On analyzing video with very small motions

Michael Dixon[1], Austin Abrams[1], Nathan Jacobs[2], Robert Pless[1]

[1] Washington University in St Louis
{msd2|abramsa|pless}@cse.wustl.edu

[2] University of Kentucky
jacobs@cs.uky.edu

Abstract

We characterize a class of videos consisting of very small but potentially complicated motions. We find that in these scenes, linear appearance variations have a direct relationship to scene motions. We show how to interpret appearance variations captured through a PCA decomposition of the image set as a scene-specific non-parametric motion basis. We propose fast, robust tools for dense flow estimates that are effective in scenes with small motions and potentially large image noise. We show example results in a variety of applications, including motion segmentation and long-term point tracking.

1. Introduction

This paper is about understanding and parameterizing motion in a class of videos that can fairly be described as very boring—long videos of scenes with changes due to very small motions. This includes video captured by a camera observing the breathing of someone asleep, watching trees wave gently in the wind, or observing a car engine as it vibrates when it starts, and also includes video from cameras whose viewpoint jitters because they are handheld or mounted on a shaky support. Within this class of videos, there are a wide variety of problem domains that require understanding and segmenting motions within the scene.

One natural intermediate representation to support these applications is the dense motion field between all frames in the video sequence. The traditional approach to solving for a dense motion field is to combine independent frame-to-frame flow estimates. This approach does not take advantage of the similarities between all frames, and is therefore needlessly slow. In addition, as we will show, it does not always give the best result.

Thus, this paper offers a fast and robust algorithm for computing dense motion estimates within this class of videos, where the motions are very small and perhaps are repeated (periodically or not) over time. The approach is based upon computing the PCA decomposition of the set
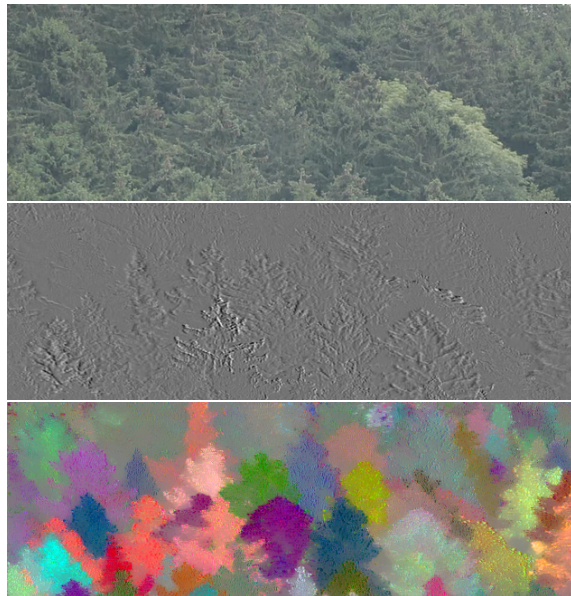

Figure 1. (top) A single frame extracted from a one minute video, which was downloaded from Vimeo [1], of gently waving trees. (middle) The first component of the PCA decomposition of the video shows characteristic patterns found in the PCA components of images of moving textures. In such scenes, PCA often fails to provide high quality image reconstructions; despite this, it still encodes the motion in the scene. (bottom) Using multiple PCA components—and interpreting them in terms of observed scene motion—enables an efficient method for segmenting a scene into locations of coherent motion, among other applications.

of images within the video. The PCA decomposition creates component images that approximately span the space of image variation—when this variation is caused by small motions, the component images are often similar to local, directional derivative filters. As one example, Figure 1 (top) shows one frame of a video of trees, and the first principle component image of the PCA decomposition. We show how a small set of such component images can be quickly translated into *motion components*, so that linear combinations of these motion components span the space of dense

motion fields. Figure 1 shows a false color image created by simply combining three of these motion components, highlighting the ability of this method to capture and segment complicated spatial patterns of coherent motion.

There are many advantages to exploiting the link between PCA and motions within a scene. First, there exist fast incremental algorithms for computing the PCA decomposition of video sequences [4], much faster than modern frame-to-frame dense optic flow methods. Second, the PCA decomposition focuses on correlated image variation and naturally suppresses many common forms of imaging noise. Third, since the PCA decomposition summarizes the variations in all of the images within the scene, extracting motion components from this decomposition simultaneously solves for the relative motion between all images in the video.

