabilistic framework to combine keypoint matching and appearance similarity to enhance estimation robustness. To model the latter, correlation coefficient between pixels is used. Despite the improved estimation accuracy, for textureless background the performance deteriorates. For large foreground, [124] tends to remove foreground, instead of background, motion. In contrast, we use edge matching as an appearance similarity measure with

a higher sensitivity and lower computational costs. Motion history-based foreground suppression minimizes its interference with homography estimation. Also, we use motion history to reduce the tendency of compensating the foreground motion.

If camera motion is modeled as 2D translation, simpler methods can be used for GMC. In [14], video stabilization is conducted using the cross-correlation between horizontal and vertical projection of the consecutive frames, by assuming that the largest variation between frames is due to 2D translation. [24] uses the same idea to estimate 2D translation. To improve the robustness to moving foreground, a RANSAC-like approach on projections of bands of the image is utilized. However, [24] fails if the foreground object is too large or the background is textureless, and the simplistic model of 2D translation is easily violated in real-world videos. Thus, we design our RGMC algorithm to minimize the effect of textureless background and large foreground on homography estimation.

3.2 Proposed Method

The main objective of Robust Global Motion Compensation (RGMC) algorithm is to be robust to the presence of predominant foreground. Thus, it is critical to suppress the foreground and rely on keypoint matches of the background for global motion estimation. We perform *f*oreground suppression by clustering motion vectors computed from keypoint matches and identifying potential clusters corresponding to the background, which are merged to provide a set of background keypoints for final homography estimation. As a key enabler for RGMC, a novel and reliable *h*omography verification model is presented to consider keypoint matching error and consistency of the edges of images after transformation, and benefit from motion history gleaned from previous frames. Fig. 3.1 shows the flowchart of the RGMC algorithm, with details presented in the following two subsections.

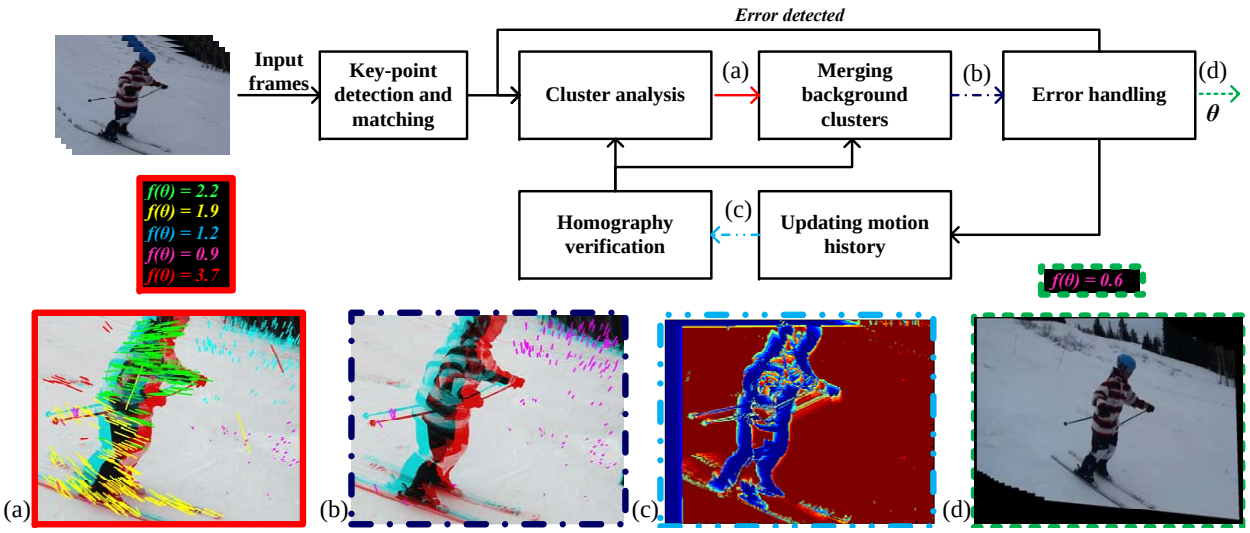


Figure 3.1: RGMC algorithm flowchart: (a) color indicates various motion vector clusters, (b) the merged cluster of background, (c) the motion history, and (d) the motion compensated video.

3.2.1 Foreground Suppression

We use SURF [10] algorithm for keypoint detection and description. To detect sufficient background keypoints, the Fast-Hessian keypoint detection threshold, τ_s, is decreased drastically. This helps in the cases of nearly uniform and textureless background, or blurred background due to rapid camera motion (e.g., videos shot by smartphones). However, this also implies that more keypoints will reside on the foreground, which calls for an effective foreground suppression.

Cluster analysis For foreground suppression, the motion vectors resulting from keypoint matches between consecutive frames are clustered. Since motion vectors on the background result from camera motion and are more *c*onsistent than foreground motion vectors, clustering will likely lead to some candidate regions from the background (see Fig. 3.1 (a)). Each cluster is analyzed separately by random subsampling of matches in that cluster and evaluating the resultant homography against the cost function, discussed in Sec. 3.2.2.

Merging background clusters Due to the zooming or motion corresponding to different planes of the background, and not knowing the optimal number of clusters a priori, we allow an over-

Algorithm 1: Robust Global Motion Compensation

Data: Frames $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$ and keypoints matches $\mathbf{D}$, prior homography θ_{t-1} and CFV $f(\theta_{t-1})$ and $f(\theta_{t-2})$

Result: Estimated homography θ_t and motion history $\mathbf{M}_t$

Compute the set of motion vectors $\mathbf{V}$ from $\mathbf{D}$;
repeat
  Cluster $\mathbf{D}$ into $\mathbf{D}_i$ $(i \in \{1,..,K\})$ based on $\mathbf{V}$, set $f_i = \infty$;
  for *i=1 to K* **do**
    while *Number of iterations* $< T_C$ **do**
      Randomly select four matching keypoints $\mathbf{Q}$ from $\mathbf{D}_i$;
      if $H(\mathbf{Q}) > p_{H,0.9}$ **then**
        Find homography $\hat{\theta}_t$;
        if *At least* λ*% of keypoints in* $\mathbf{D}_i$ *are inliers for* $\hat{\theta}_t$ **then**
          Calculate the cost function $\hat{f}$ via Eqn. 3.10;
          $f_i \leftarrow \min(\hat{f}, f_i)$.
  Regularize $\mathbf{D}_i$ to $\bar{\mathbf{D}}_i$ by randomly selecting a maximum of C matches for each cluster;
  Sort the f_i's in an ascending order and find the sorting index $j(i)$, set $m_i = \infty, (i \in \{0,..,K\})$, $i = 0$;
  repeat
    $i \leftarrow i+1$ and merge the top i clusters: $\mathbf{M}_i = \bigcup_{k=j(1)}^{j(i)} \bar{\mathbf{D}}_k$;
    while *Number of iterations* $< T_M$ **do**
      Randomly select four matching keypoints $\mathbf{Q}$ from $\mathbf{M}_i$;
      if $H(\mathbf{Q}) > p_{H,0.9}$ **then**
        Find homography $\hat{\theta}$ and calculate the cost function $\hat{f}$ via Eqn. 3.10;
        if $\hat{f} < m_i$, **then** $\theta_i \leftarrow \hat{\theta}$ and $m_i \leftarrow \hat{f}$.
  until $m_i > m_{i-1} \wedge i < K$;
  $\theta_t = \theta_{i-1}, f(\theta_t) = m_{i-1}$;
until $f(\theta_t) < \eta(f(\theta_{t-1}) + f(\theta_{t-2}))/2 \vee$ *Number of iterations* $< T_E$;
Update motion history via Eqn. 3.6 and output θ_t and $f(\theta_t)$.

clustering of K clusters. Thus, background motion vectors may be assigned to multiple clusters. To merge background clusters, based on the estimated homography and cost function value (CFV) of each cluster, a subset of the best clusters are selected to be merged in a greedy algorithm (Fig. 3.1(b)). Prior to merging, the set of keypoints belonging to each cluster are regularized by randomly selecting a maximum of C pairs for each cluster. Given that the keypoint matches in background cluster are similar, the regularization has negligible impact on the RGMC accuracy,

but remedies the case when part of the foreground (generally with a higher number of matches) is mistakenly merged to the background clusters.

Error handling For GMC applications such as video stitching or pre-processing for motion analysis, failed compensation and homography estimation for a single frame deteriorates the overall performance drastically. Since the context in consecutive frames are similar, we utilize the historical values of the cost function to assist the error handling. If the minimum CFV of homography estimation at the current frame pair is significantly higher than those of previous pairs, we repeat the estimation process with the hope that the randomness in the algorithm will recover the error.

Note that the significance of foreground suppression would be more obvious when plenty of keypoints belong to the foreground, while a few belong to the background. For instance, if foreground has 200 keypoints and background has 10, a RANSAC-like algorithm needs to run $450,000$ iterations to ensure a 90% probability of selecting a quadruplet of keypoints from background. However, by analyzing each cluster separately, RGMC efficiently focuses on background matches. Algorithm 1 summarizes the proposed RGMC algorithm. Details of the homography verification model used in the algorithm will be presented next.

3.2.2 Homography Verification Model

To evaluate the estimated homography from a quadruplet of keypoints matches, we derive a cost function that unifies the keypoint matching score, edge matching score, and the information from compensating previous frames. Denote the matching frames as $\mathbf{I}_{t-1}$ and $\mathbf{I}_t$, their candidate homography as θ_t, and the set of keypoint matches under study as $\mathbf{D}$. In Bayesian framework, similar to [124], θ_t can be estimated by maximizing

$$p(\theta_t|\mathbf{D},\mathbf{I}_t,\mathbf{I}_{t-1},\theta_{t-1}) = \frac{p(\mathbf{D},\mathbf{I}_t,\mathbf{I}_{t-1}|\theta_t,\theta_{t-1})p(\theta_t|\theta_{t-1})}{p(\mathbf{D},\mathbf{I}_t,\mathbf{I}_{t-1}|\theta_{t-1})}, \tag{3.1}$$

where θ_{t-1} is the obtained prior homography of frames $\mathbf{I}_{t-1}$ and $\mathbf{I}_{t-2}$. The $p(\theta_t|\theta_{t-1})$ is the conditional probability of θ_t given the prior homography θ_{t-1}. The denominator of Eqn. 3.1 is constant w.r.t. θ_t. By expanding the likelihood term, the homography can be verified using

$$p(\theta_t|\mathbf{D},\mathbf{I}_t,\mathbf{I}_{t-1},\theta_{t-1}) \propto p(\mathbf{D}|\mathbf{I}_t,\mathbf{I}_{t-1},\theta_t,\theta_{t-1})p(\mathbf{I}_t,\mathbf{I}_{t-1}|\theta_t,\theta_{t-1})p(\theta_t|\theta_{t-1}). \tag{3.2}$$

The term $p(\mathbf{D}|\mathbf{I}_t,\mathbf{I}_{t-1},\theta_t,\theta_{t-1}) = p(\mathbf{D}|\mathbf{I}_t,\mathbf{I}_{t-1},\theta_t)$ and represents how well the keypoint matches $\mathbf{D}$ extracted from $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$ are matched by θ_t. Knowing $\mathbf{I}_t$ is independent from θ_{t-1}, the term $p(\mathbf{I}_t,\mathbf{I}_{t-1}|\theta_t,\theta_{t-1}) = p(\mathbf{I}_t,\mathbf{I}_{t-1}|\theta_t)$, and reflects how well the frame $\mathbf{I}_t$ transformed under θ_t, denoted as $\mathbf{I}_{t|\theta_t}$, matches $\mathbf{I}_{t-1}$. Thus, the homography is estimated by minimizing,

$$\theta_t^* = \arg\min_{\theta_t}[-\ln(p(\mathbf{D}|\mathbf{I}_t,\mathbf{I}_{t-1},\theta_t)) - \ln(p(\mathbf{I}_t,\mathbf{I}_{t-1}|\theta_t)) - \ln(p(\theta_t|\theta_{t-1}))]. \tag{3.3}$$

Keypoint matching error Based on the analysis of Yan et al. [124], the keypoint matching error for inliers, $p(\mathbf{D_{in}}|\mathbf{I}_t,\mathbf{I}_{t-1},\theta_t)$, is better represented by a Laplacian model than the conventional Gaussian model. Denote (x_R^i, y_R^i) and (x_T^i, y_T^i) as the ith matching keypoint coordinates of $\mathbf{I}_t$ and $\mathbf{I}_{t-1}$ respectively, transformation of (x_R^i, y_R^i) under θ_t as (x_{RT}^i, y_{RT}^i), transformation of (x_T^i, y_T^i) under θ_t^{-1} as (x_{TR}^i, y_{TR}^i) , and $d_i \leftarrow |x_{TR}^i - x_R^i| + |y_{TR}^i - y_R^i| + |x_{RT}^i - x_T^i| + |y_{RT}^i - y_T^i|$. We use the same method as [124] to compute the keypoint matching error,

$$p(\mathbf{D}_{in}|\mathbf{I}_t,\mathbf{I}_{t-1},\theta_t) = \prod_{i=1}^{N_{in}} \frac{1}{16b^4} e^{-\frac{d_i}{b}} \tag{3.4}$$

where N_{in} is the number of inliers and the scale b is the Laplacian distribution parameter. Denoting γ_i as an indicator variable for inlier/outlier and considering that an outlier has a uniform distribution over the entire area of the frame, which is denoted as S, we have

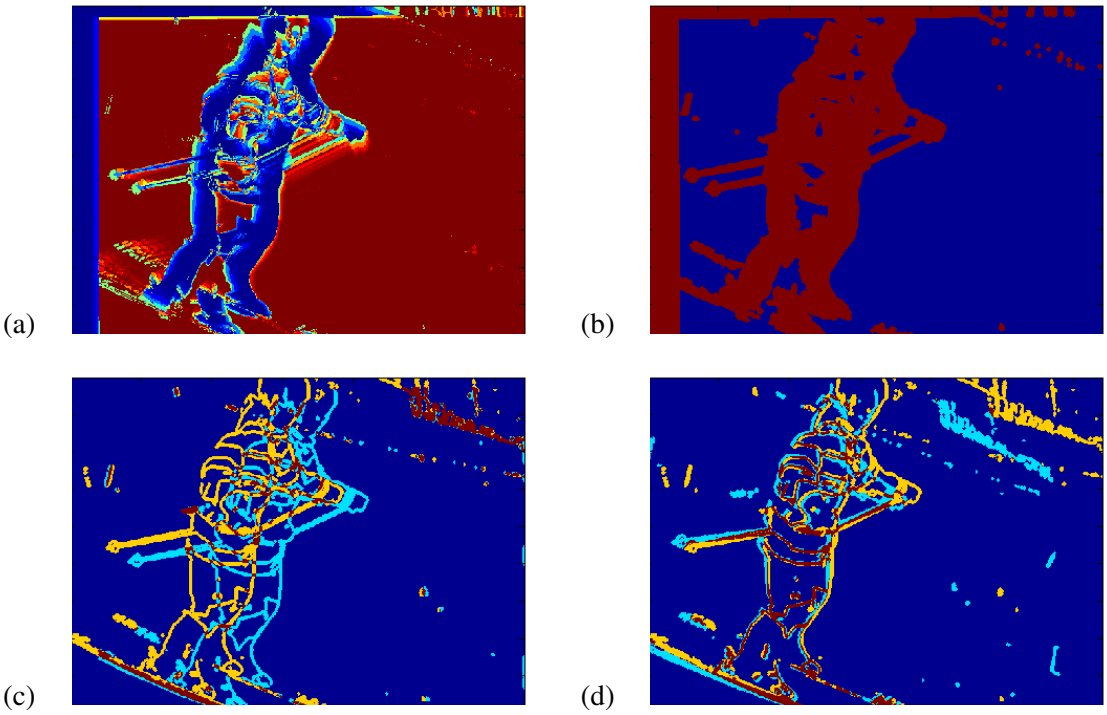


Figure 3.2: (a) Motion history $\mathbf{M}_t$, (b) Mask $\mathbf{M} = \mathbb{I}(\mathbf{M}_t > \tau)$, (c) edge matching for an accurate θ_t that matches the background, (d) edge matching for an inaccurate θ_t that matches the foreground.

$$p(\mathbf{D}|\mathbf{I}_t, \mathbf{I}_{t-1}, \theta_t) = \prod_{i=1}^{|\mathbf{D}|} [\gamma_i \frac{1}{16b^4} e^{-\frac{d_i}{b}} + (1-\gamma_i)\frac{1}{S^2}]. \tag{3.5}$$

Appearance consistency The appearance consistency under θ_t transformation, $p(\mathbf{I}_t, \mathbf{I}_{t-1}|\theta_t)$, is normally computed via pixel-based correlation [124]. We propose edge-based matching for multiple reasons. First, the pixel-based matching score is not sensitive enough for textureless background, e.g., a homography with error of few pixels displacement leads to similar scores as a perfect match. In contrast, the tolerance for error is much lower by matching the edges, which results in more accurate homography models. Although low-texture images produce few and generally noisy edge pixels, our experiments show that edge matching outperforms pixel-based correlation, even in low-texture conditions, similar to the results reported in [119]. Second, when stitching video frames based on global motion compensation, errors typically occur in mis-matched edges

at the boundary of the frames. These errors are very distracting for viewers' visual perception, and they are more likely to be remedied by edge-based appearance matching. Finally, in pixel matching, time-consuming image warping is needed for computing $\mathbf{I}_{t|\theta_t}$. Edge matching only needs to warp edge pixels in $\mathbf{I}_t$, leading to a typical $10\times$ speed-up over pixel matching.

To assure that the edge matching score reflects how well the background, not foreground, of the two frames match, we iteratively update a motion history $\mathbf{M}_t$ (see Fig. 3.1 (c)) as,

$$\mathbf{M}_t \leftarrow \alpha \mathbf{M}_{t-1} + (1-\alpha)|\mathbf{I}_{t-1} - \mathbf{I}_{t|\theta_t}|, \tag{3.6}$$

where α is a weighting scalar within 0 and 1, and $|.|$ denotes the element-wise absolute value operator. We define the edge matching score (EMS) as,

$$E(\mathbf{I}_1, \mathbf{I}_2, \mathbf{R}) = \frac{2\|\Phi(\mathbf{I}_1) \bigodot \Phi(\mathbf{I}_2) \bigodot \mathbf{R}\|_1}{\|\Phi(\mathbf{I}_1) \bigodot \mathbf{R}\|_1 + \|\Phi(\mathbf{I}_2) \bigodot \mathbf{R}\|_1 + c}, \tag{3.7}$$

where Φ is edge detection operator, $\bigodot$ is element-wise multiplication, $\mathbf{R}$ denotes the mask specifying the region of interest for EMS calculation, $\|\cdot\|_1$ computes the L_1 matrix norm, and $c(=0.001)$ is a constant to avoid division by zero. $E(\mathbf{I}_1, \mathbf{I}_2, \mathbf{R})$ ranges between 0 and 1 with 1 representing a perfect match. In Eqn. 3.3, we use $E(\mathbf{I}_{t-1}, \mathbf{I}_{t|\theta_t}, \mathbf{M})$, where $\mathbf{M} = \mathbb{I}(\mathbf{M}_t > \tau)$ is obtained by thresholding the motion history and $\mathbb{I}(\cdot)$ is an indicator function. Fig. 3.2 shows a motion history and edge matching results for two candidate θ_t's. We will later discuss how the probability model for E is obtained.

Conditional homography distribution Based on our experiments, and also prior work [24] on YouTube Action Dataset [62], the largest variation between consecutive video frames is due to 2D translation. Thus, to utilize the prior information of θ_{t-1} for a stable homography estimation, we

decompose the homography model into translation, scale, and rotation models [112]. Denote the absolute difference in components of θ_t and θ_{t-1} after decomposition as t_x and t_y for translation, Δs for scale and $\Delta\alpha$ for rotation angle. Assuming independence among components, we define

$$p(\theta_t|\theta_{t-1}) = p(t_x)p(t_y)p(\Delta s)p(\Delta\alpha). \tag{3.8}$$

Quadruplet filtering RGMC evaluates a large number of quadruplets of keypoint matches, and computes their EMS. To improve the efficiency, we filter the candidate quadruplets before the optimization of Eqn. 3.3. Intuitively, if the keypoint in the quadruplet are spatially close to each other, it is less likely to have an accurate estimate of θ_t, because homography estimation is more sensitive to the accuracy of keypoint locations. Also, background keypoints have generally a higher spatial dispersion than the foreground keypoints. Thus, only if the entropy (or dispersion) of a candidate quadruplet is above a threshold, we fully evaluate the cost function. Specifically, we use m-spacing estimate of entropy [54], similar to [24], as

$$H = \frac{1}{n}\sum_{i=1}^{n-m} \ln(\frac{n}{m}(x_{i+m} - x_i)), \tag{3.9}$$

where m is the spacing parameter (set to 1) and n is number of points. We first sort the x values prior to using them in Eqn. 3.9. Entropy estimates of x and y coordinates of the quadruplet are calculated separately and the minimum of them is the entropy of the quadruplet.

Model training Having presented the Bayesian framework, we now introduce our empirical approach to learn the various probability models. For this learning, we manually stitch 250 pairs of consecutive frames to find the best homography estimate. The labeler uses our developed GUI to match four background keypoints in two consecutive frames and fine tune the matches to visually minimize the background stitching error. The labeler also specifies a foreground mask, represent-

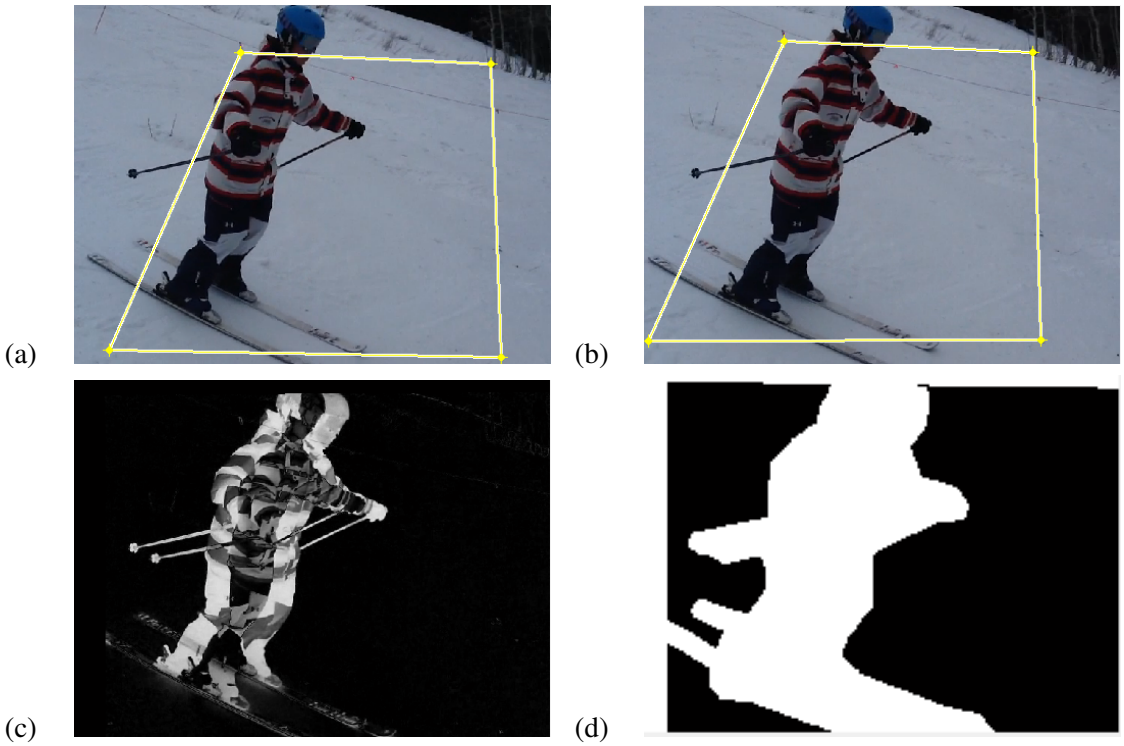


Figure 3.3: (a-b) Two consecutive frames and the matched quadruplet by the labeler, (c) the absolute difference of two frames matched via the quadruplet in (a,b), (d) manually labeled foreground mask.

ing the region resulted from foreground movement. Fig. 3.3 shows two consecutive video frames and the manually matched quadruplets. From the manually labeled sequences, we find the empirical distribution of E, t_x, t_y, Δs, $\Delta\alpha$, and H. As shown in Fig. 3.4, E, Δs, $\Delta\alpha$, and H are well approximated by a normal distribution. For H distribution, 10% percentile ($p_{H,0.9}$), reflecting the value that 90% of observed point entropies are larger than, is also shown. For t_x and t_y, Laplacian distribution is more appropriate. By plugging the probability models to Eqn. 3.3 and ignoring the constants, the final cost function is,

$$f(\theta_t) = \frac{\sum_{i=1}^{N_{in}} \frac{d_i}{b} + \sum_{i=1}^{N_{out}} \ln(S^2)}{N_{in}+N_{out}} + \frac{(E(\mathbf{I}_{t-1},\mathbf{I}_{t|\theta},\mathbf{M}) - \mu_E)^2}{2\sigma_E^2} + \lfloor \frac{(\Delta s - \mu_{\Delta s})^2}{2\sigma_{\Delta s}^2} \rfloor_T + \lfloor \frac{(\Delta\alpha - \mu_{\Delta\alpha})^2}{2\sigma_{\Delta\alpha}^2} \rfloor_T + \lfloor \frac{|\Delta t_x - \mu_{\Delta t_x}|}{b_{t_x}} \rfloor_T + \lfloor \frac{|\Delta t_y - \mu_{\Delta t_y}|}{b_{t_y}} \rfloor_T, \quad (3.10)$$

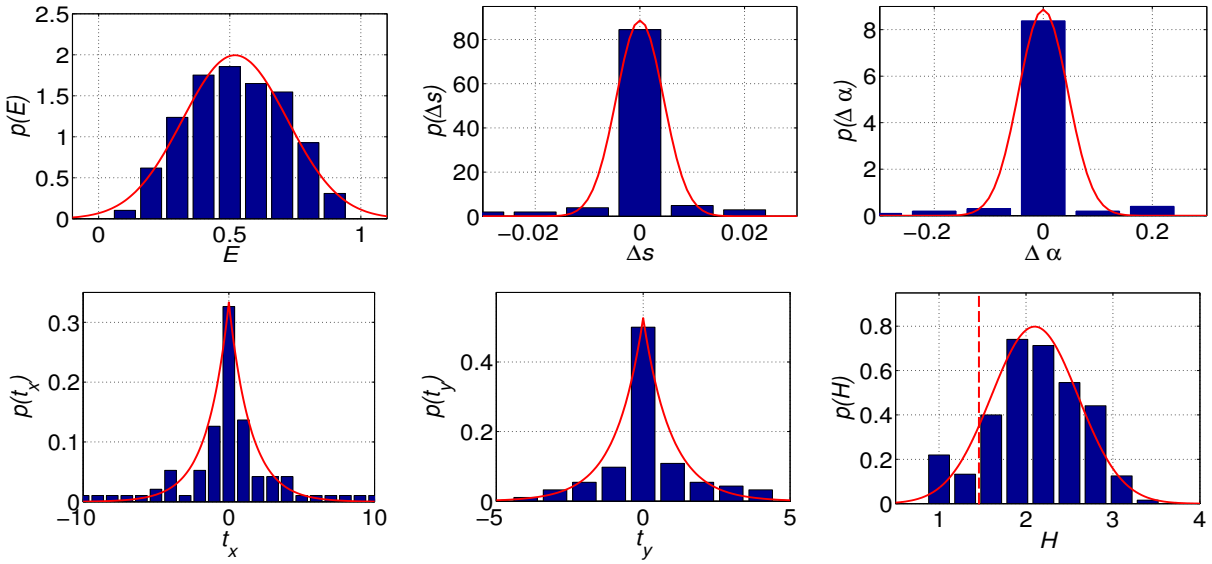


Figure 3.4: Empirical and fitted distributions for (a) $E \sim N(0.52, 0.04)$, (b) $\Delta s \sim N(0, 2 \times 10^{-5})$, (c) $\Delta\alpha \sim N(0, 2 \times 10^{-3})$, (d) $t_x \sim Laplace(0, 1.50)$, (e) $t_y \sim Laplace(0, 0.95)$, and (f) $H \sim N(2.1, 0.25)$.

where N_{out} is number of outliers and $\lfloor x \rfloor_T = \min(x, T)$ restricts the impact of prior information. Since keypoint matching error is dependent on the number of keypoints, we normalize it with the total number of keypoints. The homography θ_t is estimated by

$$\theta_t^* = \arg\min(f(\theta_t)). \tag{3.11}$$

3.3 Experimental Results

This section presents the experimental results of RGMC, and its comparison with our implementations of the RANSAC variation called MLESAC [107] and the HEASK method [124].

