Multimed Tools Appl (2010) 48:339–365
DOI 10.1007/s11042-009-0347-z

A multiple instance learning based framework for semantic image segmentation

Iker Gondra · Tao Xu

Published online: 18 August 2009

Abstract Most image segmentation algorithms extract regions satisfying visual uniformity criteria. Unfortunately, because of the semantic gap between low-level features and high-level semantics, such regions usually do not correspond to meaningful parts. This has motivated researchers to develop methods that, by introducing high-level knowledge into the segmentation process, can break through the performance ceiling imposed by the semantic gap. The main disadvantage of those methods is their lack of flexibility due to the assumption that such knowledge is provided in advance. In content-based image retrieval (CBIR), relevance feedback (RF) learning has been successfully applied as a technique aimed at reducing the semantic gap. Inspired by this, we present a RF-based CBIR framework that uses multiple instance learning to perform a semantically-guided context adaptation of segmentation parameters. A partial instantiation of this framework that uses mean shift-based segmentation is presented. Experiments show the effectiveness and flexibility of the proposed framework on real images.

Keywords Image segmentation · Multiple instance learning · Content-based image retrieval · Relevance feedback · Mean shift · Semantic gap · Adaptive segmentation · Segmentation parameters · Diverse density · Clustering

1 Introduction

Over the last four decades, the development of image segmentation algorithms has been an area of considerable research activity. The reason for such significant

I. Gondra (✉) · T. Xu
Department of Mathematics, Statistics, and Computer Science, St. Francis Xavier University, Antigonish, Nova Scotia, Canada
e-mail: igondra@stfx.ca

T. Xu
e-mail: x2006opl@stfx.ca

attention to this problem lies in its practical importance. Image segmentation is a key step towards high-level tasks such as image understanding, and serves in a wide range of applications including object recognition, scene analysis or content-based image/video retrieval.

Many image segmentation algorithms have been developed and different classification schemes have been proposed (e.g., [18]). In edge-based approaches (e.g., [14]), segmentation is based on spatial discontinuities. That is, by detecting sudden changes in local features, region boundaries can be obtained. In region-based approaches, segmentation is based on spatial similarity among pixels. Thus, a measure of region homogeneity has to be defined in advance. An approach to homogeneity-based segmentation makes use of clustering methods, which classify pixels into one of several groups. Mean shift clustering [7] is widely used in the vision community. It is derived from the Parzen window approach for nonparametric density estimation [10]. It finds the modes (dense areas) of the underlying probability density function of the image pixel values and associates with them pixels in their basin of attraction. One of the main advantages of this algorithm over most other clustering techniques is that it does not rely upon a priori knowledge of the number of clusters (e.g., the value k in the case of k-means clustering [17]). Furthermore, it does not implicitly assume any particular shape (e.g., elliptical) for the clusters. Thus, it allows for the analysis of arbitrarily structured feature spaces. However, the selection of scale parameters (window width values) remains a difficult problem with a strong impact on its performance. Some variations [6, 13] of this algorithm exploit data-driven properties to perform a more adaptive segmentation and have been used with other approaches [25]. The normalized cuts framework [22], which is capable of detecting clusters of various shapes, is an example of a clustering based approach derived from graph theory. Other important methods, that are difficult to classify under the edge-based or region-based categories, include segmentation using the Expectation-Maximization algorithm (e.g., [4]), using Markov Chains (e.g., [24]), and hybrid techniques that use both edge and homogeneity information (e.g., [11]).

Semantic segmentation, which corresponds to a partitioning of an image's pixels into regions that are semantically meaningful to people, remains a difficult and as yet largely unsolved problem. Most image segmentation algorithms extract regions satisfying some uniformity (homogeneity) criterion, which is based on low-level data-driven visual features (e.g., color, texture). Those algorithms perform well in narrow domains (e.g., medical images, frontal views of faces), where the variability of low-level visual content is limited. Unfortunately, in broader domains, homogeneous regions do not necessarily (and usually do not) correspond to semantically meaningful units. This is mainly caused by the disconnection between low-level visual features and high-level semantics, which is commonly referred to as the semantic gap. Under these circumstances, a number of learning-based approaches have been proposed to tackle this problem. In order to establish the correspondence between visual content and semantics through learning methods, a parameterized mapping from input space (e.g., features) to output space (e.g., segmentation) must be specified in advance so that the mapping can be learned by fixing its parameters during learning.

In [1, 2], an object recognition approach that compares the segmentation results with a target model is described. The output of an evaluation function is repeatedly fed back into the segmentation algorithm for tuning segmentation parameters. The system works in such a loop until the value of the function is maximized. The final set of parameter values is thought of as the most suitable one for segmenting images

that contain the target object. Genetic and hybrid algorithms are used for learning the segmentation parameters. The main drawback of the system is that both the object model and the position of the target object are assumed to be known, which makes the algorithm only suitable for limited applications. The work presented in [20] continued along this direction by developing an object recognition system that performs hill-climbing to maximize a real-valued feedback obtained from an object matching function. The system framework is given in Fig. 1. An image is first partitioned by a segmentation algorithm with a set of initial parameter values. Next, image features are extracted based on the resulting segmentation. The obtained features are then fed into the model matching function, which outputs a real-valued feedback indicating the degree of matching between the computed features and the stored object model. Segmentation parameters are then updated by a reinforcement learning algorithm that accepts the degree of matching as reinforcement. The procedure repeats until the degree of matching reaches a threshold or the number of iterations exceeds a certain upper bound. This approach still makes the assumption that the object model is known, but a fixed position is not necessary. Notice that [1, 2, 20] implement a closed-loop segmentation, which is fully dependent on the object model.

In [21], a method is described that, given semantic image segmentations, facilitates the segmentation of novel images within the same domain. The main disadvantage of this approach is that, while it does not require an explicit object model, it is still dependent on providing semantic segmentations of images that depict the same, or similar, objects as the novel images, and in a similar setting. In [3], a figure-ground learning scheme for class-based segmentation is presented. A set of training images in combination with top-down and bottom-up segmentation processes is used to extract class-relevant informative fragments. This is followed by a learning process in which each fragment is divided into figure and background. Thereafter, this representation is used to segment novel images that belong to the same class. The main drawback of the system is that the figure-ground learning process relies on two main criteria both of which depend on key parameters that have to be manually tuned from experience.

The main disadvantage of current segmentation approaches that incorporate high-level knowledge is their lack of flexibility due to the assumption that such knowledge (in the form of e.g., object models, samples of semantic segmentations, hand-chosen parameters) is provided in advance, which heavily restricts the application domain. In the context of content-based image retrieval (CBIR) [8, 23], relevance feedback (RF) learning [27] has been successfully applied as a technique aimed at reducing the semantic gap. Inspired by this, in this paper we present a

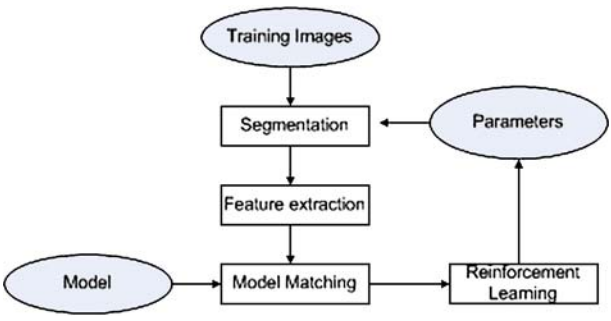


Fig. 1 A reinforcement learning-based object recognition system, where segmentation parameters are adjusted to optimize the output from the model matching function

RF-based CBIR framework that uses multiple instance learning (MIL) to perform a semantically-guided context-adaptation of segmentation parameters. The flexibility of the proposed scheme makes it very suitable for general-purpose applications. We present a partial instantiation of the proposed framework that uses mean shift clustering [7] as the segmentation algorithm. Our initial investigation into this idea was put forth in [12].

The rest of this article is organized as follows. Section 2 gives a brief introduction to MIL. The proposed framework is presented in Section 3. In Section 4, an overview of mean shift clustering is given. A partial instantiation of the proposed framework that uses mean shift clustering as the segmentation algorithm is presented in Section 5. Experimental results with real images are presented in Section 6. Finally, concluding remarks are given in Section 7.

2 Multiple instance learning

For learning problems in which there is a one-to-one correspondence between instances and labels, standard supervised learning provides an abundance of methods that can be used. Multiple instance learning (MIL) [9] receives a set of labeled *bags* as training samples. Formally, let $\mathcal{B}_i^+ = \{\mathbf{x}_{i,1}^+, \mathbf{x}_{i,2}^+, \ldots\}$ be the i^{th} positive bag, where $\mathbf{x}_{i,j}^+ \in \Re^d$ is the (feature vector representation of the) j^{th} instance in that bag. A similar notation is used for the negative bags. In a positive bag, at least one of the instances is responsible for the bag being labeled as positive, but (unlike standard supervised learning), we are not told which one(s). The objective for MIL is to find the most likely "true concept" $\mathbf{t} \in \Re^d$ in the instance space, which is as close as possible to at least one instance from each positive bag, and far from instances in the negative bags. Intuitively, $\mathbf{t}$ is a prototypical description of the common instance(s) among positive bags that is responsible for their positive labeling.

In 1998, Maron et al. [16] proposed the maximum diverse density algorithm, a probabilistic solution to MIL problems. This algorithm is perhaps the best known algorithm in MIL because of its elegant theoretical model. The idea behind it is straightforward: construct a probabilistic model to find the $\mathbf{t}$ that is close to at least one instance from each positive bag and far from all instances in negative bags. Therefore, $\mathbf{t}$ is located in a region that is not only dense in positive instances but also diverse in that it includes at least one instance from each positive bag (See Fig. 2).