Our specific contributions are (a) characterizing the types of videos and motions for which there is a simple relationship between the motions in the scene and the PCA decomposition of the image set, (b) the derivation of an algorithm to compute dense optic flow estimates between all images in videos that fit within this category, (c) the use of the concise motion description captured within the motion components and coefficients to provide new tools for motion segmentation, and (d) example results on a wide variety of scenes.

1.1. Background and Related Work

When parameterizing variation in a set of images, it is common to decompose the sources of variation into two terms, isolating appearance and motion

$$I_i(p) = A(W(p, \theta_i)) \qquad (1)$$

where $A(p)$ represents the static appearance (texture) of each point p in the scene and the motion of the texture is represented by a warp function $W(p, \theta_i) = \langle W_x(p, \theta_i), W_y(p, \theta_i) \rangle$, which transforms a point p based on parameters θ_i.

This general formulation encompasses a broad array of different models, and solving for various A and W and their corresponding parameters is a long-standing and important problem in computer vision.

Linear models that focus on appearance variations (assuming the warp W is the identity) have been extensively used for face recognition [20] or to model lighting variations in indoor and outdoor scenes [19, 12, 10]. In addition, there is significant work in modeling the dynamics of appearance change in scenes with regular variations [17], such as fixed camera shots of waves, vegetation in the wind, escalators and smoke. These dynamic texture models have been extended to segment scenes into multi-layered textures [6], and dynamic textures have also been used for background models [23].

The appearance and warp terms are inherently dependent; this forms the basis for the work on active appearance models [7], which creates piecewise linear models of the warp and the texture variation within small image regions. Explicitly considering the dependence between appearance variation and planar models of the warping function is used to accelerate the inverse compositional variant of Lucas-Kanade tracking [2]. Manifold Pursuit [16] augments the image appearance vector with image derivatives in order to automatically account for small image translations or warps.

For videos that contain both appearance and motion variations within the scene (e.g. panning over a field of waving flowers), the image alignment problem has been solved by finding the parametric image transforms that minimize the covariance matrix of the best auto-regressive model imputed by the resulting alignment [8], or using the linear dynamical system to predict pixel image intensities and using those predictions within a brightness constancy assumption [22]. It has also been noted that when the camera pan is small, there are PCA components that are similar to derivative images, and the corresponding PCA coefficient trajectory is non-periodic and seems to correlate with camera motion [8]. This property has also been used on PCA decompositions of small video blocks in order to distinguish dramatic noise from consistent motion [21]. For static cameras that see many different translations within the scene, a layered model of warps has recently shown an impressive ability to model background variations and give useful results on very complicated, real-world scenes such as background subtraction for outdoor bird feeders [13].

Thus, the previous work mostly learns linear models of the appearance variation that are tuned to a particular data set but capture motion models that follow some global parametric form (perhaps with piecewise components). One previous paper that seeks to explicitly create non-parametric models of common flow within a scene is Black et. al. [3], who show that PCA on already existing motion fields provides robust means of solving for motion fields for new images of the same scene. They use optical flow to initialize their results and then compute the PCA decomposition of the resulting flow fields. In contrast, we compute the PCA decomposition of the original image data, then show how to interpret the resulting components and coefficients as a model of the motion fields within the scene.

2. Non-Parametric Motion Components

In this work we assume that A is fixed, and explore how image motion fields can be modeled with a low dimensional linear subspace, which we refer to as *principal motion components*.

With such a model, the warp function $W(p, \theta) = \langle W_x(p, \theta), W_y(p, \theta) \rangle$ describes the displacement of each

pixel p in terms of k motion components as follows:

$$W_x(p, \theta) = p_x + M_x(p)^\top \theta \qquad (2)$$
$$W_y(p, \theta) = p_y + M_y(p)^\top \theta \qquad (3)$$

where $M_x(p)$ and $M_y(p)$ define the k-dimensional basis of possible x and y displacements at each pixel.

Because the displacement at each pixel is not required to be continuous with respect to p, these components can represent motion that cannot be described with affine or projective transformations. While this flexibility can also make it difficult to estimate the terms of such a model, since the number of parameters for each basis is twice the number of pixels, it also makes it possible to capture complex scene-specific motion models.

