Pacific Graphics 2016
E. Grinspun, B. Bickel, and Y. Dobashi
(Guest Editors)

Volume 35 (2016), Number 7

Anisotropic Superpixel Generation Based on Mahalanobis Distance

Yiqi Cai[1] and Xiaohu Guo[†1]

[1]University of Texas at Dallas, USA

Abstract

Superpixels have been widely used as a preprocessing step in various computer vision tasks. Spatial compactness and color homogeneity are the two key factors determining the quality of the superpixel representation. In this paper, these two objectives are considered separately and anisotropic superpixels are generated to better adapt to local image content. We develop a unimodular Gaussian generative model to guide the color homogeneity within a superpixel by learning local pixel color variations. It turns out maximizing the log-likelihood of our generative model is equivalent to solving a Centroidal Voronoi Tessellation (CVT) problem. Moreover, we provide the theoretical guarantee that the CVT result is invariant to affine illumination change, which makes our anisotropic superpixel generation algorithm well suited for image/video analysis in varying illumination environment. The effectiveness of our method in image/video superpixel generation is demonstrated through the comparison with other state-of-the-art methods.

1. Introduction

Images can be compactly represented by a collection of perceptually meaningful over-segments. This higher-level representation can greatly reduce the computation complexity. Compared with the multi-resolution representation, it captures structure boundaries and provides better support for region-based features [WZG*13]. Thus, superpixels have been widely used as a preprocessing step in various computer vision tasks, such as segmentation [FH04, LSK*09, ASS*12], object tracking [YLY14], stereo 3D reconstruction [MK10] and interactive image cutout [LSTS04].

Video over-segmentation generalizes the clustering from spatial pixels to *spatio-temporal* pixels. It is a challenging task since the temporal dimension can introduce camera-motion, object occlusion, non-rigid deformation, changes in scale, perspective and illumination [GCS12]. Compared with image over-segmentation, a good video superpixel representation consider additional metrics, such as spatio-temporal coherence [XC12, CWFI13] and video length scalability [GKHE10].

We discuss the generation of superpixels with spatial compactness and color homogeneity. In this paper, these two objectives are considered separately and anisotropic superpixels are generated to better adapt to local image contents. We develop an unimodular Gaussian generative model to guide the color homogeneity within a superpixel by learning local pixel color variations. It turns out maximizing the log-likelihood of our generative model is equivalent to solving an Centroidal Voronoi Tessellation (CVT) problem. Moreover, we provide the theoretical guarantee that the CVT result is invariant to affine illumination change, which makes our anisotropic superpixel generation algorithm well suited for image/video analysis in varying illumination environment. The effectiveness of our method in image/video superpixel generation is demonstrated through the comparison with other state-of-the-art methods.

2. Existing Works

Image Superpixels: Image superpixel generation algorithms can be coarsely divided into graph-based methods and clustering-based methods.

Treating image pixels as individual nodes, graph-based methods use probabilistic connections to model the probability of node's hidden object class. Naturally, the superpixel generation is to find a partition minimizing a well defined graph partition cost. The widely used methods include Nyström normalized cut [SM00, FBCM04], the Felzenszwalb-Huttenlocher method [FH04], Superpixel Lattices [MPW*08] and Weighed Aggregation [SBB00, SGS*06].

For the clustering-based methods, usually a criteria is defined to measure the appropriateness of grouping pixels into a cluster. The clustering is iteratively refined until energy convergence. Existing approaches include Meanshift [FH75, CM02], Turbopixels [LSK*09], Simple Linear Iterative Clustering (SLIC) [ASS*12], Structure Sensitive Superpixels (SSS) [WZG*13] and Manifold SLIC [YJLH16]. We consider our method as one of the clustering-based methods.

Video Superpixels: With the temporal dimension introduced, video over-segmentation is a natural extension of image over-segmentation. Depending on the scalability of video

† Corresponding author

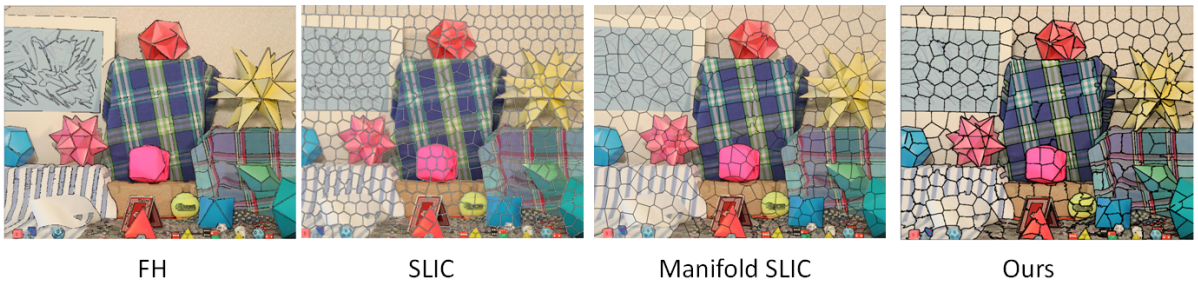

Figure 1: *An illustration of superpixels obtained by FH [FH04], SLIC [ASS*12], Manifold SLIC [YJLH16] and our method. We notice FH superpixels lack spatial compactness. Compared with SLIC and Manifold SLIC, our result provides better boundary adherence. Also the small toys are nicely captured by our method. See Sec. 6.1 for detailed comparison on segmentation error, boundary recall and achievable segmentation accuracy.*

length [GKHE10], video superpixel generation algorithms can be classified into offline algorithms and streaming algorithms.

Offline video superpixel algorithms require the video to be available in advance and short enough to fit in memory. To enforce the locality of superpixel boundaries over different frames, a popular approach [XC12] is to use volumetric representation along different frames. Due to the fact that object with non-negligible motion may contradict with the neighborhood definition of volumetric data, it has been noticed [GKHE10] that this technique does not improve the long-term spatio-temporal coherence. Thus, many algorithms [GKHE10, CWFI13] treat the temporal dimension differently and infer pixel correspondences across frames. Then the nodes in the 3D graph are connected by the inferred flow vectors with robust long-term correspondences.

To achieve video length scalability [GKHE10], streaming algorithms usually apply a window range and their results are the approximation of their corresponding offline algorithms. Our method is a streaming algorithm since it process the video frame by frame. To avoid unstable segmentation results when frames are treated independently, we optimize the superpixels of each frame from the final result of its previous frame. Due to the affine illumination invariant property in Sec. 4.4, our over-segments are optimized mainly to accommodate local structure motions.

CVT on Superpixel Generation: CVT [DFG99] has been a widely used tool to generate isotropic tessellations on surfaces. Its application on image processing was introduced by Du et al. [DGJW06]. SLIC [ASS*12] extends the concept of Voronoi cells to superpixels. More specifically, SLIC computes the CVT on the image manifold in a Euclidean space with location and color information combined. Even though SLIC produces uniform superpixels, it is observed that [WZG*13] image representation quality could be improved by adapting superpixel densities according to image contents. Thus, SSS [WZG*13] and Manifold SLIC [YJLH16] generate structure sensitive superpixels, whose size are carefully adapted w.r.t. local color variation.

