A Subspace Approach to Layer Extraction

Qifa Ke and Takeo Kanade
School of Computer Science
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 an 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. Layers in the input images will be mapped in the subspace, where it is proven that they form well-defined clusters and can be reliably identified by a simple mean-shift based clustering algorithm. Global optimality is achieved since all valid regions are simultaneously taken into account, and noise can be effectively reduced by enforcing the subspace constraint. 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 [23, 14, 2]. 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 (or non-parametric model defined by dense smooth flow field [24]). 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 three major issues of layer extraction are: (1) determination of the number of layers; (2) the model-based motion of each layer; and (3) the assignment of pixels to layers. Various approaches have been proposed for layer extraction based on motion, such as mixture model estimation with Expectation-Maximization (EM) algorithm [13, 1, 25, 24, 21], and pixel or region grouping based on a certain affinity criterion using k-means algorithm [23] or normalized graph cut [18].

Initialization (the number of models and the motion for each model) is an important but difficult step for EM approach [18, 21]. Without good initialization, EM algorithm may not converge to desired optimal solutions. A typical initialization method [1] is to divide the image into a fixed number of tiles, and use them as the initial layers for the EM algorithm. Followed by each EM iteration is the application of MDL principle to determine the number of models, which is realized as an exhaustive search in [1]. However, the initial regular tiling does not guarantee the existence of dominant motion inside each initial or intermediate layer[1], which is required for the robust motion estimation of each intermediate layer in the M-step [1]. Moreover, if one real layer is divided into different tiles, and if those tiles have different dominant motions (or without any dominant motion at all), then such an unlucky layer becomes hard to be extracted.

Grouping pixels based on local measurement does not have the similar initialization difficulty. However, grouping based on pure local measurement ignores the global constraints. Moreover, grouping in high dimensional space is often unreliable given noisy local measurements.

In this paper, we present a low dimensional 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.

Linear subspace constraints have been successfully used in computer vision. Tomasi and Kanade [20] used the rank-3 constraint in SFM. Shashua and Avidan [17] derived the linear subspace of planar homographies induced by multiple planes between a pairs of views. Zelnik-Manor and Irani [26, 27] extended the results to multiple planes across multiple views, and applied such constraints to estimate the homographies of small regions.

[1] The presence of dominant motion of the whole image is not required.

The subspace constraints to be exploited in this paper are derived from the relative affine transformations collected from homogeneous color regions. Our algorithm assumes that each 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 [4, 25, 9, 19].

Our subspace approach has the following advantages: (1) clusters in the subspace become denser and better-defined; and (2) global optimality is achieved by simultaneously taking into account all valid regions; and (3) noise in estimated motion is reduced by subspace projection, and global geometry constraint is enforced.

2. Subspace of planar homographies

This section shows that the homographies induced by 3D planar patches in a static scene, each one as 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. The redundancy is high since there exists a large number of homogeneous color regions in real images, most of which can be approximated as planar patches.

2.1. Subspace of projective homographies

Given two projective views of a static scene, any homography induce by a 3D plane in the scene can be described by [11]:

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

Here $\mathbf{v} = (v_1, v_2, v_3)^T$ defines the 3D plane[2]. $[\mathbf{e}']_\times \mathbf{A} = \mathbf{F}$ is any decomposition of the fundamental matrix $\mathbf{F}$, where $\mathbf{A}$ is a homography matrix induced by *some* plane ([11], 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* [17]. 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 planar patches [26].

2.2. Subspace of relative affine homographies

Affine camera [15] 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 can avoid computing parameters that are inherently ill-conditioned [16, 10].

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

Eq.(1) holds for affine camera as well ([11], pp.350). 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 (see the proof in appendix):

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

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. $\mathbf{v}$ is a 3-vector, and is independent of the second affine camera.

Notice an important difference between Eq.(1) and (2). Eq.(1) has an unknown scale while Eq.(2) does not. Therefore, we can define *relative affine transformation* as:

$$\Delta\mathbf{m} = \mathbf{m} - \mathbf{m}_r. \tag{3}$$

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.

