Extracting Texels in 2.1D Natural Textures

Narendra Ahuja and Sinisa Todorovic
Beckman Institute, University of Illinois at Urbana-Champaign
{ahuja, sintod}@vision.ai.uiuc.edu

Abstract

This paper proposes the problem of unsupervised extraction of texture elements, called texels, which repeatedly occur in the image of a frontally viewed, homogeneous, 2.1D, planar texture, and presents a solution. 2.1D texture here means that the physical texels are thin objects lying along a surface that may partially occlude one another. The image texture is represented by the segmentation tree whose structure captures the recursive embedding of regions obtained from a multiscale image segmentation. In the segmentation tree, the texels appear as subtrees with similar structure, with nodes having similar photometric and geometric properties. A new learning algorithm is proposed for fusing these similar subtrees into a tree-union, which registers all visible texel parts, and thus represents a statistical, generative model of the complete (unoccluded) texel. The learning algorithm involves concurrent estimation of texel tree structure, as well as the probability distributions of its node properties. Texel detection and segmentation are achieved simultaneously by matching the segmentation tree of a new image with the texel model. Experiments conducted on a newly compiled dataset containing 2.1D natural textures demonstrate the validity of our approach.

1. Introduction

This paper is about (1) identifying the basic elements of image texture, called texture elements, or texels, (2) obtaining the texel model, and (3) texel segmentation (delineation of image regions occupied by texels) by using the texel model. Our approach derives from and closely follows the fundamental notion of image texture – namely, that image texture is formed by spatial repetition of a large number of texels, which are in turn the images of a large field of spatially recurring physical texture elements in the scene.

Randomness, which is a defining feature of image texture, is the result of two physical, stochastic processes. First, the intrinsic properties of the physical texture elements are not strictly identical; they are only statistically similar as if they are samples drawn from a certain probability density function (pdf). Second, the placement of the physical elements in the scene is, in general, not strictly periodic but only statistically uniform. For example, in Fig. 1, the kiwi slices have statistically similar size, shape and color, and their placement is statistically uniform. Therefore, the models of intrinsic properties and spatial placement of image texels need to be stochastic. Another important characteristic of physical elements is their dimensionality. They may be painted patterns on the texture surface defining a 2D texture, or 3D objects extending out of the surface defining a 3D texture. As a third, hybrid case, the physical texels may be very thin patches lying along the surface, as in Fig. 1. The nearly zero thickness of these patches makes them close to 2D texels, but their ability to overlay and occlude one another lends them the 3D character; we refer to this hybrid case as 2.1D textures as in [15].

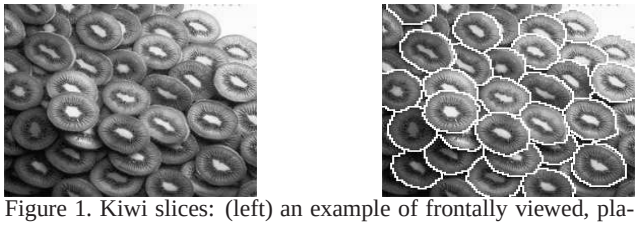

Figure 1. Kiwi slices: (left) an example of frontally viewed, planar 2.1D natural texture; (right) contours of extracted texels by our algorithm. Challenges to texel extraction: texels may partially occlude one another; texel subregions (white kiwi cores) may define a different texture, at a finer resolution, and they may be confused with the texels; occlusion and changes in illumination across the image may reduce the gray-level contrast between texels and subtexels, making their learning and segmentation difficult.

In this paper, we consider nearly planar 2.1D textures, imaged from a viewing direction which is nearly along surface normal.[1] For brevity, we will refer to them only as 2.1D textures. The image texels thus represent physical texture elements that are at a constant depth, and imaged fully

[1]The frontal viewing assumption helps us focus here on more fundamental, novel aspects of the proposed formulation aimed at more general challenges of 2.1D textures listed in the caption of Fig. 1. The frontal texture case is itself broadly relevant as is evident from most past work on texture classification, synthesis and segmentation which is focused on frontally viewed textures.

or partly depending on their mutual occlusions. Since the physical elements are themselves finite size objects, the image texels are regions, whose properties – such as geometric (e.g., area and shape), photometric (e.g., spatial color distribution), and topological (recursive embeddings of subtexel regions) properties – can be stochastically characterized. Given an image of 2.1D texture, this paper is aimed at estimating the pdf of these texel properties. Since the notion of image texture implies a large number of texels, and therefore a large number of samples from the underlying pdf's, reliable statistical estimation of these pdf's is feasible. Estimation of the pdf of intrinsic texel properties can be combined with the analysis of other aspects of texture, including properties of the textured surface (e.g., plane orientation), and stochastic rules governing the placement of physical elements on the plane, to estimate the complete texture model in more general (e.g., nonfrontal) cases. However, properties of texture surface and texel placement are outside the scope of this paper. In the rest of this section, we first review prior work, and then present an overview and main contributions of our approach.

Prior Work: Texture Modeling – Most approaches neglect to directly account for one or both basic components of image texture – namely, the intrinsic and placement characteristics of the texels. Statistical (pixel-based) approaches [2] model the joint statistical properties of features extracted at pixel locations, missing to encode the feature prior groupings imposed by texels [25]. Region based (structural) models [6] are more realistic in that they treat texture as a layout of regions, as called for by structure of the physical texture [3, 20, 14]. However, with few exceptions [3], the existing region-based models do not distinguish between regions within and across texels. For example, in [5], salient segments are extracted at multiple scales to capture photometric, geometric and structural properties of the entire texture; however, instead of identifying and modeling the basic repetitive unit of texture, every extracted segment is treated as the texture element. Many approaches exploit texture regions only implicitly, by using measures of different edge types, or of regions of different shapes, orientations or sizes, obtained by, e.g., Harris, Kadir-Brady, SIFT, or blob detectors [23, 11, 9, 13, 7]. Julesz and his collaborators [8] used a special class of features called textons (e.g., closure, linearity, end terminations, etc.) as primitives of model description, and demonstrated that the performance of the resulting models better predicted human texture perception. There are several attempts to mathematically define the notion of textons for the purpose of texture modeling. For example, in [24], texture is modeled as a superposition of Gabor base functions, which are in turn generated by a user-specified vocabulary of textons (e.g., star-shaped or T-junction templates). Finally, motivated by psychophysical research that human brain decomposes texture into its frequency and orientation subcomponents, many studies use signal processing techniques for texture modeling [10, 4, 22].