3. Preliminaries

Since our work is closely related to SLIC [ASS*12] and Mahalanobis CVT (MCVT) [RA15], we briefly introduce these works before our anisotropic superpixels.

3.1. SLIC

For a pixel $\mathbf{x} = (u, v)$ in 2D image $\mathcal{I}$, SLIC represents its color in CIELAB space $\mathbf{c}(\mathbf{x}) = (l(\mathbf{x}), a(\mathbf{x}), b(\mathbf{x}))$. The distance of two pixels is measured as the normalized Euclidean distance in $\mathcal{R}^5$ with location and CIELAB color information combined, i.e.,

$$d(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{(d_x/N_s)^2 + d_c/N_c)^2}, \tag{1}$$

where N_s and N_c are two constants to balance the inconsistency of spatial and color proximity. d_x and d_c are the spatial and color Euclidean distances respectively:

$$d_x = ||\mathbf{x_1} - \mathbf{x_2}||^2,$$
$$d_c = ||\mathbf{c}(\mathbf{x_1}) - \mathbf{c}(\mathbf{x_2})||^2.$$

Starting from k evenly sampled cluster center $\{\mathbf{x}_i\}_{i=1}^k$, SLIC uses the classic k-means algorithm to optimize the partition based on the distance measure of Eq. 1. Mathematically speaking, SLIC is an application of CVT on superpixel generation. Denote the corresponding 5D point of each pixel $\mathbf{x}$ as $\mathbf{p} = (\mathbf{x}/\sqrt{N_s}, \mathbf{c}(\mathbf{x})/\sqrt{N_c})$, SLIC partitions the image $\mathcal{I}$ by the Voronoi cells $\{\mathcal{C}_i\}_{i=1}^k$ when centroids coincide with sites. As discussed by Du et al. [DFG99], it is the minimizer of the following CVT energy function:

$$E(\{\mathcal{C}_i\}_{i=1}^k) = \sum_{i=1}^{k} \sum_{\mathbf{p} \in \mathcal{C}_i} ||\mathbf{p} - \tilde{\mathbf{p}}(\mathcal{C}_i)||^2,$$

where $\tilde{\mathbf{p}}(\mathcal{C})$ is the centroid of the cell: $\tilde{\mathbf{p}}(\mathcal{C}) = \sum_{\mathbf{p} \in \mathcal{C}} \mathbf{p}/|\mathcal{C}|$.

SLIC generates uniform and isotropic partitions due to its CVT nature. SLIC has been extended to adaptive partitions by SSS [WZG*13] and Manifold SLIC [YJLH16]. It is well known that image has an anisotropic nature of its contents. As its consequence, anisotropic diffusion [Wei96] has been proven to be an effective technique to reduce image noise without removing significant parts.

Thus, our idea is to provide an anisotropic superpixel representation where the anisotropy of each Voronoi cell is adapted according to local image contents.

3.2. MCVT

For surface segmentation with anisotropic Voronoi cells, Richter and Alexa [RA15] propose MCVT to learn the local distance metric from the embedding of the surface. MCVT adopts a variant of the Anisotropic CVT (ACVT) energy [DW05] and the metric itself is part of the unknowns to be optimized.

MCVT models the energy of each cell as the integral of distances from an observation point to all the points w.r.t. a metric to be optimized:

$$E_{\mathrm{MCVT}}(\mathcal{C}) = \min_{\hat{\mathbf{p}},\mathbf{M}=\mathbf{M}^\top,|\mathbf{M}|=1} \iint_{\mathcal{C}} (\mathbf{p}-\hat{\mathbf{p}})^\top \mathbf{M}(\mathbf{p}-\hat{\mathbf{p}})\mathrm{d}\mathbf{p}, \quad (2)$$

where the unknowns $\hat{\mathbf{p}}$ and $\mathbf{M}$ are the observation point and metric correspondingly; $\mathbf{p}$ refers to all the points in the cell. Note MCVT constrains the determinant of the metric to unity to avoid the trivial zero matrix being the optimized metric.

Considering a fixed cell $\mathcal{C}$, it is shown the observation point minimizing Eq. 2 is the centroid. Meanwhile, the optimal metric is the inverse covariance matrix normalized to have unit determinant [RA15]:

$$\begin{aligned} \hat{\mathbf{p}} &= \bar{\mathbf{p}}(\mathcal{C}), \\ \mathbf{M} &= |\mathbf{U}(\mathcal{C})|^{\frac{1}{d}}\, \mathbf{U}^{-1}(\mathcal{C}), \end{aligned} \quad (3)$$

where $\mathbf{U}(\mathcal{C})$ is the covariance matrix; $|\mathbf{U}(\mathcal{C})|$ is its determinant and d is the matrix dimension which guarantees $|\mathbf{M}| = 1$.

This solution is a continuous analogy to the Mahalanobis distance up to scale. It is pointed out [CGL*15] that the optimized metric in Eq. 3 provides the optimal anisotropy for surface approximation. Substituting the learned metric into Eq. 2, the energy of the cell can be concisely represented as:

$$E_{\mathrm{MCVT}}(\mathcal{C}) = |\mathbf{U}(\mathcal{C})|^{\frac{1}{d}}.$$

4. Anisotropic Superpixels

In this section, we formally describe our anisotropic superpixels in an energy optimization framework. The optimization method and implementation details are introduced in Sec. 5.

4.1. Problem Formulation via Energy Optimization

Given a 2D image $\mathcal{I}$ with domain Ω, we denote a tessellation of $\mathcal{I}$ with k disjoint partitions as a *k-partition* $\mathcal{P} = \{\mathcal{C}_i\}_{i=1}^k$, which satisfies $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for $i \neq j$, and $\cup_i \mathcal{C}_i = \Omega$. Clearly, not all k-partitions are suitable to serve as superpixel representations. Our goal is to define an energy function $E(\mathcal{P})$ to identify the inappropriateness. We expect the minimizer of the energy function $E(\mathcal{P})$ to characterize the following desired properties for each partition region $\mathcal{C}_i$.

i *Connectedness:* $\mathcal{C}_i$ is a simply connected region.
ii *Compactness:* For non-boundary/non-feature regions, $\mathcal{C}_i$ exhibits regular shape pattern rather than being bad-shaped.
iii *Content Awareness:* The anisotropy and size of $\mathcal{C}_i$ should be adaptive to local image contents.
iv *Boundary Preservation:* The shape of $\mathcal{C}_i$ should be adapted accordingly to preserve object boundaries if necessary.

To let a k-partition $\mathcal{P}$ be a good superpixel representation, we have two expectations: spatial compactness and color homogeneity. More specifically, these two expectations are conflicting by themselves. i.e., properties i and ii require the spatial compactness of each partition region regardless of image contents. On the contrary, properties iii and iv care its color homogeneity and encourage the necessary sacrifice on the region's shape.