We will show that the collection of all relative affine transformations across more than two views resides in a three dimensional linear subspace of $\Re^6$:

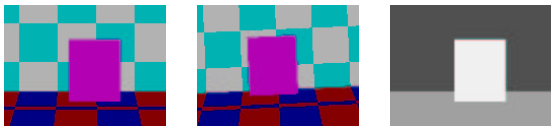
Proposition 1 *Given a static scene with k planar patches, a reference view ψ_r and another $F(F \geq 1)$ views $\{\psi_f | f = 1, ..., F\}$ of this scene, the collection of all relative affine transformations induced by these k planar patches between the reference view ψ_r and any other view ψ_f resides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

Proof: Denote the k affine transformations between reference view and view f as $\mathbf{m}_1, ..., \mathbf{m}_k$. From Eq.(2) we have $\Delta\mathbf{m}_i = \mathbf{m}_i - \mathbf{m}_r = \mathbf{e}'\mathbf{v}_i^T$, where $\mathbf{v}_i = [v_{1,i}, v_{2,i}, v_{3,i}]^T$. Reshape each $\Delta\mathbf{m}_i$ into a 6×1 column vector, and stack them into a matrix $\mathbf{W}^f_{6\times k}$. The following factorization is obvious [17]:

$$\begin{aligned}\mathbf{W}^f_{6\times k} &= \begin{bmatrix} e^f_1 & 0 & 0 \\ e^f_2 & 0 & 0 \\ 0 & e^f_1 & 0 \\ 0 & e^f_2 & 0 \\ 0 & 0 & e^f_1 \\ 0 & 0 & e^f_2 \end{bmatrix}_{6\times 3} * \begin{bmatrix} v_{1,1} & ... & v_{1,k} \\ v_{2,1} & ... & v_{2,k} \\ v_{3,1} & ... & v_{3,k} \end{bmatrix}_{3\times k} \\ &= \mathbf{E}^f_{6\times 3} * \mathbf{V}_{3\times k}\end{aligned}$$

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

$$\mathbf{W}_{6F\times k} = \begin{bmatrix} \mathbf{W}^1 \\ \mathbf{W}^2 \\ ... \\ \mathbf{W}^F \end{bmatrix}_{6F\times k} = \begin{bmatrix} \mathbf{E}^1 \\ \mathbf{E}^2 \\ ... \\ \mathbf{E}^F \end{bmatrix}_{6F\times 3} * \mathbf{V}_{3\times k} \tag{4}$$



(a) (b) (c)

449.0	261.2	0.006	0.0007	0.0002	0.0001

(d)

Figure 1. Results on synthetic sequence, where both camera and objects are moving independently: (a) and (b) two frames of the synthetic sequence; (c) the layer map by clustering in the 2-D subspace; (d) the eigenvalues of matrix $\mathbf{W}_{6\times 31}$.

The matrix dimension on the right-hand side of Eq.(4) implies that the rank of $\mathbf{W}$ is at most 3. $\diamond$

From Eq.(4) we can see that the subspace comes from the fact that multiple planes share the common camera geometry, i.e., the direction of parallel epipolar lines. The matrix $\mathbf{W}$ is built from the motions of planar patches. We can exploit high redundancy by using subspace since there exists a large number of homogeneous color regions in real images, many of which are planar patches. Multiple views have more redundancy. For the special *instantaneous* homography, there is a similar definition of relative projective homography and its subspace [27].

2.3. Dimensionality of subspace

The actual dimension of the subspace, i.e., the rank of W in Eq.(4), depends on the scene planes and the camera geometry, and could be *lower* than three. For example, if all planes 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 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.

To verify the above observation, let us consider the following situations. A 3D scene consists of three planes[3], with the table plane stationary and foreground and background planes moving upward and downward independently. At the same time, a pinhole camera is zooming out, translating horizontally, and rotating about its optical axis. Under such camera motion, each plane in the scene will induce an affine transformation. Fig.(1) shows the two rendered frames. With $F = 1$, and $k = 31$ patches (1 on foreground plane, 15 on background plane, 15 on table plane), the eigenvalues of $\mathbf{W}_{6\times 31}$, shown in Fig.(1d), clearly show that the dimension of subspace is two. In the next section we will describe in details how to derive the subspace.

[3]Each plane is made of many color patches.

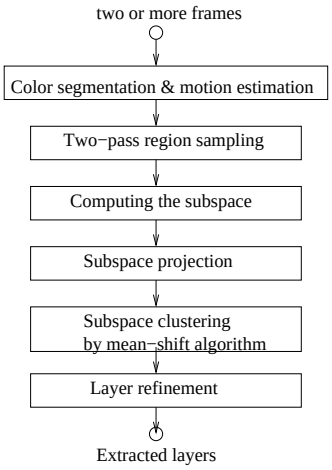


Figure 2. Overview of layer extraction algorithm.

