Unsupervised Texture Flow Estimation Using Appearance-space Clustering and Correspondence

Sunghwan Choi, *Student Member, IEEE,* Dongbo Min, *Member, IEEE,* Bumsub Ham, *Member, IEEE,* and Kwanghoon Sohn, *Senior Member, IEEE*

***Abstract*—This paper presents a texture flow estimation method that uses an appearance-space clustering and a correspondence search in the space of deformed exemplars. To estimate the underlying texture flow such as scale, orientation and texture label, most existing approaches require a certain amount of user interactions. Strict assumptions on a geometric model further limit the flow estimation to such a near-regular texture as a gradient-like pattern. We address these problems by extracting distinct texture exemplars in an unsupervised way and using an efficient search strategy on a deformation parameter space. This enables estimating a coherent flow in a fully automatic manner, even when an input image contains multiple textures of different categories. A set of texture exemplars that describes the input texture image is first extracted via a medoid-based clustering in appearance space. The texture exemplars are then matched with the input image to infer deformation parameters. Specifically, we define a distance function for measuring a similarity between the texture exemplar and a deformed target patch centered at each pixel from the input image, and then propose to use a randomized search strategy to estimate these parameters efficiently. The deformation flow field is further refined by adaptively smoothing the flow field under guidance of a matching confidence score. We show that a local visual similarity, directly measured from appearance space, explains local behaviors of the flow very well, and the flow field can be estimated very efficiently when the matching criterion meets the randomized search strategy. Experimental results on synthetic and natural images show that the proposed method outperforms existing methods.**

***Index Terms*—Texture analysis, texture exemplar, texture flow, randomized search, medoid-based clustering.**

I. Introduction

TEXTURE that consists of surfaces in real photographs conveys semantic meanings of a scene. By exploring it as a basic visual structure, various vision tasks such as pattern recognition, texture synthesis, image retrieval, and segmentation can be feasible. The texture in the natural scene usually exhibits a spatially-varying deformation which arises from the geometric variation of surfaces. Nevertheless, the human visual system (HVS) can recognize its perceptual organization very well, since an inherent visual structure embedded in the spatially (and smoothly) varying surface shows visual coherency. For example, the HVS can recognize a set of textures containing geometric deviations in orientation and scale as the same one. This phenomenon stems from the fact that the HVS organizes and groups parallel structures into coherent units [1]. The visual structure characterized by a local parallelism is typically defined as *texture flow* [1], [2]. Fig. 1 shows the examples of the texture flow estimated from a natural image by using our method.

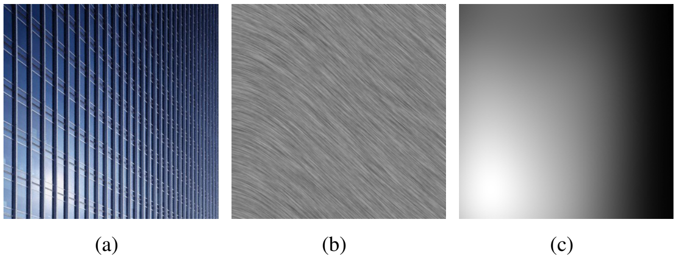


Fig. 1. Examples of texture flow estimation from a natural photograph. The input image (a) consists of repeating texture elements whose visual structure are spatially (and smoothly) varying in terms of orientation and scale. The underlying flow in image perception is decomposed as an orientation field (b) and a scale field (c).

In contrast to the perceptual mechanism of the HVS, such deviations might degrade the accuracy of computational approaches in estimating the texture flow, mainly due to a heavy computational burden and a lack of an effective model representing the *deformable* texture. Moreover, it is non-trivial to automatically estimate texture exemplars (to be used in the flow estimation) from an input image, thus making a direct application of the texture flow estimation to various vision tasks difficult. As a result, most approaches require a user-provided texture exemplar (a reference texture) to estimate underlying texture flows from natural photographs. Given synthetic or real world images, a user should specify a very precise texture exemplar with no deformation. Furthermore, in case of an image containing multiple textures, existing methods can handle a single texture at a time only when a user defines a corresponding region mask. Otherwise, the input image needs to be segmented appropriately.

In this paper, we present the unsupervised estimation of underlying texture flows in the image, by formulating the estimation as a discrete labeling problem. Our objective is to automatically extract any number of texture exemplars and to infer a dense deformation field based on a visual correspondence

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2013R1A2A2A01068338).

S. Choi and K. Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea (e-mail: shch@yonsei.ac.kr; khsohn@yonsei.ac.kr).

D. Min is with the Department of Computer Science and Engineering, Chungnam National University, Daejeon 305-764, Korea (e-mail: dbmin@cnu.ac.kr).

B. Ham is with the INRIA-Willow Project-Team and the École Normale Supérieure (CNRS/ENS/INRIA UMR 8548), 23 avenue d'Italie, 75013, Paris, France (e-mail: bumsub.ham@inria.fr). B. Ham is supported by the research grant for the European Research Council, VideoWorld.

search. From here on, we define the deformation parameter with texture label, scale and orientation for simplicity. To this end, we need to solve the following problems: 1) how to identify texture exemplars with no user intervention, and 2) how to efficiently infer the deformation parameters in a globally consistent manner. We present the texture representation model which builds upon an inherent repeatability behind texture appearance [3]. Namely, the texture appearance is formed through the evolution of some explicit texture exemplars. Based on this, we pose the extraction of texture exemplars as a mode-seeking problem: we find the most visually similar patch among all candidate sample patches in the image. It is effectively solved with the medoidshift algorithm [4]. Then, we propose a non-parametric deformation model for a texture flow estimation through a visual correspondence search using the estimated texture exemplars. One challenging problem is a heavy computational complexity incurred by an extremely large search space. We will resolve this computational issue by deploying the randomized search concept recently proposed in [5], [6]. The deformation field estimation can be significantly accelerated, while maintaining its estimation accuracy. To ensure a global coherency, an intermediate flow field is adaptively filtered with the guidance of the matching confidence.

This paper is organized as follows. Section II presents the related work, and Section III explains the inverse problem of estimating texture flows and provides the algorithm overview. Section IV describes the unsupervised extraction of texture exemplars, and the non-parametric deformation field estimation is presented in Section V. Then, the performance of the proposed method is demonstrated in Section VI. Finally, Section VII discusses limitations and concludes this paper.

II. Related Work

There is a great deal of work in a large family of texture analysis. Here, we only discuss the work most related to our approach.

One stream of existing work that is closely related to ours is texel extraction. It focuses on identifying small patches, usually known as texture exemplars. Texture exemplars are a set of fundamental sub-images which appear as spatially repeated texture elements in the scene. Ahuja *et al.* [7] proposed to precisely segment texture exemplars based on a multiscale segmentation tree. The works of [8] and [9] detected texture exemplars by discovering texture regularity. Wei *et al.* [10] extracted a compact texture exemplar that best summarizes the input image by compressing texture regions. These methods were proven to be effective in a single texture image. However, they are not able to distinguish multi-labeled texture exemplars which we address in this work. Furthermore, they are primarily designed for texture segmentation or synthesis, so not directly applicable to our task (texture flow estimation).

Texture synthesis has long been researched as another area of texture analysis. Early works [11], [12] aim to synthesize isometric textures by leveraging region-growing techniques. They start with a small exemplar image and evolve the output texture one pixel or patch at a time, while maintaining coherence of the grown region with its vicinity. Recently, several algorithms [3], [13], [14] have been devised, which incorporate texture flow to guide an anisometrically varying synthesis output. The geometric deviation of output texture varies according to the amount of the artificial deformation field. In other words, the anisotropic synthesis allows local rotation and scaling of the synthesized texture along with the deformation field. In these anisotropic synthesis methods, the deformation field is typically specified by a user. While the texture flow is a fundamental component needed for this task, there is relatively little attention in estimating the underlying deformation field from real or synthetic images. Even existing flow estimation methods [2], [15]–[17] all require a user intervention and/or handle a single textured image only.

Previous work on discovering texture deformation can be broadly classified into two categories: local approaches based on a texture descriptor with no consideration of their global consistency, and global approaches using an optimization formulation defined on high-dimensional flow labels. The local approaches typically make use of a local attribute (*e.g.*, gradient information) or a parametric model for texture representation. Kang *et al.* [18] estimated the orientation field directly from the gradient vector field. Shahar *et al.* [1] proposed to incorporate curvature information, while oriented filters were deployed to estimate a dominant orientation field for gradient-like patterns [15]. In [16], Chang and Fisher decomposed a deformed texture into explicit local attributes such as orientation and scale by utilizing a steerable pyramid. In [17], the response of the structure tensor computed at each pixel is compared to that computed from a texture exemplar in order to discriminate the dominant orientation inherent in the texture. These local approaches often utilize an over-simplified model (*e.g.*, image gradient). With an increasing irregularity it becomes more difficult to find a coherent flow field, and thus these methods are only applicable to a limited subset of textures, *e.g.*, near-regular texture images. The method proposed in [2] represents a texture feature as a linear array of thresholded pixels (*i.e.*, local binary patterns), followed by the dimension reduction strategy through the principal component analysis (PCA). This kind of representation was shown to be effective in discovering the texture flow from irregular texture images. However, similar to existing works, this method still requires a user-provided texture exemplar with no deformation, and it also deals with a single texture only.

To further enforce a global consistency in the flow estimation, a costly global optimization is often taken into account [2], [9], [16], by minimizing an objective function which combines the data constraint with a regularization term enforcing smoothness on the resultant flow fields. However, such optimization-driven methods suffer from the computational burden caused by a high-dimensional label space and/or quantization artifacts inherent in discrete labeling tasks.

To the best of our knowledge, our method is the first approach that estimates globally consistent texture flows with no user intervention. It thus enables the flow estimation in a multi-textured image, which is not feasible with existing methods. Moreover, our method directly utilizes intensity values of two patches for computing the correlation metric, unlike existing texture flow estimation approaches [2], [17] that rely on more elaborate texture descriptors. Since the texture has a stochastic property, the intensity-based correlation measure has been traditionally considered unfit in the texture flow estimation. However, it was shown in the texture synthesis literature [13] that the appearance (*e.g.*, patch) based synthesis approach produces a spatially coherent texture image very well, when the deformation parameters are taken into account. Following this observation, we will demonstrate that the local visual similarity directly measured from a simple intensity correlation metric captures local behaviors of the inherent flow very well.