Pixels carry two kinds of information: location and color. SLIC [ASS*12] adopts the normalized Euclidean distance in $\mathcal{R}^5$ by assuming location and color are homogeneous dimensions. It is worth noting that due to the heterogeneity of location and color, structure sensitive superpixels [WZG*13] incorporates a dedicated density function to capture the change of image structures.

From our point of view, decoupling location with color information clarifies their difference in essence. We introduce two different energy terms to measure the spatial compactness and color homogeneity, respectively:

$$\begin{aligned} E(\mathcal{P}) &= E_{color}(\mathcal{P}) + E_{spatial}(\mathcal{P}) \\ &= \sum_{i=1}^{k} E_{color}(\mathcal{C}_i) + \lambda E_{spatial}(\mathcal{C}_i), \end{aligned} \quad (4)$$

where λ is a constant balancing the relative importance of the expectations.

4.2. Unimodular Gaussian Generative Model

In this section, we propose a *unimodular Gaussian generative model* to measure the possibility of observing a specific pixel color in a meaningful over-segment $\mathcal{C}$. The color homogeneity energy is defined in Sec. 4.3 based on this model.

We start by introducing Gaussian generative model to superpixel generation. For each meaningful over-segment $\mathcal{C}$, it is associated a color proxy θ with two hidden parameters to be optimized $\theta = (\bar{\mathbf{c}}, \Delta)$, i.e., $\bar{\mathbf{c}}$ is the unbiased over-segment color, Δ is a $d \times d$ symmetry positive definite matrix describing the variation of the generated pixel color. Here d is the dimension of the color space. Typically we consider $d = 3$ for RGB images.

The probability density function (PDF) of θ to generate random variable $\mathbf{c}$ is modeled by the multivariate Gaussian distribution:

$$p(\mathbf{c}|\theta) = \frac{1}{(2\pi)^{\frac{d}{2}}\,|\Delta|^{\frac{1}{2}}} exp(-\frac{d(\mathbf{c},\theta)}{2}),$$

where $d(\mathbf{c},\theta)$ is the bias between a random generated pixel color $\mathbf{c}$ and the color proxy θ:

$$d(\mathbf{c},\theta) = (\mathbf{c}-\bar{\mathbf{c}})^\top \Delta^{-1} (\mathbf{c}-\bar{\mathbf{c}}). \quad (5)$$

It is possible that Δ can be overfitted to explain the high variation contents spanning over different objects. To restrict the descriptive ability of the Gaussian generative model, we introduce the unity

determinant constraint: $|\Delta| = 1$. We denote this constrained model as unimodular Gaussian generative model. Its PDF becomes:

$$p(\mathbf{c}|\theta) = \frac{1}{(2\pi)^{\frac{d}{2}}} exp(-\frac{d(\mathbf{c},\theta)}{2}).$$

4.3. Color Fitting Objective

Suppose the pixel color values are independent and identically distributed (IID). The probability that the color proxy θ generates the observed cluster $\mathcal{C}$ is:

$$p(\mathcal{C}|\theta) = \prod_{\mathbf{x}\in\mathcal{C}} p(\mathcal{I}(\mathbf{x})|\theta).$$

For clarity, we use the log-likelihood:

$$ln\, p(\mathcal{C}|\theta) = -\frac{1}{2}(d\,|\mathcal{C}|\, ln(2\pi) + \sum_{\mathbf{x}\in\mathcal{C}} d(\mathcal{I}(\mathbf{x}),\theta)),$$

where $|\mathcal{C}|$ is the number of pixels in $\mathcal{C}$.

For a fixed cluster $\mathcal{C}$, its color homogeneity is defined as the maximum log-likelihood among all possible color proxies:

$$E_{color}(\mathcal{C}) = \max_{\bar{\mathbf{c}},\Delta=\Delta^\top,|\Delta|=1} -\frac{1}{2}(d\,|\mathcal{C}|\, ln(2\pi) + \sum_{\mathbf{x}\in\mathcal{C}} d(\mathcal{I}(\mathbf{x}),\theta)).$$

When evaluating $E_{color}(\mathcal{P})$, it is clear the first term is a constant when summing over all clusters. Thus the color fitting objective has a more concise representation:

$$\begin{aligned} E_{color}(\mathcal{C}) &= \min_{\bar{\mathbf{c}},\Delta=\Delta^\top,|\Delta|=1} \sum_{\mathbf{x}\in\mathcal{C}} d(\mathcal{I}(\mathbf{x}),\theta) \\ &= \min_{\bar{\mathbf{c}},\Delta=\Delta^\top,|\Delta|=1} \sum_{\mathbf{x}\in\mathcal{C}} (\mathcal{I}(\mathbf{x})-\bar{\mathbf{c}})^\top \Delta^{-1} (\mathcal{I}(\mathbf{x})-\bar{\mathbf{c}}). \end{aligned} \tag{6}$$

Note that Eq. 6 is actually the discrete form of the MCVT energy shown in Eq. 2. Thus maximizing the log-likelihood of our generative model is equivalent to solving MCVT in a discrete manner. Not surprisingly, its optimization leads to the optimal unbiased color $\bar{\mathbf{c}}^\star$ being the mean value of the observations; $(\Delta^\star)^{-1}$ being the normalized inverse covariance matrix:

$$\begin{aligned} \bar{\mathbf{c}}^\star &= \sum_{\mathbf{x}\in\mathcal{C}} \mathcal{I}(\mathbf{x})/\,|\mathcal{C}|\,, \\ (\Delta^\star)^{-1} &= |\mathbf{U}_c(\mathcal{C})|^{\frac{1}{d}}\, \mathbf{U}_c^{-1}(\mathcal{C}). \end{aligned} \tag{7}$$

Note $\mathbf{U}_c(\mathcal{C})$ is the covariance matrix for the observed pixel color values: $\mathbf{U}_c(\mathcal{C}) = \sum_{\mathbf{x}\in\mathcal{C}} (\mathcal{I}(\mathbf{x})-\bar{\mathbf{c}})(\mathcal{I}(\mathbf{x})-\bar{\mathbf{c}})^\top$.

The proof for Eq. 7 is provided in the supplementary material for completeness. It is shown clearly in Eq. 7 that the color fitting objective has a Mahalanobis form when the unimodular Gaussian generative model is adopted to explain the observed data. Actually, the color fitting energy can be simply evaluated from the covariance matrix of the cluster by substituting Eq. 7 into Eq. 6:

$$E_{color}(\mathcal{C}) = |\mathbf{U}_c(\mathcal{C})|^{\frac{1}{d}}. \tag{8}$$

It is shown the color fitting energy, which has a Mahalanobis form similar to MCVT [RA15], has a probabilistic interpretation for superpixel generation. With the assumption of generating pixel color from its color proxy by Gaussian probability IID, the energy in Eq. 8 stands for the log-likelihood of observing all image pixels (up to a constant difference). Also, there is a fundamental difference between Eq. 8 and the MCVT energy of modeling the image as a surface. Eq. 8 only considers the color information within each over-segment. The spatial compactness requirement is discussed in Sec. 4.5.

