Space-Time Scene Manifolds

Y. Wexler D. Simakov

Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
Rehovot, 76100 Israel

Abstract

The space of images is known to be a non-linear subspace that is difficult to model. This paper derives an algorithm that walks within this space. We seek a manifold through the video volume that is constrained to lie locally in this space. Every local neighborhood within the manifold resembles some image patch. We call this the Scene Manifold *because the solution traces the scene outline. For a broad class of inputs the problem can be posed as finding the shortest path in a graph and can thus be solved efficiently to produce the globally optimal solution. Constraining appearance rather than geometry gives rise to numerous new capabilities.*

Here we demonstrate the usefulness of this approach by posing the well-studied problem of mosaicing in a new way. Instead of treating it as geometrical alignment, we pose it as an appearance optimization. Since the manifold is constrained to lie in the space of valid image patches, the resulting mosaic is guaranteed to have the least distortions possible. Any small part of it can be seen in some image even though the manifold spans the whole video. Thus it can deal seamlessly with both static and dynamic scenes, with or without 3D parallax. Essentially, the method simultaneously solves two problems that have been solved only separately until now: alignment and mosaicing.

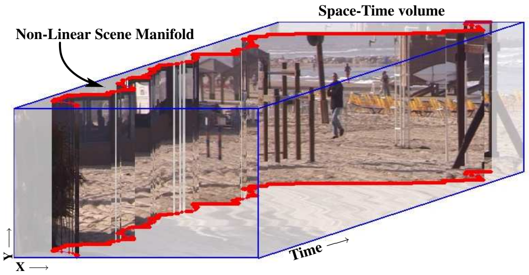
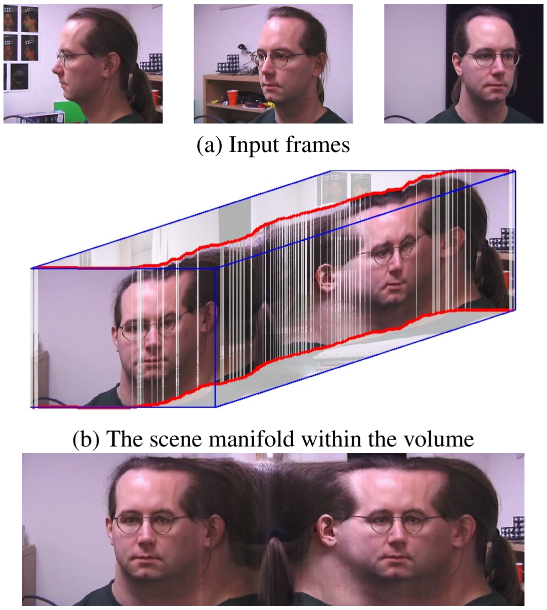


Figure 1. The scene manifold is constrained to cut the space-time volume so that to incur no distortions. See Fig. 4.

1. Introduction

The space of real images and image patches is known to be a small non-linear subspace of all pixel color combinations. This sub-space is difficult to model. Nevertheless, it is very structured and the strong constraints it induces can be very useful. Each video frame provides a concrete sample from that space and the ordered collection of all frames creates the space-time volume as is depicted in Fig. 1. There are many ways to cut through this volume. The surface of any manifold through it induces an image that is likely to contain many visual artifacts. In this work we seek to find the manifold that will incur the fewest distortions and artifacts. The parameters of the best manifold depend on the motion of the camera as well as dynamic changes in the world. Rather than use explicit geometrical constraints, we constrain the appearance of the image induced by the manifold to lie in the space of real images: *every local neighborhood in it resembles some image*. In many useful cases, there will be no distortions at all and the method described here is guaranteed to find the correct solution. We call the resulting slice the *Scene Manifold* as the solution traces the scene outline. Essentially, the method simultaneously solves two problems that have been solved only separately until now: alignment and mosaicing.