However, under certain conditions, the motion components M are linearly related to the subspace (of image space) spanned by the image set, and we can exploit this to recover a parameterization of motion. In the following section, we formalize this relationship and derive a method for globally estimating the underlying motion model for a set of images using the PCA decomposition.

3. Estimating Principal Motion Components

When object motion is the dominant cause of change in a scene, a PCA decomposition of the video will yield a low-dimensional linear representation of the appearance change caused by that motion. Although this is explicitly a representation of the variation in image intensity, the PCA components and coefficients are implicitly encoding the position of objects as well. In this section, we show how a PCA decomposition of the images, which is typically considered an *appearance* representation, can be reinterpreted to explicitly model the *positions* of the objects in a scene.

We present two approaches that use the PCA decomposition to estimate a principal component basis, and thereby, a dense motion field. The first method uses PCA coefficient trajectories as a proxy for the motion parameters and solves for the motion components M_x and M_y that best explain the image set variations. The second method directly reasons about the PCA component images, taking advantage of the observation that the PCA components of translating texture are often similar to image derivatives.

Before we begin describing our algorithms we briefly discuss the properties of PCA when applied to moving textures. We highlight how these properties are useful for motion estimation.

3.1. PCA Decomposition of Moving Textures

The PCA decomposition of a video with small motions is related to the motion in the scene; in this section we explore the relationship. Assume we are given a set of images $\mathbf{I} = \{I_1, \ldots, I_n\}$ formed by taking a constant appearance
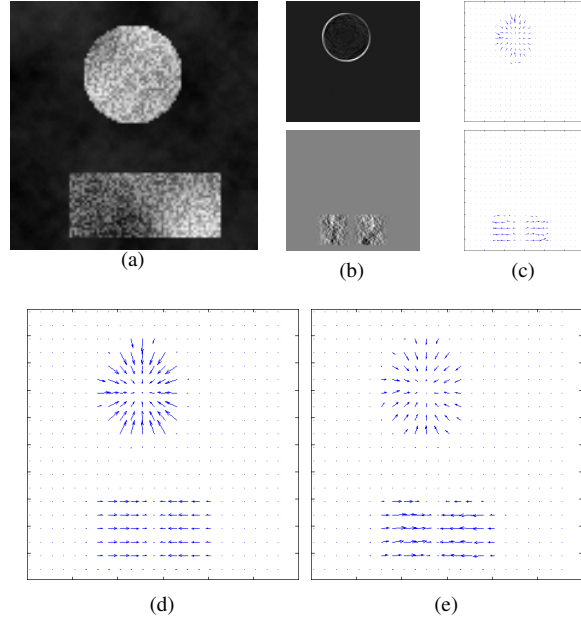

(a)   (b)   (c)

(d)   (e)

Figure 2. A synthetic example to validate the approach given in Section 3. (a) The appearance A of a synthetic sequence of images. (b) The first two principal components from the appearance variations. Because the motions in the video are small, these components are locally similar to derivatives of the appearance image. (c) The motion fields solved from the first two principal components. (d) The ground-truth warp field at $t = 34$. (e) The recovered warp. Notice that the recovered warp differs from the ground truth only in terms of scale, due to the different scale in warp coefficients θ.

(texture) $A(p)$ and warping it by $W(p, \theta_i)$. If the magnitude of the warp is small, then $\mathbf{I}$ will span a low-dimensional linear subspace. Applying PCA to the set of images yields an orthogonal basis U and a set of coefficients $\{v_1, \ldots, v_n\}$ for each image; furthermore, these are both related to scene motion.

As a real-world example consider the live stream [15] of two cute puppies, sleeping peacefully, in Figure 3. The PCA decomposition (both the components and the coefficients) of this scene are related to the subtle motions due to breathing and ear twitching. This highlights that the PCA is able to extract subtle, motion-related features of the scene.