4.4. Affine Illumination Invariant Property

Let us consider the color fitting energy for image pair $\mathcal{I}$ and $\mathcal{I}'$ in different illumination environments. For general lighting environment variation [SMT13], such as illumination intensity/direction change and ambient light change, the affine variation accounts for the relationship between pixel color values:

$$\mathcal{I}'(\mathbf{x}) = \mathbf{A}\mathcal{I}(\mathbf{x}) + \mathbf{b}, \tag{9}$$

where $\mathbf{A}$ is the linear transformation; $\mathbf{b}$ is the translation vector.

There are literatures aim to reduce the effect of illumination variation by color transformations. Normalized cross-correlation (NCC) [FVT*93] and adaptive NCC (ANCC) [HLL11] are popular transform-based methods to preprocess the image (or the support window) to have a zero mean and unit standard deviation. Usually the color channels are processed separately. Normalized correlation methods are not effective to handle illumination variation [KHKS14]. The main reason is that the affine illumination variation causes color channels to influence each other. With this cross-channel information encoded in the covariance matrix, it is shown [KHKS14] that the Mahalanobis distance is an invariant measure under affine illumination change.

Different from the previous paper [KHKS14], we show the Mahalanobis distance can be used to achieve partition invariant under affine illumination change. In the remainder of the section, we illustrate the *affine illumination invariant property* for the defined color homogeneity in Sec. 4.3: the optimal k-partition minimizing $E_{color}(\mathcal{P})$ is invariant under affine illumination change.

Lemma 1 Under affine illumination change, for every point $\mathbf{x}$ in the fixed partition $\mathcal{P}$, the bias between the pixel color and its associated optimal color proxy $\theta^\star$ is multiplied by a same constant factor, i.e., $\forall \mathbf{x} \quad d(\mathcal{I}'(\mathbf{x}),\theta'^\star) = |\mathbf{A}|^{\frac{2}{d}}\, d(\mathcal{I}(\mathbf{x}),\theta^\star)$.

Proof The optimal color proxy $\theta'^\star = (\bar{\mathbf{c}}'^\star, \Delta'^\star)$ under affine illumination change can be easily solved following Eq. 7:

$$\bar{\mathbf{c}}'^\star = \mathbf{A}\bar{\mathbf{c}}^\star + \mathbf{b}.$$

It is easy to verify $\mathbf{U}_c'(\mathcal{C}) = \mathbf{A}\mathbf{U}_c(\mathcal{C})\mathbf{A}^\top$, thus

$$\begin{aligned} (\Delta'^\star)^{-1} &= |\mathbf{A}|^{\frac{2}{d}}\, |\mathbf{U}(\mathcal{C})|^{\frac{1}{d}}\, (\mathbf{A}\mathbf{U}(\mathcal{C})\mathbf{A}^\top)^{-1} \\ &= |\mathbf{A}|^{\frac{2}{d}}\, (\mathbf{A}^\top)^{-1} (\Delta^\star)^{-1} \mathbf{A}^{-1}. \end{aligned}$$

Substituting the new optimal color proxy into Eq. 5,

$$
\begin{aligned}
& d(\mathcal{I}^{'}(\mathbf{x}), \theta^{'\star}) \\
&= d(\mathbf{A}\mathcal{I}(\mathbf{x}) + \mathbf{b}, \theta^{'\star}) \\
&= (\mathbf{A}(\mathcal{I}(\mathbf{x}) - \bar{\mathbf{c}}^{\star}))^{\top} (|\mathbf{A}|^{\frac{2}{d}} (\mathbf{A}^{\top})^{-1} (\Delta^{\star})^{-1} \mathbf{A}^{-1}) (\mathbf{A}(\mathcal{I}(\mathbf{x}) - \bar{\mathbf{c}}^{\star})) \\
&= |\mathbf{A}|^{\frac{2}{d}} (\mathcal{I}(\mathbf{x}) - \bar{\mathbf{c}}^{\star})^{\top} (\Delta^{\star})^{-1} (\mathcal{I}(\mathbf{x}) - \bar{\mathbf{c}}^{\star}) \\
&= |\mathbf{A}|^{\frac{2}{d}} d(\mathcal{I}(\mathbf{x}), \theta^{\star}) \qquad \square
\end{aligned}
$$

Lemma 2 If the k-partition $\mathcal{P}$ is a minimizer of $E_{color}(\mathcal{P})$ for image $\mathcal{I}$, $\mathcal{P}$ must also be a minimizer for image $\mathcal{I}^{'}$.

Proof Since the energy of each cell in Eq. 6 is equivalent to the MCVT energy, each partition $\mathcal{C}_i \in \mathcal{P}$ must be the Voronoi region $\mathcal{C}_i = \{\mathbf{x} \in \Omega \,|\, d(\mathcal{I}(\mathbf{x}), \theta_i^{\star}) < d(\mathcal{I}(\mathbf{x}), \theta_j^{\star}) \text{ for} \forall j \neq i\}$.

According to Lemma 1, each region is also the Voronoi region for $\mathcal{I}^{'}$, i.e., $\mathcal{C}_i = \{\mathbf{x} \in \Omega \,|\, d(\mathcal{I}^{'}(\mathbf{x}), \theta_i^{'\star}) < d(\mathcal{I}^{'}(\mathbf{x}), \theta_j^{'\star}) \text{ for} \forall j \neq i\}$. Thus, $\mathcal{P}$ is the MCVT minimizing the energy for $\mathcal{I}^{'}$. $\square$

4.5. Spatial Compactness

It has been shown that minimizing CVT energy is closely related to the maximization of the compactness of Voronoi cells [LWL*09]. We adopt this observation and apply the CVT energy on the spatial information as the regularizer in Eq. 4. Different from previous approaches, we propose the CVT energy from the perspective of covariance matrix. This reformulation via covariance matrix allows us to calculate CVT energy in the same way as the color fitting energy of Eq. 8.

For a fixed cluster $\mathcal{C}$, the CVT energy is defined as its inertia momentum:

$$E_{spatial}(\mathcal{C}) = \sum_{\mathbf{x} \in \mathcal{C}} ||\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C})||^2 .$$

It is easy to see the it is equivalent to the trace of the cluster pixels' covariance matrix:

$$
\begin{aligned}
\sum_{\mathbf{x} \in \mathcal{C}} ||\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C})||^2 &= \mathrm{Tr}(\sum_{\mathbf{x} \in \mathcal{C}} (\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C}))^{\top} (\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C})) \\
&= \sum_{\mathbf{x} \in \mathcal{C}} \mathrm{Tr}((\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C}))(\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C}))^{\top}) \qquad (10) \\
&= \mathrm{Tr}(\mathbf{U}_s(\mathcal{C})).
\end{aligned}
$$

Here $\mathbf{U}_s$ is the covariance matrix for pixels spatial coordinates: $\mathbf{U}_s(\mathcal{C}) = \sum_{\mathbf{x} \in \mathcal{C}} ((\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C}))(\mathbf{x} - \bar{\mathbf{x}}(\mathcal{C}))^{\top})$. We use Eq. 10 to regularize the spatial compactness for superpixel generation.

