A Topological Approach to Hierarchical Segmentation using Mean Shift

Sylvain Paris Frédo Durand
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

Abstract

Mean shift is a popular method to segment images and videos. Pixels are represented by feature points, and the segmentation is driven by the point density in feature space. In this paper, we introduce the use of Morse theory to interpret mean shift as a topological decomposition of the feature space into density modes. This allows us to build on the watershed technique and design a new algorithm to compute mean-shift segmentations of images and videos. In addition, we introduce the use of topological persistence *to create a segmentation hierarchy. We validated our method by clustering images using color cues. In this context, our technique runs faster than previous work, especially on videos and large images. We evaluated accuracy with a classical benchmark which shows results on par with existing low-level techniques,* i.e. *we do not sacrifice accuracy for speed.*

1. Introduction

Mean shift is a popular low-level segmentation technique for images and videos. It has been used in numerous applications such as noise removal [8], object tracking [11], 3D reconstruction [36], image and video stylization [1, 12, 35], and video editing [33]. Mean shift is not limited to images [14, 21, 22, 30], and in addition, it can be used in place of other segmentation algorithms, *e.g*. spectral methods [29, 37] and watersheds [32]. This audience motivated numerous theoretical studies that rigorously characterize its behavior [3–6, 8, 18]. However, mean shift becomes computationally expensive with large data sets such as high-resolution images and video sequences. Although acceleration techniques have been proposed [2,12,13,17,21,33,38], further improvement is still desirable: as an example, processing a second of video still requires time on the order of a couple of minutes [33]. Furthermore, creating a cluster hierarchy usable for multi-scale analysis requires an additional computational effort [7, 12, 13, 23, 33] and the theoretical properties of the multiple levels are often unclear.

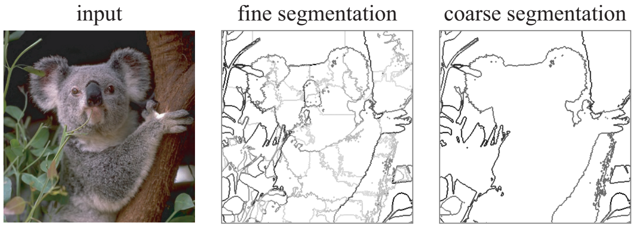

Figure 1. We compute a hierarchical segmentation using mean shift. The segmentation levels are created at no additional costs. Our tests using color features show that our algorithm is fast.

In this paper, we build upon known results of Morse theory, a branch of topology that analyzes manifolds through their singular points. We propose a new algorithm for images and videos that creates a hierarchical segmentation at the computational cost of a single-level clustering. Cheng [6] and Comaniciu and Meer [8] showed that mean shift is equivalent to a steepest ascent on a density function underlying the image data. We describe an efficient scheme to evaluate this function. This explicit representation enables a simple technique to extract the density modes corresponding to the clusters. This approach leads to a fast method to compute mean-shift segmentations. We interpret this algorithm under the light of Morse theory, and show that modes are unions of cells of the Morse-Smale complex. With the notion of *topological persistence* introduced by Edelsbrunner *et al*. [16], we build a hierarchical segmentation. We demonstrate that this procedure has advantages over recursive techniques [13, 33] and scale-space hierarchies [12, 23]. Our focus is not segmentation quality, we concentrate on computational efficiency and the creation of a hierarchy. Experiments on real data show that our algorithm achieves good performances on large images and videos using low-dimensional feature points. We evaluated our technique with a classical benchmark [24, 25] using color similarity. Our technique achieves an accuracy equivalent to previous techniques, demonstrating that our speed-up does not sacrifice precision.

2. Related Work

Mean shift was pioneered by Fukunaga and Hostetler [20]. Given a set of n feature points $\{\mathbf{x}_i\}$

and a seed point $\mathbf{y}_0$, we build a series $\{\mathbf{y}_j\}$ by computing successive averages of data points weighted by a kernel K:

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} K(\mathbf{y}_j - \mathbf{x}_i)\,\mathbf{x}_i}{\sum_{i=1}^{n} K(\mathbf{y}_j - \mathbf{x}_i)} \tag{1}$$

Classical choices for the kernel K are a Gaussian function and a step function. Cheng [6] and Comaniciu and Meer [8] demonstrated that this process is equivalent to a steepest ascent on a density function D defined by the feature points $\{x_i\}$ and a *shadow kernel* $\tilde{K}$:

$$D(\mathbf{p}) = \textstyle\sum_{i=1}^{n} \tilde{K}(\mathbf{p} - \mathbf{x}_i) \tag{2}$$

They showed that $\tilde{K}$ is directly related to K and that, in particular, the shadow of a Gaussian kernel is a Gaussian with the same bandwidth.