III. Problem Statement and Overview

Given a single- or a multi-texture image I that can be perceptually organized into $L \geq 1$ distinct regions and undergoes smoothly varying deformation in surfaces, our objective is to discover a dense deformation field $f : I \mapsto \mathbb{R}^3$ defined over all pixel coordinates $\mathbf{p} \in \Omega$ through a deformable correspondence search on an appearance space. To explore the underlying texture regularity, existing flow estimation methods typically define a texture deformation model with orientation $\theta_\mathbf{p}$ and scale $s_\mathbf{p}$ only. In order to cope with a multi-texture image in a fully automatic manner, we extend it by introducing a texture label parameter $l_\mathbf{p} \in \{1, ..., L\}$ indicating which texture exemplar each pixel belongs to:

$$f(\mathbf{p}) = (l_\mathbf{p}, \theta_\mathbf{p}, s_\mathbf{p})^T, \quad \forall \mathbf{p} \in \Omega. \tag{1}$$

Since orientation is periodic, the search range of orientation is constrained to be $0 \leq \theta_\mathbf{p} < 2\pi$. The search range of scale is preset to be in the range of $0.25 \leq s_\mathbf{p} \leq 2$. To mitigate quantization artifacts in the flow estimation, the discrete parameters are very densely sampled. Here, it is worth noting that our goal is to precisely estimate the flow vector, not the texture label parameter. In this context, the texture label parameter $l_\mathbf{p}$ is used to distinguish either 1) texture exemplars with semantically different visual structures or 2) similar texture exemplars yet with geometric and/or photometric variations. Conventionally, the texture exemplar has been defined as having no distortion [2], but there is no objective measure for defining the degree of the distortion. Thus, the texture exemplar with no distortion is always given by a user in existing approaches. Alternatively, our method attempts to find all possible texture exemplars that contain a similar *perceptual* structure (*i.e.* recognized as the same texture by an observer) yet have a certain amount of variations. This non-parametric sampling strategy can be a good choice to deal with the photometric variations which are very challenging in estimating the texture deformation field. For instance, in Fig. 2, T_2 and T_3 are perceptually similar, and can be recognized as the same texture exemplar by an observer. However, our method assigns different labels to the two exemplars, enabling a better representation of local attributes at each region. This leads to a more effective deformation field estimation.

The proposed method consists of two main stages: unsupervised texture exemplar extraction and dense deformation field estimation. In the first step, L distinct texture exemplars $\mathbf{T} = \{T_1, T_2, ..., T_L\}$ are automatically extracted using histogram-based features. They are represented as a histogram which sparsely encodes visual structure in order to reduce matching ambiguities incurred by the geometric deviation in the texture appearance. Based on histogram-based features, the distance metric space is defined using the χ^2 distance [19]. The sample patches distributed in the metric space are then clustered by using an unsupervised clustering method such as the medoidshift algorithm [4] in order to find a set of local modes on the feature space. The sample patches associated with those modes are then determined as texture exemplars $\mathbf{T}$. In the second stage, a globally coherent dense deformation field is computed based on the extracted texture exemplars $\mathbf{T}$. Please note that $\mathbf{T}$ is defined with a set of patches, and histogram-based feature vectors are not used in the inference stage any more. We cast an inverse estimation of the deformation field as a discrete labeling problem on the very densely quantized label space. It is efficiently solved with the randomized search algorithm [5], [6], enabling an efficient estimation of texture deformation without quality degeneration, *e.g.*, due to quantization artifacts which often appear in the optimization-driven discrete approaches [2]. A locally-adaptive smoothing is then applied to the intermediate deformation field, resulting in globally coherent texture flows.

IV. Unsupervised Extraction of Texture Exemplars

Before explaining the estimation of texture exemplars, let us first exploit the nature of texture synthesis. From the perspective of texture synthesis, a synthesized texture appearance is formed through the evolution of input texture exemplars, while maintaining a spatial coherency between stitched deformed textures [3]. The output texture image is hence perceptually similar to the input texture exemplars. The key idea of our approach is drawn from this inherent similarity behind the texture model: the texture exemplar T has the smallest distance in terms of visual appearance among all other sample patches. When an input image consisting of deformed textures is given, we define a texture exemplar as a representative patch that has a minimum distance with respect to deformation parameters among all other sample patches obtained from the input texture image. In this context, texture exemplars can be extracted by computing local modes among all sample patches on a metric space, where a valid distance measure between samples is defined. Fig. 2 illustrates an algorithm overview of our unsupervised texture exemplar extraction.

For ease of algorithm explanation, we first consider an input image I with a single texture exemplar, *i.e.*, $L = 1$, under the assumption that the set of possible deformation hypotheses $\mathbb{F}$ is known. Let Φ denote a set of all possible texture exemplars, *e.g.*, $\Phi = \{W_\mathbf{p} | \mathbf{p} \in \Omega\}$ consisting of all patches densely sampled from I, where Ω represents a set of 2D pixels and $W_\mathbf{p}$ is a sample patch with a radius r centered at a point $\mathbf{p}$.

A texture exemplar $T \in \Phi$ can then be estimated by minimizing the following global coherence function:

$$O(T) = \sum_{\mathbf{p} \in \Omega} \min_{f \in \mathbb{F}} d(T^{(f)}, W_\mathbf{p}), \quad T \in \Phi, \tag{2}$$

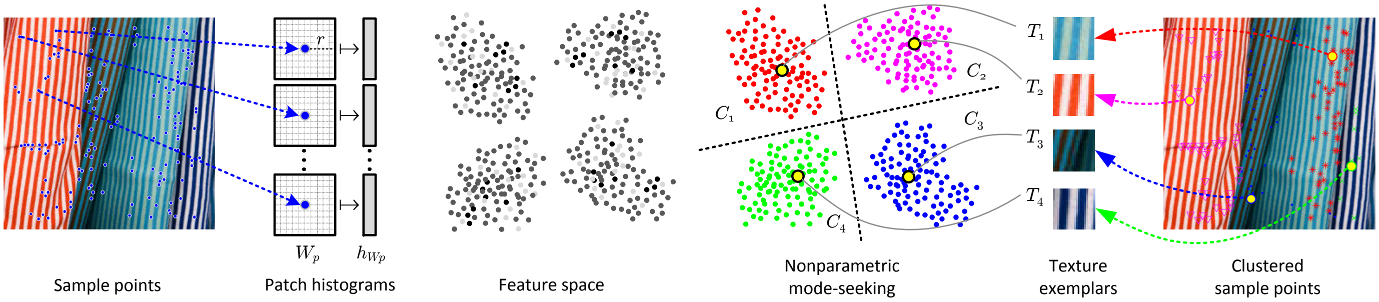


Fig. 2. Overview of the unsupervised texture exemplar extraction algorithm. The input texture image is represented as a set of sparsely sampled features based on a normalized 3D histogram. Texture exemplars $T_1, \ldots, T_4$ are then extracted from the modes of the density estimated with a Gaussian kernel in the feature space. The set of all sample patches that converge to the same mode defines the cluster $C_1, \ldots, C_4$ of that mode.

where f corresponds to a possible deformation parameter consisting of scale and orientation. A superscript (f) denotes a deformation operator, such that $T^{(f)}$ represents a patch that is deformed with respect to the parameters $f \in \mathbb{F}$. $d(\cdot,\cdot)$ is a local similarity measure between two patches, *e.g.*, Euclidian distance. Our global coherence function requires that the texture exemplar T must meet visual coherency among sample patches $W_\mathbf{p}$ for all $\mathbf{p} \in \Omega$. The coherence for each pixel $\mathbf{p}$ is measured by hypothesizing all the possible deformations $\mathbb{F}$.

One can estimate a global mode (*i.e.*, texture exemplar T) by minimizing the coherence function (2) over all possible sample patches Φ. However, such an exhaustive search is computationally expensive, since the original deformation parameters actually exist on the continuous space, and thus the discretized search space $\mathbb{F}$ is typically very huge. Also, the number of candidate texture exemplars Φ is proportional to the image size, thus leading to huge computational cost.

A. Histogram-based Feature Vectors

Our strategy to overcome this problem is to use a sparse representation of the input texture, where a patch histogram representing the joint distribution of lightness (I_L) and the opponent colors (I_a and I_b), which are expressed in the CIELAB color space, is adopted as a feature vector. This representation leads to a certain invariance against geometric variations by scale and orientation. It thus allows one to measure correlation between two patches without explicitly considering their possible deformation hypotheses. In addition, we compute the feature vectors only at sparsely sampled interest points, since relevant visual cues for capturing the deformation field are mostly concentrated around these interest points [9].

Other applications such as image classification [20] typically require using a more sophisticated representation (*e.g.*, bag-of-features [21]) to deal with a problem defined over a general scene. In contrast, our approach takes as an input a specific type of scene that consists of repeated texture elements only. This inherent repeatability behind the texture appearance suggests that texture elements (sample patches) show a similar color distribution, even when they undergo a certain amount of geometric variations (scale and orientation). Therefore, our simplified representation for a texture image is sufficient enough to extract texture exemplars.

For defining a feature vector for a patch $W_\mathbf{p}$, let us denote $h_{W_\mathbf{p}}(\mathbf{b})$ as a bin of 3D histogram:

$$h_{W_\mathbf{p}}(\mathbf{b}) = \frac{1}{|W_\mathbf{p}|} \sum_{\mathbf{q} \in W_\mathbf{p}} \delta(\mathbf{b}, \mathbf{u}(\mathbf{q})), \tag{3}$$

where $\mathbf{u}(\mathbf{q}) = (I_L(\mathbf{q}), I_a(\mathbf{q}), I_b(\mathbf{q}))^T$. The indicator function $\delta(\mathbf{m},\mathbf{n}) = 1$ when $\mathbf{m} = \mathbf{n}$, and 0 otherwise. We define a feature vector $h_{W_\mathbf{p}}$ as a normalized 3D histogram representing a generic color distribution with appropriate quantization levels $[q_1, q_2, q_3]$. In this paper, the quantization levels for histogram generation are set to $[10, 5, 5]$: they are not tuned for particular images but chosen to capture general aspects of texture. Here, the feature vector can be viewed as concatenated bins of the histogram, *i.e.*, 250-dimensional vector.