3. Layer extraction algorithm

Fig.(2) shows the steps of layer extraction algorithm. The input is two or more images, with one of them selected as the reference view frame. The reference image is segmented based on static color information. It is in general safer to over-segment, so that each segment corresponds to a single planar patch. Then an affine or translational motion is estimated with respect to each other frame for each color segment. Then the region sampling algorithm will select valid color segments, and the affine motions from these selected color segments are used to compute the linear subspace. Data points in the subspace are then generated by projecting the affine motion into the subspace. We use the mean-shift based clustering technique [5, 7] to derive the initial layers. Finally, the un-selected color segments are assigned to layers in the layer refinement step.

3.1. Color segmentation and motion estimation

Our layer extraction algorithm assumes that pixels inside each color segment belong to the same layer, and the motion of each color segment can be described by a 2D parametric model, such as affine or projective homography[4]. We use the color segmentation algorithm proposed by [6]. Since color segmentation is not our final goal, over-segmentation has been used here in order to assure the validity of the above assumption to the largest extend. Such assumption is generally valid for over-segmented images of natural scenes, and has been successfully used in motion analysis and stereo [4, 25, 19].

For every color segment in the reference frame, we directly estimate a parametric motion using a simple hierarchical model-based approach with robust estimation [3, 1,

[4]Note that color segmentation is applied only on the reference image. We directly estimate the motion of each region without doing region correspondence between reference image and other images.

4]. In our experiment, translational or affine model is estimated depending on the area support of each color segment.

Large color segments usually still have enough intensity variation to estimate affine motions. For a segment with little intensity variation, a translational motion can still be reliably estimated from the boundaries of color segment, if there is not occlusion.

3.2. Two-pass region sampling

To derive the subspace, we must select regions to be used to build the matrix $\mathbf{W}$ in Eq.(4). Those regions must be the ones for which affine motions are estimated, and in general, they should uniformly distribute over the reference frame, so that each layer in the image domain will have enough samples and form a dense cluster in the feature space where clustering is performed.

A straightforward region sampling method is to divide the reference frame into small $n \times n$ blocks, and then select the blocks where an affine motion can be estimated [23]. Since affine motions are usually not available or erroneous in *small* textureless blocks, a layer containing large homogeneous color regions will not have enough number of samples to become a single dense cluster in the feature space. On the other hand, a layer with rich texture may have much more samples and the clustering algorithm may bias toward such layer.

To deal with the above problems while at the same time uniformly sample the reference image, we design a two-pass sampling approach based on color segmentation, as illustrated in Fig.(3). In the first pass, color segments for which affine motions have been estimated are selected as region samples[5]. The remaining un-selected areas are used in the second pass. Such remaining areas usually have rich texture and contain many small color segments where only translational motions are available. In the second pass, the reference image is divided into $n \times n$ blocks ($n = 20$ in our experiments). For each block containing more than 80% of un-selected pixels, we re-estimate an affine motion using the un-selected pixels inside this block. If the intensity residual of such estimated motion is small, the un-selected color segments inside such block are chosen as region samples.

3.3. Computing subspace

Computing the subspace of homographies involves building and factorizing the matrix $\mathbf{W}$ in Eq.(4), which has been constructed from the affine transformations of the k selected region samples: $\mathbf{m}_i, i = 1, 2, ..., k$.

There are three important implementation details in building $\mathbf{W}$:

- We can choose one color region with large area support and good motion estimation as the reference plane. In practice, we found the average transformation $\overline{\mathbf{m}} = \frac{1}{k}\sum_{i=1}^{k} \mathbf{m}_i$ serves as a good reference affine transformation induced by some "virtual" plane[6].

- The area of each selected color segment is to be taken into account. For a selected color segment $\mathbf{m}_i$ containing n pixels, we reshape $\Delta\mathbf{m}_i$ into a 6×1 column vector, and then put n columns of $\Delta\mathbf{m}_i$ into $\mathbf{W}$[7]. In other words, regions with larger area have larger weights. Obviously adding such weight does not change the rank of $\mathbf{W}$.

- We scale the different components in the affine transformation, such that a unit distance along any component in the parameter space corresponds to approximately a unit displace at the image boundaries [23]. Such scaling makes the subspace approximately *isotropic*. We use the image width as the scale factor. Specifically, the matrix $\mathbf{W}_{6\times k}$ is left-multiplied by the following scale matrix:

$$\mathbf{S} = \begin{bmatrix} w & 0 & 0 & 0 & 0 & 0 \\ 0 & w & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & 0 & w & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Again, such linear transformation does not change the rank of $\mathbf{W}$, or the dimension of the subspace. Let us denote $\tilde{\mathbf{W}} = \mathbf{S}\mathbf{W}$. In practice, we found that $\mathbf{S}$ is not a sensitive parameter. The final results do not change for a wide range of the w in matrix $\mathbf{S}$.

