Computer Vision and Image Understanding 113 (2009) 446–458

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Image description with features that summarize

J.J. Corso *, G.D. Hager

Computational Interaction and Robotics Lab, Department of Computer Science, The Johns Hopkins University, Baltimore, MD 21218, USA

ARTICLE INFO

Article history:
Received 6 February 2007
Accepted 19 November 2008
Available online 25 December 2008

Keywords:
Image matching
Segmentation
Interest point operator
Feature space
Feature detector

ABSTRACT

We present a new method for describing images for the purposes of matching and registration. We take the point of view that large, *coherent* regions in the image provide a concise and stable basis for image description. We develop a new algorithm for feature detection that operates on several *projections* (feature spaces) of the image using kernel-based optimization techniques to locate local extrema of a continuous scale-space of image regions. Descriptors of these image regions and their relative geometry then form the basis of an image description. The emphasis of the work is on features that summarize image content and are highly robust to viewpoint changes and occlusion yet remain discriminative for matching and registration.

We present experimental results of these methods applied to the problem of image retrieval. We find that our method performs comparably to two published techniques: Blobworld and SIFT features. However, compared to these techniques two significant advantages of our method are its (1) stability under large changes in the images and (2) its representational efficiency.

1. Introduction

In this paper, we consider the problem of matching (or registering) differing views of a scene to each other. This, problem, which has received an immense amount of attention over the last decade, is currently solved using two different approaches.

One set of approaches, pioneered by Schmid and Mohr [1] and extended in many recent papers (Section 2), makes use of local region descriptors for indexing and matching. The general idea of such approaches is to locate regions of high information content using an interest operator, and to then create indices for matching. The key to good performance is to create interest operators and match indices that are insensitive to geometric and photometric image distortions. The advantage of the approach is generally the robustness of matching to occlusion, changes in lighting, and moderate changes of viewpoint. The disadvantages are the need to identify such local image regions, and (typically) the use of only grayscale image projections. In particular, large areas of the image are potentially discarded as "untextured" and therefore unusable by the method. In addition, since the local methods emphasize feature discriminability, they generally have a high storage cost. Mikolajczyk and Schmid [2] evaluated the performance of several local image descriptors. Their evaluation tested the descriptors' stability to rotation, scaling, affine transformations, and illumination changes. The study showed that SIFT [3] features performed the best over all conditions.

Another set of approaches, exemplified by Malik et al. [4,5] instead represent images through segmentation. This approach is particularly appealing for image retrieval problems where the goal is to find similar, rather than exactly matching, images. The advantages are that large areas of the image tend to be stable across large changes in viewpoint and can be matched in a spatially approximate manner, and the associated representation tends to have a smaller storage cost since the goal is to summarize image content. The disadvantage is that it is necessary to have an efficient yet stable segmentation process, and to find image cues that are themselves stable over variations in pose and lighting.

In our work, we consider a "middle ground." Namely, our goal is to create interest operators that focus on homogeneous regions, and local image descriptors for these regions. Intuitively, we propose to locate a sparse set of large, homogeneous regions (features) that can be simply characterized to summarize the image content. To this end, we perform a sparse image segmentation, and then index images based on the results of that segmentation. The segmentation is performed in parallel on several scalar image projections (feature spaces) using kernel-based optimization methods. The optimization evaluates both the size (large regions tend to have high stability across widely disparate views) and the coherency (e.g., similar color, texture, depth, or image gradient) of region content. Once a region is located, its description is composed of simple kernel-weighted statistics of the coherent content. This description is concise, and it is stable under drastic changes in viewpoint, and it is insensitive to photometric changes provided the initial image

* Corresponding author. Present address: Computer Science and Engineering, State University of New York at Buffalo, 201 Bell Hall, Buffalo, NY 14260, USA.
E-mail addresses: jcorso@cse.buffalo.edu (J.J. Corso), hager@cs.jhu.edu (G.D. Hager).

1077-3142/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2008.11.009

projections are. We claim that using such sparse regions provide a plausible trade-off between the discriminating power of local techniques and the summarizing efficiency of global methods. Finally, since we compute multiple image regions, images can be geometrically registered in a manner similar to interest point-based registration.

In principle, our method is most similar to Greenspan et al. [6] and Schaffalitzky and Zisserman [7]. [6] use a mixture distribution as the underlying continuous representation of the image. As we will discuss in Section 3.1, we define a similar mixture distribution as our underlying representation. However, [6] estimate the mixture distribution jointly for the whole image by using an expectation maximization algorithm with a minimum description length (MDL) term. Their model is derived on a fixed feature space (space and color). Matching is done via a global KL distance score. While our ultimate representation is similar, our algorithms are philosophically different: [6] seek a global representation and do joint optimization, while we seek a more local representation optimizing each feature point independently (using kernel techniques). Furthermore, there is no natural way to tune their algorithm to yield models with as many computed features as our models because they incorporate an MDL term into the objective function. We match part-by-part while [6] compute a global distance score. Finally, we define a general feature space that a system engineer can tune, which does not require any change on the model estimation; [6] fixes the feature space.

[7] use the texton segmentation [5] and create a texture-region descriptor that is invariant to affine geometric and photometric transformations. They robustly estimate the epipolar geometry of a wide baseline system by matching these regions. While emphasis on scene-retrieval and registration based on regions as opposed to points is similar to their work, we differ in the region detection and description. The remainder of this paper first presents a literature survey of local and global methods, then details our kernel-based segmentation methods and last, provides comparative experimental results supporting the claim that the proposed *features that summarize* provide a plausible balance between local features and global segmentation.

2. Related work in image modeling

In this section, we discuss two of the basic approaches at modeling images: local, pixel-level modeling and global, entire-image modeling. We also briefly discuss the important issue of scale selection.

2.1. Local methods

Local, pixel-level methods focus on finding salient points in images. As discussed in Section 1, the general idea of such approaches is to locate regions of high texture content using an interest operator, and to then create indices for matching. The first so-called interest operator was proposed by Moravec [8], which detects points with high-contrast neighborhoods and is rotationally invariant. Another rotationally invariant interest operator is the Harris corner detector [9], which performs a local gradient eigenanalysis to select points with neighborhoods whose gradient is varying in both image dimensions. The Harris detector has a high repeatability rate [10], which is important since the interest points will be matched across images.

The pioneering work of Schmid and Mohr [1] emphasizes the invariance properties of both detection and description of the interest points and the local regions surrounding them. They use local, differential grayvalue invariants in a multiscale representation at a number of interest points (Harris corners) in the image for description. The invariants are the local differential *jets* from Koenderink and van Doorn [11]. Their representation is robust to similarity transforms and partial visibility. Additionally, they include information from multiple scales yielding a scale-invariant (up to the scale quantization) representation. To match, they propose a fast multidimensional hash-table voting algorithm that is robust to mismatches and outliers.

Schmid and Mohr's work gave rise to numerous related techniques, which we summarize next. Lowe [12,3] proposed a scale-invariant feature transform (SIFT). The interest point, or *key*, locations are identified with a staged filtering approach that searches through a discrete scale-space [13] for minima and maxima of a difference-of-Gaussian function. For representation, the image neighborhood around each key location is assigned a canonical orientation in accordance with the local image gradients. Then, the feature vector is constructed by *orientation planes*. A local image region can be separated into a set of orientation planes each consisting of only the gradients corresponding to that orientation. The keys are invariant to image translation, scaling and rotation, and partially invariant to illumination changes. Since the keys have a relatively high storage cost, Ke and Sukthankar [14] proposed an extension of Lowe's SIFT [12] method, PCA-SIFT. While SIFT patch descriptors are constructed by smoothed orientation histograms, the PCA-SIFT patch descriptors is based on the projection of the patch gradient maps into a low-dimensional eigenspace.

In recent years, many researchers have proposed affine-invariant interest points and features. Lazebnik et al. [15] detect interest points in the images by local maxima of the Laplacian in scale-space. Then, for each maxima, the local image region, at the appropriate scale, is then normalized based on the second-moment matrix resulting in affine-invariant patches. The normalized patches are represented using intensity-domain spin images, a two-dimensional histogram with axes of brightness and distance from the patch center.

Tuytelaars and van Gool [16] proposed detection of regions by finding intensity-based local extrema, constructing an irregularly shaped blob, and then fitting an affine invariant ellipse (equivalent to the irregularly shaped blob up to the second-order moments). The regions are described by *Generalized Color Moments*, which implicitly characterize the shape, intensity, and color distribution of the region pattern in a uniform manner. They couple the image description with Harris corners and apply it to the wide-baseline stereo problem. Related work in wide-baseline stereo using interest region based techniques include the Maximally Stable Extremal Regions by Matas et al. [17,18] and the scale-invariant, normalized affine-corner pairs of Tell and Carlsson [19].

The last set of techniques we discuss uses a maximization of the Shannon entropy measure [20] in the image signal to detect and characterize salient points. The idea is to define saliency in terms of local signal complexity. Gilles [21] uses salient image patches to register aerial images; he uses the entropy of local image patch histograms to characterize saliency. However, Gilles fixed a global-scale for the size of the patches per image. For the case of aerial satellite imagery where an affine geometry assumption is plausible, the fixed scale is acceptable, but in the general case, it is not. Kadir and Brady's scale-saliency technique [22] extended Gilles's saliency detector to incorporate patch scale. They search for clusters of high-entropy in scale-space and use them as the interest points. Hare and Lewis [23] use the scale-saliency interest points for image matching.