The point $\mathbf{t}$ can be located by maximizing the following conditional probability over all points $\mathbf{x}$ in the feature space [16]

$$\Pr\left(\mathbf{x} = \mathbf{t} | \mathcal{B}_1^+, \ldots, \mathcal{B}_n^+, \mathcal{B}_1^-, \ldots, \mathcal{B}_m^-\right).$$

Assuming a uniform prior probability over $\mathbf{t}$ and applying Bayes' rule, this is equivalent to maximizing the likelihood

$$\arg\max_{\mathbf{x}} \Pr\left(\mathcal{B}_1^+, \ldots, \mathcal{B}_n^+, \mathcal{B}_1^-, \ldots, \mathcal{B}_m^- | \mathbf{x} = \mathbf{t}\right).$$

Making the additional assumption that each bag is conditionally independent given the target concept $\mathbf{t}$, this can written in product form

$$\arg\max_{\mathbf{x}} \prod_i \Pr\left(\mathcal{B}_i^+ | \mathbf{x} = \mathbf{t}\right) \prod_i \Pr\left(\mathcal{B}_i^- | \mathbf{x} = \mathbf{t}\right).$$

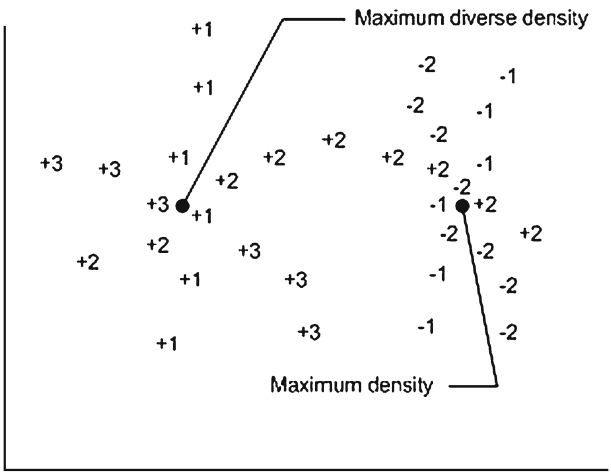


Fig. 2 The idea is to find areas that are close to at least one instance from every positive bag and far from instances in negative bags. In a two-dimensional feature space, each positive instance is denoted by the + sign with its bag number. Negative instances are similarly represented except that the − sign is used. The true concept point where the diverse density is maximized is not necessarily the one with the maximum density

Again, with the assumption of a uniform prior probability over $\mathbf{t}$ and applying Bayes' rule once more, this becomes

$$\arg\max_{\mathbf{x}} \prod_i \Pr\left(\mathbf{x}=\mathbf{t}|\mathcal{B}_i^+\right) \prod_i \Pr\left(\mathbf{x}=\mathbf{t}|\mathcal{B}_i^-\right).$$

In our approach, in order to estimate $\Pr(\mathbf{x}=\mathbf{t}|\mathcal{B}_i)$, the noisy-or model [19] is used. This model is based on two assumptions. First, for $\mathbf{t}$ to be the true concept, it is caused (and thus close to) at least one of the instances in each of the positive bags. Second, the probability of an instance not being the target concept is independent of the probability of any other instance not being the target concept. Based on these assumptions, we have

$$\Pr\left(\mathbf{x}=\mathbf{t}|\mathcal{B}_i^+\right) = \Pr\left(\mathbf{x}=\mathbf{t}|\mathbf{x}_{i,1}^+, \mathbf{x}_{i,2}^+, \ldots\right) = 1 - \prod_j \left(1 - \Pr\left(\mathbf{x}=\mathbf{t}|\mathbf{x}_{i,j}^+\right)\right).$$

Likewise, since all instances in a negative bag are negative, we have

$$\Pr\left(\mathbf{x}=\mathbf{t}|\mathcal{B}_i^-\right) = \Pr\left(\mathbf{x}=\mathbf{t}|\mathbf{x}_{i,1}^-, \mathbf{x}_{i,2}^-, \ldots\right) = \prod_j \left(1 - \Pr\left(\mathbf{x}=\mathbf{t}|\mathbf{x}_{i,j}^-\right)\right).$$

If we model the probability of an instance being the target concept as related to the distance between them, then we have

$$\Pr\left(\mathbf{x}=\mathbf{t}|\mathbf{x}_{i,j}\right) = \exp\left(-\left\|\mathbf{x}_{i,j}-\mathbf{x}\right\|^2\right).$$

Because the multiplication of probabilities might result in a number that is too small to represent on a computer, as suggested in [16], the negative log-likelihood is used instead. Thus, a gradient descent method can be used to search for the global minimum, where the diverse density is maximized. In order to avoid the gradient descent method getting trapped into a local optimum, multiple iterations might be required. Fortunately, according to the definition, the global minimum point is made of contributions from some set of positive bags. Thus, if we start a gradient descent from every instance in a positive bag, one of them is likely to be closest to the global minimum point $\mathbf{t}$, contribute the most to it and have a descend directly to it [16].

3 Proposed framework

In the context of content-based image retrieval (CBIR) [8, 23], relevance feedback (RF) learning [27] has been successfully applied as a technique aimed at reducing the semantic gap. It works by gathering semantic information from user interaction. The simplest form of RF is to indicate which images in the retrieval set are relevant. Thus, the amount of knowledge that is given as input consists of only one additional bit of information which indicates the image as positive (class) or negative (non-class). Based on this, context adaptation is performed by adjusting the retrieval scheme parameters to better serve the user's semantic intent. This process iterates until the user is satisfied with the retrieved images or stops searching. Thus, RF learning can be seen as a form of supervised learning that finds relations between high-level semantic interpretations and low-level visual properties. Hence, it attempts to reduce the semantic gap by tailoring the retrieval scheme parameters to the narrow image domain the user has in mind when searching for a particular type of images. Inspired by this, we realized that the RF obtained from a regular query session can also be exploited to automatically estimate segmentation scheme parameters. That is, in the same way that RF has been successfully used to perform context adaptation of retrieval scheme parameters, it can also be used to perform context adaptation of segmentation scheme parameters. A description of the proposed framework is given next.

The proposed framework (See Fig. 3) is an extension of a generic CBIR system that uses RF learning to improve retrieval performance. The image/feature database

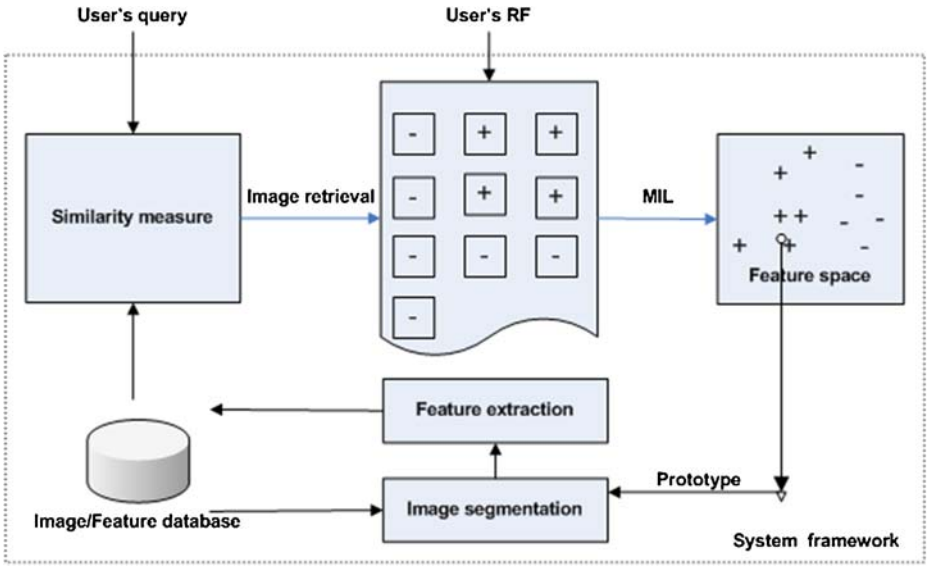


Fig. 3 Proposed framework

stores images, their segmentations, and their corresponding region-based visual feature sets. Obviously, the quality of a region-based representation is directly affected by the quality of the corresponding segmentation. The image segmentation and feature extraction components implement the particular image segmentation and region-based visual feature extraction algorithms. After the user submits a query image to the system through the user interface, a region-based image similarity measure (e.g., [5, 15]) is used to retrieve the closest images in the database and form the retrieval set. As is usual in RF-based CBIR systems, the user then labels each of the images in the retrieval set as either positive or negative. This RF is then used to perform context adaptation for both the retrieval and the segmentation scheme. The retrieval scheme can be adjusted by using any of the available RF learning algorithms [27].

For a given query image, we assume that the user's decision to label an image in the retrieval set as positive is based on the presence of a particular object of interest (OOI) in the image. Similarly, a user labels an image in the retrieval set as negative if none of the objects in the image correlates with the user's semantic intent. Based on these assumptions, at the end of the query session, the set of cumulative positive and negative images is used to find common (or similar) regions in the positive images that do not appear in (or are very distinct from) regions in the negative images. It is hypothesized that the commonalities (i.e., the regions) among positive images correspond to fragments of the OOI. The feature vectors of those regions are then used for obtaining object-specific knowledge on the distribution of the OOI's visual appearance. This knowledge can then be used to perform a semantically-guided context adaptation of segmentation parameters. Subsequently, the initial segmentation of the OOI in each of the positive images in the database can be revised. Specifically, we use multiple instance learning (MIL) [9] to identify the OOI and to find a prototypical feature vector for it. Intuitively, the "true concept" is a prototypical description of the common OOI among positive images that is responsible for their positive labeling. We also find an estimate of the maximum acceptable variability from the prototypical representation for the OOI, which we call the tolerance.

Formally, let $\mathcal{B}_i^+ = \{\mathbf{x}_{i,1}^+, \mathbf{x}_{i,2}^+, \ldots\}$ be segmentation information for the i^{th} positive image in the retrieval set, where $\mathbf{x}_{i,j}^+ \in \Re^d$ is the (feature vector representation of the) j^{th} region. A similar notation is used for the negative images. Note that this is just part of the segmentation information that is kept for each image in the image/feature database. Let the retrieval set be $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$, where $\mathcal{B}^+$ and $\mathcal{B}^-$ are the sets of positive and negative images respectively. By using the maximum diverse density algorithm (described in Section 2), we can find the mean representation $\mathbf{t} \in \Re^d$ (i.e., the true concept that is responsible for the labels of all positive and negative images) for the OOI. To fully characterize the OOI's distribution in the feature space, we also need to estimate a range centered about $\mathbf{t}$, within which every region will be classified as an instance of the OOI. Clearly, this range defines the maximum acceptable variability from $\mathbf{t}$ for the OOI, which is termed as the tolerance $v \in \Re$. Thus, in a high-dimensional feature space, each OOI is uniquely represented by a pair of $(\mathbf{t}, v)$ which actually defines a hypersphere with $\mathbf{t}$ as the origin and v as the radius. Any feature vector that falls into the same hypersphere is considered an instance of this OOI. Figure 4 depicts this idea graphically.

