A Robust Subspace Approach to Layer Extraction

Qifa Ke and Takeo Kanade
Computer Science Department
Carnegie Mellon University
{ke+,tk}@cs.cmu.edu

Abstract

Representing images with layers has many important applications, such as video compression, motion analysis, and 3D scene analysis. This paper presents a robust subspace approach to reliably extracting layers from images by taking advantages of the fact that homographies induced by planar patches in the scene form a low dimensional linear subspace. Such subspace provides not only a feature space where layers in the image domain are mapped onto denser and better-defined clusters, but also a constraint for detecting outliers in the local measurements, thus making the algorithm robust to outliers. By enforcing the subspace constraint, spatial and temporal redundancy from multiple frames are simultaneously utilized, and noise can be effectively reduced. Good layer descriptions are shown to be extracted in the experimental results.

1. Introduction

Decomposing an image sequence into layers has been proposed as an efficient video representation for coding, motion and scene analysis, and 3D scene representation [26, 17, 2, 22]. There are two types of layers: 2D layer and 3D layer. A 2D layer consists of 2D sub-images such that pixels within the same layer share common 2D parametric transformation. A 3D layer consists of a 3D plane equation, the texture of that plane, a per-pixel opacity map and depth-offset [2]. Extracting 3D layers usually requires the knowledge of camera motion, which is essentially a structure from motion (SFM) task, a non-trivial task for computer vision, and may not be necessary for some applications such as video coding, where 2D layers are usually sufficient. This paper focuses on 2D layer extraction from uncalibrated images.

The major issues of layer extraction are: (1) model initialization, including the number of layers and the model-based motion of each layer; and (2) the determination of spatial support for each layer. A nature approach is to formulate the layer extraction as a MLE or MAP estimation problem [12, 6, 1, 27, 25, 16], which is then optimized by the Expectation-Maximization (EM) algorithm. The number of layers is usually determined by some model selection criteria, e.g., MDL [6, 1]. Model initialization is a critical but difficult step in order for the EM algorithm to converge to desired optimal solutions [21, 25, 15].

Another category of approaches is to group pixels or regions into layers based on the affinity of local measurements, e.g., the k-means algorithm [26], or the normalized graph cut [21]. Grouping pixels based on pure local measurements does not have the initialization difficulty. However, such approach ignores the global constraints and tends to make early commitments to noisy local measurements. Moreover, grouping in high dimensional space is often unreliable given noisy local measurements.

In this paper, we present a low dimensional robust linear subspace approach which can exploit the global spatial-temporal constraints. We formulate the layer extraction problem as clustering in the low dimensional subspace where clusters become denser, better-defined, and thus more reliably identifiable. Such subspace also provides a constraint for detecting outliers in the local measurements, resulting in a robust layer extraction algorithm.

Linear subspace constraints have been successfully used in computer vision. Tomasi and Kanade [24] used the rank-3 constraint in Structure from Motion (SFM). Shashua and Avidan [20] derived the linear subspace of planar homographies induced by multiple planes between a pairs of views. Zelnik-Manor and Irani [28, 29] extended the results to multiple planes across multiple views, and applied such constraints to estimate the homographies of small predefined regions.

The subspace constraints to be exploited in this paper are derived from the relative affine transformations collected from large number of local regions across multiple frames. To avoid over-fitting in computing the local measurements, we design a local process to improve the motion estimation results, and to detect outliers at early stage. In the final stage of assigning pixels to layer models, we take

into account the spatial coherence by using *over-segmented* color regions instead of individual pixels. We assume that each over-segmented homogeneous color region is a planar patch. Such assumption is generally valid for images of natural scenes, and has been extensively used in motion analysis and stereo [3, 7, 23]. To deal with the case when such assumption is violated, we enforce the fact that the shape of each layer in the image domain is coherent or changes *gradually* across time [22].

2. Subspace of Relative Affine Homographies

This section shows that the homographies induced by 3D planar patches in a static scene, each represented by a column vector in the parameter space, reside in a low dimensional linear subspace. Such subspace comes from the fact that multiple planar patches in the scene share the common global camera geometry.