A natural domain to test this approach is the well-studied problem of mosaicing, where a video sequence is represented by one image. Previous takes on this problem focused on the geometrical aspects, whether in the $2D$ image space or the $3D$ world coordinates. The geometrical alignment was often followed by some post-processing to eliminate seams. Our approach addresses the mosaicing problem from a new angle. It seeks to find a manifold that cuts through the space-time volume in a way that will minimize visual artifacts. This reliance on appearance, rather than

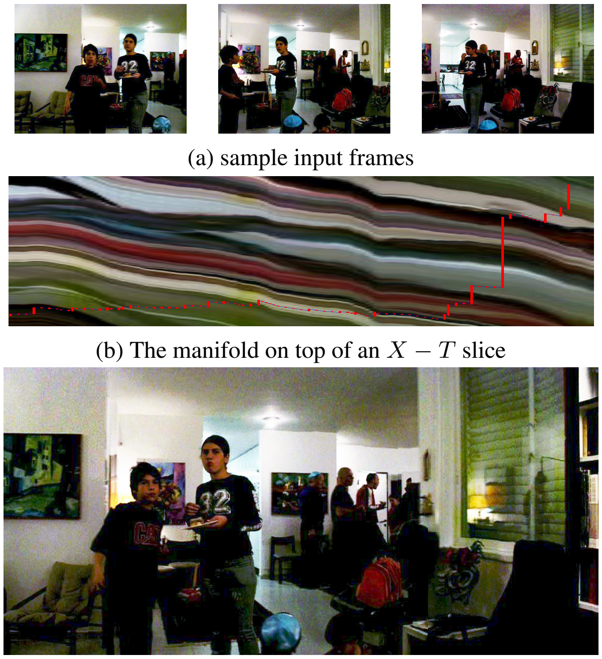


(a) Input frames

(b) The scene manifold within the volume

(c) The image induced by the scene manifold

Figure 2. "Head" sequence from [7]. Note how the manifold in **(b)** traces the face and head, thus recovering the surface correctly and resulting in a "peeled out" figure exterior. The reader is encouraged to compare this result with the one in [7].



(a) sample input frames

(b) The manifold on top of an $X - T$ slice

(c) The image induced by the scene manifold

Figure 3. "Dinner" sequence. Dynamic scene with moving camera. The $X - T$ slice in (b) shows a top view of the space-time volume. The convention here is that time is along the horizontal axis.

geometry, allows the method to deal seamlessly with both *static and dynamic* scenes, *with or without parallax*. The algorithm finds the optimal manifold for a large class of videos by posing the appearance constraint as an efficient shortest-path computation in a graph.

2. Previous work

The closest work to this one is probably [1] where the goal is also the creation of mosaics. Each video frame is parametrically aligned and a fixed strip in it is fused to form a mosaic. As the alignment is local, the resulting global $2D$ coordinate system on the mosaic does not necessarily correspond to a plane in the world but can span a manifold in $3D$. Note that this should not be confused with the Scene Manifold discussed here which lies within the space-time volume. The reliance on parametric image alignment means that the method is only suitable to mostly-static scenes with no parallax. In our approach those restrictions do not exist and the manifold can also cross the same frame more than once as is shown in Fig. 7.

In [7], constraints on the light rays in $3D$ are used to classify the family of stereo images for perspective and multi-perspective images. The work concentrates on the geometrical aspects of the projection and assumes a rigid world setting. Under the motion model there, a *straight* cut through the volume produces mosaics. Other cuts through this volume may also be interesting as shown in [12]. Fig. 2 shows that our approach can also handle cases when the camera moves around an object to capture its 360^o appearance and create a multiple-center-of-projection image automatically, without any geometric assumptions or explicit alignment.