3.3.1 Dataset

We select 50 videos from SVW dataset [89], where 24 videos are used for model learning in Sec. 3.2.2, and the rest for testing. SVW contains videos of amateurs practicing a sport, shot

Figure 3.5: Sample frames of the test videos in (a) SVW, (b) HMDB51, and (c) Holleywood2 datasets.

using smartphone by ordinary people. Thus, highly unconstrained SVW is an excellent example of user-generated videos with predominant foreground of humans. We also use 10 videos from Holleywood2 [72] and 15 videos from HMDB51 [49] datasets[1]. In total, 51 videos are used for quantitative evaluation with sample frames shown in Fig. 3.5. [2]

3.3.2 Parameters

In all the experiments, we have the same fixed parameter setting, i.e., $\tau = 0.5$, $C = 50$, $T_C = 50$, $T_M = 100$, $T_E = 2$, $K = 10$, $\eta = 1.5$, $\alpha = 0.5$, $\lambda = 70\%$, and $T = 100$. Our experiments show that RGMC is robust to variation of parameters. The most important parameter is K. Large values of K increase the computational cost. On the other hand, K should be large enough so that foreground, background, and erroneous matches are mapped to different clusters. As a trade-off, we use $K = 10$.

[1]For these two datasets, videos are temporally trimmed around the signature motion in the video, practically disabling effect of our motion history module. In HMDB51, similar to many existing datasets such as UCF101 [100], the video resolution is only 320×240, thus GMC suffers from both video content and the low resolution.

[2]In HMDB51 [49] and UCF101 [100] datasets, only very low resolution videos ($320 * 240$) are *publicly* available for which GMC basically suffers considerably from the resolution and in UCF-Sports [102], camera is static. Thus, we limit our qualitative evaluation to SVW and available motion-compensated videos in HMDB51 dataset.

Algorithm	Ground Truth	MLESAC		HEASK		RGMC				
Setting	–	DT	LT	DT	LT	$(20,50)$	$(\mathbf{50},\mathbf{100})$	$(100,200)$	D-M	D-E
BRE ($\times 10^{-3}$)	**7.59**	**15.65**	18.59	17.33	**14.24**	11.77	10.11	**10.02**	11.60	11.25

Table 3.1: Impact of different settings on average BRE for each algorithm. DT and LT denote default ($\tau_s = 1000$) and lowered ($\tau_s = 100$) detection threshold in SURF algorithm, respectively. For RGMC $\tau_s = 100$ is used and 3 different setting of (T_C, T_M) are reported. D-M and D-E denote default setting of $(T_C, T_M) = (50, 100)$ with motion history and error handling turned off, respectively.

3.3.3 Evaluation metric

For accuracy evaluation, we have manually matched a quadruplet of keypoints and found the ground truth homography θ_0 for a total of 350 pairs of consecutive frames in challenging periods in 51 test videos. The same GUI described in Sec. 3.2.2 is used to obtain θ_0 and the foreground mask. We denote the intersection of the complement of this mask, i.e., the background mask, and the region covered by $\mathbf{I}_{t|\theta_0}$, as $\mathbf{B}$. We quantify the consistency of frames $\mathbf{I}_t$ and $\mathbf{I}_{t-1|\theta}$ (grayscale frames with pixels ranging between 0 and 1) using the background region error (BRE), $\varepsilon = \frac{1}{\|\mathbf{B}\|_1} \| |(\mathbf{I}_{t-1} - \mathbf{I}_{t|\theta_t})| \bigodot \mathbf{B} \|_1$.

3.3.4 Accuracy assessment

Table 4.1 represents the average BRE on test videos for different algorithms. Due to random nature of algorithms, we repeat each experiment 5 times and report the average performance. To ensure that comparisons are fair, we decrease the keypoint detection thresholds also for baseline methods. HEASK has better performance with lowered threshold and thus we use this setting for the experiments. We also report results for different iteration numbers T_C and T_M for RGMC and as a trade-off between accuracy and efficiency, select $(T_C, T_M) = (50, 100)$ as default values for RGMC. In addition, we turn off the modules of *Motion History* and *Error Handling* in RGMC alternatively, to verify that their existence is helpful. Fig. 3.6 shows two consecutive frames of

three sample videos matched by different algorithms, along with the ground truth matching. As shown, RGMC produces very accurate background matching. Fig. 3.7 represents the average per-video BRE, sorted by the BRE of ground truth matching. As shown, RGMC performance is very robust and in most videos RGMC matching error is very close to the ground truth value. Finally, Fig. 3.8 compares stitching results on a sample video using different algorithms. It is worth noting that since a cascade of homographies are used for GMC and stitching of video frames, propagation of errors of matching consecutive frames, gives rise to inaccuracy as the length of the input video increases. Also, coexistence of textureless background and large foreground (in terms of the total number of pixels covered by the foreground), may cause failure in the RGMC algorithm, especially if the foreground motion exists starting the initial frames.

3.3.5 Computational cost

For the comparison with baseline methods, we test Matlab implementations of algorithms on a PC with Intel i5-3470@2GHz CPU. The average time for matching frame pair of size $720 \times 1,280$ (480×854) by MLESAC, HEASK, and RGMC is 2.0 (0.3), 53.1 (21.3), and 4.3 (2.3) seconds, respectively. We also have a C++ implementation of RGMC using the OpenCV libraries, which takes 1.4 (0.7) seconds for matching frame pair of size $720 \times 1,280$ (480×854)[3].

3.3.6 Qualitative evaluation

In addition to the aforementioned quantitative study, we also perform the qualitative evaluation on *u*nlabeled videos to demonstrate the severity of the predominant foreground issue in real-world videos, and the superiority of RGMC on a large scale dataset. For each video, we run a GMC algorithm, visually observe the motion-compensated videos, and claim a *f*ailure if an instable background is observed (e.g., Fig. 3.8 (a,b)). We observe a failure rate of 32% by the MLESAC method

[3]Source code is available at http://www.cse.msu.edu/∼liuxm/RGMC

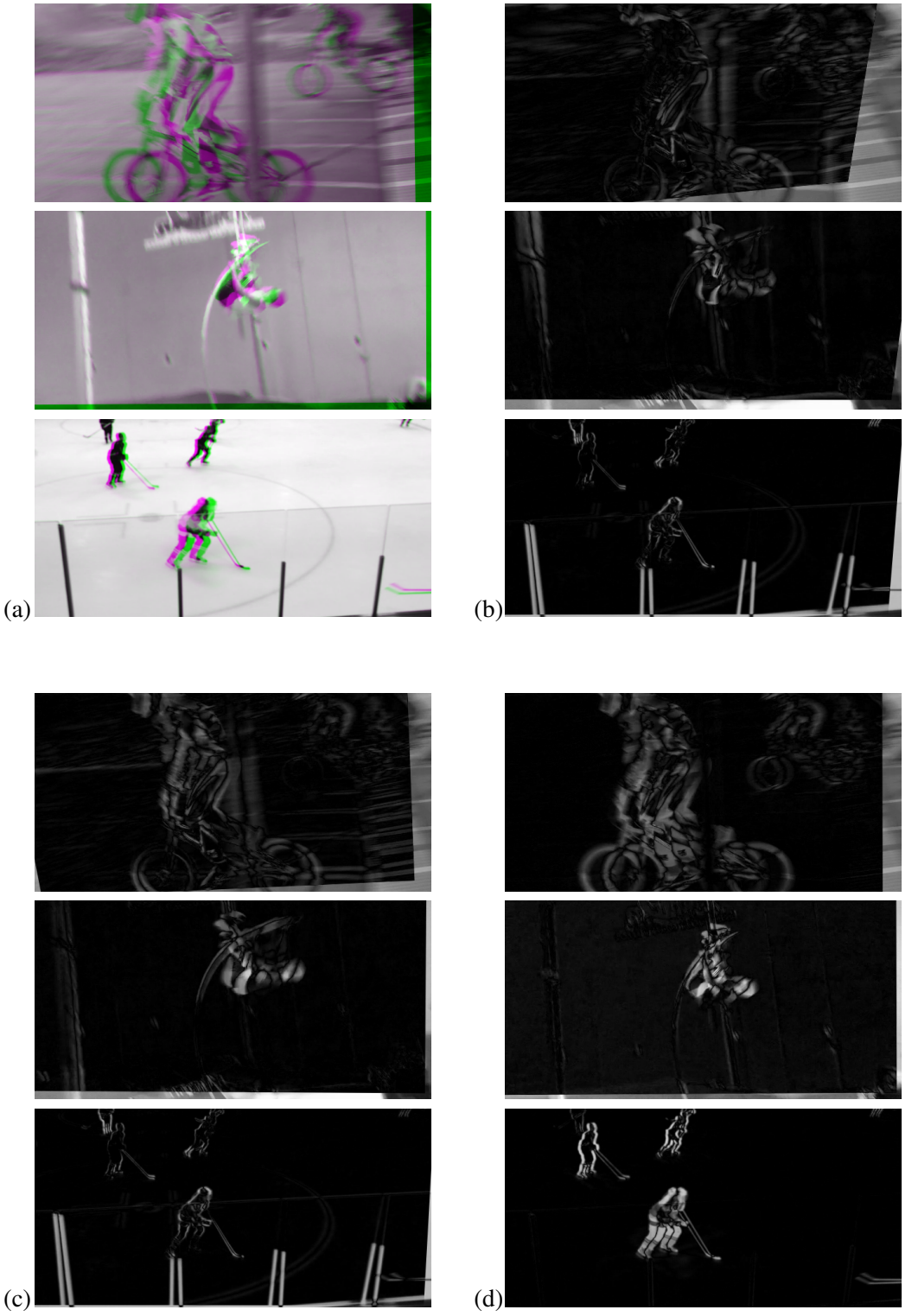


Figure 3.6: Each row shows GMC results of two consecutive frames from video ID S17, S19, and S9 by (a) manual labeling, (b) MLESAC, (c) HEASK, and (d) RGMC. In (a), colorful pixels show the pixels that are different between overlaid frames. In (b-d), the pixel brightness indicates the difference.

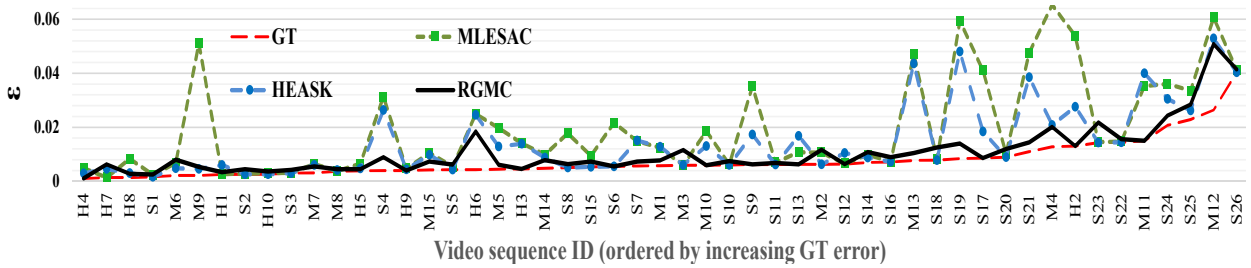
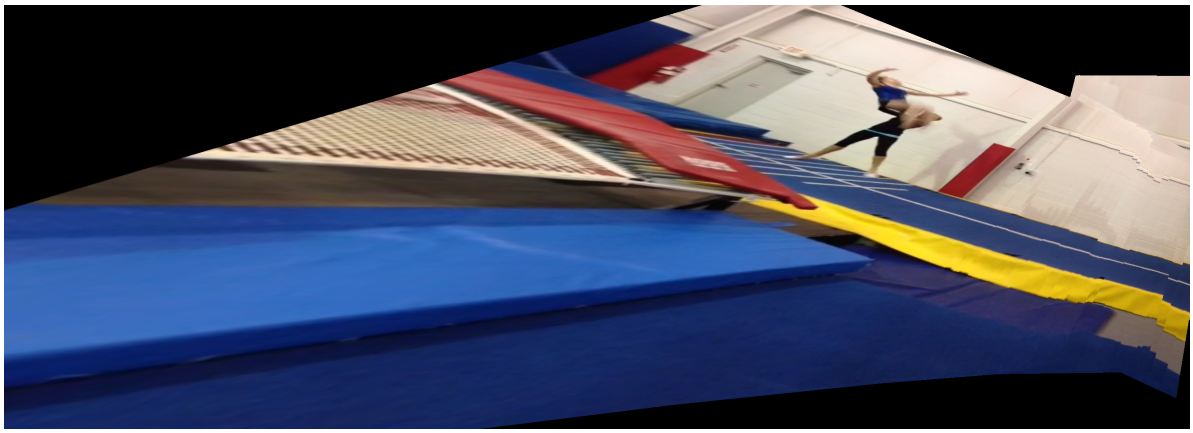
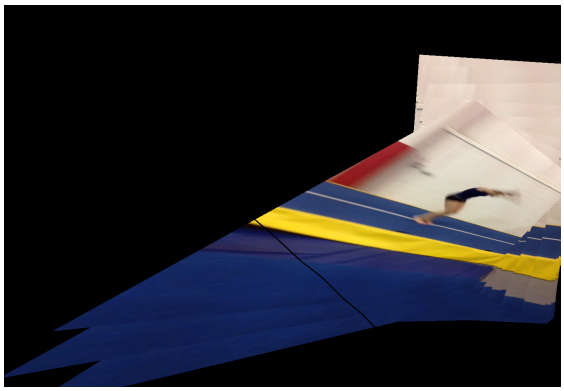


Figure 3.7: Per-video BRE using the optimal setting for each algorithm compared with ground truth (GT) matching BRE.

(a)



(b)



(c)

Figure 3.8: A 40-frame sequence of gymnastics backflips in textureless background stitched using (a) MLESAC, (b) HEASK, and (c) RGMC. Consistency of the background shows the superiority of RGMC. For HEASK, stitching up to frame #10 is shown, after which the stitching drastically fails.

among 225 videos from three categories of cartwheel, dive and dribble in HMDB51 dataset. Further, a 35% failure rate by MLESAC is observed from 500 videos of SVW dataset; in contrast on the same data our RGMC has merely a 5% failure rate.

3.4 Conclusions

We presented a robust global motion compensation (RGMC) algorithm that delivers reliable results in the presence of predominant foreground and textureless or blurry background, enabling its application to real-world unconstrained videos. By foreground suppression, RGMC is able to tolerate large foreground and occlusion. Also, the proposed method successfully handles keypoint matching with a very low matching threshold, required for GMC in low texture background. This is achieved by clustering motion vectors, and analyzing each cluster to identify matches pertaining to the background. A novel homography verification model is proposed to support the RGMC. Extensive experiments and comparison with manually matched ground truth and baseline methods demonstrate the superiority of RGMC.

Chapter 4

Temporally Robust Global Motion Compensation

As discussed in Chapter 3, Global motion compensation (GMC) removes the impact of *intentional* and *unwanted* camera motion in the video, transforming the video to have *static* background with the only motion coming from foreground objects. Video stabilization is a closely related problem where *unwanted* camera motion, such as vibration, is removed, leaving a smooth camera motion in the output video. It is important to note that the final product of GMC is a video with *static* background *throughout the entire video*. This sets a high bar on accuracy requirement for estimation of transformations to the global coordinate, despite foreground motion and appearance ambiguities. GMC can be re-purposed for video stabilization (VS) and mosaicing, but not vice versa - given the accuracy requirement. GMC is an essential module for processing videos from *n*on-stationary cameras, which are abundant due to emerging mobile sensors, e.g., wearable cameras, smartphones, and camera drones. First, the resultant *motion panorama* [8], as if virtually generated by a static camera, is by itself appealing for visual perception. More importantly, many vision tasks benefit from GMC. For instance, dense trajectories [114] are shown to be superior when camera motion is compensated [117]. Otherwise, camera motion interferes with human motion, rendering the analysis problem very challenging. Accurate and consistent GMC allows reconstruction of a “stitched" background [74], and subsequently segmentation of foreground [103, 113]. This helps

Figure 4.1: Schematic diagrams of proposed TRGMC and existing sequential GMC algorithms, and resultant motion panorama for a video shot by panning the camera up and down. Background continuity breaks easily in the case of the sequential GMC [88].

multi-object tracking by mitigating the unconstrained problem of tracking multiple in-the-wild objects, to tracking objects with a static background [99].

In existing GMC work [9, 24, 88], frames are transformed to a global motion-compensated coordinate (GMCC), by *sequentially* processing input frames. For a pair of consecutive frames, the mapping transformation is estimated, and by accumulating the transformations, a *composite* global transformation of each frame to the GMCC is obtained. However, the sequential processing scheme causes frequent GMC failures for multiple reasons: 1) Sequential GMC is only as strong as the *w*eakest pair of consecutive frames. A single frame with high blur or dominant foreground motion can cause the rest of the video to fail. 2) Generally, multiple planes exist in the scene. The common assumption of a single homography will accumulate residual errors into remarkable errors. 3) Even if the error of consecutive frames is in a sub-pixel scale, due to the *multiplication* of several homography matrices, the error can be significant over time [74]. These problems are especially severe when processing long videos and/or the camera motion becomes more complicated. E.g.,

when the camera pans to left and right repeatedly, or severe camera vibration exists, the GMC error is obvious by exhibiting discontinuity on the background (see Fig. 4.1 for an example). Although RGMC algorithm discussed in Chapter 3 improves GMC robustness and considerably decreases rate of drastic failures, still accumulation of error degrades RGMC performance. This degradation is more obvious when video length increases and camera motion is more complicated.

To address the issues of sequential GMC, we propose a temporally robust global motion compensation (TRGMC) algorithm which by *joint* alignment of input frames, estimates accurate and temporally consistent transformations to GMCC. The result can be rendered as a motion panorama that maintains perceptual realism despite complicated camera motion (Fig. 4.1). TRGMC densely connects pairs of frames, by matching local keypoints. Joint alignment (a.k.a. congealing) of these frames is formulated as an optimization problem where the transformation of each frame is updated iteratively, such that for each *link* interconnecting a keypoint pair, the spatial coordinates of two end points are identical. This novel *k*eypoint-based congealing, built upon succinct keypoint coordinates instead of high-dimensional appearance features, is the core of TRGMC. Joint alignment not only leads to the temporal consistency of GMC, but also improves GMC stability by using redundancy of the information. The improved stability is crucial for GMC, especially in the presence of considerable foreground motion, motion blur, non-rigid motion like water, or low-texture background. The joint alignment scheme also provides capabilities such as coarse-to-fine alignment, i.e., alignment of the keyframes followed by non-keyframes, and appropriate weighting of keypoints matches, which cannot be naturally integrated in sequential GMC. Our quantitative experiments reveal that TRGMC pushes the alignment error close to human performance.

In summary, this chapter makes the following contributions:

- An algorithm for *joint* alignment of video frames is proposed to produce a globally motion compensated video where, despite the complicated camera movement and considerable

foreground motion, the background appears to be static over the progression of time.

- A keypoint-based congealing algorithm aligns the spatial coordinates of keypoints for an image stack. It extends congealing applications from spatially cropped objects (faces and letters) to complex motion-rich video frames.

- The capabilities and applications of TGRMC are demonstrated. Our collected video dataset, manual labels, and the code will be publicly available.

4.1 Previous Work

TRGMC is related to many techniques in different aspects. We first review them and then compare our work with existing GMC algorithms.

Firstly, homography estimation from keypoint matches is crucial to many vision tasks, e.g., image stitching, registration, and GMC. A main challenge of homography estimation from keypoint matches is the false matches due to appearance ambiguities. Robust methods are proposed to handle the outliers, such as RANSAC [35] and its variants [17, 106, 107]. Some methods also directly reject false matches [57, 70]. The hybrid methods [88, 124] combines appearance similarity and keypoint matches in a probabilistic framework. *A*ll methods estimate a homography for a frame pair. In contrast, in TRGMC, instead of direct calculation of homography transformation for each pair of frame, we jointly optimize the set of homographies which map the set of input frames into a global coordinate, such that the keypoints over a wide range of temporal distance are aligned well. Thus, TRGMC leverages the redundant background matches over time to better handle outliers.

Image stitching (IS) and panoramic image mosaicing share similarity with GMC. IS aims to minimize the distortions and ghosting artifact in the overlap region. Many works utilize multiple homographies, instead of a *single* homography, due to existence of multiple scene planes [36,

69, 104, 105, 109, 133]. Some recent works focus also on the parallax issue, by using a hybrid model that uses homograhy for non-overlapping parallax-free regions and allow some local non-projective deviation to account for parallax and avoid stitching artifacts [59, 60, 127]. Li *et al*let@tokeneonedot [58] generate panoramas from motion-blurred videos. In these works, input images have much less overlap than GMC. On the other hand, video mosaicing takes in a video which raster scans a wide angle *static* scene, and produces a single *static* panoramic image [90, 92, 97]. When the camera path forms a 2D scan [92] or a 360° rotation [90], global refinement is performed via bundle adjustment (BA) [108], which ensures an artifact-free panoramic image, assuming a static scene. Although a byproduct of TRGMC is a similar static reconstruction of the scene, TRGMC focuses on efficient generation of an appealing video, where background consistently appears static for visual perception (in contrast to an image), for a *highly dynamic* scene. The important feature of such a video is that the only apparent motion in the video will rise from foreground motion. While one may use BA to estimate camera pose and then transformation between frames, our experiments reveal that BA is not reliable for videos with foreground motion and is less efficient than TRGMC. Further, BA estimates 3D location of keypoints while TRGMC needs 2D registration. Thus, by using BA, a harder problem needs to be solved which is unnecessary for the purpose of global motion compensation. Hence, image/video mosaicing and GMC have different application scenarios and challenges.

Another related topic is the panoramic video [31, 43, 44, 77, 128]. For instance, Perazzi *et al*let@tokeneonedot [77] create a panoramic video from an array of stationary cameras by generalizing parallax-tolerant image stitching to video stitching. The fast video stitching in [128] can handle proper stitching of objects at varying depths. Jiang and Gu [44] propose an algorithm for stitching multiple video streams into a single panoramic video with spatial-temporal content-preserving warping. In this work, for alignment of video frames, a spatial-temporal local warping

is proposed, which locally aligns frames from different videos while maintaining the temporal consistency. While these works focus on stitching *multiple* synchronized videos, GMC creates a motion panorama from a *single non-stationary* camera. Unlike GMC, video panoramas do not require the resultant video to have a stationary background.

Video stabilization (VS) is a closely related but different problem. TRGMC can be re-purposed for VS, but not vice versa, due to the accuracy requirement. Given the accurate mapping to a global coordinate using TRGMC, VS would mainly amount to cropping out a smooth sequence of frames and handling rendering issues such as parallax. Among different categories of VS, 2D VS methods calculate consecutive warping between the frames and have similarities with *sequential* GMC, but any estimation error will not cause severe degradation in VS as long as it is smoothed. While TRGMC targets *long-term staticness of the background*, VS mainly cares about *smoothing* of camera motion, not *removing* it. In other words, TRGMC imposes a stronger constraint on the result, which is background staticness by complete camera motion removal in comparison to VS which deals with camera motion smoothing. This strict requirement differentiates TRGMC also from Re-Cinematography [38]. Also, large occlusion by the foreground may result in VS failure, however TRGMC handles this challenge by utilizing redundancy of background information in the joint alignment scheme.

Congealing aims to jointly align a stack of images from one object class, e.g., faces and letters [53, 64]. Congealing iteratively updates the transformations of all images such that the entropy [53] or Sum of Squared Differences (SSD) [18] of the images, is minimized. However, despite many extensions of congealing [19, 41, 50, 68, 96], almost all prior work define the energy based on the *appearance features* of two images. Since congealing is based on image-based processing, it requires moderate initial alignment and is sensitive to intra-class variation and background clutter [50]. In [41], by incorporating deep learning into the congealing alignment frame-

work, a combination of unsupervised joint alignment with unsupervised feature learning is proposed. Through deep learning, authors obtain features that can represent the image at differing resolutions based on network depth, and that are tuned to the statistics of the specific data being aligned. In [50], a heuristic local feature based algorithm is proposed to rigidly align object class images to a seed image. Best matching local features are selected as object landmark. To overcome the problem of false matches, iteratively a minimal subset of matches are selected, homography is estimated, and image points are transformed to the seed coordinate. Using a spatial scoring algorithm, scores of the features matching are accumulated within a preset distance limit, resulting to refined landmarks. Finally, the other images are aligned to the seed using only the best landmarks. Note that [50] uses a heuristic local feature based algorithm to rigidly align object class images. In contrast we formulate the joint alignment of keypoints as an optimization problem and solve it in a principal way. Cox *et al*let@tokeneonedot [18] employ a sum of squared differences (SSD) cost solved by Gauss-Newton gradient decent. Unlike entropy-based congealing, the details of each image in the stack are used for alignment, rather than relying on the average image of the stack. Our experiments on GMC show that appearance-based congealing is inefficient and sensitive to initialization and foreground motion. Therefore, we propose a novel keypoint-based congealing algorithm minimizing the SSD of corresponding *keypoint coordinates*, instead of appearance features, to gain considerable efficiency enhancement and robustness to initialization.Further, most prior works apply to a spatially cropped object such as faces, while we deal with complex video frames with dynamic foreground and moving background, at a higher spatial-temporal resolution.

There are a few existing sequential GMC works, where the main problem is to accurately estimate a homography transformation between consecutive frames, given challenges such as appearance ambiguities, multi-plane scene, and dominant foreground [8,24,88]. Bartoli *et al*let@tokeneonedot [9] first estimate an approximate 4-degree-of-freedom homography, and then refine it. Sakamoto *et*

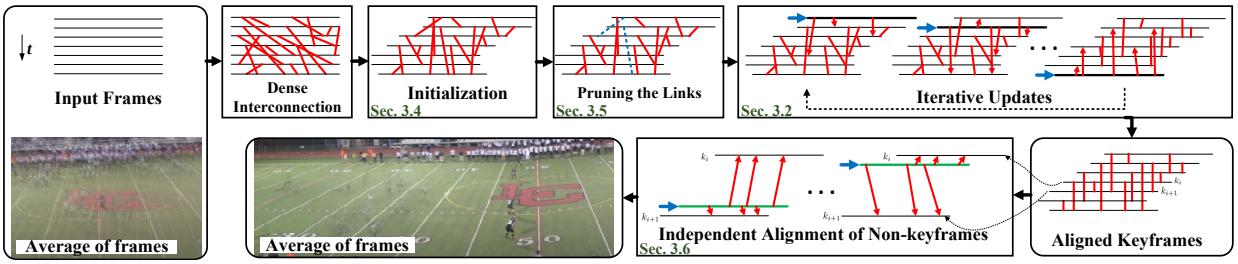


Figure 4.2: Flowchart of the TRGMC algorithm.

*al*let@tokeneonedot [90] generate a 360° panorama from an image sequence. Assuming that multiplication of all consecutive homographies results in the *identity* mapping, and homography has only 5 degrees of freedom, the camera rotation matrix has 3 degrees of freedom, to which are added the focal lengths before and after the camera rotation, all the homographies are optimized jointly to prevent error accumulation. In contrast, TRGMC employs an 8-degree-of-freedom homography. Although using homography in the case of considerable camera translation and large depth variation results in parallax artifacts, using a higher degrees-of-freedom homography than prior works allows TRGMC to better handle camera panning, zooming, and translation. Safdarnejad *et al*let@tokeneonedot [88] incorporate edge matching into a probabilistic framework that scores candidate homographies. Although [24, 88] improve the robustness to foreground, error accumulation and failure in a single frame pair still deteriorate the overall performance. Thus, TRGMC targets robustness of the GMC in terms of both the presence of foreground and long-term consistency by joint alignment of frames.

4.2 Proposed TRGMC Algorithm

The core of TRGMC is the novel keypoint-based congealing algorithm. Our method relies on densely interconnecting the input frames, regardless of their temporal offset, by matching the detected SURF [10] keypoints at each frame. We refer to these connections, shown in Fig. 4.2, as *links*. Frames are initialized to their approximate spatial location by only 2D translation (Sec. 4.2.4).

We rectify the keypoints such that majority of the links have end points on the background region. Then the congealing applies appropriate transformation to each frame and the links connected to it, such that the spatial coordinates of the end-points of each link are as similar as possible. In Fig. 4.2, this translates to having the links as parallel to the $t-$axis as possible.

For efficiency and robustness, TRGMC processes an input video in two stages. Stage one selects and jointly aligns a set of keyframes. The keyframes are frozen, and then stage two aligns each remaining frame to its two encompassing keyframes. The remainder of this section presents the details of the algorithm.

4.2.1 Formulation of keypoint-based congealing

Given a stack of N frames $\{\mathbf{I}^{(i)}\}$, with indices $i \in \mathbb{K} = \{k_1, ..., k_N\}$, the keypoint-based congealing is formulated as an optimization problem,

$$\min_{\{\theta_i\}} \varepsilon = \sum_{i \in \mathbb{K}} [\mathbf{e}_i(\theta_i)]^\intercal \Omega^{(i)} [\mathbf{e}_i(\theta_i)], \tag{4.1}$$

where θ_i is the transformation parameter from frame i to GMCC, $\mathbf{e}_i(\theta_i)$ collects the pair-wise alignment errors of frame i relative to all the other frames in the stack, and $\Omega^{(i)}$ is a weight matrix.