Prior Work: Texel Extraction – Much work on texel extraction mirrors the limitations of texture modeling. For example, a highly restrictive assumption is made in [23] that texels are homogeneous, structureless blobs, which are darker or lighter than the background. In [3], a texel is represented as a union of disc-shaped regions found by a multiscale blob detector. In recent work, texels are characterized by the occurrence of feature points associated with them, or by correlations of image windows, neither of which delineates the texels. Sometimes, edge fragments are used to locate the vicinity of a texel. For instance, in [16, 21], texels are represented by Harris corners and homogeneous-intensity regions enclosed by Canny edges, while in [7], by MSER points and normalized-cross-correlation patches. In [13], texels are characterized by SIFT descriptors. In [12], the user is required to provide a parallelogram-shaped texel template for detecting similar texels in a given texture. None of the above methods precisely segments the texels.

Motivation: We use regions as features for texel modeling and extraction. Regions offer several advantages over lower-dimensional, local features, such as interest points or edge fragments. Regions could precisely delineate the texel's contours, not just its vicinity. The higher dimensionality of regions makes them richer descriptors of texel's geometric and photometric properties, and their detection more stable to small illumination and viewpoint changes. Unlike local features, regions facilitate capturing within-texel structure and spatial context of texels.

Overview of Our Approach: (1) We first obtain a multiscale segmentation of the texture image. The image is then represented by a segmentation tree, whose nodes encode the geometric and photometric properties of the image regions, and whose structure captures their recursive containment. The spatial layout of the regions can be easily derived from the segmentation tree. Each texel appears as a finite-size subtree in the segmentation tree, because a large number of texels, required to comprise a texture image, limits the texel size to a fraction of the image size, and also the recursive subregion embedding visible within a texel is lower bounded by the pixel size. (2) Since texture is characterized by the presence of a large number of texels, the segmentation tree must contain many subtrees having similar node and structural properties. To identify texels, therefore, we find the largest group of the largest subtrees across the image that match. (3) Some matching subtrees may represent different parts of only partly visible texels. All such subtrees must clearly be parts of the unknown subtree representing the entire texel. The texel subtree is constructed by fusing (registering and mosaicing) all matched, partial as well as full, texel subtrees into a tree-union. A pdf is estimated that captures statistical properties of the

matching sets of nodes in the tree-union, which ultimately serves as the generative model of the texel. Fig. 2 illustrates Steps 2 and 3. (4) Given a new image of the same texture, texels are extracted by matching the learned texel model with the image's segmentation tree. Matches found locate texels, and being union of regions, they simultaneously delineate their exact boundaries.

Towards the implementation of these steps, we propose a new learning algorithm for estimating the texel generative model. Our algorithm involves simultaneous estimation of both model structure and model pdf, which is commonly viewed as the most challenging problem of statistical generative modeling. We formulate it as an optimization problem to find the optimal model structure and pdf for which the description length of the data is minimum (MDL). This is accomplished in an iterative procedure that alternates between the following two steps: (1) the Expectation-Maximization (EM), where the model pdf is estimated for a given tree-union structure, and (2) tree matching, where the tree-union is estimated for the previously computed model pdf.

Contributions: (1) We employ a generative model of arbitrarily structured texels comprising a 2.1D texture. The model encodes the random geometric, photometric and topological properties of texels. (2) A new learning algorithm is proposed for simultaneous estimation of both structure and property-pdfs defining the texel model. (3) We estimate structural, geometric, and photometric properties of the complete (unoccluded) texel from partially visible texel occurrences in a 2.1D texture without any supervision. (4) Experimental validation is conducted on a new dataset consisting of 2.1D natural textures, which we have compiled to exercise various parts of the proposed algorithms. This dataset complements the existing benchmark sets (e.g., Brodatz, CUReT, CMU Near-regular Textures, etc.) since most of them do not contain 2.1D textures, or have only 2.1D textures with (near-)regular placement of texels. (5) Most prior work replaces texel extraction with the simpler problem of locating a point or points associated with each texel. Therefore, most existing methods cannot precisely delineate the texel boundaries. To the best of our knowledge, this paper presents the first formulation and solution to unsupervised learning and segmentation of texels in 2.1D textures.

Next we describe our image representation, in Sec. 2, followed by texel detection in Sec. 3. The texel generative model is defined in Sec. 4, while learning of the model structure and pdf's is described in Sec. 5. Experimental validation is discussed in Sec. 6.

2. Region Properties for Texel Modeling

An input image is represented by the segmentation tree, T, obtained using the multiscale segmentation algorithm of [1], which parses the image into homogeneous regions, at all degrees of homogeneity present, regardless of the regions' contrast, size, shape and location. Nodes at upper levels correspond to larger regions, while their children nodes capture embedded, smaller details. Any cutset of the tree corresponds to one possible image segmentation, while parent-child relationships capture recursive region embedding. The number of nodes (150–200),[2] branching factor (0–5), and the number of levels (10–15) in different parts of the tree are image dependent . A vector, $\boldsymbol{y}_{ij}$, of region properties is associated with node i. Many of the properties of node i are expressed relative to those of i's parent-region j, to allow rotation and scale invariance in texel detection. These properties are as follows: (1) gray-level contrast g_{ij}; (2) area ratio $a_{ij} \triangleq A_i/A_j$, where A_i and A_j are the areas of i and j; (3) displacement between the centroid locations (x, y) of i and j, $\overrightarrow{\Delta}_{ij} \triangleq \frac{1}{\sqrt{A_j}}[(\overrightarrow{x}_j - \overrightarrow{x}_i) + (\overrightarrow{y}_j - \overrightarrow{y}_i)]$; (4) area dispersion of i over its children $k \in C(i)$, $AD_i \triangleq \sum_{k \in C(i)} (a_k - \text{mean}(\{a_k\}))^2$; (5) shape context histogram $\boldsymbol{h}_i = \{h_i(z)\}_{z=1 \ldots Z}$, computed by parsing the image into $Z=40$ pie slices, each of which begins at the centroid of i, and subtends the same angle $2\pi/Z$, and by counting the total number of pixels of region i that fall in pie slice z; the slice having the largest histogram value is is said to contain the estimate of the major axis of region i; (6) tilt angle α_{ij} between major axes of i and j. In summary, given that j is the parent of i, the vector associated with node i is $\boldsymbol{y}_{ij} = [g_{ij}, a_{ij}, \overrightarrow{\Delta}_{ij}, AD_i, \alpha_{ij}, \boldsymbol{h}_i]^{\mathrm{T}}$.