Accordingly, our global coherence function can be reformulated using the histogram based similarity measure as follows:

$$\tilde{O}(T) = \sum_{\mathbf{p} \in \Omega_s} \tilde{d}(h_T, h_{W_\mathbf{p}}), \quad T \in \Phi_s, \tag{4}$$

where Ω_s is a set of sparsely sampled points using interest point detector [9]. So, the set of candidate texture exemplars is also defined as $\Phi_s = \{W_\mathbf{p} | \mathbf{p} \in \Omega_s\}$. Note that, unlike (2), our relaxed global coherence function (4) does not need to consider all possible deformation hypotheses $f \in \mathbb{F}$. $\tilde{d}(\cdot,\cdot)$ is a metric function for measuring a similarity between two histograms. We use the χ^2 distance [19] which is given by

$$\tilde{d}(h_1, h_2) = 2 \sum_{b=1}^{B} \frac{h_1(b) h_2(b)}{h_1(b) + h_2(b)}, \tag{5}$$

where $B = 250$ is the number of bins used in the histogram, and $h_x(b)$ represents a value at bin b of the histogram h_x.

B. Texture Exemplar Extraction

So far, we have presented a method for extracting a single texture exemplar by minimizing (4), when $L = 1$. This method is further extended into a generalized texture exemplar estimation, *i.e.*, L is unknown. We can estimate a set of texture exemplars $\mathbf{T} = \{T_1, T_2, ..., T_L\}$ as well as L by computing all the local modes (including the global mode).

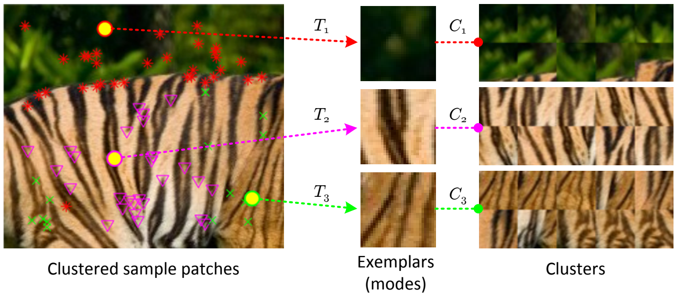
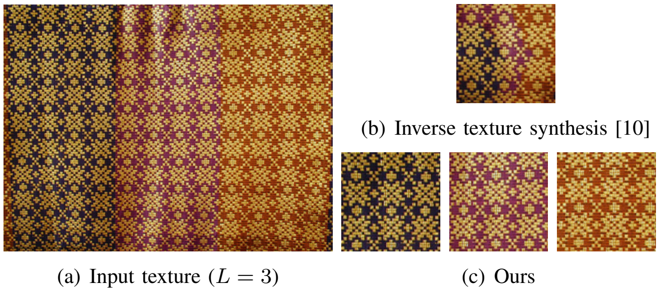


Fig. 3. Examples of texture exemplar extraction via the medoidshift clustering. Once sparsely sampled interest points form the feature space, texture exemplars are then defined as the local modes of the underlying density estimated with the kernel φ from the feature space. $L = 3$ explicit texture exemplars are automatically extracted in this example. The clustered sample patches in C_x show strong correlation to the corresponding texture exemplar T_x.


(a) Input texture ($L = 3$)

(b) Inverse texture synthesis [10]

(c) Ours

Fig. 4. Comparisons of extracted texture exemplars using inverse texture synthesis [10] and our method. (a) The input texture shows $L = 3$ photometric variations. (b) A single texture exemplar is extracted by [10] that best summarizes a contextual sense of the given texture, while (c) three explicit texture exemplars that cover $L = 3$ physical texture elements are completely extracted by our method. As can be seen in (c), our approach allows an automatic separation of visual cues prior to the inference of deformation field.

Each mode corresponds to a local minimum and represents a texture exemplar. In short, this minimization task can be seen as an unsupervised mode-seeking problem on the sample domain Φ_s. The set of all sample patches that converges to the same mode is defined as the cluster of the mode. The set of candidate exemplars Φ_s can be partitioned into different coherent clusters $C_1, ..., C_L$ based on histogram similarity measures. Here, we adopt the medoidshift algorithm [4], which can work on the non-Euclidean metric space (*e.g.*, χ^2 distance employed in this paper), unlike the meanshift like clustering approaches [22].

More specifically, we estimate texture exemplars that converge to local minima which exist on the following density function corresponding to (4). For all $\mathbf{p} \in \Omega_s$,

$$T_{\mathbf{p}} = \arg\min_{V\in\Phi_s} \sum_{\mathbf{q}\in\Omega_s} \tilde{d}\left(h_V, h_{W_{\mathbf{q}}}\right) \varphi_\sigma\left(\tilde{d}\left(h_{W_{\mathbf{q}}}, h_{W_{\mathbf{p}}}\right)\right), \qquad (6)$$

where φ_σ is an exponential kernel with a bandwidth parameter $\sigma = 0.2$ used to evaluate the underlying density, *i.e.*, $\varphi_\sigma(x) = \exp\left(-\frac{|x|}{\sigma}\right)$. The patch $T_{\mathbf{p}} \in \Phi_s$ represents the mode of the patch $W_{\mathbf{p}}$ that has the minimum weighted distance to all other patches in Φ_s, such that it is a minimizer of the objective function (6). Note that, with $\varphi_\sigma = 1$, the function (6) becomes exactly the same as what provides a global mode. By evaluating (6) for all sample patches in Φ_s, the trajectory of each sample patch $W_{\mathbf{p}}$ is evolved toward a local mode $T_{\mathbf{p}}$ for some $\mathbf{p} \in \Omega_s$. Here, we define the set of points representing local modes as $\bar{\Omega}_s$, such that $\bar{\Omega}_s \subset \Omega_s$. Accordingly, the set of texture exemplars is defined as $\mathbf{T} = \{W_{\mathbf{p}} | \mathbf{p} \in \bar{\Omega}_s\}$. It is worth noting that the trajectories $\mathbf{p} \in \bar{\Omega}_s$ are constrained to pass through the sample points Ω_s, so the modes should belong to points in the sample set Ω_s [4].

To avoid over-fragmentation of the resultant clusters, the minimization of (6) is iteratively performed on the constrained sample points $\Omega_s^{(0)}, \Omega_s^{(1)}, \ldots, \Omega_s^{(t)}$, where at each iteration t the sample points are redefined with their previous modes: $\Omega_s^{(t)} \triangleq \bar{\Omega}_s^{(t-1)}$ with initial values $\Omega_s^{(0)} = \Omega_s$. Namely, for every iteration step, input sample patches are replaced with their modes $\Omega_s^{(t)}$ at the previous iteration, and then the clustering process proceeds with new $\Omega_s^{(t)}$ as the sample patches.

For ease of implementation, we can formulate (6) as a matrix form [4]:

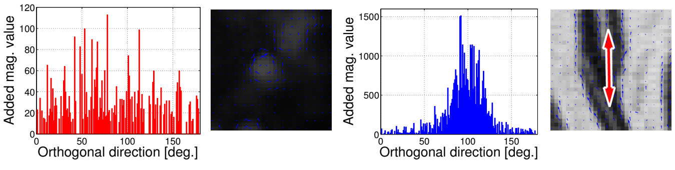
$$\mathbf{S}^{(t)}(i,j) = \sum_{k=1}^{N^{(t)}} \tilde{d}(h_{W[j]}, h_{W[k]}) \varphi\left(\tilde{d}(h_{W[i]}, h_{W[k]})\right), \qquad (7)$$

where $\mathbf{S}^{(t)}$ is an $N^{(t)} \times N^{(t)}$ symmetric matrix at the t^{th} iteration with $N^{(t)} = |\Omega_s^{(t)}|$ and $W[i]$ denotes a patch located at i^{th} sample point in $\Omega_s^{(t)}$. Each entry along the i^{th} column of $\mathbf{S}^{(t)}$ contains the sum of weighted distance from all other sample patches for $W[i]$. Once the matrix $\mathbf{S}^{(t)}$ is constructed, the local mode of iteration t for the i^{th} sample patch $W[i]$ is denoted by the index $i^{(t)}$ with the minimum value in the i^{th} column of $\mathbf{S}^{(t)}$, *i.e.*, $i^{(t)} = \arg\min_j \mathbf{S}^{(t)}(i,j)$. This step repeats until no further change occurs at iteration t'. The index set of final modes is then defined as $\Lambda_s = \{i^{(t')} | 1 \leq i \leq N^{(t')}\}$. Finally, texture exemplars $T_1, \ldots, T_L$ are extracted as $W[i]$ for all $i \in \Lambda_s$, and the number of extracted texture exemplars is $L = |\Lambda_s|$.

Fig. 3 shows the results of texture exemplars extracted by our method. The extracted texture exemplars summarize clustered patches well in terms of visual appearance. It is important to note that the extracted texture exemplars have a slightly different context from those of other approaches. For example, the texture compaction method in [13] aims to extract a single compact exemplar that best depicts a contextual sense of a given texture. Hence, this method describes the image as a synthesized result with a single texture only. When a multi texture image as in Fig. 4(a) is given, three texture elements are combined together as in Fig. 4(b), giving poor flow estimation results in our case. In contrast, our method treats the image locally by clustering input patches. Thus, our method enables an unsupervised estimation of L and an automatic separation of visual cues, as shown in Fig. 4(c). We found this to be effective in the texture flow estimation, since extracted multiple texture exemplars provide a wider range of coverage for particular texture elements than a single one.