Fraundorfer and Bischof [24] argue that one should fuse the characterization with the detection because if they are separate, then it is possible the detector may find regions whose description will not be salient. [24] use geometric features (specifically, Harris corners [9]) to describe the local image patch. Hence, they improve the robustness of the patch description to photogrammetric and

geometric changes [25]. They incorporate Kadir and Brady's [22] scale-selection technique and find that their method significantly reduces the number of false matches.

Mikolajczyk and Schmid [2] evaluated the performance of several local image descriptors. Their evaluation tested the descriptors' stability to rotation, scaling, affine transformations, and illumination changes. The study showed that SIFT [3] features performed the best over all conditions. Thus, we use the SIFT method in our comparative analysis (Section 7) to represent the class of local methods.

2.2. Global methods

The global methods attempt to capture the most important content of the image as a whole. Such methods, in general, attempt to form a low order summary of the image. One set of methods, which we do not cover in this paper, use global histograms to represent the image (e.g., [26]). We discuss those methods that use image segmentation as a representation. Since the segmentation literature is broad, we further restrict it to the subset the emphasizes an image retrieval application because of the direct relation to this work. This approach is exemplified by Malik et al. [4,5]. They attempt to group pixels that roughly correspond to objects therefore allowing image-to-image matching at the object level. In the modeling, they incorporate color, texture, and position features into a Gaussian mixture model (GMM). They use the Expectation-Maximization [27] with the Minimum Description Length principle [28,29] for GMM estimation and model selection. Similarly, Greenspan et al. [30,6] use GMM modeling in a 5D color and spatial feature space to model images and video. For image-to-image comparison, they directly compare the GMM using a novel approximation of the Kullback–Leibler distance Mo and Wilson [31] extend these ideas in a multiresolution GMM.

Ruiz et al. [32] generalize the notion of the scale-space blob [13] to include color information. The scale-space blobs are analogous to a full image segmentation. They use the automatic scale-selection principle based on extrema of the normalized Laplacian [33]. Neural networks are used to learn the image categories.

Schaffalitzky and Zisserman [7] describe a texture-region descriptor that is invariant to affine geometric and photometric transformations and insensitive to the shape of the texture region. In the affine-normalized region, they compute a rotationally and scale invariant description using a statistical approach that creates a histogram of the dominant gradient at each pixel (for each scale). In their approach, *detection* of regions is solved by standard texture-based image segmentation [5].

2.3. The scale issue

Scale is a crucial parameter in the analysis of objects in images. In our case, there are two essential notions of scale: the *integration* scale of the image content (e.g., texture or edges), and the scale of an associated spatial kernel function used to summarize image content. In both cases, there is no universally accepted method for choosing an optimal scale. In our work, we focus primarily on determining the correct scale of a spatial kernel for summarizing image content.

Lindeberg proposed a set of scale-selection principles [33] for feature detection and image matching, and a technique [13] for building a gray-level blob and scale-space blob representation of an image. Comaniciu et al. [34] proposed the variable bandwidth mean shift to solve this problem (in the context of kernel-based density estimation [35]). Collins [36] applied Lindeberg's general scale-selection principles [33] to extend the kernel-based mean shift tracking to refine the scale of the object being tracked. Okada et al. [37] presented a method for the creation of an anisotropic, Gaussian scale-space by extending Lindeberg's [33] isotropic scale-space methods.

3. Image modeling

A *coherent* region in an image is a connected set of (relatively) homogeneous pixels. Coherency is indifferent to the character of the homogeneity. For example, the image of a plaid shirt with a colorful, checkered pattern is considered coherent. This notion of region coherency is the basis for image segmentation and has been studied in various methodologies for multiple decades: e.g., the piecewise constant Potts model [38] and the piecewise smooth Mumford and Shah model [39]. We will use the coherency idea in multiple image feature spaces in Section 4; currently, we consider single, scalar (e.g., grayscale) images (discrete signals).

3.1. The model

We denote the image $\mathbf{I} \doteq \{\mathscr{I}, I\}$ where $\mathscr{I}$ is a finite set of n pixel locations (points in $\mathbb{R}^2$), I is the map $\mathscr{I} \to \mathscr{X}$ (here, $\mathscr{X}$ is some arbitrary value space). We model the appearance of a region as a constant value α with additive i.i.d. noise assumed to be zero-mean Gaussian $\mathscr{N}(0, \psi^2)$. The region is assumed to be connected; spatially, we describe a region with an anisotropic Gaussian kernel:

$$K(i, r) = \frac{1}{2\pi|\mathbf{\Psi}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(i-\boldsymbol{\mu})^{\mathsf{T}}\mathbf{\Psi}^{-1}(i-\boldsymbol{\mu})\right), \tag{1}$$

where $i \in \mathscr{I}$ and $r \doteq \{\boldsymbol{\mu} \in \mathbb{R}^2, \mathbf{\Psi} \in \mathrm{GL}(2), \mathbf{\Psi} = \mathbf{\Psi}^{\mathsf{T}}\}$ fully describe the anisotropic Gaussian. We restrict the discussion to normalized kernels: $\sum_{i \in \mathscr{I}} K(i, r) = 1$. We note that this restriction is only an approximation since we are working with sampled data (see Appendix A). The spatial parameters r and the appearance parameter α fully describe the region.

We use the Gaussian to spatially describe the region because it has relatively few parameters, is sufficient to estimate regions of arbitrary size and orientation, and does not require the precise estimation of object boundaries. Pixels are weighted based on their proximity to the spatial mean of the Gaussian kernel.

The complete model for describing an image is a set of region pairs $\{(r_i, \alpha_i) : i = 1, \ldots, m\}$. One can conceptualize this model as a mixture distribution [40] in a joint feature-spatial space with Gaussian components and uniform mixing weights.

3.2. Model estimation

The free parameters of the model are the number of regions and both the spatial and appearance parameters for each region. The most popular [40] technique to estimate a mixture model is the Expectation-Maximization method [27]. While it has guaranteed convergence, it is very sensitive to initialization and requires the number of components as input. In our formulation, the number of components corresponds to the number of coherent regions, which is a data dependent variable. As in [4,6], one can apply the minimum description length principle [28,29]. However, there remains a problem due to the regular sampling of the pixels, which violates the assumption that the location samples are normally distributed.

Instead of taking this joint approach to estimating the model, we propose a local approach that defines an interest operator for coherent regions. We assume the number of regions is unknown and derive an objective function on the five parameters of the spatial anisotropic Gaussian. Given a set of region parameters r, we estimate the appearance parameter α by minimizing the following error term

$$\alpha^* = \arg\min_{\alpha} \sum_{i \in \mathscr{I}} K(i, r)(I(i) - \alpha)^2, \tag{2}$$

which follows directly from the assumed Gaussian noise distribution and the use of a spatial weighting kernel. The minimum is the kernel-weighted estimate of the appearance parameter:

$$\alpha^* = \sum K(i,r)I(i). \tag{3}$$

Plugging this kernel-weighted mean back into Eq. (2) and switching the arguments to the region parameters r, we arrive at the kernel-weighted variance, which is used to estimate the r parameters.

$$\begin{aligned} r^* &= \arg\min_r \sum K(i,r)(I(i)-\alpha)^2 \\ &= \arg\min_r \sum K(i,r)I(i)^2 - \left[\sum K(i,r)I(i)\right]^2. \end{aligned} \tag{4}$$

However, this error function is unstable and has zero variance as its minimum, which is *degenerate* (the function would be minimized when the region is only sampling the value at a single pixel). To regularize the error function, we include a second, additive term. While many regularizing choices are possible, we choose one that has a physical meaning. We minimize the squared distance between the kernel and the image. Since we consider, normalized kernels, we include a scale factor β for the kernel.

$$\min \sum [\beta K(i,r) - I(i)]^2. \tag{5}$$

We include a normalizing factor $\frac{1}{n}$ and a multiplier γ that is set by hand to weight the two terms of the function. We combine (4) and (5) to yield the final objective function:

$$\arg\min_{\beta,\boldsymbol{\mu},\boldsymbol{\Psi}} \sum K(i,r)I(i)^2 - \alpha^2 + \frac{\gamma}{n}\sum [\beta K(i,r) - I(i)]^2. \tag{6}$$

By taking a binomial expansion and discarding both constant and higher order terms, we use the following function to approximate (6) [41]:

$$\arg\min_{\boldsymbol{\mu},\boldsymbol{\Psi}} \frac{\sum K(i,r)I(i)^2}{\alpha^2} + \frac{\tau}{|\boldsymbol{\Psi}|^{\frac{1}{2}}}, \tag{7}$$

where τ is a weighting factor between the two terms. We show the derivation and proof in Appendix A. We note the appealing form of this function. It is the sum of a homogeneity term and a scale term, which are precisely the two characteristics we wish to focus on in the coherent regions. Since the kernels are defined continuously, standard optimization methods can be used to minimize (7) such as gradient descent and newton minimization.

3.3. Initialization

We initialize the regions using conventional blob-finding techniques. Marr and Hildreth [42] first proposed the use of the Laplacian of a Gaussian (LoG) for distinguishing homogeneous regions from the drastic changes in intensity that separate them. More recently, Lowe [3], among others [15], used a Difference of a Gaussian (DoG) to approximate the LoG filter. They construct a dense, discrete scale-space of DoG responses and then perform an explicit search for stable points (local extrema in space and scale).