We use SVD algorithm to factorize the matrix $\tilde{\mathbf{W}}$:

$$\tilde{\mathbf{W}}_{6\times k} = \mathbf{U}_{6\times 6}\mathbf{\Sigma}_{6\times 6}\mathbf{V}^T_{6\times k} \quad (5)$$

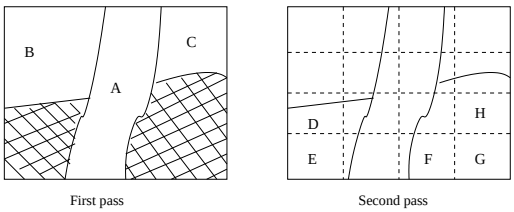


Figure 3. Two-pass sampling. Solid lines in the figures show the boundaries of color segments. In the first pass, color segments A, B, C are selected. The remaining color segments are small. In the second pass, the image is divided into $n \times n$ blocks. Blocks D-H are selected since they contain more than 80% of un-selected pixels. Affine motion for each selected block is estimated based only on the unselected pixels inside it.

[5] A simple outliers detection is applied here. Regions with large registration error are considered as outliers.

[6] Notice that $\overline{\mathbf{m}}$ is induced by some world plane (either real or virtual) if and only if there exists $\mathbf{F} = [\mathbf{e}']_\times \overline{\mathbf{m}}$, where $\mathbf{F}$ is the fundamental matrix[11].

[7] If we do not use the average transformation $\overline{m}$ as reference, we need to subtract mean from each column.

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 [12]:

$$sqrt(\frac{\sum_{i=0}^{d}\alpha_i^2}{\sum_{i=0}^{6}\alpha_i^2}) > t \quad (6)$$

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

The linear subspace is defined by the first d columns of U, which are the bases of the subspace. The motions of the region samples are projected into this subspace as $\Sigma_{d\times d}V^T_{d\times k}$, where each column becomes a feature point in the d-dimensional subspace.

3.4. Layer initialization by subspace clustering

We now apply a clustering algorithm to the data points in the d-dimensional subspace for initial layers. The mean-shift based clustering algorithm, proposed by Commaniciu and Meer [6, 7], has been successfully applied to color segmentation and non-rigid object tracking [6, 8]. We adopt this algorithm because: (1) it is non-parametric and robust; (2) it can automatically derive the number of clusters and the cluster centers. Refer to [6, 7] for a clear description and details on this algorithm.

A critical parameter in this clustering algorithm is the window radius r of mean shift. This parameter determines the resolution of segmentation. We will show results over a range of r.

3.5. Layer refinement & post-processing

Once we have the initial layers given by subspace clustering, we re-estimate an affine motion for each initial layer by using all of the region samples inside that layer. Then we re-assign every color segment[8] to the layer that predicts its motion best. This layer refinement is similar to one EM iteration in its goal, but without the probabilistic notion.

There are some spurious small regions, largely due to outliers. We have an optional post-processing step to remove such regions, by assigning them to their neighbors with similar motions. Such post-processing is desirable since a small number of compact layers are preferable for applications such as video compression.

4. Experimental results

This section presents the experimental results of two real image sequences: *flower garden* and *mobile & calendar*.

There are two parameters that need to be specified. One is the noise level parameter t in Eq.(6) for determining the dimension of the subspace. In the following experiments, both sequences were found to have a two-dimensional subspace with $t = 95\%$. The other parameter is the window radius r. It is a critical parameter in the mean-shift based clustering algorithm. The value of this parameter can be derived from the covariance matrix of $\tilde{\mathbf{W}}$. According to [6], in our experiments it is to be set proportional to $\sigma = \sqrt{trace(cov(\tilde{\mathbf{W}}))}$. We have found by experiments that $r = 0.3\sigma$ produces the desired results. We will also show different layer extraction results by varying r over a wide range of $[0.3\sigma, 1.3\sigma]$.

[8] Including the color segments that are not selected in the two-pass region sampling step.

4.1. *flower garden* sequence

Fig.(4a) and Fig.(4b) show two frames of the *flower garden* sequence, where the scene is static and the camera is translating approximately horizontally.

Fig.(4c) shows the color segmentation result on the reference image by applying the color segmentation algorithm with over-segmentation class proposed in [6]. Fig.(4d) shows the region samples selected by the two-pass sampling algorithm, and the initial layers via mean-shift clustering in the subspace. The black regions are un-selected regions. Notice that most of the occlusion regions are not selected, perhaps due to the two-pass sampling algorithm. Four layers (which roughly correspond to tree, branch, house, and flower bed) have been identified by the clustering algorithm, with window radius $r = 0.3\sigma_{garden}$, where $\sigma_{garden} = 4.5$. The tree layer and the branch layer contain large color segments and are easier to extract. Notice that the flower bed and the house consist of mostly small regions. The subspace clustering successfully identifies them as two separate layers.