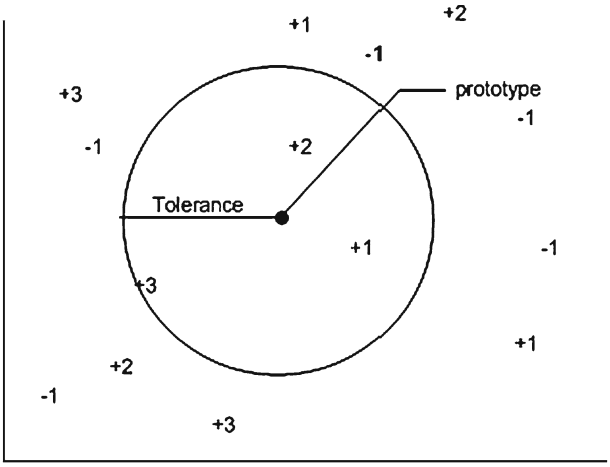


Fig. 4 In a two-dimensional feature space, the OOI's distribution is uniquely characterized by a circle, the origin of which is the prototype learned through the maximum diverse density algorithm and the radius is the maximum acceptable variability for that object model, i.e. the tolerance. In this example with three positive images and one negative image, the *circle* is the smallest one that is centered at the true concept and contains at least one region from each positive image

In order to obtain a reasonable estimate for v, for each $\mathcal{B}_i^+$ in $\mathcal{B}^+$, the most similar $\mathbf{x}_{i,j}^+$ with respect to $\mathbf{t}$ is found and the distance of the least similar one of all such regions (from all images in $\mathcal{B}^+$) is chosen as the value of v. Formally, the computation of v is given by

$$v = \max_{0 \le i \le |\mathcal{B}^+|} \left(\min_{0 \le j \le |\mathcal{B}_i^+|} \left(\left\| \mathbf{x}_{ij}^+ - \mathbf{t} \right\|^2 \right) \right).$$

Algorithm 1 summarizes the computation of both $\mathbf{t}$ and v, which characterize the OOI.

Algorithm 1 ModelLearning

Input: $\mathcal{B}$
Output: $\mathbf{t}$, v

```
Run maximum diverse density algorithm on B to obtain t
v ← −∞
for each B_i^+ ∈ B^+ do
    min ← ∞
    for each x_ij^+ ∈ B_i^+ do
        If ( ||x_ij^+ − t||^2 < min ) then
            min ← ||x_ij^+ − t||^2
        end if
    end for
    if (min > v) then
        v ← min
    end if
end for
```

The prototypical feature vector, tolerance, and positive images in the retrieval set are then passed to the image segmentation and feature extraction components to perform a re-segmentation of the OOI in those images. This is done in a way that is specific to the particular image segmentation and feature extraction components that are used in the framework. A partial instantiation of this framework that uses mean shift clustering as the segmentation algorithm is presented in Section 5.

As previously mentioned, the main disadvantage of current segmentation approaches that incorporate high-level knowledge is their lack of flexibility due to the assumption that such knowledge is provided in advance, which heavily restricts the application domain. That is, those approaches use a "closed-loop segmentation" in which segmentation is improved by automatically adjusting parameters using existing object models. Notice that, in the proposed framework, machine learning is used to estimate model parameters of the particular object that is to be segmented. Thus, the context adaptation of segmentation parameters and re-segmentation process requires no additional input knowledge and is thus very flexible and completely transparent to the users of the CBIR system.

4 Mean shift clustering

Mean shift clustering [7] is widely used in the vision community. It is derived from the Parzen window approach for nonparametric density estimation [10]. The Parzen window approach (kernel density estimation) to estimating an unknown probability density function $p(\mathbf{x})$ is the most popular density estimation method. As a nonparametric procedure, it can be used with arbitrary distributions and without the assumption that the shape of the underlying density is known. Given a set of sample points $\mathbf{x}_i, i = 1, \ldots, n$ in d-dimensional space $\Re^d$, the number of points falling in a d-dimensional hypercube with edge length h and centered at $\mathbf{x}$ is given by

$$\sum_{i=1}^{n} k\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

where $k(\mathbf{u})$ is the following window function (kernel)

$$k(\mathbf{u}) = \begin{cases} 1 \ |u_j| \leq \frac{1}{2}; \forall j = 1, \ldots, d \\ 0 \text{ otherwise.} \end{cases}$$

Thus an estimate of the space-averaged density at a point $\mathbf{x}$ is given by

$$\hat{p}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} k\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right). \tag{1}$$

Because the window width h determines the volume (i.e., h^d) of the hypercube, it has a strong effect on the accuracy of the estimate $\hat{p}(\mathbf{x})$ [10]. Although an estimate of $p(\mathbf{x})$ can be obtained by (1), sometimes, we are more interested on the modes (high-density areas) of $p(\mathbf{x})$ rather than on $p(\mathbf{x})$ itself. The mean shift procedure finds such

modes without estimating $p(\mathbf{x})$ in advance. Assuming a radially symmetric kernel function $K(\mathbf{u})$, (1) can be rewritten as

$$\hat{p}(\mathbf{x}) = \frac{c}{nh^d} \sum_{i=1}^{n} K\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \tag{2}$$

where c is a normalization constant [7]. The modes of $p(\mathbf{x})$ are located among the zeros of its gradient $\nabla p(\mathbf{x})$, which is estimated by the gradient of (2)

$$\nabla \hat{p}(\mathbf{x}) = \frac{2c}{nh^{d+2}} \sum_{i=1}^{n} (\mathbf{x} - \mathbf{x}_i) K'\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right). \tag{3}$$

Introducing $g(\mathbf{x}) = -K'(\mathbf{x})$ into (3) yields

$$\nabla \hat{p}(\mathbf{x}) = \frac{2c}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right]. \tag{4}$$

The first term of the product in (4) is proportional to $p(\mathbf{x})$ and the second term is the mean shift (i.e., the difference between the weighted mean and $\mathbf{x}$, the center of the window)

$$\mathbf{m}_\mathbf{x} = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{\mathbf{x}-\mathbf{x}_i}{h} \right\|^2 \right)} - \mathbf{x} \right]. \tag{5}$$

From (4) and (5), the following is obtained

$$\mathbf{m}_\mathbf{x} = \frac{1}{2} h^2 c \frac{\nabla \hat{p}(\mathbf{x})}{\hat{p}(\mathbf{x})}$$

which shows that the mean shift at $\mathbf{x}$ is proportional to the density gradient estimate. Therefore, $\mathbf{m}_\mathbf{x}$ always points in the gradient-increasing direction of $p(\mathbf{x})$. Thus $\mathbf{m}_\mathbf{x}$ can define a path leading to a stationary point (mode) of the estimated density. The mean shift procedure is obtained by successive computation of $\mathbf{m}_\mathbf{x}$ and translation of the window by $\mathbf{m}_\mathbf{x}$. An important property of this procedure which makes it unique from other gradient-based algorithms is that, when moving along such path, there is no need to specify the step size explicitly [7].

The fact that the mean shift procedure results in a walk along the direction of increasing gradient towards the nearest mode makes it an ideal tool for cluster analysis. For image segmentation, it was first introduced in [7] and soon became one of the most popular image segmentation techniques. There are two major steps involved: an application of the mean shift procedure on the image pixels to locate all convergence points, followed by a clustering step to merge all convergence points (and associated pixels) on the same basin of attraction into regions. Briefly, each image pixel is represented in the *joint* domain by a $(2+p)$-dimensional vector, which is the concatenation of its 2 spatial coordinates (in the *spatial* domain) with its p-dimensional color space representation (in the *range* domain, usually the 3-dimensional L*u*v* color space). In the first step, the window (a product of two

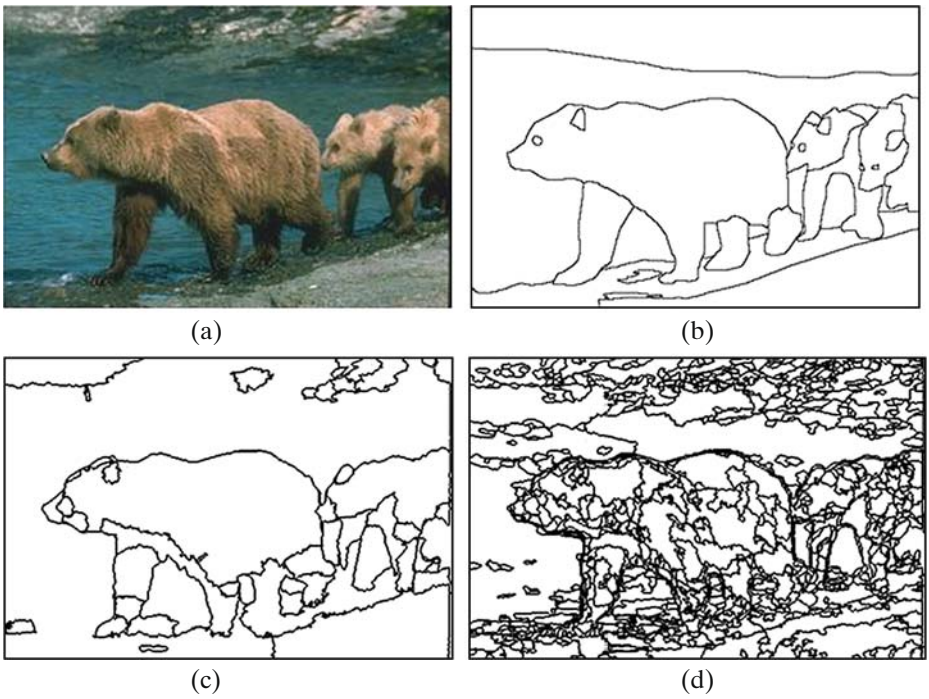


Fig. 5 **a** Original image; **b** human segmentation; **c** mean shift segmentation with $(h_s, h_r) = (12,12)$; **d** mean shift segmentation with $(h_s, h_r) = (5,5)$

kernels with window width parameters h_s and h_r) is initialized at each individual pixel location and moves in the direction of the maximum increase in the joint density gradient, until convergence. Thus, h_s and h_r are the employed kernel bandwidths. In the second step, convergence points that are closer than h_s in the spatial domain and h_r in the range domain are assigned to the same region. Finally, each pixel is assigned the region label of its corresponding convergence point.