As our objective is based on appearance, it is also related to the domain of creation and synthesis of textures. The famous work of [10] samples from the space of texture patches to synthesize textures. As it only samples from that space, it does not need to model it in any way. In [8] both $3D$ geometry and $2D$ image texture were used for synthesizing new views. In our work the appearance constraint is stronger and is computed efficiently in closed-form, thus further geometrical considerations are unnecessary. The work of [9] and other works derived from it (such as [11]) use maximum-flow formulation to combine image and video patches. Note that this formulation requires the patches to be pre-aligned and so are not applicable here.

The recent work of [2] deals with the creation of a video mosaic that nicely extends the idea of mosaicing. It modifies the space-time volume so that sweeping a (multi)linear cut through it will produce a pleasing video with a larger spatial extent than the original. This works for stochastic motion texture but does not seem to extend as is for structured motion, such as the moving people in Fig. 3.

The approach taken here is very different for a number of reasons. First, *no alignment* is assumed. Instead we consider several possible neighborhoods for each patch and decide among them during the global search. Second, the patches used here are *very small* (1-pixel wide) and so do not contain perspective effects (e.g. parallax) - thus no post-correction is needed. Third, the method presented here is guaranteed to find the *global optimum* within this domain in an efficient way.

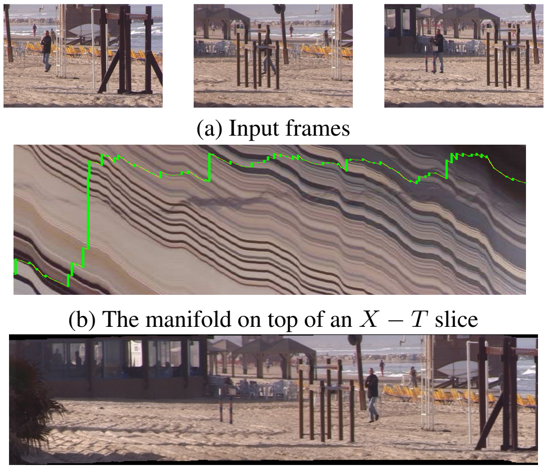
3. The Algorithm

We seek a manifold that will cut through the space-time volume, from the first frame to the last, in a way that does not incur any visual artifacts, i.e. *looks locally like a real image*. The motivation behind this definition is that each image is a sample from the space of real images and the resulting manifold should be embedded in this space.

The main difficulty in this task arises from the fact that the space-time volume, as shown in Fig. 1, is not symmetrical in all dimensions. Each $X - Y$ slice is an image and so, by definition, contains a valid projection of the world. This means that the manifold can pass freely within each image. In contrast, the time dimension T is more problematic and the manifold cannot always cross from one image to another. Between the video frames, the camera may move in the world and the scene may change as a whole or only locally in some areas. According to the above definition, neighboring patches in one image can be neighbors in the manifold whereas successive pairs from two images may or may not form a valid, artifact-free, image patch and hence are undesirable.

In this work we restrict ourselves to manifolds that cut the volume to form a ruled surface which we assume to be vertically aligned. This scenario is common and appears, for example, when the camera pans and translates *mostly* along one direction (e.g. the ground plane). Under this assumption, the manifold is uniquely defined by a path in a $2D$ profile, such as the curve on top of Fig. 1. This path encodes the appearance and placement of the manifold. Each point in the path represents a vertical strip in the manifold (a one-pixel wide column of pixels). The path also induces an ordering on the manifold. A point in it has two neighbors, one before it and one after it. Thus the quality of a manifold depends only on local neighborhoods within it. In Section 3.4 we show how to extend the graph to handle vertical motion as well.

While the number of paths in the volume is exponentially large, we do not need to check all of them. The locality allows us to use dynamic programming to find the global optimum efficiently. For a patch in one image we test several possible transitions onto the next images. Each transition can be scored according to the appearance it induces on the manifold. To find the optimal path we construct graph where the nodes represent image strips and the edges are the possible transitions. The optimal manifold is then the shortest path in this graph.



(a) Input frames

(b) The manifold on top of an $X - T$ slice

(c) The image induced by the scene manifold

Figure 4. "Beach" sequence. The camera is panning with vertical jitter which is handled by the algorithm. All the motion in the scene is handled seamlessly.