2.1. Projective Homography

Given two projective views of a static scene, any homography induced by a 3D plane in the scene can be described by [10]:

$$\mathbf{H}_{3\times 3} \cong \mathbf{A}_{3\times 3} + \mathbf{e}'\mathbf{v}^T \tag{1}$$

Here $\mathbf{v} = (v_1, v_2, v_3)^T$ defines the 3D plane[1]. $[\mathbf{e}']_\times \mathbf{A} = \mathbf{F}$ is any decomposition of the fundamental matrix $\mathbf{F}$, where $\mathbf{A}$ is a homography matrix induced by *some* plane ([10], pp.316).

Given k planes in the scene, we have k homography matrices $\mathbf{H}_i, i = 1, 2, ..., k$. Suppose we construct a matrix $\mathbf{W}_{9\times k}$ by considering each $\mathbf{H}_i$ as a column vector. The rank of $\mathbf{W}$ is known to be at most *four* [20]. In other words, all homographies between two projective views span a *four* dimensional linear subspace of $\Re^9$. This result was extended to the case of multiple projective views, and has been used to accurately estimate the homographies for small predefined planar patches [28].

2.2. Relative Affine Homography

In this section, we derive the subspace constraints for affine camera models. Affine camera [18] is an important model usable in practice. One advantage of affine camera is that it does not require calibration. Moreover, when perspective effect is small or diminishes, using affine camera model avoids computing parameters that are inherently ill-conditioned [19, 9].

Given uncalibrated cameras, it is known that the projective homography can only be determined up to an unknown scale. This is not the case for affine cameras. In affine camera, the 2D affine transformation can be *uniquely* determined, and we can rewrite Eq.(1) as [10, 15]:

$$\mathbf{m}_{2\times 3} = \mathbf{m}_\mathbf{r} + \mathbf{e}'\mathbf{v}^T. \tag{2}$$

Here $\mathbf{m}_\mathbf{r}$ is the affine transformation induced by the reference plane. $\mathbf{e}' = (e_1, e_2)^T$, where $(e_1, e_2, 0)$ is the direction of epipolar lines in homogeneous coordinate in the second camera. The 3-vector $\mathbf{v}$ representing the plane is independent of the second affine camera.

Notice that Eq.(1) has an unknown scale while Eq.(2) does not. We can define *relative affine transformation* as:

$$\Delta\mathbf{m} = \mathbf{m} - \mathbf{m}_r = \mathbf{e}'\mathbf{v}^T. \tag{3}$$

where $\mathbf{m}_r$ is the affine transformation induced by the reference plane. The reference plane can be either a real plane or a virtual plane.

2.3. Subspace of Relative Affine Homographies

We will show that the collection of all relative affine transformations across two or more views resides in a three dimensional linear subspace:

Proposition 1 *Given a static scene with k planar patches, a reference view ψ_r and other $F(F \geq 1)$ views $\{\psi_f | f = 1, ..., F\}$ of this scene, the collection of all relative affine transformations induced by these k planar patches between the reference view ψ_r and any other view ψ_f resides in a three dimensional linear subspace of $\Re^{6F}$.*

Proof: Denote the k affine transformations between the reference view and view f as $\mathbf{m}_{f,1}, ..., \mathbf{m}_{f,k}$. From Eq.(2) we have $\Delta\mathbf{m}_{f,i} = \mathbf{m}_{f,i} - \mathbf{m}_{f,r} = \mathbf{e}'_f\mathbf{v}_i^T$, where $\mathbf{v}_i = [v_{1,i}, v_{2,i}, v_{3,i}]^T$. Reshape each $\Delta\mathbf{m}_{f,i}$ into a 6×1 column vector, and stack them into a matrix $\mathbf{W}^f_{6\times k}$. The following factorization is obvious [20]:

$$\begin{aligned}\mathbf{W}^f_{6\times k} &= \begin{bmatrix} e_{f,1} & 0 & 0 \\ 0 & e_{f,1} & 0 \\ 0 & 0 & e_{f,1} \\ e_{f,2} & 0 & 0 \\ 0 & e_{f,2} & 0 \\ 0 & 0 & e_{f,2} \end{bmatrix} * \begin{bmatrix} v_{1,1} & \dots & v_{1,k} \\ v_{2,1} & \dots & v_{2,k} \\ v_{3,1} & \dots & v_{3,k} \end{bmatrix} \\ &= \mathbf{E}^f_{6\times 3} * \mathbf{V}_{3\times k} \end{aligned} \tag{4}$$

where $\mathbf{V}$ is common to all views. Therefore, we have:

$$\mathbf{W}_{6F\times k} = \begin{bmatrix} \mathbf{W}^1 \\ \mathbf{W}^2 \\ \dots \\ \mathbf{W}^F \end{bmatrix}_{6F\times k} = \begin{bmatrix} \mathbf{E}^1 \\ \mathbf{E}^2 \\ \dots \\ \mathbf{E}^F \end{bmatrix}_{6F\times 3} * \mathbf{V}_{3\times k} \tag{5}$$

[1] We ignore the degenerate case where a plane is projected into a line in the image.

The matrix dimension on the right-hand side of Eq.(5) implies that the rank of $\mathbf{W}$ is at most 3. $\diamond$

For the special *instantaneous* homography, it is known that there is a similar definition of relative projective homography and its subspace [29]. Instantaneous motion is not required here. The affine camera is allowed to undergo large motion (e.g., rotation) between frames.

The actual dimension of the subspace, i.e., the rank of W in Eq.(5), depends on the scene and the camera geometry, and could be *lower* than three. For example, if all planes in the scene are parallel to each other (not necessary front-parallel), or if there is only one plane in the scene, then the subspace dimension is *one* instead of three.

Another important fact is that the assumption of static scenes for deriving Eq.(5) is a sufficient condition but *not a necessary* one. This means that even with moving objects in the scene, we may still have a low dimensional linear subspace.

3. Algorithm for Layer Extraction Using Subspace

Our algorithm to layer extraction has the following four major steps: 1) construct the measurement matrix; 2) compute the low dimensional linear subspace using robust SVD; 3) cluster the local measurements into initial layer models in the subspace; 4) assign regions to layers, and post-process to refine the layer supports. Outliers in local measurements are detected in both Step 1 and 2.

In Step 1 and 4, we need to compute motion-compensated residuals (pixel-wised) for a given region. Given a multi-frame sequence, we need to sum up its motion compensated residuals between the reference frame and every other frame in the sequence. Although the use of multiple frames introduces temporal redundancy and decreases ambiguity, it unfortunately increases the chance for a region to be occluded, which in turn results in unreliable average residuals.

We use the temporal selection technique [14] to deal with the above occlusion problem. The motion compensated residual of region r between the reference frame and every other frame in the sequence are sorted in ascending order. Only the first half part of the sorted results are used to compute the final average residuals for region r.

3.1. Measurement Matrix Construction

To derive the subspace, we must collect local measurements (the affine motion of each local region/block) to build the measurement matrix $\mathbf{W}$ in Eq.(5). We divide the reference image (the image being segmented) into small $n \times n$ overlapped blocks ($n = 16$ in our experiments), where adjacent blocks are overlapped with each other by half of the block size [2]. The affine motion directly estimated from a small block often over-fits the data inside that block, and could differs greatly from the global optimal motion, i.e., the corresponding layer model. Simply increasing the block size reduces over-fitting effect, but at the same time increases the chance that such block contains multiple motions, and therefore no longer corresponds to a planar patch in the scene.

To deal with the above problem, we design a local process to gradually expanding each small block into a larger *k-connected component (KCC)*. Each *KCC* should cover the original starting block, while at the same time the residuals inside it should be less than some small pre-defined value ε. Being *k-connected* discards the thinly connected pixels, which are usually textureless and are from other layers other than the layer containing the starting block. The shape of a *KCC* could be irregular. Given a $n \times n$ block r in the reference frame, its KCC is derived by the following local process:

- Step 1 (motion estimation): Compute the homography of current KCC w.r.t. every frame in the sequence, and compute its motion compensated residuals using temporal selection [14].
- Step 2 (expanding/shrinking): Remove pixels with residuals larger than ε from current KCC. For each pixel p on the boundary of current KCC, a $n \times n$ ($n = 5$ in our experiments) test window w is centered at p. w is added to (removed from) KCC if the majority of the pixels inside w have residuals less (larger) than ε. In our experiments, Step 2 is repeat four times in one iteration to save computation time.
- Step 3 (evaluation): Enforce the *k-connected* requirement. If the resulted KCC becomes stable (the change of area or motion parameters is small enough) or the maximum number of iterations has been reached, then the block r is declared as an inlier. If the resulted KCC does not cover the original block r, then r is marked as an outlier. Otherwise, goto Step 1.

The parameter ε specifies the noise level to be tolerated. It depends on the geometry (the planarity) and the texture of the underlying layer, and is set to a conservative small value (15 in our experiments, with the pixel intensity range of $[0, 255]$). Fig.(1) shows an example of the above local process, given a ten-frame sequence. The small rectangle in Fig.(1a) shows the initial block r. Fig.(1b) shows the residuals [3] after compensating the motion estimated based

[2] Overlapped blocks can effectively deal with occlusions or other noises. For example, if a block is occluded (or contains multiple motions), it is often the case that at least one of its overlapped blocks is not occluded.

[3] The residuals are scaled up for visualization.

on pixels inside r. Although pixels inside r are well compensated, those pixels outside r but inside the same layer of r (flower bed) are not well compensated. Fig.(1c) shows the *k-connected component* after five iterations of the above process. Fig.(1d) is the final converged result. We can see that the pixels in the same layer are much better compensated in (c) and (d) than that in (b), which indicates a better global motion estimation result under the tolerable noise level ε. In our experiments, we find that three to five iterations are enough to obtain stabilized motion parameters. For our purpose, we do not require the local process to converge. Three iterations are enough to produce the desired local measurements. In Fig.(1e), the initial region r contains two motions (the tree and the flower garden). The resulted KCC does not cover r, therefore r is marked as an outlier.

3.2. Robust Subspace Computation

The subspace can be derived by factorizing the matrix $\tilde{\mathbf{W}}$ in Eq.(5) using SVD:

$$\tilde{\mathbf{W}}_{6F\times k} = \mathbf{U}_{6F\times 6F}\mathbf{\Sigma}_{6F\times 6F}\mathbf{V}^T_{6F\times k} \tag{6}$$

The diagonal of $\mathbf{\Sigma}$ contains the eigenvalues α_i of $\tilde{\mathbf{W}}$ in decreasing order. The actual rank of $\tilde{\mathbf{W}}$ depends on the camera and the planes in the scene, and is detected by [11]:

$$\frac{\sum_{i=0}^{d}\alpha_i^2}{\sum_{i=0}^{6F}\alpha_i^2} > t \tag{7}$$

where d is the rank of $\tilde{\mathbf{W}}$, and t determines the noise level we want to tolerate.

The linear subspace S is defined by the first d columns of U, which are the bases of the subspace. The remaining $(6F-d)$ columns form the bases of the residual space, which is orthogonal to S, and is denoted as $S^{\perp}$. SVD assumes Gaussian noise model and is sensitive to outliers, which often exist in motion data. The existence of subspace provides a constraint for outlier detection, making the subspace computation robust.

There are two kinds of outliers:

- data with extreme values that inflate the covariance matrix of $\tilde{\mathbf{W}}$.
- data that can not be represented by the subspace S, and have large projection values in the residual space $S^{\perp}$.

The detection of the above outliers is based on the Mahalanobis distance d_i^2 of the i-th data point m_i:

$$d_i^2 = (\mathbf{m_i}-\overline{\mathbf{m}})^T\mathbf{C}^{-1}(\mathbf{m_i}-\overline{\mathbf{m}}) = \sum_{p=1}^{6F} v_{i,p}^2 \tag{8}$$

where $\mathbf{C}$ is the covariance matrix of $\tilde{\mathbf{W}}$. The derivation of Eq.(8) is based on $\mathbf{C}^{-1} = \mathbf{U}\Sigma^{-2}\mathbf{U}^T$ and $\mathbf{m_i}-\overline{\mathbf{m}} = \mathbf{U}\Sigma\mathbf{v_i}$.