To detect seed points, we likewise create a coarse, discrete scale-space of isotropic DoG responses by sampling a few (in our experiments, just 2) large scales. This coarse sampling is sufficient for seed detection because we later refine each candidate seed and localize it in both space and scale. Similar to Lowe, we look for local extrema in the DoG response to detect seeds. However, since we are coarsely sampling scale-space, we analyze each 2D DoG-response separately (Lowe searches for extrema in 3D scale-space). Our search will result in many spurious seed extrema, which will converge to the nearest true extrema in the optimization of (7).

We define a seed with three parameters: $\boldsymbol{\mu}$ is set to the spatial location of the extrema point, and the $\boldsymbol{\Psi}$ is set to the product of the 2×2 identity and one-third of the scale of the LoG filter. Intuitively, this one-third scale factor shrinks the kernel to the homogeneous region at the filter's center. In contrast, Lowe scales the region by a factor of 1.5 because the SIFT keys function best in regions of high contrast (the region including its surrounding areas, for example). Fig. 1 shows a comparison of our scaling and Lowe's scaling with respect to the LoG function.

3.4. Merging

Different seed points may converge to the same minimum of the objective function (7), and since the optimization is independent for each seed point, we must account for this issue in a post-processing step. It is possible to do a more sophisticated initialization procedure that would reduce the need for a merging process. Essentially, there is a trade-off between the complexity of the initialization and the necessity of a merging post-process. Since we have a continuous objective function that can be minimized with efficient techniques, we are conservative and choose a simpler initialization that will result in multiple seeds converging to the same minimum.

Let $\mathscr{R}$ denote the set of active regions in an image. For a region $R \in \mathscr{R}$, denote the parameters by $\boldsymbol{\theta}(R) \doteq \{\boldsymbol{\mu}, \boldsymbol{\Psi}\}$. Since the regions are described by anisotropic Gaussian functions, we use the Kullback–Leibler (KL) divergence function [20] and sum it in both directions to make it symmetric. Since we want to find regions that have converged to the same local minimum, we set a threshold τ let two regions be *equivalent* if their KL distance is less than τ. Such a threshold must be used due to numerics and discretization in the floating-point representation. Define an empty set of merged regions $\hat{\mathscr{R}} = \emptyset$, and merge with the following algorithm:

(1) For each region $R \in \mathscr{R}$.
(2) For each region $S \in \mathscr{R} \setminus R$
(3) If $d(R,S) < \tau$, remove S from $\mathscr{R}$
(4) Add R to $\hat{\mathscr{R}}$.

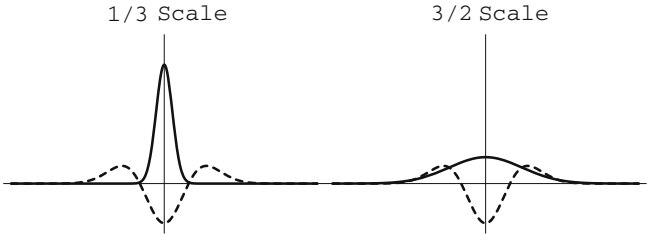

Fig. 1. A comparison of the region scaling between our homogeneous regions (one-third) and Lowe's SIFT keys (1.5). The LoG kernel is shown as a dotted line with the region size as a solid line.

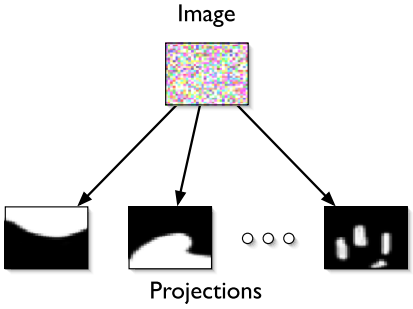

Fig. 2. Explanation of data-flow in image dimensionality reduction.

Fig. 3. Example pixel projections. (Left) Original image. (Middle) RGB linear combination with coefficients $(-1,1,-1)$. (Right) RGB pixel likelihood with color $(0,1,0)$.

From our observations, we have found the number of regions was significantly reduced (about 25% on average) after the merging procedure. The procedure is insensitive to τ, which we set ($\tau = 2.0$) based on empirical experiments measuring stability of region merging. Once two regions are merged, we set their parameters to the arithmetical average.

4. Multiple feature spaces

Images are complex entities; they are the result of numerous physical and stochastic processes and live in a very high dimensional space. To make image analysis tractable, we project the images into a lower dimensional space. Each dimension in the projected space captures a single image character like red-ness, or stripy-ness. This idea is related to filtering techniques like using a bank of Gabor filters as the basis (e.g. [43]), or dimensionality reduction with principle components analysis (e.g. [44]). However, we differ in that there is an underlying assumption that the image has been constructed from a set of unknown scalar image processes. The goal in our approach is to define a set of projections that can approximate these unknown underlying processes.

Essentially, each projection defines a new *feature space* (see Fig. 2). The intuition is that various projection functions will map a region of consistent image content to a uniform image patch in the scalar field: e.g., there is some texture and/or color projection function such that an image of a plaid shirt will be mapped to a relatively uniform scalar field. Thus, by choosing appropriate scalar projections, we can capture coherent image content of varying character. To that end, define a function $S : \mathscr{X} \times \mathscr{I} \mapsto \mathbb{R} \times \mathscr{I}$ that *projects* the d-dimensional image $\mathbf{I}$ to a one-dimensional scalar field $\mathbf{J}$. The scalar image $\mathbf{J}$ is indexed by the same pixel locations $\mathscr{I}$ and is thus comprised of $\{\mathscr{I}, J\}$, where $J : \mathscr{I} \mapsto \mathbb{R}$. We will simply write $J(i)$ instead of $S(\mathbf{I})(i)$ in the following examples.

There are two classes of projections: those that operate independently on single pixels and those that operate over neighborhoods of pixels. The methodology we propose is general and the construction of these projections is application dependent. We do not restrict the projections: they may be non-linear, and they may be dependent.

4.1. Pixel projections

A pixel projection is one that operates individually on the pixels without considering any neighborhood information. While limited in utility, they do not affect any invariance properties of the detection.

4.1.1. Linear combinations of pixel color

A simple linear combination of image bands can define a useful feature space. Given three coefficients $\{c_r, c_g, c_b\}$ on the pixel color components, the definition is

$$\mathbf{J}(i) = c_r \mathbf{I}_r(i) + c_g \mathbf{I}_g(i) + c_b \mathbf{I}_b(i), \quad \forall i \in \mathscr{I}. \tag{8}$$

Fig. 3 (middle) shows an example. Such a discrete set of linear combinations is used by Collins and Liu [45] in a tracking framework. Each vector of coefficients creates a feature space, and they propose a Fisher discriminant-like ratio to choose the best feature space for the current image frame.

4.1.2. Pixel color likelihood

A second pixel projection models a feature space as a Gaussian process in color-space. Then, the projection computes the likelihood on each pixel. Given a color, $\mathbf{c}$ and an estimated covariance $\mathbf{\Sigma}$, the likelihood function is written:

$$\mathbf{J}(i) = \exp\left(-\frac{1}{2}(\mathbf{I}(i) - \mathbf{c})^{\mathsf{T}} \mathbf{\Sigma}^{-1} (\mathbf{I}(i) - \mathbf{c})\right), \quad \forall i \in \mathscr{I}. \tag{9}$$

Fig. 3 (right) gives an example of the likelihood function.

4.2. Neighborhood projections

Neighborhood projections can be more powerful than the pixel projections because they incorporate information from multiple pixels in a single calculation. However, the neighborhood projections affect the invariance properties of the detection. For example, for the detection to be scale invariant, we would need to know the per-pixel *integration* scale (i.e. the size of the local neighborhood needed to completely model the local image texture). While some heuristic methods have been presented to estimate this local scale [4], its calculation is error-prone, especially near object boundaries.

In addition to the two neighborhood projections we discuss below, various linear filters can be used as projection functions as well. These include gradient operator, Gabor [46] functions, and template-matching kernels.

4.2.1. Grayscale variance

The neighborhood variance is a simple texture operator. Let $N(i) \subset \mathscr{I}$ define the set of neighborhood pixels for $i \in \mathscr{I}$ with cardinality n. For pixel i, the variance is

$$J(i) = \frac{1}{n} \sum_{j \in N(i)} \left(I(j) - \frac{1}{n} \sum_{j \in N(i)} I(j) \right)^2. \tag{10}$$

4.2.2. Local orientation coherency

A second texture projection measures the local orientation coherency, or the *stripy-ness*, of the neigborhood. Jahne [47, p. 357] suggests a ratio of a linear combination of the eigenvalues of the structure tensor. Denote the local image gradients of $\mathbf{I}$ in the x and y direction as $\mathbf{I}_x$ and $\mathbf{I}_y$, respectively. Then, for pixel i and its neighborhood N, the structure tensor T is

$$T(i) = \begin{bmatrix} \sum_{N(i)} I_x^2 & \sum_{N(i)} I_x I_y \\ \sum_{N(i)} I_x I_y & \sum_{N(i)} I_y^2 \end{bmatrix}. \tag{11}$$

Let $\lambda_1 \geqslant \lambda_2$ be the eigenvalues of $T(i)$. Then, if there is a ideal local orientation, one eigenvalue is zero, $\lambda_1 > \lambda_2 = 0$. Considering image noise, the ideal case will never happen. For dominant local orientation $\lambda_1 \gg \lambda_2$. Otherwise, if there is isotropic local orientation (including the case of little gradient information), $\lambda_1 \approx \lambda_2$. Thus, this suggests using the following ratio to analyze the presence of dominant local gradient:

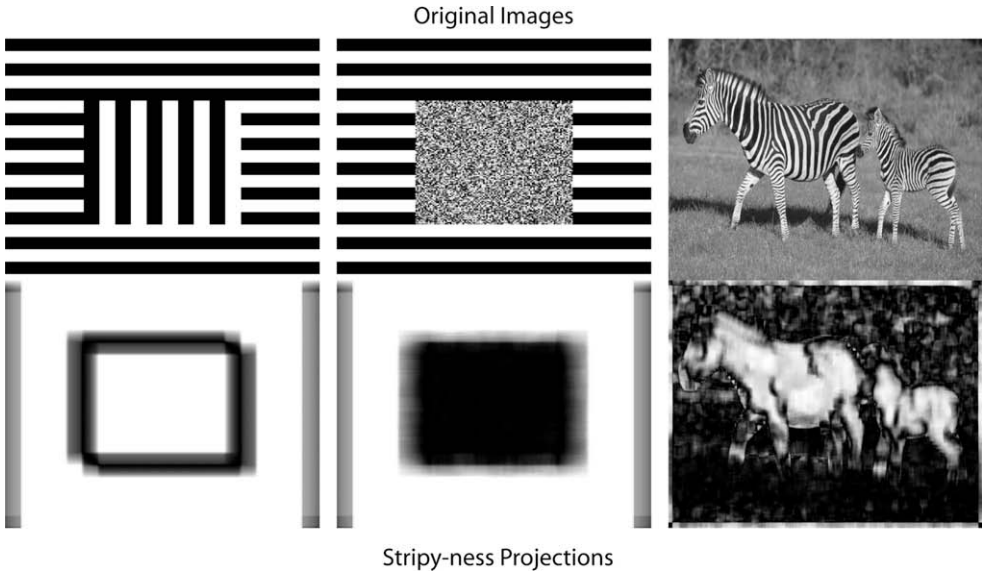

Fig. 4. Examples of the stripy-ness (local orientation coherency) projection. The grayscale images are on top with the corresponding projections below. In the projections, white means more stripy.

$$J(i) = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2. \quad (12)$$

For the case of dominant local orientation, then this ratio is near 1, and conversely, for the case of no dominant local orientation, this ratio tends to 0. Care must be taken in the implementation to avoid division-by-zero; in our implementation, we threshold on very low gradients. We give three examples of this stripy-ness projection in Fig. 4.

5. The complete detection algorithm

In this section, we summarize the complete algorithm for extracting coherent regions. The local minima of a continuous scale-space define representative coherent regions in the image description. For a given input image **I**, define a set of scalar projections $\mathscr{B} = \{S_1, \ldots, S_b\}$. For each projection $p \in \mathscr{B}$, define an initial, empty set of regions $\mathscr{R}_p$ and carry out the following steps:

(1) Detect seeds (Section 3.3).
(2) Independently, minimize the function in Eq. (7) to refine each seed.
(3) Add convergent regions to $\mathscr{R}_p$.
(4) Merge $\mathscr{R}_p$ (Section 3.4).

After computing the b region sets, we compute region descriptions (Section 6). In Fig. 5, we show examples of the algorithm running on four different image projections.

6. Region description

In this section we discuss a few potential approaches to region description; the description approaches are compared in Section 7.3. Recall that at this step in the algorithm, we have b sets of regions $\mathscr{R}_1 \ldots \mathscr{R}_b$, (one for each feature space).

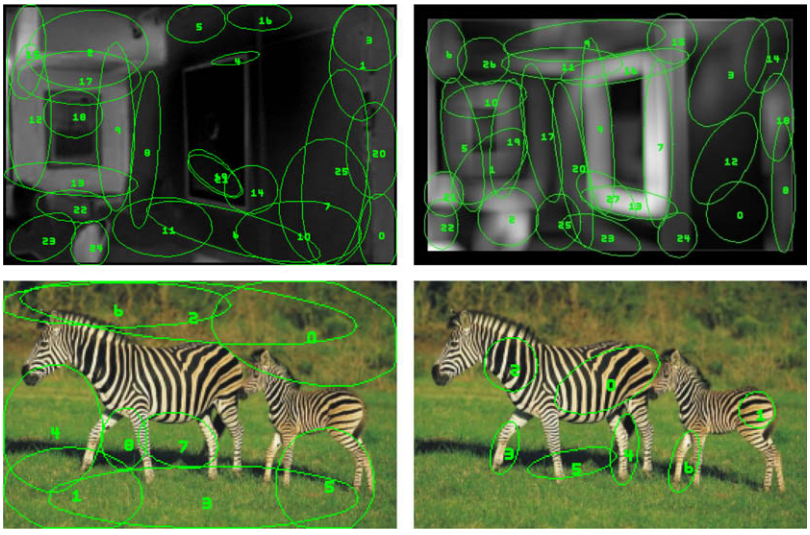

Fig. 5. Examples of the algorithm running on four different image projections. Top-left is the green color projection introduced in Fig. 3 (right). Top-right is a neighborhood variance projection on the same image from Fig. 3(left). Bottom-left is a "grassy" color projection. Bottom-right is the stripy-ness projection. (For interpretation of color mentioned in this figure the reader is referred to the web version of the article.)

6.1. Single appearance with cooccurrence

The first description is the appearance model we have already discussed. Here, we assume that the character is a constant value in one of the projected images with zero-mean, normally distributed noise. We can either fix or estimate the variance. Because we have not restricted the development to independent projections, we must explicitly model the cooccurence information relationship between different projections. In this context, cooccurence simply means that the same region (spatial) has been detected in two or more different projections.

We augment each region with a parameter-list indicating from which projection it was detected. Denote the parameter-list of projection(s) for a region R by $L(R)$. Then, we define a new region set that is the union of all detected regions detected $\mathscr{R} = \bigcup_{p=1}^{b} \mathscr{R}_p$. To evaluate the cooccurence information, we enumerate through each pair of regions in $\mathscr{R}$. For each pair of regions (R, S), we evaluate their spatial proximity with the KL distance. If $d(R, S) < \tau$, the regions are said to cooccur. In this case, we define a new region T that inherits its parameters from R and S: the spatial parameters set to those of R (which are equivalent since $d(R, S) < \tau$) and the projection list is combined by taking the union $L(T) = L(R) \bigcup L(S)$. We remove R and S from $\mathscr{R}$ and add T to $\mathscr{R}$.

After checking each region pair in $\mathscr{R}$, we compute the appearance description for the remaining regions. For each region $R \in \mathscr{R}$, describe it by computing the kernel-weighted mean (3) under each projection in its list $L(R)$.

6.2. Appearance in all projections

The first method for describing the appearance of the regions creates a summarization of the image based on the exact projections used in the segmentation. However, such a representation may not be discriminative enough for certain problems, like image-to-image matching. A slightly more discriminative approach is to sample the appearance properties in all projections regardless of a region's originating projection.

Take the union of all regions detected $\mathscr{R} = \bigcup_{p=1}^{b} \mathscr{R}_p$. Then, for each region in $\mathscr{R}$ sample the kernel-weighted mean over all projections $\mathscr{B}$ creating a b dimensional vector. This is the representation we have used in [41]. Although this description yields good results (Section 7), it is invalid to assume a single, homogeneous value in every projection. Therefore, one must also measure the kernel-weighted variance in each projection, which is shown to improve matching results (Section 7.3).

6.3. Properties

The region description is implicitly invariant to rotation and translation in the image because it is simply a set of kernel-weighted statistics. However, as the detection is sensitive to scale if a neighborhood projection is used, the description is also sensitive to scale. Likewise, if the scalar projection is designed to extract *vertical* texture (*y-gradient* in the image), then the region's description under this projection is no longer rotationally invariant or robust to affine distortion. A rotated image will yield a completely different region description under this projection. However, we note that one can enforce some degree of invariance (e.g., affine [15]) by explicitly incorporating it based on the spatial parameters of the region before computing the description. We do not explore this idea in this paper.

Given each of the description methods described above, the maximum number of appearance parameters per regions is $2b$ for b projections. It is clear that the image description is concise (linear). Thus, the storage requirement for our technique will not prohibit its scaling to large or very large databases.

7. Experiments

We use an image retrieval experimental paradigm. Thus, for a given database of images, we apply our coherent region extraction to each image independently. The image descriptions are then stored in a database and a querying protocol is established. To perform retrieval, for each image in the dataset, we query the database, and a sorted list of *matching* images is returned with the best match first.

For the experiments, we use a dataset of 48 images taken of an indoor scene from widely varying viewpoints and with drastic photometric variability (a subset of the dataset is shown in Fig. 6). We hand-labeled the datasets; two images are said to be *matching* if there is any area of overlap between them (hence the comparatively small data set size).

We use the standard precision–recall graphs to present the matching results. The precision is defined as the fraction of true-positive matches from the total number retrieved and the recall is the fraction of matching images that are retrieved from the total number of possible matches in the database. Thus, in the ideal case, the precision–recall graph is a horizontal line at 100% precision for all recall rates.

Denote the three bands of the input image as R, G, B and S as their grayscale projection. Unless otherwise noted, we use a set of five projections: the three opponent color axes $\{(R+G+B)/3,\ (R-B)/3$, and $(2G-R-B)/4\}$ which have been experimentally shown by [48] to perform well in color segmentation, a neighborhood variance measure in S with a window size of 16, and an orientation coherency measure in S with a window size of 16.