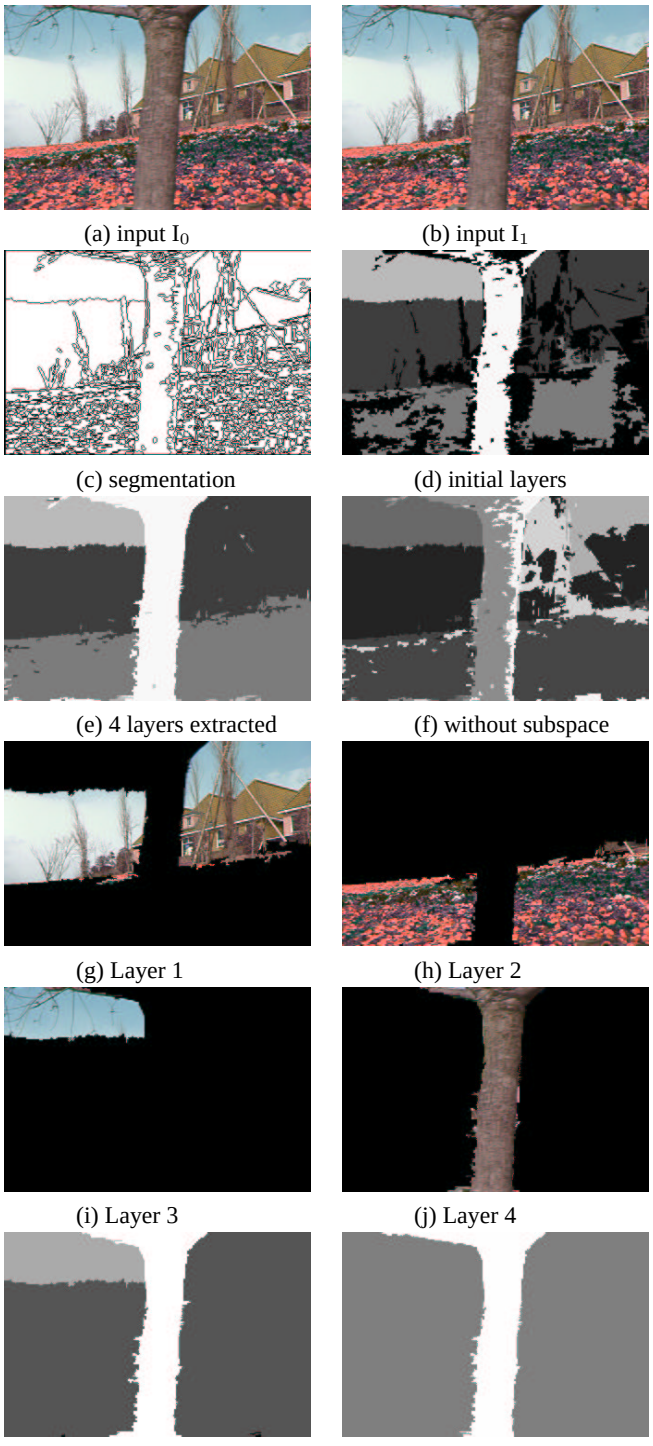
Fig.(4e) shows the four layers after the layer refinement step but without post processing. Every initially unselected color segments has been assigned to one of the layers.

Fig.(4g-j) shows the four layers where the small spurious regions are assigned to neighbor regions based on motion affinity by the post processing step.

4.2. *mobi* sequence

The *mobile & calendar* sequence is used to show that static scene assumption in the analysis of Section 2 is a sufficient condition but *not a necessary one*. In this sequence, the train is pushing a rotating ball leftwards, and the calendar is pulled upwards, while camera is panning and tracking the train.

Fig.(5d) shows the region samples and initial layers by mean shift clustering with $r = 0.3\sigma_{mobi}$, where $\sigma_{mobi} = 3.2$. Again we notice that most of the occlusion regions are in the un-selected black regions. Fig.(5e) shows the result of layer refinement but without post processing. Note that the ball (in the lower middle) is extracted successfully. Although its area support is small, its motion is distinct and it forms a separate cluster in the subspace. In previous work of layer extraction on this sequence, for example in [1], the ball layer tends to be missed since its motion is not dominant in any initial or intermediate layer.