For the class of linear warp functions $W(p, \theta_i) = \hat{M}(p)^\top \theta_i$ estimating the motion parameters θ_i is straightforward; they can be recovered from $\mathbf{I}$ (with possible permutation/rescaling) by first solving for the linear subspace spanned by $\mathbf{I}$ and then projecting each image I_i onto the basis. And, to the extent that the linear subspace has been recovered correctly, there exists a motion basis M such that $W(p, \theta_i) = \hat{M}(p)^\top \theta_i = M(p)^\top v_i$. In other words, if the magnitude of the underlying warp is small (which is the case with subtle motions), then the warp can be expressed
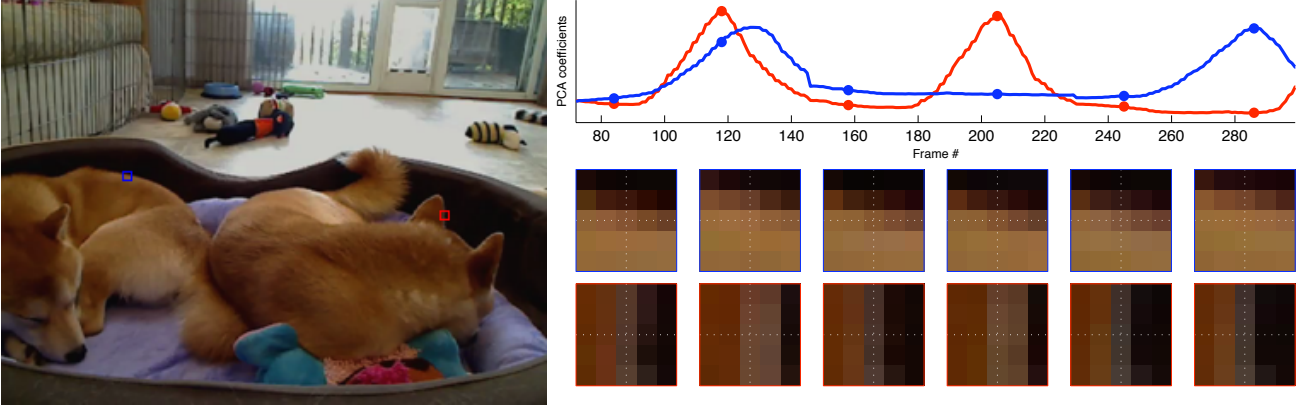
Figure 3. (left) A frame from a video of two shiba inus sleeping. (top right) The first two PCA coefficients through time. The images corresponding to the blue and red dots are shown in the two rows underneath. The first row is zoomed in on the blue rectangle, and the second row is zoomed in on the red rectangle. This shows that under subtle movements, PCA components code for motion.

as a linear combination of an unknown motion basis M and the coefficients v_i from the original PCA decomposition. We validate this result in Figure 2 by creating a synthetic image set generated from a known A and W, and recovering W from the resulting image set.

In the following section, we show how these motion-related features can be used to solve for motion components that enable direct reasoning about the motion of objects.

3.2. Estimating a Motion Model from PCA Coefficients

In this section, we show how to use the PCA coefficients v to recover the motion basis M of a linear warp function W from a set of images $\mathbf{I}$. Our basic assumption (from Equation 1) is that each image is some arbitrary warp of a basic texture A, so that $I_i(p) = A(W(p, v_i))$. Alternatively, we can approximately reconstruct the static texture from any specific image by applying the inverse image warp, so that for any i, $A(p) = I_i(W^{-1}(p, v_i))$.

This lets us factor out the appearance term to relate specific images, for example: $I_i(W^{-1}(p, v_i)) = I_{i+1}(W^{-1}(p, v_{i+1}))$, and we can directly express one image in terms of the previous image as:

$$I_i(W^{-1}(W(p, v_{i+1}), v_i)) \approx I_{i+1}(p). \qquad (4)$$

If the warp function W is linear then the composition of the two warps above is additive, we can simplify the approximate equality in Equation 4 to

$$I_i(W(p, \Delta v_i), i) \approx I_{i+1}(p) \qquad (5)$$

where $\Delta v_i = v_{i+1} - v_i$.

This allows us to write the relationship between the temporal difference image and the warp at frame i:

$$I_i(W(p, \Delta v_i), i) - I_i(p) \approx I_{i+1} - I_i \approx I_i^t(p) \qquad (6)$$

Replacing W with the linear model defined in Equations 2 and 3, and using the first order Taylor series to relate image warps to image derivatives, we can approximate Equation 6 as:

$$I_i^x(p)M_x(p)^\top \Delta v_i + I_i^y(p)M_y(p)^\top \Delta v_i \approx I_i^t(p) \qquad (7)$$

where I^x, I^y, and I^t denote the spatio-temporal derivatives of I.