Image Segmentation Clustering of the $\{\mathbf{x}_i\}$ set is achieved by placing a seed point $\mathbf{y}_0^i$ at each $\mathbf{x}_i$. Feature points whose corresponding series converge to the same limit are grouped. For image clustering, each pixel is assigned a feature point $\mathbf{x}_i$. For instance, to account for the pixel position (x_i, y_i) and color (r_i, g_i, b_i), 5D vectors are used: $\mathbf{x}_i = (x_i, y_i, r_i, g_i, b_i)$. In this context, $\mathbb{R}^5$ is called the *feature space*. The feature points are segmented according to their limit point, which directly translates into a clustering of the pixels. Since, the $\mathbf{y}_j^i$ sequences are ascending the density function D [6, 8], this process is equivalent to grouping the points according to the density mode which they belong to. Comaniciu and Meer [8] remarked that Gaussian kernels yield better results at the price of less tractable computation. A major contribution of this paper is to design an efficient algorithm to compute a mean-shift segmentation using a Gaussian kernel. Comaniciu *et al.* [10] and Wang *et al.* [34] also showed that adapting the kernel to the local structure of the feature points improve the results at the cost of more computation. We keep such extension as future work.

Gaussian Mean Shift Carreira-Perpiñán related the Gaussian mean shift to the EM algorithm [4] and to spectral clustering [3]. He showed that the convergence rate is lower near mode boundaries [3], that boundaries can have complex shapes [4], and that there can be modes containing no feature points [5]. Comaniciu and Meer [8] demonstrated that the path $\{\mathbf{y}_j\}$ is smooth, and that mean-shift is a robust estimator. Van de Weijer and van den Boomgaard pointed out a link with bilateral filtering [31]. Fashing and Tomasi recast it as a bound optimization [18].

On the practical side, a brute-force computation of the $\{\mathbf{y}_j\}$ sequences is computationally expensive because of the repeated weighted averages (Eq. 1) (each $\mathbf{y}_j$ queries all $\mathbf{x}_i$). To speed up the process, Comaniciu and Meer [8] use axis-aligned box windows. This produces many limit points and adjacent points are merged as a post-process. This algorithm is then equivalent to graph partitioning [19]. Elgammal *et al.* [17] factor the computation of the Gaussian functions. Georgescu *et al.* [21] perform fast nearest-neighbor queries with spatially coherent hash tables. Yang *et al.* [38] and Carreira-Perpiñán [2] accelerate the process by applying Newton iterations. Comaniciu and Meer [8] and Carreira-Perpiñán [2] further reduce the computational cost with dedicated downsampling schemes. All these acceleration techniques deal with the feature points and the mean-shift iterations (Eq. 1). In comparison, we concentrates on the density function D (Eq. 2) that we represent using a coarse grid following similar arguments to Comaniciu and Meer, and Carreira-Perpiñán. One of our contribution is to classify most pixels without iterating, thereby sidestepping the bottlenecks of the previous methods.

Hierarchical Segmentation A cluster hierarchy is a powerful tool to analyze data at multiple scales. Several approaches exist to construct such a multi-level structure.

Recursive segmentation has been initially proposed to accelerate computation [13, 23, 33]. Small clusters are created using a small kernel, and larger segments are formed by applying mean shift on the cluster centroids. The advantages are a faster computation since only a few neighbors are considered at each step, and a tree structure useful for multi-scale tasks [12]. The downside is that the hierarchy levels are arbitrary chosen and a level cannot be added without altering all the subsequent levels. At the cost of more computation, Leung *et al.* [23] compute segmentations for several bandwidths. The clusters become bigger with larger kernels and by construction the levels are independent. But successive segmentations do not form a hierarchical tree anymore because large clusters are not guaranteed to be unions of small clusters [5, 12, 23]. Comaniciu addresses this issue by determining the “strength” of the boundaries between clusters and merging segments weakly separated [7]. This produces a hierarchical structure and one can access any level without modifying the hierarchy. However, it involves a computationally expensive process to find the saddles between density modes [9].

Our approach is along the line of Comaniciu’s saddle detection with the major advantage that we find saddle points at a negligible cost since we explicitly compute the density function. Furthermore, our merging criterion enables on-the-fly computation of any hierarchical level.

Contributions

This paper introduces the following contributions:

▷ A new fast algorithm to compute a Gaussian mean-shift segmentation of an image or a video from an explicit representation of the underlying density function.

▷ Efficient numerical schemes to evaluate this density function and extract its modes.

▷ A hierarchical segmentation based on a topological analysis of mean shift.

We do not aim for better segmentation accuracy and use simple color features. We focus on computational efficiency and the creation of a hierarchy.

3. Background on Morse Theory

Mean-shift sequences converge to local maxima of the underlying density function D. Thus, the shape of the density function, *e.g.* its maxima and saddles, is of primary importance. This motivates us to analyze mean shift with Morse theory which is a vast framework to study the topology of manifolds. In this section, we introduce the concepts which our approach is based on. Figure 2 shows the various entities we use. We refer to dedicated books for an in-depth introduction [26, 27].

Morse-Smale Complex This section defines the topological entities that we handle later. We refer to Edelsbrunner's article [15] for formal definitions.

We study a manifold $\mathbb{M}$ which we view as a terrain to gain intuition. That is, we consider $\mathbb{R}^2$ as a horizontal xy plane, with a height function h defined at each point. We are interested in the steepest-slope paths. These paths end at flat points where $\nabla h = 0$. These locations are either a local maximum (a summit), a saddle, or a local minimum. We name them *critical points*. For a given local maximum $\mathbf{m}$, we define its *stable manifold* as the set of points being on a steepest-slope path ending at $\mathbf{m}$. Equivalently, we define *unstable manifolds* associated to local minima. A *Morse-Smale* cell is the intersection of a stable manifold and an unstable manifold, *i.e.* all points in a Morse-Smale cell are on paths ending at the same two critical points. The *Morse-Smale complex* is the collection of the Morse-Smale cells.