C. Reference Orientation Assignment

After texture exemplars are extracted, the reference orientation (RO) ϕ_l is assigned to each texture exemplar T_l



(a) OGH for T_1 (b) RO for T_1 (c) OGH for T_2 (d) RO for T_2

Fig. 5. Results on the automatic assignment of reference orientation (RO) using orthogonal gradient histogram (OGH). Given the texture exemplars T_1 and T_2 of Fig. 3, (a) and (c) represent the computed OGH for T_1 and T_2, respectively. Selected reference orientations are respectively (b) $\phi_1 = 52°$ (or $232°$) and (d) $\phi_2 = 91°$ (or $271°$).

($l = 1, ..., L$). This procedure is needed to visualize the texture flow. Although a *relative* orientation between the texture exemplar and each patch from an input image is estimated in the next section, our method still requires estimating the RO to visualize an *absolute* orientation of the estimated texture flow in a visually natural manner.

Fig. 5 shows the results of the ROs selected using the orthogonal gradient histogram (OGH) proposed in this section. The input texture exemplars T_1 and T_2 are extracted from the image in Fig. 3. In case of well-structured texture exemplars like T_2, the RO is obtained automatically by taking a direction with a maximum frequency from the OGH. Specifically, we use a histogram of possible orientations from gradient information, where each bin is linearly spaced by $1°$. Supposing that a gradient vector field $\mathbf{g}(\mathbf{p}) = (g_x(\mathbf{p}), g_y(\mathbf{p}))^T$ of a particular texture exemplar T is given, one can compute an orthogonal direction $\theta^{\perp}(\mathbf{p})$ of a vector perpendicular to the image gradient $\mathbf{g}$ by, for example, $\theta^{\perp}(\mathbf{p}) = \arctan(g_y(\mathbf{p}), -g_x(\mathbf{p}))$. We add the magnitude value $m(\mathbf{p}) = ||\theta^{\perp}(\mathbf{p})||_2$ to the corresponding bin of $\theta^{\perp}(\mathbf{p})$. When $m(\mathbf{p}) < \tau$, we skip the binning of $\theta^{\perp}(\mathbf{p})$ ($\tau = 10$ in this paper). Once the histogram is computed, the RO is set to the direction that corresponds to the maximum magnitude (peak) in the histogram. Originally, the orientation field is represented with 2π-periodicity, but considering the bi-directional property of the texture flow, we represent it with π-periodic orientation [1].

In Fig. 5(d), the RO for T_2 is computed as $\phi_2 = 91°$ which corresponds to the peak in the histogram of Fig. 5(c). In case of the unstructured texture exemplar T_1 as in Fig. 5(b), however, no distinct orientation is observed due to homogeneous textures as shown in Fig. 5(a). For such cases, the RO may be assigned manually by a user. Otherwise, for a fully automatic estimation, we can simply assume that this exemplar contains no meaningful texture if the standard deviation of its normalized OGH exceeds a pre-defined threshold $\kappa = 3$, and the pixels belonging to this texture label are also considered invalid in the following deformation estimation. It should be noted that the RO estimation is just for visualizing the texture flow estimated in the following section, and thus estimating relative orientations is still feasible without the RO.

V. Non-parametric Deformation Field Estimation

We now focus on the non-parametric estimation of the deformation field, and expose the strategy for casting this inverse problem as a globally consistent deformable matching using the randomized search concept. Once texture exemplars are extracted, we infer the underlying deformation field f by matching texture exemplars with target patches for all pixels in the given texture image.

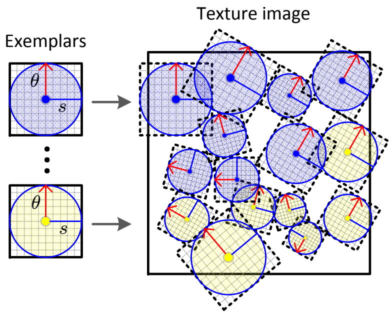


Fig. 6. Non-parametric deformation model for estimating the underlying deformation field. The local visual similarity between texture exemplars and target patches in the image is directly measured from the appearance space, where rotation and scale transformations with varying parameters are taken into account to find best matches.

A. Non-parametric Deformation Model

Our inference algorithm based on a non-parametric deformation model aims to best match a texture exemplar T_l with a deformed target patch, centered at a pixel $\mathbf{p}$ from I, by varying the amount of rotation and scale, as shown in Fig. 6. Let E_I and E_G denote respectively a distance function for measuring intensity and gradient similarity of two patches as:

$$E_I(\mathbf{p}, l, \theta, s) = \sum_{\mathbf{q} \in \mathcal{N}(T_l)} \left\| T_l(\mathbf{q}) - I(\phi_{\mathbf{p}}^{(\theta,s)}(\mathbf{q})) \right\|^2, \quad (8)$$

$$E_G(\mathbf{p}, l, \theta, s) = \sum_{\mathbf{q} \in \mathcal{N}(T_l)} \left\| \nabla T_l(\mathbf{q}) - \nabla I(\phi_{\mathbf{p}}^{(\theta,s)}(\mathbf{q})) \right\|^2, \quad (9)$$

where ∇ is a gradient operator and $\mathcal{N}(T_l)$ is a set of relative pixel locations with setting the center of the texture exemplar T_l to an origin. $\phi_{\mathbf{p}}^{(\theta,s)}(\cdot)$ denotes a warping operator with respect to rotation θ and scale s, which yields

$$\phi_{\mathbf{p}}^{(\theta,s)}(\mathbf{q}) = \mathbf{p} + s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{q}. \quad (10)$$

Based on these intensity-based distance measures, the visual similarity of two patches is defined as follows:

$$\mathcal{V}(\mathbf{p}, l, \theta, s) = E_I(\mathbf{p}, l, \theta, s) + E_G(\mathbf{p}, l, \theta, s). \quad (11)$$

The deformation field $f(\mathbf{p})$ is then estimated by minimizing the following objective function:

$$f(\mathbf{p}) = \mathop{\arg\min}_{(l,\theta,s) \in \mathbb{F}} \mathcal{V}(\mathbf{p}, l, \theta, s), \quad (12)$$

where $\mathbb{F}$ is the set of all possible parameters with respect to label l, rotation θ and scale s. Indeed, this minimization problem can be simply solved by exhaustively searching over the discretized label space of $\mathbb{F}$. However, such an exhaustive search is computationally expensive, since the number of possible deformation parameters is typically very huge.

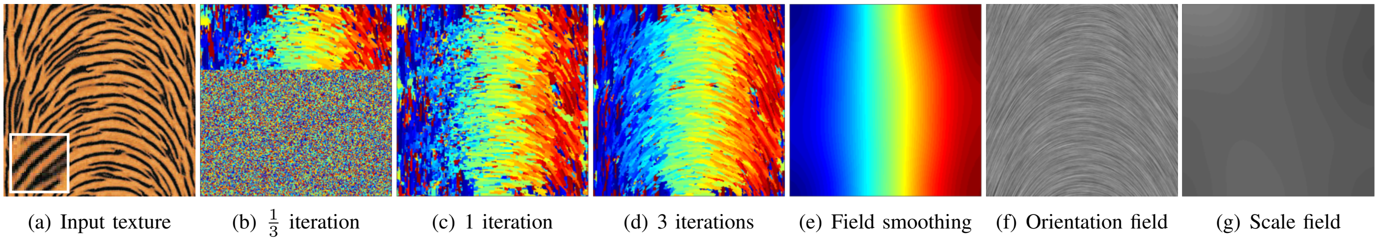


Fig. 7. Overview of the randomized inference. Given the input texture (a) with the extracted $L = 1$ texture exemplar (overlaid over (a)), our randomized inference algorithm iteratively updates the deformation field (b)-(d) from random initialization, followed by local field smoothing (e). It produces globally consistent orientation (f) and scale (g) fields. Note that orientation fields are encoded in colors for visualization.

B. Randomized Inference

To handle the computational complexity introduced by a large number of candidate labels, we deploy the randomized search algorithm [5], which is proven to be highly efficient in the high-dimensional discrete label search. Fig. 7 presents the overview of our randomized inference algorithm. Instead of an exhaustive search over all possible parameters, we smartly traverse *parts* of it using a randomized cooperative hill climbing strategy: *propagation* and *random search*. A basic motivation of the randomized search algorithm [5] is rather simple: if the deformation field is initialized by random labels, then correct labels are likely to exist among the set of these random labels. A good guess of some pixels guides the rest of pixels so that they also have a good guess by propagating its current labels to the vicinity. This randomized inference process iteratively updates the deformation field f until convergence. For each iteration, good guesses are examined in scan order by alternating between propagation and random search. It should be noted that our algorithm randomly selects arbitrary (floating) values for the orientation and scale within the given search range. Therefore, although discrete, our method does not suffer from severe quantization artifacts while maintaining its runtime efficiency.

Propagation. In the first step, a propagation proceeds in order to improve an intermediate deformation label $f(\mathbf{p})$ by considering current best label pairs $\Psi_\mathbf{p}$ of its neighboring pixels including itself. For instance, $\Psi_\mathbf{p} = \{f(\mathbf{p}), f(\mathbf{p} - (1,0)), f(\mathbf{p} - (0,1))\}$ on odd-numbered iteration. The hypothesis test is then performed as follows:

$$f(\mathbf{p}) \leftarrow \underset{(l,\theta,s)\in\Psi_\mathbf{p}}{\arg\min}\ \mathcal{V}(\mathbf{p}, l, \theta, s), \qquad (13)$$
$$c(\mathbf{p}) \leftarrow \mathcal{V}(\mathbf{p}, f(\mathbf{p}))$$

where $\mathcal{V}$ is the visual similarity distance defined in (11) and $\leftarrow$ is an assignment operator. Intuitively, the current deformation label $f(\mathbf{p})$ is replaced with the label that provides the smallest matching cost among candidate labels $\Psi_\mathbf{p}$. Also, the smallest matching cost is stored in the distance map $c(\mathbf{p})$, which will be used to guide the field smoothing in the next section. This process helps improve the convergence, since neighboring pixels tend to have similar orientation and scale in natural images. On even-numbered iteration, the propagation is performed in reverse scan order: $\Psi_\mathbf{p} = \{f(\mathbf{p}), f(\mathbf{p}+(1,0)), f(\mathbf{p}+(0,1))\}$.