The effectiveness of mean shift-based segmentation tightly depends on the selection of window width values for both the spatial (h_s) and range (h_r) domains. A selection of small values easily results in an excessive number of regions while a selection of large values might blur salient details or even incur an incorrect merging of irrelevant regions (See Fig. 5).

5 Proposed framework with mean shift-based segmentation

We have implemented a partial instantiation of the proposed framework (See Fig. 3) that uses mean shift clustering as the segmentation algorithm. We chose mean shift [7] because it is widely used in the vision community. This segmentation algorithm also has a relatively small number of adjustable parameters (i.e., only the scale (window width) parameters), which largely reduces the search space for optimal parameter settings. However, as previously discussed, the selection of scale

parameters (window width values) remains a difficult problem with strong impact on its performance. Thus, a good choice of values for those parameters is critical.

Initially, every image in the database is processed by the feature extraction component. In the mean shift-based segmentation algorithm [7], the joint domain representation of every pixel in the image is obtained by concatenating its 2 spatial coordinates to its p-dimensional color space representation. Thus, every pixel in the image is represented by a $(2+p)$-dimensional vector. However, we have observed that, in many cases, the color and spatial information are not sufficient for good segmentation performance. In this paper we alleviate this problem by first preprocessing the image to generate a "texture map", which we previously proposed in [26], based on a simple scanline technique which finds approximately repetitive patterns and is described next.

A scanline is defined as a sequence of pixels that a straight line (over an image and in a given direction) passes through. If we put this sequence of pixels into a Cartesian coordinate system with x-axis as indices and y-axis as one-channel intensity values, we obtain a two-dimensional signal (See Fig. 6). Let $S_\alpha = \{p_1, \ldots, p_n\}$ be a sequence of n scanned pixels with equal-interval indices starting from 1 along a certain direction α. Because each color channel will be analyzed and processed

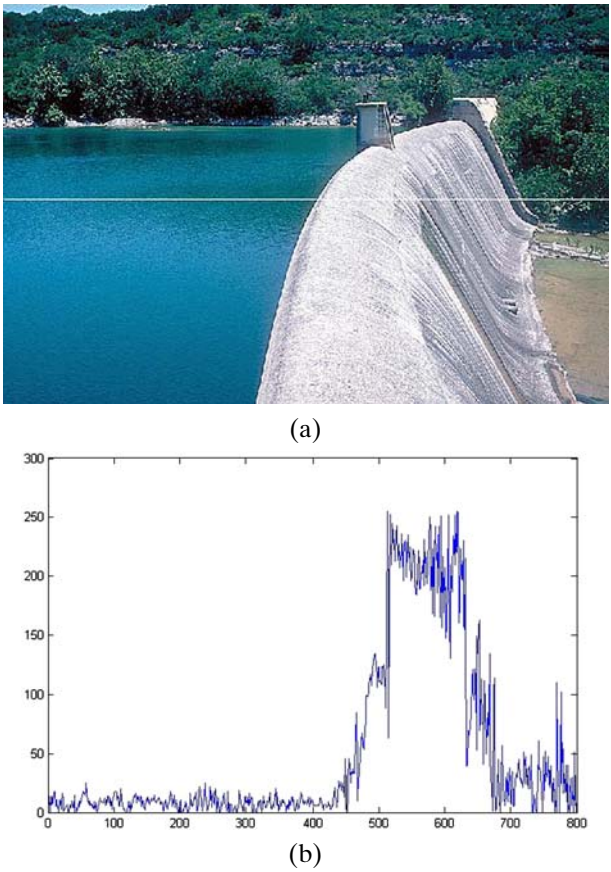


Fig. 6 **a** Original image with a *horizontal scanline* (in white color); **b** the pixel-intensity chart of this scanline on one color channel

individually, we simply use p_i to refer to the pixel intensity value for one of the color channels of the i^{th} pixel. We define the following measure as the *center of pattern* for a given scanline S_α

$$C = \frac{\sum_{i=1}^{n} p_i i}{\sum_{i=1}^{n} p_i} \tag{6}$$

This is analogous to the definition of the *center of mass* in physics.

Intuitively, the calculated C is an integer index indicating the balance point of S_α. Thus if S_α is a piece of a strictly repetitive pattern segment, the following equation will be satisfied with a small balance error e

$$C = \frac{n}{2} + e \tag{7}$$

In the case that $e \approx 0$, the pattern center of S_α is expected to approach the pixel with index $1 + n/2$. Now, if we relax the restriction to allow some uniformly distributed variety (noise) along the whole pattern segment, it is easy to observe that (7) still holds as long as the added variety does not break the overall balance. In that case, we call such a pattern an *approximately repetitive pattern*. In most cases the balance error e indicates the perfectness of repetition for a given pattern segment. Intuitively, a small absolute value for e means an approximately balanced pattern. In contrast, a large absolute value implies that the given pattern is not coherent everywhere, which suggests that the incoherence has to be trimmed off in order to retain its homogeneity (balance). Based on this observation, the following pruning rule is used for extracting a homogenous pattern from a given scanline: for a sequence of scanned pixels S_α, if C obtained by (6) cannot satisfy (7) with $e < \tau$ (τ is a predefined balance threshold), then we prune the scanline by e pixels either on the right side or left side of S_α. The balance error e can be simply calculated by

$$e = \frac{n}{2} - C \tag{8}$$

If e is positive, then e pixels on the most *left* side of S_α will be pruned off because some part on the *right* side (i.e., pixels with higher indices) of S_α is *denser*. For the same reason, a negative e leads to a pruning of $|e|$ ($|\cdot|$ as absolute operator) pixels on the most *right* side of S_α. The pruned scanline will go through the same procedure until a balance is finally achieved and thus a homogenous pattern $P_\alpha \subseteq S_\alpha$ is acquired. All pixels truncated off will be concatenated to form a new scanline and the same procedure will be repeated until all pixels are associated with patterns. Eventually, a scanline is divided into a set of consecutive non-overlapping patterns. This rule guarantees that during each iteration of pruning, only one side of a scanline will be trimmed off, which largely protects the integrity of texture patterns.

However, it is obvious that even if the balance threshold τ is restricted to zero, a satisfactory C from Eq. 7 does not guarantee the existence of an *approximately repetitive pattern* because there are cases (See Fig. 7) of patterns having their pattern centers approximately at the centers but not presenting any coherent texture patterns at all. Hence, this pruning rule only works for limited cases. A simple way to overcome the problem is to detect homogenous patterns in a concatenating manner. More precisely, the algorithm starts with the first l data points in a scanline, where the

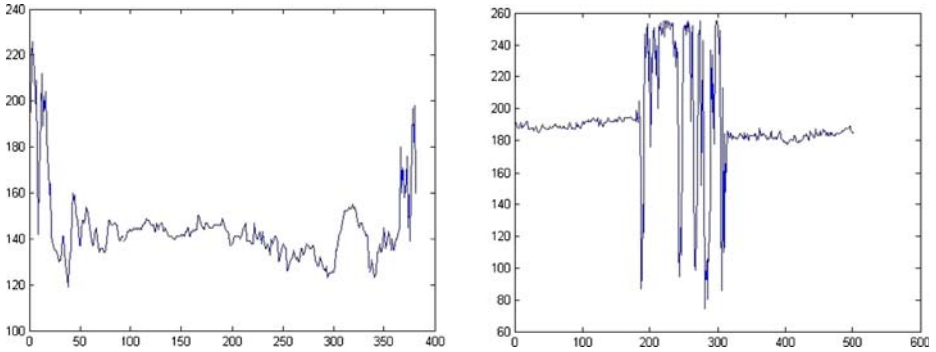


Fig. 7 Scanlines that have heterogeneous patterns inside but still with their pattern centers approximately at the center of scanlines: **a** a piece of scanline segment with dense areas at two sides; **b** a piece of scanline segment with dense area concentrated around the center

initial window length l should be selected large enough to capture texture patterns. If Eq. 7 over this l-point segment is satisfied, then it considers the next l data points in the scanline. If the center of pattern for these $2l$ data points still satisfies (7) with $e < \tau$, then we concatenate these two data sequences and repeat the same procedure until the balance is broken. Whenever the balance is not held, we simply cut half the current l (i.e., $l = 0.5l$) and continue with the same concatenating rule. Once l is reduced to zero, which probably implies a hit of the boundary between two distinct patterns, it means a texture pattern has been located from the starting position to the breaking point where l becomes zero. The algorithm then restarts from the last breaking point and keeps doing this until every data point is associated with a pattern. Finally, we obtain a set of consecutive nonoverlapping patterns from the input scanline. Figure 8 shows the results of applying the algorithm to a given scanline. Although the first pattern (from pixel 0 to 400) is divided into four segments, distinct patterns are successfully separated from others and, more importantly, all boundaries are preserved. For 2D images, due to the variety of texture patterns in different orientations, a multi-directional scanline operation surely generates a more

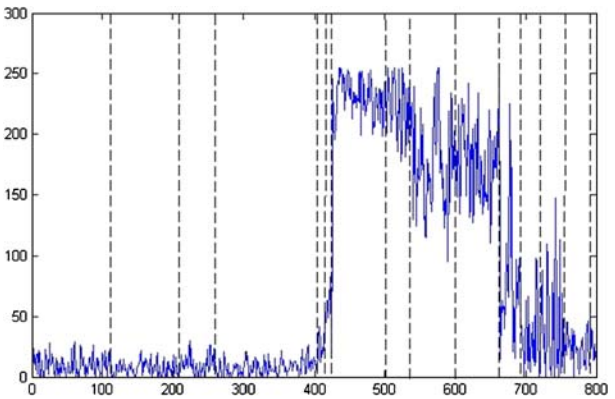


Fig. 8 A given scanline is divided into several pattern segments (separated by *dashed lines*)

informative representation than a single-directional one. In our experiments, a four-directional operation (namely 0, 45, 90 and 135 degrees) was employed.