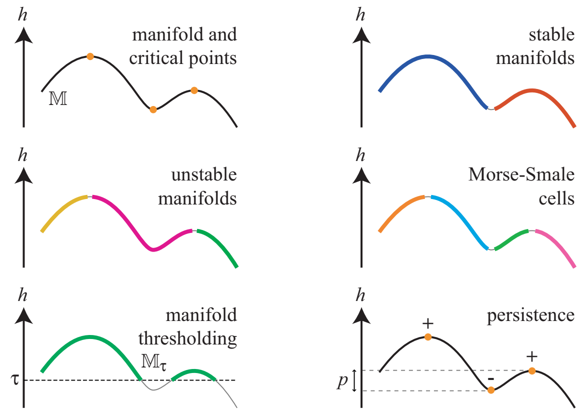

Figure 2. Topological entities shown on a 1D example.

If we know the Morse-Smale complex, for any location, we can tell where we end if we walk up or down.

Topological Persistence The notion of *persistence* quantifies the stability of a topological feature. Considering the terrain analogy, a summit is persistent if it is separated from other summits by low saddles; that is, if one has to walk down a lot before climbing up to the next summit.

Let $\mathbb{M}_\tau$ be the set of all the points higher than τ: $\{u \in \mathbb{M} \mid h(u) > \tau\}$. We observe the topology of $\mathbb{M}_\tau$ as τ goes down from $+\infty$ to $-\infty$. A known result of Morse theory is that $\mathbb{M}_\tau$ changes topology only when the threshold reaches a critical point, *i.e.* when $\tau = h(a)$ with $a \in \mathbb{M}$ a critical point. If the change creates a topological feature, the critical point is said *positive*, if it removes a feature, it is said *negative*. For instance, when τ crosses a local maximum, a new component appears in $\mathbb{M}_\tau$. Thus, a local maximum is a positive critical point. When τ crosses a saddle between two maxima, their components merge, thereby removing a component. This saddle is a negative critical point. Each topological feature is associated to a couple (a, b) of positive and negative critical points. The *persistence* p of a feature is the height difference $|h(a) - h(b)|$. Intuitively, as τ decreases, topological features appear and disappear, and the persistence of a feature is its life time in this process. Another interpretation of the persistence is the "how much h needs to be perturbed so that the feature would not exist." Figure 2 shows a 1D example. Edelsbrunner *et al.* [15, 16] showed that the Morse-Smale complex of $\mathbb{M}$ induced by h can be simplified by canceling the couples (a, b) in order of increasing persistence. Intuitively, the small variations of h are considered irrelevant and the topological features that exist only because of these small variations are canceled. We will use persistence in Section 5.

4. Mean Shift from Density Modes

We build on Cheng [6] and Comaniciu *et al.* [8] who showed that a mean-shift sequence (Eq. 1) corresponds to a steepest ascent on the density D (Eq. 2). Mean-shift segmentation groups the points converging to the same local maximum. Thus, extracting the modes of D is equivalent since points in the same mode converge to the same maximum and belong to the same mean-shift cluster. We demonstrate in the result section that this approach is faster for large kernels.

Algorithm Overview First, we assign a feature point $\mathbf{x}_i$ to each pixel. This step consists in "glueing" the spatial coordinates x and y with the color components expressed in a color space such as RGB or CIE-Lab to form 5D vectors. Then, we compute the density function on a coarse grid. Using this explicit representation, we extract the density

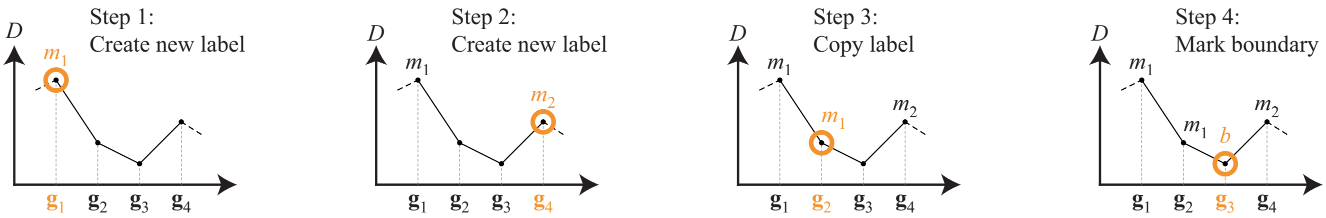

Figure 3. Mode extraction. We process the grid nodes $\mathbf{g}_k$ up to down which leads to the order: $\mathbf{g}_1$, $\mathbf{g}_4$, $\mathbf{g}_2$, $\mathbf{g}_3$. The node $\mathbf{g}_1$ has no label in its neighborhood, thus we create a new label m_1. $\mathbf{g}_4$ is in the same situation and is assigned a new label m_2. The label m_1 is the only one in the neighborhood of $\mathbf{g}_2$, thus $\mathbf{g}_2$ is assigned this label. $\mathbf{g}_3$ is adjacent to two different labels m_1 and m_2 which means it is a contact point between two different modes. We mark it with a special boundary label b.

modes that separate them. Finally, we group pixels whose feature points are in the same mode. The density evaluation and the mode extraction are the core of our technique, and we describe them in Sections 4.1 and 4.2.