We define the alignment error of frame i as the SSD between the spatial coordinates of the endpoints of all links connecting frame i to the other frames, instead of the SSD of appearance [18]. Specifically, as shown in Fig. 4.3, we denote coordinates of the start and the end point of each link k connecting frame i to the frame $d_k^{(i)} \in \mathbb{K} \backslash \{i\}$ as $(x_k^{(i)}, y_k^{(i)})$ and $(u_k^{(i)}, v_k^{(i)})$, respectively. For simplicity, we omit the frame index i in θ_i. Thus, the error $\mathbf{e}_i(\theta)$ is defined as,

$$\mathbf{e}_i(\theta) = [\Delta X_i(\theta)^\intercal, \Delta Y_i(\theta)^\intercal]^\intercal, \tag{4.2}$$

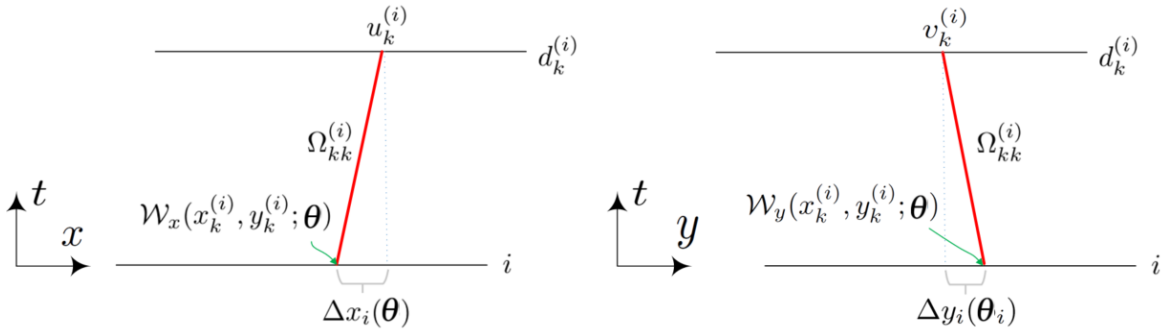


Figure 4.3: The notation used in TRGMC.

where

$$\Delta X_i(\theta) = \tilde{\mathbf{w}}_i^{(x)} - \mathbf{u}^{(i)}, \quad \Delta Y_i(\theta) = \tilde{\mathbf{w}}_i^{(y)} - \mathbf{v}^{(i)}, \tag{4.3}$$

are the errors in $x-$ and $y-$ axes. The vectors $\tilde{\mathbf{w}}_i^{(x)} = [\mathscr{W}_x(x_k^{(i)}, y_k^{(i)}; \theta)]$ and $\tilde{\mathbf{w}}_i^{(y)} = [\mathscr{W}_y(x_k^{(i)}, y_k^{(i)}; \theta)]$ denote the x and $y-$ coordinates of $(x_k^{(i)}, y_k^{(i)})$ warped by the parameter θ, respectively. The vectors $\mathbf{u}^{(i)} = [u_k^{(i)}]$ and $\mathbf{v}^{(i)} = [v_k^{(i)}]$ denote the coordinates of the end points. Similarly, the vectors $\mathbf{x}^{(i)} = [x_k^{(i)}]$ and $\mathbf{y}^{(i)} = [y_k^{(i)}]$ denote the coordinates of the start points. If N_i links emanate from frame i, $\mathbf{e}_i$ is a $2N_i-$dim vector. $\Omega^{(i)}$ is a diagonal matrix of size $2N_i \times 2N_i$ which assigns a weight to each element in $\mathbf{e}_i$. The parameter θ has 2, 6, or 8 elements for the cases of 2D translation, affine transformation, or homography, respectively. In this chapter, we focus on homography transformation which is a projective warp model, parameterized as,

$$\begin{bmatrix} \mathscr{W}_x(x_k^{(i)}, y_k^{(i)}; \theta) \\ \mathscr{W}_y(x_k^{(i)}, y_k^{(i)}; \theta) \\ 1 \end{bmatrix} = \overbrace{\begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{bmatrix}}^{\theta} \begin{bmatrix} x_k^{(i)} \\ y_k^{(i)} \\ 1 \end{bmatrix}. \tag{4.4}$$

Although the homography model assumes the planar scene and this assumption may be violated in real world [127], we identify the problem of temporal robustness to be more fundamental for GMC than the inaccuracies due to a *single* homography. Also, videos for GMC are generally

swiped through the scene with high overlap, thus the discontinuity resulted from this assumption is minor.

4.2.2 Optimization solution

Equation 6.1 is a non-linear optimization problem and difficult to minimize. Following [18], we linearize this equation by taking the first-order Taylor expansion around θ. Starting from an initial θ, the goal is to estimate $\Delta\theta$ by,

$$\arg\min_{\Delta\theta} [\mathbf{e}_i(\theta) + \frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\Delta\theta]^{\mathsf{T}} \Omega^{(i)} [\mathbf{e}_i(\theta) + \frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\Delta\theta] + \gamma \Delta\theta^{\mathsf{T}} \mathscr{I} \Delta\theta, \tag{4.5}$$

where $\Delta\theta^{\mathsf{T}} \mathscr{I} \Delta\theta$ is a regularization term, with a positive constant γ setting the trade-off. We observe that without this regularization, parameter estimation may lead to distortion of the frames. The indicator matrix $\mathscr{I}$ is a diagonal matrix specifying which elements of $\Delta\theta$ need a constraint. We use $\mathscr{I} = diag([1,1,0,1,1,0,1,1])$ to specify that there is no constraint on the translation parameters of the homography, but the rest of parameters should remain small.

By setting the first-order derivative of Eqn. 6.4 to zero, the solution for $\Delta\theta$ is,

$$\Delta\theta = \mathbf{H}_R^{-1} \frac{\partial \mathbf{e}_i(\theta)^{\mathsf{T}}}{\partial \theta} \Omega^{(i)} \mathbf{e}_i(\theta), \tag{4.6}$$

$$\mathbf{H}_R = \frac{\partial \mathbf{e}_i(\theta)^{\mathsf{T}}}{\partial \theta}\, \Omega^{(i)} \frac{\partial \mathbf{e}_i(\theta)}{\partial \theta} + \gamma \mathscr{I}. \tag{4.7}$$

Using the chain rule, we have $\frac{\partial \mathbf{e}_i(\theta)}{\partial \theta} = \frac{\partial \mathbf{e}_i(\theta)}{\partial \mathscr{W}} \frac{\partial \mathscr{W}}{\partial \theta}$. Knowing that the mapping has two components as $\mathscr{W} = (\mathscr{W}_x, \mathscr{W}_y)$, and the first half of $\mathbf{e}_i$ only contains x components and the rest only y

components, we have,

$$\frac{\partial \mathbf{e}_i(\theta)}{\partial \mathscr{W}} = \begin{bmatrix} \mathbf{1}_{N_i} & \mathbf{0}_{N_i} \\ \mathbf{0}_{N_i} & \mathbf{1}_{N_i} \end{bmatrix}, \tag{4.8}$$

where $\mathbf{1}_{N_i}$ and $\mathbf{0}_{N_i}$ are N_i-dim vectors with all element being 1 and 0, respectively. For homography transformation, $\frac{\partial \mathscr{W}}{\partial \theta} = \frac{\partial(\mathscr{W}_x, \mathscr{W}_y)}{\partial(p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8)}$ is given by,

$$\frac{\partial \mathscr{W}}{\partial \theta} = \begin{bmatrix} \tilde{\mathbf{w}}_i^{(x)} & \tilde{\mathbf{w}}_i^{(y)} & \mathbf{1}_{N_i} & \mathbf{0}_{N_i} & \mathbf{0}_{N_i} & \mathbf{0}_{N_i} & -\mathbf{u}^{(i)}\tilde{\mathbf{w}}_i^{(x)} & -\mathbf{u}^{(i)}\tilde{\mathbf{w}}_i^{(y)} \\ \mathbf{0}_{N_i} & \mathbf{0}_{N_i} & \mathbf{0}_{N_i} & \tilde{\mathbf{w}}_i^{(x)} & \tilde{\mathbf{w}}_i^{(y)} & \mathbf{1}_{N_i} & -\mathbf{v}^{(i)}\tilde{\mathbf{w}}_i^{(x)} & -\mathbf{v}^{(i)}\tilde{\mathbf{w}}_i^{(y)} \end{bmatrix}. \tag{4.9}$$

At each iteration, and for each frame i, $\Delta\theta$ is calculated and the start points of all the links emanating from frame i are updated accordingly. Similarly, for all links with end points on frame i, the end point coordinates are updated. [1]

We use the SURF [10] algorithm for keypoint detection with a low detection threshold, $\tau_s = 200$, to ensure sufficient keypoints are detected even for low-texture backgrounds. We use the nearest-neighbor ratio method [65] to match keypoints and form links between each pair of keyframes.

Keyframe selection We select keyframes at a constant step of Δf, i.e., from every Δf frames, only one is selected. Based on the experimental results, as a trade-off between accuracy and efficiency, we use $\Delta f = 10$ in TRGMC.

4.2.3 Weight assignment

We have defined all parameters in the problem formulation, except the weights of links, $\Omega^{(i)}$. We consider two factors in setting $\Omega^{(i)}$. Firstly, the keypoints detected at larger scales are more likely to be from background matches, since they cover coarser information and larger image patches. Thus, to be robust to foreground, the early iterations should emphasize links from larger-scale

[1] In algorithm implementation, it is important to store the original coordinates of the detected keypoints and apply the *composite* transformations accumulated in all the iterations to update the coordinates of the start and end points of the links. Otherwise, accumulation of numerical errors will harm the performance.

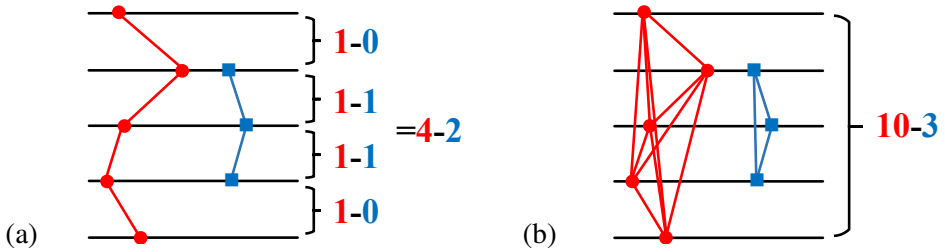


Figure 4.4: Comparison of the ratios of **background**-**foreground** matches for (a) sequential GMC and (b) TRGMC.

keypoints, which forms a coarse-to-fine alignment. We normalize the scales of all keypoints such that the maximum is 1, and denote the minimum of the normalized scales of the two keypoints comprising the link k as s_k. Then, $\Omega_{k,k}^{(i)}$ is set proportional to s_k.

Secondly, for each frame i, the links may be made either to all the previous frames, denoted as *backward* scheme, or both the previous and upcoming frames, denoted as *backward-forward* scheme. The former is for potential real time application, whereas the latter for offline video processing. These schemes are implemented by assigning different weights to backward and forward links,

$$\Omega_{k,k}^{(i)} = \begin{cases} (\beta.s_k)^{r^q}; \text{ if } d_k^{(i)} < i \quad \text{(Backward links)} \\ (\alpha.s_k)^{r^q}; \text{ if } d_k^{(i)} > i \quad \text{(Forward links)} \end{cases} \tag{4.10}$$

where $0 < \alpha, \beta < 1$, q is the iteration index, and $0 < r < 1$ is the rate of change of the weights. Note that the alignment errors in x and $y-$axes have the same weights, i.e., $\Omega_{k+N_i,k+N_i}^{(i)} = \Omega_{k,k}^{(i)}$. After a few iterations, the weights of all the links will be restored to 1. In the backward scheme, we set $\alpha = 0$.

4.2.4 Initialization

Initialization speeds up the alignment and decreases the false keypoint matches. The objective is to roughly place each frame at the appropriate coordinates in the GMCC. For initialization, we

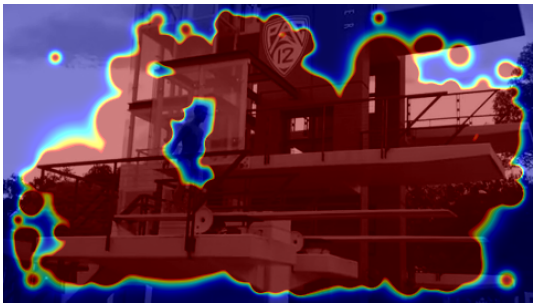
(a)  (b) 

Figure 4.5: (a) The input frame, (b) the reliability map, with the red color showing higher reliability.

align the frames based only on rough estimation of translation without considering rotation, skew, or scale. We use the average of the motion vectors in matching two consecutive frames as the translation. Using this simple initialization, even if the camera has in-plane rotation, estimated 2D translations are zero, which is indeed correct and does not cause any problem for TRGMC. Given the estimated translation, approximate overlap area of each pair of frames is calculated, and only the keypoints inside the overlap area are matched, reducing number of false matches due to appearance ambiguities.

4.2.5 Outlier handling

Links may become outliers for two reasons: (i) the keypoints reside on foreground objects not consistent with camera motion; (ii) false links between different physical locations are caused by the low detection threshold and similar appearances.

In order to prune the outliers, we assume that the motion vectors of background matches, i.e., background links, have consistent and smooth patterns, caused by camera motion such as pan, zoom, tilt, whereas, the outlier links will exhibit arbitrary pattern, inconsistent with the background pattern. Specifically, we use Ma *et al*let@tokeneonedot [70] method to prune outlier links by imposing a smoothness constraint on the motion vector field[2]. This method outperforms RANSAC

[2]We use the implementation provided by the authors and default parameters.

if the set of keypoint matches contains a large proportion of outliers. Since keyframes have larger relative time difference than consecutive frames, the foreground motion is accentuated and more distinguishable from camera motion. This helps with better pruning of the foreground links. At each stage that the keypoints from a pair of frames are matched to form the links, we perform the pruning.

Congealing of an image stack also increases the proportion of background matches over the outliers - another way to suppress outliers. The keypoints on background are more likely to form longer range matches than the foreground ones, due to non-rigid foreground motion. Hence, when $\binom{N}{2}$ combinatorial pairs of frames are interconnected, there are a lot more background matches (Fig. 4.4).

4.2.6 Alignment of non-keyframes

The keyframes alignment provides a set of temporally consistent motion compensated frames, which are the basis for aligning non-keyframes. We refer to keyframes and non-keyframes with superscripts i and j, respectively. For a non-keyframe j between the keyframes k_i and k_{i+1}, its alignment is a special case of Eqn. 6.1, with indices $\mathbb{K} = \{j\}$, and the destination of the links $d_k^{(j)} \in \{k_i, k_{i+1}\}$, i.e., only θ_j of frame j is updated while the keyframes remain fixed. Each non-keyframe between keyframes k_i and k_{i+1} is aligned independently.

However, given the small time offset between j and $d_k^{(j)}$, the observed foreground motion may be hard to discern. Also, frame j is linked only to two keyframes, thus there is no redundancy of background information to improve robustness to foreground motion. Therefore, we need a different means of outlier handling. We handle this issue by assigning higher weights to links that are more likely to be connected to the background.

For each keyframe i, we quantify how well the links emanating from frame i are aligned with

other keyframes. If the alignment error is small, i.e., $\varepsilon_k^{(i)} = \left|\mathscr{W}_x(x_k^{(i)}, y_k^{(i)}; \theta) - u_k^{(i)}\right| + \left|\mathscr{W}_y(x_k^{(i)}, y_k^{(i)}; \theta) - v_k^{(i)}\right| < \tau$, the link k is more likely on the background of frame i and thus, more reliable for aligning non-keyframes. We create a *reliability map* for each keyframe i, denoted as $\mathbf{R}^{(i)}$ (Fig. 4.5). For each link k with $\varepsilon_k^{(i)} < \tau$, a Gaussian function with $\mu_k = (x_k^{(i)}, y_k^{(i)})$ and $\sigma_k = cs_k$ is superposed on $\mathbf{R}^{(i)}$, where the constant c is 20. We define,

$$\mathbf{R}_{m,n}^{(i)} = \left\lceil \left\lfloor \sum_{k \in \mathbb{B}_i} e^{-\frac{\left(m - x_k^{(i)}\right)^2 + \left(n - y_k^{(i)}\right)^2}{2\sigma_k^2}} \right\rfloor_1 \right\rceil_\eta, \tag{4.11}$$

where $\mathbb{B}_i = \{k | \varepsilon_k^{(i)} < \tau\}$, $\eta > 0$ is a small constant (set to 0.1), $\lceil x \rceil_\eta = \max(x, \eta)$ and $\lfloor x \rfloor_1 = \min(1, \eta)$. Now, we assign the weight of the links connecting frame j to the keyframe $d_k^{(j)}$ at the coordinate $(u_k^{(j)}, v_k^{(j)})$, as the reliability map of the keyframe at the endpoint, $\Omega_{k,k}^{(j)} = \left(\mathbf{R}^{(a)}_{u_k^{(j)}, v_k^{(j)}}\right)^{r^q}$, where $a = d_k^{(j)}$.

We summarize the TRGMC algorithm in Algorithm 2.

4.3 Experimental Results

We now present qualitative and quantitative results of the TRGMC algorithm and discuss how different computer vision applications will benefit from TRGMC.

4.3.1 Baselines and details

We choose three sequential GMC algorithms as the baselines for comparison: MLESAC [107] and HEASK [124] both based on our own implementation, and RGMC [88] based on the authors' Matlab code available online. We implement TRGMC in Matlab, and will publish the code. Denoting the video frames of $w \times h$ pixels, we set the parameters as $\gamma = 0.1wh$, $T_1 = 300$, $\tau_1 = 5 \times 10^{-4}$, $T_2 = 50$, $\tau_2 = 10^{-4}$, $r = 0.7$, $\tau = 1$, $\Delta f = 10$, and $\beta = 1$. For the backward-forward scheme we set

Algorithm 2: TRGMC Algorithm

Data: A set of input frames $\{\mathbf{I}^{(m)}\}_{m=1}^{M}$
Result: A set of homography matrices $\{\theta_m\}_{m=1}^{M}$
/* Align keyframes (Sec. 4.2.2) */
Specify $\mathbb{K} = \{k_1, ..., k_N\}$ and initialize (Sec. 4.2.4);
Match keypoints of all frames $i \in \mathbb{K}$ densely;
Prune links (Sec. 4.2.5) and set weights (Eqn. 4.10);
Store links' start and end coordinates in $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{u}_i, \mathbf{v}_i)$;
repeat
    forall the $i \in \mathbb{K}$ **do**
        Compute $\Delta\theta_i$ (Eqn. 4.6), update θ_i, $\mathbf{x}_i$ and $\mathbf{y}_i$;
        Update $(\mathbf{u}_m, \mathbf{v}_m)$ according to θ_i for $m \in \mathbb{K} \backslash \{i\}$;
        Update weights (Eqn. 4.10);
    $q \leftarrow q + 1$;
until $q < T_1$ *or* $\left(\frac{1}{N}\sum_{i \in \mathbb{K}} ||\Delta\theta_i||^2 > \tau_1\right)$;
/* Align non-keyframes (Sec. 4.2.6) */
Compute reliability map $\mathbf{R}^{(i)}$ for $i \in \mathbb{K}$;
for $i = 1 : N - 1$ **do**
    forall the $j \in \{k_i + 1, ..., k_{i+1} - 1\}$ **do**
        Match keypoints in j with $d^{(j)} \in \{k_i, k_{i+1}\}$;
        Prune links (Sec. 4.2.5) and set weights $\Omega_{k,k}^{(j)}$;
        Store links' coordinates in $(\mathbf{x}_j, \mathbf{y}_j)$ and $(\mathbf{u}_j, \mathbf{v}_j)$;
        repeat
            Compute $\Delta\theta_j$ (Eqn. 4.6), update θ_j, $\mathbf{x}_j$ and $\mathbf{y}_j$;
            Update weights (Eqn. 4.10), $q \leftarrow q + 1$;
        until $q < T_2$ *or* $\left(||\Delta\theta_j||^2 > \tau_2\right)$;

$\alpha = 1$ and for the backward scheme $\alpha = 0$.

4.3.2 Datasets and metric

Given there is no public dataset for quantitative GMC evaluation, we form a dataset composed of 40 challenging videos from SVW [89] and 15 videos from UCF101 [100], termed "quantitative dataset". SVW is an extremely unconstrained dataset including videos of amateurs practicing sports, and is also captured by amateurs via smartphone. The min. and max. spatial size of videos are 240 and 480 pixels, respectively. The average, min., and max. length of the videos are 14, 3,

Algorithm	MLESAC	HEASK	RGMC	TRGMC		GT*
Setting	–	–	–	BF*	B*	–
Avg. BRE	0.116	0.110	0.097	**0.058**	0.060	0.038
Efficiency (s/f)	0.17	7.47	3.47	0.64	0.41	–

Table 4.1: Comparison of GMC algorithms on quantitative dataset (*GT: Ground truth, BF: Backward-Forward, B: Backward).

and 45 seconds, captured at 25 or 30 FPS. In addition, we form another "qualitative dataset" with 200 *unlabeled* videos from SVW, in challenging categories of boxing, diving, and hockey.

To compare GMC over different temporal distances of frames, for each video of length M in the quantitative dataset, we manually align all 10 possible pairs from the 5-frame set, $\mathbb{F} = \{1, 0.25M, 0.5M, 0.75M, M\}$, as long as they are overlapping, and specify the background regions. For this, a GUI is developed for a labeler to match 4 points on each frame pair, and fine tune them up to a half-pixel accuracy, until the background difference is minimized. Then, the labeler selects the foreground regions which subsequently identify the background region. Similar to [88], we quantify the consistency of two warped frames $\mathbf{I}^{(i)}(\theta_i)$ and $\mathbf{I}^{(j)}(\theta_j)$ (0 to 1 grayscale pixels) via the background region error (BRE),

$$\text{BRE}(i,j) = \frac{1}{\|\mathbf{M_B}\|_1} \| |(\mathbf{I}^{(i)}(\theta_i) - \mathbf{I}^{(j)}(\theta_j))| \odot \mathbf{M_B} \|_1, \tag{4.12}$$

where $\odot$ is element-wise multiplication and $\mathbf{M_B}$ is the background mask for the intersection of two warped frames.

4.3.3 Quantitative evaluation

Average of BRE over all the temporal frames pairs is shown in Table 4.1. TRGMC outperform all the baseline methods with considerable margin. The *backward-forward (BF)* scheme has a slightly better accuracy than the *backward (B)* scheme, and is also more stable based on our visual

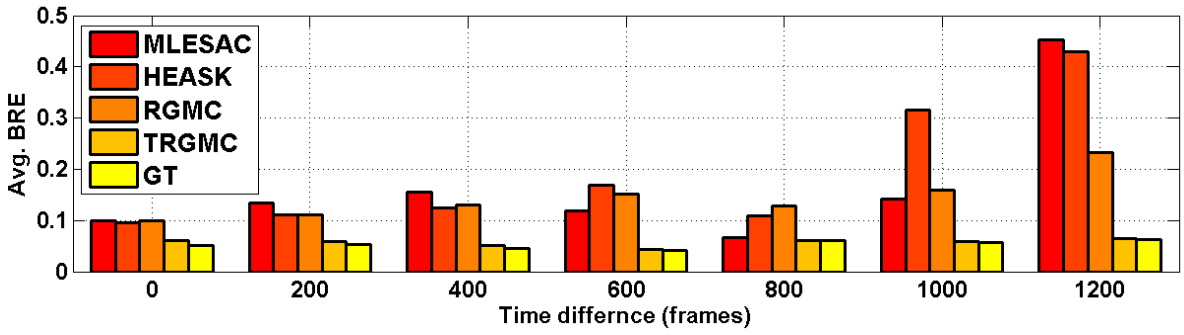


Figure 4.6: Average BRE of frame pairs versus the time difference between the two frames.

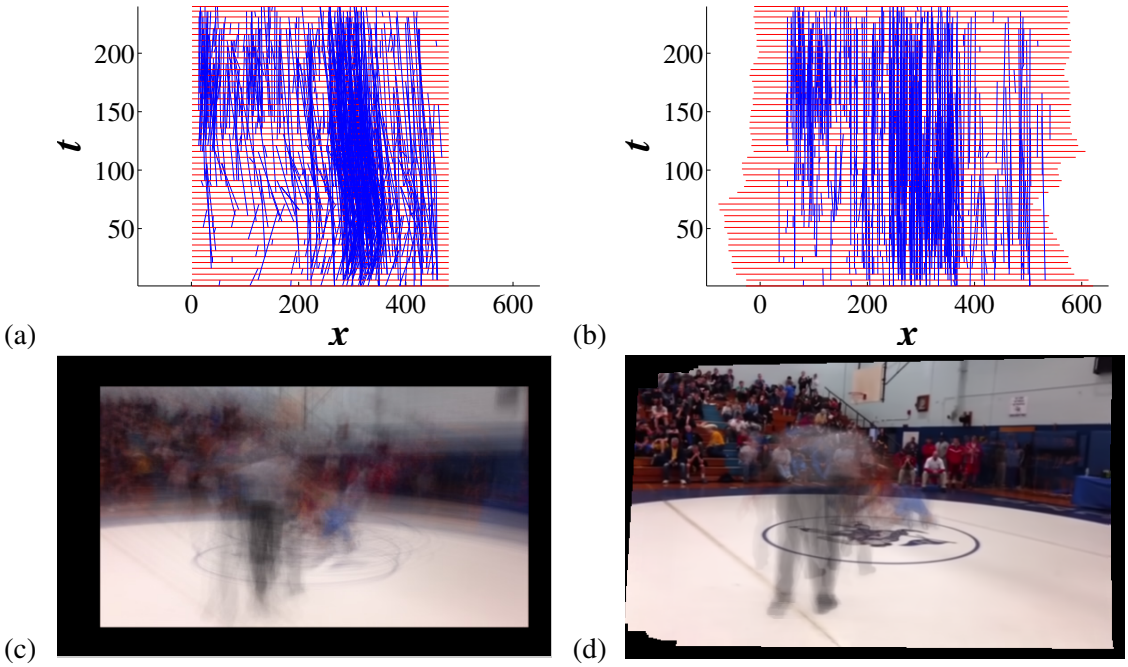


Figure 4.7: Top view of the frames and links (a) before and (b) after TRGMC. The parallel links in (b) show successful *spatial* alignment of keypoints. Average of frames (c) before and (d) after TRGMC. For better visibility, we show up to 15 links emanated per frame.

observation. Thus, we use BF as the default scheme for TRGMC.

To illustrate how the accumulation of errors over time affects the final error, Fig. 4.6 summarizes the average error versus the time difference between the frames in $\mathbb{F}$. This shows that TRGMC error is almost constant over a wide temporal distance between the frames. Thus, even if a frame is not aligned accurately, the error is not propagating to all the frames after that. However, in sequential GMC, the error increases as the time difference increases.

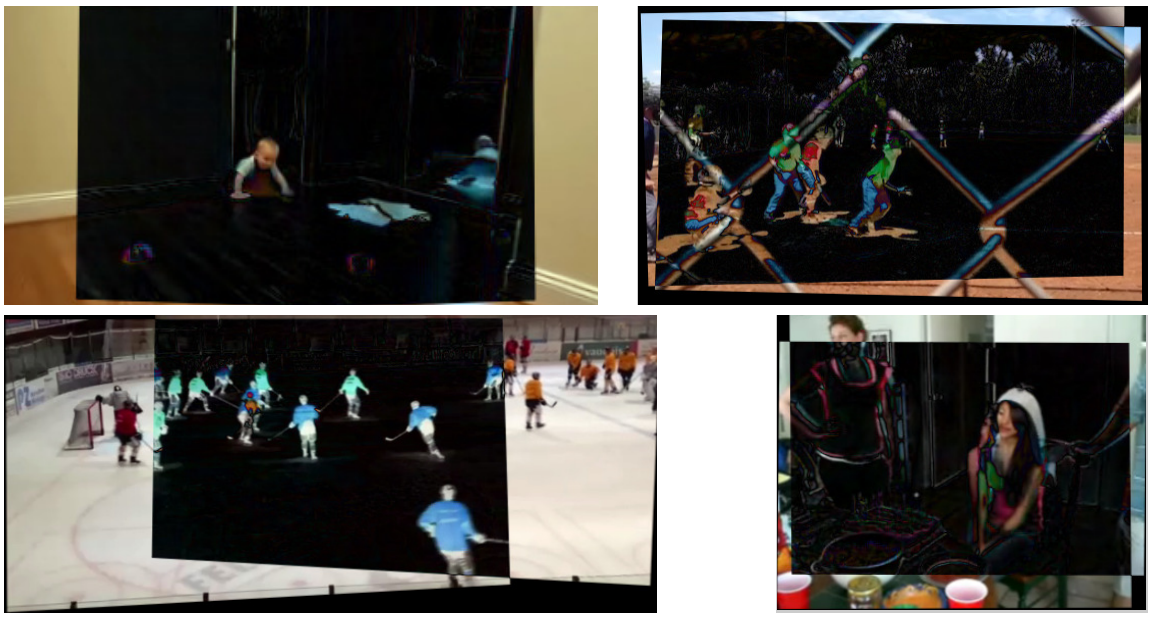


Figure 4.8: Composite image formed by overlaying the frame n on frame 1 for several videos after TRGMC. Left to right, top to bottom, n is equal to 144, 489, 912, 93, respectively. In the overlap region the difference between the frames is shown.