For matching, we take a *consistent nearest neighbor* approach. Given a pair of images $\mathbf{I}_1,\ \mathbf{I}_2$ and their corresponding region sets $\mathscr{R}_1,\ \mathscr{R}_2$ computed from the same set of projections $\mathscr{B}$, the matching score between the two images is defined as the number of consistent nearest neighbor region pairs. A consistent nearest neighbor region pair is defined as a pair of regions with each being mutual nearest neighbors in a brute force search through both region sets. To be concrete, for region $R \in \mathscr{R}_1$, solve the following function

$$R_2^* = \arg\min_{R_2 \in \mathscr{R}_2} D(R, R_2), \tag{13}$$

where D is a distance function between the two region descriptions. Then, for the nearest neighbor R_2^*, solve the following function

$$R_1^* = \arg\min_{R_1 \in \mathscr{R}} D(R_1, R_2^*). \tag{14}$$

The match $\{R, R_2^*\}$ is considered consistent match if and only if $R_1^* = R$.

From the possible descriptions previously discussed (Section 6), there are two candidates for the distance function. First, if the kernel-weighted means are used to describe the regions, then a simple sum of squared distance measure is sufficient. Second, if the kernel-weighted variances are included in the description, then the more appropriate measure is the KL distance. Additionally, if the cooccurence information is maintained, and the descriptions stored are dependent on the projection in which the regions were extracted, then the distance is only valid between regions that have been extracted from an equivalent set of projections. The distance between regions that have been extracted from different projections is set to infinity.

7.1. Detection properties

The coherent regions have a number of good properties: stability/invariance, conciseness, and scalability. Since the image description is composed of a number of independent regions, like other local descriptor methods [1], it is robust to occlusion (shown

Fig. 6. A subset of the indoor dataset (chosen arbitrarily) used in the retrieval experiments.

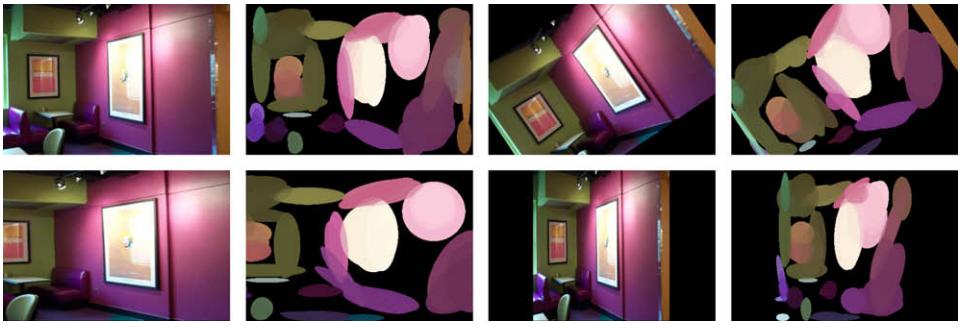

Fig. 7. The coherent regions extracted are robust to affine distortion of the image. The top-left is the original image, top-right is a rotation, bottom-left is an increase in the aspect ratio, and bottom-right is a reduction in the aspect ratio.

experimentally in Section 7.7). In addition, using the kernel functions to weight the region statistics increases the robustness since it weights pixels based on their distance from the region center and avoids the difficult problem of boundary localization.

We claim that the detection is robust to affine distortions in the image. In Fig. 7 we show the extracted regions using the RGB projection for exposition. To qualitatively analyze the detection, we have transformed by different affine maps: (top-right) is rotation, (bottom-left) increasing the aspect ratio, (bottom-right) reducing the aspect ratio. From the figure, we see that roughly the same regions are extracted.

We have also performed two experiments to quantitatively measure the detection invariance. For these experiments, a point projection is used, which is rotation, translation, and scale invariant. In the first experiment (Fig. 8), we analyze the detection repeatability under rotations in the image. For this experiment, to detect if a region is re-detected, we use only the spatial param-

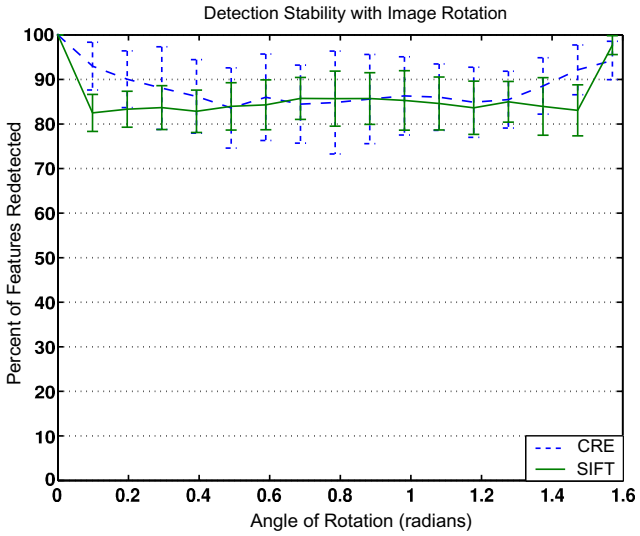

Fig. 8. Detection repeatability experiment for rotated images. Our method is labeled CRE (Coherent Region Extraction).

Table 1
Detection repeatability under random affine transformations of varying complexity. Our method is CRE (Coherent Region Extraction).

Grade	CRE (%)	SIFT(%)
1	95	93
2	92	91
3	88	89
4	89	88
5	87	86

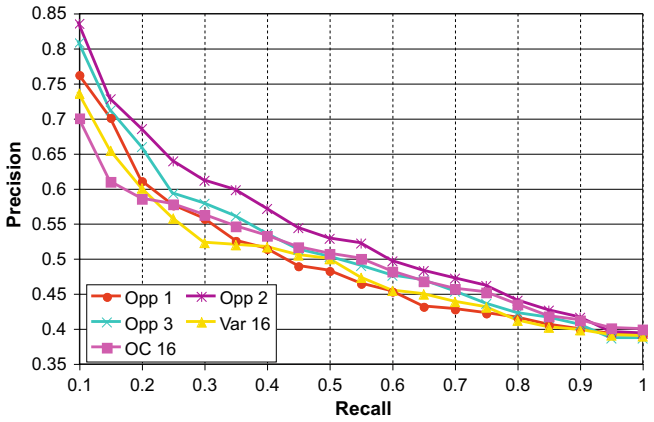

Fig. 9. Graph showing precision–recall for each of the five projections used in the experiments (independently).

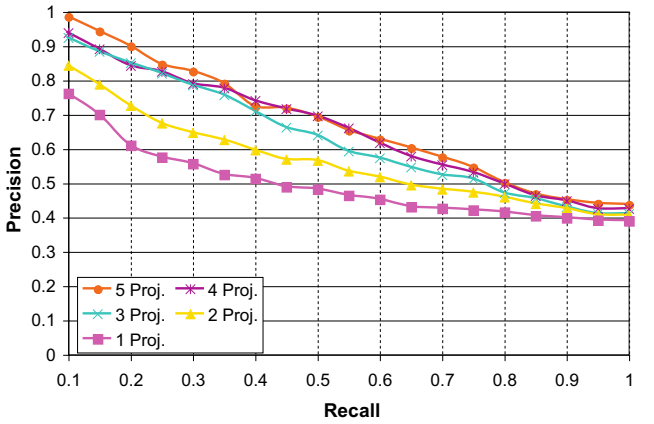

Fig. 10. Graph showing precision–recall as the number of projections (feature spaces) is varied.

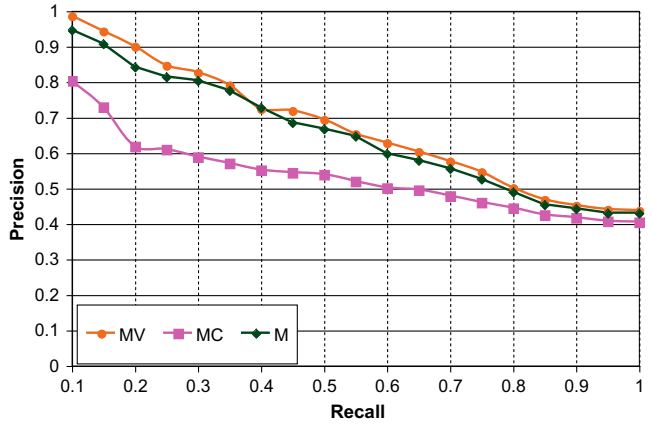

Fig. 11. Graph showing precision–recall for different region description algorithms.

eters (since we know the underlying geometric transformation and can directly invert it). We compare it against the SIFT method on the same images. We find the two methods perform comparably. In the second experiment (Table 1), we distort the image by an affine transformation chosen at random with varying complexity (five grades). The simplest transformations, grade 1, included scale changes (1 ± 0.1) and rotations $(\pm \frac{\pi}{16}$ radians). The most complex transformations, grade 5, included scale changes (1 ± 0.5), rotations $(\pm \frac{\pi}{2}$ radians), and skew (1 ± 0.3).

7.2. Projections analysis

The projections define the feature spaces in which the image will be analyzed for coherent content. The representative power of the resulting feature spaces is dependent on the projections used. In the following two experiments we analyze the matching sensitivity to the projections we have used in the experiments.