3.1. Definitions

Let $\mathcal{V}$ be a video sequence, which is an ordered set of images $\{\Psi_f\}_1^N$. Each image in turn contains an ordered set of strips $\psi_f^1 ... \psi_f^K$ and so, the space-time volume is the set of strips $\{\psi_f^i\}$ (i is coordinate on the spatial X axis, f is coordinate on the time axis). Using these notations, the desired scene manifold can be denoted by a list of strips $\{\psi_{f_j}^{k_j}\}_{j=1}^L$. The width of the strips is set to one pixel as wider strips may form naturally when needed. We seek to find the scene manifold which cuts this space without any visual artifacts or distortions. These can only arise by stacking together pairs of strips that do not appear in the space of real images and so we define a cost measure for any pair of strips (ψ_f^k, ψ_g^l) as

$$D(\psi_f^i, \psi_g^j) = \min_n \min_{UV \subset \Psi_n} ||UV - \psi_f^i \psi_g^j|| \qquad (1)$$

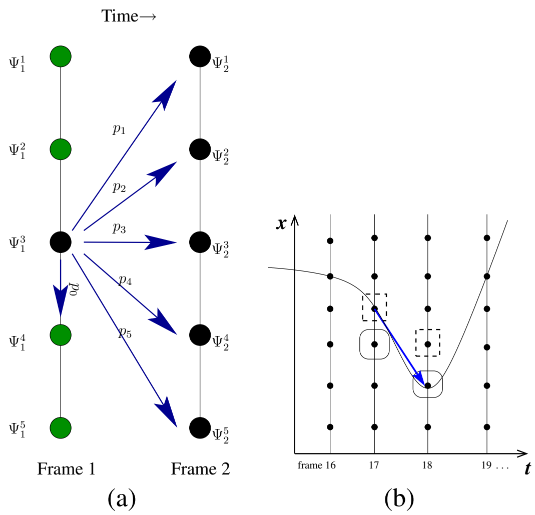
That is, a pair of strips are valid neighbors if, when placed side-by-side, they are similar to some pair in one of the video images (a similar formula was used in [6] but was formulated there in statistical terms which did not lead to a closed-form solution). We can get a good upper bound on $D(\cdot, \cdot)$ by

$$D(\psi_f^i, \psi_g^j) \quad \leq \quad C(\psi_f^i, \psi_g^j) = \qquad (2)$$

$$= \min \left\{ \begin{array}{l} ||\psi_f^i \psi_g^j - \psi_f^i \psi_f^{i+1}||, \\ ||\psi_f^i \psi_g^j - \psi_g^{j-1} \psi_g^j|| \end{array} \right\}$$

$$= \min \left\{ ||\psi_g^j - \psi_f^{i+1}||, ||\psi_f^i - \psi_g^{j-1}|| \right\}$$

This bound is fairly tight since we already know two locations where the cost is close to the global minimum. The cost function (2) allows us to define a graph where the cost of each edge efficiently encodes the cost of cutting the volume along it.



(a) (b)

Figure 5. Graph construction. The graph encodes possible strip transitions. Each circle in **(a)** denotes a strip and the edges represent possible transitions between them. In **(a)**, a pixel in Frame 1 has one natural transition within the same frame (p_0) and 5 possible transitions to the next frame ($p_{1,\ldots,5}$). **(b)** shows the manifold as it passes in the volume. The graph arc in (b) will have a small cost either if the broken rectangles are similar or if the solid ovals are similar (Eq. 2). In both cases a broken rectangle and a solid oval can be seamlessly put side by side.