4.1. Efficient Computation of the Density Function

Our computational scheme exploits two properties of Gaussian kernels: they produce smooth, low-frequency results, and they are separable.

Since we use a Gaussian kernel G_σ, the density D is a sum of Gaussian functions.Thus, from a signal-processing standpoint, D is band-limited. Although the Gaussian kernel does not completely cut the high frequencies, it ensures that their amplitude is negligible. Similarly to Paris and Durand for bilateral filtering [28], we build upon this property and use a coarse representation of the density function.

In practice, with d-dimensional feature points $\mathbf{x}_i$, we evaluate the density function D on a d-dimensional, regularly spaced grid. We choose the grid step to be equal to the kernel parameter σ. We also exploit the separability of the Gaussian kernel to perform d 1D convolutions (one along each axis) instead of an expensive d-dimensional convolution. In summary, we first bin the feature points in a regular grid, then filter the bin values with a separable Gaussian.

4.2. Mode Extraction

In the previous step, we explicitly sampled the density function D. We name $\{\mathbf{g}_k\}$ the positions of the grid cells, and the density function is represented by the values $\{D(\mathbf{g}_k)\}$. In this section, we extract the modes of D since they contain points that converge to the same local maximum under mean-shift iterations. We associate a label m_l to each local maximum, and mark each grid node $\mathbf{g}_k$ with its corresponding label. The proposed algorithm is similar in spirit to watershed [32] and topological filtration [15].

We sort the nodes $\mathbf{g}_k$ in order of decreasing density and label them one by one. Let $\bar{\mathbf{g}}$ be the currently processed node. All the nodes already labelled have been processed before and thus have a higher density. Conversely, the unprocessed nodes have a lower density. We name $\mathcal{N}_{\bar{\mathbf{g}}} = \{\mathbf{a}_k\}$ the set of nodes adjacent to $\bar{\mathbf{g}}$ in the grid including diagonal neighbors, *i.e.* $\|\bar{\mathbf{g}} - \mathbf{a}_k\|_\infty = \sigma$. To label $\bar{\mathbf{g}}$, we count the number of different labels in $\mathcal{N}_{\bar{\mathbf{g}}}$ and distinguish three cases:

▷ *Zero label*: All neighbors of $\bar{\mathbf{g}}$ have a density lower than $D(\bar{\mathbf{g}})$. $\bar{\mathbf{g}}$ is a local maximum; we create a new label and assign it to $\bar{\mathbf{g}}$.

▷ *One label* m: All neighbors with a density greater than $D(\bar{\mathbf{g}})$ belong to the same mode m. The ascent process necessarily leads to this mode. $\bar{\mathbf{g}}$ is labeled with m.

▷ *Two or more labels*: $\bar{\mathbf{g}}$ is on a boundary between several modes. We mark it with a special boundary label b. Furthermore, if two modes m_1 and m_2 are in contact for the first time, $\bar{\mathbf{g}}$ is a saddle point (the highest contact point between two modes). We keep this information for later use. We do not count the b label when we examine the neighborhood $\mathcal{N}_{\bar{\mathbf{g}}}$. Although this may introduce minor variations of the boundary location, this ensures that the boundary codimension remains 1 (that is, the boundaries have no "volume"), and allows us to classify a larger number of nodes.

Figure 3 illustrates this algorithm on a simple 1D example. Note that higher-dimensional cases are more complex because saddle points do not exist in 1D. Figure 5 shows that most pixels can be classified using this technique.

Refining Boundaries The boundary nodes result in unclassified pixels (Fig. 5). These areas can be filled with a classical mean-shift iteration scheme. We start a series $\mathbf{y}_j^i$ at each unclassified point $\mathbf{x}_i$. We stop iterating when $\mathbf{y}_j^i$ enters a region with a non-boundary label m, and mark the data point with m. Although we use an iterative scheme, the number of iterations is limited since we do not need to reach or even approach convergence. Furthermore, the mean computation (Eq. 1) can be accelerated by adapting the numerical scheme proposed by Paris and Durand for bilateral filtering [28].

5. Hierarchical Segmentation

The previous section produces a partition of the d-dimensional data space that induces a segmentation of the

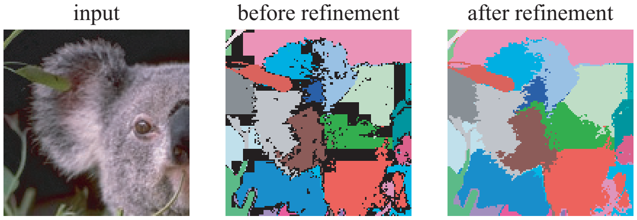

Figure 5. After mode extraction, most pixels are segmented, only boundaries remain unclassified (black pixels in middle image). These pixels are clustered using an iterative process that stops as soon as it reaches a labelled region (result shown on the right).

input image. We now build a hierarchical structure by successively merging pairs of clusters. To this end, we define the *persistence* of the boundary between two modes and merge the weakly separated modes.