4.3.4 Qualitative evaluation

While quantitative results are comprehensive, the number of videos is limited by the labeling cost. Thus, we further compare TRGMC and the best performing baseline, RGMC, on the larger qualitative dataset. The resultant motion panoramas were *visually* investigated and categorized into three cases: good, shaking, and failed (i.e., considerable background discontinuity). The comparison in Tab. 4.2 again shows the superiority of TRGMC.

Figure 4.7 shows the *links* of a sample video processed by TRGMC, and the average frames, before and after processing. Initialization module is disable for generating this figure to better illustrate how well the spatial coordinate of the keypoints are aligned, resulting in links parallel to the $t-$ axis. This video also shows how GMC might be utilized for video stabilization. Figure 4.8 shows a composite image formed by overlaying the last frame (or a far apart frame with enough overlap) on frame 1 for several videos, after TRGMC. In the overlap region, difference between the two frames is shown, to demonstrate how well the background region matches for the frames

with large temporal distance.

4.3.5 Computational efficiency

Table 4.1 also presents the average time for processing each frame for each method, on a PC with an Intel i5-3470@3.2GHz CPU, and 8GB RAM. While obtaining considerably better accuracy than HEASK or RGMC, TRGMC is on average 15 times faster than HEASK and 7 times faster than RGMC. MLESAC is $\sim$3 times faster than TRGMC, but with twice the error. For TRGMC, the backward scheme is 50% faster than forward-backward, since it has approximately half the links of BF.

4.3.6 Accuracy vs. efficiency trade-off

Fig. 4.9 presents the error and efficiency results for a set of 5 videos versus the keyframe selection step, Δf. For this set, the ground truth error is 0.049. As a sweet spot in the error and efficiency trade-off, we use $\Delta f = 10$ for TRGMC. This figure also justifies the two stage processing scheme in TRGMC, as processing frames at a low selection step Δf, is costly in terms of efficiency, but only improves the accuracy slightly.

Alg. \Perfromance	Good	Shaking	Failed
RGMC	64%	33%	3%
TRGMC	**93%**	5%	2%

Table 4.2: Comparison of GMC algorithms on qualitative dataset.

4.4 Conclusions

We proposed a temporally robust global motion compensation (TRGMC) algorithm by joint alignment (congealing) of frames, in contrast to the common sequential scheme. Despite complicated

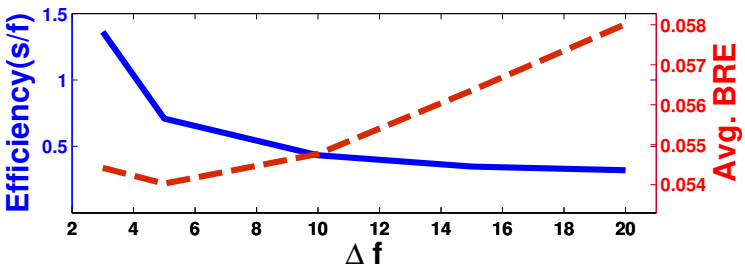


Figure 4.9: Error and efficiency vs. the keyframe selection step, Δf.

camera motions, TRGMC can remove the *intentional* camera motion, such as pan, as well as *unwanted* motion due to vibration on handheld cameras. Experiments demonstrate that TRGMC outperforms existing GMC methods, and applications of TRGMC.

The enabling assumption of TRGMC is that the camera motion in the direction of the optical axis is negligible. For instance, TRGMC will not work properly on a video from a wearable camera of a pedestrian, since in the global coordinate the upcoming frames grow in size and cause computational and rendering problems. Similar to panorama images, the best results are achieved if the optical center of the camera has negligible movement during the capturing, making a homography-based approximation of camera motion appropriate. However, if the optical center moves in the perpendicular direction to the optical axis (e.g., a camera following a swimmer), TRGMC still works well, but rendering the results in the form of motion panorama will be degraded by parallax effect.

Chapter 5

Global Motion Compensation Applications

There are a wide range of applications which can benefit from a robust and accurate global motion compensation (GMC) algorithm. Basically, many algorithms which are tailored to work only with static cameras benefit from GMC, as GMC transforms the video from a freely moving camera to a video from a pseudo-static camera in which background pixels are static over progression of time. Besides, some byproducts of GMC such as background reconstructions and motion panoramas themselves provide an interesting visualization of the captured video.

In these chapter, we briefly investigate these applications and then in Chapter 6 propose an algorithms for spatio-temporal alignment of non-overlapping sequences which is enabled by TRGMC.

5.1 Motion panorama

By sequentially reading input frames, applying the transformation found by TRGMC, and overlaying the warped frames on a sufficiently large canvas, a motion panorama is generated. Furthermore, it is possible to reconstruct the background using the warped frames *first* (as will be discussed later), and overlay the frames on that, to create a more impressive panorama. The last frame on the video generated such, can be referred to as a panoramic mosaic [101]. Figure 5.1 shows a few exemplar panoramas along with the camera motion pattern. For all the input videos of length M, we apply $(\frac{1}{2}(\theta_1 + \theta_M))^{-1}$ to the transformations found by TRGMC to normalize the result and have a better view of the scene in a smaller spatial area.

5.2 Raster scan of scenes/Image mosaic

We may swipe the camera through a large scene in a raster scan fashion and use TRGMC to reconstruct a big image mosaic. Note that this scenario is non-trivial since the accumulated error can be obvious when the raster scan comes back to the original camera position. The long term robustness presented by TRGMC is crucial in this scenario.

5.3 Background reconstruction

Background reconstruction is important for removing occlusions, or detecting foreground [74]. To reconstruct the background, a weighted average scheme is used to weight each frame by the *reliability map*, $\mathbf{R}^{(i)}$, which assigns higher weights to background. Since the minimum value of $\mathbf{R}^{(i)}$ is a positive constant η, if no reliable keyframe exists at a coordinate, all the frames will have equal weights. Specifically, the background is reconstructed by $\mathbf{B} = \frac{\sum_{i\in\mathbb{K}} \mathbf{R}^{(i)}(\theta_i)\mathbf{I}^{(i)}(\theta_i)}{\sum_{i\in\mathbb{K}} \mathbf{R}^{(i)}(\theta_i)}$, where $\mathbf{R}^{(i)}(\theta_i)$ and $\mathbf{I}^{i}(\theta_i)$ are the reliability map and the input frame warped using the transformation θ_i. Using our scheme, reconstructed background in Fig. 5.2 is sharper and less impacted by the

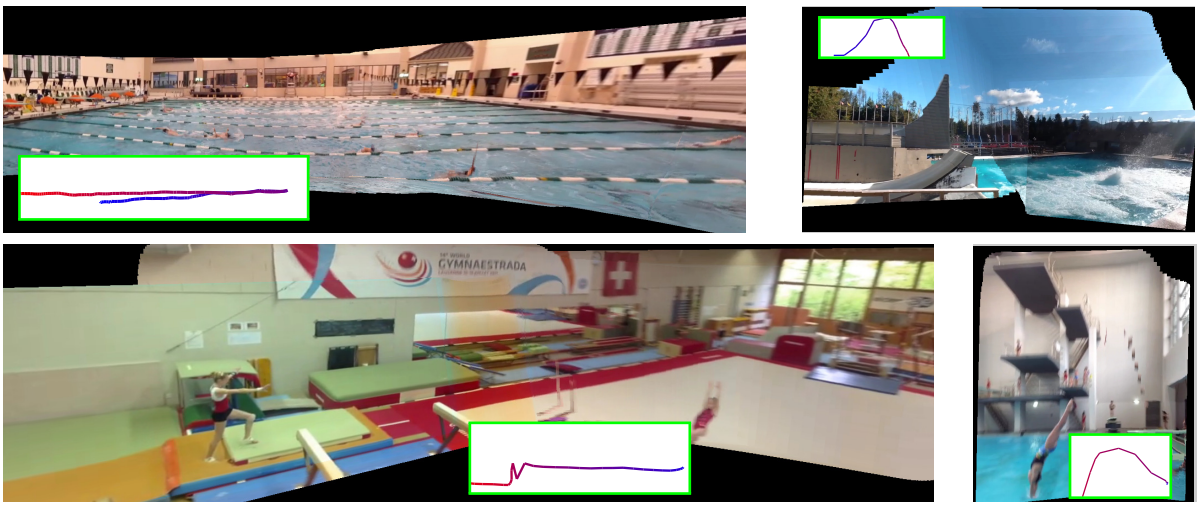


Figure 5.1: Temporal overlay of frames from different videos processed by TRGMC. Trajectory of the center of image plane over time is overlaid on each plot to show the camera motion pattern, where color changes from blue to red with progression of time.

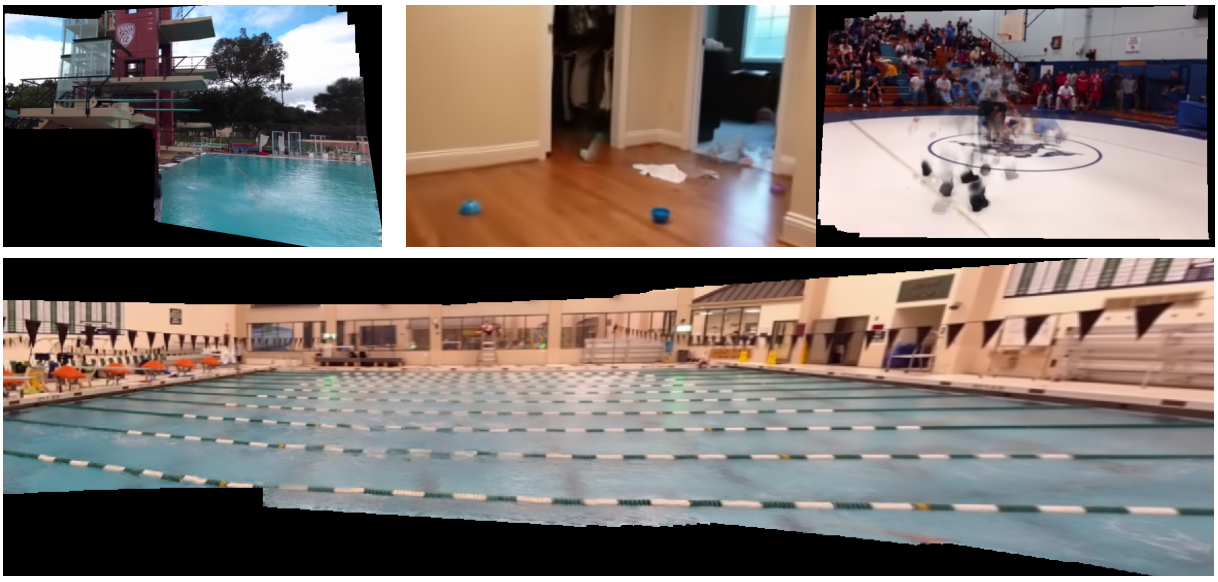


Figure 5.2: Background reconstruction results.

foreground.

5.4 Foreground segmentation

The reliable background reconstruction result **B** as calculated in Section 5.3 along with the GMC result of frame $\mathbf{I}^{(i)}$, i.e., θ_i, can be easily used to segment the foreground by thresholding the difference image, $|\mathbf{B} - \mathbf{I}^{(i)}(\theta_i)|$ (See Fig. 5.3).

5.5 Human action recognition

State of the art human action recognition heavily relies on analysis of human motion. For instance, the dense trajectories algorithm [114] for motion analysis reveals its power when camera motion is compensated in the input video, either as pre-processing step, or internally [117]. Otherwise, camera motion interferes with human motion, making the analysis problem very challenging. In [117], camera motion is compensated by detecting human and removing motion vectors due to human motion, and relying on RANSAC algorithm for outlier rejection. However, this internal GMC

(a)  (b) 

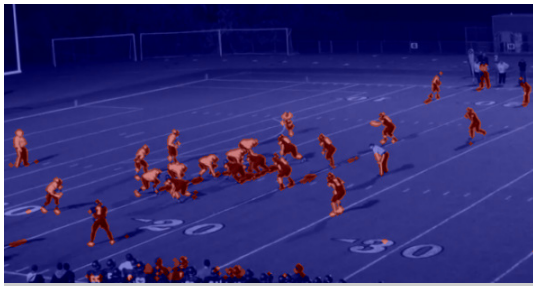
(c) 

Figure 5.3: Foreground segmentation: (a) Input frame, (b) reconstructed background, (c) difference of (a,b) on (a).

(a)  (b) 

Figure 5.4: Dense trajectories of the (a) original video, and (b) TRGMC-processed video.

requires accurate human detection, which has a high failure rate in videos in the wilds, specially for highly articulated human body in sports videos, and loses performance when number of false matches increases. Fig. 5.4 illustrates the difference of dense trajectories calculated on an input video with and without application of TRGMC. As shown in this figure, utilizing TRGMC before extraction of dense trajectories effectively suppresses the camera motion effect.

5.6 Multi-object tracking (MOT)

When appearance cues for tracking are ambiguous, e.g., tracking players in team sports like football, motion cues gain extra significance [26, 55]. MOT is comprised of two tasks, data association by assigning each detection a label, and trajectory estimation – both highly affected by camera motion. TRGMC can be applied to remove camera motion and thus, revive the power of tracking algorithms relying on motion cues. To verify the impact of TRGMC, we manually label the locations of all players in 566 frames of a football video (the one in Fig. 5.3) and use this ground truth detection results to study how MOT using [3] benefits from TRGMC. Fig. 5.5 compares the trajectories of players over time with and without applying TRGMC. Comparing number of label switches, this qualitatively demonstrates improvement of a challenging MOT scenario using TRGMC. Also, the Multi-Object Tracking Accuracy (MOTA) [12] achieved for the original video and the video processed by TRGMC are 63.79% and 84.23%, respectively.

5.7 Spatio-temporal alignment of non-overlapping sequences

Spatio-temporal alignment of multiple videos is an important computer vision problem with a wide range of applications. Previous works study different aspects and scenarios of the spatio-temporal alignment. Given the capabilities of the accurate GMC provided by TRGMC, it is possible to design algorithms for spatio-temporal alignment of sequences in new scenarios. Specifically, since it is possible to transform each given sequence to a global coordinate and also reconstruct the background for each video sequence, it is possible to register the background images, and subsequently register each frame of each sequence with other sequences. To this end, we propose a new algorithm for spatio-temporal alignment of sequences, for *non-overlapping sequences* (NOS), which is presented in details in Chapter 6. Targeted NOS are captured by *freely* and *independently* panning

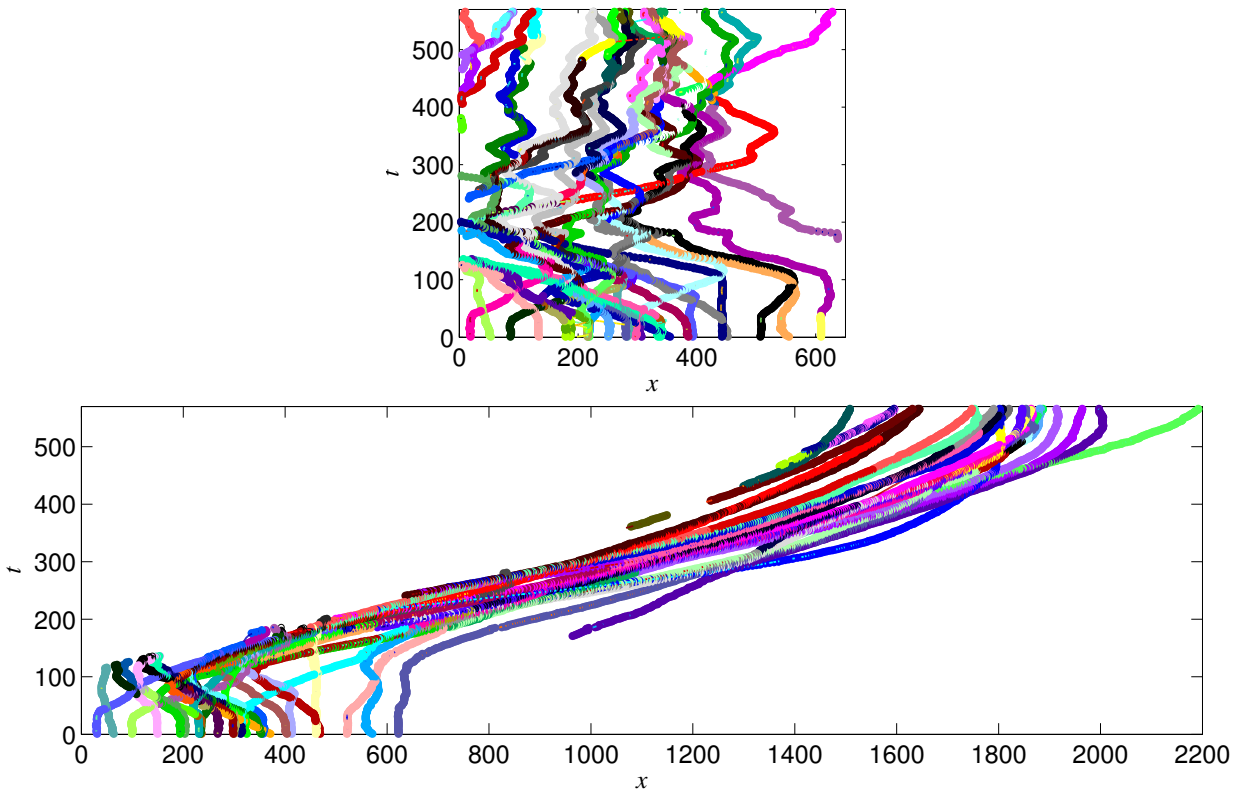


Figure 5.5: Multi-player tracking using [3] for a football video with camera panning to the right, before (top) and after processing by TRGMC (bottom).

cameras from *nearby* viewpoints. In NOS, sequences might not have any pair of frames that have spatial overlap and belong to the same world time instant. More interestingly, sequences might even not cover some common regions of the same scene over the progression of time.

Our algorithm uses TRGMC to map each frame to a camera-motion-removed video and reconstruct the background for each sequence, independently. These potentially *non-overlapping* backgrounds are aligned via appearance cues and also the prediction that ***where*** a moving object leaving field of view of a camera will appear in field of view of another camera. Given the spatial alignment, we predict ***when*** a moving object leaving field of view of one camera will appear in field of view of another to come up with the temporal. We mathematically formulate this prediction and estimate the temporal synchronization.

Chapter 6

Spatio-Temporal Alignment of Non-Overlapping Sequences from Independently Panning Cameras

6.1 Introduction

Spatio-temporal alignment of multiple videos [16, 27, 28, 33, 39, 76, 80, 95, 120] is a well-studied vision problem with a wide range of applications, e.g., human action recognition [82, 110], video editing [120], markerless motion capture [39], video mosaicing, change detection [27], and abandoned object detection [48]. Previous works study different aspects and scenarios of the spatio-temporal alignment. Some works target sequences from the *same* scene but *different* viewpoints [39, 76]. Some can handle sequences recorded at *different* times by independent moving cameras that follow a *similar* trajectory [28, 33, 120]. The seminal work of Caspi and Irani [16] studies spatially non-overlapping sequences when two fixed cameras move *jointly* in space.

Our work covers a novel unexplored aspect of spatio-temporal alignment of sequences, for *non-overlapping sequences* (NOS). Targeted NOS are captured by *freely* and *independently* panning cameras, from *nearby* viewpoints, with limited translation, especially in optical axis direction. In NOS, sequences might not have any pair of frames that have spatial overlap and belong to the same world time instant. More interestingly, sequences might even not cover some common regions

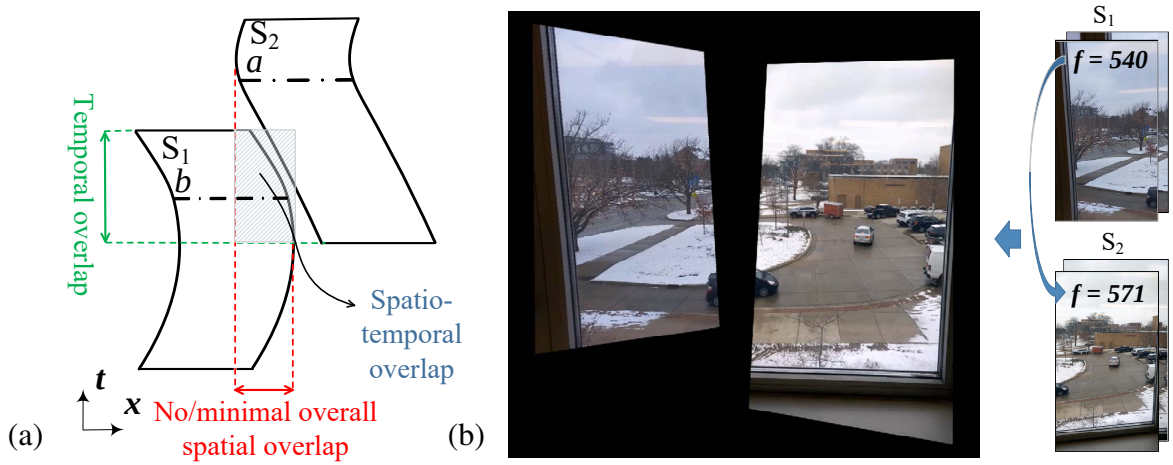


Figure 6.1: (a) Top view of spatio-temporal FOV of two moving cameras capturing sequences S_1 and S_2; Non-overlapping sequences (NOS) may not even cover some common spatial region over the progression of time, i.e, no overall spatial overlap will exist. (b) Spatio-temporal alignment of NOS results in displaying sequences from multiple freely panning cameras in a common coordinate and at the correct time shift.

of the same scene over the progression of time. In other words, if we reconstruct the observed background by these sequences, the backgrounds may be non-overlapping, i.e., in Fig. 6.1 (a), overall spatial overlap does not exist.

Given the ubiquitousness of smartphones and wearcams, NOS are increasingly common. When amateur users unsynchronizedly shoot videos of an event, aligning these videos leads to a single comprehensive video, with greater spatial and temporal spans (Fig. 6.1 (b)). This resultant video is essentially a panoramic video, shot by smartphones, without the need to fix the cameras to each other or use tripods, with the best visual presentation achieved if there exists even a tiny overlap. Further, in cases of crime actions or violations where many witnesses capture videos from the incident, each sequence may cover part of the story. Aligning these videos into a unified *large-scale* 3D volume provides a better grasp of the full picture.

The existing spatio-temporal alignment algorithms fail in the case of NOS, since even if there is some overall spatial overlap, spatial alignment of apparently overlapping frames, as Fig. 6.1 (a) shows for frames a and b, obviously violates the temporal alignment. However, by decompos-

ing the task to spatial alignment first and then temporal alignment based on scene dynamics, the problem can be solved. In general our proposed algorithm assumes NOS satisfy the following two assumptions. 1) Although the sequences do not need to have any corresponding frames that share a common scene at the same world time stamp, and no overall overlap as in Fig. 6.1 (a), they cover nearby parts of a scene from similar view angles. 2) There are moving objects in the scene which move from the field of view (FOV) of one camera to FOV of other cameras, or if the sequences happen to have overlap, have motion in the overlap region.

Our algorithm uses global motion compensation to map each frame to a camera-motion-removed video and reconstruct the background for each sequence, independently. With the two assumptions, these potentially *non-overlapping* backgrounds are aligned via appearance cues and also the prediction that ***where*** a moving object leaving FOV of a camera will appear in FOV of another camera. Collection of the former mappings and the latter background alignment, can put each frame in each sequence in correct spatial alignment w.r.t. frames from other sequences. Given the spatial alignment and the assumption 2, we predict ***when*** a moving object leaving FOV of one camera will appear in FOV of another. We mathematically formulate this prediction and estimate the temporal synchronization.

In summary, this chapter makes these contributions:

◇ A new scenario in spatio-temporal alignment of sequences is identified and targeted.

◇ A spatial alignment algorithm for NOS via alignment of reconstructed backgrounds and consistency of objects movement is proposed.

◇ The trajectory of moving objects with smooth path are used as a clue for temporal alignment of NOS.

6.2 Previous Work

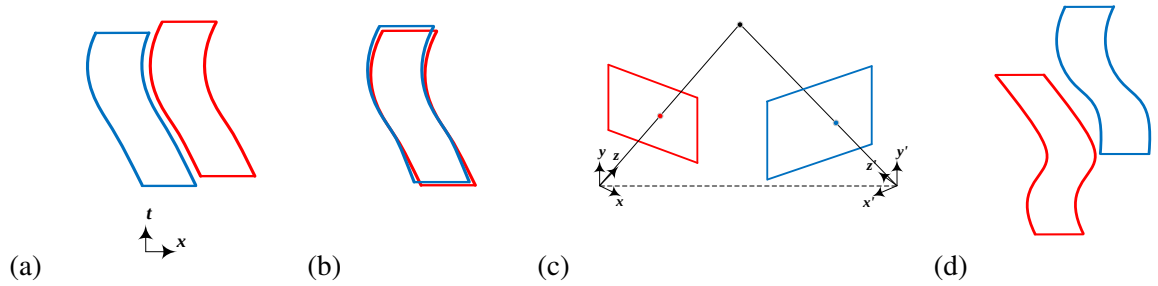
The prior work in spatio-temporal alignment of sequences mostly differ in their *a*ssumptions and *s*cenarios, e.g., the camera movement (static, jointly moving, or moving), camera view-point (similar or distinct), extent of overlap in sequences, and extent of similarity of camera motion paths. The work of [34] presents an excellent taxonomy of these assumptions, one of which is that, to align sequences from the same event captured by freely moving cameras, coherent scene appearance is assumed. We lift this assumption by handling non-overlapping sequences, although we do assume negligible camera movement in the optical axis direction. We now review key scenarios in prior work.

6.2.1 Jointly moving cameras

Caspi and Irani align spatially non-overlapping sequences when two closely *attached* cameras move *jointly* in space (Fig. 6.2 (a)) [16]. Assuming cameras share the same projection center, their relationship is modeled as a fixed homography **H**, estimated based on the idea that the apparent motion in camera 1 is related to camera 2 with **H**. Esquivel et al. [32] relax the projection center assumption and calibrate a multi-camera rig from non-overlapping views, assuming synchronized sequences. In contrast, the cameras in NOS can pan freely with no overlap, which is substantially more challenging.

6.2.2 Cameras following similar trajectories

The authors of [28, 33, 34, 120] align sequences recorded at *different* times by independent moving cameras that follow a *similar* trajectory (Fig. 6.2 (b)). [28] assumes one sequence as the reference and the other sequences entirely contained (temporally) within the reference. The alignment is formulated as an energy minimization problem alternately solved for temporal and spatial alignment



(a) (b) (c) (d)

Figure 6.2: Various scenarios in spatio-temporal alignment of sequences: (a) jointly moving cameras, (b) independently moving cameras at different times following similar trajectories, (c) stationary cameras with different viewpoints, (d) the proposed independently panning cameras with non-overlapping sequences.

parameters and is evaluated on four sets of real videos. In [120] an interactive method for nonlinear temporal video alignments is proposed for video editing. All these methods require coherent scene appearance and are not capable of handling sequences from moving cameras with no overlap in FOV — the targeted scenario of NOS.

6.2.3 Stationary cameras at different views

Padua et al. [76] target $n \geq 2$ sequences from the *same* scene but *different* viewpoints (Fig. 6.2 (c)). The *stationary* cameras allow the estimated camera's epipolar geometry remain fixed. Motion trajectories are used as cues for both spatial and temporal alignment. Experimental results are provided for 5 sequences, however, as the proposed method is not dependent on a specific tracker, for each sequence, the optimal tracker is chosen based on the application in hand.

6.2.4 Time synchronization

Many prior works have focused on time synchronization of sequences. Assuming the known 3D object location and calibrated stationary cameras, [15] synchronizes non-overlapping sequences of these cameras. Gaspar et al. [37] propose a synchronization algorithm for the case that two

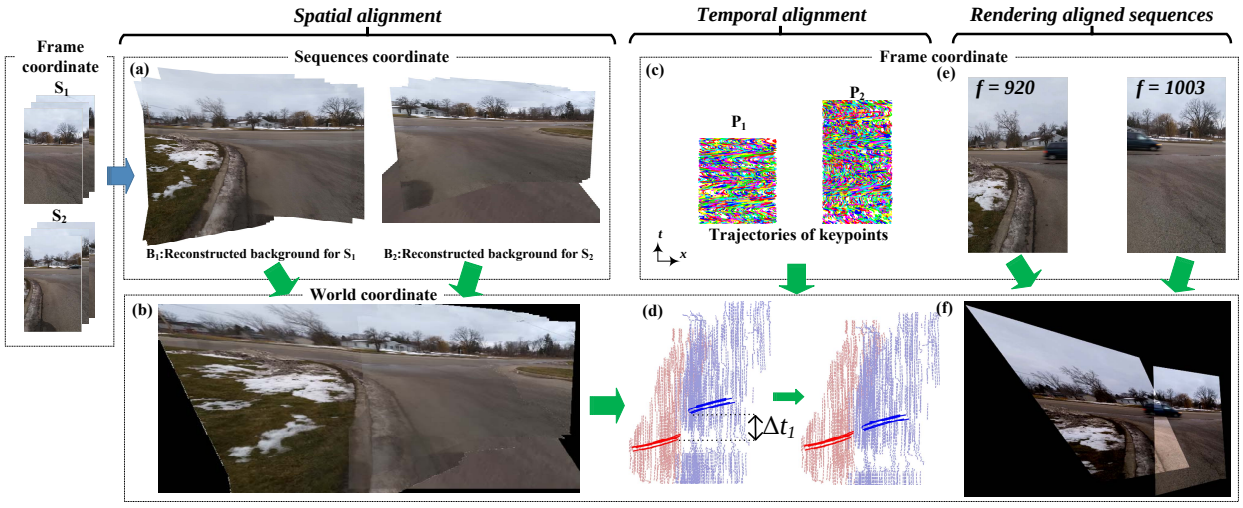


Figure 6.3: Flowchart of our spatio-temporal alignment algorithm. First, spatial alignment is performed by background reconstruction for each sequence (a) and aligning the backgrounds (b). Second, given the spatial alignment parameters, keypoint trajectories (c) are mapped to the world coordinate and the best temporal alignment in terms of continuity of moving object trajectories is found (d). Finally, spatio-temporal alignment parameters are used for displaying the sequence in a world coordinate system and at the correct time shift (e).

cameras move independently, even if different features are tracked in two sequences. It assumes the known intrinsic camera parameters and two visible rigid objects in both sequences, whose relative motion is used for synchronization. Lu and Mandal [67] model the video temporal alignment as a spatio-temporal discrete trajectory alignment problem. The method is evaluated on synthetic trajectories and 10 pairs of real videos. Our method also relies on existence of at least one moving object for temporal synchronization. In fact, without spatial overlap between FOVs, any temporal alignment algorithm has to track moving objects or egomotion [34]. However, we can work with non-overlapping sequences where the same moving object is not visible at the same time in all sequences, without relying on camera calibration or known moving object location.