5. Image/Video Superpixel Generation

5.1. k-partition Optimization

From the perspective of energy optimization, there are two widely used approaches for MCVT computation: the Lloyd relaxation scheme [Llo82] and the quasi-Newton approach [LWL*09]. Yet, our problem is in a discrete manner with pixel clustering, this optimization process can be greatly accelerated with a variational merging-swapping framework [CGZM15].

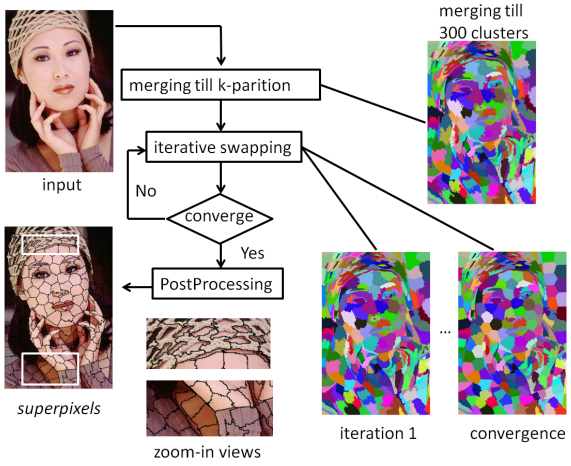

Figure 2: *Illustration of image superpixel generation by computing optimal k-partition.*

The merging step reduces the partition number from the number of pixels to k in a greedy Quadric Error Metric (QEM) fashion [GH97]. Initially, each pixel is treated as an individual cluster. Clusters may choose to merge with its direct neighbors with the amount of increased energy as the merging cost, i.e., for a cluster-pair merging$(\mathcal{C}_i, \mathcal{C}_j) \rightarrow \mathcal{C}_k$, the total increase amount is simply $E(\mathcal{C}_k) - E(\mathcal{C}_i) - E(\mathcal{C}_j)$. All possible cluster-pairs are stored in a min-heap with the least cost pair performed at each time. Only a local computation is needed to update the heap after each merging.

The swapping step optimizes the k-partition by relaxing the pixel binding during successive merging. It is encouraged to swap pixels to neighboring cluster if the swapping decreases the energy. These pixel swappings can be launched in an iterative scheme [CGZM15] until energy convergence. In each iteration, boundary pixels are tested on whether their swappings could possibly decrease the total energy, i.e., for a boundary pixel $\mathbf{x} \in \mathcal{C}_i$ neighboring with $\mathcal{C}_j$, the energy changes from the state of $(\mathcal{C}_i, \mathcal{C}_j)$ to $(\mathcal{C}_i - \mathbf{x}, \mathcal{C}_j + \mathbf{x})$. If a pixel can be swapped to multiple neighboring clusters, the one with largest energy decrement would be selected.

5.2. Image/Video Superpixel Generation Overview

Image superpixels can be generated directly using the merging-swapping scheme described in Sec. 5.1. Like SLIC [ASS*12], our k-partition optimization does not enforce cluster connectivity. For "orphaned" pixels disconnected from the main component of the cluster, we compute the merging costs with their direct neighbor clusters. These pixels are assigned to the least cost neighbor cluster using the merging operation defined in Sec.5.1. Clusters are guaranteed to be connected after the post-processing. Fig. 2 illustrates image superpixel generation process.

Since our color fitting objective has taken the illumination change into account, the partition of successive video frames only change slightly with the assumption of only small changes in the scene. It is natural to extend the discrete variational optimization approach

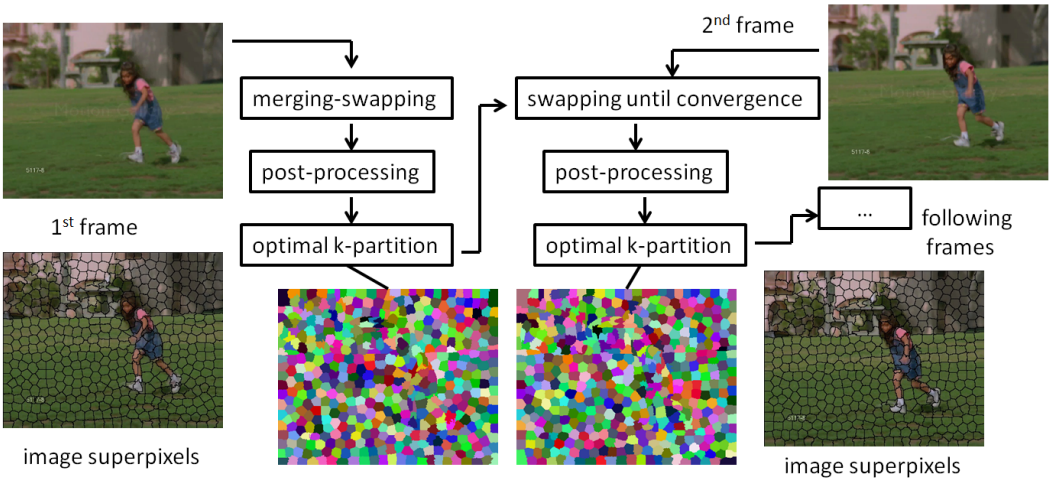

Figure 3: *Flowchart of the proposed video superpixel generation framework based on the swapping operations.*

to video superpixels. The optimal k-partition of the first frame is computed with the same approach as image superpixel generation. For successive frames, only swapping operations are required for local update on the k-partition of the previous frame. Note our approach is a streaming method for video processing. The required memory remains constant regardless of the video length. The flow chart of our proposed video superpixel generation framework is listed in Fig. 3.

5.3. Implementation Details

During the superpixel generation, we track two sets of controids and covariance matrices with respect to color and pixel coordinates respectively. For merging or swapping operations, the update of these statistics can be efficiently assembled from the altered clusters in $\mathbf{O}(1)$ computation, without the need to sum over all pixels. We provide the update rules in Appendix. The change of energies are also easy to evaluate from these matrices using Eq. 8 and Eq. 10.

Merging Initialization: When each over-segment has less than $d+1$ distinct pixel colors, its covariance matrix is degenerate and the color homogeneity fails to learn an effective proxy. Thus it becomes a waste to start the merging process with each pixel as an individual cluster. For our implementation, the merging step is initialized by dividing the image into blocks, i.e., the image is equidistantly divided into $25k$ blocks. With this initialization, the number of cluster-pair merging operation is reduced to $\mathrm{O}(k)$. Even though object boundaries are not well preserved with the block initialization, we noticed the first few iterations of the swapping step will cover this shortcoming of the acceleration.

For the space complexity of our algorithm, the clusters' covariance matrices and the min-heap require additional space. Using the block initialization, this additional space is also reduced to $\mathrm{O}(k)$.

Regularizer Weight: In Eq. 4, the regularizer weight λ determines the importance of the spatial compactness in the final result. Its value is automatically adjusted in our algorithm according to the image/video content. At the beginning, the merging step starts with the empirical value $\lambda = 0.17$. Then this regularizer weight is updated by setting the two objective equally important $\lambda E_{spatial}(\mathcal{P}) = E_{color}(\mathcal{P})$ when a sketch of the superpixels available, i.e., for image superpixel generation, λ is updated at the end of the merging step; for video superpixel generation, λ is updated every frame using the previous optimized partition.