Notice that this takes the form of the classical optical flow constraint, where the flow terms are replaced with the product of the known difference of PCA coefficients Δv_i and an unknown motion basis M. Following the Horn-Schunck formulation [11], we add a spatial smoothness constraint on the flow, yielding the following pair of equations:

$$(I_i^x(p)^2 + \alpha^2)\Delta v_i^\top M_x(p) + I_i^x(p)I_i^y(p)\Delta v_i^\top M_y(p)$$
$$= \alpha^2 \Delta v_i^\top \bar{M}_x(p) - I_i^x(p)I_i^t(p) \qquad (8)$$

$$I_i^x(p)I_i^y(p)\Delta v_i^\top M_y(p) + (I_i^y(p)^2 + \alpha^2)\Delta v_i^\top M_y(p)$$
$$= \alpha^2 \Delta v_i^\top \bar{M}_y(p) - I_i^y(p)I_i^t(p) \qquad (9)$$

where the parameter α is a spatial regularization constant, and $\bar{M}_x(p)$ and $\bar{M}_y(p)$ are a local average of M_x and M_y in the neighborhood surrounding p.

Using this formulation, we linearly estimate the warp function W by choosing the motion basis $M(p)$, at each pixel and for all images, that minimizes the least squared error with respect to Equations 8 and 9.

3.3. Estimating a Motion Model from PCA Components

An alternative interpretation of the PCA , we observe that the principal components $U(p)$ describe the partial derivatives of the images with respect to v. From the standard

4

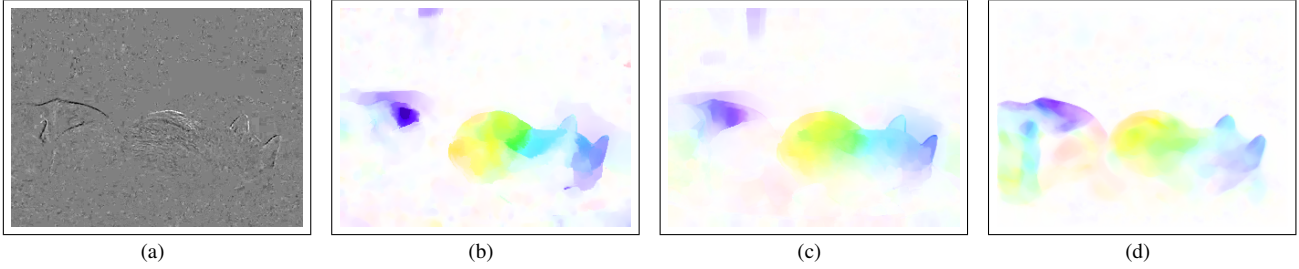|			
(a)	(b)	(c)	(d)

Figure 4. Comparison of our optical flow estimation method to a state-of-the-art technique [18]. (a) Original pair of frames of the shiba inus data set. (b) The optical flow computed by [18]. (c) Multiple frames of optical flow estimates from (b) projected onto a motion PCA basis [3] (d) Our results. Notice that our result is free of artifacts, especially at the top of the frame and between the ears of the dog on the right.

PCA reconstruction, we have that

$$I_i(p) = A(W(p, v_i)) \approx U(p)v_i, \qquad (10)$$

and therefore

$$\frac{\partial}{\partial v} A((W(p, v_i)) \approx U(p). \qquad (11)$$

Replacing W with the linear model defined in Equations 2 and 3 we get

$$A^x(p)M_x(p) + A^y(p)M_y(p) \approx U(p) \qquad (12)$$

where A^x and A^y are the spatial derivatives of A.