Random Search. In the second step, a random search proceeds to prevent the estimated parameters from being trapped in local minima. We update the current optimal label $f(\mathbf{p})$ by a sequence of random trials which are constructed by sampling around $f(\mathbf{p})$ at an exponentially decreasing distance as

$$(l^i_\mathbf{p}, \theta^i_\mathbf{p}, s^i_\mathbf{p})^T = (l_\mathbf{p}, \theta_\mathbf{p}, s_\mathbf{p})^T + \alpha^i \mathbf{R}_i \mathbf{Z}, \quad i = 0, 1, 2, \ldots, \quad (14)$$

where $\mathbf{R}_i$ is a 3×3 diagonal matrix whose diagonal entries are uniform random numbers in $[-1, 1]$, α^i is the i^{th} exponential of a ratio $\alpha = 0.5$, and $\mathbf{Z} = (L, \pi, 2.0)^T$ is the maximum search range. The index i increases until the orientation search radius as the second entry in $\alpha^i\mathbf{Z}$ is below 1. Using this sequence, the current label $f(\mathbf{p})$ is refined if the target random pair has a smaller cost. Note that, in terms of energy minimization, our randomized inference method shares similar principles with the recent work of [23], called PatchMatch Belief Propagation (PMBP). Interestingly, the work of [23] showed that the random sampling and propagation steps of PatchMatch [5], [6] are related to steps in a special form of belief propagation.

Fig. 7 shows the intermediate results of the orientation field during iterations. Starting from random orientations of Fig. 7(b), the orientation field is progressively evolved during iterations. As a result, the resultant orientation field is locally aligned with the visual structure of the input texture image as in Fig. 7(d). However, the inference mechanism is inherently local, thus often missing a spatial coherency in the estimation. Instead of using a costly global optimization, we resolve this problem by implicitly imposing the smoothness prior on the deformation field via a local filtering approach, which will be detailed in the following section.

C. Field Smoothing

After the deformation parameters are inferred through the randomized search, the nonlinear vector field smoothing is performed with the guidance of the distance map $c(\mathbf{p})$ as a matching confidence in order to enforce global consistency. We extend the work of [18] by introducing the scale field as well as the orientation field into the smoothing process as a two-tuple flow vector $\bar{f}$. We also introduce the normalized matching cost $\bar{c}(\mathbf{p})$, computed by the randomized inference, as the guide signal for adaptive smoothing, making the smoothing more robust against matching outliers. Intuitively, dominant flow vectors having smaller matching costs are preserved, while weak flow vectors having larger costs are directed to follow neighboring dominant ones. It should be noted that the vector field smoothing is performed on the scale and

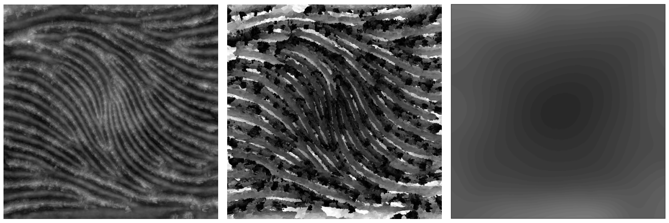

(a) Distance map $\bar{c}$ (b) Initial scale field (c) Refined scale field

Fig. 8. Results on scale smoothing to Fig. 11(a). (a) the normalized distance map $\bar{c}$, (b) initial scale field obtained by the randomized inference, and (c) the refined scale field by our field smoothing.

orientation fields only. It is because the texture label field is just used as intermediate parameters to deal with multi texture images having a certain amount of geometric and/or photometric variations, not to precisely segment perceptually similar regions. The field smoothing is defined as follows:

$$\bar{f}^{m+1}(\mathbf{p}) = \mathbf{K}^{-1}(\mathbf{p}) \sum_{\mathbf{q}\in\mathcal{N}_f(\mathbf{p})} \mathbf{W}(\mathbf{p},\mathbf{q})\bar{f}^{m}(\mathbf{q}), \quad m = 1, ..., \eta, \tag{15}$$

where $\bar{f}^m$ is an intermediate result at the m^{th} iteration, and $\bar{f}^1 = (\theta_\mathbf{p}, s_\mathbf{p})^T$ represents the scale and orientation estimated through the randomized inference. For further enhancing a global coherency of the intermediate flow field, the field smoothing is iteratively applied η times. $\mathbf{K}^{-1}(\mathbf{p})$ is a diagonal matrix for a normalization and $\mathcal{N}_f(\mathbf{p})$ denotes the neighborhood of $\mathbf{p}$. $\mathbf{W}$ is a 2×2 diagonal weighting matrix, which is defined as:

$$\mathbf{W}(\mathbf{p},\mathbf{q}) = \begin{bmatrix} w_r(\mathbf{p},\mathbf{q})w_d(\mathbf{p},\mathbf{q}) & 0 \\ 0 & w_r(\mathbf{p},\mathbf{q}) \end{bmatrix}, \tag{16}$$

where w_r and w_d represent the range kernel and the direction kernel, respectively. The range kernel w_r encourages dominant orientations and scales to be preserved during smoothing, which is defined as follows:

$$w_r(\mathbf{p},\mathbf{q}) = \frac{1}{2}(1 + \tanh(\bar{c}(\mathbf{p}) - \bar{c}(\mathbf{q}))), \tag{17}$$

where $\bar{c}(\mathbf{p})$ represents a normalized matching distance across an entire image. $\tanh(\cdot)$ is a monotonically increasing function with respect to the distance difference $\bar{c}(\mathbf{p}) - \bar{c}(\mathbf{q})$, and thus bigger weights are assigned to the neighboring pixels $\mathbf{q}$ whose matching distances are lower than that of the center $\mathbf{p}$. Accordingly, the pixels having lower matching distances contribute more in the filtering of the flow field. Note that the guide signal for calculating adaptive weight w_r is the matching distance, while the work of [18] uses the gradient magnitude. The direction kernel w_d helps tighter alignment of neighboring orientations, which is defined as:

$$w_d(\mathbf{p},\mathbf{q}) = |\cos(\theta_\mathbf{p} - \theta_\mathbf{q})|. \tag{18}$$

The direction weight increases as the difference of two orientations approaches to 0 or π. Note that w_d is only applicable to the orientation field, since the scale field is not directional. In Fig. 7(e), our smoothing improves global coherency of the orientation field, resulting in a good continuation of texture flows. In addition, as shown in Fig. 8, our smoothing helps remove outliers in the scale estimation as in Fig. 8(b), which are caused by severe variations on the texture appearance.

VI. Experimental Results

In this section, we validate the performance of the proposed method on various texture images including both synthetic and natural photographs. Test images were selected which undergo sufficient deformation such as rotation and scale transformations, and also show a clear distinction enough to be interpreted by the HVS. In addition, they contain at least one coherent region ($L \geq 1$) in order to evaluate the validity of our exemplar extraction method. To verify the applicable extent of the proposed method, we applied it to various types of texture images, which involve regular textures having a lattice structure, near regular textures like line patterns, irregular textures with apparent orientation, and stochastic textures with no obvious orientation. The proposed method was implemented in the MATLAB and was simulated on a PC with Quad-core CPU 2.93GHz. In all experiments, the window radius r and the quantization levels $[q_1, q_2, q_3]$ in computing histogram-based features (Sec. IV-A) are set to $r = 15$ (unless otherwise stated) and $[10, 5, 5]$, respectively. The kernel bandwidth in the medoidshift clustering (Sec. IV-B) is set to $\sigma = 0.2$. The maximum iteration of the randomized inference process (Sec. V-B) is set to 3. For each test image, the window size $\mathcal{N}_f$ and the maximum iteration η of the field smoothing (Sec. V-C) are empirically set in the range of $[15, 25]$ and $[5, 15]$ according to image resolution, respectively. For flow visualization, the line integral convolution (LIC) [24] is used.

Since our method is inherently local, we mainly compare the proposed method with state-of-the-art local methods including the edge tangent filter [18] (ETF) and the statistical invariance (SI) method [17]. In addition, although the objective is slightly different, the deformed lattice detection (DLD) algorithm [9], which aims to discover a lattice structure, is also compared for the evaluation of regular-type textures. It is important to notify that all these flow estimation approaches (except DLD) require a user interaction, while our method is fully automatic.

A. Evaluation on Single Texture Images ($L = 1$)

Fig. 9 shows the estimated deformation fields on both natural and synthetic textures, when the input image has a single texture element. All texture exemplars are automatically extracted by our method, and thus no human assistance is involved in the estimation. The resultant deformation fields are visually consistent with the human perception in terms of scale and orientation. As shown in the top row of Fig. 9, the proposed method can capture the inherent flows of the circular pattern very well. It is also observed that the resulting scale field is represented well as a convex shape that bulges inward. For the input image with a regular-type texture as in the 'Roof tiles' image, our method still produces globally consistent flows.