In Appendix A, we demonstrate that this definition of persistence is equivalent to the topological persistence, and that we actually simplify the underlying Morse-Smale complex as illustrated on Figure 4. Thus, our technique achieves a topological simplification that accounts for noise and inaccuracies in the density function.

Boundary Persistence We consider two modes m_1 and m_2 with local maxima $\mathbf{m}_1$ and $\mathbf{m}_2$ and densities $D(\mathbf{m}_1) > D(\mathbf{m}_2)$. Let $\mathbf{s}_{12}$ be the saddle between these modes. We define the *boundary persistence* p_b of the boundary between m_1 and m_2 as:

$$p_\mathrm{b}(m_1, m_2) = D(\mathbf{m}_2) - D(\mathbf{s}_{12}) \tag{3}$$

Simplification To simplify the segmentation, we fix a threshold thr and merge clusters whose boundary persistence p_b is less than thr. To build a hierarchy, we increase thr from 0 to $+\infty$. This construction is fast since local maxima and saddles have been detected during mode extraction (Sec. 4.2) and because we do not rerun the mean-shift algorithm. It produces a true hierarchical structure since large clusters are unions of smaller ones. As an alternative, simplification can be performed on-the-fly, that is, the merger decision can be taken during mode extraction (App. B).

Unlike recursive techniques [13, 33], our approach does not require additional computation and any level can be accessed arbitrarily without altering the hierarchy. And compared to increasing the kernel bandwidth [23], persistence-based simplification is guaranteed to produce a hierarchy.

5.1. Controlling the Hierarchy

In the following paragraphs, we expose two ways to control the way the hierarchy is built.

Merging Large Clusters Using the point density to define the boundary persistence favors early fusions of small clusters since the height difference between the maximum and the saddle is bounded. Depending on the application, it may be desirable to merge early large clusters, *e.g.* to group sky pixels in a single big segment. To this end, we use a modified density function $\tilde{D} = f(D)$ to compute the persistence. The proposed technique and the associated properties hold as long as f is an increasing function because it does not change the mode definition nor modifies the segmentation before simplification. $\tilde{D} = \log D$ efficiently balances large and small clusters.

Application-Specific Hierarchy The persistence-based hierarchy accounts only for the topological structure of the feature space. Applications may induce additional domain-specific objectives. Typically, when dealing with images, color is a strong structural cue and it makes sense to favor the fusion of clusters of similar color. This is incorporated in our framework by using a modified persistence $\tilde{p}$. For color images, one can define $\tilde{p} = p \times \|\Delta C\|$ where p is the topological persistence and $\|\Delta C\|$ is the color distance between the two considered modes. One should be aware that modifying the persistence in such way breaks the equivalence between off-line and on-the-fly hierarchy. Nonetheless, the achieved results are often visually more satisfying.

In a probabilistic context, one could use the criterion proposed by Comaniciu *et al.* [9]. We keep such extension as future work.

6. Results

Running Time We timed our algorithm on an AMD Opteron 2.6GHz with 1MB of cache with an 8-megapixel picture (Fig. 6). On-the-fly simplification is about twice faster than the off-line algorithm to access the same hierarchy level because there are fewer boundary pixels requiring

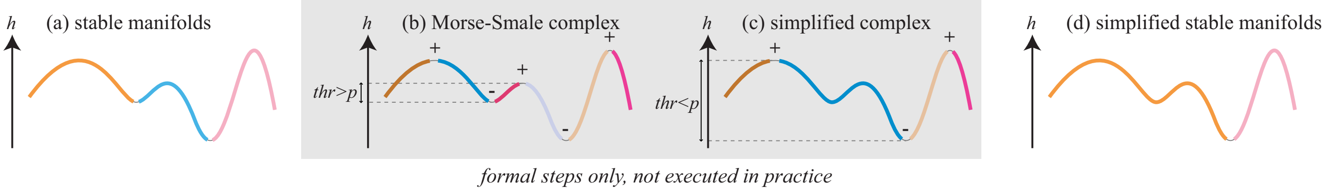

Figure 4. Formal simplification process on a 1D example. Formally, we deal with the Morse-Smale complex (b) underlying the stable manifolds which represent the density modes (a). We cancel the pairs of critical points with a persistence p less than a threshold thr (b,c), and extract the stable manifolds of the resulting complex (d). We demonstrate that the mode simplification can be done without explicitly computing the Morse-Smale complex, *i.e.* we go directly from (a) to (d).

mean-shift iterations. The cluster-merging step is negligible compared to the other steps. Moreover, our algorithm runs faster with larger bandwidth since it allows for coarser sampling of the feature space. This property is advantageous on large images containing millions of pixels. For these pictures, typical spatial bandwidths are tens of pixels and above. We compared with the EDISON system [8] whose running times increase quadratically with the spatial bandwidth: 2min 16s for 16 pixels, 39min for 64 pixels, and so on. We also tested the graph-based method [19] which is among the fastest techniques to compute a single-level segmentation. It consistently runs in $\approx$ 18s for all settings. Our algorithm becomes faster for spatial bandwidths greater than ≈ 64 pixels while producing a hierarchical structure. Table 1 shows the time used by each step. The off-line computation is dominated by the iterative boundary refinement. However these 15s has to be compared to the 39min required by EDISON which is purely iterative. On-the-fly computation speeds up this step by an order of magnitude because there are fewer boundary pixels.