3.2. Graph Construction

Let $\mathcal{G} = (V, E)$ denote a graph where the nodes $V = \{\psi_f^i\}$ are the $K \cdot N$ image strips and the edges $E \subseteq V \times V$ encode the possible transitions between the strips. Each edge has an associated transition cost $C : E \to \Re$ as defined in Eq.(2). Consider for example the two patches ψ_1^3 from the first image and ψ_2^4 from the second image, denoted by circles in Fig. 5. The error of this placement is small if the pair ψ_1^3, ψ_2^4 appears together in some image. Instead of performing a costly search, we compute the similarity of ψ_1^3 and ψ_2^3, and if they are similar, then ψ_2^4 can be placed next to ψ_1^3 (just like it is already placed next to ψ_2^3). Thus, the cost of a transition from node ψ_i^j to ψ_k^l is directly related to the similarity between ψ_i^j to ψ_k^{l-1} as is given in Eq. (2).

As we assume the frames come from a video sequence, there is no need to add to the graph all possible edges. Instead, we only compute edges between nearby patches in space and in time. This limit depends on the maximal motion velocity that is expected, which is typically five pixels and frames.

3.3. Global Optimization

When the edge costs are computed, two special nodes need to be added to mark the beginning and end points of the manifold. For most examples presented here, the goal is to find a path from the first frame to the last so that the whole video sequence is represented. The node v_{start} is connected to all the nodes in the first frame with zero cost and v_{end} is connected to all the nodes in the last frame with zero cost. The shortest path between them contains the manifold with the least amount of distortions.

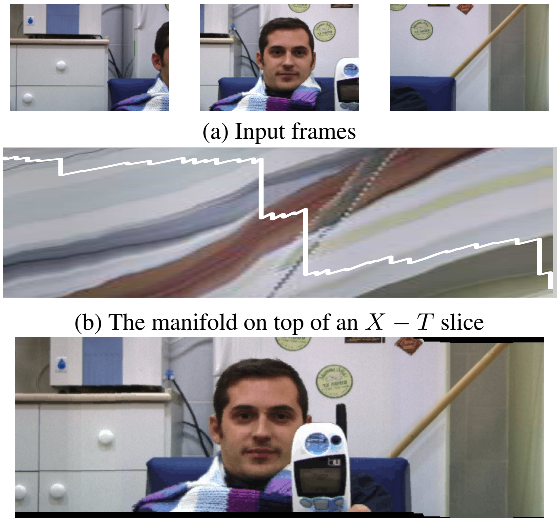
In Fig 7 we show how one can use this method to create a synopsis of a scene where specific events at different locations and times are combined together. The path is forced to pass through these points so that they are included in the manifold as explained in Section 4.

Now, each path from v_{start} to v_{end} corresponds to a manifold. The edge weights encode the transition costs and we seek the one with the least distortions. The well known shortest-path algorithm of [5] can now be employed to find this global optimum efficiently. The resulting minimal-cost path can be directly translated to a mosaic by stacking together the strips corresponding to $\{\psi_{f_j}^{k_j}\}_{j=1}^{L}$ side by side. The above construction guarantees that this manifold has the least cumulative visual errors.

3.4. On-Line Graph Construction

We have introduced the algorithm in two steps (graph construction and optimization) only for the sake of simplicity. In practice, there is no need to pre-compute the whole graph but instead it is constructed on-line during the shortest-path algorithm. As a node is visited, the cost edges connected to it are computed and added to a priority queue. The advantage of such computation is that most possible transitions incur a very high cost in Eq. (2) and so will never be taken. Hence most edge costs need not be computed at all and the complexity is proportional to the manifold length.

This saving is more noticeable for more complex motion types, including vertical shifts. For example, if one wishes to handle vertical shake as well, a patch ψ_f^j may match several vertical shifts of ψ_g^l and so several nodes are needed for each strip, multiplying the number of nodes in the graph by the number of allowed shifts. While the number of nodes grows rapidly, the number of nodes actually visited ones does not.