For each image pixel, the 5-dimensional joint domain representation is obtained by concatenating its 2 spatial coordinates to its 3-dimensional color space representation in the L*u*v* color space. Then, for each of the 3 color channels, a "texture map" of the image is generated by replacing the original pixel intensity value with the average of the intensity values of all pixels on the same scanline (i.e., the value of each pixel on a particular color channel is replaced by the mean color of the texture pattern it belongs to). In this sense, the final "texture map" (on all 3 color channels) is actually a blurred version of the original image (See Fig. 9).

We have observed that, in many cases, mean shift-based segmentation has a better segmentation performance when run on this preprocessed version of the original image which, besides color and spatial information, also considers texture. In order to illustrate this we compared segmentation results based on these two image representations: the preprocessed texture-map-based version of the image and the original color-based representation. We used the mean shift-based segmentation algorithm [7] with different combinations of bandwidth parameters (e.g., $\{h_s = 3, h_r = 3\}$, $\{h_s = 4, h_r = 4\}$, and so on until $\{h_s = 10, h_r = 10\}$). The parameter set that resulted in the best performance for the mean shift-based segmentation algorithm [7] was chosen. Figure 10 shows a visual comparison between some of the segmentations generated with the original color-based representation and the derived texture-map-based representation. As we can observe, except for the images of 'oranges', all segmentations based on the original color-based representation were severely over-segmented due to the presence of complex texture while those based on the texture map representation did not have the same problem because texture had been smoothed.

The "texture map" of every image is then segmented by using mean shift-based segmentation [7]. As explained in Section 3, after the user submits a query image to the system through the user interface, a region-based image similarity measure is used to retrieve the closest images in the database and form the retrieval set. As is usual in RF-based CBIR systems, the user then labels each of the images in the retrieval set as either positive or negative. For a given query image, we assume that the user's decision to label an image in the retrieval set as positive is based on the

Fig. 9 Texture maps

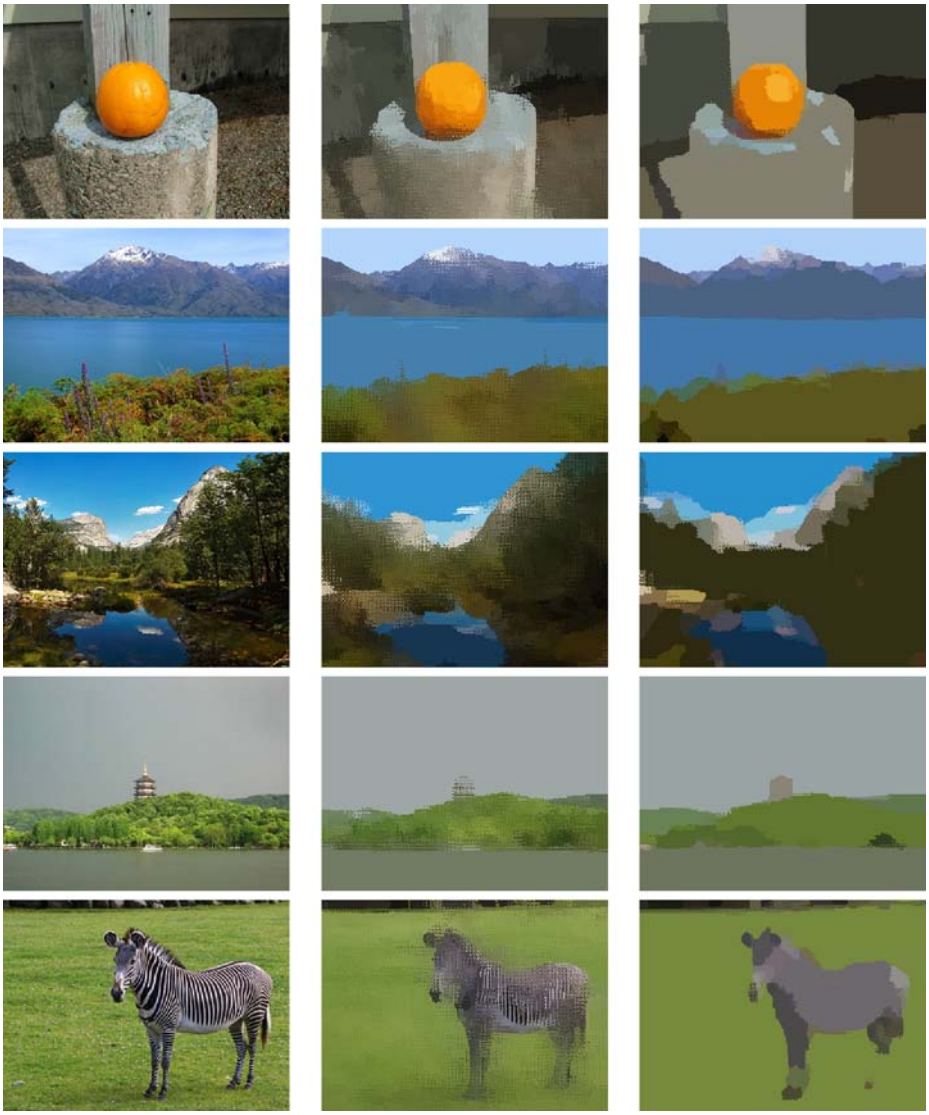


Fig. 10 Comparison between segmentation results based on color and texture map representations: **a** original images; **b** segmentations based on color information; **c** segmentations based on texture map. All segmentations are produced by the mean shift segmentation algorithm. It is easy to observe that segmentations based merely on color can barely deal with images containing complex texture patterns

presence of a particular object of interest (OOI) in the image. Similarly, a user labels an image in the retrieval set as negative if none of the objects in the image correlates with the user's semantic intent. As explained in Section 3, we use MIL [9] to identify the OOI and to find a prototypical feature vector **t** for it. We also find an estimate

of the maximum acceptable variability from the prototypical representation for the OOI, which we call the tolerance and denote by $v \in \Re$ (See Section 3 for details).

As explained in Section 4, in the first step of mean shift-based segmentation, the window is initialized at each pixel location and moves in the direction of the maximum increase in the joint density gradient, until convergence. Let $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots\}$, where $\mathbf{c}_i \in \Re^5$ is a convergence point, be the results obtained after the first step. Let the superscripts s and r in $\mathbf{c}_i^s$ and $\mathbf{c}_i^r$ denote the 2-dimensional spatial and the 3-dimensional range components of $\mathbf{c}_i$ respectively. In the second step of standard mean shift-based segmentation, convergence points that are closer than h_s in the spatial domain and h_r in the range domain receive the same region label. Instead, we use $\mathbf{t}$ and v to define an adaptive merging criteria for fragments of the OOI. This is equivalent to performing an adaptive selection of the scale (window width) parameters. Thus, implicitly, an object-specific adaptive adjustment of the (h_s, h_r) parameters is performed. Algorithm 2 summarizes this adaptive merging procedure.

Algorithm 2 Merging

Input: $\mathcal{C}$,h_r,h_s,$\mathbf{t}$,v
Output: Each convergence point $\mathbf{c}_i \in \mathcal{C}$ is assigned to a region

for each $\mathbf{c}_i \in \mathcal{C}$ **do**
  $\mathbf{c}_i.regionLabel \leftarrow 0$
  $\mathbf{c}_i.OOI \leftarrow FALSE$
end for
$regionLabel \leftarrow 1$
for each $\mathbf{c}_i \in \mathcal{C}$ **do**
  if $\left(\left\|\mathbf{c}_i^r - \mathbf{t}\right\|^2 \leq v\right)$ **then**
    $\mathbf{c}_i.OOI \leftarrow TRUE$ ▷ part of OOI
  end if
  if $(\mathbf{c}_i.regionLabel == 0)$ **then**
    $\mathbf{c}_i.regionLabel \leftarrow regionLabel$
    $regionLabel \leftarrow regionLabel + 1$
  end if
  for $\mathbf{c}_j \leftarrow \mathbf{c}_i.nextNeighbor(\mathbf{c}_i^s, h_s)$ **do**
    if $\left(\left\|\mathbf{c}_j^r - \mathbf{t}\right\|^2 \leq v\right)$ **then**
      $\mathbf{c}_j.OOI \leftarrow TRUE$ ▷ part of OOI
    end if
    if $(\mathbf{c}_i.OOI \;\&\; \mathbf{c}_j.OOI)$ **then**
      $\mathbf{c}_j.regionLabel \leftarrow \mathbf{c}_i.regionLabel$
    else if $(!\mathbf{c}_i.OOI \;\&\; !\mathbf{c}_j.OOI)$ **then**
      if $\left(\left\|\mathbf{c}_i^r - \mathbf{c}_j^r\right\|^2 \leq h_r\right)$ **then**
        $\mathbf{c}_j.regionLabel \leftarrow \mathbf{c}_i.regionLabel$
      end if
    end if
  end for
end for

In Line 7, $\mathbf{c}_i$ is assumed to be part of the OOI if its distance (in the range domain) to $\mathbf{t}$ is not larger than v. In Line 14, each $\mathbf{c}_j$ whose distance (in the spatial domain) to $\mathbf{c}_i$ is not larger than h_s is examined. In Line 15, it is determined whether $\mathbf{c}_j$ is part of the OOI. In Line 19, if both $\mathbf{c}_i$ and $\mathbf{c}_j$ are assumed to be part of the OOI, they receive the same region label. Otherwise, if neither $\mathbf{c}_i$ nor $\mathbf{c}_j$ are part of the OOI, the default fixed merging criteria is used in Line 21. In the final step (i.e., after Algorithm 2) each pixel is assigned the region label of its corresponding convergence point.