In Fig. 9, we show the representative power of each of the five projections that we use in the experiments. The three opponent axes are labeled Opp, the variance projection Var 16, and the orientation coherency projection OC 16. The graph indicates that the color projections are more representative of the image content in the test database than the two texture projections. The orientation coherency projection performs the worst initially, but, for greater recall rates, it improves with respect to the other projections. This change is because the images we use have very few regions of stripy texture, and thus, the color is more discriminative for low recall rates. However, for higher recall rates, the stripy region information is less ambiguous than the remaining color information. In Fig. 9, the Opp 1 projection is, essentially, the grayscale image; it is interesting to note that while it performs better than the variance and orientation coherency for recall rates up to 20%, for the remaining recall rates, it performs the worst. This degradation is due to the high variation in the lighting conditions between the images, and the raw grayscale data is not very robust to such photometric variation.

In Fig. 10, we show the effect of varying the number of projections used in the image description. For Proj. 1, we just use the grayscale image. For Proj. 2, we use the grayscale image and the variance projection with a neighborhood size of 16. For Proj. 3, we use the three opponent color axes, and for Proj. 4, we add the variance with neighborhood size 16. Proj. 5 is the same set of projections used in all the other experiments. We find that the addition of multiple projections greatly improves the retrieval accuracy.

7.3. Description comparison

In this experiment, we compare the candidate region descriptions from Section 6. The three descriptions we compare are:

MC—Kernel-weighted mean and cooccurence modeling. Here, we make explicit use of the projections from which the regions are extracted.
M—A kernel-weighted mean from each projection.
MV—A kernel-weighted mean and variance from each projection.

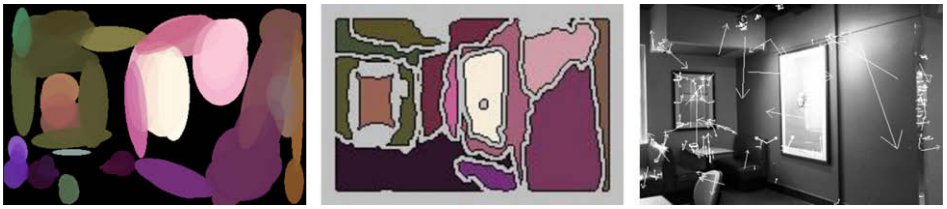

Fig. 12. Image representation for the three methods on the same image. For our technique (left) and Blobworld (middle), a color representation is used. For SIFT (right), the key locations, scales, and orientations are rendered by arrows.

Fig. 11 shows that the kernel-weighted mean and variance from each projection performs the best of the three techniques. One would expect the **MC** description to perform better since it explicitly incorporates the projections from which the regions are extracted. However, the resulting description is not as discriminative as those resulting from other two description algorithms.

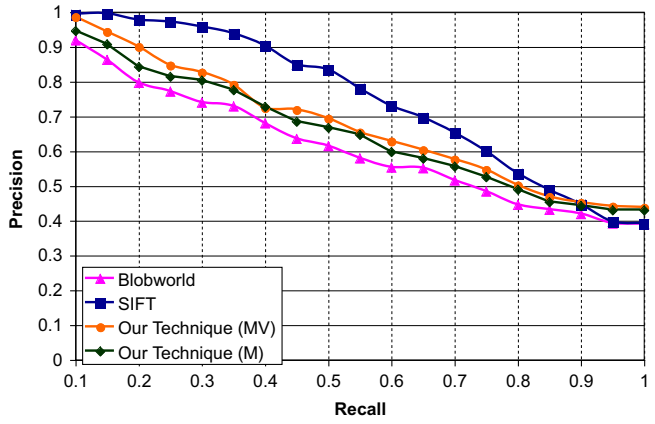

Fig. 13. Comparison between our technique and other published techniques.

Table 2
Comparison of average per-image storage for the three techniques.

	Average number of elements	Size per element (in words)	Average size (in words)
Our technique (M)	333	5	1665
Our technique (MV)	333	10	3330
Blobworld	9	239	2151
SIFT	695	32	22,260

7.4. Retrieval comparison

We compare our technique to two representative techniques for local and global image description: SIFT keys [3] and Blobworld [4], respectively. Fig. 12 gives a visualization of the different representations. SIFT is an example of a local, affine-insensitive and scale-invariant interest point descriptor. Note, that additional geometric constraints are plausible for both our method and SIFT key matching, but we do not employ any of them in order to keep the comparisons between methods fair. Blobworld is an example of using segmented image regions as the description. To measure matches using their provided source code, we used blob-to-blob queries. For a query image $\mathbf{I}$ with regions $r_1, \ldots, r_n$, we queried the database independently for each region r_i and maintained accumulators for each image. The final matches for the query image were those images with the highest accumulators after queries for all n regions had been issued.

Fig. 13 presents the precision–recall graph (average for querying on all images in the database) for each of the methods. For retrieval, we find the SIFT keys outperform the other two methods. This result agrees with the study by Mikolajczyk and Schmid [2]. Our method (**MV**) outperforms the Blobworld technique by about 6.5% precision on average. As we will discuss in the next section, the better SIFT performance on this initial retrieval experiment is likely a function of its high storage requirements, which are an order of magnitude more than our method and Blobworld. SIFT exploits such a redundant representation to give improved results; however, the scalability of SIFT is questionable.

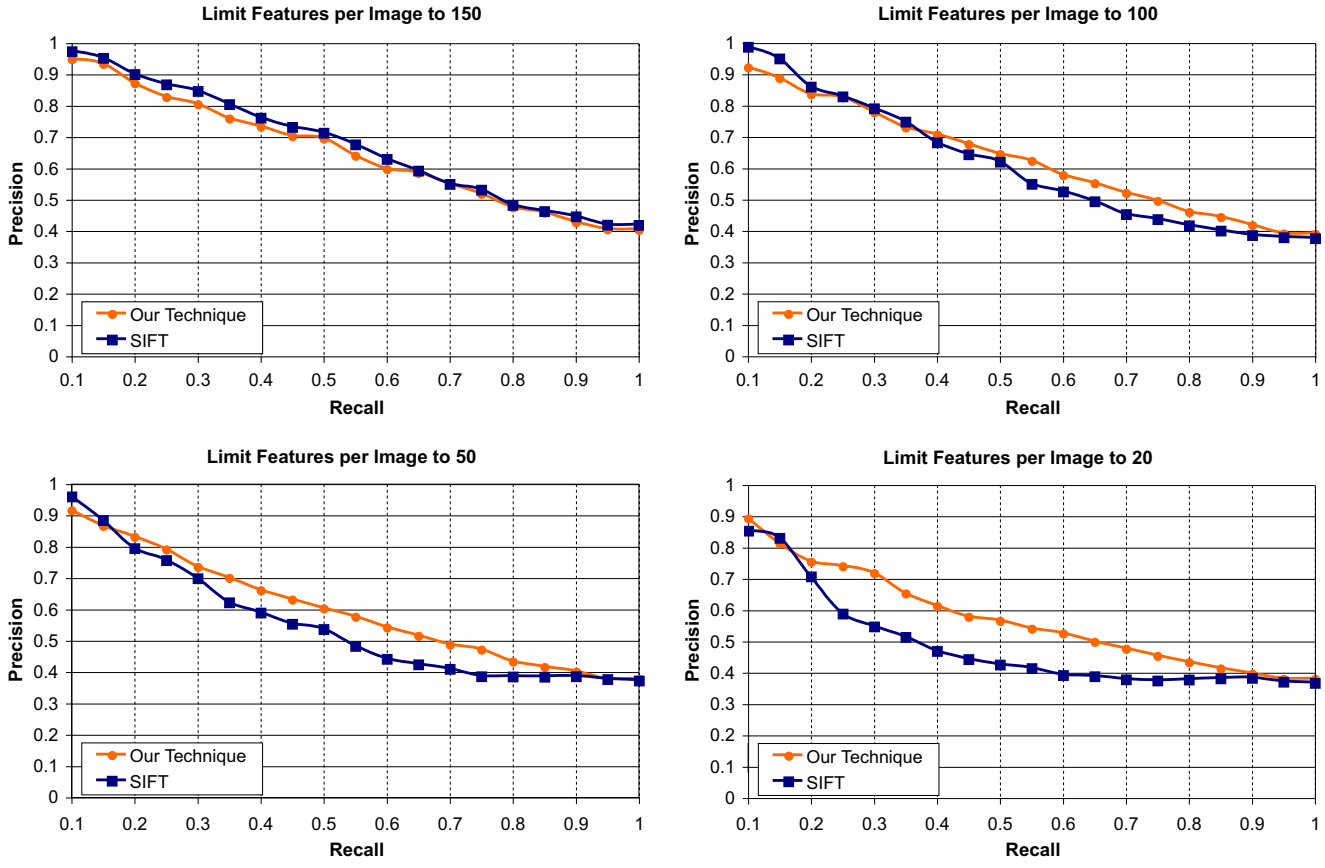

Fig. 14. Retrieval comparison for four different feature subset sizes.

7.5. Storage comparison

As an image or a scene database grows, querying it becomes more difficult and better indices or searching algorithms are required. In Table 2, we compare the storage efficiency for the three methods. We see that our method generates a data-size on the same order as Blobworld, which is far less than the SIFT approach. This data reflects the available source code for Blobworld and SIFT. It should be noted that the SIFT keys store 128 1-byte elements while the other two methods use 4-byte (1-word) floating point elements. We have not experimented with quantizing the storage for our technique to further reduce the size.