Swapping Parallelization: The swapping test of all boundary pixels in each iteration can be performed parallelly [CGZM15]. We adopt a GPU implementation to speed up this process. For superpixel generation in Sec. 6.3, we achieve around 13 fps on videos with resolution 400×320.

6. Experiments

Our algorithm is implemented using Microsoft Visual C++ 2010. For the hardware platform, the experiments are run on a desktop computer with Intel(R) Core i7-4770 CPU with 3.40GHz, 32GB DDR3 RAM, and NVIDIA GeForce GTX 660 GPU with 2 GB GDDR5 video memory.

6.1. Illumination Variation

In this section, we demonstrate our superpixels are robust against environment illumination changes. The Dolls image pair $(\mathcal{I}_1, \mathcal{I}_2)$ are picked from Middlebury dataset [Hir07] under different illumination settings. We manually transform $\mathcal{I}_1$ to $\mathcal{I}_1^{'}$ using Eq. 9 by randomly generated parameters. Note due to image data precision, the produced image $\mathcal{I}_1^{'}$ is only an approximated affine transformation with truncations. Fig. 4 shows the 500 superpixels Manifold SLIC and Our results on the three images respectively.

To provide a quantitative measurement on the robustness of superpixels, we adopt the achievable segmentation accuracy by assigning each superpixel region a different segmentation label. Relative to a partition $\mathcal{P}$, the *partition robustness* of $\mathcal{P}^{'}$ is defined as:

$$PR(\mathcal{P}, \mathcal{P}^{'}) = \frac{1}{N} \sum_{\mathcal{C}_i \in \mathcal{P}} \max_{\mathcal{C}_j \in \mathcal{P}^{'}} \{|\mathcal{C}_i \cap \mathcal{C}_j|\},$$

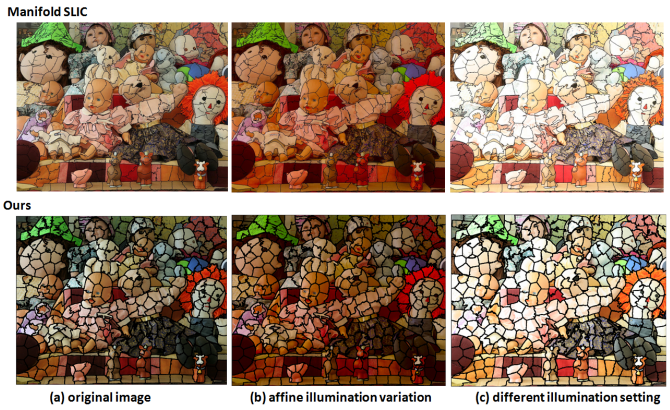

Figure 4: *Superpixels of the Dolls under different illumination settings. (a) Image $\mathcal{I}_1$ selected from Middlebury dataset. (b) Image $\mathcal{I}_1^{'}$ produced by the affine illumination transformation of $\mathcal{I}_1$. (c) Image $\mathcal{I}_2$ selected by different illumination setting.*

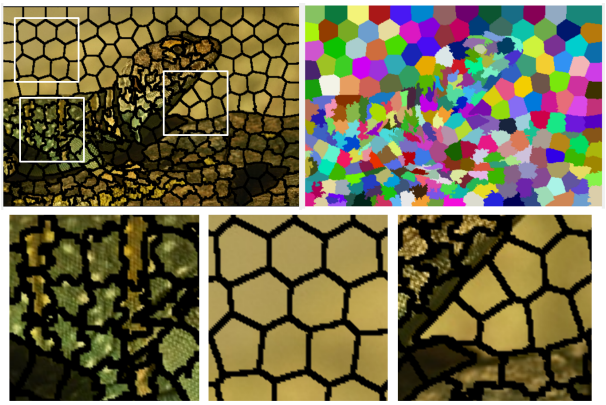

Figure 5: *Superpixel generation on the Chameleon with 500 over-segments.*

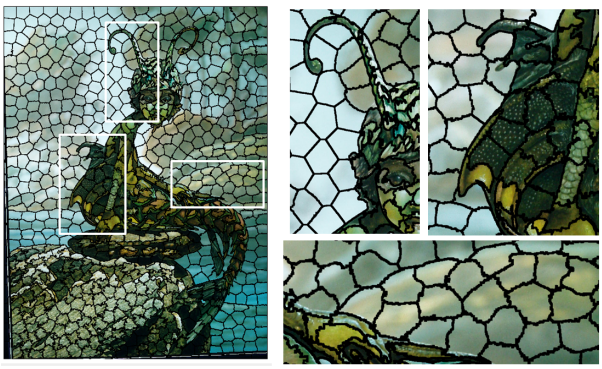

Figure 6: *Superpixel generation on the Mermaid with 1000 over-segments.*

Figure 7: *Representative superpixel generation examples on the INRIA Holidays dataset.*

where N is the total number of pixels. Note for $\mathcal{P}$ and $\mathcal{P}^{'}$ with the same partition number, 100% partition stableness can only be achieved by exactly the same partition.

Let the partition on $\mathcal{I}_1$ be the partition to be evaluated on. Manifold SLIC's partition robustness on $\mathcal{I}_1^{'}$ and $\mathcal{I}_2$ are 68.3% and 62.7%, respectively. While we have much higher robustness with 81.5% and 78.6% correspondingly.

6.2. Image Superpixels

Our superpixel generation algorithm is tested on the Middlebury dataset [Hir07], INRIA Holidays dataset [JDS08] and BSDS500 dataset [AMFM11]. As shown in Fig. 4,5, 6 and 7, our superpixels satisfy the desired properties listed in Sec. 4.1, i.e., adhere to image boundaries while maintain the spatial compactness. Generally speaking, hexagonal tiling is expected in color homogeneous regions. While in regions with color variations, these over-segments adapts its shape and anisotropy to preserve the object boundaries.

The anisotropy adaptation is clearly illustrated in the antenna region of the Mermaid image. Also, the shape adaption can be clearly seen from the text region in Fig. 7.

We compare our anisotropic superpixel with several representative methods on the BSDS500 benchmark [AMFM11]. These methods include FH [FH04], SLIC [ASS*12], Manifold SLIC [YJLH16], Turbopixels [LSK*09] and VCells [WW12]. Following Manifold SLIC, 200 images are randomly selected and evaluated against the provided ground truth segmentation.