Under the assumption that data are sampled from an underlying elliptic normal distribution with covariance matrix $\mathbf{C}$, d_i^2 follows χ^2 distribution with N degrees of freedom [13]. All data samples with d_i^2 lie outside the p-th percentage point of the corresponding χ^2_N distribution ($N = 6F$ is the degrees of freedom) are marked as outliers.

A problem with the above distance measurement d_i^2 is that it may not give enough weight to the bases of the residual space $S^{\perp}$, which usually identify the outliers that violate the correlation structure imposed by the bulk of data, but not necessarily inflate the covariance matrix. For this reason, we also look at the residual space $S^{\perp}$ by examining the following value:

$$o_i^2 = \sum_{p=d+1}^{6F} v_{i,p}^2$$

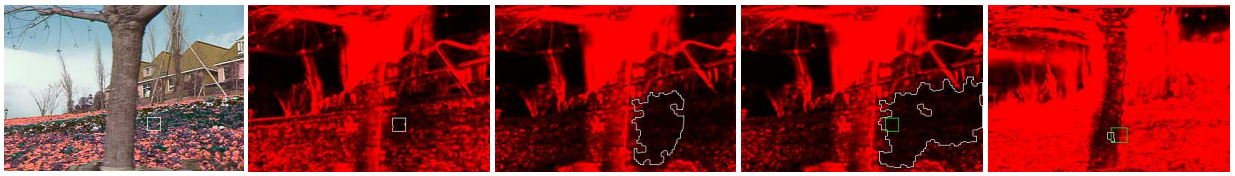
where d is the rank of $\tilde{\mathbf{W}}$; and o_i^2 follows χ^2_N distribution [8], with degrees of freedom $N = 6F - d$.

Our algorithm for robust subspace computation consists of the following steps:

- Step 1: Use SVD to compute an initial subspace.
- Step 2: Compute d_i^2 and o_i^2 for each data point. Data whose d_i^2 and o_i^2 are outside the p-th confidence interval of χ^2 distribution are marked as outliers.
- Step 3: Apply SVD to the set of inliers to recompute the subspace.
- Step 4: Repeat Step 2 and 3 until the set of inliers stabilizes.