For faster performance, the dimensionality of the feature space can be reduced using principal component analysis. On color images (Fig. 7), our tests showed that the original 5D space can be reduced to 4D at almost no loss. Reducing to 3D removes some medium-contrast details, yet the results are still sufficient for a number of applications. PCA dramatically reduces running times as shown on Figure 6 and Table 1. For practical use, reduction to 4D yields the most valuable tradeoff accuracy versus running time. Removing two dimensions should be reserved to applications where performances are critical. PCA is also useful for movies to reduce the required memory. On 6s of videos (46 megapixels), reducing the 6D $xytLab$ space to 5D yields quality results in 5min51s, and reducing to 4D achieves satisfying results in 40s. For comparison, Wang *et al*. [33] reported running times of about 10min on a similar sequence.

Accuracy We tested our algorithm with the Berkeley benchmark [24, 25] that compares computer-generated clusterings to man-made segmentations. Our goal is to check whether our algorithm sacrifices accuracy for performances.

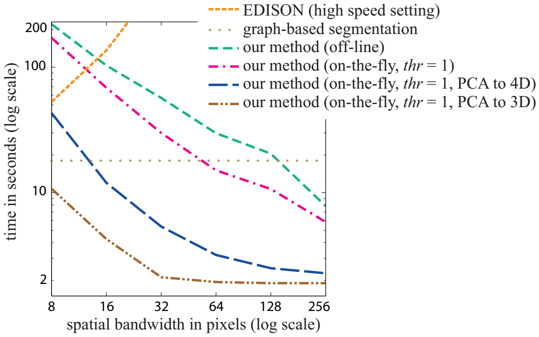

Figure 6. Running times of our algorithm and existing techniques on an 8-megapixel image with a color bandwidth equal to 5% of the intensity range, and an increasing spatial bandwidth.

	Off-line	On-the-fly	On-the-fly
		thr $= 1$	*thr* $= 1$, PCA to 4D
Gaussian	6.08s	6.07s	0.59s
Sort	0.67s	0.67s	0.03s
Mode ext.	1.73s	1.81s	0.22s
Boundaries	15.58s	1.04s	1.02s
Other	5.23s	5.24s	1.15s
Total	29.29s	14.83s	3.01s

Table 1. Timing of each step: Gaussian estimation, density sample sort, mode extraction, boundary refinement, and other tasks such as data structure management. We use the same 8-megapixel picture as Figure 6 and a spatial bandwidth of 64 pixels.

Recall that we do not claim any improvement of the segmentation quality. We chose a classical test scenario based on color cues in the CIE-Lab color space, that is: $\mathbf{x}_i = (x_i, y_i, L_i, a_i, b_i)$. We tested our method in a low-level context: we did not “train” our algorithm, and we used the same parameters for all images. Our technique behaves as expected in this scenario (Fig 8): colorful pictures are well segmented whereas textured and camouflaged objects such as the soldiers and the tigers are poorly detected. The benchmark numerically evaluates accuracy using the *F measure* that equals 1 for exact matches with man-made clusters, and decreases with false and missing boundaries (details in [24]). Martin *et al*. report F values between 0.56 and 0.59 using color cues [24]. Our technique achieves 0.61 on average which is consistent with Martin’s study. In summary, our algorithm achieves segmentations on par with existing techniques while computing a hierarchy and being significantly faster.

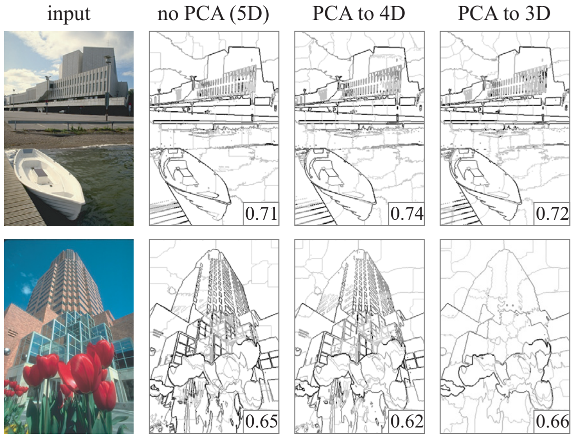

Figure 7. The dimensionality of the feature space can reduced with PCA. Removing one dimension is a good tradeoff since it does not incur visible defects and significantly shortens running times. Further speed-up can be obtained by removing two dimensions although some medium-contrast features are lost (see the windows on the second row). This option is thus more suitable for time-critical applications. The F measure [24] is indicated in the lower-right corner (higher is better, cf. text).

7. Conclusions

We have described a new algorithm to compute a mean-shift segmentation. By recasting the process in Morse theory, we have demonstrated that a hierarchical structure can be computed at a negligible cost. Our approach is especially useful for large images and video sequences where existing techniques are limited. Our technique is as precise as previous work, yet significantly faster and it produces a multi-level hierarchy. These properties make our technique a meaningful choice for many practical applications such as image and video editing, and content recognition.