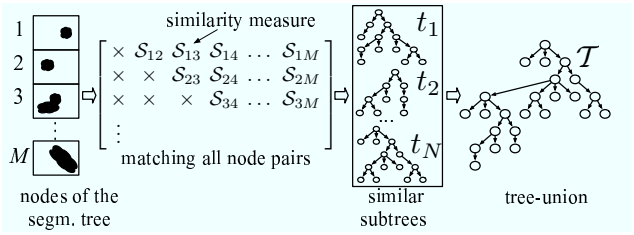

Figure 2. An input image is represented by the segmentation tree, and then all pairs of its M nodes are matched. Frequently occurring, similar subtrees are viewed as candidate texels, which are then fused into the tree-union, representing the texel model.

3. Texel Detection

The segmentation tree T contains all segments present in the image, including texels. They can be detected as a set of disjoint, similar subtrees rooted closest to the root of T. Therefore, any smaller, similar subtrees, corresponding to the repetitive subtexel structure, are not detected separately, since they are contained within the identified parent texel subtrees. The texel subtrees are discovered by matching all possible node pairs in the segmentation tree, and selecting those node pairs closest to the root whose attributes match,

[2] The values in parentheses are obtained for 2.1D textures considered in Sec 6.

and the same holds for their descendant nodes. For matching, we use the well-known graph matching algorithm of [18, 17]. For each pair of nodes $(i,i')\in T\times T$, $i\neq i'$, the algorithm matches nodes in the subtrees rooted at i and i' so as to maximize the total similarity value $\mathcal{S}_{ii'}$ computed over the subtrees. Formally, given that the matching algorithm has selected a bijection $f=\{(i_0,i'_0),(i_1,i'_1),\ldots,(i_n,i'_n)\}$, where the descendant nodes $i_1,\ldots,i_n$ of $i=i_0$ are paired with the descendant nodes $i'_1,\ldots,i'_n$ of $i'=i'_0$, the similarity $\mathcal{S}_{ii'}$ between the subtrees is defined as

$$\mathcal{S}_{ii'} \triangleq \mathcal{S}(f) = \textstyle\sum_{(i,i')\in f}(r_i+r_{i'}-m_{ii'}) \ , \qquad (1)$$

where r_i is the saliency of node i, and $m_{ii'}$ is the cost of matching i and i' (both also called edit-costs), both defined in terms of region properties $\boldsymbol{y}_{ij}$, as explained in Sec. 5. The complexity of computing $\mathcal{S}_{jj'}$ is $O(n^2)$ where n is the number of descendant nodes under j and j' [18, 17].

After finding the similarity measures $\mathcal{S}_{ii'}$, $\forall(i,i')\in T\times T$, $i\neq i'$, we analyze $\mathcal{S}_{ii'}$ values to identify texel candidates. From (1), $\mathcal{S}_{ii'}$ values become larger for nodes i and i' closer to the root of the segmentation tree (i.e., for larger regions in the image). Consequently, the highest $\mathcal{S}$ values may not correspond to the similarity measure between texel subtrees, but between their groupings into supertexels. Since supertexels occupy larger image areas than texels, the frequency of their occurrence is significantly smaller than that of texels. Therefore, texel detection amounts to finding a cluster of $\mathcal{S}$ values that are both large and frequent among the similarity measures obtained. To this end, we analyze the modes and valleys of the frequency histogram of $\mathcal{S}$ values, $\mathcal{H}(\mathcal{S})$, and select as texel candidates all subtree pairs whose $\mathcal{S}$ values fall in the mode with the largest product of $\mathcal{S}$ values and their frequencies, $\text{texel\_mode} \triangleq \arg \max_{\text{mode}\in\text{histogram\_modes}} \sum_{\mathcal{S}\in\text{mode}} \mathcal{S}\cdot\mathcal{H}(\mathcal{S})$.

4. Specification of the Texel Generative Model

The set of detected texel candidates, $\mathbb{D}=\{t_1,t_2,...,t_N\}$, may contain entire as well as partial views of the texels. To fuse all these views into a compact model of the entire, unoccluded texel, we find the union $\mathcal{T}$ of the subtrees in $\mathbb{D}$. The tree-union is the smallest directed acyclic graph that contains every tree in a given set. Since the texel structure and region properties are stochastic, we characterize $\mathcal{T}$ with a pdf, as defined below.

Our generative model, depicted in Fig. 3, represents an acyclic Bayesian network. Each node i in $\mathcal{T}$ is associated with a hidden random variable x_i taking discrete values in $\{1,\ldots,K\}$, where K is the input parameter. x_i is assumed statistically dependent on i's parent x_j, which is encoded by the Markov chain of transition probabilities $P(x_i|x_j)$, and the root prior $P(x_1)$. The variability of tree-union structure is captured by the observable random variable N_i, representing the number of children of node i, which is generated by x_i according to $P(N_i|x_i)\triangleq\lambda_{x_i}\mathrm{e}^{-\lambda_{x_i}N_i}$. The vector of region properties $\boldsymbol{y}_{ij}$ associated with node i, defined in Sec. 2, is generated by x_i and x_j according to the Gaussian distribution, $P(\boldsymbol{y}_{ij}|x_i,x_j)\triangleq\mathcal{N}(\boldsymbol{y}_{ij};\boldsymbol{\mu}_{x_ix_j},\Sigma_{x_ix_j})$. The set of model parameters, Ω, is assumed to be identical for all nodes i in $\mathcal{T}$: $\Omega\triangleq\{P(x_1),P(x_i|x_j),\boldsymbol{\mu}_{x_ix_j},\Sigma_{x_ix_j},\lambda_{x_i}\}$. Then, the joint likelihood of $\mathcal{T}$ is given by

$$\begin{aligned}P(\mathcal{T}|\Omega)=&P(x_1)\textstyle\prod_{j\in\mathcal{T}}\lambda_{x_j}\exp(-\lambda_{x_j}N_j)\\ &\cdot\textstyle\prod_{i=1}^{N_j}P(x_i|x_j)\mathcal{N}(\boldsymbol{y}_{ij};\boldsymbol{\mu}_{x_ix_j},\Sigma_{x_ix_j}).\end{aligned} \qquad (2)$$