6 Experimental results

The objective of our experiments was to obtain evidence as to whether the proposed framework is indeed capable of learning more informed segmentation parameters. Note that it is difficult to perform meaningful performance comparisons with [1, 2, 20, 21], all of which require high-level knowledge (in the form of e.g., object models, samples of strong segmentations, hand-chosen parameters) to be provided in advance, which heavily restricts the application domain. Thus, in order to have a fair comparison, we compare the revised mean shift-based segmentations generated with our proposed framework with the initial segmentations generated by mean shift-based segmentation. We chose 5 random image categories. Each image category contains 30 to 40 positive images and an equal number of negative images. Hence, each image category simulates a possible initial retrieval set in response to a user's query (for images of that particular category) under the proposed CBIR framework.

The images in column (a) of Figs. 11, 12, 13, 14, 15 are sample positive images from each category ('oranges', 'forests', 'zebras', 'tigers', and 'pineapples' respectively). The images in column (b) are the corresponding ground-truth segmentations. The images in column (c) are the corresponding initial segmentations of those positive images in the database which, as explained in Section 5, is obtained by running standard mean shift-based segmentation on the "texture map" of each image. Different values for the (h_s, h_r) parameters were considered and the values that were determined to be the best (moderate merging of irrelevant regions and moderate over-segmentation), were chosen (and resulted in the segmentations in column (c)). The revised segmentations obtained with our proposed framework are shown in column (d).

An intuitive way of assessing the performance of an image segmentation algorithm is to compare the segmentations it produces against ground-truth (human-generated) segmentations of the same images. The ground-truth segmentations of the images (in column (b)) resulted in the semantic segmentation of the OOI (i.e., the entire orange, forest, zebra, tiger or pineapple) as a single region. It can be seen that each of the segmentations of the OOI in column (d) is "larger" (more of the OOI is included) than its counterpart in column (c). Thus, the revised segmentations agree more with the ground-truth segmentations.

Let $\mathcal{S} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots\}$, where $\mathcal{R}_i$ is the set of pixels in a region, be the segmentation of an image. The segmentation quality measure A is defined as

$$A = \max_{1 \le i \le |S|} \frac{|\mathcal{R}_i \cap \mathcal{G}|}{|\mathcal{G}|} \times \frac{|\mathcal{R}_i \cap \mathcal{G}|}{|\mathcal{R}_i|}$$

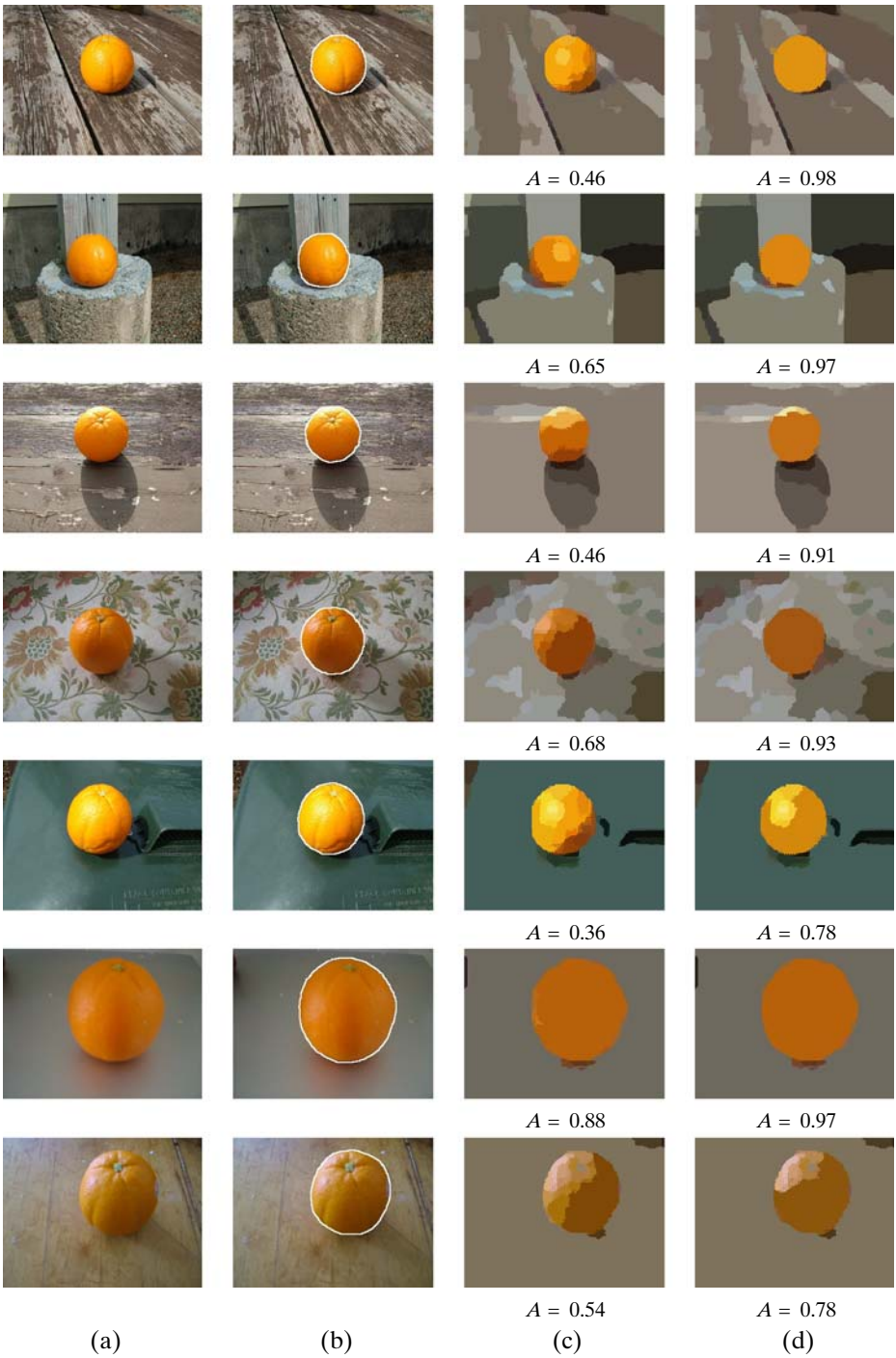


Fig. 11 Experiments on 'oranges': **a** Sample positive images in retrieval set; **b** Ground-truth segmentation; **c** Initial mean shift segmentation; (d) Revised mean shift segmentation

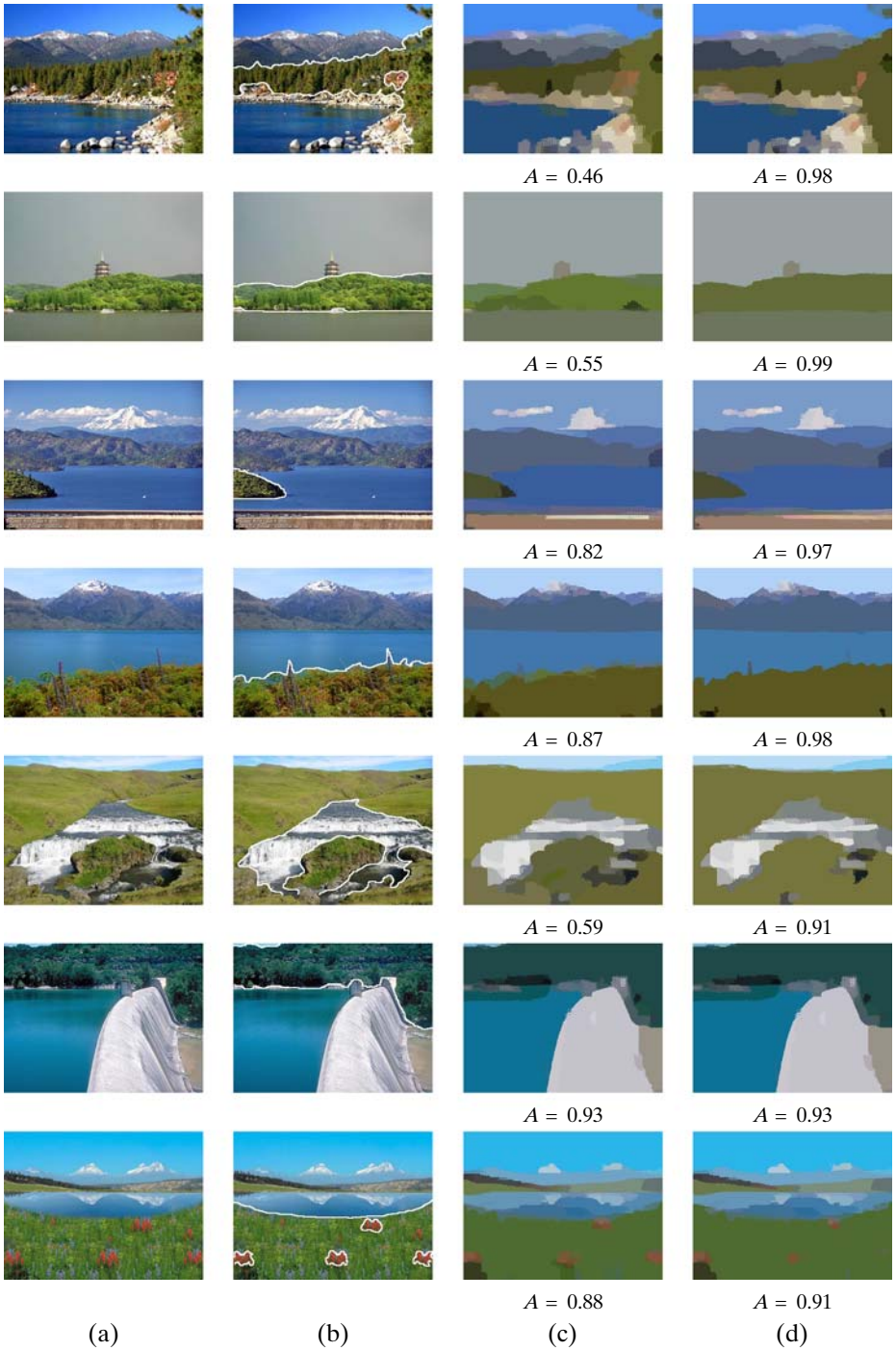


Fig. 12 Experiments on 'forests': **a** Sample positive images in retrieval set; **b** Ground-truth segmentation; **c** Initial mean shift segmentation; **d** Revised mean shift segmentation