Although $A(p)$ is an unknown, for scenes with subtle variation the mean image, $\bar{I}(p)$ can serve as an approximation. Replacing A with $\bar{I}$ gives

$$\bar{I}^x(p)M_x(p) + \bar{I}^y(p)M_y(p) \approx U(p) \qquad (13)$$

As before, adding a spatial regularization term gives the following system of equations from which you can solve for M_x and M_y using the Horn-Schunck method:

$$(\bar{I}^x(p)^2 + \alpha^2)M_x(p) + \bar{I}^x(p)\bar{I}^y(p)M_y(p) \qquad (14)$$
$$= \alpha^2 \bar{M}_x(p) - \bar{I}^x(p)U(p)$$

$$\bar{I}^x(p)\bar{I}^y(p)M_y(p) + (\bar{I}^y(p)^2 + \alpha^2)M_y(p) \qquad (15)$$
$$= \alpha^2 \bar{M}_y(p) - \bar{I}^y(p)U(p)$$

When $\bar{I}$ is a good approximation of A, this approach offers a simpler and more efficient way of solving for motion components. For all experiments presented in this paper, this method gave comparable results to the one presented in the previous section.

4. Rapid, Dense Optical Flow Estimation

We have presented a method for solving for consistent motion components for scenes with small, repeated motions. These motion components map the coefficients from

the appearance basis onto flow fields, thereby giving a simple method for estimating the optical flow of the entire video. Each video frame i has coefficients v_i computed from the initial solution for the PCA appearance basis. These serve as the parameters of our warp model and the motion components compute the displacement of this pixel relative to a mean image as $(M_x(p)v_i, M_y(p)v_i)$. In addition, if we define Δv_i as the difference in the coefficients from the previous frame, we can compute the flow of all points p in our image at frame i:

$$F_x(p, i) = M_x(p)\Delta v_i \qquad (16)$$
$$F_y(p, i) = M_y(p)\Delta v_i \qquad (17)$$

Since the coefficients v_i are solved globally over all frames, the resulting frame-to-frame flow estimates are not subject to drift.

Additionally, the motion bases give a rapid method to generate flow fields from new images, assuming that the new images arise from the same linear motion basis. Given a new image I, we project I onto the basis vectors U to get corresponding coefficients v. We can use these coefficients in Equations 16 and 17 to get updated positions, or use Δv to get flow. Each of these steps is a matrix multiplication corresponding to $O(k)$ multiplications per pixel so we can estimate dense optical flow for new images very quickly.

5. Results and Applications

Examples of principal motion components automatically extracted from a wide range of video data sets can be seen in Figure 6.

5.1. Comparison to Traditional Optical Flow

We performed a qualitative comparison of our method for computing dense optical flow with a more traditional approach. For the traditional approach we used the high-quality implementation provided by Sun [18]. Figure 4 shows results using this method, our method, and a hybrid method, similar to [3] that projects many instances of

5

frame-to-frame optical flow onto a common motion basis. Notice that using PCA on frame-to-frame optical flow preserves repeated artifacts caused in the individual frame estimates. Our method, which uses global appearance information from the entire sequence to solve for a PCA basis, does not suffer from such artifacts.

5.2. Motion Segmentation

In this section, we use our motion components, M, to estimate pixel-wise affinities, with the goal of segmenting the scene into regions with similar motion. We adopt the framework of Brox and Malik [5], which approaches this problem by clustering point trajectories. Like our approach, their method uses affinity propagation to cluster pixels, but they rely on computationally expensive and notoriously challenging point tracking to estimate affinities between points.

We efficiently compute the affinity between a pair of pixels p, q using the flow fields F_x, F_y, which are defined in Section 4 and are derived directly from our motion components. We define an affinity that favors clusters composed of nearby pixels with similar motion:

$$S(p,q) = \exp\left(\frac{-d^2(p,q)}{\sigma^2}\right)$$

The distance function, d, is the maximum distance between the displaced locations of the original points p, q:

$$d(p,q) = \max_t d_t(p,q) \tag{18}$$

where $d_t(p,q)$ is the product of the image space distance, $\|p - q\|$, and the difference in the pixel-wise flow field, $d_t(p,q) = \|(F_x(p,t), F_y(p,t)) - (F_x(q,t), F_y(q,t))\|$.

We then cluster the pixels using affinity propagation [9] on our affinity matrix S. In our experiments, we set σ to be 0.2 of the maximum image dimension and the cluster preference to be the median of S. Figure 5 shows an example segmentation. This result demonstrates that using only motion fields we can obtain high-quality motion segmentation in a challenging case without explicit point tracking. In practice, we do not consider all pixels when performing affinity propagation. As is common practice, we reduce the memory requirements by first identifying a set of exemplar pixels and clustering the exemplars. We use hierarchical k-means to find an evenly distributed set of exemplars.