Fig. 8 illustrates the quantitative comparison on undersegmentation error, boundary adherence and achievable segmentation accuracy [XC12]. Undersegmentation error [ASS*12] measures the fraction of pixels exceeding the ground truth boundaries when the superpixels are mapped on. Boundary recall [ASS*12] measures the fraction of ground truth boundaries falling within two pixels of the superpixel boundary. Achievable segmentation accuracy [XC12] is the upper bound of segmentation accuracy for a superpixel represen-

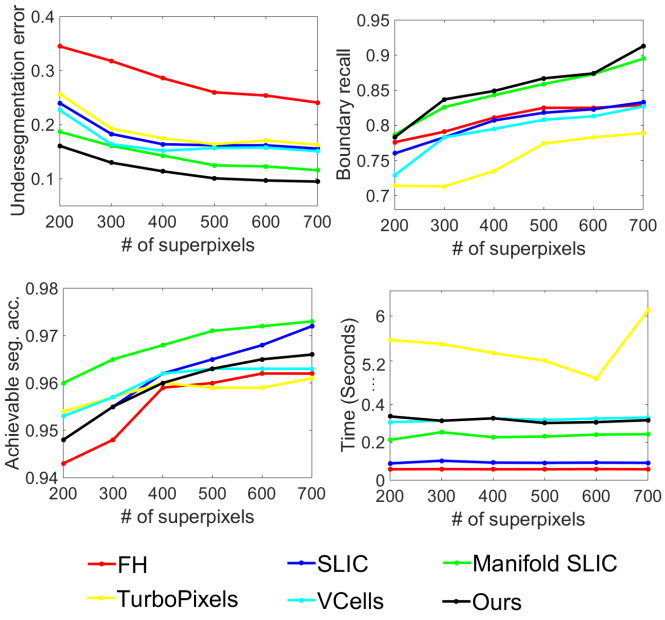

Figure 8: *Evaluation statistics on the BSDS500 benchmark. Compared with FH, SLIC, Manifold SLIC, Turbopixels and VCells, our results have the least undersegmentation error and the highest boundary recall rate. Among all these comparison methods, Manifold SLIC has the most similar quality to ours.*

tation. The result shown in Fig. 8 shows our method provides the least undersegmentation error and almost the same level of boundary recall with Manifold SLIC. For achievable segmentation accuracy, our algorithm performs at the same level with VCells and SLIC. While Manifold SLIC provides the best performance. For runtime performance, our algorithm is at the same level with Manifold SLIC.

6.3. Video Superpixels

For video superpixel generation, we compare our framework in Sec. 5.2 with Temporal Superpixels (TSP) [CWFI13] and the top two 3D graph-based methods according to Xu and Corso [XC12]. The results are reported on the mixture of SegTrack dataset [TFNR12] and Chen dataset [XC12].

Since there is no volumetric representation of the video data in our method, we evaluate the metric using the 2D criteria in Sec. 6.1. For the spatio-temporal coherence measurement in video superpixels, we adopt the mean duration metric [XC12] to measure the number of frames a superpixel exists in. These metrics w.r.t. the number of superpixels are illustrated in Fig. 9. It can be seen that our framework has the best boundary adherence performance due to its least undersegmentation error, comparable boundary recall and highest achievable segmentation accuracy. Also, the mean duration indicates our superpixels are able to track objects in videos for longer periods.

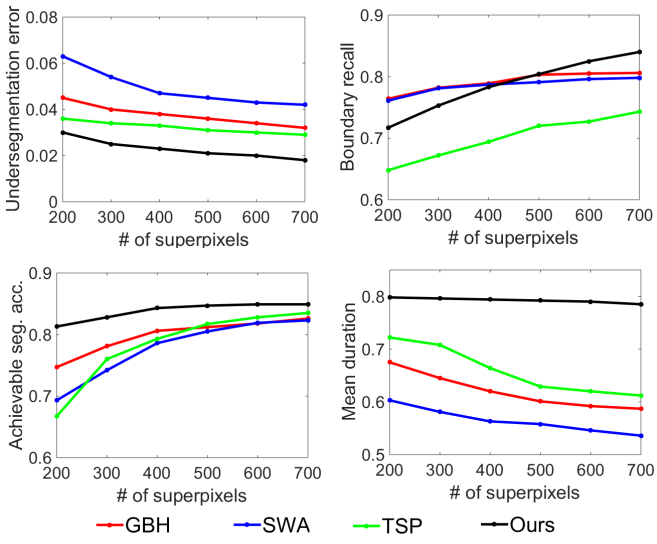

Figure 9: *Evaluation statistics on the SegTrack dataset [TFNR12] and Chen dataset [XC12].*

7. Conclusion

In this paper, we have presented a unimodular Gaussian generative model for the measurement of color homogeneity of superpixel generation. Using this model, the color homogeneity measurement is a MCVT energy, where the anisotropic metric is learned from local pixel color variations. We have demonstrated our model is invariant to affine illumination change, which is a desired property under evolving illumination environments. Also, we quantitatively shown that our framework outperforms other supervoxel methods in image/video analysis.

8. Appendix

This efficient update of covariance matrices requires to keep track of the pixel number and centroid of the clusters. We illustrate the update for $\mathbf{U}_s$, $\mathbf{U}_c$ can be updated in a similar approach.

For a merging operation $(\mathcal{C}_i, \mathcal{C}_j) \to \mathcal{C}_k$:

$$\bar{\mathbf{x}}(\mathcal{C}_k) = \frac{|\mathcal{C}_i|\,\bar{\mathbf{x}}(\mathcal{C}_i) + |\mathcal{C}_j|\,\bar{\mathbf{x}}(\mathcal{C}_j)}{|\mathcal{C}_i| + |\mathcal{C}_j|},$$

$$\begin{aligned}\mathbf{U}_s(\mathcal{C}_k) =& \mathbf{U}_s(\mathcal{C}_i) + \mathbf{U}_s(\mathcal{C}_j) + |\mathcal{C}_i|\,(\bar{\mathbf{x}}(\mathcal{C}_k) - \bar{\mathbf{x}}(\mathcal{C}_i))(\bar{\mathbf{x}}(\mathcal{C}_k) - \bar{\mathbf{x}}(\mathcal{C}_i))^\top \\ &+ |\mathcal{C}_j|\,(\bar{\mathbf{x}}(\mathcal{C}_k) - \bar{\mathbf{x}}(\mathcal{C}_j))(\bar{\mathbf{x}}(\mathcal{C}_k) - \bar{\mathbf{x}}(\mathcal{C}_j))^\top.\end{aligned}$$

For a swapping operation which swaps a boundary pixel $\mathbf{x}$ from $\mathcal{C}_i$ to $\mathcal{C}_j$. Denote $\mathcal{C}_{i'} = \mathcal{C}_i - \mathbf{x}$, and $\mathcal{C}_{j'} = \mathcal{C}_j \cup \mathbf{x}$:

$$\begin{aligned}\mathbf{U}_s(\mathcal{C}_{i'}) =& \mathbf{U}_s(\mathcal{C}_i) - |\mathcal{C}_{i'}|\,(\bar{\mathbf{x}}(\mathcal{C}_{i'}) - \bar{\mathbf{x}}(\mathcal{C}_i))(\bar{\mathbf{x}}(\mathcal{C}_{i'}) - \bar{\mathbf{x}}(\mathcal{C}_i))^\top \\ &- (\bar{\mathbf{x}}(\mathcal{C}_i) - \mathbf{x})(\bar{\mathbf{x}}(\mathcal{C}_i) - \mathbf{x})^\top,\end{aligned}$$

$$\mathbf{U}_s(\mathcal{C}_{j'}) = \mathbf{U}_s(\mathcal{C}_j) + \left|\mathcal{C}_j\right| (\bar{\mathbf{x}}(\mathcal{C}_{j'}) - \bar{\mathbf{x}}(\mathcal{C}_j))(\bar{\mathbf{x}}(\mathcal{C}_{j'}) - \bar{\mathbf{x}}(\mathcal{C}_j))^\top + (\bar{\mathbf{x}}(\mathcal{C}_{j'}) - \mathbf{x})(\bar{\mathbf{x}}(\mathcal{C}_{j'}) - \mathbf{x})^\top.$$