6.3 Proposed Method

We discussed the assumptions for the proposed spatio-temporal alignment of NOS in Section 6.1. The intrinsic and extrinsic camera parameters are not required. Also, the cameras might be un-

synchronized, i.e., the capture starts at different times, with possibly distinct frame rates, and are panned freely and independently. However, best results are achieved by small camera baseline and limited translation of cameras, especially in the optical axis direction.

The proposed algorithm has two stages, (1) spatial alignment (Fig. 6.3(b)), which relies on the reconstructed backgrounds' appearance and consistency of movement of objects across the sequences, (2) temporal alignment (Fig. 6.3(d)), which uses the continuity of objects' trajectories to synchronize the videos. Our method is feature-based, relying on keypoint correspondence for the first stage and keypoint trajectories for the first and second stage.

6.3.1 Notations

As shown in Fig. 6.3, *frame coordinate* refers to the pixel coordinate in the input video, *sequence coordinate* to the global coordinate of the reconstructed background of one video, and *world coordinate* to the global coordinate of *all* input videos where the final aligned video is rendered. We denote the coordinates and time stamps in the frame coordinate with plain letters, in the sequence coordinate with $\sim$ over the notation, e.g., $\tilde{\mathbf{x}}$, and in the world coordinate with double $\sim$, e.g., $\tilde{\tilde{\mathbf{x}}}$. Accordingly, a transformation from the frame to sequence coordinate has $\sim$ over the notation, and a transformation from the sequence to world coordinate has double $\sim$. We use superscript for the sequence number and subscript for, either the frame number or trajectory number. E.g., $\tilde{\theta}_i^s$ is the transformation of frame i in sequence s from the frame coordinate to sequence coordinate.

6.3.2 Spatial alignment

We break down the spatial alignment to two phases. First, for each sequence, we map all the frames to the sequence coordinate, via global motion compensation (GMC), which also produces a reconstructed background mosaic (Fig. 6.3(a)). A crucial assumption for successful GMC is the

camera having small motion in the optical axis direction. Second, image alignment is conducted on the reconstructed backgrounds and maps them to the world coordinate (Fig. 6.3(b)). However, if the backgrounds are non-overlapping, common image alignment cannot be used. Thus, a new alignment scheme is proposed in Sec. 6.3.2.3.

6.3.2.1 Global motion compensation

GMC removes any intentional or unwanted camera motion in a sequence, creating a video with static background [86, 88]. Essentially, GMC estimates a per frame transformation to the sequence coordinate. This work utilizes the TRGMC algorithm [86], discussed in Chapter 5, which handles dynamic scenes and estimates the transformations by joint alignment of input frames. TRGMC first detects SURF [11] keypoints in each frame, and performs keypoint matching to densely interconnect all frames, regardless of their temporal offset. These connections are referred as *links*. Then the keypoint-based congealing applies appropriate transformation to each frame and its links, such that the spatial coordinates of the end-points of each link are as similar as possible.

For the convenience of readers, we briefly introduce the keypoint-based congealing. Given a stack of N frames $\{\mathbf{I}_i\}$, with indices $i \in \mathbb{K} = \{k_1, ..., k_N\}$, keypoint-based congealing is formulated as an optimization problem,

$$\min_{\{\tilde{\theta}_i^s\}} \varepsilon^s = \sum_{i \in \mathbb{K}} [\mathbf{e}_i(\tilde{\theta}_i^s)]^\intercal \Omega_i^s [\mathbf{e}_i(\tilde{\theta}_i^s)], \tag{6.1}$$

where $\tilde{\theta}_i^s$ is an 8-dim homography transformation parameter from frame i of sequence s to the sequence coordinate, $\mathbf{e}_i(\tilde{\theta}_i^s)$ collects pair-wise alignment errors of frame i relative to all other frames, and Ω_i^s is a weight matrix. Since TRGMC uses homography transformation, it works best with nodal camera motion. In the case of camera translation, TRGMC still works by matching the dominant background, although the result may downgrade with parallax.

The alignment error of frame i relative to all other frames is the sum of squared differences (SSD) between the coordinates of the endpoints of all links connecting keypoints of frame i to keypoints of other frames. The coordinates of the start and end point of each link k starting from frame i are donated as $(x_{i,k}, y_{i,k})$ and $(u_{i,k}, v_{i,k})$, respectively. The error $\mathbf{e}_i(\tilde{\theta}_i^s)$ is,

$$\mathbf{e}_i(\tilde{\theta}_i^s) = [\triangle\mathbf{x}_i(\tilde{\theta}_i^s)^\intercal, \triangle\mathbf{y}_i(\tilde{\theta}_i^s)^\intercal]^\intercal, \tag{6.2}$$

where

$$\triangle\mathbf{x}_i(\tilde{\theta}_i^s) = \tilde{\mathbf{w}}_i^{(x)} - \mathbf{u}_i, \qquad \triangle\mathbf{y}_i(\tilde{\theta}_i^s) = \tilde{\mathbf{w}}_i^{(y)} - \mathbf{v}_i, \tag{6.3}$$

are the errors in x and $y-$axes. The vectors $\tilde{\mathbf{w}}_i^{(x)} = [\mathscr{W}_x(x_{i,k}, y_{i,k}; \tilde{\theta}_i^s)]$ and $\tilde{\mathbf{w}}_i^{(y)} = [\mathscr{W}_y(x_{i,k}, y_{i,k}; \tilde{\theta}_i^s)]$ denote the x and $y-$coordinates of $(x_{i,k}, y_{i,k})$ warped by the parameter $\tilde{\theta}_i^s$, respectively. The vectors $\mathbf{u}_i = [u_{k,i}]$ and $\mathbf{v}_i = [v_{k,i}]$ denote the coordinates of the end points.

Equation 6.1 is solved by taking the Taylor expansion around $\tilde{\theta}_i^s$ and finding the increment $\Delta\tilde{\theta}_i^s$ that minimizes,

$$\arg\min_{\Delta\tilde{\theta}_i^s} [\mathbf{e}_i(\tilde{\theta}_i^s) + \frac{\partial\mathbf{e}_i(\tilde{\theta}_i^s)}{\partial\tilde{\theta}_i^s}\Delta\tilde{\theta}_i^s]^\intercal\Omega_i^s[\mathbf{e}_i(\tilde{\theta}_i^s) + \frac{\partial\mathbf{e}_i(\tilde{\theta}_i^s)}{\partial\tilde{\theta}_i^s}\Delta\tilde{\theta}_i^s] + \gamma\Delta\tilde{\theta}_i^{s\intercal}\mathscr{I}\Delta\tilde{\theta}_i^s, \tag{6.4}$$

where $\Delta\tilde{\theta}_i^{s\intercal}\mathscr{I}\Delta\tilde{\theta}_i^s$ is a regularization term which stabilizes the changes to the transformation, with a positive constant γ setting the trade-off. The indicator matrix $\mathscr{I}$ is a diagonal matrix specifying which elements of $\Delta\tilde{\theta}_i^s$ need a constraint.

By setting the first-order derivative of Eqn. 6.4 to zero, a closed-form solution for $\Delta\tilde{\theta}_i^s$ is obtained. After enough iterations, $\tilde{\theta}_i^s$ will be the transformation mapping frame i of the input video to the sequence coordinate.

6.3.2.2 Spatial alignment of overlapping backgrounds

Given the $\tilde{\theta}_i^s$ for all the input videos, we follow [86] to reconstruct the backgrounds B^s for them. If there exists enough overlap between the backgrounds, common image alignment algorithms may be used. Specifically, we estimate the transformation $\tilde{\tilde{\theta}}^s$ that maps the background of sequence s to the world coordinate, by matching SURF keypoints on background images via the vector field consensus algorithm [70]. In summary, a point with the homogeneous coordinate $(x, y, 1)$ in frame i of sequence s is mapped to the sequence coordinate of sequence s, denoted as $(\tilde{x}, \tilde{y}, 1)$, and the world coordinate of all sequences, denoted as $(\tilde{\tilde{x}}, \tilde{\tilde{y}}, 1)$,

$$\begin{bmatrix} \tilde{\tilde{x}} \\ \tilde{\tilde{y}} \\ 1 \end{bmatrix} = \tilde{\tilde{\theta}}^s \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \tilde{\tilde{\theta}}^s \tilde{\theta}_i^s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{6.5}$$

Thus, the transformation $\tilde{\tilde{\theta}}^s \tilde{\theta}_i^s$ conducts spatial alignment for frame i in sequence s. Given the homography transformation of $\tilde{\tilde{\theta}}^s$, as the cameras' baseline increases, the dominant background plane is aligned, and the foreground may be affected by parallax in the final composite video.

6.3.2.3 Spatial alignment of non-overlapping sequences

With freely panning cameras, it is likely that the backgrounds of sequences have no overlap, or the overall overlap is too small to reliably estimate the spatial alignment, flagged with noisy keypoint matches using vector field consensus [70]. One potential solution is to extrapolate the background images, and align the extrapolated images, similar to [79]. However, our experiments reveal that this is not reliable. First, extrapolation introduces many artifacts [7], or blurred areas [2, 79], leading to poor keypoint matching. Second, extrapolation in horizontal direction, helps with alignment in vertical direction, but leaves lots of ambiguity in horizontal alignment. Third, a rigid Euclidean

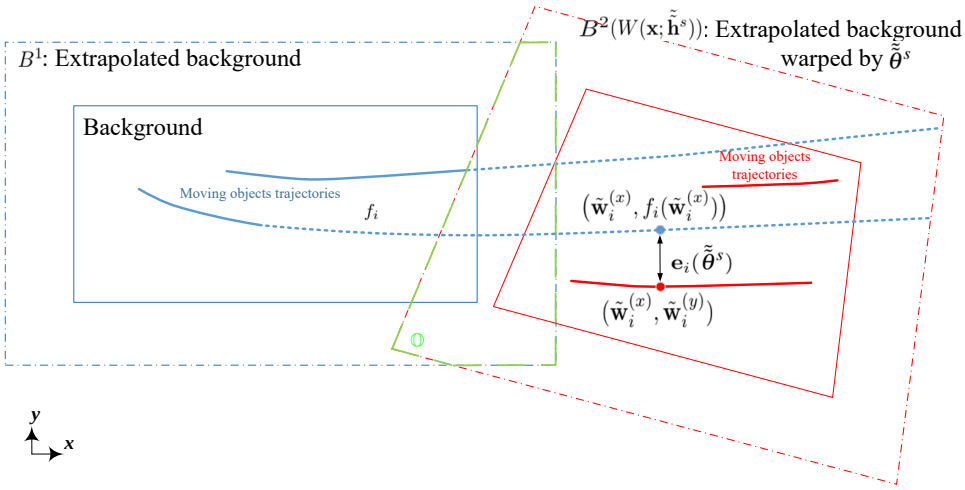


Figure 6.4: Spatial alignment of non-overlapping sequences using background extrapolation and smoothness of object trajectories.

transformation, as in [79], does not suffice for a proper background alignment.

On the other hand, how objects move across the sequences in the *spatial* world coordinate, irrespective of temporal synchronization, provides hints for spatial alignment (Fig. 6.4). There is ambiguity in the exact spatial alignment, however, as more objects move across the sequences and in more diverse directions, the ambiguity is decreased.

To enable spatial alignment for non-overlapping sequences, we propose a spatial alignment algorithm that combines both aforementioned ideas. We first extrapolate the background images of all sequences. Then, we perform *m*otion tracking to obtain trajectories of all keypoints in each sequence. By transforming the trajectories to the sequence coordinate using $\tilde{\theta}_i^s$ and filtering out static trajectories, we collect moving object trajectories. We create *m*otion tracks by matching moving object trajectories across sequences. Finally, we incrementally update the transformation applied to the background images to increase the motion track smoothness in the world coordinate, while ensuring that this update will not violate the appearance consistency of *extrapolated* backgrounds

in the overlap region.

Motion tracking We perform tracking in consecutive frames to form the trajectories. Among various schemes, we prefer the keypoint-based tracking for two reasons. 1) Object-based tracking requires detecting generic objects on each frame, which could be error-prone and inefficient. 2) Our experiments and also the analysis in [61] reveal that optical flow-based tracking such as dense trajectories [115] leads to spurious motion trajectories close to the motion boundaries. We use SURF [11] keypoint detector due to superior performance on blurry images, in comparison to SIFT [65]. To detect newly emerging objects, we start tracking all the keypoints on frame i who have no corresponding matches from frame $i-1$.

Denote the jth trajectory in sequence s as $P_j^s = [\mathbf{x}_j^s, \mathbf{y}_j^s, \tilde{\mathbf{t}}_j^s]$, where $\mathbf{x}_j^s$ and $\mathbf{y}_j^s$ are the frame coordinates, and $\tilde{\mathbf{t}}_j^s$ is the time stamp. To handle sequences at different frame rates, $\tilde{\mathbf{t}}_j^s$ should be the absolute time unit such as milliseconds not frame number. We then compute the trajectory $\tilde{P}_j^s$ in the sequence coordinate via $\tilde{\theta}_i^s$. In this coordinate, trajectories of moving and stationary keypoints are easily distinguishable, as sequence coordinates of static objects remain constant over time (Fig. 6.3(d), bold vs. dashed lines). Denoting the trajectory length as l_j^s and width and height of the sequence as w^s and h^s, we omit stationary trajectories if,

$$\frac{1}{l_j^s}\sum_{k=1}^{l_j^s-1}\left(\frac{|\tilde{\mathbf{x}}_{j,k}^s - \tilde{\mathbf{x}}_{j,k+1}^s|}{w^s} + \frac{|\tilde{\mathbf{y}}_{j,k}^s - \tilde{\mathbf{y}}_{j,k+1}^s|}{h^s}\right) < \tau_1, \tag{6.6}$$

where τ_1 is a threshold for the total displacement of the tracked object, and $\tilde{\mathbf{x}}_{j,k}^s$ and $\tilde{\mathbf{y}}_{j,k}^s$ denote the kth element in the vectors $\tilde{\mathbf{x}}_j^s$ and $\tilde{\mathbf{y}}_j^s$, respectively.

Creating motion tracks We describe each moving object trajectory j of sequence s with two SURF descriptors, one for the keypoint starting the trajectory $\mathscr{S}_j^s$ and one for the one ending it $\mathscr{E}_j^s$. To match two trajectories j and k from sequences s_1 and s_2, a classical keypoint matching algo-

rithm [65] is used to match all 4 combinations of keypoints, i.e., $(\mathscr{S}_j^{s_1}, \mathscr{S}_k^{s_2})$, $(\mathscr{E}_j^{s_1}, \mathscr{E}_k^{s_2})$, $(\mathscr{S}_j^{s_1}, \mathscr{E}_k^{s_2})$ and $(\mathscr{E}_j^{s_1}, \mathscr{S}_k^{s_2})$, and the minimum distance decides a match. This way, more robustness against view point variation is achieved, as the nearby keypoints of the trajectories (in the world coordinate) will be the deciding factor in trajectory matching. We call each set of the matched trajectories a *track*, denoted by Π_k. For simplicity of notation, we assume that the trajectories contributing to a track have been re-indexed such that $\Pi_k = \{\tilde{\tilde{P}}_k^s; s \in \{1, ..., S\}\}$. For a certain sequence s, $\tilde{\tilde{P}}_k^s$ might be empty, i.e., no trajectories from this sequence is part of the track Π_k. Note that not all trajectories should be matched to form tracks, as they might be due to noise or objects with non-smooth motion path. Sec. 6.3.3.2 presents a method to remove non-smooth trajectories.

Spatial alignment formulation For simplicity, we discuss the alignment of 2 sequences, as more sequences may be aligned in the same manner, sequentially. Also, we set $\tilde{\tilde{\theta}}^1 = I_{3\times 3}$ and use θ for $\tilde{\tilde{\theta}}^2$ to avoid cluttered equations. Given N tracks indexed by i, and extrapolated backgrounds B^1 and B^2, the goal is to find a transformation θ which maps B^2 to B^1, such that the pixel contents of the extrapolated background are consistent in the overlap region $\mathbb{O}(\theta)$ and trajectories of sequence 2 reside on the *extension* of trajectories in sequence 1. For image extrapolation, we use PatchMatch algorithm [7]. To further improve extrapolation results, for extrapolating each background, we use contents of *both* background images. Then, we formulate the problem as an optimization problem (Fig. 6.4),

$$\min_{\theta} \sum_{\mathbf{x}\in\mathbb{O}(\theta)} \left[B^2(\mathscr{W}(\mathbf{x};\theta)) - B^1(\mathbf{x})\right]^2 + \beta \sum_i \mathbf{e}_i(\theta)^\intercal \mathbf{e}_i(\theta), \tag{6.7}$$

where $\mathscr{W}(\mathbf{x};\theta)$ warps $\mathbf{x}$ by the transformation θ, and $\mathbf{e}_i(\theta)$ represents how far trajectory i of sequence 2 is from spatial extension of matching trajectory in sequence 1. The first term in Eqn. 6.7 is similar to Lucas-Kanade algorithm [5], operated only in the overlapping area. To define $\mathbf{e}_i(\theta)$, we fit a line, which based on our experiments works better than fitting polynomials, to the ith tra-

jectory in sequence 1 (in the sequence coordinate), denoted by $f_i(x)$. The vector $\mathbf{e}_i(\theta)$ collects the y-distance between each point on the ith trajectory in sequence 2, after warped by θ, and the fitted curve,

$$\mathbf{e}_i(\theta) = [\tilde{\mathbf{w}}_i^{(y)} - f_i(\tilde{\mathbf{w}}_i^{(x)})], \tag{6.8}$$

where $\tilde{\mathbf{w}}_i^{(x)} = [\mathscr{W}_x(x_{i,2}, y_{i,2}; \theta)]$ and $\tilde{\mathbf{w}}_i^{(y)} = [\mathscr{W}_y(x_{i,2}, y_{i,2}; \theta)]$ are the warped $\tilde{x}$ and $\tilde{y}-$coordinates of the ith trajectory in sequence 2 in the sequence coordinate.

The optimization problem is solved by taking the Taylor expansion around θ and finding the increment $\Delta\theta$ by,

$$\begin{aligned}\arg\min_{\Delta\theta} \sum_{\mathbf{x}\in\mathbb{O}(\theta)} \left[B^2(\mathscr{W}(\mathbf{x};\theta)) + \nabla B^2 \frac{\partial \mathscr{W}}{\partial \theta}\Delta\theta - B^1(\mathbf{x})\right]^2 \\ + \beta \sum_i \left[\mathbf{e}_i(\theta) + \frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\Delta\theta\right]^\mathsf{T} \left[\mathbf{e}_i(\theta) + \frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\Delta\theta\right] + \alpha \Delta\theta^\mathsf{T} \mathscr{I} \Delta\theta, \end{aligned} \tag{6.9}$$

where $\Delta\theta^\mathsf{T}\mathscr{I}\Delta\theta$ is a regularization term penalizing some special changes on $\Delta\theta$ controlled by $\mathscr{I}$ and a positive constant α. By setting $\mathscr{I} = diag([0,0,1,0,0,1,0,0])$, we penalize large changes on translation elements of $\Delta\theta$, so that frames are first aligned by warping them rather than translating them. Based on our experiments, this leads to more stable result. We initialize the algorithm by setting the sequences side by side (spatially) with the two possible layouts, and use the alignment result of the layout with lower final cost. The solution to Eqn. 6.9 is,

$$\Delta\theta = \mathbf{H}^{-1}\left(\sum_{\mathbf{x}\in\mathbb{O}(\theta)} \left[\nabla B^2 \frac{\partial \mathscr{W}}{\partial \theta}\right]^\mathsf{T} \left[B^1(\mathbf{x}) - B^2(\mathscr{W}(\mathbf{x};\theta))\right] - \beta \sum_i \left[\frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\right]^\mathsf{T} \mathbf{e}_i(\theta)\right), \tag{6.10}$$

in which

$$\mathbf{H} = \sum_{\mathbf{x}\in\mathbb{O}(\theta)} \left[\nabla B^2 \frac{\partial \mathscr{W}}{\partial \theta}\right]^\mathsf{T} \left[\nabla B^2 \frac{\partial \mathscr{W}}{\partial \theta}\right] + \beta \sum_i \left[\frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\right]^\mathsf{T} \left[\frac{\partial \mathbf{e}_i(\theta)}{\partial \theta}\right] + \alpha\mathscr{I}. \tag{6.11}$$

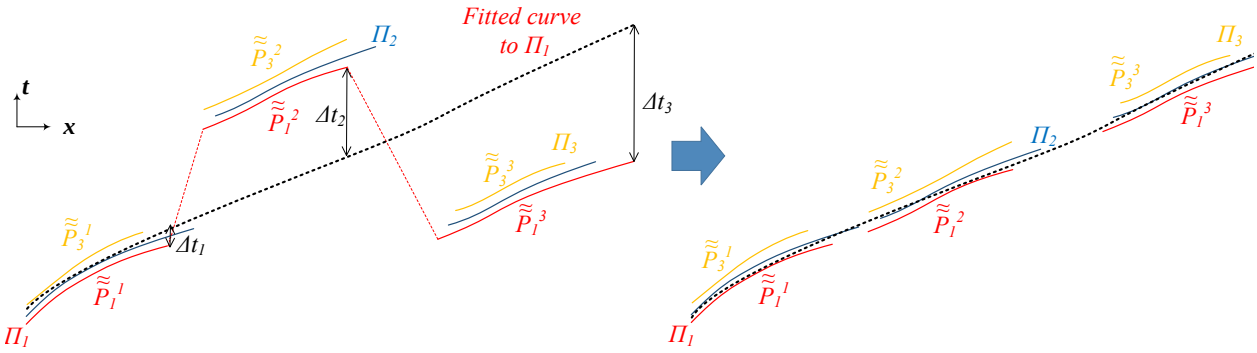


Figure 6.5: Trajectories, tracks, and fitted space-time curve to the tracks from 3 videos.

Here $\frac{\partial \mathscr{W}}{\partial \theta}$ is the Jacobian evaluated at $\mathbf{x}$. By the chain rule,

$$\frac{\partial \mathbf{e}_i(\theta)}{\partial \theta} = \frac{\partial \mathbf{e}_i(\theta)}{\partial \mathscr{W}} \frac{\partial \mathscr{W}}{\partial \theta} = [-f'(\tilde{\mathbf{w}}_i^{(x)}), \mathbf{1}] \frac{\partial \mathscr{W}}{\partial \theta}. \tag{6.12}$$

in which

$$\frac{\partial \mathbf{e}_i(\theta)}{\partial \mathscr{W}} = \frac{\partial \mathbf{e}_i(\theta)}{\partial \mathscr{W}_x, \mathscr{W}_y} = [-f'(\tilde{\mathbf{w}}_i^{(x)}), \mathbf{1}]. \tag{6.13}$$

6.3.3 Temporal alignment

NOS are assumed to have moving objects, without which the temporal alignment is neither necessary nor possible. Given moving objects and spatial alignment results, the temporal alignment of NOS amounts to estimating *w*hen an object will appear in FOV of another camera, after it moves out of the current FOV. If both cameras observe the object's motion at the same time, the problem is easier. For this purpose, we create *m*otion tracks as discussed in Sec. 6.3.2. Then, we *e*stimate the temporal offset between sequences such that trajectories from the identical object follow a continuous path in $\tilde{\tilde{x}} - \tilde{\tilde{t}}$ and $\tilde{\tilde{y}} - \tilde{\tilde{t}}$ coordinates, i.e., the motion tracks are smooth. Since not all trajectories are due to moving objects, we *f*ilter motion trajectories with non-smooth paths, before matching trajectories.

6.3.3.1 Estimation of temporal offset

Given the collection of tracks, the objective is to make each track a smooth curve, by shifting the temporal coordinates of the contributing trajectories appropriately (Fig. 6.5). For S sequences, $\tilde{\tilde{x}}-$ coordinate of trajectories forming the kth track is the vector $[\tilde{\tilde{\mathbf{x}}}_k^1, \tilde{\tilde{\mathbf{x}}}_k^2, ..., \tilde{\tilde{\mathbf{x}}}_k^S]^\intercal$. $\tilde{\tilde{y}}-$ coordinate of each track is defined similarly. We assume that by temporally shifting each sequence s for Δt_s, the sequences are temporally aligned. To estimate Δt_s, we fit a polynomial curve of degree m to time stamps versus $\tilde{\tilde{x}}$ and $\tilde{\tilde{y}}-$coordinates of *each* track independently and estimate the time shifts, in order to achieve the lowest curve fitting error. Here, we discuss only the $\tilde{\tilde{t}}-\tilde{\tilde{x}}$ curve, and the $\tilde{\tilde{t}}-\tilde{\tilde{y}}$ curve is similar.

We denote the trajectory coordinates of sequence s and all the power terms of the polynomial space-time curve as

$$\tilde{\tilde{\mathbf{x}}}_k^{s(m)} = [\mathbf{1}_k^s, \tilde{\tilde{\mathbf{x}}}_k^s, [\tilde{\tilde{\mathbf{x}}}_k^s]^2, \cdots, [\tilde{\tilde{\mathbf{x}}}_k^s]^m], \tag{6.14}$$

where $\mathbf{1}_k^s$ is a l_k^s-dim vector of all ones, and $[.]^m$ denotes an element-wise power operation. For the track k, all the required terms of the polynomial space-time curve are collected in a matrix $\tilde{\tilde{X}}_k$ of size $\sum_s l_k^s \times (m+1)$, and all the time stamps in a vector $\tilde{\tilde{T}}_k(\Delta\mathbf{t})$ of length $\sum_s l_k^s$,

$$\tilde{\tilde{X}}_k = \begin{bmatrix} \tilde{\tilde{\mathbf{x}}}_k^{1(m)} \\ \tilde{\tilde{\mathbf{x}}}_k^{2(m)} \\ \vdots \\ \tilde{\tilde{\mathbf{x}}}_k^{S(m)} \end{bmatrix}, \tilde{\tilde{T}}_k(\Delta\mathbf{t}) = \begin{bmatrix} \tilde{\mathbf{t}}_k^1 + \Delta t_1 \\ \tilde{\mathbf{t}}_k^2 + \Delta t_2 \\ \vdots \\ \tilde{\mathbf{t}}_k^S + \Delta t_S \end{bmatrix}. \tag{6.15}$$

We denote the coefficients of the kth polynomial curve fitting to the kth track as $\mathbf{c}_k = [c_q]; q \in \{0, ..., m\}$. We can estimate the coefficients by solving a linear system, $\arg\min_{\mathbf{c}_k} \parallel \tilde{\tilde{T}}_k(\Delta\mathbf{t}) - \tilde{\tilde{X}}_k \mathbf{c}_k \parallel$.

