Image and Video Segmentation by Anisotropic Kernel Mean Shift

Jue Wang, Bo Thiesson, Yingqing Xu, and Michael Cohen

Microsoft Research (Asia and Redmond)

Abstract. Mean shift is a nonparametric estimator of density which has been applied to image and video segmentation. Traditional mean shift based segmentation uses a radially symmetric kernel to estimate local density, which is not optimal in view of the often structured nature of image and more particularly video data. In this paper we present an anisotropic kernel mean shift in which the shape, scale, and orientation of the kernels adapt to the local structure of the image or video. We decompose the anisotropic kernel to provide handles for modifying the segmentation based on simple heuristics. Experimental results show that the anisotropic kernel mean shift outperforms the original mean shift on image and video segmentation in the following aspects: 1) it gets better results on general images and video in a smoothness sense; 2) the segmented results are more consistent with human visual saliency; 3) the algorithm is robust to initial parameters.

1 Introduction

Image segmentation refers to identifying homogenous regions in the image. Video segmentation, in this paper, means the joint spatial and temporal analysis on video sequences to extract regions in the dynamic scenes. Both of these tasks are misleadingly difficult and have been extensively studied for several decades. Refer to [9,10,11] for some good surveys. Generally, spatio-temporal video segmentation can be viewed as an extension of image segmentation from a 2D to a 3D lattice. Recently, mean shift based image and video segmentation has gained considerable attention due to its promising performance.

Many other data clustering methods have been described in the literature, ranging from top down methods such as K-D trees, to bottom up methods such as K-means and more general statistical methods such as mixtures of Gaussians. In general these methods have not performed satisfactorily for image data due to their reliance on an a priori parametric structure of the data segment, and/or estimates of the number of segments expected. Mean shift's appeal is derived from both its performance and its relative freedom from specifying an expected number of segments. As we will see, this freedom has come at the cost of having to specify the size (bandwidth) and shape of the influence kernel for each pixel in advance.

The difficulty in selecting the kernel was recognized in [3,4,12] and was addressed by automatically determining a bandwidth for spherical kernels. These

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 238–249, 2004.

approaches are all purely *data driven*. We will leverage this work and extend it to automatically select general elliptical (anisotropic) kernels for each pixel. We also add a priori knowledge about typical structures found in video data to take advantage of the extra freedom in the kernels to adapt to the local structure.

1.1 Mean Shift Based Image and Video Segmentation

Rather than begin from an initial guess at the segmentation, such as seeding points in the K-means algorithm, mean shift begins at each data point (or pixel in an image or video) and first estimates the local density of similar pixels (i.e., the density of nearby pixels with similar color). As we will see, carefully defining "nearby" and "similar" can have an important impact on the results. This is the role the kernel plays.

More specifically, mean shift algorithms estimate the local density *gradient* of similar pixels. These gradient estimates are used within an iterative procedure to find the peaks in the local density. All pixels that are drawn upwards to the same peak are then considered to be members of the same segment.

As a general nonparametric density estimator, mean shift is an old pattern recognition procedure proposed by Fukunage and Hostetler [7], and its efficacy on low-level vision tasks such as segmentation and tracking has been extensively exploited recently. In [1,5], it was applied for continuity preserving filtering and image segmentation. Its properties were reviewed and its convergence on lattices was proven. In [2], it was used for non-rigid objects tracking and a sufficient convergence condition was given. Applying mean shift on a 3D lattice to get a spatio-temporal segmentation of video was achieved in [6], in which a hierarchical strategy was employed to cluster pixels of 3D space-time video stack, which were mapped to 7D feature points (position(2), time(1), color(3), and motion(1)).