References

[AMFM11] ARBELAEZ P., MAIRE M., FOWLKES C., MALIK J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 5 (2011), 898–916. 7

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell. 34*, 11 (2012), 2274–2282. 1, 2, 3, 5, 7

[CGL*15] CAI Y., GUO X., LIU Y., WANG W., MAO W., ZHONG Z.: Curvature-metric-free surface remeshing via principle component analysis. *CoRR abs/1510.03935* (2015). URL: http://arxiv.org/abs/1510.03935. 3

[CGZM15] CAI Y., GUO X., ZHONG Z., MAO W.: Dynamic meshing for deformable image registration. *Computer-Aided Design 58* (2015), 141 – 150. 5, 6

[CM02] COMANICIU D., MEER P.: Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 603–619. 1

[CWFI13] CHANG J., WEI D., FISHER III J. W.: A video representation using temporal superpixels. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2013). 1, 2, 8

[DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal voronoi tessellations: Applications and algorithms. *SIAM Rev. 41*, 4 (1999), 637–676. 2

[DGJW06] DU Q., GUNZBURGER M., JU L., WANG X.: Centroidal voronoi tessellation algorithms for image compression, segmentation, and multichannel restoration. *J. Math. Imaging Vis. 24*, 2 (2006), 177–194. 2

[DW05] DU Q., WANG D.: Anisotropic centroidal voronoi tessellations and their applications. *SIAM J. Scientific Computing 26*, 3 (2005), 737–761. 3

[FBCM04] FOWLKES C., BELONGIE S., CHUNG F., MALIK J.: Spectral grouping using the nyström method. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 2 (2004), 214–225. 1

[FH75] FUKUNAGA K., HOSTETLER L.: The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory 21*, 1 (1975), 32–40. 1

[FH04] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Efficient graph-based image segmentation. *Int. J. Comput. Vision 59*, 2 (2004), 167–181. 1, 2, 7

[FVT*93] FAUGERAS O., VIÉVILLE T., THERON E., VUILLEMIN J., HOTZ B., ZHANG Z., MOLL L., BERTIN P., MATHIEU H., FUA P., BERRY G., PROY C.: *Real-time correlation-based stereo : algorithm, implementations and applications*. Research Report RR-2013, INRIA, 1993. 4

[GCS12] GALASSO F., CIPOLLA R., SCHIELE B.: Video segmentation with superpixels. In *Asian Conference on Computer Vision* (2012). 1

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH '97, pp. 209–216. 5

[GKHE10] GRUNDMANN M., KWATRA V., HAN M., ESSA I.: Efficient hierarchical graph-based video segmentation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 2141–2148. 1, 2

[Hir07] HIRSCHMÃIJLLER H.: Evaluation of cost functions for stereo matching. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2007). 6, 7

[HLL11] HEO Y. S., LEE K. M., LEE S. U.: Robust stereo matching using adaptive normalized cross-correlation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 4 (2011), 807–822. 4

[JDS08] JEGOU H., DOUZE M., SCHMID C.: Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision: Part I* (2008), ECCV '08, pp. 304–317. 7

[KHKS14] KIM S., HAM B., KIM B., SOHN K.: Mahalanobis distance cross-correlation for illumination-invariant stereo matching. *IEEE Transactions on Circuits and Systems for Video Technology 24*, 11 (2014), 1844–1859. 4

[Llo82] LLOYD S. P.: Least squares quantization in PCM. *IEEE Transactions on Information Theory 28* (1982), 129–137. 5

[LSK*09] LEVINSHTEIN A., STERE A., KUTULAKOS K. N., FLEET D. J., DICKINSON S. J., SIDDIQI K.: Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell. 31*, 12 (2009), 2290–2297. 1, 7

[LSTS04] LI Y., SUN J., TANG C.-K., SHUM H.-Y.: Lazy snapping. *ACM Trans. Graph. 23*, 3 (2004), 303–308. 1

[LWL*09] LIU Y., WANG W., LÉVY B., SUN F., YAN D.-M., LU L., YANG C.: On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph. 28*, 4 (2009), 101:1–101:17. 5

[MK10] MIČUŠÍK B., KOŠECKÁ J.: Multi-view superpixel stereo in urban environments. *Int. J. Comput. Vision 89*, 1 (2010), 106–119. 1

[MPW*08] MOORE A. P., PRINCE S. J. D., WARRELL J., MOHAMMED U., JONES G.: Superpixel lattices. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2008), pp. 1–8. 1

[RA15] RICHTER R., ALEXA M.: Mahalanobis centroidal Voronoi tessellations. *Computers & Graphics 46*, 0 (2015), 48–54. 2, 3, 4

[SBB00] SHARON E., BRANDT A., BASRI R.: Fast multiscale image segmentation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2000), vol. 1, pp. 70–77. 1

[SGS*06] SHARON E., GALUN M., SHARON D., BASRI R., BRANDT A.: Hierarchy and adaptivity in segmenting visual scenes. *Nature 442*, 7104 (2006), 810–813. 1

[SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 8 (2000), 888–905. 1

[SMT13] SONG X., MUSELET D., TRÃL'MEAU A.: Affine transforms between image space and color space for invariant local descriptors. *Pattern Recognition 46*, 8 (2013), 2376 – 2389. 4

[TFNR12] TSAI D., FLAGG M., NAKAZAWA A., REHG J. M.: Motion coherent tracking using multi-label mrf optimization. *Int. J. Comput. Vision 100*, 2 (2012), 190–202. 8

[Wei96] WEICKERT J.: Anisotropic diffusion in image processing, 1996. 2

[WW12] WANG J., WANG X.: Vcells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 6 (2012), 1241–1247. 7

[WZG*13] WANG P., ZENG G., GAN R., WANG J., ZHA H.: Structure-sensitive superpixels via geodesic distance. *International Journal of Computer Vision 103*, 1 (2013), 1–21. 1, 2, 3

[XC12] XU C., CORSO J. J.: Evaluation of super-voxel methods for early video processing. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2012), pp. 1202–1209. 1, 2, 7, 8

[YJLH16] YONG-JIN LIU CHENGCHI YU M. Y., HE Y.: Manifold slic: A fast method to compute content-sensitive superpixels. *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (2016), To appear. 1, 2, 7

[YLY14] YANG F., LU H., YANG M.: Robust superpixel tracking. *IEEE Trans. Image Processing 23*, 4 (2014), 1639–1651. 1