(a) input I_0 (b) input I_1

(c) segmentation (d) initial layers

(e) 4 layers extracted (f) without subspace

(g) Layer 1 (h) Layer 2

(i) Layer 3 (j) Layer 4

(k) (m)

Figure 4. Results of *flower garden* sequence. (a) and (b) Two frames of the sequence; (c) Color segmentation map; (d) Selected regions and initial layer map by clustering in the 2D subspace, where black indicates un-selected regions; (e) Layers after refinement; (f) Noisy layers extracted using the original six dimensional parameter space instead of subspace; (g)-(j) Four layers extracted after post-processing; (k) & (m) Layer maps by increasing the window radius of mean-shift algorithm.

4.3. Increasing window radius

In this experiment, we vary the window radius r to see how the segmentations of different resolutions are derived. Fig.(4k) and (4m) show the layer maps obtained when increasing the window radius to $0.7\sigma_{garden}$ and $1.3\sigma_{garden}$ respectively[9]. Notice that in Fig.(4m), part of the branch layer is erroneously merged into the background layer. Fig.(5k) and (5m) are for *mobi* sequence.

The functionality of parameter r is similar to the "coding length" of MDL [1]. However, r is easier to understand and is more natural to set, in a way similar to the variance of Gaussian in [25].

4.4. Comparison with clustering without using subspace

To demonstrate the advantages of using subspace, we also show the results of layer extraction without using subspace. To make the window radius comparable in both cases, we have scaled them by the following factor:

$$s = \frac{sqrt(\alpha_0^2 + \alpha_1^2 + ... + \alpha_5^2)}{sqrt(\alpha_0^2 + \alpha_1^2)} \tag{7}$$

where α_i's are the eigenvalues of $\tilde{\mathbf{W}}$.

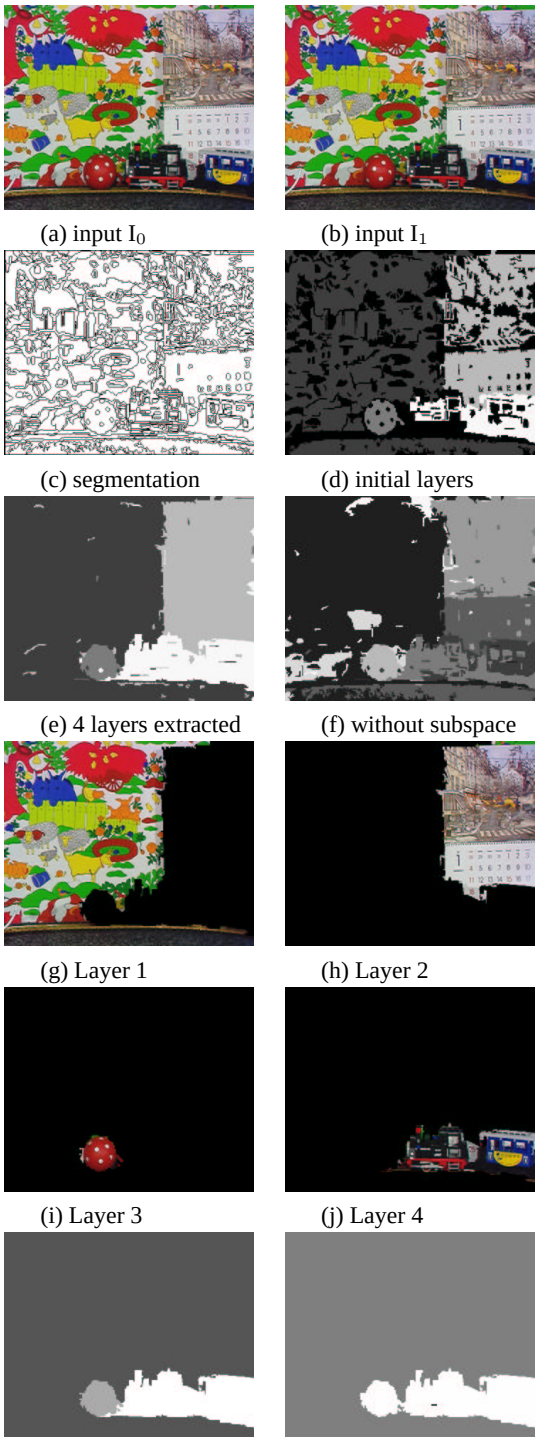
Fig.(4f) and Fig.(5f) are the results of clustering in the original six-dimensional affine parameter space, with $r = s \times 0.3\sigma$. Some layers are split into two or more layers, possibly due to the fact that in the high dimensional space, the data are sparser and the cluster are not as well defined as in the low dimensional space. Also some regions are assigned to wrong layers.