The application of mean shift to an image or video consists of two stages. The first stage is to define a *kernel* of influence for each pixel x_i. This kernel defines a measure of intuitive *distance* between pixels, where distance encompasses both spatial (and temporal in the case of video) as well as color distance. Although manual selection of the size (or *bandwidth*) and shape of the kernel can produce satisfactory results on general image segmentation, it has a significant limitation. When local characteristics of the data differ significantly across the domain, it is difficult to select globally optimal bandwidths. As a result, in a segmented image some objects may appear too coarse while others are too fine. Some efforts have been reported to locally vary the bandwidth. Singh and Ahuja [12] determine local bandwidths using Parzen windows to mimic local density. Another variable bandwidth procedure was proposed in [3] in which the bandwidth was enlarged in sparse regions to overcome the noise inherent with limited data.

Although the size may vary locally, all the approaches described above used a radially symmetric kernel. One exception is the recent work in [4] that describes the possibility of using the general local covariance to define an asymmetric kernel. However, this work goes on to state, "Although a fully parameterized covariance matrix can be computed.., this is not necessarily advantageous.." and then returns to the use of radially symmetric kernels for reported results.

The second iterative stage of the mean shift procedure assigns to each pixel a *mean shift point*, $M(x_i)$, initialized to coincide with the pixel. These mean shift points are then iteratively moved upwards along the gradient of the density function defined by the sum of all the kernels until they reach a stationary point (a *mode* or hilltop on the virtual terrain defined by the kernels). The pixels associated with the set of mean shift points that migrate to the (approximately) same stationary point are considered to be members of a single segment. Neighboring segments may then be combined in a post process.

Mathematically, the general multivariate kernel density estimate at the point, x, is defined by

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_H(x - x_i) \tag{1}$$

where the n data points x_i represent a sample from some unknown density f, or in the case of images or video, the pixels themselves.

$$K_H(x) = |H|^{-1/2}K(H^{-1/2}x) \tag{2}$$

where $K(z)$ is the d-variate kernel function with compact support satisfying the regularity constraints as described in [13], and H is a symmetric positive definite $d \times d$ bandwidth matrix. For the radially symmetric kernel, we have

$$K(z) = c\,k(||z||^2) \tag{3}$$

where c is the normalization constant. If one assumes a single global spherical bandwidth, $H = h^2 I$, the kernel density estimator becomes

$$\hat{f}(x) = \frac{1}{n(h)^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{4}$$

For image and video segmentation, the feature space is composed of two independent domains: the *spatial/lattice* domain and the *range/color* domain. We map a pixel to a multi-dimensional feature point which includes the p dimensional spatial lattice ($p = 2$ for image and $p = 3$ for video) and q dimensional color ($q = 3$ for L*u*v color space). Due to the different natures of the domains, the kernel is usually broken into the product of two different radially symmetric kernels (superscript s will refer to the spatial domain, and r to the color range):

$$K_{h^s,h^r}(x) = \frac{c}{(h^s)^p(h^r)^q}k^s\left(\left\|\frac{x^s}{h^s}\right\|^2\right)k^r\left(\left\|\frac{x^r}{h^r}\right\|^2\right) \tag{5}$$

where x^s and x^r are respectively the spatial and range parts of a feature vector, k^s and k^r are the profiles used in the two domains, h^s and h^r are employed bandwidths in two domains, and c is the normalization constant. With the kernel from (5), the kernel density estimator is

$$\hat{f}(x) = \frac{c}{n(h^s)^p(h^r)^q}\sum_{i=1}^{n} k^s\left(\left\|\frac{x^s - x_i^s}{h^s}\right\|^2\right)k^r\left(\left\|\frac{x^r - x_i^r}{h^r}\right\|^2\right) \tag{6}$$

As apparent in Equations (5) and (6), there are two main parameters that have to be defined by the user for the simple radially symmetric kernel based approach: the spatial bandwidth h^s and the range bandwidth h^r. In the variable bandwidth mean shift procedure proposed in [3], the estimator (6) is changed to

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{c}{(h_i^s)^p\ (h_i^r)^q} k^s \left(\left\| \frac{x^s - x_i^s}{h_i^s} \right\|^2 \right) k^r \left(\left\| \frac{x^r - x_i^r}{h_i^r} \right\|^2 \right) \tag{7}$$

There are now important differences between (6) and (7). First, potentially different bandwidths h_i^s and h_i^r are assigned to each pixel, x_i, as indicated by the subscript i. Second, the different bandwidths associated with each point appear within the summation. This is the so-called sample point estimator [3], as opposed to the *balloon* estimator defined in Equation (6). The sample point estimator, which we will refer to as we proceed, ensures that all pixels respond to the same global density estimation during the segmentation procedure. Note that the sample point and balloon estimators are the same in the case of a single globally applied bandwidth.