5. Learning Model Structure and Parameters

Learning the model structure $\mathcal{T}$ and parameters Ω is conducted simultaneously in an iterative procedure, where for a given model structure we estimate the model parameters, and then use these parameters to re-estimate the model structure. This iterative learning is guided by the minimum description length (MDL) principle. Related to ours is the approach to learning a mixture of tree-unions from a given set of trees [19]. The likelihood of their mixture model is defined as a product of the sampling probabilities of nodes in the trees from the set, since these nodes are assumed sampled from the model as Bernoulli trials. Unlike [19], we explicitly model the transition (Markovian) probabilities between parent-child node pairs in $\mathcal{T}$, and thus enforce a correct sampling of node hierarchy from our generative model. In the sequel, we first explain how to estimate model parameters for a given model structure, then discuss how to learn the tree-union structure by using the parameter estimates, and finally present the entire learning algorithm.

Learning Model Parameters: Given the model structure (i.e., tree-union), which is a directed acyclic graph, Ω can be learned using the EM algorithm, in conjunction with the belief propagation. The EM algorithm consists of the E-step and M-step that are iterated alternatively until the objective function, Q, reaches convergence.

Let $\boldsymbol{N}=\{N_i\}$ and $\boldsymbol{Y}=\{\boldsymbol{y}_{ij}\}$ denote all observables, and $\boldsymbol{X}=\{x_i\}$, all hidden variables of $\mathcal{T}$. In the E-step, the expectation of the joint log-likelihood $Q(\Omega|\Omega^{(\tau)})=\mathbb{E}\left[\log P(\boldsymbol{X},\boldsymbol{N},\boldsymbol{Y}|\Omega)|\boldsymbol{N},\boldsymbol{Y},\Omega^{(\tau)}\right]$ is computed. From (2), we have

$$\begin{aligned}Q(\Omega|\Omega^{(\tau)})=&P(x_1|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})\log P(x_1)\\ &+\textstyle\sum_j P(x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})[\log\lambda_{x_j}-\lambda_{x_j}N_j]\\ &+\textstyle\sum_{i,j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})\log P(x_i|x_j)\\ &+\textstyle\sum_{i,j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})\log\mathcal{N}(\boldsymbol{y}_{ij};\cdot).\end{aligned} \qquad (3)$$

To obtain posterior marginals $P(x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})$ and $P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})$, appearing in (3), we use the belief propagation algorithm presented in Appendix. After the posterior marginals are computed, they are plugged in (3), which concludes the E-step.

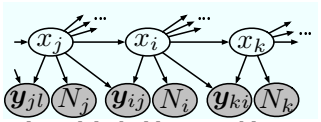

Figure 3. The texel model: hidden variables x_i and x_j form the Markov chain, and generate observables: region properties $\boldsymbol{y}_{ij}$, and the branching factor N_i.

In the M-step, we update $\Omega^{(\tau+1)}=\arg\max_{\Omega} Q(\Omega|\Omega^{(\tau)})$. By setting $\partial Q(\Omega|\Omega^{(\tau)})/\partial\Omega=0$ and accounting for the Lagrange multipliers, we obtain the following update rules:

$$\begin{aligned}
P^{(\tau+1)}(x_1) &= P(x_1|\boldsymbol{Y},\boldsymbol{N}),\\
P^{(\tau+1)}(x_i|x_j) &= \tfrac{1}{n_{ij}}\textstyle\sum_{i,j}\frac{P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})}{P(x_j|\boldsymbol{Y},\boldsymbol{N})},\\
\boldsymbol{\mu}_{x_ix_j}^{(\tau+1)} &= \tfrac{\sum_{i,j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})\boldsymbol{y}_{ij}}{\sum_{i,j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})},\\
\Sigma_{x_ix_j}^{(\tau+1)} &= \tfrac{\sum_{i,j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})(\boldsymbol{y}_{ij}-\boldsymbol{\mu}_{x_ix_j}^{(\tau+1)})(\boldsymbol{y}_{ij}-\boldsymbol{\mu}_{x_ix_j}^{(\tau+1)})^{\mathrm{T}}}{\sum_{i,j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})},\\
\lambda_{x_i}^{(\tau+1)} &= \tfrac{\sum_i P(x_i|\boldsymbol{Y},\boldsymbol{N})}{\sum_i P(x_i|\boldsymbol{Y},\boldsymbol{N})N_i},
\end{aligned} \tag{4}$$

where n_{ij} is the number of child-parent pairs (i,j) in $\mathcal{T}$.

The E-step and M-step are performed alternatively until $\min_{x_i,x_j}|\boldsymbol{\mu}_{x_ix_j}^{(\tau+1)}-\boldsymbol{\mu}_{x_ix_j}^{(\tau)}|/\boldsymbol{\mu}_{x_ix_j}^{(\tau)}<10^{-4}$, which takes only a few iterations in our experiments. The EM is guaranteed to converge to a local maximum or a saddle point $\hat{\Omega}$.

Learning Model Structure: The tree-union $\mathcal{T}$ is constructed by sequentially adding the texel subtrees $t\in\mathbb{D}$ to the model. This is done by matching t with the current estimate $\mathcal{T}^{(\tau)}$, and by adding and appropriately connecting to $\mathcal{T}^{(\tau)}$ the unmatched nodes from t, which yields $\mathcal{T}^{(\tau+1)}$. For matching t and $\mathcal{T}^{(\tau)}$, we use the same algorithm of [18, 17], described in Sec. 3. From (1), it follows that learning the model structure $\mathcal{T}$ depends on the definitions of node saliency r_i, and the cost of node matching $m_{ii'}$. Our goal is to estimate r_i and $m_{ii'}$, so that the learning yields the optimal model structure. This is accomplished by posing tree-union learning as an optimization problem with the objective to minimize data description length. In the sequel, we derive r_i and $m_{ii'}$.