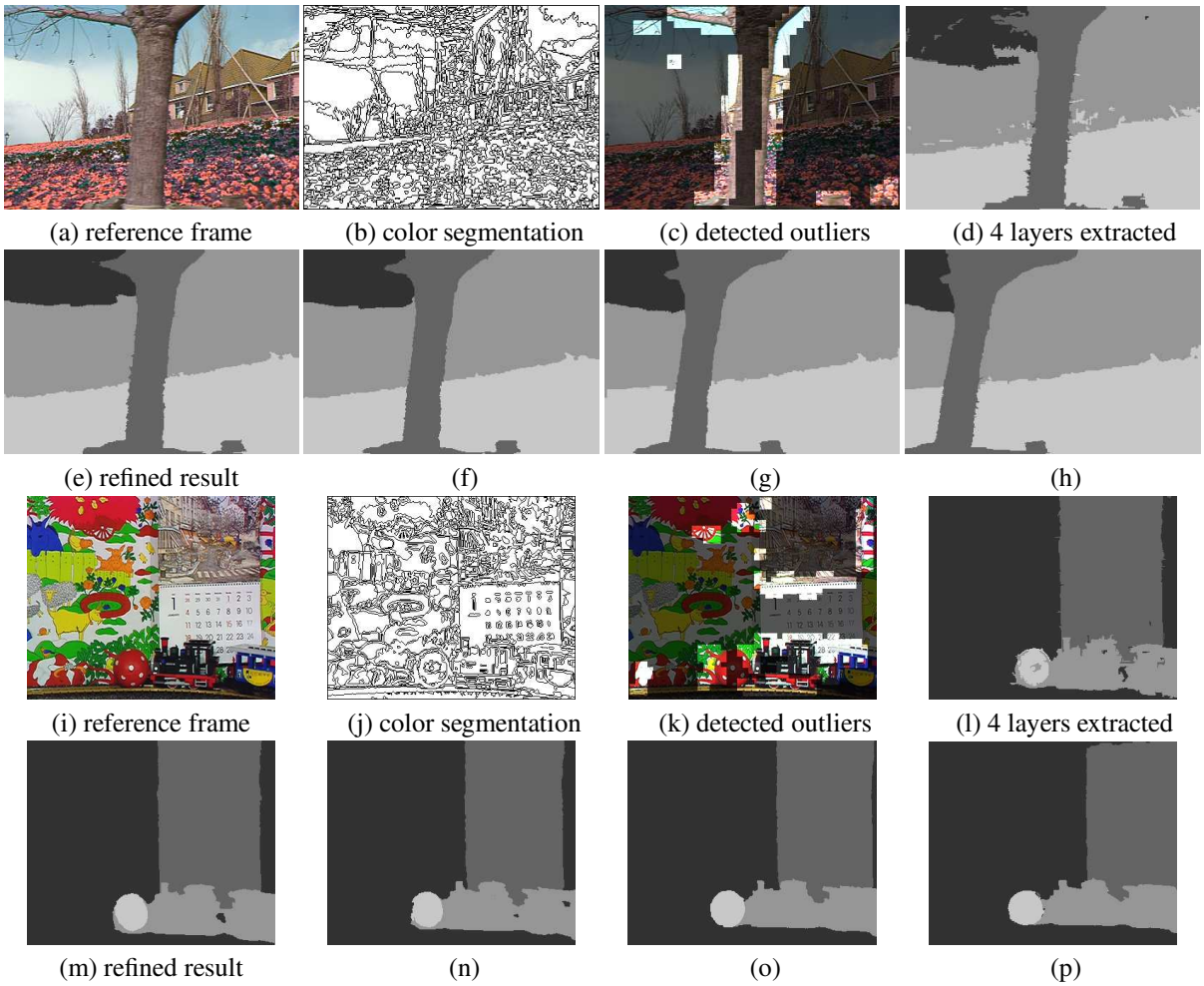
3.3. Model Initialization by Subspace Clustering

We now apply a clustering algorithm to the data points (projected local measurements) in the d-dimensional subspace for initial layer models (cluster centers). We adopt the mean shift based clustering algorithm, proposed by Commaniciu and Meer [4, 5], because: (1) it is non-parametric and robust; and (2) it can automatically derive the number of clusters and the cluster centers. Refer to [4, 5] for a clear description and details on this algorithm. Here we point out two implementation details. First, the initial seeds for the mean shift algorithm are chosen to uniformly distribute across the image domain. Second, the window radius r can be derived from the covariance matrix of $\tilde{\mathbf{W}}$. According to [4], in our experiments it is to be set proportional to $\sigma = \sqrt{trace(cov(\tilde{\mathbf{W}}))}$. We have found by experiments that a wide range of r produces the desired results, due to the use of multiple frames.



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

Figure 1. Deriving the *k-connected component (k=5)*: (a) initial region r given by the white rectangle; (b) residuals after compensation based on the motion estimated using the initial region r; (c) residuals and *k-connected component* after five iterations of the local process; (d) converged residuals and *k-connected component*; (e) outlier region (final KCC does not cover the original block).



(a) reference frame (b) color segmentation (c) detected outliers (d) 4 layers extracted

(e) refined result (f) (g) (h)

(i) reference frame (j) color segmentation (k) detected outliers (l) 4 layers extracted

(m) refined result (n) (o) (p)

Figure 2. Layer extraction results. (a): reference frame; (b): color over-segmentation result; (c) detected outliers in local measurements; (d): four initial layers extracted; (e): refined results on (d); (f-h): final results on another three frames of the flower-garden sequence. (i)–(p) are the results on mobile & calendar sequence.