1.2 Motivation for an Anisotropic Kernel

During the iterative stage of the mean shift procedure, the mean shift points associated with each pixel climb to the hilltops of the density function. At each iteration, each mean shift point is attracted in varying amounts by the sample point kernels centered at nearby pixels. More intuitively, a kernel represents a measure of the likelihood that other points are part of the same segment as the point under the kernel's center. With no a priori knowledge of the image or video, actual distance (in space, time, and color) seems an obvious (inverse) correlate for this likelihood; the closer two pixels are to one another the more likely they are to be in the same segment.

We can, however, take advantage of examining a local region surrounding each pixel to select the size and shape of the kernel. Unlike [3], we leverage the full local covariance matrix of the local data to create a kernel with a general elliptical shape. Such kernels adapt better to non-compact (i.e., long skinny) local features such as can be seen in the monkey bars detail in Figure 2 and the zebra stripes in Figure 5. Such features are even more prevalent in video data from stationary or from slowly or linearly moving cameras. When considering video data, a spatio-temporal slice (parallel to the temporal axis) is as representative of the underlying data as any single frame (orthogonal to the temporal axis). Such a slice of video data exhibits stripes with a slope relative to the speed at which objects move across the visual field (see Figures 3 and 4). The problems in the use of radially symmetric kernels is particularly apparent in these spatio-temporal slice segmentations. The irregular boundaries between and across the stripe-like features cause a lack of temporal coherence in the video segmentation.

An anisotropic kernel can adapt its profile to the local structure of the data. The use of such kernels proves more robust, and is less sensitive to initial parameters compared with symmetric kernels. Furthermore, the anisotropic kernel provides a set of handles for application-driven segmentation. For instance, a user may desire that the still background regions be more coarsely segmented while the details of the moving objects to be preserved when segmenting a video sequence. To achieve this, we simply expand those local kernels (in the color and/or spatial dimensions) whose profiles have been elongated along the time dimension. By providing a set of heuristic rules described below on how to modulate the kernels, the segmentation strategy can be adapted to various applications.

2 Anisotropic Kernel Mean Shift

2.1 Definition

The *Anisotropic Kernel Mean Shift* associates with each data point (a pixel in an image or video) an anisotropic kernel. The kernel associated with a pixel adapts to the local structure by adjusting its shape, scale, and orientation. Formally, the density estimator is written as

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^r(H_i^s)^q} k^s\left(g(x^s, x_i^s, H_i^s)\right) k^r\left(\left\| \frac{x^r - x_i^r}{h^r(H_i^s)} \right\|^2\right) \tag{8}$$

where $g(x^s, x_i^s, H_i^s)$ is the Mahalanobis metric in the spatial domain:

$$g(x^s, x_i^s, H_i^s) = (x_i^s - x^s)^T H_i^{s^{-1}} (x_i^s - x^s) \tag{9}$$

In this paper we use a spatial kernel with a constant profile, $k^s(z) = 1$ if $|z| < 1$, and 0 otherwise. For the color domain we use an Epanechnikov kernel with a profile $k^r(z) = 1 - |z|$ if $|z| < 1$ and 0 otherwise. Note that in our definition, the bandwidth in color range h^r is a function of the bandwidth matrix in space domain H_i^s. Since H_i^s is determined by the local structure of the video, h^r thus varies from one pixel to another. Possibilities on how to modulate h^r according to H^s will be discussed later.