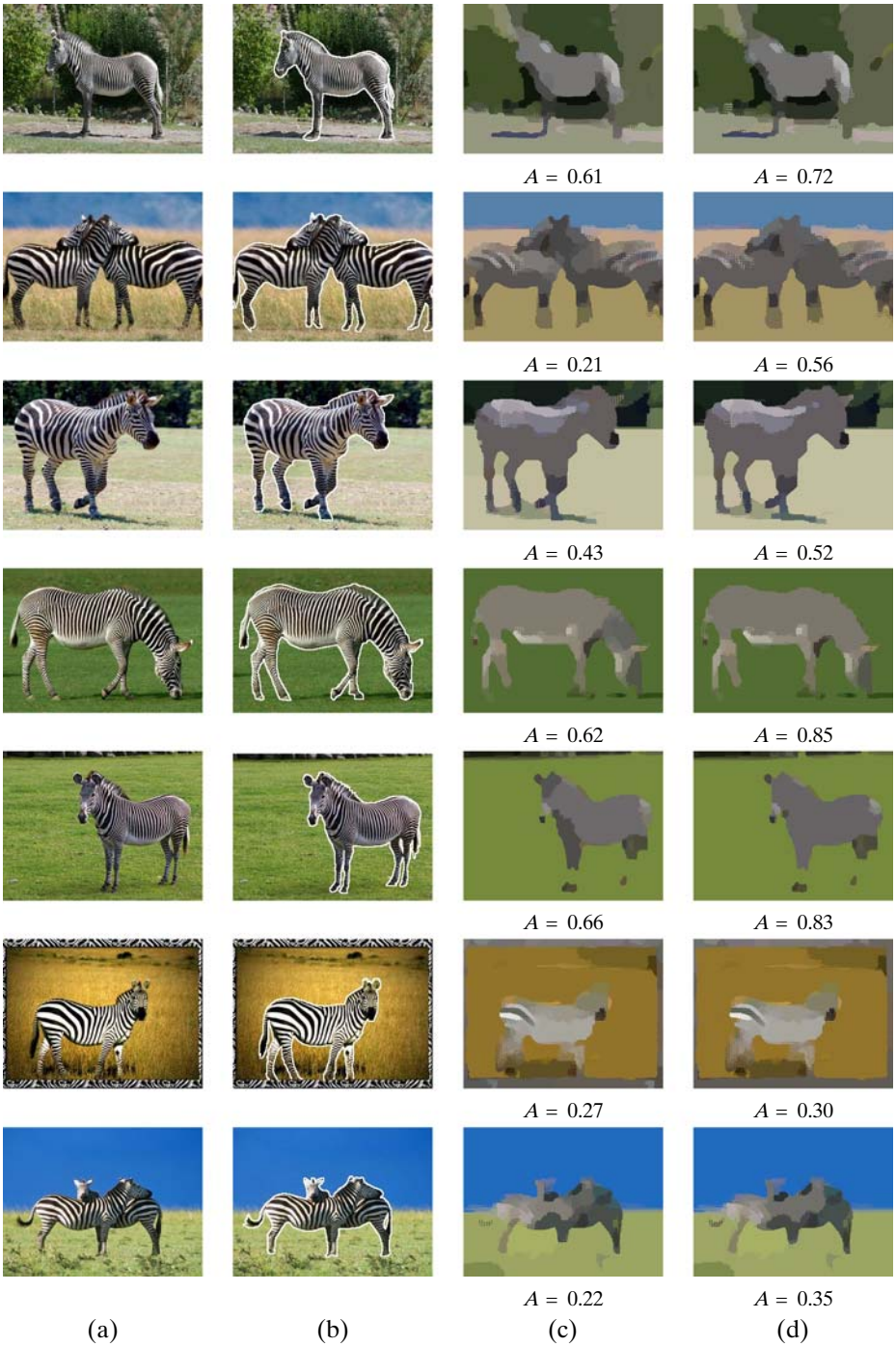


Fig. 13 Experiments on 'zebras': **a** Sample positive images in retrieval set; **b** Ground-truth segmentation; **c** Initial mean shift segmentation; **d** Revised mean shift segmentation

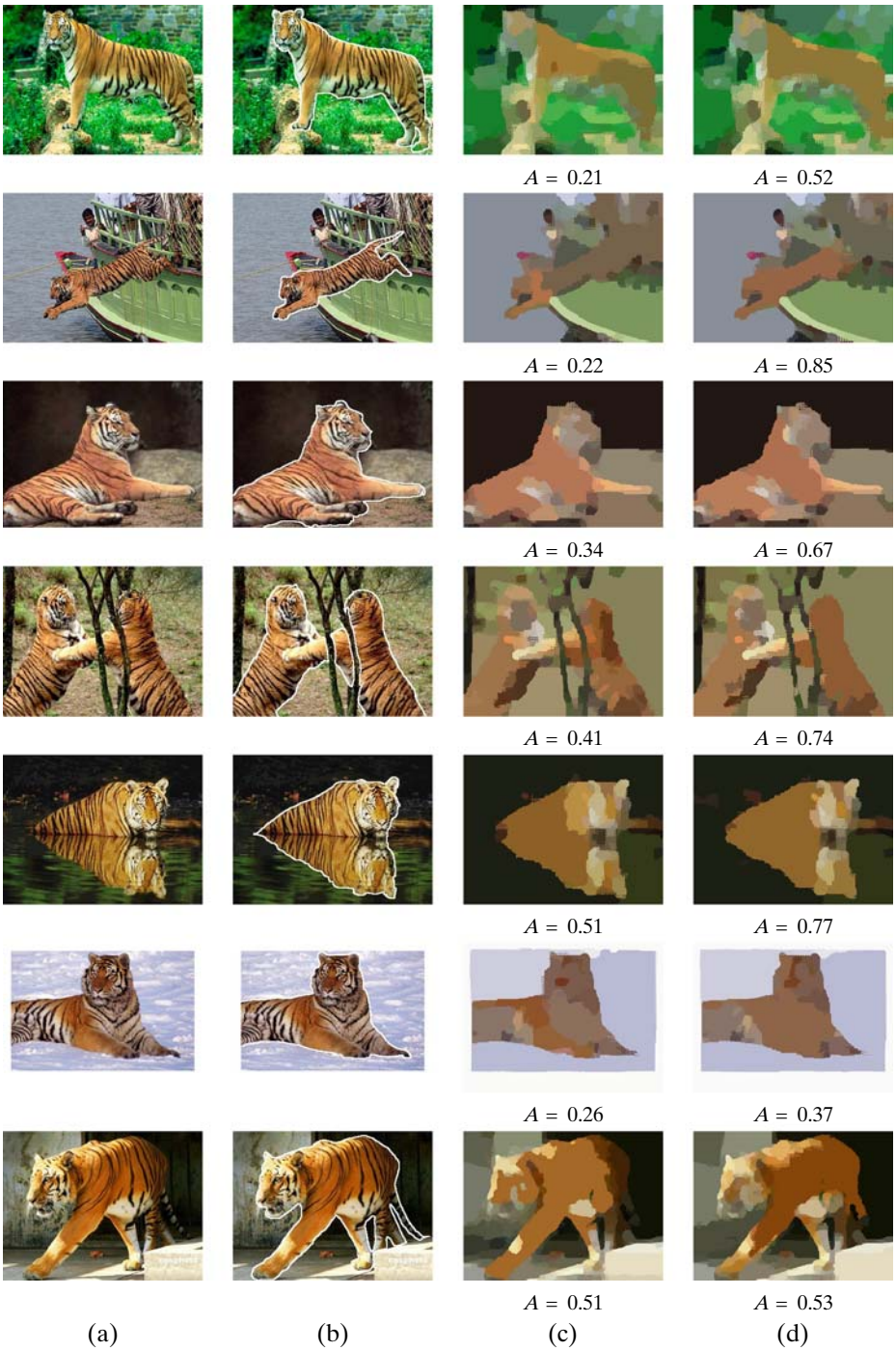
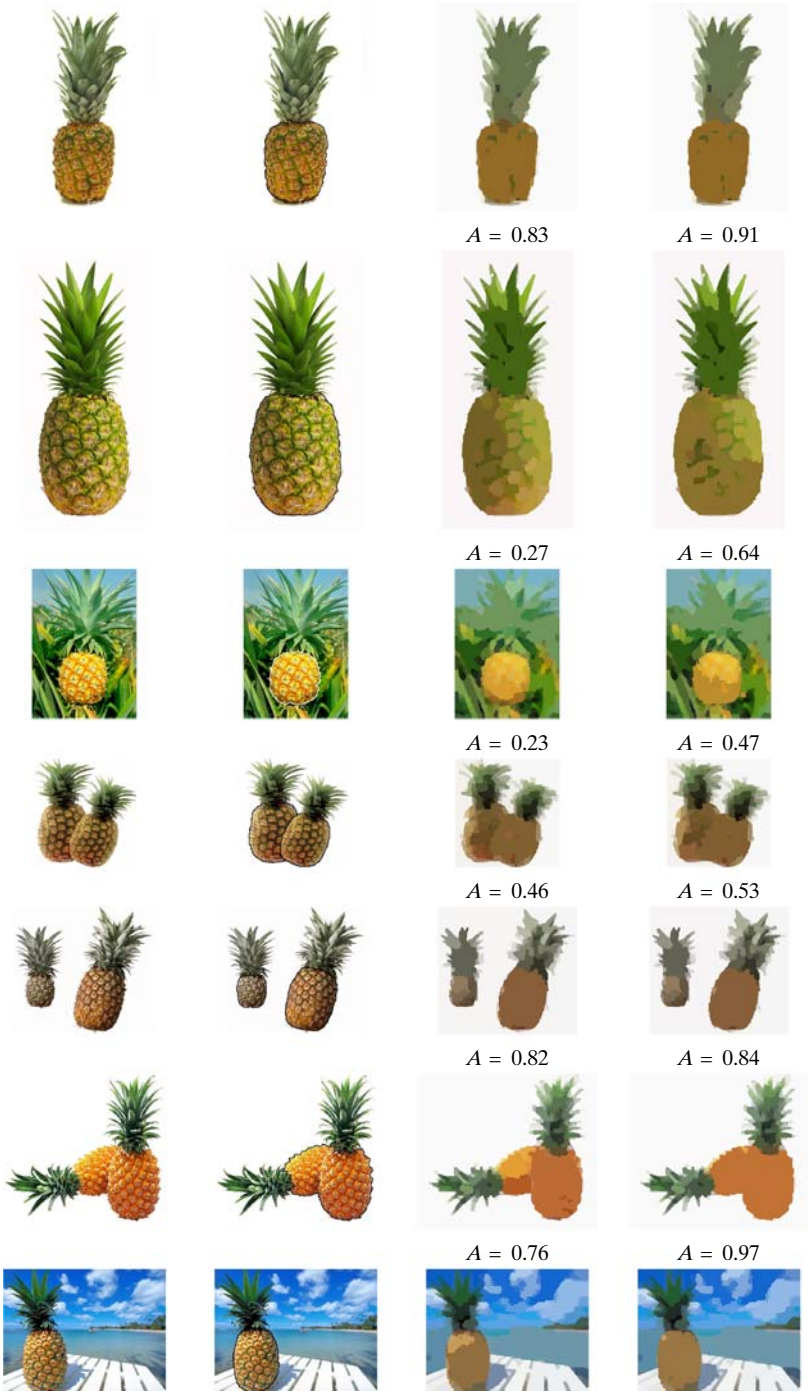