We begin with the standard definition of the MDL of subtrees in $\mathbb{D}$:

$$\mathcal{L}\triangleq -\log P(\mathcal{T}|\Omega)-\log P(|\mathcal{T}|)+\tfrac{1}{2}|\Omega|\log|\mathbb{D}|, \tag{5}$$

where $P(\mathcal{T}|\Omega)$ is given by (2), $P(|\mathcal{T}|)$ is the prior of the number of nodes in $\mathcal{T}$, and $|\Omega|=3K^2+2K$ (Sec. 4). $P(|\mathcal{T}|)$ is assumed to be the exponential distribution with parameter Λ. Also, all tree-unions with $|\mathcal{T}|$ nodes are assumed equally likely. Since the number of ordered trees with n nodes is $\frac{1}{n}\binom{2n-2}{n-1}\xrightarrow{n\to\infty}4^n$, we obtain $P(|\mathcal{T}|)=\kappa e^{-\Lambda|\mathcal{T}|}4^{-|\mathcal{T}|}$, where κ is a normalization constant. Let $\mathcal{L}_j$ denote per-node description cost, whose definition follows from (2):

$$\mathcal{L}_j\triangleq\lambda_{x_j}N_j-\textstyle\sum_{i=1}^{N_j}\log[P(x_i|x_j)\mathcal{N}(\boldsymbol{y}_{ij};\boldsymbol{\mu}_{x_ix_j},\Sigma_{x_ix_j})]. \tag{6}$$

For the roots in $\mathcal{T}$, $\mathcal{L}_1$ also includes $-\log P(x_1)$. Then, from (5) and (2), $\mathcal{L}$ can be expressed in terms of per-node description costs as

$$\mathcal{L}=\textstyle\sum_{j\in\mathcal{T}}\mathcal{L}_j+|\mathcal{T}|(\Lambda+\log 4)+\frac{1}{2}|\Omega|\log|\mathbb{D}|, \tag{7}$$

where constants irrelevant to minimization of $\mathcal{L}$ are dropped. The expression in (7) shows that $\mathcal{L}$ is directly proportional to the number of nodes in $\mathcal{T}$. Thus, minimizing $\mathcal{L}$ amounts to finding $\mathcal{T}$ with the fewest nodes, which also preserves the original node adjacency and hierarchical relationships in $\mathbb{D}$. Indeed, such a graph is the tree-union, which can be constructed using the matching algorithm of Sec. 3, with r_i and $m_{ii'}$ defined so that $\mathcal{L}$ is minimized. To specify r_i and $m_{ii'}$, we relate the expressions for description length (7) and similarity measure (1), as discussed in the sequel.

When learning $\mathcal{T}^{(\tau+1)}$ from $\mathcal{T}^{(\tau)}$ and $t\in\mathbb{D}$, two nodes $i\in\mathcal{T}^{(\tau)}$ and $i'\in t$ may either be matched (case 1), or left unmatched (case 2). In case 1, i and i' form a joint node $\overline{ii'}$ to which we associate the vector of region properties $\boldsymbol{y}_{\overline{ii'}\cdot}=\text{mean}(\boldsymbol{y}_{i\cdot},\boldsymbol{y}_{i'\cdot})$. In case 2, node $i'\in t$ is added to $\mathcal{T}^{(\tau+1)}$, while i remains intact. From (7), description length advantage between the two cases is given by $\mathcal{A}(i,i')=\mathcal{L}_{\text{case 2}}-\mathcal{L}_{\text{case 1}}=\mathcal{L}_i+\mathcal{L}_{i'}-\mathcal{L}_{\overline{ii'}}$. It follows that the set of matches $f=\{(i,i')|i\in\mathcal{T}^{(\tau)},i'\in t\}$ that minimizes $\mathcal{L}$ also maximizes the advantage function

$$\mathcal{A}(f)=\max_f\textstyle\sum_{(i,i')\in f}(\mathcal{L}_i+\mathcal{L}_{i'}-\mathcal{L}_{\overline{ii'}}). \tag{8}$$

From (1) and (8), we conclude that maximizing $\mathcal{A}(f)$ can be identified with maximizing the similarity measure $\mathcal{S}$ by the matching algorithm of Sec. 3. This allows us to define the information theoretic node saliency and matching cost:

$$r_i\triangleq\mathcal{L}_i,\quad m_{ii'}\triangleq\mathcal{L}_{\overline{ii'}}, \tag{9}$$

where $\mathcal{L}_i$ and $\mathcal{L}_{\overline{ii'}}$ are given by (6). The definitions of r_i and $m_{ii'}$ in (9) guarantee that the texel model will respect the node-hierarchy constraints present in $\mathbb{D}$, and will have the MDL among all possible models. The overall learning algorithm is summarized in Alg. 1.

6. Experimental Results

For experimental validation, we carefully selected a new dataset containing 80 homogeneous 2.1D natural textures. All textures are on a nearly planar surface, and have been imaged from a direction nearly normal to the plane. The texels are all 3D, but their thickness and depth differences are much smaller than their distance from the camera. Therefore, they offer a good approximation to 2.1D textures. Each texture class is represented by three 320×240 images in the dataset, samples of which are shown in Figs. 1, 4, 5. The three images are obtained by cropping a single large texture image. The number of texels per

Algorithm 1: Learning the Texel Model

Input : $\mathbb{D}=\{t_1,\ldots,t_N\}$, $K\in\{2,3,4,\ldots\}$, $\Lambda\in[\frac{1}{200},\frac{1}{50}]$

Output: Model structure $\hat{\mathcal{T}}$, and model parameters $\hat{\Omega}$

1 Initialization of $\Omega^{(0)}$: Set $P(x_1)$ and $P(x_i|x_j)$ as uniform distributions; $\boldsymbol{\mu}_{x_ix_j}$=mean($\{\boldsymbol{y}_{ij}\}$), $\Sigma_{x_ix_j}$=cov($\{\boldsymbol{y}_{ij}\}$); $\lambda_{x_i}=1/$mean(# of children per node in $\mathbb{D}$);