6. Conclusions

In this paper, we introduce *principal motion components*, a non-parametric linear model for motion that estimates a warp function by reinterpreting principal components of the raw image data, which are typically used for modeling *appearance* variations. This model is particularly useful for scenes that have very small but consistent motions.
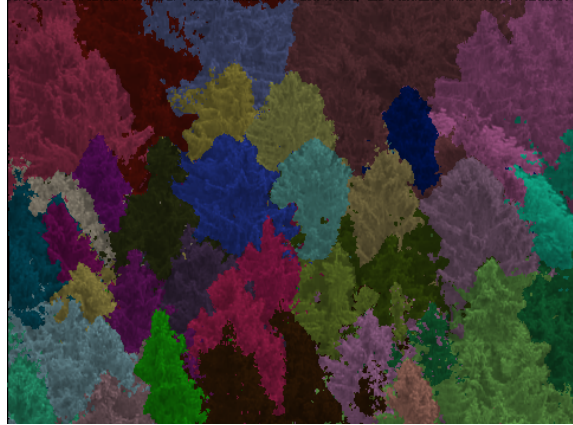

Figure 5. Performing motion segmentation on the data set from Figure 1 into 30 clusters. The full video is provided in the supplemental material.

Our model derives motion components directly from the PCA decomposition, which represents the dominant variations over the entire image set. As such, our model works well in areas where frame-to-frame differences are too difficult to track with more typical optical flow algorithms. Furthermore, since flow vectors can be computed directly from the PCA coefficient space, determining the flow of a new image in the sequence is very fast.

Using pixel-wise affinities, we can leverage principal motion components to segment the scene into regions with similar motion. This is a particularly promising application, because our method of affinity calculation does not depend on problematic point tracking through time, and is instead based on the principle motion components optimized globally over all frames.

Our example results show promise for analyzing video that is very boring visually, but which captures important variations. Finding the motions of the puppies is a proxy for automatic breathing monitoring for babies, and capturing vibrational patterns (as visualized in the car engine) is an important problem in the monitoring of manufacturing plants. We encourage the reader to also look at our supplementary video, which gives additional results. The supplemental video is available at http://research.engineering.wustl.edu/~abramsa/projects/motionPCA/.
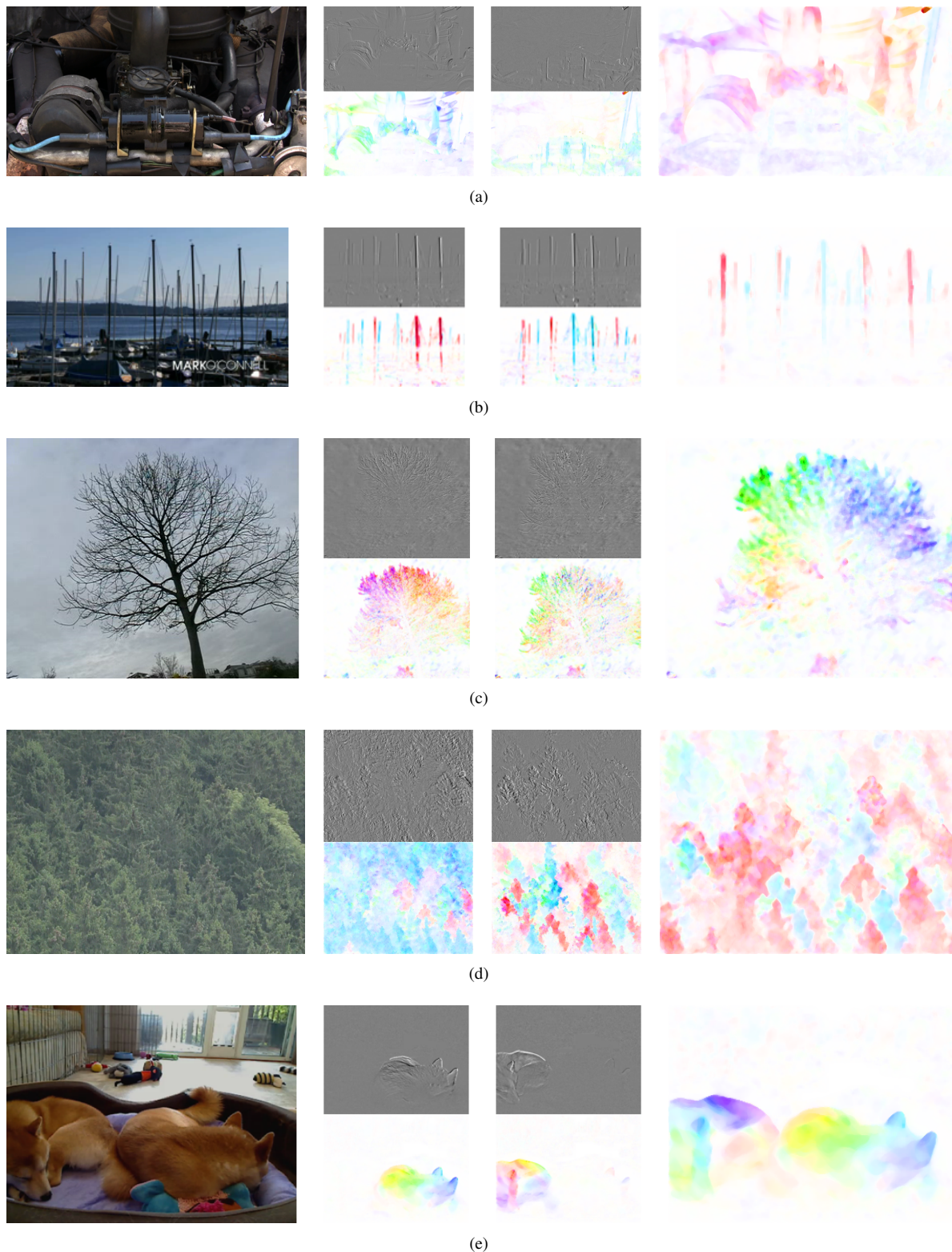
Acknowledgments