Acknowledgement The authors thank David Cohen-Steiner and Steven Gortler for their insights on topological persistence; Ted Adelson, Bryan Rusell, and Yair Weiss for their help with the paper; and Jiawen Chen and Emily Whiting for appearing in the video. This work was supported by a National Science Foundation CAREER award 0447561 "Transient Signal Processing for Realistic Imagery," an NSF Grant No. 0429739 "Parametric Analysis and Transfer of Pictorial Style," and a grant from Royal Dutch/Shell Group. Frédo Durand acknowledges a Microsoft Research New Faculty Fellowship and a Sloan Fellowship.

References

[1] A. Bousseau, M. Kaplan, J. Thollot, and F. Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *Int. Symp. on Non-Photorealistic Animation and Rendering*, 2006.

[2] M. Á. Carreira-Perpiñán. Acceleration strategies for Gaussian mean-shift image segmentation. In *Computer Vision and Pattern Recognition*. 2006.

[3] M. Á. Carreira-Perpiñán. Fast nonparametric clustering with Gaussian blurring mean-shift. In *Int. Conf. on Machine Learning*, 2006.

[4] M. Á. Carreira-Perpiñán. Gaussian mean shift is an EM algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, to appear.

[5] M. Á. Carreira-Perpiñán and C. K. I. Williams. On the number of modes of a Gaussian mixture. In *Scale-Space Methods in Computer Vision*, 2003.

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1995.

[7] D. Comaniciu. Image segmentation using clustering with saddle point detection. In *Int. Conf. on Image Processing*. 2002.

[8] D. Comaniciu and P. Meer. A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 2002.

[9] D. Comaniciu, V. Ramesh, and A. D. Bue. Multivariate saddle point detection for statistical clustering. In *Eur. Conf. on Computer Vision*, 2002.

[10] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Int. Conf. on Computer Vision*. 2001.

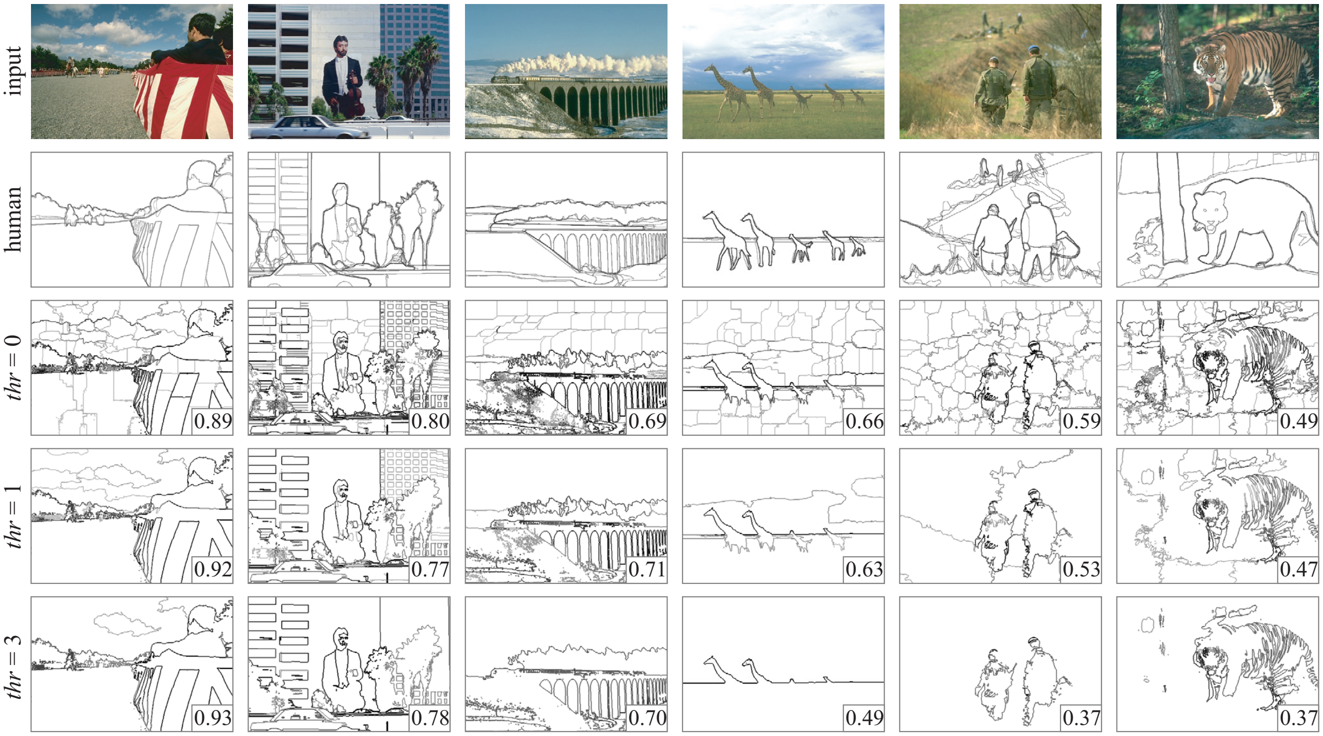

Figure 8. Test images from the Berkeley database [25]. We chose a standard scenario based color cues. Our technique behaves as expected: colorful pictures are well segmented whereas camouflaged objects such as the soldiers and the tiger are poorly detected. The F measure [24] is indicated in the lower-right corner (higher is better, cf. text for details). The gray level of a boundary indicates its persistence (dark boundaries are more persistent).