3.4. Assigning Color Regions to Models and Layer Refinements

Once we are given the initial layer models (the cluster centers), we can assign each color segment to the best layer model. To utilize the spatial coherence, we assign over-segmented color regions to layers, instead of individual pixels. To deal with occlusions, temporal selection [14] is used here to determine the best layer model for a color region. Very fine-grained over-segmentation [4] has been used to assure that each homogeneous color segment is contained inside only one layer. However, we still observe some spurious regions, largely due to noises in the images, or the disparities between color segment boundaries and motion boundaries in some places. Such spurious regions appear at different random positions in different frames, and can be removed by using the fact that each layer corresponds to a *rigid and consistent* plane in the scene, such that its shape in the image domain is coherent or changes *gradually* across time [22].

4. Experimental Results and Applications

We apply our algorithm to two real image sequences: *flower garden* and *mobile & calendar*. To segment each frame, its 10 neighbored frames are used as input to our algorithm. Both sequences were found to have a three-dimensional subspace of the original space $\Re^{60}$, with the noise level parameter in Eq.(7) set to $t = 97\%$.

4.1. Layer Extraction Results

Fig.(2a) shows one frame of the *flower garden* sequence, where the scene is static and the camera is translating approximately horizontally. Fig.(2b) shows the very fine-grained color over-segmentation result. Even the tree, which has visually homogeneous color, has been over-segmented into many color regions. Fig.(2c) highlights the outliers detected by our algorithm. Motions estimated from blocks containing multiple motions are mostly identified as outliers. Fig.(2d) shows the four cluster centers (with assigned color regions) derived by the mean shift clustering in the subspace. Color segments that are occluded in some frames are still assigned to the right layers due to the use of temporal selection in this step. Fig.(2e) shows the layers refined by enforcing shape coherence across time. Fig.(2f-h) show the layer extraction results on three other frames in this sequence. Four layers corresponding to the house, flower bed, tree, and tree branch are consistently extracted in this sequence.

The rest of the images in Fig.(2) show the results of applying the same algorithm and parameters to the *mobile & calendar* sequence. In this sequence, the train is pushing the ball (rotating), and the calendar is moving up. At the same time, the camera is zooming out and tracking the train. Four layers are identified, namely the background, the ball, the train, and the calendar. Notice that the ball has small support, but distinct motion, which tends to be missing in other previous work, for example in [1].

4.2. Application: Video Compression

We show the preliminary results of applying layer representation to video compression. Each input video segment is compactly represented by layer mosaics [4]. Fig.(3a & b) shows the four layer mosaics of the flower garden sequence (30 frames). We are able to compress the original video sequence from about $7MB$ to about $40KB$ [5]. Fig.(3c) shows the recovered frame based on the layer representation, whose original frame is shown in Fig.(2a). The remaining images in Fig.(3) show the similar results for *mobile & calendar* sequence. We are able to compress it from about $9MB$ (30 frames) to about $45KB$. Note that higher compression ratio can be achieved with longer sequence.

5. Conclusions

We have presented a robust subspace approach to extracting 2D layers from image sequence. Our subspace approach has the following advantages: (1) clusters in the subspace become denser and better-defined; (2) robustness is achieved by both the local process (Section 3.1) and the subspace constraint; (3) spatial and temporal constraints from multiple frames are simultaneously utilized by using subspace; and (4) noise in estimated motion is reduced by subspace projection. Together with mean shift based clustering algorithm, we have demonstrated that the use of low dimensional subspace leads to good layer descriptions on real images.