(a) Input frames

(b) The manifold on top of an $X - T$ slice

(c) The image induced by the scene manifold

Figure 6. Translating camera from [2]. Note the resulting mosaic contains no noticeable distortions as the path chooses the correct transition in each step thus avoiding parallax artifacts. This example also demonstrates both horizontal and vertical alignment

3.5. Coarse-To-Fine Computation

Another improvement of the algorithm is the use of a coarse-to-fine framework. A spatio-temporal Gaussian pyramid is built from the input video. Each pyramid level is down-scaled to contain half the resolution in the spatial and in the temporal dimensions thus each level contains $\frac{1}{8}$ of the pixels of the level below it. At the coarsest level a complete graph is built and the optimal path is extracted. The path is then transformed to the next resolution and only strips close to that path in space and in time will be used to build the graph for the higher resolution level.

The main advantage of this multi-scale framework is that the similarity computation in the coarser levels captures global features and so consistency of larger video features will be maintained. The refinements on subsequent levels make the small adjustments necessary to ensure consistency of the finer details. As the refined path is constrained to lie in the vicinity of the coarser one, global features cannot be disrupted even if they would agree locally.

Another advantage of the multi-scale process is computational cost. The graph contains one node per strip so each subsequent pyramid level reduces the graph to have $\frac{1}{4}$ the number of nodes. Typically, when the computation reaches the finest level, roughly 5% of the strips are used, thus resulting in a substantial speedup.

4. Experiments

We have implemented the algorithm in Matlab and have conducted several experiments to test it. Please visit the project's website at: `http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeSceneManifolds`

The "Dinner" sequence in Fig. 3 contains a natural setting. The camera translates while people are moving about. The parallax and the motion of the people can clearly be seen in the $X - T$ slice in 3(b) (where time is the horizontal axis). In the resulting mosaic the two children on the left were taken from the same image as is evident in the large vertical line on the path in 3(b). As they move non-rigidly, any stitch between frames passing through the children would incur visible seam in the mosaic, which is automatically avoided by our algorithm.

The "Cricket" sequence in Fig. 7 demonstrates the creation of a synopsis mosaic. In this example we manually picked three points, two of which are in one frame. The path was constrained to pass through three points at different times and so must include these events as shown in Fig. 7(d). Note that the objects (in this case people) do not need to be segmented or even accurately located as the algorithm will automatically ensure that they are not cut or distorted.

We have included here two known sequences for reference. The "Head" sequence in Fig. 2 was taken from [7]. It was taken by a rotating camera that surrounds a person's head. As the head surface is not cylindrical, the distance from it to the camera changes. The freedom to choose the transition locations allows the resulting manifold to avoid foreshortening effects that were apparent in [7]. A close examination of Fig. 2(b) shows that that it wraps the head accurately. The sequence in Fig. 6 was taken from [2]. The camera moves horizontally from left to right and the scene contains substantial parallax, as is seen by the varying angles on the $X - T$ slice in 6(b). The resulting manifold contains minor visual artifacts on the left-hand side. The manifold was allowed to move vertically ± 5 pixels as can be seen on the top and bottom boundaries in 6(c).

5. Summary and Conclusions

We have presented the notion of the Scene Manifold, that cuts through the space-time volume so as to minimize visual artifacts. The resulting manifold will lie in the subspace of real images from the video itself whenever possible and will minimize the distance from it otherwise. We have demonstrated the strength of this approach by constructing mosaics of very challenging, highly dynamic videos, without the use of any geometrical information and without assuming alignment.

6. Acknowledgments

The authors would like to thank Michal Irani for her many insightful remarks. This research was supported in part by the Moross Laboratory for Vision and Motor Control at the Weizmann Institute of science.

References

[1] S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," *CVPR* pp. 338-343, 1997.

[2] A. Rav-Acha and Y. Shor and S. Peleg, "Mosaicing with Parallax using Time Warping" *Second IEEE Workshop on Image and Video Registration (IVR'04)*, 2004. http://www.cs.huji.ac.il/∼alexis/

[3] M. Irani and B. Rousso and S. Peleg, "Recovery of Ego-Motion Using Region Alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 19(3), pp. 268-272, 1997

[4] A. Zomet and D. Feldman and S. Peleg and D. Weinshall, "Mosaicing New Views: The Crossed-Slits Projection" *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 25(6), pp 741-754, 2003

[5] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs." *Numerische Math.* Vol 1, pp. 269-271, 1959.