2 Initialize $r_i^{(0)}\triangleq||\boldsymbol{y}_{i\cdot}||$, and $m_{ii'}^{(0)}\triangleq|r_i^{(0)}-r_{i'}^{(0)}|$;

3 Construct $\mathcal{T}^{(0)}$ from $\{t_1,t_2\}$ by using $r_i^{(0)}$ and $m_{ii'}^{(0)}$, where t_1 and t_2 are randomly selected from $\mathbb{D}$;

4 Set $\tau=0$, $\mathbb{D}=\mathbb{D}\backslash\{t_1,t_2\}$;

5 **while** $\mathbb{D}\neq\emptyset$ **do**

6 Construct $\mathcal{T}^{(\tau+1)}$ from $\mathcal{T}^{(\tau)}$ and t by using $r_i^{(\tau)}$ and $m_{ii'}^{(\tau)}$, where t is selected from $\mathbb{D}$ such that $\mathcal{L}$ is minimum;

7 Compute $\Omega^{(\tau+1)}$ as in (4) ;

8 Compute $r_i^{(\tau+1)}$ and $m_{ii'}^{(\tau+1)}$ as in (9) ;

9 $\mathbb{D}=\mathbb{D}\backslash\{t\}$; $\tau=\tau+1$;

10 **end**

11 $\hat{\mathcal{T}}=\mathcal{T}^{(\tau)}$, $\hat{\Omega}=\Omega^{(\tau)}$;

image ranges from 14 to 175, depending on the texture class. The dataset represents the following challenges (Figs. 4, 5): (1) inter-texel occlusions, (2) texel subregions can be viewed as texture at finer resolution (artichokes); (3) repetitive texel substructure may be confused with the texel (vertical stripes on the fish); (4) large variations in texel appearances (fish); (5) the texels may appear as low contrast regions (bees), thin regions (pine-trees), and their contours may form complex topologies (four contours of the leeks meet at one point), all of which are difficult to segment; (6) the background surface along which physical texture elements lie may be completely/heavily occluded by the elements (bees), which violates the assumption of some prior work that texels appear against the background; and (7) illumination may vary across the image (fish).

The texel model is learned on one out of three available images per texture class. Texel extraction is performed in the remaining two test images by matching the learned texel model with the two segmentation trees. The matches found with sufficiently large similarity measures are adjudged as texels. The ground truth is obtained by manually delineating the contours of all texels present in the image. A detected texel is said to be false positive if the XOR of its area with the true texel area is larger than their intersection. Segmentation error per true positive is defined as the ratio between the XOR, and union of its area with the true texel area. Average segmentation error is defined as the mean of texel segmentation errors on all true positives in the image. For all texture classes, we perform texel extraction experiments three times, each time using a different training image, and report the average results here. For the purpose of showing specific results, Figs. 4, 6 , and Table 1 use the similarity-measure threshold that yields the highest F-measure, $F\triangleq 2\cdot\text{Precision}\cdot\text{Recall}/(\text{Precision}+\text{Recall})$.

Texel segmentation: Figs. 4 and 5 illustrate high accuracy in detecting and segmenting the fully and partially visible texels. Extracted texels are shown by drawing the outer contours of the visible parts on the original. Performance is good even in cases when the texel edges are jagged and blurred (e.g., bees), and when several overlapping texels form a complex region topology (e.g., daisies). Figs. 4 and 5 also show examples where our texel extraction fails. Texels that are not detected, for the most part, have low intensity contrasts with the surround, and thus do not form texel-characteristic subtrees in the segmentation tree that can be matched with the texel model. Due to low contrast, parts of one texel may be occasionally confused as parts of the neighboring texels (petals of daisies). Also, severely occluded texels may not be detected, since the similarity measure of their matches with the texel model may not be sufficiently large (e.g., occluded oranges in Fig. 5).

Accuracy: The results in Table 1 and Fig. 6 are averages over all 80 textures in the dataset. Fig. 6 gives performance comparison of the information-theoretic edit-costs, given by (9), against the standard heuristic definition of edit-costs, given in Step 2 of Alg. 1. When the heuristic edit-costs are used, there is no need to estimate the model pdf's, and thus the learning algorithm in this case does not include Steps 7–8 of Alg. 1, i.e., it is equivalent to the algorithms presented in [17]. Learning the information-theoretic edit-costs yields significantly better performance.

Sensitivity and Run-time: Our texel extraction algorithm uses only two input parameters: (1) the mean number of nodes in the tree-union $1/\Lambda$, and (2) the number of hidden-variable states K. Λ controls the flat prior distribution $P(|\mathcal{T}|)$. Therefore, our algorithm is insensitive to a wide range of values $\Lambda\in[\frac{1}{200},\frac{1}{50}]$. Sensitivity to K is illustrated in Table 1 and Fig. 6. As K increases both the detection and segmentation become better, but with a major increase in complexity of Alg. 1 ($O(|\mathcal{T}|K^2)$, see Appendix). For example, learning takes approximately 6-15min for K=3, and 10-25min for K=6, with C-code on a 2.8GHz 2GB PC. Matching the tree-union with the segmentation tree takes 10-30s, depending on the number of nodes in these graphs.

7. Conclusions

We have presented what to our knowledge is the first attempt at solving texel detection and segmentation, for the case of homogeneous, frontally viewed, 2.1D, natural textures. This is accomplished by a new learning algorithm that combines tree matching, belief propagation on acyclic graphs, and EM, to learn structural, geometric, and photometric properties of the complete (unoccluded) texel from its partially visible occurrences in a 2.1D texture, without any supervision. Tree matching and tree-union learning have already been demonstrated to construct good representations under changes in illumination and view directions

[17]. Here we show that these algorithms also handle occlusions well. We plan to extend this work to cover nonfrontal and nonplanar textures with occlusion.