[11] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 2003.

[12] D. DeCarlo and A. Santella. Stylization and abstraction of photographs. In *ACM SIGGRAPH*, 2002.

[13] D. DeMenthon. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Statistical Methods in Video Processing Workshop*, 2002.

[14] K. Der, R. Sumner, and J. Popović. Inverse kinematics for reduced deformable models. *ACM Trans. on Graphics*, 2006.

[15] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 2003.

[16] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Computational Geometry*, 2002.

[17] A. Elgammal, R. Duraiswami, and L. S. Davis. Efficient kernel density estimation using the fast Gauss transform for computer vision. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 2003.

[18] M. Fashing and C. Tomasi. Mean shift is a bound optimization. *IEEE Trans. on Pattern Analysis Machine Intell.*, 2005.

[19] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *Int. Jour. of Computer Vision*, 2004.

[20] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. in Information Theory*, 1975.

[21] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Int. Conf. on Computer Vision*, 2003.

[22] D. L. James and C. D. Twigg. Skinning mesh animations. *ACM Trans. on Graphics*, 2005.

[23] Y. Leung, J.-S. Zhang, and Z.-B. Xu. Clustering by scale-space filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000.

[24] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 2004.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Int. Conf. on Computer Vision*. 2001.

[26] Y. Matsumoto. *An Introduction to Morse Theory*. American Mathematical Society, 2002.

[27] J. Milnor. *Morse Theory*. Princeton University Press, 1963.

[28] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *Eur. Conf. on Computer Vision*, 2006.

[29] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1997.

[30] O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3D motion estimation via mode finding on lie groups. In *Int. Conf. on Computer Vision*, 2005.

[31] J. van de Weijer and R. van den Boomgaard. Local mode filtering. In *Computer Vision and Pattern Recognition*, 2001.

[32] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1991.

[33] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Video cutout. *ACM Trans. on Graphics*, 2005.

[34] J. Wang, B. Thiesson, Y. Xu, and M. F. Cohen. Image and video segmentation by anisotropic mean shift. In *Eur. Conf. on Computer Vision*, 2004.

[35] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen. Video tooning. *ACM Trans. on Graphics*, 2004.

[36] Y. Wei and L. Quan. Region-based progressive stereo matching. In *Computer Vision and Pattern Recognition*, 2004.

[37] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Int. Conf. on Computer Vision*. 1999.

[38] C. Yang, R. Duraiswami, D. DeMenthon, and L. Davis. Mean-shift analysis using quasi-Newton methods. In *Int. Conf. on Image Processing*, 2003.

A. Equivalence with Topological Persistence

The simplification process using the boundary persistence p_b is actually equivalent to applying a persistence-driven simplification of the Morse-Smale complex induced by the density function D on $\mathbb{R}^d$.

Sketch of Proof: Directly from the definitions (Sec. 3), using the density D as height function h, the density modes are stable manifolds and thus unions of Morse-Smale cells. Since stable manifolds are defined by the local maximum where the steepest-slope paths converge to, only pair cancellations involving a maximum have an impact on the mode decomposition. When we analyze the topological changes of $\mathbb{M}_\tau$ when τ decreases, local maxima always correspond to the creation of an isolated connected component. By definition, there is no higher point in their neighborhood. Connected components disappear when they get "attached" to another component whose maximum is higher. This event occurs at a saddle point. Thus, a local maximum is always linked to a saddle point. In conclusion, stable manifolds are only affected by cancellation of pairs involving a saddle point $\mathbf{s}_{12}$ between two maxima $\mathbf{m}_1$ and $\mathbf{m}_2$ with $D(\mathbf{m}_1) > D(\mathbf{m}_2)$. The associated persistence p is equal to $D(\mathbf{m}_2) - D(\mathbf{s}_{12}) = p_b(m_1, m_2)$. □

B. On-the-fly Simplification

The off-line construction of a hierarchy level T first computes the whole segmentation and then increases the simplification threshold thr from 0 to the value T. We demonstrate that modes can be merged as soon as the saddle between them is reached during mode extraction, and that this yields the same result as the off-line technique.

Sketch of Proof: We consider a saddle $\mathbf{s}_{12}$ between modes m_1 and m_2 with local maxima $\mathbf{m}_1$ and $\mathbf{m}_2$ and $D(\mathbf{m}_1) > D(\mathbf{m}_2)$ such that $p_b(m_1, m_2) < T$. We observe the mode-extraction process when it reaches $\mathbf{s}_{12}$. We assumes that modes are merged on-the-fly. Merging m_1 and m_2 does not alter the persistence of the m_1 boundaries since $\mathbf{m}_1$ is the highest point of $m_1 \cup m_2$. However, the persistence of the m_2 boundaries is modified. For the equivalence to hold, the $m_1|m_2$ boundary must be the first one removed by the off-line technique. Since the nodes are processed in decreasing order of density, $\mathbf{s}_{12}$ is the highest saddle involving m_2, therefore the weakest one. Hence it would be removed first as thr increases from 0 to T. □