Since all tracks share the same $\Delta\mathbf{t}$, we can efficiently solve for all tracks jointly,

$$\mathbf{c}^*, \Delta\mathbf{t}^* = \arg\min_{\mathbf{c},\Delta\mathbf{t}} \| \tilde{\tilde{T}}(\Delta\mathbf{t}) - \tilde{\tilde{X}}\mathbf{c} \|, \tag{6.16}$$

in which

$$\tilde{\tilde{X}} = \begin{bmatrix} \tilde{\tilde{X}}_1 & 0 & \cdots & 0 \\ 0 & \tilde{\tilde{X}}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\tilde{X}}_K \end{bmatrix}, \tilde{\tilde{T}}(\Delta\mathbf{t}) = \begin{bmatrix} \tilde{\tilde{T}}_1(\Delta\mathbf{t}) \\ \tilde{\tilde{T}}_2(\Delta\mathbf{t}) \\ \vdots \\ \tilde{\tilde{T}}_K(\Delta\mathbf{t}) \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_K \end{bmatrix}. \tag{6.17}$$

Here, $\tilde{\tilde{X}}$ is a $N_K \times K(m+1)$ matrix where $N_K = \sum_k \sum_s l_k^s$ is the count of keypoints in all K tracks. We alternatively estimate $\mathbf{c}$ and $\Delta\mathbf{t}$, until the change in $\Delta\mathbf{t}$ is negligible. We first estimate $\mathbf{c}$, with fixed $\Delta\mathbf{t}$. Since $N_K \gg K(m+1)$, this linear system is over-constrained for $\mathbf{c}$. We solve $\mathbf{c}$ by Orthogonal-triangular decomposition, which is numerically more accurate than the pseudo inverse of $\tilde{\tilde{X}}$. Then, for a given $\mathbf{c}^*$, we set Δt_s as the average of residuals from the keypoints in trajectories belonging to sequence s,

$$\Delta t_s = -\frac{1}{N_s}(\tilde{\tilde{T}} - \tilde{\tilde{X}}\mathbf{c}^*)^\intercal \mathscr{I}_s, \tag{6.18}$$

where $\mathscr{I}_s$ is a binary indicator vector with an element equal to 1 if the corresponding row in $\tilde{\tilde{T}}$ comes from a trajectory in sequence s, and $N_s = \| \mathscr{I}_s \|_1$ is the count of such rows.

6.3.3.2 Motion trajectory filtering

As mentioned before, not all trajectories are resulted from object motion with a smooth path. In other words, some trajectories might be due to noise in keypoint locations while the camera moves. So, before matching trajectories across sequences and collecting them to a track, we filter out the

Sequence ID	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	S1	S2	S3	S4	S5
Camera baseline (m)	1	3	1	1	1	1	1	1	5	10	0	0	0	0	0
Temporal error (s)	0.13	0.07	0.07	0.13	0.07	0.07	0.10	0.03	0.07	0.07	0	0.03	0.07	0.03	0.07
Spatial error (pixel)	-	-	-	-	-	-	-	-	-	-	2	3	7	2	2

Table 6.1: Temporal and spatial alignment error in seconds and pixels, respectively, for real (R) and synthetic (S) sequences.

trajectories that cannot be well approximated with a smooth path, by fitting the order-m polynomial to the trajectory,

$$\mathbf{c}_k^{s*} = \arg\min_{\mathbf{c}_k^s} \| \tilde{\mathbf{t}}_k^s - \tilde{\tilde{\mathbf{x}}}_k^{s(m)} \mathbf{c}_k^s \|_2, \tag{6.19}$$

and thresholding the total fitting residual to remove non-smooth trajectories, i.e., $\frac{1}{l_k^s} \| \tilde{\mathbf{t}}_k^s - \tilde{\tilde{\mathbf{x}}}_k^{s(m)} \mathbf{c}_k^{s*} \|_1 < \tau_2$.

6.4 Experimental results

In this section, we present the experimental setup and both quantitative and qualitative results. Note that since NOS is a novel scenario for spatio-temporal alignment of sequences, there is no prior work for comparison. We set $\beta = 100$, $\alpha = 10^3$, $m = 3$ for the temporal curve fitting step, and $\tau_1 = 0.03$ and $\tau_2 = 0.15$ for trajectory filtering.

6.4.1 Dataset

Given that there is no public dataset in this new scenario, we collect a NOS dataset including ten real-world sequence sets, and five synthetic sequence sets. Real sets are captured by two or three people using handhold smartphones with the distance between the cameras, i.e. baseline, as shown in Tab. 6.1. Synthetic sets provide sequences for which the ground truth result are exactly known, and are created by taking a sequence and cropping out two spatio-temporal tubes from the 3D sequence volume. This emulates the case of independently panning cameras with almost identical optical centers. To simulate a freely panning camera and hand shake, the spatial region used for each tube at each frame has a fixed size of 640×360 pixels, but the region location has an additive

zero-mean Gaussian noise. Also, if the original video is stationary, the regions shift in $x-$direction to create a pan-like effect.

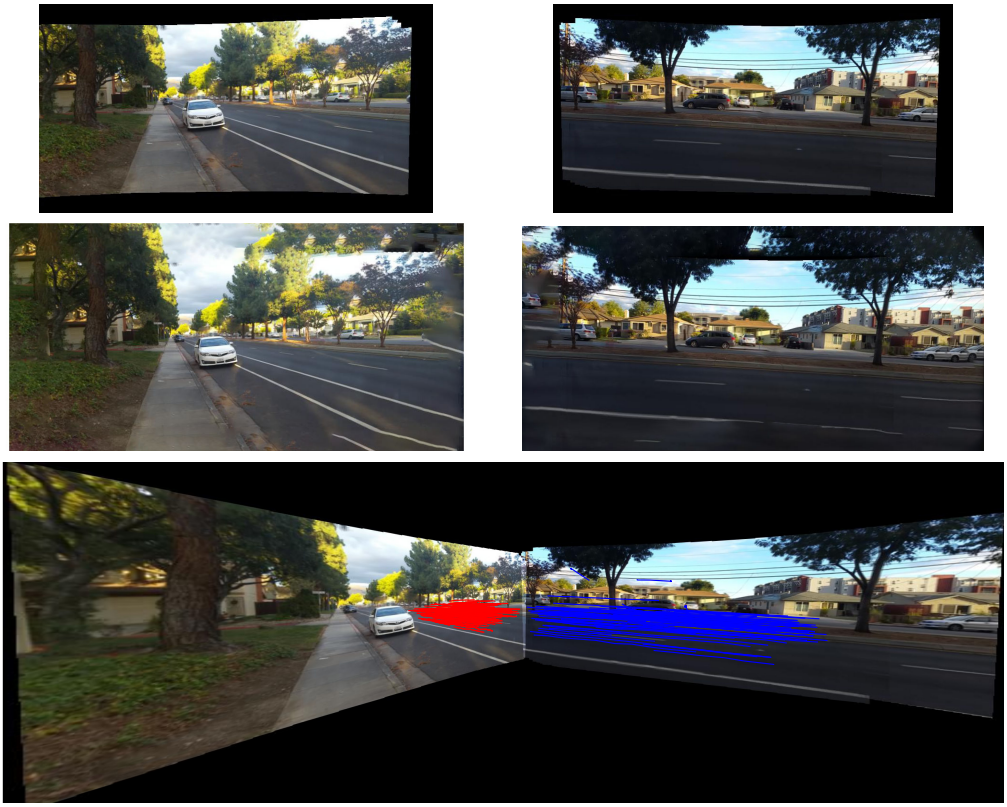


Figure 6.6: Spatial alignment of non-overlapping sequences. Top to bottom: reconstructed backgrounds of two sequences with negligible overlap, extrapolated backgrounds, and aligned background with trajectory of moving objects overlaid on the background.

6.4.2 Qualitative results

Figure 6.6 presents the reconstructed backgrounds along with image extrapolation results. Further, it is shown how the backgrounds are transformed so that moving object trajectories have smooth path.

Figure 6.7 shows the alignment results for three sets of real sequences with some overall spatial overlap. Similarly, Figure 6.9 shows the alignment results for three sets of real sequences with no/minimal overall spatial overlap. For each set, two or three sample frames with moving objects

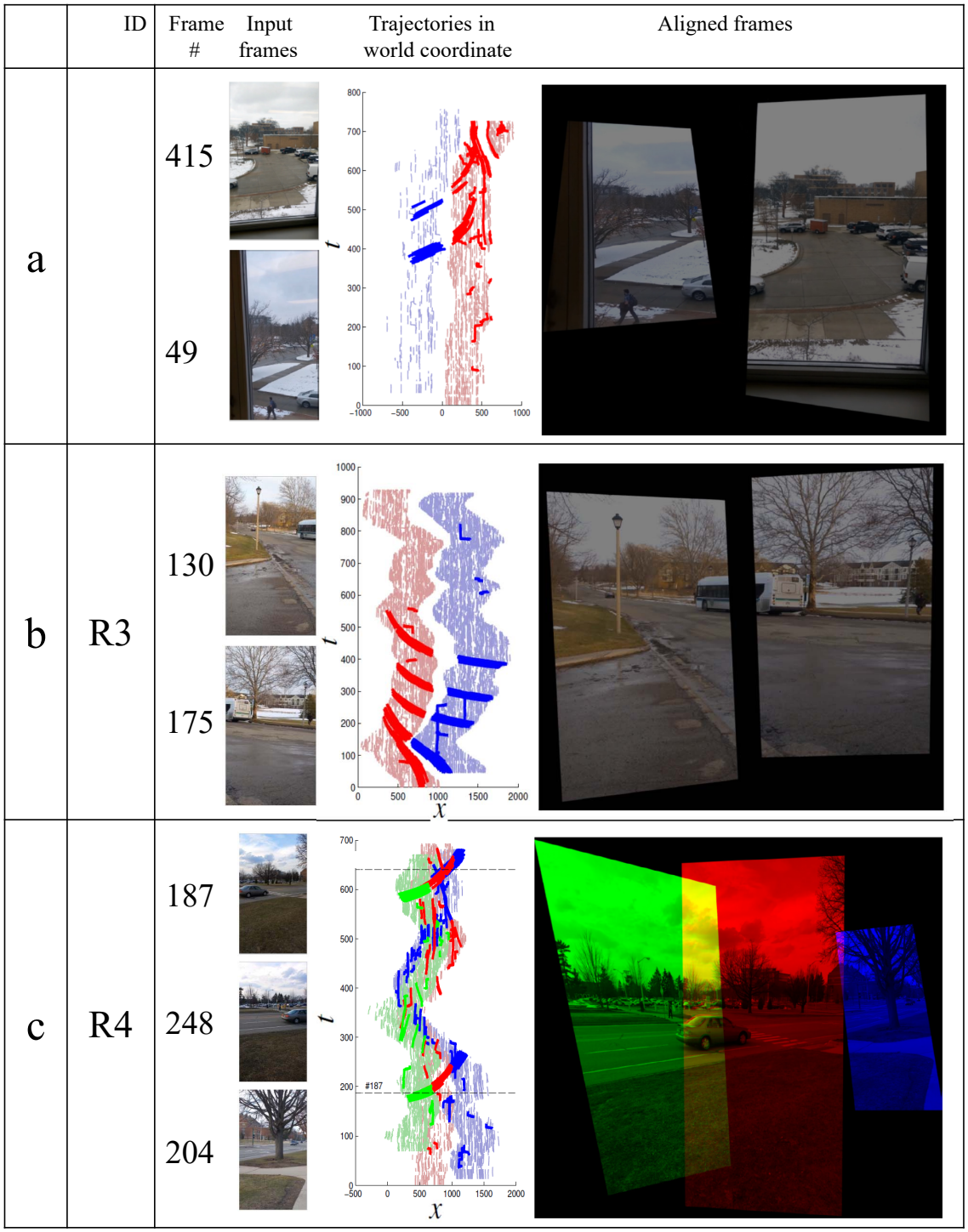


Figure 6.7: Each row shows spatio-temporal alignment results on a set of real NOS, with *some* overall spatial overlap. For each sequence, input frames at the estimated time shift and trajectories of moving objects in the world coordinate are shown. The input frames are transformed to the world coordinate to make a composite image via alpha blending.

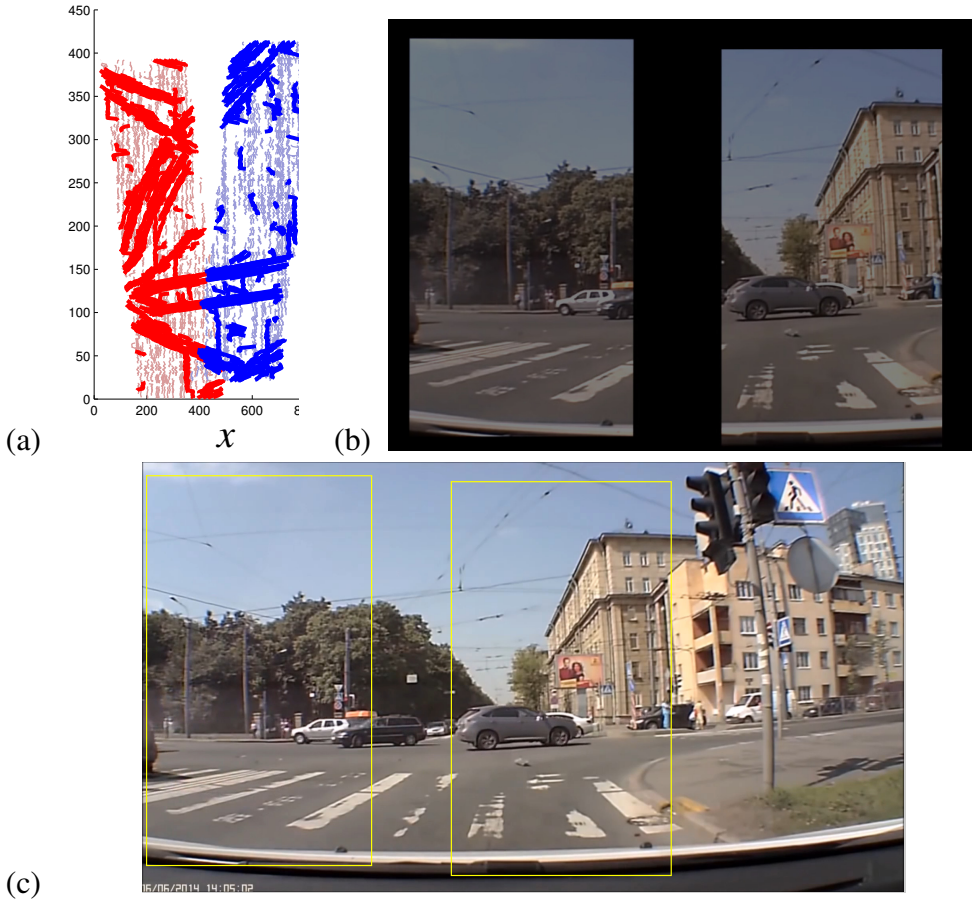


Figure 6.8: Results for two synthetic NOS from an accident footage (S2). (a) Trajectories of moving objects, (b) aligned input frames, (c) original frame where the synthetic frames are cropped.

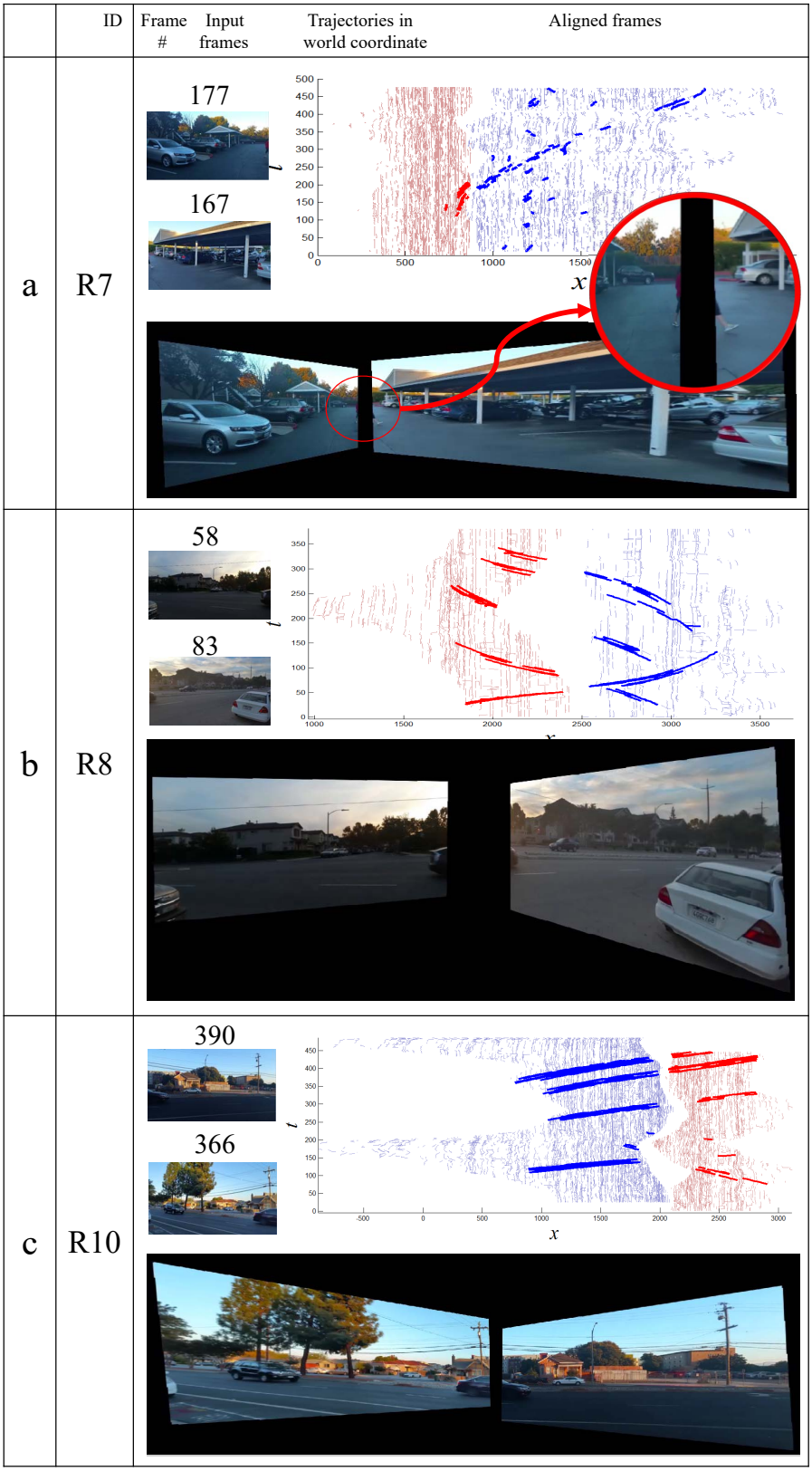


Figure 6.9: Each row shows spatio-temporal alignment results on a set of real NOS, with *no* overall spatial overlap. For each sequence, input frames at the estimated time shift and trajectories of moving objects in the world coordinate are shown. The input frames are transformed to the world coordinate to make a composite image via alpha blending.

are shown, at the time shift estimated by the proposed algorithm. Also, keypoint trajectories from both sequences in the world coordinate after spatio-temporal alignment are shown. Trajectories of moving objects have considerable extent in the $x-$direction, whereas trajectories of stationary objects are roughly parallel to $t-$axis. Finally, the two input frames are warped to the world coordinate to make a composite image. Although the input frames may not have direct overlap, perceived continuity of the scene and also relative location of the moving objects, demonstrate capabilities of the proposed algorithm and the application scenarios. Note that in all test sequences cameras move freely and independently, as shown by the range of trajectories in the world coordinate. For the case of Fig. 6.9(a), the sequences are non-overlapping, but only a person is tracked moving across the FOVs. Thus, as shown in this figure, spatial alignment has some error, which consequently affects the accuracy of the temporal alignment.

Figure 6.8 represents a synthetic set where two sequences are created from a video of a car accident. The two cropped frames after spatio-temporal alignment are shown in a composite image and for comparison, the corresponding frame from the original video is also shown, demonstrating the accuracy of the spatio-temporal alignment.

6.4.3 Quantitative results

To quantitatively evaluate the proposed algorithm, we compare the alignment errors with the ground truth. For the case of synthetic sets, the original video from which the synthetic sequences are cropped, provides the ground truth location of the center points of the cropped frames. We measure the spatial location error of each aligned frame w.r.t. the ground truth location and report sum of absolute errors in x and $y-$direction, averaged over the sum of the length of the sequences, as the spatial alignment error. Also, since we create the synthetic sequences, the ground truth time shift is known. For real sets, when the input frames do not have overlap, quantifying the spatial

error is not feasible. For quantification of temporal alignment, we manually align the sequences by relying on visual cues such as body pose, moving object location relative to background landmarks, and consistency of appearance of moving objects in the composite image. Table 6.1 provides the quantified temporal and spatial errors. As may be observed, temporal alignment works well even when the camera baseline distance increases, although the final consolidated result may suffer from parallax.

6.4.4 Computational cost

The main computational cost of the proposed algorithm comes from TRGMC. On average, for a video of 15-second long, we spend 450 seconds on TRGMC and background reconstruction, using a PC with an Intel i5-3470@3.2GHz CPU, and 8GB RAM. Spatial alignment is independent of sequence length and takes $\sim$162 seconds on average for NOS. Finally, temporal alignment takes about 13 seconds on average over the database.

6.4.5 Limitations

Violation of assumptions, especially existence of moving objects with a trajectory which spans FOVs of multiple cameras, results in alignment failures. Furthermore, when relying on non-rigid or articulated moving objects for alignment, many keypoints are not tracked long enough due to change of appearance, making alignment difficult. Also, in this case, matching trajectories among different sequences is less reliable and error prone. Since the algorithm is independent of the type of tracking involved, other tracking algorithms can be investigated in the future. Furthermore, alignment of non-overlapping background images suffers from ambiguity and is error prone, although the proposed algorithms makes use of available cues to conduct this task.

6.5 Conclusions

We proposed an algorithm for spatio-temporal alignment of sequences, referred to as non-overlapping sequences (NOS), from freely panning cameras for which FOVs of the cameras might not even observe some common regions over progression of time. This new scenario of video alignment is useful in reconstructing events, incidents, or crime scenes from multiple amateur-captured sequences, or creation of panoramic videos from cooperative users via handheld cameras without the need for tripods. The spatial alignment of our algorithm relies on reconstructing background for each sequence and aligning the backgrounds. When backgrounds are non-overlapping, the spatial alignment uses clues from smoothness of moving objects' paths and coherent appearance of background after image extrapolation. Smoothness of trajectory of moving objects is also utilized as a clue for temporal alignment. Our experiments demonstrate capabilities of the proposed method, despite the challenging scenario of NOS.

Chapter 7

Conclusion and Future Work

7.1 Conclusions and discussions

In this research, algorithms for global motion compensation are proposed to remove effect of camera motion and help with magnifying motions of interest in the videos from moving cameras. In this regard, we proposed two robust global motion compensation algorithms, namely RGMC and TRGMC. RGMC delivers reliable results in the presence of predominant foreground and textureless or blurry background, enabling its application to real-world unconstrained videos. By foreground suppression, RGMC is able to tolerate existence of large foreground and occlusion. Also, the proposed method successfully handles keypoint matching with a very low matching threshold, required for GMC in low texture background, or poorly illuminated scene. This is achieved by clustering motion vectors, and analyzing each cluster to identify matches pertaining to the background. Further, a novel homography verification model is proposed to support the RGMC. This model unifies keypoint matching error and consistency of the edges of images after transformation, and benefits from motion history gleaned from previous frames to ensure that in case of large foreground, foreground motion is not compensated instead of the camera motion affecting the background. Extensive experiments and comparison with ground truth obtained by manually matching the frames and baseline methods demonstrate the superiority of RGMC.

Furthermore, we proposed a temporally robust global motion compensation (TRGMC) algorithm by joint alignment (congealing) of frames, in contrast to the common sequential scheme.

This is done by dense connection of keypoints throughout all the frames and iteratively applying transformation on each frame such that the keypoints are spatially aligned. Despite complicated camera motions, TRGMC can remove the *intentional* camera motion, such as panning, as well as *unwanted* motion due to vibration on handheld cameras. Redundancy of information in the joint alignment of the stack of input images makes TRGMC capable of dealing with foreground motion and false matches without imposing further processing cost. Also, due to the joint alignment scheme, in TRGMC, existence of blurry, low texture, or poorly illuminated frames will not lead to total failure of GMC for all the upcoming frames. Beyond the robustness gained, experiments demonstrate that TRGMC outperforms existing GMC methods in terms of accuracy.

It is worth noting that the enabling assumption of any global motion compensation algorithm relying on homography estimation, is that the camera motion in the direction of the optical axis is negligible. For instance, TRGMC will not work properly on a video from a wearable camera of a pedestrian, since in the global coordinate the upcoming frames grow in size and cause computational and rendering problems. Similar to panorama images, the best results are achieved if the optical center of the camera has negligible movement during the capturing, making a homography-based approximation of camera motion appropriate. However, if the optical center moves in the perpendicular direction to the optical axis (e.g., a camera following a swimmer), TRGMC still works well, but rendering the results in the form of motion panorama will be degraded by parallax effect.

Subsequently, we investigated the challenging problem of spatio-temporal alignment of multiple video sequences, captured by freely panning handheld cameras. We identified and tackled a novel scenario of this problem referred to as Non-Overlapping Sequences (NOS) and proposed a solution which is only feasible given the reliable global motion compensation and background reconstruction via TRGMC. NOS are captured by multiple freely panning handheld cameras whose

field of views might even have no direct spatial overlap. However, over the progression of time, there are nearby regions in the scene that are observed by the cameras independently and probably at distinct time instants. In contrary to many existing works, this assumption is less restrictive than common region being observed by field of views of different cameras over progression of time, and obviously much less restrictive than the common requirement of direct spatial overlap between frames from different cameras. With the popularity of mobile capturing devices such as smartphones and wearable cameras, NOS rise when multiple cooperative users capture a dynamic scene, such as public events, to create a panoramic video or when it is desired to consolidates multiple footages of an incident or crime scenes into a single video. The proposed method makes it possible to better reconstruct the events or crime scenes captured by amateur users, and obtain a better understanding of the incident.

For this novel scenario, we first spatially align the sequences by reconstructing the background of each sequence using TRGMC algorithm and then registering these backgrounds across the sequences, even if the backgrounds are not overlapping. To this end, the reconstructed background images are first extrapolated. Then, a cost function is defined and minimized such that while extrapolated backgrounds are aligned well, trajectory of moving objects leaving field of view of one camera and entering field of view of another camera are spatially smooth. Given the spatial alignment, and assuming smoothness of trajectories of moving objects, such as cars or pedestrians, the sequences are temporally synchronized, such that the trajectories of moving objects across sequences are consistent with the prediction of when a moving object leaving the field of view of a camera, would appear in the field of view of another camera.

Finally, to develop algorithms for analyzing user-generated videos, unconstrained and representative datasets are of great significance. For this purpose, we collected a dataset of *Sports Videos in the Wild (SVW)*, consisting of videos captured by users of a leading sports training smart-

phone app (Coach's Eye®) while practicing a sport or watching a game. The dataset contains 4000+ videos selected by reviewing $\sim$85,000 videos and consists of 30 sports categories and 44 actions. Videos of sports practice, which frequently happens outside the typical sports field, have huge intra-class variations due to background clutter, unrepresentative environment, existence of different training equipment and most importantly, imperfect actions. On the other hand, using smartphones for video capturing by ordinary people, in comparison to videos captured by professional crew for broadcasting, leads to challenges due to camera vibration and motion, occlusion, view point variation, and poor illumination. Given various manual labels, this dataset can be used for a wide range of computer vision applications, such as action recognition, action detection, genre categorization, and spatio-temporal alignment. On the sport genre categorization problem, we also design the evaluation protocol and evaluate three different methods to provide baselines for future works.

7.2 Future work

TRGMC is a powerful and useful algorithm which can be an essential module in utilizing many existing algorithms designed for static cameras, for handheld and moving cameras. However, many applications require high processing rate. In future, we will further investigate fast algorithms for global motion compensation, and will investigate how a fast GMC algorithm benefits other computer vision applications.

7.2.1 Speed-up TRGMC

One potential direction is to keep the alignment framework to be joint, however not as dense as TRGMC. More specifically, TRGMC has $O(n^2)$ complexity due to dense linking of all the n frames

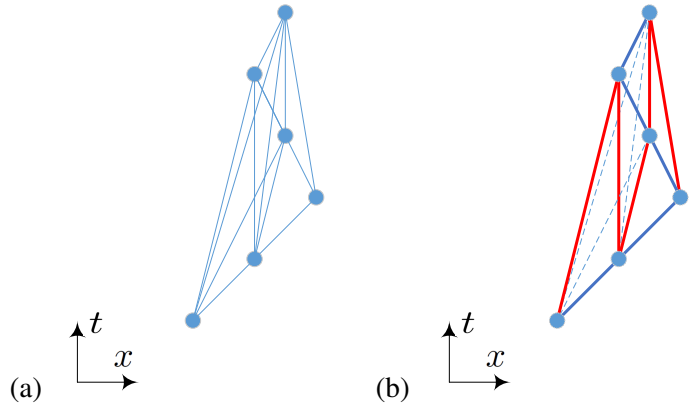


Figure 7.1: Comparison of the links made by (a) TRGMC with (b) the links which might be made via the proposed speed-up scheme. Chosen links by minimum spanning tree are shown in red and links between adjacent frames are shown in blue.

in the stack. Decreasing number of links such that each frame is only linked to a constant number of other frames, for instance the two temporally neighboring frame and a few frames with longer temporal distance to enforce temporal consistency, will improve the complexity to $O(n)$. Selection of the best linking choice can be casted to finding a minimum spanning tree in a weighted graph $G = (V, E)$. Each frames in the alignment stack is a node of the graph G, and each edge of the graph connecting node i to j has a weight w_{ij} showing suitability of the matching pair (i, j) to be included in the joint alignment. To enforce long term robustness of the alignment, including frames with larger temporal distance is preferred. Thus, if the normalized temporal difference of frames i and j is denoted as t_{ij}, we define $w_{ij} \propto (1 - t_{ij})$. Furthermore, frames with largest spatial overlap are better matched with each other, and are less affected with view angle change. So, after initialization step of TRGMC which results to finding approximate translation between the frames, it is possible to collect the sequential translations to obtain the translation between arbitrary frames i and j and find the normalized overlap area between these frames, denoted as a_{ij}. Thus, the weights might also be proportional to the overlap area, as $w_{ij} \propto (1 - t_{ij})(1 - a_{ij})$. Figure 7.1 compares the links made by TRGMC with the links which might be made via the proposed speed-up scheme. Beyond the links of the minimum spanning tree, it is also beneficial to link the sequential frames to enforce

both temporally long term and local consistencies.