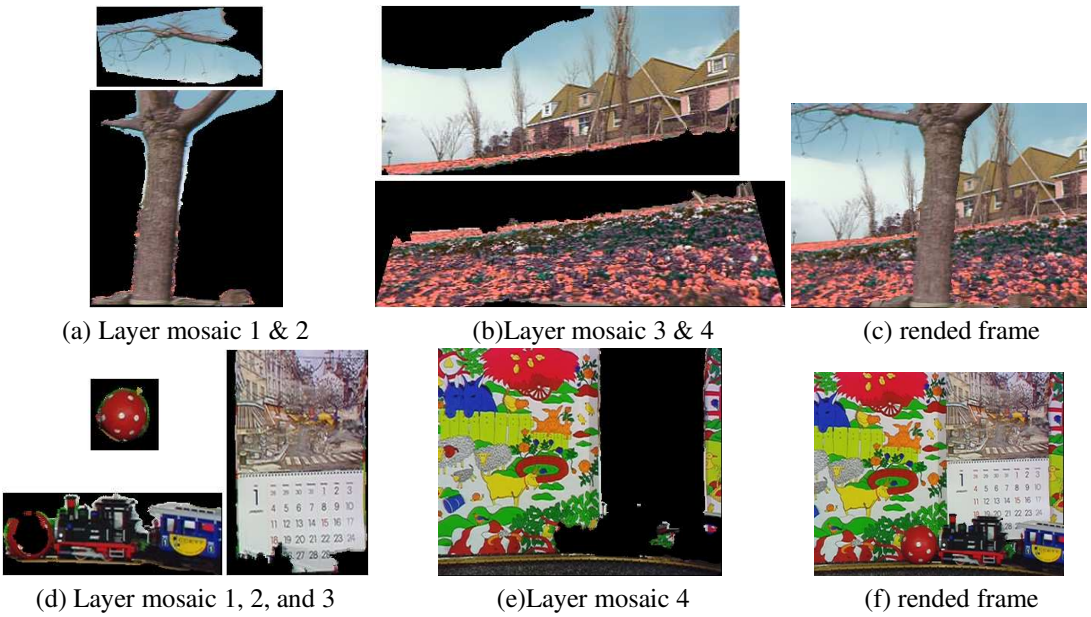
Finally, note that in this paper we model layers with 2D parametric motion. In such model, an object (e.g., human body) with non-rigid or articulate motion will be segmented into multiple layers. To be able to segment such object as one single layer, we need to increase the complexity of the layer model and combine other cues in addition to motion cues, which is still not a well-defined task, and is part of our future work.

References

[1] S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV95*.

[4] Sprite VOP in MPEG-4

[5] We do not encode the residual signals for the layers.



(a) Layer mosaic 1 & 2 (b)Layer mosaic 3 & 4 (c) rended frame

(d) Layer mosaic 1, 2, and 3 (e)Layer mosaic 4 (f) rended frame

Figure 3. Layer mosaics and synthesized frame. (a) & (b) layer mosaics of the flower garden sequence; (c): a recovered frame based on the layer representation; (d)–(c) same results on mobile & calendar sequence.

[2] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR98*, 1998.

[3] M. J. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10), 1996.

[4] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *CVPR97*.

[5] D. Comaniciu and P. Meer. Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 2(1), 1999.

[6] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *PAMI*, 17(5), May 1995.

[7] M. Gelgon and P. Bouthemy. A region-level graph labeling approach to motion-based segmentation. In *CVPR97*.

[8] R. Gnanadesikan and J. Kettenring. Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, 28:81–124, March 1972.

[9] C. Harris. Structure-from-motion under orthographic projection. In *ECCV90*.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[11] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV99*.

[12] A. Jepson and M. Black. Mixture models for optical flow computation. In *CVPR93*.

[13] I. T. Jolliffe. *Principal Components Analysis*. Springer, 1986.

[14] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR 2001*.

[15] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *CVPR 2001*.

[16] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *CVPR01*.

[17] M. Lee, W. Chen, C. Lin, C. Gu, T. Markoc, S. Zabinsky, and R. Szeliski. A layered video object coding system using sprite and affine motion model. *CirSysVideo*, 7(1), 1997.

[18] J. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[19] L. Shapiro. *Affine Analysis of Image Sequences*. Cambridge University Press, 1995.

[20] A. Shashua and S. Avidan. The rank 4 constraint in multiple (over 3) view geometry. In *ECCV96*.

[21] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV'98*.

[22] H. Tao, H. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *PAMI*, 24(1):75–89, January 2002.

[23] H. Tao and H. S. Sawhney. Global matching criterion and color segmentation based seereo. In *WACV2000*.

[24] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2), 1992.

[25] P. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. In *ICCV99*.

[26] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3(5), 1994.

[27] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR97*.

[28] L. Zelnik-Manor and M. Irani. Multi-view subspace constraints on homographies. In *ICCV99*.

[29] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *PAMI*, 22(10), 2000.