Next, we show the results of an experiment that compares the retrieval rates for the SIFT method with our method when they store an equivalent (or nearly equivalent) number of features. In this case, we are still not storing the same amount data since the length of the SIFT keys are 128 bytes and the length of our descriptors is 5, which is dependent on the number of projections (we use the standard five projections and the **M** description). As suggested by Lowe [3], the larger (in spatial scale) SIFT keys are generally more stable and robust to noise. Thus, to reduce the number of SIFT keys stored, we keep the largest. In choosing the subset of features for our method, we rely on the value of the objective function for each region, which incorporates both scale and homogeneity. We have shown in [49] that this method is better than scale alone. In Fig. 14, we show the precision–recall graph for four different subset sizes: 150, 100, 50, and 20. The two methods perform comparably at 150 feature with the SIFT method slightly outperforming the coherent regions. However, in the next three graphs, we find our technique drops in precision slightly for smaller subset sizes, but the SIFT method drops at a much faster rate. One can infer from these results that while SIFT is highly discriminative, it relies on large quantities of redundant data.

7.6. Robustness to affine distortion

In Section 7.1 we discussed the properties of our representation, and we claimed that it is robust to affine transformations of the image. To test this claim, we changed the aspect ratio of each image in the entire dataset and re-computed the coherent regions and SIFT keys. We performed a complete dataset query (same as above) and measured the precision–recall (Fig. 15) when querying with these distorted images. We used the **MV** description method. We experimented with aspect ratio changes of 0.5, 0.75. From the graphs, we see that our method is robust to the image distortion. At the extreme cases, it outperforms the SIFT method, which drops substantially.

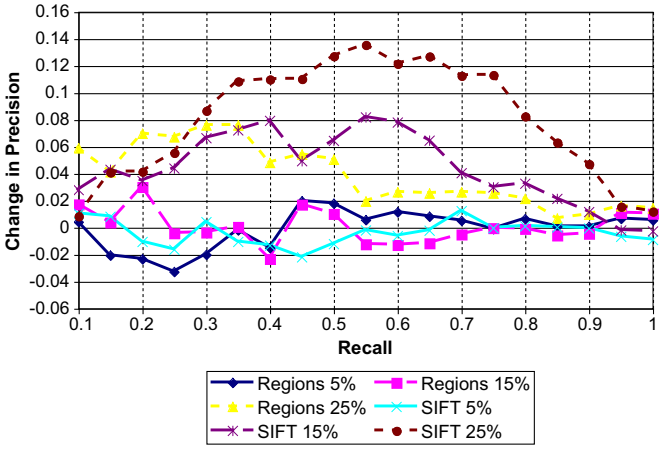

Fig. 16. Graph showing the change in precision under partial occlusion for our technique and the SIFT method. 0 change in precision means the occlusion has no effect on the retrieval, negative change in precision means the occlusion actually improved the rate, and positive change means the occlusion caused the retrieval rates to degrade.

7.7. Robustness to occlusion

As mentioned earlier, one of the benefits of the local interest-operator techniques is their robustness to occlusion since an entire image (or object) is represented as a set of independent (and local) measurements. Likewise, our method summarizes an image as a set of independent regions. To simulate the occlusion, we choose a rectangle independently at random in each image and turn all the pixel intensities in that region to 0. In Fig. 16, we compare the robustness to occlusion of our method and the SIFT method as we vary the amount of occlusion. To compute the change in precision we subtract the precision with occlusion from the precision without occlusion. Thus, 0 change in precision means the occlusion has no effect on the retrieval, negative change in precision means the occlusion actually improved the rate, and positive change means the occlusion caused the retrieval rates to degrade. An improvement is possible for small partial occlusions when the occluder masks an ambiguous image region. Essentially, a "smaller" change in precision means more robustness to the occlusion. We compare the same occlusion sizes: 5%, 15%, and 25%. We find that our technique is more robust to the occlusion in this experiment than the SIFT technique for the same respective occluders.

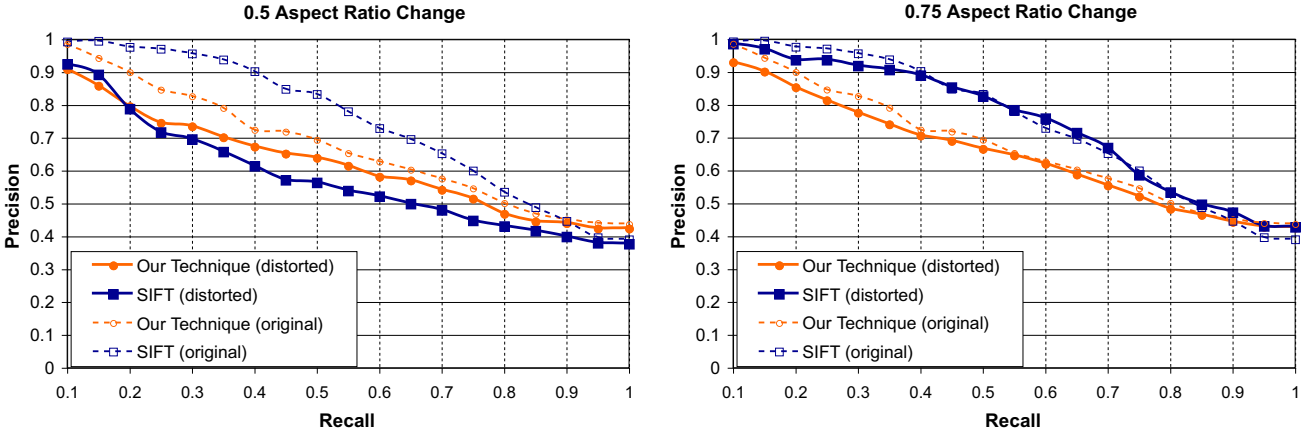

Fig. 15. Graph showing precision–recall for our technique and the SIFT method when querying with distorted images from the database.

8. Conclusion

We have presented a novel method for image representation using a kernel-based, sparse image segmentation and description method. The method is general in that it permits a variety of feature spaces that are represented as scalar image projections. We create a continuous scale-space of anisotropic regions with coherent image content. The regions are robust under drastic viewpoint changes and varying photometric conditions. Our experiments indicate that the methods are stable, reliable, and efficient in terms of both computation and storage. In particular, the use of spatial kernels admits efficient, optimization-based methods for segmentation and image matching.

The main contribution of this paper is the exploration of features that both summarize content and are moderately discriminative rather than features that are highly discriminative. In this sense, we integrate ideas from traditional segmentation into the interest point concepts thereby creating an interest region operator. However, it is clear that both approaches have their benefits and drawbacks. Thus, we envision an integrated framework where each interest point has a spectrum of representations associated with it: at one end of the spectrum is a summarizing representation and at the other end is the discriminating representation. Such a representation-spectrum will scale much better than either independently with the large image databases and demand for real-time applications we are beginning to find now and will increasingly see in the future.

Acknowledgments

This work was performed while Corso was with the Computational Interaction and Robotics Lab at Johns Hopkins University. This material is based upon work supported by the National Science Foundation under Grant No. 0112882. We are grateful to the group at UC Berkeley (Blobworld) and Lowe (SIFT) for providing copies of their source code/binaries on the world-wide-web thereby enabling our experiments.

Appendix A. Derivation of final objective function

We show that Eq. (7) approximates Eq. (6) (see [49] for a full discussion). We approximate the discrete sum of the Gaussian (1) over the image with the continuous integral:

$$\sum_{i\in\mathcal{I}} K(i,r) \approx \int\int K(\mathbf{x},r)d\mathbf{x} = 1, \tag{A.1}$$

where the integral is over $\mathbb{R}^2$ and $\mathbf{x}\in\mathbb{R}^2$. This integral is computed in closed-form, and, since the regions are generally smaller than and contained by the image, the approximation captures the majority of the region-area. We also need the squared kernel:

$$\sum_{i\in\mathcal{I}} K(i,r)^2 \approx \int\int K(r,\mathbf{x})^2 d\mathbf{x} = \frac{1}{4\pi|\mathbf{\Psi}|^{\frac{1}{2}}}. \tag{A.2}$$

First, we solve for the scale factor in the template term of Eq. (6).

$$\begin{aligned}
&\arg\min_{\beta} \sum[\beta K(i,r) - I(i)]^2 \\
&0 = \sum[\beta K(i,r) - I(i)]K(i,r) \\
&\beta = \frac{\sum I(i)K(i,r)}{\sum K(i,r)^2} \\
&\beta \approx 4\pi\alpha|\mathbf{\Psi}|^{\frac{1}{2}}.
\end{aligned} \tag{A.3}$$

Next, we simplify Eq. (6):

$$\begin{aligned}
&\arg\min_{\beta,\boldsymbol{\mu},\mathbf{\Psi}} \sum K(i,r)I(i)^2 - \alpha^2 + \frac{\gamma}{n}\sum[\beta K(i,r) - I(i)]^2 \\
&\arg\min_{\boldsymbol{\mu},\mathbf{\Psi}} \sum K(i,r)I(i)^2 - \alpha^2 + \frac{\gamma}{n}[4\pi|\mathbf{\Psi}|^{\frac{1}{2}}\alpha^2 - 8\pi|\mathbf{\Psi}|^{\frac{1}{2}}\alpha^2] + \text{const} \\
&\arg\min_{\boldsymbol{\mu},\mathbf{\Psi}} \frac{\sum K(i,r)I(i)^2}{\alpha^2} - \gamma\frac{4\pi|\mathbf{\Psi}|^{\frac{1}{2}}}{n} + \text{const}.
\end{aligned} \tag{A.4}$$