Fig. 10 presents subjective evaluation results on a synthetic image. The results of Paris *et al.* (Fig. 10(b)) and ETF (Fig. 10(c)) methods are poor since they exploit local gradient

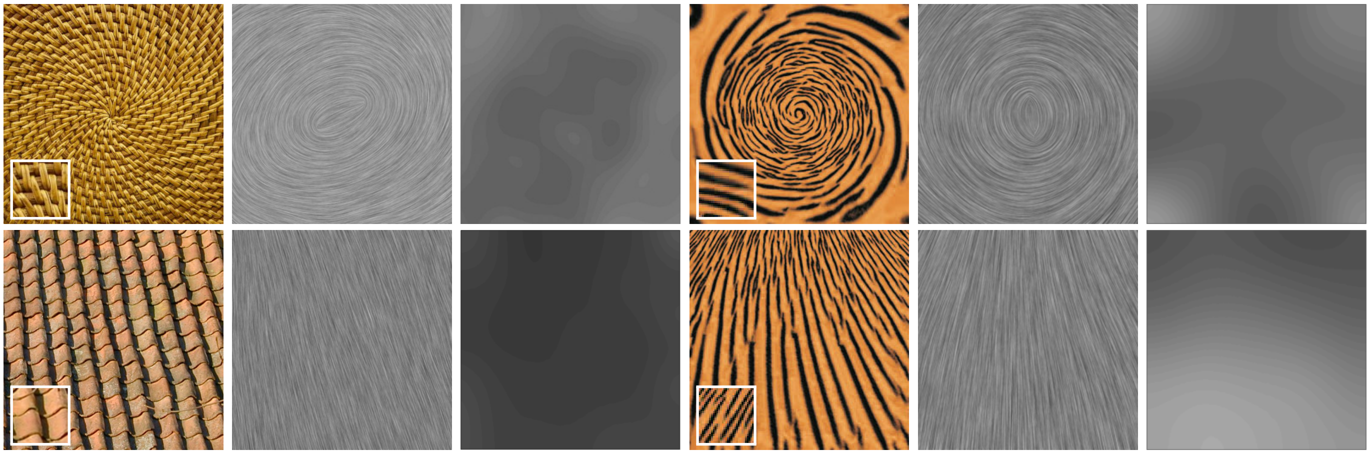
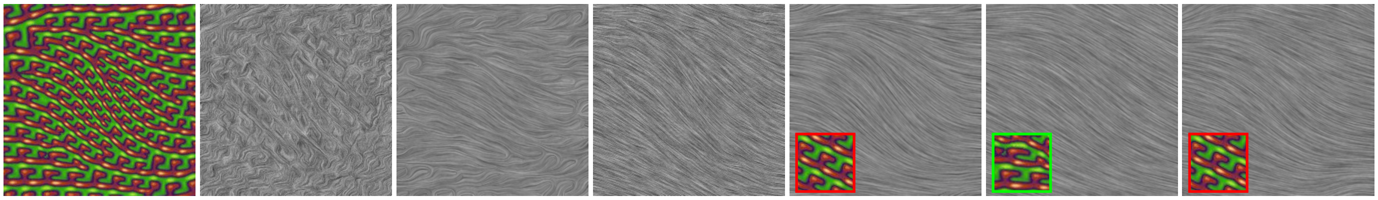
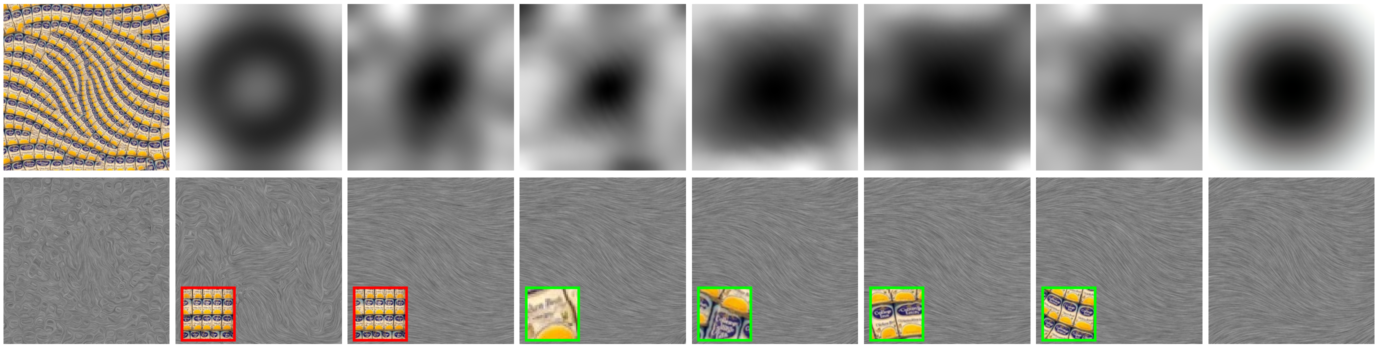


Fig. 9. Experimental results on natural and synthetic images: (from left to right) input texture images ($L = 1$), estimated orientation and scale fields. The extracted texture exemplars are overlaid over the input image.



(a) Input texture (b) Paris *et al.* (c) ETF (d) ITS (e) SI (f) Ours (g) Ours-GT

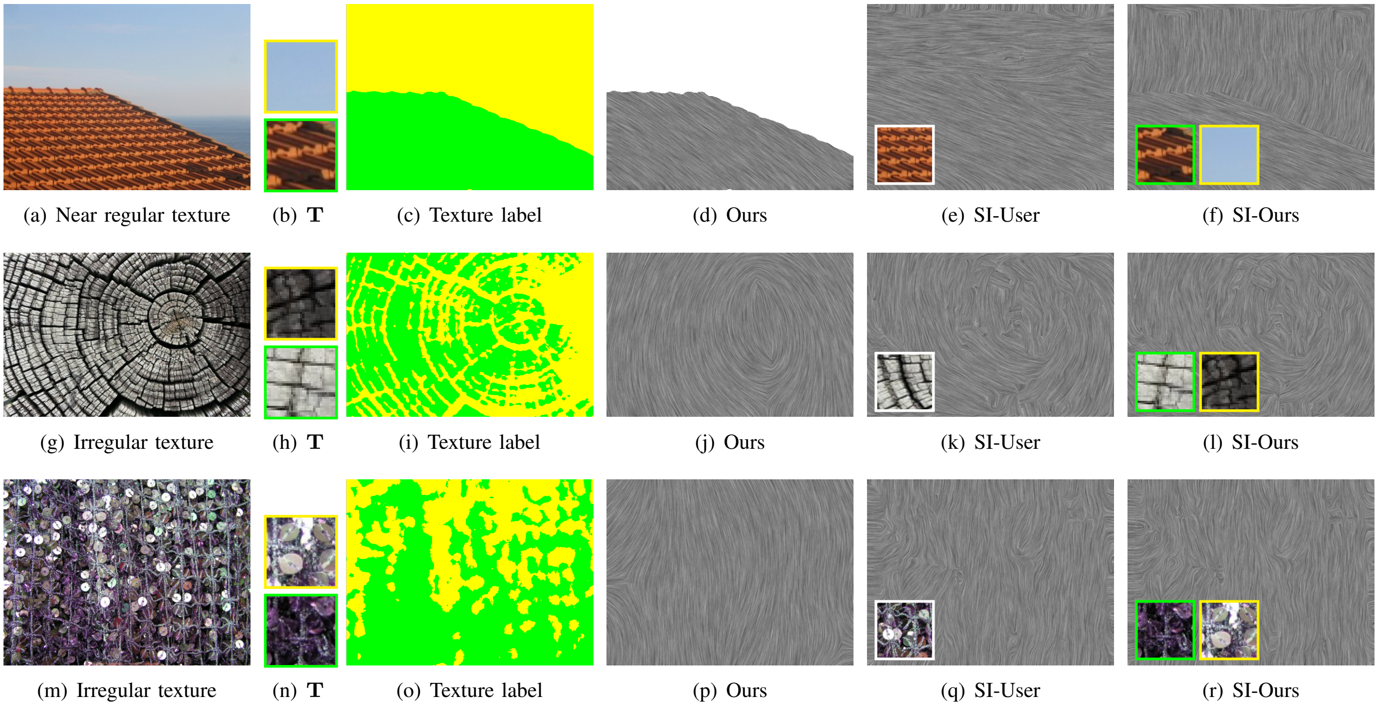
Fig. 10. Comparisons of the proposed method with state-of-the-art local methods. (a) The input texture shows a near regular geometry. (b)-(f) Estimated orientation fields obtained by (b) Paris *et al.* [15], (c) ETF [18], (d) inverse texture synthesis (ITS) [10], (e) SI [17], and (f) the proposed method. Texture exemplars are overlaid over the corresponding flow fields. Note that the input exemplar provided to (e) SI method and (g) the proposed method is the ground truth (GT) with no deformation, while the one provided to (f) ours is automatically extracted from (a) the input image using our automatic exemplar extractor.



(a) ETF (b) SI (c) Ours-GT (d) $r = 15$ (e) $r = 25$ (f) $r = 35$ (g) $r = 45$ (h) GT

Fig. 11. Comparisons of the estimated scale and orientation fields with ground truth (GT) data. (a) Input texture and the estimated orientation field by ETF [18]. (b),(c) Estimated scale and orientation fields by (b) SI method [17] and (c) the proposed method using the ground truth exemplar (red). (d)-(g) Estimated scale and orientation fields by the proposed method using automatically extracted exemplars (green) with (d) $r = 15$, (e) $r = 25$, (f) $r = 35$, and (g) $r = 45$. (h) The ground truth scale and orientation fields. Note that the scale fields are normalized.

information only, and thus the deformable structure of the texture is not taken into account. In addition, although the ETF method [18] employs vector smoothing similar to ours, it fails to capture a desired flow field due to inaccurate local estimates. The inverse texture synthesis (ITS) method in Fig. 10(d) requires a cumbersome initialization using a manual specification of sparse texture flows from a user. Our method shows comparable performance to the SI method of Fig. 10(e). The ground truth exemplar overlaid on Fig. 10(e) should be provided manually in the SI method, since it is designed to use either a carefully selected or a ground truth exemplar. In contrast, our method uses the automatically extracted exemplar as shown in Fig. 10(f). Since the extracted exemplar is sampled directly from the input image, it still contains a certain deformation compared to the ground truth exemplar that are not distorted. Nevertheless, the proposed method outperforms other methods or at least shows a comparable quality, indicating that our texture exemplar defined as a local mode explains the local behavior of the underlying flow field well. When the proposed method uses the ground truth


(a) Near regular texture (b) **T** (c) Texture label (d) Ours (e) SI-User (f) SI-Ours

(g) Irregular texture (h) **T** (i) Texture label (j) Ours (k) SI-User (l) SI-Ours

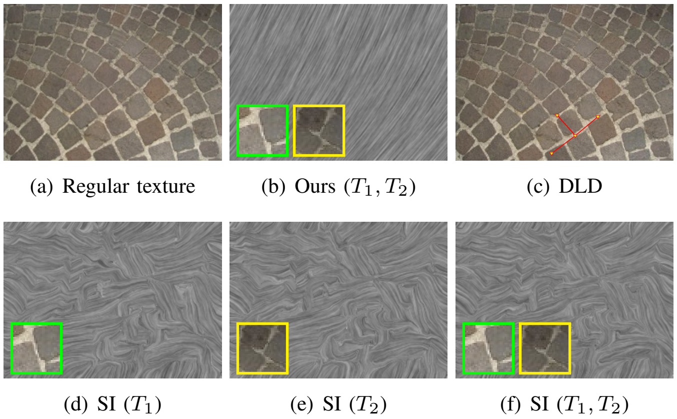
(m) Irregular texture (n) **T** (o) Texture label (p) Ours (q) SI-User (r) SI-Ours

Fig. 12. Results of the proposed method with the SI method [17] in estimating $L = 2$ multi texture images. (from left to right column) Input images, extracted texture exemplars, estimated texture label maps, results of the proposed method, results of the SI method using a user-provided texture exemplar, and results of the SI method using extracted texture exemplars by our method.

exemplar instead of the automatically extracted one, the best performance is achieved as shown in Fig. 10(g). Here, different orientations of two texture exemplars in Figs. 10(e) and (f) do not matter in the visualization of the flow field, since the absolute orientation is displayed using the RO of each texture exemplar, as explained in Section IV-C.