[6] Y. Wexler and E. Shechtman and M. Irani, "Space-Time Video Completion", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Vol. 1, pp. 120-127, 2004.

[7] S. Seitz and J. Kim, "The Space of All Stereo Images" *IJCV, Marr Prize Special Issue*, Vol. 48(1), pp. 21-38, 2002. http://grail.cs.washington.edu/projects/stereo/

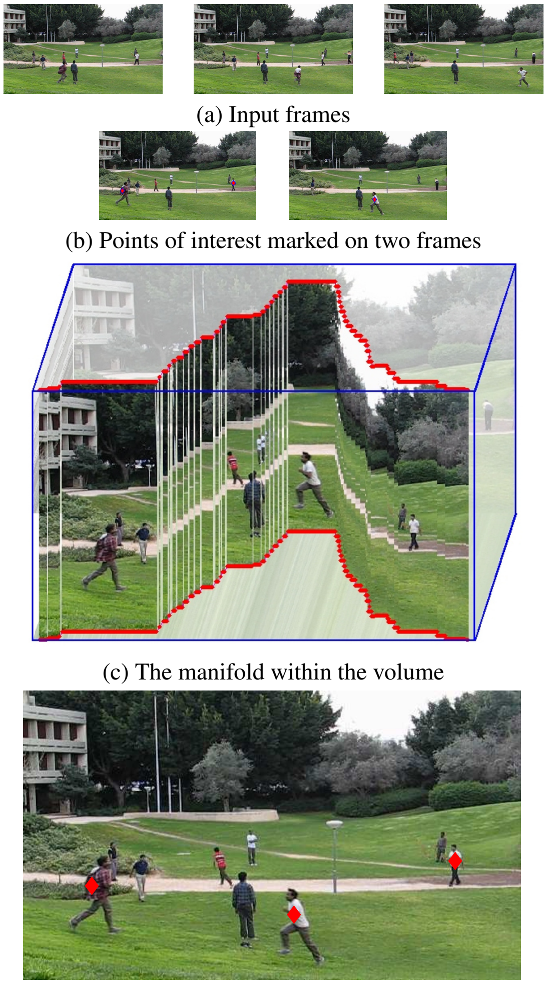
[8] A.W. Fitzgibbon and Y. Wexler and A. Zisserman. "Image-based rendering using image-based priors" *International Conference on Computer Vision (ICCV)*, pp. 1176-1183, 2003.

[9] V. Kwatra and A. Schdl and I. Essa and G. Turk and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts", *ACM Transactions on Graphics, SIGGRAPH 2003*, Vol. 22(3), pp. 277-286, 2003.

[10] A.A. Efros and T.K. Leung, "Texture Synthesis by Non-parametric Sampling" *IEEE International Conference on Computer Vision*, pp. 1033-1038, 1999.

[11] A. Agarwala, C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, R. Szeliski, "Panoramic Video Textures" *ACM Transactions on Graphics, SIGGRAPH 2005*, 2005.

[12] A. Klein, P. Sloan, A. Finkelstein and M. Cohen", "Stylized video cubes", *ACM SIGGRAPH Symposium on Computer Animation*, pp. 15-22, 2002.



(a) Input frames

(b) Points of interest marked on two frames

(c) The manifold within the volume

(d) The image induced by the scene manifold

Figure 7. A synopsis example. This input is from a static camera covering a cricket match. Three people were selected in the input by manually choosing 3 points in the space-time volume as shown in **(c)**. The resulting manifold contains a synopsis of the event and the points are marked in red. Note that no segmentation is needed as the manifold automatically includes the entire objects.