We assume that $\alpha^2 \neq 0$. Finally, we show that Eq. (7) is a first-order approximation to Eq. (6). We use a special case of the binomial series expansion [50]:

$$\begin{aligned}
(1-x)^{-r} &= \sum_{k=0}^{\infty} \frac{(r)_k}{k!}(-x)^k \\
&= 1 + rx + \frac{1}{2}r(r-1)x^2 + \frac{1}{6}r(r-1)(r-2)x^3 + \cdots
\end{aligned}$$

We have used the Pochhammer symbol $(r)_k = r(r+1)\cdots(r+k-1)$. The series converges for $|x|<1$. For the case $r=1$, we have $(1-x)^{-1} = 1 + x + x^2 + \cdots$ Let $\tau = \frac{n}{4\pi\gamma}$, and write $B = \frac{4\pi\gamma}{n}|\mathbf{\Psi}|^{\frac{1}{2}}$.

$$\begin{aligned}
\frac{\tau}{|\mathbf{\Psi}|^{\frac{1}{2}}} &= \frac{n}{4\pi\gamma|\mathbf{\Psi}|^{\frac{1}{2}}} = B^{-1} = (1-(1-B))^{-1} \\
&= 1 + (1-B) + (1-B)^2 + \cdots \approx -B = -\frac{4\pi\gamma}{n}|\mathbf{\Psi}|^{\frac{1}{2}}.
\end{aligned} \tag{A.5}$$

For the binomial expansion, we must ensure $|(1-B)|<1$. We derive bounds for γ to ensure $0<B<2$. Note $|\mathbf{\Psi}|^{\frac{1}{2}} > 0$, and $n>0$. The lower bound is clearly $\gamma > 0$. The upper bound derivation follows:

$$\begin{aligned}
&B < 2 \\
&\frac{4\pi\gamma}{n}|\mathbf{\Psi}|^{\frac{1}{2}} < 2 \\
&\gamma < \frac{n}{2\pi}|\mathbf{\Psi}|^{\frac{1}{2}}.
\end{aligned} \tag{A.6}$$

Assuming (A.5) and (A.6) shows that the objective function defined in Eq. (7) is an first-order approximation of Eq. (6).

We now discuss the bound (A.6) to show that the approximation holds in our experiments (Section 7). Recall n is the number of pixels in the image, which is on the order of 10^5. We implement (7) with the $\tau = 1$. Then, by definition, $\gamma = \frac{n}{4\pi}$. Given the bound (A.6), we can determine for what size regions this approximation holds, i.e. we get a bound for $\mathbf{\Psi}^{\frac{1}{2}}$:

$$\begin{aligned}
&\gamma < \frac{n}{2\pi}|\mathbf{\Psi}|^{\frac{1}{2}} \\
&\frac{n}{4\pi} < \frac{n}{2\pi}|\mathbf{\Psi}|^{\frac{1}{2}} \\
&\frac{1}{2} < |\mathbf{\Psi}|^{\frac{1}{2}}.
\end{aligned} \tag{A.7}$$

Therefore, the approximation holds for all but extremely small regions. Since the unit is the pixel, the lower bound roughly corresponds to regions that are smaller than a pixel.

References

[1] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (5) (1997) 530–535.
[2] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 257–264.
[3] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[4] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using Expectation-Maximization and its application to image querying, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1026–1038.

[5] J. Malik, S. Belongie, J. Shi, T. Leung, Textons, contours, and regions: cue combination in image segmentation, in: International Conference on Computer Vision, 1999.
[6] H. Greenspan, J. Goldberger, L. Ridel, A continuous probabilistic framework for image matching, Computer Vision and Image Understanding 84 (2001) 384–406.
[7] F. Schaffalitzky, A. Zisserman, Viewpoint invariant texture matching and wide baseline stereo, in: International Conference on Computer Vision, vol. 2, 2001, pp. 636–643.
[8] H.P. Moravec, Visual mapping by a Robot Rover, in: International Joint Conference on Artificial Intelligence, 1979, pp. 598–600.
[9] C. Harris, M. Stephens, A combined corner and edge detector, in: ALVEY Vision Conference, 1988, pp. 147–151.
[10] C. Schmid, Appariement d'images par invariants locaux de niveaux de gris, Ph.D. Thesis, Institut National Polytechnique de Grenoble, France, 1996.
[11] J.J. Koenderink, A.J. van Doorn, Representation of local geometry in visual systems, Biological Cybernetics 55 (1987) 367–375.
[12] D. Lowe, Object recognition from local scale-invariant features, in: International Conference on Computer Vision, 1999.
[13] T. Lindeberg, Scale-Space Theory in Computer Vision, Kluwer Academic Publishers, 1994.
[14] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, Tech. Rep. 15, Intel, 2003.
[15] S. Lazebnik, C. Schmid, J. Ponce, Affine-invariant local descriptors and neighborhood statistics for texture recognition, in: International Conference on Computer Vision, 2003, pp. 649–656.
[16] T. Tuytelaars, L.V. Gool, Wide baseline stereo matching based on local, affinely invariant regions, in: British Machine Vision Conference, 2000.
[17] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: British Machine Vision Conference, 2002.
[18] J. Matas, S. Obdrzalek, O. Chum, Local affine frames for wide-baseline stereo, in: International Conference on Pattern Recognition, 2002.
[19] D. Tell, S. Carlsson, Wide baseline point matching using affine invariants computed from intensity profiles, in: European Conference on Computer Vision, 2000, pp. 814–828.
[20] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley and Sons, 1991.
[21] S. Gilles, Robust description and matching of images, Ph.D. Thesis, University of Oxford, 1998.
[22] T. Kadir, M. Brady, Saliency, scale and image description, International Journal of Computer Vision 43 (2) (2001) 83–105.
[23] J.S. Hare, P.H. Lewis, Scale saliency: applications in visual matching, tracking and view-based object recognition, in: Distributed Multimedia Systems/Visual Information Systems, 2003, pp. 436–440.
[24] F. Fraundorfer, H. Bischof, Detecting distinguished regions by saliency, in: Scandinavian Conference on Image Analysis, 2003, pp. 208–215.
[25] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, International Journal of Computer Vision 37 (2) (2000) 151–172.
[26] M.J. Swain, D.H. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.
[27] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B 39 (1) (1977) 1–38.
[28] P. Grunwald, A tutorial introduction to the minimum description length principle, in: P. Grnwald, I. Myung, M. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications, MIT Press, 2005.
[29] J. Rissanen, Modeling by shortest data description, Automatica 14 (5) (1978) 465–471.
[30] H. Greenspan, J. Goldberger, A. Mayer, Probabilistic space-time video modeling via piecewise GMM, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (3) (2004) 384–397.
[31] X. Mo, R. Wilson, Video modelling and segmentation using Gaussian mixture models, in: International Conference on Pattern Recognition, vol. 3, 2004, pp. 854–857.
[32] D.G. Ruiz, H. Takahashi, M. Nakajima, Ubiquitous image categorization using color blobs, in: Nicograph, 2003.
[33] T. Lindeberg, Principles for automatic scale selection, Handbook on Computer Vision and Applications, vol. 2, Academic Press, Boston, USA, 1999, pp. 239–274.
[34] D. Comaniciu, V. Ramesh, P. Meer, The variable bandwidth mean shift and data-driven scale selection, in: International Conference on Computer Vision, vol. 1, 2001, pp. 438–445.
[35] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.
[36] R. Collins, Mean-shift blob tracking through scale space, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003.
[37] K. Okada, D. Comaniciu, A. Krishnan, Scale selection for anisotropic scale-space: application to volumetric tumor characterization, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. 594–601.
[38] R. Potts, Some generalized order–disorder transformation, Proceedings of the Cambridge Philosophical Society 48 (1952) 106–109.
[39] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Communications on Pure and Applied Mathematics XLII (1989) 577–685.
[40] G.J. McLachlan, K.E. Basford, Mixture Models: Inference and Applications to Clustering, Marcel Dekker, Inc., 1988.
[41] J.J. Corso, G.D. Hager, Coherent regions for concise and stable image description, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005.
[42] D. Marr, E. Hildreth, Theory of edge detection, in: Royal Society of London B, vol. 290, 1980, pp. 199–218.
[43] T. Randen, J.H. Husoy, Filtering for texture classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4) (1999) 291–311.
[44] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.
[45] R. Collins, Y. Liu, On-line selection of discriminative tracking features, in: International Conference on Computer Vision, vol. 1, 2003, pp. 346–352.
[46] D. Gabor, Theory of communication, Journal of the IEE 3 (93) (1946) 429–457.
[47] B. Jahne, Digital Image Processing, fifth ed., Springer, 2001.
[48] Y. Ohta, T. Kanade, T. Sakai, Color information for region segmentation, Computer Graphics and Image Processing 13 (3) (1980) 222–241.
[49] J.J. Corso, Techniques for vision-based human–computer interaction, Ph.D. Thesis, The Johns Hopkins University, 2005.
[50] E.W. Weisstein, Binomial Series, From Mathworld—A Wolfram Web-Resource. Available from: <http://www.mathworld.wolfram.com/BinomialSeries.html>.