The bandwidth matrix H_i^s is symmetric positive definite. If it is simplified into a diagonal matrix with equal diagonal elements, (i.e., a scaled identity), then H_i^s models the radially symmetric kernels. In the case of video data, the time dimension may be scaled differently to represent notions of equivalent "distance" in time vs. image space. In general, allowing the diagonal terms to be scaled differently allows for the kernels to take on axis aligned ellipsoidal shapes. A full H_i^s matrix provides the freedom to model kernels of a general ellipsoidal shape oriented in any direction. The Eigen vectors of H_i^s will point along the axes of such ellipsoids. We use this additional freedom to shape the kernels to reflect local structures in the video as described in the next section.

2.2 Kernel Modulation Strategies

Anisotropic kernel mean shift give us a set of handles on modulating the kernels during the mean shift procedure. How to modulate the kernel is application related and there is not an uniform theory for guidance. We provide some intuitive heuristics for video data with an eye towards visually salient segmentation. In the case of video data we want to give long skinny segments at least an equal chance to form as more compact shapes. These features often define the salient features in an image. In addition, they are often very prominent features in the spatio-temporal slices as can be seen in many spatio-temporal diagrams. In particular, we want to recognize segments with special properties in the time domain. For example, we may wish to allow static objects to form into larger segments while moving objects to be represented more finely with smaller segments.

An anisotropic bandwidth matrix H_i^s is first estimated starting from a standard radially symmetric diagonal H_i^s and color radius h^r. The neighborhood of pixels around x_i is defined by those pixels whose position, x, satisfies

$$k^s(g(x, x_i, H_i^s)) < 1;\; k^r\left(\left\| \frac{x - x_i}{h^r(H_i^s)} \right\|^2\right) < 1 \tag{10}$$

An analysis of variance of the points within the neighborhood of x_i provides a new full matrix $\bar{H}_i^s$ that better describes the local neighborhood of points.

To understand how to modulate the full bandwidth matrix $\bar{H}_i^s$, it is useful to decompose it as

$$\bar{H}_i^s = \lambda D A D^T \tag{11}$$

where λ is a global scalar, D is a matrix of normalized Eigen vectors, and A is a diagonal matrix of Eigen values which is normalized to satisfy:

$$\prod_{i=1}^{p} a_i = 1 \tag{12}$$

where a_i is the i^{th} diagonal elements of A, and $a_i \geq a_j$, for $i < j$. Thus, λ defines the overall volume of the new kernel, A defines the relative lengths of the axes, and D is a rotation matrix that orients the kernel in space and time.

We now have intuitive handles for modulating the anisotropic kernel. The D matrix calculated by the covariance analysis is kept unchanged during the modulation process to maintain the orientation of the local data. By adjusting A and λ, we can control the spatial size and shape of the kernel. For example, we can encourage the segmentation to find long skinny regions by diminishing the smaller Eigen values in A as

$$a_i' = \begin{cases} a_i^{3/2} & : \quad a_i <= 1 \\ \sqrt{a_i} & : \quad a_i > 1 \end{cases}, i = 2, ..., p \tag{13}$$

In this way the spatial kernel will stretch more in the direction in which the object elongates. To create larger segments for static objects we detect kernels oriented along the time axis as follows. First, a scale factor s_t is computed as

$$s_t = \alpha + (1 - \alpha) \prod_{i=1}^{p-1} d_1(i)^2 \tag{14}$$

where d_1 is the first Eigen vector in D, which corresponds with the largest Eigen value a_1. $d_1(i)$ stands for the ith element in d_1, which is the x, y and t component of the vector when $i = 1, 2, 3$, respectively. α is a constant between 0 and 1. In our system, we set α to 0.25. The product in the above equation corresponds to the cosine of the angle between the first Eigen vector and the time axis. If the stretch direction of the kernel is close to the time axis, the scale factor is close to a small value α. Otherwise if the stretch direction is orthogonal to the time axis, then s_t is close to 1. The matrix A is thus changed as

$$a'_i = a_i \cdot s_t, i = 2, ..., p \qquad (15)$$

After the matrix A is modified by (13) and/or (14), the global scalar λ is changed correspondingly as

$$\lambda' = \lambda \prod_{i=1}^{p} \left(\frac{a_i}{a'_i} \right) \qquad (16)$$