Fig. 11 shows additional flow estimation results. The ground truth exemplar (red) is used for the SI method as well. For quantitative evaluation, we measure the root mean square (RMS) error on the estimated orientation and scale fields against the ground truth data which is generated by the anisotropic texture synthesis algorithm [13]. Note that both estimated and ground truth scale fields are normalized prior to evaluation. Table I presents the results of the quantitative evaluation. The RMS errors of orientation and scale are respectively 5.71° when $r = 25$ and 0.13 when $r = 45$, while other methods exceed about 30° and 0.27, respectively. The test image as in the top row of Fig. 11(a) has complex structures, and thus such gradient-based local methods fail to produce correct flows.

We also evaluated the performance of our method by varying the window radius $r = 15, 25, 35, 45$ when texture exemplars are extracted. As shown in Figs. 11(d)-(g), the scale in the extracted texture exemplars becomes broader as r increases. This, however, does not affect the quality on the final orientation field. The extracted texture exemplar is the most visually similar patch from well-structured regions (interest points), making it discriminative when an orientation field is estimated on appearance space. In case of scale estimation, it is observed that using a large scale exemplar produces slightly better results.


(a) Regular texture (b) Ours (T_1, T_2) (c) DLD

(d) SI (T_1) (e) SI (T_2) (f) SI (T_1, T_2)

Fig. 13. Performance comparison on a regular-type texture. (a) The input texture shows a lattice structure. (b) The estimated orientation fields by the proposed method. The extracted texture exemplars T_1 (green) and T_2 (yellow) are overlaid over (b). (c) The detected lattice structure by DLD [9]. (d)-(f) The estimated orientation fields obtained by SI method [17] using (d) T_1, (e) T_2, and $\{T_1, T_2\}$.

B. *Evaluation on Multi Texture Images* ($L \geq 2$)

Fig. 12 shows the results obtained by our method and the state-of-the-art exemplar-based method [17], when the input image consists of multiple textures. Test images include the near regular texture as in Fig. 12(a), the irregular texture with apparent orientation as in Fig. 12(f), and the stochastic texture with no obvious orientation as in Fig. 12(k). In all test images, the number of extracted texture exemplars is $L = 2$. Different textural elements are combined together in the image, leading to the complex structure. In existing flow

TABLE I
QUANTITATIVE EVALUATION OF FIG. 11 BASED ON RMS ERROR METRIC

Method	Orientation Error	Scale Error
ETF [18]	31.1647°	N.A
SI with GT [17]	33.4883°	0.2713
Ours $r = 15$	8.1788°	0.1977
Ours $r = 25$	**5.7080°**	0.1633
Ours $r = 35$	7.0586°	0.2409
Ours $r = 45$	6.8479°	**0.1252**
Ours with GT	7.5818°	0.1722

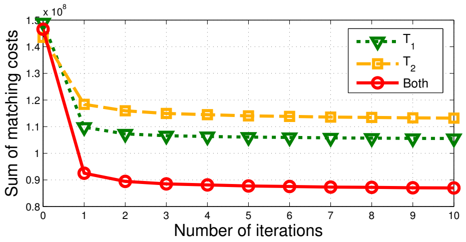


Fig. 14. Cost profiles of our non-parametric sampling process. Given the input image of Fig. 13(a), the sum of matching costs is measured at each iteration of the randomized inference. Using two texture exemplars T_1 (green) and T_2 (yellow) as in Fig. 13(b) gives the lowest matching cost, enabling more accurate and robust matching performance in the existence of texture variations.

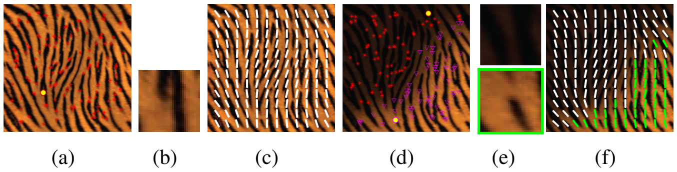


Fig. 15. Results of the proposed method under different illumination conditions. When an input texture under (a) uniform or (d) varying illumination is given, a set of relevant texture exemplars is extracted as (b) $L = 1$ single patch or (e) $L = 2$ multiple patches. Using (e) $L = 2$ exemplars, (f) the desired orientation field is successfully estimated from (d) the image under shadows, which is comparable to (c) the one estimated from (a) a clean texture.

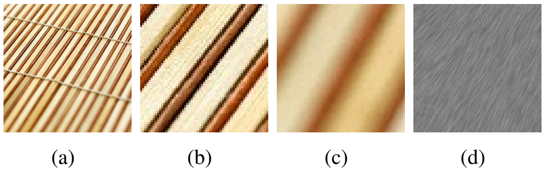


Fig. 16. Results of the proposed method with the input image containing blurred regions. (a) Input image containing blur regions, (b),(c) the extracted $L = 2$ texture exemplars, and (d) the estimated orientation field.

estimation approaches, it is typically assumed that a single exemplar contains discriminative visual cues that covers an entire image. This assumption, however, is unfit to the cases of multi-texture images. Thus, the input image should be manually segmented (*e.g.*, [2], [17]), and a suitable texture exemplar should be provided for each region. Contrarily, our method is directly applicable to these images by automatically estimating deformation fields of each coherent region using corresponding exemplars which represent texture attributes for each region very well. This unsupervised sampling strategy is very effective in that distinct visual cues increase the discriminative power of matching in the inference. As shown in Fig. 12(f), while local regions are highly irregular under substantial contrast changes, our method can cope with such challenging textures as two texture exemplars effectively cover the distinct texture attributes of the image. Moreover, even when the texture is stochastic with no obvious orientation as in Fig. 12(k), our method can produce a globally coherent flow field as in Fig. 12(n). Our per-pixel labeling framework also allows one to estimate texture label parameters as shown in Figs. 12(c), (h), and (m). In Fig. 12(a), the 'sky' regions have no obvious orientation, and thus they are invalid in estimating the orientation field. Such regions are detected in the texture label field (yellow in Fig. 12(c)), where their validity is automatically determined by the RO assignment process. As shown in Fig. 12(d), orientations for the 'sky' regions are not estimated, while the SI method has no ability to validate correct regions. Note that the orientation field estimated by the SI method using a single texture exemplar were shown together in Figs. 12(e), (j), and (o) to show how this method works in multi-texture images.

For fair comparison, we also report the results of the SI method [17] with multiple texture exemplars extracted by the proposed method. We modified the SI method to allow using the same number of exemplars as distinct texture regions detected by our deformation field estimation method. Specifically, the SI method was performed separately on each labeled region, and the resulting flow fields were re-combined to produce the final flow field for the entire image. As shown in Figs. 12(f),(l), and (r), the SI method is not able to produce a coherent flow field even with the multiple texture exemplars that cover each of the pre-labeled regions.

Fig. 13 shows the results for regular-type textures, where the flow estimation is difficult due to an orientation ambiguity. As a result, the orientation field of the SI method [17], which heavily relies on image gradients, is misleaded by ambiguous local structures. Indeed, the flow estimation in regular-type textures can be seen as a special case of a lattice detection [9]. However, if the image undergoes significant orientation and scale changes, they also fail to produce a convincing result as in Fig. 13(d). In contrast, our method captures the inherent lattice-type regularity well. The advantage of our unsupervised sampling strategy can also be explained from energy minimization perspective. Fig. 14 presents cost profiles in estimating the deformation field of Fig. 13(a) using various combinations of input texture exemplars, *i.e.* $\mathbf{T} = \{T_1\}$, $\{T_2\}$, or $\{T_1, T_2\}$. Using $L = 2$ texture exemplars achieves a faster convergence than other single texture exemplar usages. In addition, it gives the lowest matching cost, indicating more accurate and robust matching performance in the existence of texture variations.

Another advantage of our method is a robustness against illumination variations by virtue of the unsupervised sampling strategy. In general, the inference is heavily affected by illumination and exposure changes, in particular by shadows. Existing approaches usually require a clean texture that has uniform illumination [2] or performing an illumination decomposition as a pre-processing [17]. Instead, we deal with this challenge by extracting texture exemplars that can cover particular texture elements of those problematic regions under shadows. Fig. 15 shows a texture image with illumination changes. While a single texture exemplar as in Fig. 15(b) is extracted from the input image under uniform illumination, two exemplars covering the parts under different illumination as in Fig. 15(e) are extracted from Fig. 15(d). Using these appropriate exemplars, the desired orientation field is successfully estimated under shadows as shown in Fig. 15(f). When the image contains blurred regions as shown in Fig. 16, our method is able to produce a desired flow field as well.

C. Trade-off Analysis with Existing Approaches

We analyze the computational efficiency of our method. First, we compare the runtime of our deformation field estimation process with that of existing approaches, since texture exemplars are provided by a user in the existing approaches. Given an input texture of size 256×256 and an exemplar of size 64×64, the running times of deformation estimation are respectively around 4 seconds (ETF) and 10 seconds (SI) on average, while our method takes 79 seconds (75 seconds for randomized inference and 4 seconds for vector field smoothing). For a complex input texture as in Fig. 11(a), these two local approaches fail to produce a coherent flow field. These local methods directly calculate the deformation field with a over-simplified model (*e.g.*, intensity gradient), and thus they run faster than ours but cannot discriminate correct orientations in the presence of complex structure. Moreover, the ETF [18] does not consider the scale field. In contrast, though inherently local, our method formulates the deformation field estimation as the per-pixel labeling framework based on a non-parametric deformation model. This labeling algorithm shares a similar spirit with several global optimization-driven approaches, but our randomized search strategy along with the vector field smoothing enables a much faster inference, with a comparable estimation quality to global approaches. For instance, the global approaches typically take about $10 \sim 20$ minutes [2]. In addition, our inference algorithm requires only a little extra memory for storing a distance map, unlike existing optimization-driven approaches [2], [9] that typically require a huge memory usage to handle the high-dimensional label space.