Fig. 14 Experiments on 'tigers': **a** Sample positive images in retrieval set; **b** Ground-truth segmentation; **c** Initial mean shift segmentation; **d** Revised mean shift segmentation


A = 0.83
A = 0.91
A = 0.27
A = 0.64
A = 0.23
A = 0.47
A = 0.46
A = 0.53
A = 0.82
A = 0.84
A = 0.76
A = 0.97
A = 0.74
A = 0.90
(a)
(b)
(c)
(d)

◀ **Fig. 15** Experiments on 'pineapples': **a** Sample positive images in retrieval set; **b** Ground-truth segmentation; **c** Initial mean shift segmentation; **d** Revised mean shift segmentation

where $\mathcal{G}$ is the set of pixels corresponding to the OOI in the ground truth segmentation. The measure A can be simply thought of as the percentage of pixels in agreement with the ground truth segmentation over the OOI. Thus $0 \leq A \leq 1$ is to be maximized.

As shown in Figs. 11–15, for each segmentation in column (c), the value of A is higher than the value of A for the corresponding segmentation in column (b). Notice that, for example, for the first image (first row) in Fig. 11, the brightness on the upper part of the orange results in this fragment being segmented as a different region with standard mean shift-based segmentation. On the other hand, since bright orange fragments also appear in other positive images, our approach learns that such fragments should be part of the orange.

The overall performance measures by averaging A for all 30–40 positive images in each category are given Table 1. As can be observed from the experimental results, the proposed framework is indeed capable of performing a more semantic image segmentation. This is because once (through the use of multiple instance learning) we obtain object-specific knowledge on the distribution of the OOI's visual appearance, this knowledge can be used to perform a semantically-guided context adaptation of segmentation parameters.

Notice that, as a side effect of processing a query, the initial segmentations (in column (c)) in the database are replaced with the revised segmentations (in column (d)). This will have a direct positive impact on the retrieval performance of future queries that involve the same OOI. Also, the re-segmentation is done offline (i.e., after processing the user's query) and is thus completely transparent to the user. Notice that, differently from other segmentation approaches that incorporate high-level knowledge, our proposed framework is much more flexible in the sense that it does not depend on the assumption that such knowledge (in the form of e.g., object models, samples of strong segmentations, hand-chosen parameters) is provided in advance, which heavily restricts the application domain. Instead, high-level knowledge (in the form of information on the OOI's visual appearance distribution) is learned as a by-product of users' interaction with the system.

Table 1 Overall segmentation performance for different image categories

Category	Average A for initial segmentations	Average A for revised segmentations
Oranges	0.72	0.91
Forests	0.65	0.88
Zebras	0.38	0.68
Tigers	0.34	0.60
Pineapples	0.57	0.76

7 Conclusions and future work

A content-based image retrieval (CBIR) framework that uses multiple instance learning (MIL) to achieve a more semantic image segmentation was presented. A segmentation algorithm is used to generate an initial segmentation of the images in the database. After processing a query, the user gives the usual relevance feedback (RF) by labeling each of the images in the retrieval set as positive or negative, based on whether or not it contains a particular object of interest (OOI). This feedback is then used in conjunction with MIL to obtain object-specific knowledge on the distribution of the object's visual appearance. This knowledge is then used to perform a semantically-guided context adaptation of segmentation parameters. The initial segmentation of the OOI in each of the positive images in the database is then revised. We presented a partial instantiation of this framework that uses mean shift clustering, which is widely used in the vision community, as the segmentation algorithm.

Experimental results on a variety of real images demonstrated that the proposed framework is indeed capable of performing a more semantic image segmentation. This is because once (through the use of MIL) we obtain object-specific knowledge on the distribution of the OOI's visual appearance, this knowledge can be used to perform a semantically-guided context adaptation of segmentation parameters.

The main disadvantage of current segmentation approaches that incorporate high-level knowledge is their lack of flexibility due to the assumption that such knowledge (in the form of e.g., object models, samples of strong segmentations, hand-chosen parameters) is provided in advance, which heavily restricts the application domain. In contrast to those methods, our approach is much more flexible in the sense that it does not require high-level knowledge to be provided in advance. Instead, it automatically learns such knowledge as a side effect of the user's interaction with the system.

In the context of CBIR, both short-term learning (based on RF supplied by the current user), and long-term learning (based on accumulated RF) have been used in the past. In this paper, we have only exploited short-term learning to perform a semantically-guided context adaptation of segmentation parameters. Our future work will focus on the possibility of also incorporating long-term learning to further improve on the quality and efficacy of the segmentation process.

References

1. Bhanu B, Lee S, Das S (1995) Adaptive image segmentation using genetic and hybrid search methods. IEEE Trans Aerosp Electron Syst 31(4):1268–1291
2. Bhanu B, Lee S, Ming J (1995) Adaptive image segmentation using a genetic algorithm. IEEE Trans Syst Man Cybern 25(12):1543–1567
3. Borenstein E, Ullman S (2004) Learning to segment. In: European conference on computer vision, pp 315–328
4. Carson C, Belongie S, Greenspan H, Malik J (2002) Blobworld: color and texture-based image segmentation using EM and its applications to image querying and classification. IEEE Trans Pattern Anal Mach Intell 24(8):1026–1038

5. Chen Y, Wang J (2002) A region-based fuzzy feature matching approach to content-based image retrieval. IEEE Trans Pattern Anal Mach Intell 24(9):1252–1267
6. Comaniciu D (2003) An algorithm for data-driven bandwith selection. IEEE Trans Pattern Anal Mach Intell 25(2):281–288
7. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
8. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2):1–60
9. Dietterich TG, Lathrop RH, Lozano-Pérez T (1997) Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 89(1–2):31–71
10. Duda RO, Hart PE, Stork DG (2000) Pattern classification, chapter 4, 2nd edn. Wiley, New York
11. Fan J, Yau DKY, Elmagarmid AK, Aref WG (2001) Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Trans Image Process 10(10):1454–1466
12. Gondra I, Xu T (2008) Adaptive mean shift-based image segmentation using multiple instance learning. In: Proceedings of IEEE international conference on digital information management, pp 716–721
13. Gu IYH, Gui V (2001) Colour image segmentation using adaptive mean shift filters. In: Proceedings of IEEE international conference on image processing, vol 1, pp 726–729
14. Guy G, Medioni G (1996) Inferring global perceptual contours from local features. Int J Comput Vis 20(1–2):113–133
15. Li J, Wang J, Wiederhold G (2000) IRM: integrated region matching for image retrieval. In: Proceedings of ACM conference on multimedia, pp 147–156
16. Maron O, Lozano-Pérez T (1998) A framework for multiple-instance learning. In: Jordan MI, Kearns MJ, Solla SA (eds) Advances in neural information processing systems, vol 10. MIT, Cambridge
17. McQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297
18. Pal NR, Pal SK (1993) A review of image segmentation techniques. Pattern Recogn 26(9):1277–1294
19. Pearl, J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco
20. Peng J, Bhanu B (1998) Closed-loop object recognition using reinforcement learning. IEEE Trans Pattern Anal Mach Intell 20(2):139–154
21. Schnitman Y, Caspi Y, Cohen-Or D, Lischinski D (2006) Inducing semantic segmentations from an example. In: Asian conference on computer vision, pp 384–393
22. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
23. Smeulders AW, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380
24. Tu Z, Zhu S.C (2002) Image segmentation by data-driven Markov chain and Monte Carlo. IEEE Trans Pattern Anal Mach Intell 24(5):657–673
25. Wang Y, Yang J, Peng N (2006) Unsupervised color-texture segmentation based on soft criterion with adaptive mean-shift clustering. Pattern Recogn Lett 27:386–392
26. Xu T, Gondra I (2009) Texture map: an effective representation for image segmentation. In: Proceedings of the C3S2E conference. ACM, New York, pp 197–203
27. Zhou XS, Huang TS (2003) Relevance feedback in image retrieval: A comprehensive review. Multimedia Syst 8(6):536–544

Iker Gondra is currently a faculty member at the Department of Mathematics, Statistics, and Computer Science, St. Francis Xavier University, Nova Scotia, Canada. He completed his PhD degree in Computer Science at Oklahoma State University, Oklahoma, USA in 2005. The main focus of his research has been on the application of machine learning to content-based image retrieval and image segmentation.

Tao Xu recently completed his M.Sc. degree in Computer Science at the Department of Mathematics, Statistics, and Computer Science, St. Francis Xavier University, Nova Scotia, Canada. Before the beginning of his graduate studies, he worked in the computer industry for several years. His research interests include machine learning and theoretical computer science.