To keep the *analysis resolution* in the color domain consistent with that in space domain, the bandwidth in the color domain is changed to

$$h^r(H_i^s) \leftarrow \sqrt{\frac{\lambda'}{\lambda}} \cdot h^r(H_i^s) \qquad (17)$$

The effect is to increase the color tolerance for segments that exhibit a large stretch, typically along the time axis (i.e., are static in the video).

2.3 Algorithm

The anisotropic mean shift segmentation is very similar to the traditional mean shift segmentation algorithm. The only difference is that a new anisotropic spatial kernel and space dependent kernel in the color domain are determined individually for each feature point prior to the main mean shift procedure. Recall that when kernels vary across feature points, the sample point estimator should be used in the mean shift procedure (note subscripts j within summation in step 4. The sample point anisotropic mean shift algorithm is formally described below. Steps 1-3 are the construction of kernels and steps 4-6 is the main mean shift procedure for these kernels.

1. Data and kernel initialization.
   – Transfer pixels into multidimensional (5D for image, 6D for video) feature points, x_i.
   – Specify initial spatial domain parameter h_0^s and initial range domain parameter h_0^r.
   – Associate kernels with feature points, initialize means to these points.
   – Set all initial bandwidth matrices in the spatial domain as the diagonal matrix $H_i^s = (h_0^s)^2 I$. Set all initial bandwidths in the range domain as $h^r(H_i^s) = h_0^r$.

2. For each point x_i, determine the anisotropic kernel and related color radius:
   - Search the neighbors of x_i to get all the points $x_j, j = 1, ..., n$ that satisfy the constraints of kernels:
   $$k^s(g(x_i, x_j, H_i^s)) < 1; k^r\left(\left\|\frac{x_i - x_j}{h^r(H_i^s)}\right\|^2\right) < 1$$
   Update the bandwidth matrix H_i^s as:
   $$H_i^s \leftarrow \frac{\Sigma_{j=1}^n \left\|\frac{x_i^r - x_j^r}{h^r(H_i^s)}\right\|^2 (x_j^s - x_i^s)(x_j^s - x_i^s)^T}{\Sigma_{j=1}^n \left\|\frac{x_i^r - x_j^r}{h^r(H_i^s)}\right\|^2}$$
   - Modulate H_i^s as discussed in the previous section. For image segmentation, apply the modulations for exaggerating eccentricity (13) and modifying overall scale (16) sequentially; for video segmentation, sequentially apply the modulations for eccentricity (13), scaling for static segments (15), and overall scale (16).
   - Modulate color tolerance $h^r(H_i^s)$ as described in (17).
3. Repeat step (2) a fixed number of times (typically 3).
4. Associate a mean shift point $M(x_i)$ with every feature point (pixel), x_i, and initialize it to coincide with that point. Repeat for each $M(x_i)$
   - Determine the neighbors, x_j, of $M(x_i)$ as in (18) replacing x_i with $M(x_i)$.
   - Calculate the mean shift vector summing over the neighbors:
   $$M_v(x_i) = \frac{\Sigma_{j=1}^n (x_j - M(x_i)) \left\|\frac{M(x_i^r) - x_j^r}{h^r(H_j^s)}\right\|^2}{\Sigma_{j=1}^n \left\|\frac{M(x_i^r) - x_j^r}{h^r(H_j^s)}\right\|^2}$$
   - Update the mean shift point:
   $$M(x_i) \leftarrow M(x_i) + M_v(x_i)$$

   until $M_v(x_i)$ is less than a specified epsilon.
5. Merge pixels whose mean vectors are approximately the same to produce homogenous color regions.
6. Optionally, eliminate segments containing less than a given number of pixels.