Appendix: Belief Propagation on Tree-Union

The E-step in the EM algorithm requires posterior marginals $P(x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})$ and $P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N},\Omega^{(\tau)})$, appearing in (3). We compute them by using the standard belief propagation (BP), derived below. Let $\boldsymbol{Y}_i$ and $\boldsymbol{N}_i$ denote all observables down the tree-union under node $i\in\mathcal{T}$, including $\boldsymbol{y}_{ij}$ and N_i. Let (i,j) denote child-parent node pair, and $C(i)$, the set of children of i. Then, we derive the BP algorithm on $\mathcal{T}$ as presented in Algorithm 2.

Algorithm 2: Belief Propagation on $\mathcal{T}$

Input : Current estimate $\Omega^{(\tau)}$ of model parameters, K
Output: Posteriors $P(x_i|\boldsymbol{Y},\boldsymbol{N})$ and $P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})$

1 Compute $\forall i,j\in\mathcal{T}$, $x_i,x_j\in\{1,\ldots,K\}$: $P(\boldsymbol{y}_{ij}|x_i,x_j)$, and $P(N_i|x_i)$ using $\Omega^{(\tau)}$, as specified in Sec. 4;

2 Compute $\forall(i,j)\in\mathcal{T}$ top-down:
$P(x_i,x_j)=P(x_i|x_j)P(x_j)$; $P(x_i)=\sum_{x_j}P(x_i|x_j)P(x_j)$;

3 Compute $\forall(i,j)\in\mathcal{T}$ bottom-up:

$$P(x_i,x_j|\boldsymbol{Y}_i,\boldsymbol{N}_j)\propto P(N_j|x_j)P(\boldsymbol{y}_{ij}|x_i,x_j)P(x_i,x_j)\cdot\prod_{c\in C(i)}\sum_{x_c}\frac{P(x_c,x_i|\boldsymbol{Y}_c,\boldsymbol{N}_i)}{P(x_i)};$$

4 Compute $\forall(i,j)\in\mathcal{T}$ top-down:
$P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})=\dfrac{P(x_i,x_j|\boldsymbol{Y}_i,\boldsymbol{N}_j)P(x_j|\boldsymbol{Y},\boldsymbol{N})}{\sum_{x_i}P(x_i,x_j|\boldsymbol{Y}_i,\boldsymbol{N}_j)}$,
$P(x_i|\boldsymbol{Y},\boldsymbol{N})=\sum_{x_j}P(x_i,x_j|\boldsymbol{Y},\boldsymbol{N})$.

In Step 3 of Alg. 2, "$\propto$" means that equality holds up to a multiplicative quantity, which is canceled out in Step 4. Complexity of the EM is defined by complexity of Alg. 2, which is $O(|\mathcal{T}|K^2)$; $|\mathcal{T}|$ is the number of tree-union nodes.

Acknowledgment

The support of the Office of Naval Research under grant N00014-06-1-0101 is gratefully acknowledged.

References

[1] N. Ahuja. A transform for multiscale image segmentation by integrated edge and region detection. *IEEE TPAMI*, 18(12):1211–1235, 1996.

[2] N. Ahuja and B. J. Schachter. Image models. *ACM Comput. Surv.*, 13(4):373–397, 1981.

[3] D. Blostein and N. Ahuja. Shape from texture: Integrating texture-element extraction and surface estimation. *IEEE TPAMI*, 11(12):1233–1251, 1989.

[4] O. G. Cula and K. J. Dana. 3D Texture recognition using bidirectional feature histograms. *IJCV*, 59(1):33–60, 2004.

[5] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *ICCV*, pages 716–723, 2003.

[6] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.

[7] J. H. Hays, M. Leordeanu, A. A. Efros, and Y. Liu. Discovering texture regularity as a higher-order correspondence problem. In *ECCV*, volume 2, pages 522–535, 2006.

[8] B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981.

[9] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE TPAMI*, 27(8):1265–1278, 2005.

[10] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

[11] T. K. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. In *ECCV*, volume 1, pages 546–555, 1996.

[12] W.-C. Lin and Y. Liu. Tracking dynamic near-regular textures under occlusion and rapid movements. In *ECCV*, volume 2, pages 44–55, 2006.

[13] A. Lobay and D. Forsyth. Shape from texture without boundaries. *IJCV*, 67(1):71–91, 2006.

[14] M. Mirmehdi and M. Petrou. Segmentation of color textures. *IEEE TPAMI*, 22(2):142–159, 2000.

[15] M. Nitzberg and D. Mumford. The 2.1-D sketch. In *ICCV*, pages 138–144, 1990.

[16] F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, volume LNCS 1681, pages 165–181, 1999.

[17] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, volume 1, pages 927–934, 2006.

[18] A. Torsello and E. R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Pattern Recogn. Lett.*, 24(8):1089–1097, 2003.

[19] A. Torsello and E. R. Hancock. Learning shape-classes using a mixture of tree-unions. *IEEE TPAMI*, 28(6):954–967, 2006.

[20] M. Tuceryan and A. Jain. Texture segmentation using Voronoi polygons. *IEEE TPAMI*, 12(2):211–216, 1990.

[21] A. Turina, T. Tuytelaars, and L. V. Gool. Efficient grouping under perspective skew. In *CVPR*, pages 247–254, 2001.

[22] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1-2):61–81, 2005.

[23] H. Voorhees and T. Poggio. Computing texture boundaries from images. *Nature*, 333:364–367, 1988.

[24] S.-C. Zhu, C.-E. Guo, Y. Wang, and Z. Xu. What are textons? *IJCV*, 62(1-2):121–143, 2005.

[25] S. C. Zhu, Y. N. Wu, and D. Mumford. Filters, random fields, and maximum entropy (FRAME): Towards a unified theory for texture modeling. *Int. J. Comp. Vision*, 27(2):107–126, 1998.