5. Conclusion

We have presented a subspace approach to extracting 2D layers from image sequence. The low dimensional subspace makes the cluster better-defined and easier to extract. It also effectively reduces noise introduced in the step of motion estimation. The local spatial coherence is also exploited by assigning color segments to layers, instead of assigning individual pixels. Together with the mean-shift based clustering algorithm, we have demonstrated that the use of low dimensional subspace leads to good layer descriptions on real images.

The results shown in this paper are based only on two views. For multiple views, the algorithm presented in this paper can be readily applied without any change. The only difference between two-view and multiple-view cases is the format of matrix $\tilde{\mathbf{W}}$. The multi-view algorithm will produce better results, as long as for each color segment in the reference view, its motions to other views can be estimated. We are currently experimenting with the case of multiple views.

[9] Further increasing r will eventually results in a layer map with only one layer in it.



(a) input I_0 (b) input I_1

(c) segmentation (d) initial layers

(e) 4 layers extracted (f) without subspace

(g) Layer 1 (h) Layer 2

(i) Layer 3 (j) Layer 4

(k) (m)

Figure 5. Results of *mobile & calendar* sequence. (a) and (b) Two frames of the sequence; (c) Color segmentation map; (d) Selected regions and initial layer map by clustering in the 2D subspace, where black indicates un-selected regions; (e) Layers after refinement; (f) Noisy layers extracted using the original six dimensional parameter space instead of subspace; (g)-(j) Four layers extracted after post-processing; (k) & (m) Layer maps by increasing the window radius of mean-shift algorithm.

In this paper, we used SVD to compute the subspace. Given Gaussian noise, SVD achieves global optimality in the sense of least square error. If the data contain outliers, robust algorithm can be used for deriving the subspace [22].

Acknowledgements

Thanks go to Harry Shum, Simon Baker, Martial Hebert, and Alan Lipton for helpful discussions and comments on the paper. We would also like to thank the anonymous reviewers for their feedback. This work was supported in part by DiamondBack Vision, Inc.

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*.

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

[3] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV92*.

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

[5] Y.Z. Cheng. Mean shift, mode seeking, and clustering. *PAMI*, 17(8), 1995.

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

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

[8] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR00*.

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

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

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

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

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

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

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

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

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

[18] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV98*.

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

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

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

[22] F. Torre and M. J. Black. Robust principal component analysis for computer vision. In *ICCV2001*.

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

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

[25] Y. Weiss and E.H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR96*.

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

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

Appendix

Parametric representation of affine transformation: *Given a pair of affine cameras ψ_r, ψ', and a reference plane π_r, we can represent any other affine transformation $\mathbf{m}_{2\times 3}$ induced by a plane π_m by:*

$$\mathbf{m} = \mathbf{m_r} + \mathbf{e}'\mathbf{v}^T,$$

where $\mathbf{m_r}$ is the affine transformation induced by reference plane π_r, $\mathbf{e}' = (e_1, e_2)^T$, and the homogeneous coordinates $(e_1, e_2, 0)$ is the direction of epipolar lines in camera ψ'. $\mathbf{v}^T = (v_1, v_2, v_3)$ is a 3-vector independent of camera ψ'.

Proof: Without loss of generality, let us choose three non-collinear points $[P_0, P_1, P_2]$ on 3D plane π_r. We ignore the degenerate case where a plane projects onto a line in the camera imaging plane. $[P_0, P_1, P_2]$ projects onto three non-collinear points $[p_0, p_1, p_2]$ in camera ψ_r, and $[p'_0, p'_1, p'_2]$ in camera ψ', where $p_i = (x, y)^T$ and $p'_i = (x', y')^T$ are 2D image coordinates. There exist three non-collinear points $[P'_0, P'_1, P'_2]$ on plane π_m that will also project onto $[p_0, p_1, p_2]$ in camera ψ_r. Denote the image points of $[P'_0, P'_1, P'_2]$ in camera ψ' as $[p''_0, p''_1, p''_2]$, as shown in Fig.(6).