2.4 Initial Scale Selection

As in traditional mean shift image segmentation, the anisotropic kernel mean shift segmentation algorithm also relies on two initial parameters: the initial bandwidths in space and range domain. However, since the bandwidth matrices H_i^s and the bandwidth in range domain $h^r(H_i^s)$ are adaptively modulated, the proposed algorithm is more robust to the initial parameters.

To further increase the robustness, one may also adopt the semiparametric scale selection method described in [3]. The system automatically determines an initial spatial bandwidth for each kernel associated with a point. The user is thus required to set only one parameter: the bandwidth h_0^r in range domain. The local scale is given as the bandwidth that maximizes the norm of the normalized mean shift vector. Refer to [3] for the detailed description and proof.

3 Results

We have experimented with the anisotropic mean shift procedure outlined above on a number of video and still imagery. The first set of images are taken from a short 10 second video of a girl swinging on monkey bars taken from a stationary camera. We first examine a ten frame sequence. We segmented the frames in three ways: 1) each individually with a standard radially symmetric kernel, 2) segmenting the 3D block of video with radially symmetric kernels, and 3) with 3D anisotropic kernels. The results are shown in Figure 1 along with summed pairwise differences between frames. The expected temporal coherence from the stationary camera is faithfully captured in the anisotropic case. A detail of the monkey bars (Figure 2) shows how salient features such as the straight bars are also better preserved. Finally, we show the comparison of symmetric vs. anisotropic kernels on spatio-temporal slices from the monkey bars sequence (Figure 3) and the well known garden sequence (Figure 4) that show much improved segmentation along the trajectories of objects typically found in video. A last example run on a zebra image shows improvement as well in capturing long thin features.

3.1 Robustness

The anisotropic kernel mean shift is more robust to initial parameters than the traditional mean shift. To test this, we correlated the number of segmented regions to the *analysis resolution* on the monkey bars spatio-temporal slice. We fixed h_r to be 6.5 (in the 0 to 255 color space) in both cases. The analysis resolution is then defined as h^s for the fixed symmetric kernels, and the average λ value from the decomposition of the H_i^s in equation (11). As expected, the number of segments increases as the analysis resolution decreases in both cases (see Figure 2). However, the slope is almost twice as steep in the radially symmetric case as with the anisotropic kernel. This indicates that the traditional algorithm is more sensitive to initial parameters than the proposed algorithm. Furthermore, by incorporating the scale selection method, the algorithm automatically selects initial spatial bandwidth.

4 Discussion

Mean shift methods have gained popularity for image and video segmentation due to their lack of reliance on a priori knowledge of the number of expected segments. Most previous methods have relied on radially symmetric kernels. We have shown why such kernels are not optimal, especially for video that exhibits long thin structures in the spatio-temporal slices. We have extended mean shift to allow for anisotropic kernels and demonstrated their superior behavior on both still images and a short video sequence.

The anisotropic kernels plus the sample point density estimation both make the inner loop of the mean shift procedure more complex. We are currently