Figure 6. Some example image sequences and their computed flows. For (a)-(e), we show (left) an example image, (top center) the first two principal component images, (bottom center) their corresponding principal motion components M, and (right) the computed flow for an image in the sequence.

References

[1] 1 Minute Vimeo Project. http://vimeo.com/groups/1minute.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(1):221 – 255, March 2004.

[3] M. Black, Y. Yacoob, A. Jepson, and D. Fleet. Learning parameterized models of image motion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[4] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proc. European Conference on Computer Vision*, pages 707–720, 2002.

[5] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *Proc. European Conference on Computer Vision*, 2010.

[6] A. Chan and N. Vasconcelos. Variational layered dynamic textures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 484–498. Springer, 1998.

[8] A. Fitzgibbon. Stochastic rigidity: Image registration for nowhere-static scenes. In *Proc. IEEE International Conference on Computer Vision*, 2001.

[9] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[10] R. Garg, H. Du, S. M. Seitz, and N. Snavely. The dimensionality of scene appearance. In *Proc. IEEE International Conference on Computer Vision*, 2009.

[11] B. K. P. Horn and B. G. Schunck. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981.

[12] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[13] T. Ko, D. Estrin, and S. Soatto. Warping background subtraction. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[14] Mark O' Connell. http://www.artbeats.com/clips/rac-fh106-60/hd.

[15] SFShiba. http://www.ustream.tv/sfshiba.

[16] A. Shashua, A. Levin, and S. Avidan. Manifold pursuit: A new approach to appearance based recognition. In *Proc. International Conference on Pattern Recognition*, 2002.

[17] S. Soatto, G. Doretto, and Y. N. Wu. Dynamic textures. In *Proc. IEEE International Conference on Computer Vision*, 2001.

[18] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[19] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz. Factored time-lapse video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), Aug. 2007.

[20] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Neuroscience*, 3(1):71–86, 1991.

[21] N. Verbeke and N. Vincent. A PCA-based technique to detect moving objects. In *Proc. of the Scandinavian Conference on Image Analysis (SCIA)*, pages 641–650, Berlin, Heidelberg, 2007. Springer-Verlag.

[22] R. Vidal and A. Ravich. Optical flow estimation and segmentation of multiple moving dynamic textures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[23] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proc. European Conference on Computer Vision*, 2003.