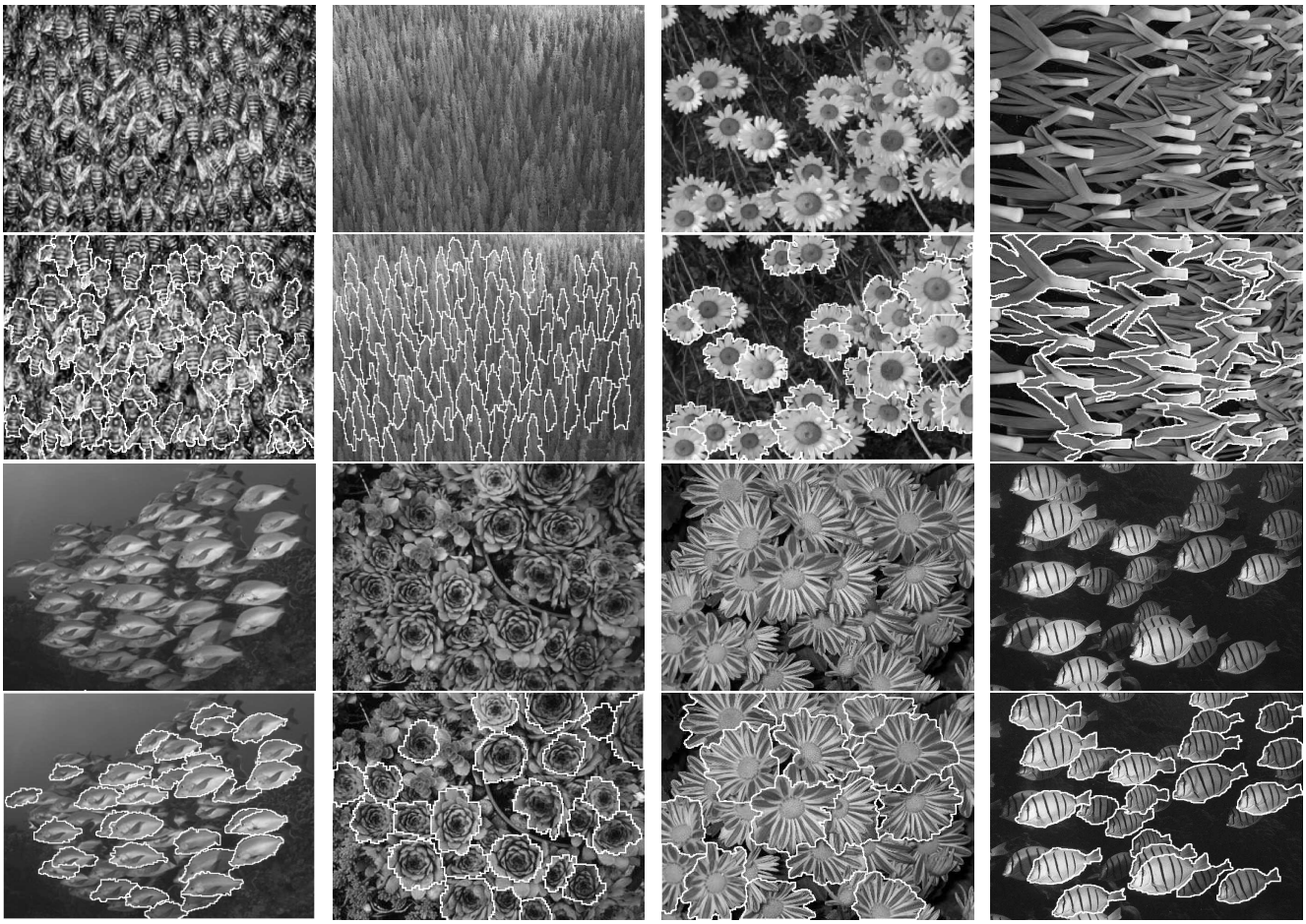

Figure 4. Results of texel extraction on eight 2.1D natural textures from our dataset of 80 ($K{=}4, \Lambda{=}\frac{1}{100}$). The contours of the detected texels are overlaid on the originals. Challenges to the algorithm include low-contrast regions between neighboring texels, thin and elongated texel regions, jagged and blurred texel edges, complex topologies of the texel boundaries, inter-texel occlusions, and varying illumination.

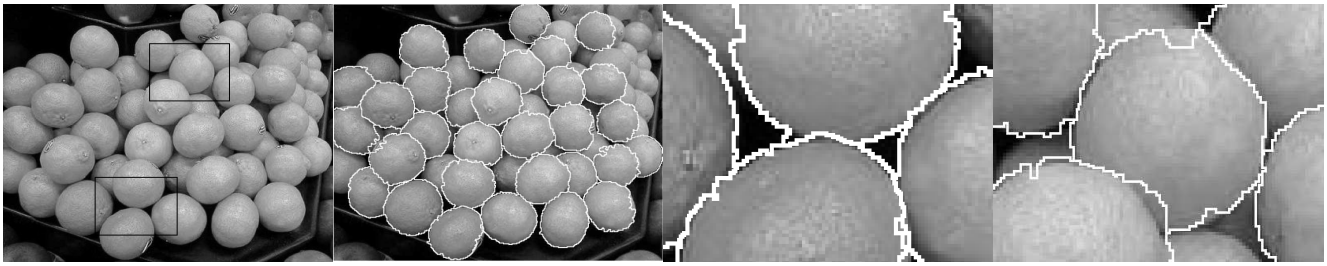

Figure 5. Texel extraction usually fails due to severe occlusion and low intensity contrast with neighboring texels. The matches of occluded texels with the texel model have lower similarity measures than the threshold. The low contrast texels are not even represented by the segmentation tree, and thus cannot be detected. The two right images show zoomed-in details of the black windows in the original image.

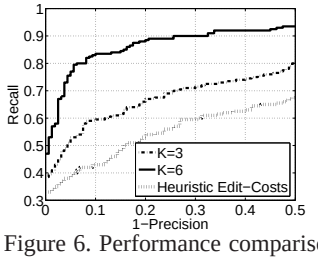

Table 1. Average Recall, Precision, and Segmentation Error (in %)

	$K{=}2$	$K{=}3$	$K{=}4$	$K{=}5$	$K{=}6$	Heuristic Edit-Costs
Recall	61.3±9.5	67.5±10.1	75.5±8.7	79.4±6.2	82.6±7.3	59.6±11.2
Precision	68.2±17.8	79.8±11.8	82.5±9.4	85.4±7.1	90.2±6.3	73.2±16.3
Segm. Error	29.8±14.5	25.4±13.1	20.3±9.4	17.9±10.4	16.7±8.1	19.9±11.4

Figure 6. Performance comparison of the information-theoretic edit-costs, given by (9), against the standard heuristic definition of edit-costs, given in Step 2 of Alg. 1.