Since an affine transformation is uniquely determined by three pairs of non-collinear corresponding points, we have:

$$\begin{bmatrix} p'_0 & p'_1 & p'_2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_r \\ 0\,0\,1 \end{bmatrix} * \begin{bmatrix} p_0 & p_1 & p_2 \\ 1 & 1 & 1 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} p''_0 & p''_1 & p''_2 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m}_{2\times 3} \\ 0\,0\,1 \end{bmatrix} * \begin{bmatrix} p_0 & p_1 & p_2 \\ 1 & 1 & 1 \end{bmatrix} \tag{9}$$

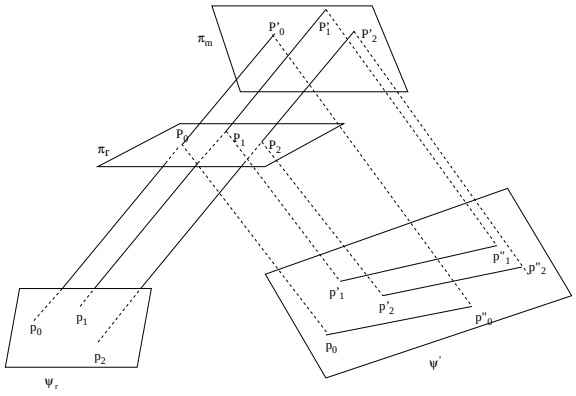


Figure 6. The relationship between 3D planes and affine cameras.

Since affine camera has parallel projection, $\left[\overline{P_0P'_0}, \overline{P_1P'_1}, \overline{P_2P'_2}\right]$ are three parallel line segments. Parallelism is preserved by affine camera. Therefore, $\left[\overline{P_0P'_0}, \overline{P_1P'_1}, \overline{P_2P'_2}\right]$ will project onto parallel line segments $\left[\overline{p'_0p''_0}, \overline{p'_1p''_1}, \overline{p'_2p''_2}\right]$ (epipolar lines) in affine camera ψ' whose projection matrix is $\{\mathbf{M'}_{2\times 3}, \mathbf{T}'\}$. Denote $\overline{p_ip_j} = p_j - p_i$. We have:

$$\begin{aligned} \left[\overline{p'_0p''_0}, \overline{p'_1p''_1}, \overline{p'_2p''_2}\right] &= \mathbf{M}' * \left[\overline{P_0P'_0}, \overline{P_1P'_1}, \overline{P_2P'_2}\right] \\ &= \mathbf{M}' * \mathbf{D} * [k_0, k_1, k_2]\,, \end{aligned} \tag{10}$$

where $\mathbf{D}$ (unit 3-vector) denotes the direction of parallel lines $\overline{P_0P'_0}, \overline{P_1P'_1}, \overline{P_2P'_2}$, and $\left[\overline{P_0P'_0}, \overline{P_1P'_1}, \overline{P_2P'_2}\right] = \mathbf{D} * [k_0, k_1, k_2]$, with k_i denoting the length of line segment $\overline{P_iP'_i}$. $[k_0, k_1, k_2]$ is independent of camera ψ'.

Denote $\mathbf{e}' = [e_1, e_2]^T = \mathbf{M}' * \mathbf{D}$ (It is obvious that $[e_1, e_2, 0]^T$ is the direction of epipolar lines in homogeneous coordinates in camera ψ'). From Eq.(10) we have:

$$\begin{aligned} \left[p''_0, p''_1, p''_2\right] &= \left[p'_0, p'_1, p'_2\right] + \left[\overline{p'_0p''_0}, \overline{p'_1p''_1}, \overline{p'_2p''_2}\right] \\ &= \left[p'_0, p'_1, p'_2\right] + \mathbf{e}' * [k_0, k_1, k_2] \end{aligned} \tag{11}$$

Substitute Eq.(11) and Eq.(8) into Eq.(9), we have:

$$\begin{aligned} &\begin{bmatrix} \mathbf{m}_{2\times 3} \\ 0\,0\,1 \end{bmatrix} * \begin{bmatrix} p_0 & p_1 & p_2 \\ 1 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{m}_r \\ 0\,0\,1 \end{bmatrix} * \begin{bmatrix} p_0 & p_1 & p_2 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} \mathbf{e}' \\ 0 \end{bmatrix} * [k_0, k_1, k_2] \end{aligned} \tag{12}$$

Since $[p_0, p_1, p_2]$ are non-collinear points, the matrix $P_{3\times 3} = \begin{bmatrix} p_0 & p_1 & p_2 \\ 1 & 1 & 1 \end{bmatrix}$ is non-singular and $P^{-1}_{3\times 3}$ exists. Therefore, from Eq.(12), we have:

$$\mathbf{m} = \mathbf{m}_r + \mathbf{e}' * [v_0, v_1, v_2] \tag{13}$$

Here $\left[\mathbf{e}'^T, 0\right]$ is the direction of epipolar lines in homogeneous coordinate in camera ψ', and $\mathbf{v}^T = [v_0, v_1, v_2] = [k_0, k_1, k_2] * P^{-1}_{3\times 3}$. It is obvious that the 3-vector $\mathbf{v^T}$ is independent of the second camera ψ'. $\diamond$