7.2.2 Joint alignment and outlier rejection

In each keypoint matching stage between two given frames, TRGMC utilizes an outlier rejection via Ma *et al*let@tokeneonedotmethod of vector field consensus [70]. This operation is repeated frequently, and takes ~ 10 the keypoint matching step itself. So, one potential direction to improve the efficiency is by embedding the outlier detection and rejection in the joint alignment formulation via latent variables or appropriate weighting of each link. For instance, for each iteration of the algorithm in which homographies are updated to decrease the alignment error, nonconforming links which do not connect background keypoints, and thus have increased error with the homography update, may be identified.

APPENDICES

Appendix A

Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis

Considering the enormous creation rate of user-generated videos on websites like YouTube, there is an immediate need for automatic categorization, recognition and analysis of videos. To develop algorithms for analyzing user-generated videos, unconstrained and representative datasets are of great significance. For this purpose, we collected a dataset of *Sports Videos in the Wild (SVW)*, consisting of videos captured by users of a leading sports training smartphone app (Coach's Eye®) while practicing a sport or watching a game. The dataset contains 4000+ videos selected by reviewing $\sim$85,000 videos and consists of 30 sports categories and 44 actions. Videos of sports practice, which frequently happens outside the typical sports field, have huge intra-class variations due to background clutter, unrepresentative environment, existence of different training equipment and most importantly, imperfect actions. On the other hand, using smartphones for video capturing by ordinary people, in comparison to videos captured by professional crew for broadcasting, leads to challenges due to camera vibration and motion, occlusion, view point variation, and poor illumination. Given various manual labels, this dataset can be used for a wide range of computer vision applications, such as action recognition, action detection, genre categorization, and spatio-temporal alignment. On the sport genre categorization problem, we design the evaluation protocol and evaluate three different methods to provide baselines for future works.

A.1 Introduction

The amount of digital videos being created is increasing exponentially, e.g., YouTube has reached the upload rate of 100 hours of video per minute. A great deal of this growth is due to the tremendous popularity of smartphones and ubiquitous Internet access. This means that *a*mateur-user-generated videos form the new trend in content generation. Thus, there is an immediate need for robust algorithms to automatically analyze and retrieve videos.

Many computer vision problems are data-driven and the existence of representative and realistic datasets are necessary for developing robust algorithms. Therefore, there has been a trend from research on *c*ontrolled datasets toward *u*nconstrained datasets. For instance, recent face recognition research focuses on datasets like Labeled Faces in the Wild (LFW) [42] rather than controlled datasets like FERET [78]. Similarly, for human action recognition, datasets with less controlled videos, e.g., Hollywood2 [71], HMDB [49] and UCF101 [100], are gaining popularity, compared with staged datasets like KTH [93] or Weizmann [13]. While these datasets ([49, 71, 100]) are from YouTube videos and movies and thus have unconstrained environment and actions relative to staged datasets, many of the videos are captured *p*rofessionally. Therefore, in aspects like camera vibration, view angle variation, and illumination, they are bound to common practices of filmmaking. On the other hand, specifically for sports videos, most videos in public datasets are representative of successful completion of the actions that may not truly reflect the highly complex and diverse real-world sports activities. Finally, for sports videos, due to strong correlation of background and the actions in existing datasets, the state-of-the-art performance on genre categorization is very high.

Given the explosion of user-generated videos and the lack of real-world datasets for the research community, we present a highly unconstrained dataset of sports videos, called *Sport Videos in the*

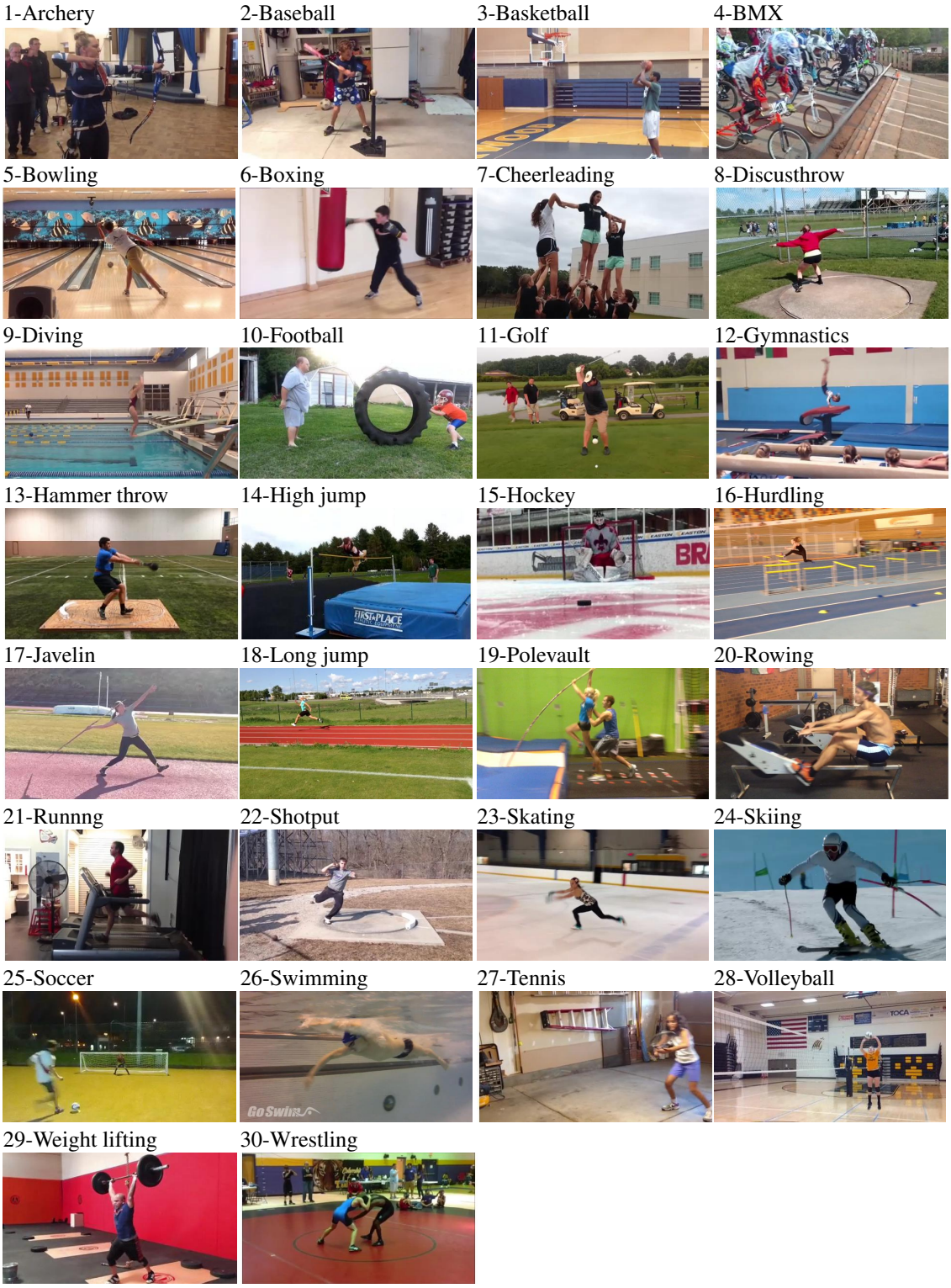


Figure A.1: Sample frames from all 30 sports categories of SVW.

Wild (SVW). SVW is comprised of videos captured *solely* with smartphones by users of Coach's Eye® smartphone app, a leading app for sports training developed by TechSmith corporation. The app allows users to conveniently capture videos whenever they practice a sport or watch a game. Fig. A.1 shows sample frames from different categories of SVW. Being captured by smartphone and by ordinary people, along with the fact that many videos are of practices of amateurs, not professional athletes, makes SVW the most unconstrained dataset of sports and action videos.

SVW is annotated to serve for multiple purposes. For *a*ction recognition, videos are labeled with 44 different actions and timespan of each action. To make the dataset appropriate for *a*ction detection, videos are not trimmed around each action, instead, time stamps are provided. In addition, more than 50% of the videos are annotated with spatio-temporal bounding boxes around each and every action in the video. For *s*ports genre categorization, each video is labeled with generic name of the sport being practiced, resulting in 30 sports categories.

For the sport genre categorization problem, we design the evaluation protocol and compare the performance of three algorithms on the proposed dataset as baselines for future research. First, the performance of the state-of-the-art motion-based dense trajectories algorithm [114] is reported. Second, purely context-based algorithm of describing videos with SIFT features [66] is presented. Finally, experiments using a motion-assisted context-based algorithm are conducted. All data, including the dataset, labels, evaluation protocol, and experimental results, will be *p*ublicly available to the research community for future research[1].

[1] http://cvlab.cse.msu.edu/svw-download.html

A.2 Related Work

Table 1 summarizes different aspects of most popular action recognition (AR) datasets. To the best of our knowledge, there is no publicly available dataset for sports genre categorization. Among existing datasets, HMDB [49] and UCF101 [100] are the most challenging ones in terms of having unconstrained videos.

KTH [93] and Weizmann [13] datasets contain simple actions and their AR accuracies are reported to be above 90% [49]. IXMAS [121] contains staged actions captured by 5 calibrated cameras, where an AR accuracy of 93.5% is reported in [116].

UCF Sports [102] and Olympic [75] are the only datasets that cover just sports activities. While the environment is not controlled, the videos are *c*aptured by professional crew, the actions are *p*erformed by professional athletes, and the background is restricted to official sports fields. As noted in [49], the actions in these datasets are highly distinguishable from shape cues alone. For Olympic, an accuracy of 91.1% is reported in [117]. Having limited number of categories and distinct activities in each category, a recognition rate of 98% is reported in [49] for UCF Sports using the information from static joint locations alone.

Hollywood2 [71] dataset is gathered from 69 movies and is labeled for both action recognition and scene understanding. Being selected from movies, it contains unconstrained environment and actions while benefiting from professional capturing. Its main restrictions include the limited number of actions and the fact that clips extracted from the same movie share similar scenes. In [117], an AR accuracy of 64.3% is reported for Hollywood2.

For UCF50 [83], Kuehne et al. [49] suggest that low-level features are as predictive as mid-level features and Wang et al. [117] report a 91.2% AR accuracy. As an extension of UCF50, UCF101 has 101 categories and is the largest AR dataset available [100]. Being collected from

YouTube, the actions are fairly unconstrained, but no comment can be made about the capturing process. Karpathy et al. [46] report a 66% AR accuracy for UCF101 (80% for the sports group). Probably due to the low resolution of source videos, all clips are normalized to the relatively low resolution of 320×240. At the mean clip length of 7.2 second, UCF101 is fairly short compared to SVW, making it less suitable for action detection problems.

HMDB [49] is collected by looking for non-ambiguous human actions in Internet videos and movies. As a quality standard, selection of videos has been constrained to having a single action per clip and 40% of the clips are not affected by camera motion. The dataset is prepared in two versions of original videos and stabilized videos and good performance is reported for stabilization. In [117], an accuracy of 57.2% is reported for HMDB. HMDB is very challenging due to not only the unconstrainedness of the dataset, but also having multiple shots in a single clip, where both factors contribute to the low AR accuracy.

Although existing datasets have some levels of unconstrained actions and environment, there is still more complexity in real-world videos that need to be represented in research datasets. Specifically for sports videos, current datasets do not provide highly unconstrained conditions. For UCF101, one of the most challenging datasets, sport videos achieve the highest recognition rate ([46, 100]) among different types of videos. This is claimed to be due to distinctiveness of sports motions and less cluttered background in official sports field than other types of actions, which does not hold for sports in the wild. Considering high performances reported for UCF Sports, Olympic, and sports groups of UCF101, SVW specifically fills the research gap for analyzing *c*hallenging sports videos. On the other hand, uploading a video to YouTube implies that the action of desire has been successfully performed and completed in the video. But for a completely unconstrained video, there might be failure cases (e.g., batting practice). In addition, unlike UCF101 or Hollywood2, in SVW no two videos are trimmed from a single footage captured by users, which keeps

Table A.1: Comparison of multiple datasets for action recognition (AR), scene understanding (SU), and genre categorization (GC).

Dataset	Purpose	Categ. #	Clip #	Avg. length	Unconst. actions	Unconst. capturing	Camera vibration	Orient.	Sources
KTH [93]	AR	6	100	NA	No	No	No	Lands,	Staged
Weizmann [13]	AR	9	9	NA	No	No	No	Lands,	Staged
IXMAS [121]	AR	11	30	NA	No	No	No	Lands,	Staged
UCF Sports [102]	AR	9	14+	NA	Yes	No	No	Lands,	Broadcast TV
Olympic [75]	AR	16	50	NA	Yes	No	No	Lands,	YouTube
Hollywood2 [71]	AR SU	12 10	61+ 62+	NA	Yes	No	No	Lands,	Movies
UCF50 [83]	AR	50	100+	NA	Yes	No	Slight	Lands,	YouTube
HMDB [49]	AR	51	101+	NA	Yes	No	Slight	Lands,	Movies & Internet
UCF101 [100]	AR	101	100+	7.2	Yes	No	Slight	Lands,	YouTube
SVW	AR GC	44 30	50+ 110+	11.6	Yes	Yes	Yes	Lands, & Port.	Smartphone

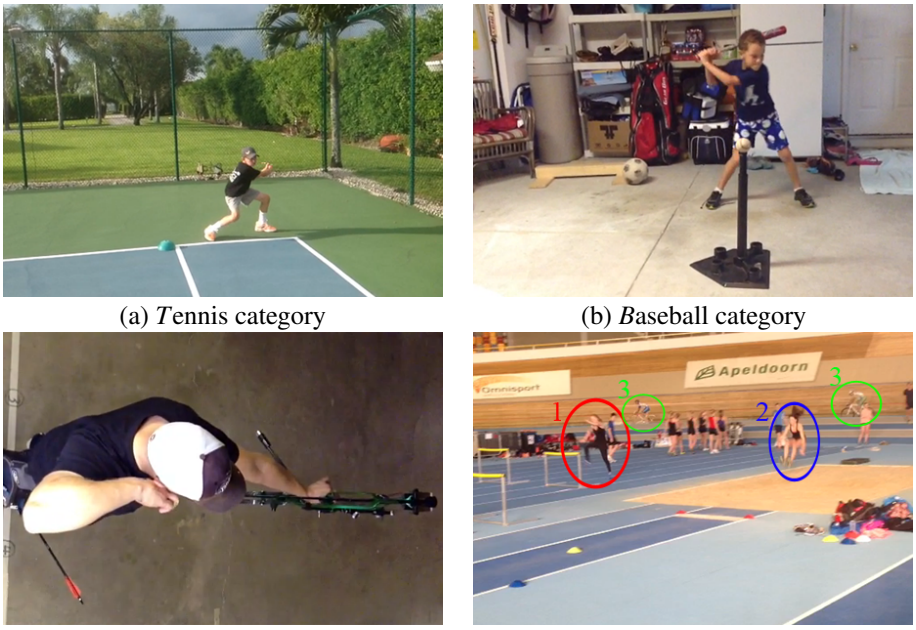
the variance of the actions, environment, and shooting conditions in the dataset as high as possible. Furthermore, due to highly unconstrained environment and illumination condition as well as a high rate of scene occlusion by people, video stabilization of SVW is very challenging and our experiments show a high failure rate of stabilization using the common RANSAC algorithm [35]. Finally, the video resolution and clip length of SVW are larger than all the current datasets, and SVW includes both lanscape and portrait orientaion of videos.

A.3 Sports Videos in the Wild (SVW) Dataset

A.3.1 Dataset details and statistics

Dataset collection SVW is selected from the videos captured by ordinary users of Coach's Eye® smartphone-based sports app developed by TechSmith corporation, when users practice a sport or watch a game. The users can review the videos and compare them with those of coaches or professional athletes side by side. A user may also upload the videos to the app server for other users to review and comment on his sports training progress. At the time of writing this paper, an average of 4 videos per minute are being uploaded to the app server by users, and among $700,000$ uploaded videos, users have marked $\sim$418,000 as publicly usable. Due to the highly nonuniform distribution of sports categories, $85,000$ videos from the public set have been reviewed and labeled to collect enough videos for 30 sports category and 44 action categories with at least 110 and 50 videos per category, respectively.

Challenges of SVW Compared to broadcasting videos, sports videos in the wild have many unique challenges for visual analysis, due to both the imperfect practices of *a*mateur players and unprofessional capturing by *a*mateur users. Firstly, the static image context is less discriminative for categorization. For example, in a video of tennis forehand drill (Fig. A.2 (a)), no assumption can be made about existence of the racquet (and in some cases the tennis court). The only reliable clue may be the unique motion characteristics of the hands. Secondly, in these videos, existence of training equipment is more likely than the broadcasting videos (Fig. A.2 (b)). On the other hand, cluttered backgrounds as well as common environments also cause difficulties in unconstrained sports videos; There are many SVW videos that the sport is practiced inside the house, in the garage, or in the backyard (Fig. A.2 (b)). Thirdly, unprofessional capturing by amateur users intro-



(a) *T*ennis category (b) *B*aseball category

(c) *A*rchery category (d) *H*urdling category

Figure A.2: SVW challenges: (a) Related equipment does not exist, (b) Background is cluttered and uncorrelated with the sport, (c) Uncommon camera angles increase the intra-class variations, (d) Multiple sports co-exist (1: Hurdling, 2: Long jump, 3: Cycling).

duces additional challenges like extreme camera vibration, improper camera movement, occlusion from audience, judges and fences due to improper camera location, and uncommon view angles (Fig. A.2 (c)). Finally, for amateur videos, it is more probable to have multiple activities in a single video (Fig. A.2 (d)).

It is important to note that unlike other action recognition datasets that are recently widely used, multiple actions defined in SVW may come from a *single* sport (see Fig. A.3). In other words, while the environment is quite similar for these subsets of actions, movements are completely different. This introduces further challenges in visual analysis of sports videos in the wild for the purpose of action recognition. On the other hand, this arises difficulties for genre categorization. Each sport category has huge intra-class variation due to containing multiple actions that can appear at any timespan of the whole video length.

Figure A.3: Annotated actions categories ([343, 359, Forearm], [380, 400, Set], [438, 454, Spike]) within a video from *V*olleyball genre category. Since distinct actions from the same sport genre may share a common field, visual appearance alone is not enough for action recognition in SVW.

Dataset labelling Videos are manually labeled in a two-round scheme. First, for each clip, 6 frames uniformly sampled across the video length constitute a montage, which is saved as an image. A GUI equipped with a button for each category shows the saved montage and records the pressed button from the labeler. In the next round, all labeled clips are reviewed one by one. Clips over 1-minute long are trimmed to loosely cover representative motions, but not precisely around the action of interest so that the dataset is also suitable for action detection. To prepare SVW for action recognition, at least 50% of the videos are reviewed closely to annotate *a*ll pre-defined actions within a clip and their corresponding timespans. The same videos are also annotated with bounding boxes around each action in the video for *a*ction detection application. Fig. A.3 represents how different actions within a clip are annotated with the label and time stamps.

For each video clip, we also label various meta tags. Fig. A.4 represents the distribution of the number of participents in videos, commonality/uniqueness of the action environment, and the camera view angles for 30 sports categories. Meta tags reveal that 19% of SVW videos are affected by considerable camera vibration and the videos of three categories have the highest rates of training equipment usage, Running (9%), Weitgh lifting (9%), and Boxing (4%). Multipe activities in a single video are more common for categories such as Hurdling, High jump, Running, Weight lifting, and Diving.

Spatial resolution normalization The resolution of the original videos varies from 480×272 to 1280×720 (irrespective of video orientation) with 640×360 being the most common size. Since for some analysis algorithms variation of video sizes might result in the confusion of scene scales, a normalized version of the dataset is provided along with the original one. Having both landscape and portrait orientations in the dataset, normalized clips have the maximum size (width or height) of 480 pixels.

Evaluation protocol In line with UCF101 and HMDB, three splits of 70% training and 30% testing are generated for the genre categorization application of SVW. We designate the splits by aiming to evenly distribute different actions, camera view angles, and field characteristics over the splits. The genre categorization accuracy is used as the performance metric and is defined as the fraction of testing videos whose genres are correctly classified.

A.3.2 Potential applications of SVW

Action recognition Due to the huge number of video content available online and the desire to content understanding, a great deal of effort has been focused on the problem of action recognition

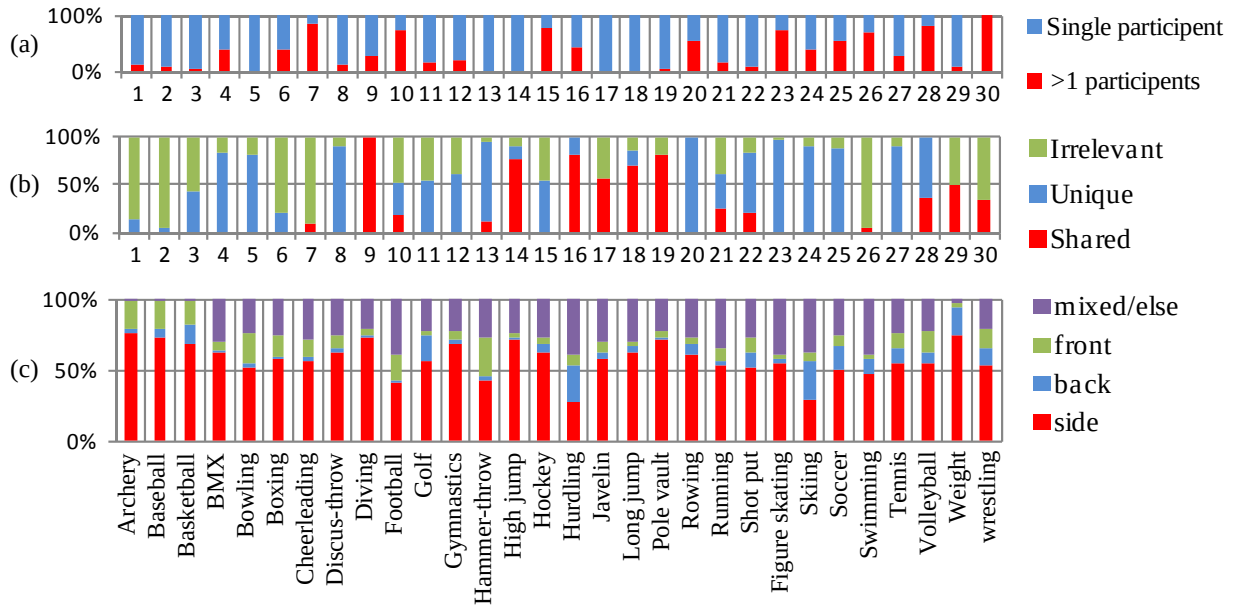


Figure A.4: Distribution of (a) number of participents in videos, (b) aspects of the action field and (c) camera views angles, in 30 categories. *I*rrelevant field is a field that from its appearance, the sports category cannot be deduced (e.g., practicing in the backyard). *S*hared field refers to the condition in which from just field appearance, more than one sports category might be inferred (e.g., track and field sports). *U*nique field is the one that just from field context, the corresponding sports category can be conjectured (e.g., Bowling tracks).

from videos [45, 51, 52, 71, 81, 122]. Inherently, for sports videos, action recognition is a subset of the genre categorization problem, i.e., for the former, labels for a single action are available but for the latter, a group of different actions within each sport are all labeled with the genre of the sport, resulting in higher intra-class variations.

Action detection Although there has been great emphasis on action recognition, the action detection problem has not been extensively studied. Action detection by itself and as part of *recognition by detection* systems [85] is an important problem to be tackled. Especially, in real-world videos, actions of interest may cover a relatively short period of a video and it is important to be able to detect these actions. Existing approaches use rather simple datasets with short videos [47, 125] or proprietary datasets [25]. SVW enables researchers to push the limit of action detection toward more realistic videos.

Genre categorization Sports genre categorization is vastly studied for broadcasting TV channels videos [29, 73, 118, 126, 129, 130]. In these works, it is assumed that sports occur in sports arena (implicitly assuming the existence of specific equipment and field lining) and are captured by professional TV broadcasting crew. Low-level features like color, motion, and histogram of edge directions are used for categorization. Our experiments show that this type of approaches does not perform well on sports in the wild. On the other hand, in [87], authors report superior performance of the dense trajectories method [114] for genre categorization of unconstrained proprietary videos. This paper aims to provide a dataset of such videos. Well-known sports-only datasets of UCF Sports [102] and Olympic [75], include specific actions not generic sports categories, and have been reported to achieve $\sim$90% accuracy ([116, 117]) (The method in [116] achieves $\sim$62% accuracy on SVW.). Thus, a challenging video dataset for this application is highly desirable.

Spatio-temporal alignment Given two video sequences of the same action, spatio-temporal alignment is defined as finding the spatial and temporal coordinate transformation that maps the actions of interest in one video to those of the other [4, 111]. For the case of sports videos, spatio-temporal alignment of actions enables effective comparison of actions performed by different people. This is specifically useful for the purposes of sports training and grading. Furthermore, the *j*oint alignment of multiple videos, from either one user over time or a diverse set of users, allows us to study the temporal evolving and inter-subject variations of a particular action, which are novel research problems by themselves. Having unconstrained videos where action of interest may happen at any temporal segment of the video, SVW serves as a realistic and challenging dataset for the alignment problem.

Table A.2: Performances (genre categorization accuracy) of different baseline algorithms on SVW.

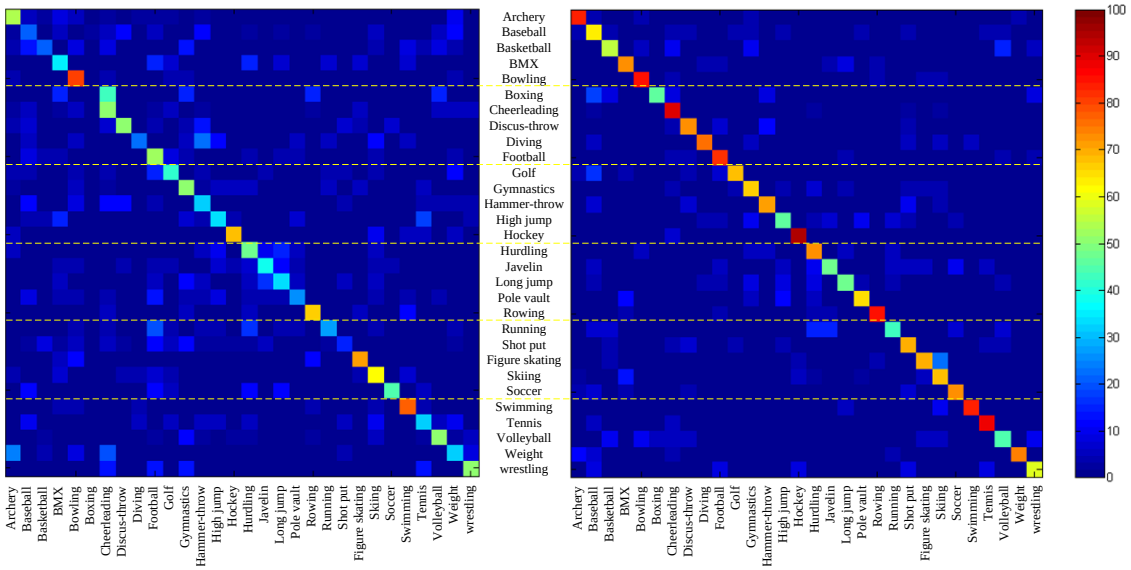
Method	Motion-based	Context-based	Motion-assisted context
Performance	61.53%	37.08%	39.13%

A.4 Baseline Experiments

In this section, we present the performances of three different algorithms for the genre categorization problem on SVW. The first algorithm summarizes features extracted from dense trajectories [114] using the widely used Bag of Words (BoW) approach [20]. The second algorithm analyzes the context of video frames using the BoW on the SIFT features [66]. The third one, a motion-assisted context-based algorithm, segments the moving and stationary pixels using trajectory information and then analyzes the appearance of these two groups of pixels separately. To the interest of computational cost and memory, for all methods, a two-level bottom-up codebook generation scheme is used [130]. At the first layer, for each class, a set of codewords are generated using K-means clustering. At the second layer, codewords of all classes are aggregated and by another round of clustering, the final codewords are obtained. We use Support Vector Machine (SVM) as the classifier for all the algorithms. Table A.2 summarizes the genre categorization accuracies of the baseline algorithms. Unlike UCF Sports, which is conjectured in [49] to be equally predictable using contextual or motion information due to the fact that many sports in UCF Sports are location-specific, and similarly Olympic dataset, sports videos in the wild are better recognized using motion features due to existence of many practice videos in environments uncorrelated with the activities. However, the accuracy achieved by motion-based algorithm is relatively low due to miscellaneous aforementioned challenges of SVW. Fig. A.5 represents confusion matrices of context-based and motion-based algorithms. The contrast of off-diagonal elements indicates the potential benefits of fusing these two algorithms. More detail on all three algorithms follows.