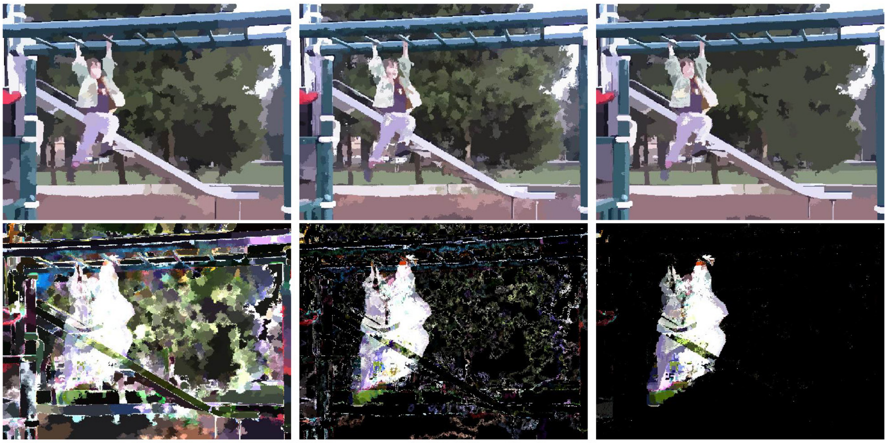

Fig. 1. First row: Segmentation for 2D radially symmetric kernel, 3D symmetric kernel, 3D anisotropic kernel. Note the larger background segments in the anisotropic case while preserving detail in the girl. Second row: total absolute differences across nine pairs of subsequent frames in a ten frame sequence, 2D, 3D radially symmetric, 3D anisotropic. Note the clean segmentation of the moving girl from the background.

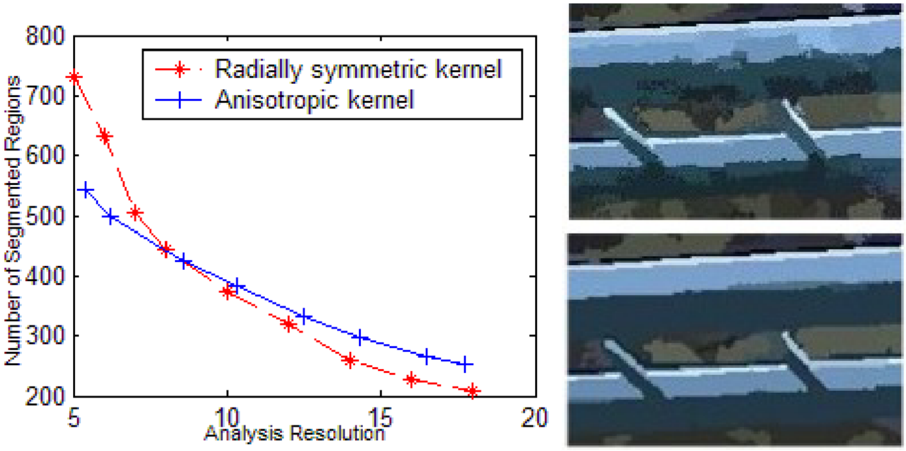

Fig. 2. Left: Robustness results. Right: Monkey Bar detail between 3D radially symmetric kernel result (top) and anisotropic result (bottom).

working on ways to make this more efficient by recognizing pixels that move together early in the iterative process.

It would be nice to have a formal way to objectively analyze the relative success of different mean shift segmentation procedures. Applications such as determining optical flow directly from the kernel orientations might provide a

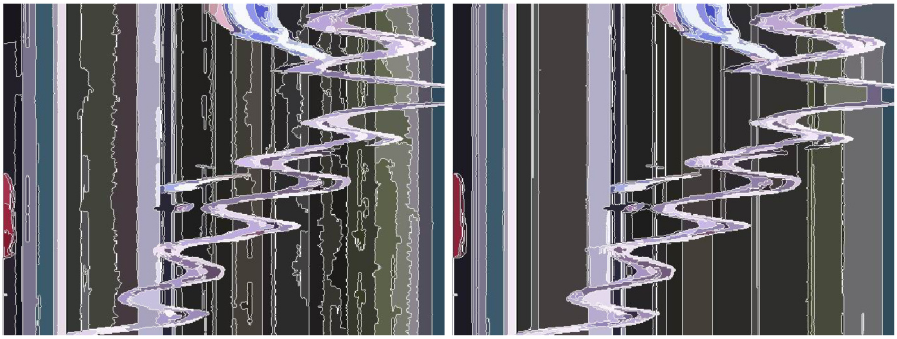

Fig. 3. Spatio-temporal slice of 10 second video segmented by radially symmetric kernel mean shift (left, 384 segments) and anisotropic kernel mean shift (right, 394 segments). Note the temporal coherence indicated by the straight vertical segmentation.