Table A.3: Performance of different combinations of trajectory descriptors on SVW.

Descriptors	**s**	HOG	HOG + **s**	MBH	MBH + HOG	MBH + HOG + **s**
Performance	44.65%	56.45%	60.43%	58.12%	60.69%	61.53%



(a) (b)

Figure A.5: Confusion matrices of (a) context-based and (b) motion-based categorization algorithms.

A.4.1 Motion-based algorithm

For motion-based algorithm, the state-of-the-art approach of dense trajectories is used [114]. The BoW approach on top of dense trajectory based features has been reported to outperform those of space-time interest points on various datasets [116]. This approach consists of three main steps: video stabilization, trajectory extraction and description, and BoW representation of trajectory information. We use implementations in [117] for the second step.

A.4.1.1 Video stabilization

Frame by frame motion stabilization is achieved by matching interest points on consecutive frames and applying RANSAC [35] to obtain the affine transformation between frames. Due to issues

such as poor illumination, moving subjects and audience, and uniform or non-rigid backgrounds (like water), the failure rate of video stabilization is quite high, which deteriorates the overall performance of the motion-based algorithm.

A.4.1.2 Dense trajectories

As proposed in [114], dense trajectories are extracted at multiple spatial scales. Each point $p_t = (x_t, y_t)$ at frame t is tracked to the next frame $t+1$ by performing the median filtering in a dense optical flow field $\mathbf{W} = (u_t, v_t)$, $p_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (K * \mathbf{W})|_{(\overline{x_t}, \overline{y_t})}$, where K is the median filtering kernel and $(\overline{x_t}, \overline{y_t})$ is the rounded position of (x_t, y_t). Trajectories are started from the sample points on a grid spaced by W pixels (set to 5). The length of each trajectory is limited to L (set to 15), and after reaching this length, the trajectory is removed from the tracking process and new sample points are tracked.

A.4.1.3 Trajectory descriptors

The shape of the trajectories can be used as a representative feature, especially for sports analysis. In [114], the displacements of trajectory, $\Delta p_t = (x_{t+1} - x_t, y_{t+1} - y_t)$, over L consecutive frames are concatenated to be a vector, $\hat{\mathbf{s}} = [\Delta p_t, ..., \Delta p_{t+L-1}]$, which is further normalized to be a trajectory descriptor $\mathbf{s} = \hat{\mathbf{s}} / \sum_{j=t}^{t+L-1} \|\Delta p_j\|$. Similar to [114], the video volume of a neighborhood of each trajectory is aligned and the resultant volume is described by using the Motion Boundary Histogram (MBH) [23] and the Histogram of Oriented Gradients (HOG) [22]. Table A.3 shows the performance of different combinations of descriptors. The highest accuracy of 61.53% is achieved by combining MBH, HOG, and $\mathbf{s}$ descriptors.

A.4.1.4 Context-based algorithm

We follow the algorithm in [130] for analyzing the videos using only the static contextual information. In this algorithm, we sample one frame per one-second length of video and use the BoW representation of SIFT descriptors for categorization. For this algorithm, no inter-frame information is utilized, thus video stabilization is not required. For dictionary learning, we use 10 videos per category. In our experiments, the size of codebook is set to 4000. As represented in Table A.2, the categorization accuracy of 37.08% is achieved using this method.

A.4.1.5 Motion-assisted context algorithm

Along with the idea in [83], we augment the context-based method with the information of moving and stationary pixels. This can be loosely considered as foreground-background segmentation using motion information. For this purpose, the mean position of trajectories of the stabilized videos, for which the standard deviation of the trajectory points is beyond a threshold, is considered as a moving point. The decision about a moving point at a certain frame is propagated to 15 frames before and after the frame on which the trajectory ends. Having groups of moving and stationary pixels ready, SIFT descriptors and BoW representation are calculated for them separately and the resulting histograms are concatenated to represent the video. This algorithm achieves an accuracy of 39.13% (Table A.2), which is slightly better than the algorithm using context information only.

A.4.1.6 Discussion

Comparing $\sim$62% accuracy achieved by motion-based algorithm of dense trajectories on SVW with $\sim$91% accuracy obtained by applying the same method to both UCF50 and Olympic Sports datasets [117], demonstrates that SVW is a very challenging sports video dataset. In addition, comparing accuracy obtained by applying motion-based and context-based algorithms ($\sim$62% vs

$\sim$39%) reveals that in SVW, motion is the main cue for categorization and action recognition. While the motion-assisted context based algorithm results in $\sim$39% accuracy for SVW, as reported in [83], similar method achieves accuracy of $\sim$67% for UCF50. This essentially suggests that background and equipment appearance in SVW is not as informative as in UCF50 dataset. In [46], 80% accuracy is reported for Sports group of UCF101 dataset. Considering all these results, we may conclude that although sports videos feature unique movements, analysis of truly unconstrained videos is still challenging and needs further research.

A.5 Conclusions

To advance computer vision research, and to push the limits of various video analysis problems toward more realistic and unconstrained scenarios happening in the real world, representative and unconstrained datasets are essential. In this regard, we introduced Sports Videos in the Wild (SVW), as a very challenging real-world dataset of sports videos available for genre categorization, action detection, action recognition, and spatio-temporal alignment. We evaluated three different baseline algorithms for sports genre categorization. Experimental results suggest that due to uncorrelatedness of environment and actions in SVW, as well as amateur capturing of the videos, the presented SVW dataset is indeed the most challenging sports and action dataset available.

Appendix B

Publications

- **S. Safdarnejad** and X. Liu, "Spatio-temporal alignment of minimally overlapping sequences from independently moving cameras", in Proc. of Computer Vision and Pattern Recognition Conf. (CVPR) 2017.
- **S. Safdarnejad**, Y. Atoum, and X. Liu, "Temporally Robust Global Motion Compensation by Keypoint-based Congealing", in Proc. of European Conf. on Computer Vision (ECCV) 2016.
- **S. Safdarnejad**, X. Liu, and L. Udpa, "Robust Global Motion Compensation in Presence of Predominant Foreground", in Proc. of British Machine Vision Conf. (BMVC) 2015, Swansea, UK, 2015. **(Best poster award)**
- **S. Safdarnejad**, X. Liu, and L. Udpa, "Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis", in Proc. of IEEE Conf. on Automatic Face and Gesture Recognition (FG) 2015, Ljubljana, Slovenia, 2015.
- P. Banerjee, **S. Safdarnejad**, L. Udpa, S. S. Udpa, NDT Techniques: Signal and Image Processing, in Reference Module in Materials Science and Materials Engineering, Elsevier, 2015.
- **S. Safdarnejad**, Z. Su, C. Ye, L. Udpa, S. Udpa, "Analysis of EC-GMR Data for Detection of Cracks under Fasteners (CUF)", Int. Journal of Applied Electromag. and Mechanics, 2015.
- **S. Safdarnejad**, X. Liu, and L. Udpa, "Genre Categorization of Amateur Sports Videos in the Wild", in Proc. of Int. Conf. of Image Processing (ICIP) 2014, Paris, France, 2014.
- **S. Safdarnejad**, O. Karpenko, L. Udpa, S. Udpa, "A Robust Multi-frequency Mixing Algorithm for Suppression of Rivet Signal in GMR Inspection of Riveted Structures", in Proc. of Quantitative Nondestructive Evaluation (QNDE) 2015, Minneapolis, MN, 2015, AIP Publishing.
- P. Banerjee, **S. Safdarnejad**, L. Udpa, S. Udpa, "Ensemble of Classifiers for Confidence-rated Classification of NDE Signal", in Proc. of Quantitative Nondestructive Evaluation (QNDE) 2015, Minneapolis, MN, 2015, AIP Publishing.

- Z. Su, A. Efremov, **S. Safdarnejad**, A. Tamburrino, L. Udpa, S. Udpa, "Optimization of Coil Design for Near Uniform Interrogating Field Generation", in Proc. of Quantitative Nondestructive Evaluation (QNDE) 2015, Minneapolis, MN, 2015, AIP Publishing.

- P. Banerjee, **S. Safdarnejad**, L. Udpa, S. Udpa, "Investigation of a Comprehensive Confidence Measure in NDE", in Proc. of Review of Progress in Quantitative Nondestructive Evaluation (QNDE) 2014, Boise, ID, 2014, AIP Publishing.

- O. Karpenko, **S. Safdarnejad**, G. Dib, L. Udpa, S. Udpa, A. Tamburrino, "Image Processing Algorithms for Automated Analysis of GMR Data from Inspection of Multilayer Structures", in Proc. of Quantitative Nondestructive Evaluation (QNDE) 2014, Boise, ID, 2014, AIP Publishing.

- **S. Safdarnejad**, L. Udpa, S. Udpa, C. Buynak, G. Steffes, E. Lindgren, and J. Knopp, "Statistical Algorithms for Eddy Current Signal and Noise Analysis", in Proc. of Review of Progress in Quantitative Nondestructive Evaluation (QNDE) 2013, Baltimore, MD, 2013, AIP Publishing.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Anubhav Agarwal, CV Jawahar, and PJ Narayanan. A survey of planar homography estimation techniques. *Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12*, 2005.

[2] Amit Aides, Tamar Avraham, and Yoav Y Schechner. Multiscale ultrawide foveated video extrapolation. In *Computational Photography (ICCP), 2011 IEEE International Conference on*, pages 1–8. IEEE, 2011.

[3] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1926–1933. IEEE, 2012.

[4] Serge Ayer and Martin Vetterli. Method and system for combining video sequences with spatio-temporal alignment, 2001. US Patent 6,320,624.

[5] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[6] Adam Barclay and Hannes Kaufmann. FT-RANSAC: Towards robust multi-modal homography estimation. In *Proc. IAPR Workshop PRRS*, pages 1–4. IEEE, 2014.

[7] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, pages 29–43. Springer, 2010.

[8] Adrien Bartoli, Navneet Dalal, Biswajit Bose, and Radu Horaud. From video sequences to motion panoramas. In *Proc. Conf. Motion and Video Computing Workshops*, pages 201–207. IEEE, 2002.

[9] Adrien Bartoli, Navneet Dalal, and Radu Horaud. Motion panoramas. *Computer Animation and Virtual Worlds*, 15(5):501–517, 2004.

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proc. European Conf. Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[11] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proc. European Conf. Computer Vision (ECCV)*, pages 404–417. Springer, 2006.

[12] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *J. Image and Video Process.*, 2008:1, 2008.

[13] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Proc. Int. Conf. Computer Vision (ICCV)*, volume 2, pages 1395–1402. IEEE, 2005.

[14] Angelo Bosco, Arcangelo Bruna, Sebastiano Battiato, Giuseppe Bella, and Giovanni Puglisi. Digital video stabilization through curve warping techniques. *IEEE Trans. Consumer Electronics*, 54(2):220–224, 2008.

[15] Darlan N Brito, Flávio LC Pádua, Rodrigo L Carceroni, and Guilherme AS Pereira. Synchronizing video cameras with non-overlapping fields of view. In *XXI Brazilian Symp. Computer Graphics and Image Processing*, pages 37–44. IEEE, 2008.

[16] Yaron Caspi and Michal Irani. Aligning non-overlapping sequences. *Int. J. Computer Vision*, 48(1):39–51, 2002.

[17] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. *Pattern Recognition*, pages 236–243, 2003.

[18] Mark Cox, Sridha Sridharan, Simon Lucey, and Jeffrey Cohn. Least squares congealing for unsupervised alignment of images. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[19] Mark Cox, Sridha Sridharan, Simon Lucey, and Jeffrey Cohn. Least-squares congealing for large numbers of images. In *Proc. Int. Conf. Computer Vision (ICCV)*, pages 1949–1956. IEEE, 2009.

[20] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[21] Xinyi Cui, Qingshan Liu, and Dimitris Metaxas. Temporal spectral residual: fast motion saliency detection. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 617–620. ACM, 2009.

[22] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[23] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. European Conf. Computer Vision (ECCV)*, pages 428–441. Springer, 2006.

[24] Oscar Déniz, Gloria Bueno, E Bermejo, and Rahul Sukthankar. Fast and accurate global motion compensation. *Pattern Recognition*, 44(12):2887–2901, 2011.

[25] Konstantinos G Derpanis, Mikhail Sizintsev, Kevin Cannons, and Richard P Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1990–1997. IEEE, 2010.

[26] Caglayan Dicle, Octavia Camps, and Mario Sznaier. The way they move: tracking multiple targets with similar appearance. In *Proc. Int. Conf. Computer Vision (ICCV)*, pages 2304–2311. IEEE, 2013.

[27] Ferran Diego, Daniel Ponsa, Joan Serrat, and Antonio M López. Video alignment for change detection. *IEEE Trans. Image Proc.*, 20(7):1858–1869, 2011.

[28] Ferran Diego, Joan Serrat, and Antonio M López. Joint spatio-temporal alignment of sequences. *IEEE Trans. Multimedia*, 15(6):1377–1387, 2013.

[29] Ling-Yu Duan, Min Xu, Qi Tian, Chang-Sheng Xu, and Jesse S Jin. A unified framework for semantic shot classification in sports video. *Multimedia, IEEE Transactions on*, 7(6):1066–1083, 2005.

[30] Rahul Dutta, Bruce Draper, and J Ross Beveridge. Video alignment to a common reference. In *IEEE Winter Conf. WACV*, pages 808–815. IEEE, 2014.

[31] Motaz El-Saban, Mostafa Izz, Ayman Kaheel, and Mahmoud Refaat. Improved optimal seam selection blending for fast video stitching of videos captured from freely moving devices. In *Proc. Int. Conf. Image Processing (ICIP)*, pages 1481–1484. IEEE, 2011.

[32] Sandro Esquivel, Felix Woelk, and Reinhard Koch. Calibration of a multi-camera rig from non-overlapping views. In *Pattern Recognition*, pages 82–91. Springer, 2007.

[33] Georgios D Evangelidis and Christian Bauckhage. Efficient and robust alignment of unsynchronized video sequences. In *Pattern Recognition*, pages 286–295. Springer, 2011.

[34] Georgios D Evangelidis and Christian Bauckhage. Efficient subframe video alignment using short descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(10):2371–2386, 2013.

[35] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[36] Junhong Gao, Seon Joo Kim, and Michael S Brown. Constructing image panoramas using dual-homography warping. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 49–56. IEEE, 2011.

[37] Tiago Gaspar, Paulo Oliveira, and Paolo Favaro. Synchronization of two independently moving cameras without feature correspondences. In *Proc. European Conf. Computer Vision (ECCV)*, pages 189–204. Springer, 2014.

[38] Michael L Gleicher and Feng Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 5(1):2, 2008.

[39] Nils Hasler, Bodo Rosenhahn, Thorsten Thormahlen, Michael Wand, Jürgen Gall, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 224–231. IEEE, 2009.

[40] Yuwen He, Bo Feng, Shiqiang Yang, and Yuzhuo Zhong. Fast global motion estimation for global motion compensation coding. In *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, volume 2, pages 233–236. IEEE, 2001.

[41] Gary Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems (NIPS)*, pages 764–772, 2012.

[42] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[43] Muhammad Twaha Ibrahim, Rehan Hafiz, Muhammad Murtaza Khan, Yongju Cho, and Jihun Cha. Automatic reference selection for parametric color correction schemes for panoramic video stitching. In *Advances in Visual Computing*, pages 492–501. Springer, 2012.

[44] Wei Jiang and Jinwei Gu. Video stitching with spatial-temporal content-preserving warping. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 42–48. IEEE, 2015.

[45] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo. Trajectory-based modeling of human actions with motion reference points. In *Proc. European Conf. Computer Vision (ECCV)*, pages 425–438. Springer, 2012.

[46] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014.

[47] Tae-Kyun Kim and Roberto Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(8):1415–1428, 2009.

[48] Hui Kong, Jean-Yves Audibert, and Jean Ponce. Detecting abandoned objects with a moving camera. *IEEE Trans. Image Process.*, 19(8):2201–2210, 2010.

[49] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. HMDB51: A large video database for human motion recognition. In *High Performance Comput. Sci. Eng.*, pages 571–582. Springer, 2013.

[50] Jukka Lankinen and Joni-Kristian Kämäräinen. Local feature based unsupervised alignment of object class images. In *Proc. British Mach. Vision Conf. (BMVC)*, volume 1, 2011.

[51] Ivan Laptev. On space-time interest points. *Int. J. Comput. Vision*, 64(2-3):107–123, 2005.

[52] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[53] Erik G Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(2):236–250, 2006.

[54] Erik G Learned-Miller et al. ICA using spacings estimates of entropy. *J. Machine Learning Research*, 4:1271–1295, 2003.

[55] José Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011.

[56] Qian Li, Shifeng Chen, and Beiwei Zhang. Predictive video saliency detection. In *Pattern Recognition*, pages 178–185. Springer, 2012.

[57] Xiangru Li and Zhanyi Hu. Rejecting mismatches by correspondence function. *Int. J. Comput. Vision*, 89(1):1–17, 2010.

[58] Yunpeng Li, Sing Bing Kang, Neel Joshi, Steve M Seitz, and Daniel P Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2424–2431. IEEE, 2010.

[59] Chung-Ching Lin, Sharathchandra U Pankanti, Karthikeyan Natesan Ramamurthy, and Aleksandr Y Aravkin. Adaptive as-natural-as-possible image stitching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1163. IEEE, 2015.

[60] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. Smoothly varying affine stitching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 345–352. IEEE, 2011.

[61] Ce Liu, William T Freeman, and Edward H Adelson. Analysis of contour motions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 913–920, 2006.

[62] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos in the wild. In *Proc. IEEE Conf. CVPR*, pages 1996–2003. IEEE, 2009.

[63] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proc. IEEE Conf. CVPR*, pages 4209–4216. IEEE, 2014.

[64] Xiaoming Liu, Yan Tong, and Frederick W. Wheeler. Simultaneous alignment and clustering for an image ensemble. In *Proc. Int. Conf. Computer Vision (ICCV)*, pages 1327–1334. IEEE, 2009.

[65] David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 60(2):91–110, 2004.

[66] David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[67] Cheng Lu and Mrinal Mandal. A robust technique for motion-based video sequences temporal alignment. *IEEE Trans. Multimedia*, 15(1):70–82, 2013.

[68] Simon Lucey, Rajitha Navarathna, Ahmed Bilal Ashraf, and Sridha Sridharan. Fourier lucas-kanade algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1383–1396, 2013.

[69] Jiayi Ma, Jun Chen, Delie Ming, and Jinwen Tian. A mixture model for robust point matching under multi-layer motion. *PloS one*, 9(3):e92282, 2014.

[70] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014.

[71] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2929–2936. IEEE, 2009.

[72] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Proc. IEEE Conf. CVPR*, 2009.

[73] C Krishna Mohan and B Yegnanarayana. Classification of sport videos using edge-based features and autoassociative neural network models. *Signal, Image and Video Processing*, 4(1):61–73, 2010.

[74] Eduardo Monari and Thomas Pollok. A real-time image-to-panorama registration approach for background subtraction using pan-tilt-cameras. In *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance (AVSS)*, pages 237–242. IEEE, 2011.

[75] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. European Conf. ECCV*, pages 392–405. Springer, 2010.

[76] Flávio LC Pádua, Rodrigo L Carceroni, Geraldo AMR Santos, and Kiriakos N Kutulakos. Linear sequence-to-sequence alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):304–320, 2010.

[77] F Perazzi, A Sorkine-Hornung, H Zimmer, P Kaufmann, O Wang, S Watson, and M Gross. Panoramic video from unstructured camera arrays. In *Computer Graphics Forum*, volume 34, pages 57–68. Wiley Online Library, 2015.

[78] P. J. Phillips, H. Moon, P. J. Rauss, and S. Rizvi. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, October 2000.

[79] Yair Poleg and Shmuel Peleg. Alignment and mosaicing of non-overlapping images. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012.

[80] Dmitry Pundik and Yael Moses. Video synchronization using temporal signals from epipolar lines. In *Proc. European Conf. Computer Vision (ECCV)*, pages 15–28. Springer, 2010.

[81] Cen Rao and Mubarak Shah. View-invariance in action recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–316. IEEE, 2001.

[82] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *Int. J. Computer Vision*, 50(2):203–226, 2002.

[83] Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[84] Xiaobo Ren, Tony X Han, and Zhihai He. Ensemble video object cut in highly dynamic scenes. In *Proc. IEEE Conf. CVPR*, pages 1947–1954. IEEE, 2013.

[85] Sreemanananth Sadanand and Jason J Corso. Action bank: A high-level representation of activity in video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1234–1241. IEEE, 2012.

[86] S. Morteza Safdarnejad, Yousef Atoum, and Xiaoming Liu. Temporally robust global motion compensation by keypoint-based congealing. In *Proc. European Conf. Computer Vision (ECCV)*, pages 101–119. Springer, 2016.

[87] S. Morteza Safdarnejad, Xiaoming Liu, and Lalita Udpa. Genre categorization of amateur sports videos in the wild. In *Proc. Int. Conf. ICIP*. IEEE, 2014.

[88] S. Morteza Safdarnejad, Xiaoming Liu, and Lalita Udpa. Robust global motion compensation in presence of predominant foreground. In *Proc. British Machine Vision Conf. (BMVC)*, 2015.

[89] S. Morteza Safdarnejad, Xiaoming Liu, Lalita Udpa, Brooks Andrus, John Wood, and Dean Craven. Sports videos in the wild (SVW): A video dataset for sports analysis. In *Proc. Int. Conf. Automatic Face and Gesture Recognition (FG)*, pages 1–7. IEEE, 2015.

[90] Masatoshi Sakamoto, Yasuyuki Sugaya, and Kenichi Kanatani. Homography optimization for consistent circular panorama generation. In *Advances in Image and Video Technology (PSIVT)*, pages 1195–1205. Springer, 2006.

[91] Pekka Sangi, Jari Hannuksela, Janne Heikkilä, and Olli Silvén. Sparse motion segmentation using propagation of feature labels. In *VISAPP (2)*, pages 396–401. Citeseer, 2013.

[92] Harpreet S Sawhney, Steve Hsu, and Rakesh Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proc. European Conf. Computer Vision (ECCV)*, pages 103–119. Springer, 1998.

[93] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, volume 3, pages 32–36. IEEE, 2004.

[94] Hae Jong Seo and Peyman Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of vision*, 9(12):15, 2009.

[95] Joan Serrat, Ferran Diego, Felipe Lumbreras, and José Manuel Álvarez. Synchronization of video sequences from free-moving cameras. In *Pattern Recognition and Image Analysis*, pages 620–627. Springer, 2007.

[96] Fatemeh Shokrollahi Yancheshmeh, Ke Chen, and Joni-Kristian Kamarainen. Unsupervised visual alignment with similarity graphs. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2908. IEEE, 2015.

[97] Heung-Yeung Shum and Richard Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Proc. Int. Conf. Computer Vision (ICCV)*.

[98] Aljoscha Smolić, Yuriy Vatis, Heiko Schwarz, and Thomas Wiegand. Long-term global motion compensation for advanced video coding. In *ITG-Fachtagung Dortmunder Fernsehseminar*, pages 213–216, 2003.

[99] Francesco Solera, Simone Calderara, and Rita Cucchiara. Learning to divide and conquer for online multi-target tracking. *arXiv preprint arXiv:1509.03956*, 2015.

[100] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[101] Drew Steedly, Chris Pal, and Richard Szeliski. Efficiently registering video into panoramic mosaics. In *Proc. Int. Conf. Computer Vision (ICCV)*, volume 2, pages 1300–1307. IEEE, 2005.

[102] M Sullivan and M Shah. Action mach: Maximum average correlation height filter for action recognition. In *Proc. IEEE Conf. CVPR*, pages 1–8. IEEE, 2008.

[103] Yunda Sun, Bo Li, Baozong Yuan, Zhenjiang Miao, and Chengkai Wan. Better foreground segmentation for static cameras via new energy form and dynamic graph-cut. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, volume 4, pages 49–52. IEEE, 2006.

[104] Zygmunt L Szpak, Wojciech Chojnacki, and Anton van den Hengel. Robust multiple homography estimation: An ill-solved problem. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2132–2141. IEEE, 2015.

[105] Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with j-linkage. In *Proc. European Conf. ECCV*, pages 537–547. Springer, 2008.

[106] Ben J Tordoff and David W Murray. Guided-mlesac: Faster image transform estimation by using matching priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1523–1535, 2005.

[107] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Comput. Vision and Image Understanding*, 78(1):138–156, 2000.

[108] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 1999.

[109] Hirofumi Uemura, Seiji Ishikawa, and Krystian Mikolajczyk. Feature tracking and motion compensation for action recognition. In *Proc. British Mach. Vision Conf. (BMVC)*, pages 1–10, 2008.

[110] Yaron Ukrainitz and Michal Irani. *Aligning sequences and actions by maximizing space-time correlations*. Springer, 2006.

[111] Yaron Ukrainitz and Michal Irani. *Aligning sequences and actions by maximizing space-time correlations*. Springer, 2006.

[112] M Vargas and E Malis. Deeper understanding of the homography decomposition for vision-base control. *Unité de recherche INRIA Sophia Antipolis (France)*, 2007.

[113] CK Wan, BZ Yuan, and ZJ Miao. A new algorithm for static camera foreground segmentation via active coutours and GMM. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[114] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proc. IEEE Conf. CVPR*, pages 3169–3176. IEEE, 2011.

[115] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. IEEE, 2011.

[116] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vision*, 103(1):60–79, 2013.

[117] Heng Wang, Cordelia Schmid, et al. Action recognition with improved trajectories. In *Proc. Int. Conf. Computer Vision (ICCV)*, 2013.

[118] Jinjun Wang, Changsheng Xu, and Engsiong Chng. Automatic sports video genre classification using pseudo-2d-hmm. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, volume 4, pages 778–781. IEEE, 2006.

[119] Lu Wang, Ulrich Neumann, and Suya You. Wide-baseline image matching using line signatures. In *Proc. Int. Conf. ICCV*, pages 1311–1318. IEEE, 2009.

[120] Oliver Wang, Christopher Schroers, Henning Zimmer, Markus Gross, and Alexander Sorkine-Hornung. Videosnapping: Interactive synchronization of multiple videos. *ACM Trans. on Graphics (TOG)*, 33(4):77, 2014.

[121] Daniel Weinland, Edmond Boyer, and Remi Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Proc. Int. Conf. Computer Vision (ICCV)*, pages 1–7. IEEE, 2007.

[122] Shandong Wu, Omar Oreifej, and Mubarak Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *Proc. Int. Conf. Computer Vision (ICCV)*, pages 1419–1426. IEEE, 2011.

[123] Yawen Xue, Xiaojie Guo, and Xiaochun Cao. Motion saliency detection using low-rank and sparse decomposition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1485–1488. IEEE, 2012.

[124] Qing Yan, Yi Xu, Xiaokang Yang, and Truong Nguyen. HEASK: Robust homography estimation based on appearance similarity and keypoint correspondences. *Pattern Recognition*, 47(1):368–387, 2014.

[125] Junsong Yuan, Zicheng Liu, and Ying Wu. Discriminative video pattern search for efficient action detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1728–1743, 2011.

[126] Xun Yuan, Wei Lai, Tao Mei, Xian-Sheng Hua, Xiu-Qing Wu, and Shipeng Li. Automatic video genre categorization using hierarchical svm. In *Proc. Int. Conf. Image Processing (ICIP)*, pages 2905–2908. IEEE, 2006.

[127] Jordi Zaragoza, Tat-Jun Chin, Quoc-Huy Tran, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving dlt. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1285–1298, 2014.

[128] Wei Zeng and Hongming Zhang. Depth adaptive video stitching. In *Proc. IEEE Conf. Computer and Information Science (ICIS)*, pages 1100–1105. IEEE, 2009.

[129] Ning Zhang, Ling-Yu Duan, Qingming Huang, Lingfang Li, Wen Gao, and Ling Guan. Automatic video genre categorization and event detection techniques on large-scale sports data. In *Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research*, pages 283–297. IBM Corp., 2010.

[130] Ning Zhang, Ling-Yu Duan, Lingfang Li, Qingming Huang, Jun Du, Wen Gao, and Ling Guan. A generic approach for systematic analysis of sports videos. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):46, 2012.

[131] Sheng-hua Zhong, Yan Liu, Feifei Ren, Jinghuan Zhang, and Tongwei Ren. Video saliency detection via dynamic consistent spatio-temporal attention modelling. In *AAAI*, 2013.

[132] Yaping Zhu, Natan Jacobson, Hong Pan, and Truong Nguyen. Motion-decision based spatiotemporal saliency for video sequences. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 1333–1336. IEEE, 2011.

[133] Marco Zuliani, Charles S Kenney, and BS Manjunath. The multiransac algorithm and its application to detect planar homographies. In *Proc. Int. Conf. ICIP*, volume 3, pages III–153. IEEE, 2005.