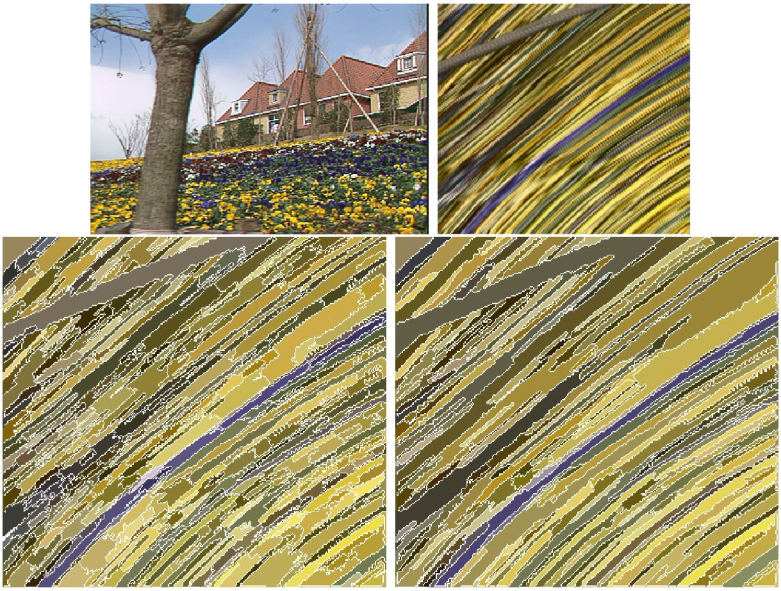

Fig. 4. Well known garden sequence frame and an epipolar slice. Radially symmetric and Anisotropic segmentation (267 and 266 segments).

useful metric. We also look forward to applying our methods to one of our original motivations; automatically producing cartoon-like animations from video.

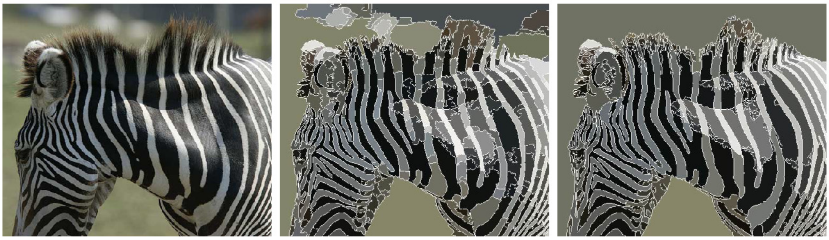

Fig. 5. Zebra photograph. Segmentation with radially symmetric and anisotropic kernels (386 and 387 segments).

References

1. Comaniciu, D., Meer, P.: Mean shift analysis and applications. Proc. IEEE Int. Conf. on Computer Vision, Greece (1999) 1197-1203.
2. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (2000) 142-151.
3. DeMenthon, D., Megret, R.: The variable bandwidth mean shift and data-driven scale selection. Proc. IEEE 8th Int. Conf. on Computer Vision, Canada (2001) 438-445.
4. Comaniciu, D.: An Algorithm for Data-Driven Bandwidth Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 2, February 2003 (2003).
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. on PAMI (2002) 603-619.
6. DeMenthon, D., Megret, R.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (2000) 142-151.
7. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Information Theory **21** (1975) 32–40.
8. Lorensen, W.E., Cline, H.E.: Marching Cubes: a high resolution 3D surface reconstruction algorithm. Proc. ACM SIGGRAPH 1987, (1987) 163-169.
9. Megret, R., DeMenthon, D.: A Survey of Spatio-Temporal Grouping Techniques. Technical report: LAMP-TR-094/CS-TR-4403, University of Maryland, College Park (1994).
10. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recognition **26** 9 (1993) 1277–1294.
11. Skarbek, W., Koschan, A.: Colour Image Segmentation: A survey. Technical report, Technical University Berlin (1994).
12. Singh, M., Ahuja, N.: Regression Based Bandwidth Selection for Segmentation using Parzen Windows. Proc. IEEE International Conference on Computer Vision (2003) **1** 2–9.
13. Wand, M., Jones, M.: Kernel Smoothing. Chapman & Hall (1995) p. 95.