D. Applications

Estimated deformation fields are directly applicable in the tasks of texture manipulation such as anisotropic texture synthesis [13] and re-texturing [17]. In this section, we introduce two image processing applications based on our deformation fields: flow based image retrieval and unsupervised texture segmentation.

1) Flow-based Image Retrieval: Inspired by the HOG descriptor [25] which shows an excellent performance on detecting objects, we attempt to find the most similar image, where a visual similarity is measured in a slightly different manner from that of existing descriptors. We define the similarity with an inherent regularity in terms of orientation and scale, so that an image, which undergoes a similar deformation to that of a query image, is retrieved among a set of test images.

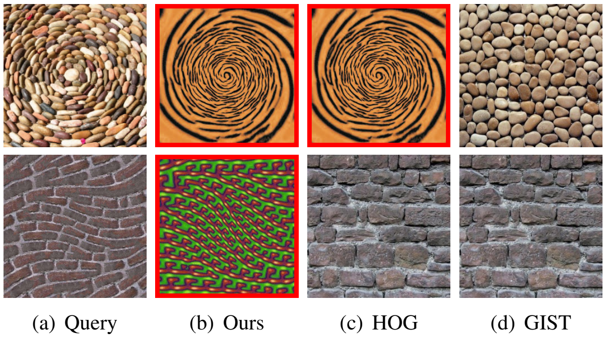
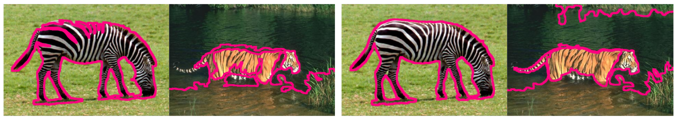


Fig. 17. Flow-based image retrieval. Best matching results obtained by our method, HOG, and GIST are presented in (b), (c), and (d), respectively. The correct retrieval results were marked with a red border.

The inherent regularity of an image is characterized by the distribution of local orientation and scale. For this representation, the image is divided into 4-by-4 regions, called cells, and each cell accumulates a local 1D histogram of orientations over the pixels inside the cell. Each pixel within the cell gives a weighted vote to its corresponding bin of the local histogram. The amount of contribution for voting is determined by the corresponding scale value $s_{\mathbf{p}}(\mathbf{p} \in \Omega)$. The feature vector is then formed with concatenated 16 local histograms. Euclidean distance is measured between feature vectors to retrieve the most similar image. Note that the rationale behind using the scale as a weighting factor in the final feature representation is that this helps capturing of informative orientations. In our experiment, local flows having small scale have very little contributions in retrieving desired image matches. Thus, using larger scale parameters is more effective to represent the local shape of an inherent flow structure. This is similar to the one used in the HOG representation [25]. For example, the HOG determines the presence/absence of informative edges by using the magnitude of gradient.

We simply test the proposed retrieval scheme using 20 manually selected test images. A query image is chosen from the test set, and the most similar image is then retrieved among 19 images. The retrieval results are shown in Fig. 17 with a comparison to the existing approaches using the HOG [25] and the GIST [26]. For subjective evaluation, we experimentally extracted the most visually-similar results, which were most frequently selected by twelve users. In 'Pebble' image containing a simple circular deformation field, the HOG works very well, since such deformation fields are estimated relatively well by using image gradients only. But, when it comes to a query image with more complicated textures, this simple HOG-based approach produces an inaccurate result, *e.g.*, for 'Brick' image. The GIST based approach also fails to retrieve visually similar images, since it was originally developed for representing the general aspect of spatial properties in the image such as the intensity distribution. In contrast, our approach always provides the images with visually similar flows, regardless of the degree of texture complexity.

2) Unsupervised Texture Segmentation: The estimated texture label field $l_{\mathbf{p}}(\mathbf{p} \in \Omega)$ can be used for a better initialization in the unsupervised segmentation algorithm such as [27]. Typical methodologies used for unsupervised segmentation rely



(a) Donoser *et al.* [27]

(b) Ours

Fig. 18. Comparisons of the results on the unsupervised texture segmentation using [27]. (a) The results obtained by the original method in [27] and (b) the results of [27] with the initialization using the estimated label field.

on a representative color model for discriminating dominant regions, and thus they often show unsatisfactory results in grouping highly textured regions, as shown in Fig. 18(a). To be specific, the method in [27] devises the sub-segmentation integration approach where each of sub-segmentation phases automatically computes the Gaussian Mixture model (GMM) representing the color distribution of each region-of-interest (ROI) and each pixel in the image. However, the estimation of ROIs relies only on color distributions. Thus, the original method [27] often fails to estimate accurate segmentation results, when there are complex texture boundaries. We resolve this limitation by deploying the estimated texture label field. The ROIs are initialized by the estimated texture label field, where only reliable texture labels $\tilde{l}_{\mathbf{p}}$ are assigned in the initialization. We define a reliable texture label $\tilde{l}_{\mathbf{p}}$ whose matching cost $c(\mathbf{p})$ is lower than the mean matching cost across the entire image. This enables more coherent estimation of GMMs. Fig. 18(b) shows the segmentation results that are significantly improved by our initialization strategy. Note that our deformation model is based on the visual similarity, and thus the texture label parameter encodes both color and texture distributions.

VII. Discussions and Conclusions

This paper has addressed the inverse estimation of the underlying texture deformation field based on the non-parametric visual correspondence mechanism. The unsupervised sampling strategy in extracting texture exemplars allows one to estimate deformation fields of multi texture images in a fully automatic manner. It is also beneficial to deal with various imaging conditions like non-uniform illumination. The efficient randomized search enables the direct application of the non-parametric deformable texture model to a high-dimensional search space, and the locally-adaptive vector field smoothing provides an excellent alternative for costly optimization based approaches. More importantly, our method is the first automated approach that is capable of estimating the texture flow field in a multi-texture image.

There are some limitations in our approach, though. Our method is not able to produce a convincing result when the input texture undergoes severe affine deformations, *e.g.* shears. Like other exemplar-based methods, our deformation model is established on the typical rotation and scale deformation scenario. Hence, the matching quality might decrease in such severely deformed regions. However, the proposed method can be naturally extended to incorporate the affine deformation model, since the randomized inference is very effective in yielding a solution in the high-dimensional search space [28].

References

[1] O. Ben-Shahar and S. Zucker, "The perceptual organization of texture flow: a contextual inference approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 4, pp. 401–417, April 2003.

[2] Y.-W. Tai, M. S. Brown, and C.-K. Tang, "Robust estimation of texture flow via dense feature sampling," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[3] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 795–802, 2005.

[4] Y. A. Sheikh, E. A. Khan, and T. Kanade, "Mode-seeking by medoid-shifts," in *Proc. IEEE Int. Conf. on Computer Vision*, 2007.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[6] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *Proc. European Conf. on Computer Vision*, 2010.

[7] N. Ahuja and S. Todorovic, "Extracting texels in 2.1d natural textures." in *Proc. IEEE Int. Conf. on Computer Vision*, 2007.

[8] J. Hays, M. Leordeanu, A. A. Efros, and Y. Liu, "Discovering texture regularity as a higher-order correspondence problem," in *Proc. European Conf. on Computer Vision*, 2006.

[9] M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu, "Deformed lattice detection in real-world images using mean-shift belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1804–1816, 2009.

[10] L.-Y. Wei, J. Han, K. Zhou, H. Bao, B. Guo, and H.-Y. Shum, "Inverse texture synthesis," *ACM Trans. Graph.*, vol. 27, no. 3, p. 52, 2008.

[11] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. on Computer Vision*, 1999.

[12] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. ACM SIGGRAPH*, 2001.

[13] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 541–548, 2006.

[14] Y. Liu, W.-C. Lin, and J. Hays, "Near-regular texture analysis and manipulation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 368–376, 2004.

[15] S. Paris, H. M. Briceño, and F. X. Sillion, "Capture of hair geometry from multiple images," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 712–719, 2004.

[16] J. Chang and J. Fisher, "Analysis of orientation and scale in smoothly varying textures," in *Proc. IEEE Int. Conf. on Computer Vision*, 2009.

[17] X. Liu, L. Jiang, T.-T. Wong, and C.-W. Fu, "Statistical invariance for texture synthesis," *IEEE Trans. Vis. and Comput. Graph.*, vol. 18, no. 11, pp. 1836–1848, 2012.

[18] H. Kang, S. Lee, and C. K. Chui, "Flow-based image abstraction," *IEEE Trans. Vis. and Comput. Graph.*, vol. 15, no. 1, pp. 62–76, 2009.

[19] M. Hein and O. Bousquet, "Hilbertian metrics and positive definite kernels on probability," in *Proc. IEEE Int. Conf. on Artificial Intelligence and Statistics*, 2005.

[20] K. Grauman and T. Darrell, "Unsupervised learning of categories from sets of partially matching image features," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[21] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. European Conf. on Computer Vision*, 2004.

[22] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[23] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "PBMP: Patchmatch belief propagation for correspondence field estimation," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 2–13, 2014.

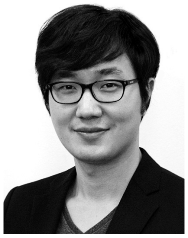
[24] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," in *Proc. ACM SIGGRAPH*, 1993.

[25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[26] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[27] M. Donoser and H. Bischof, "Roi-seg: Unsupervised color segmentation by combining differently focused sub results," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[28] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo-stereo matching with slanted support windows." in *Proc. British Machine Vision Conference*, 2011.

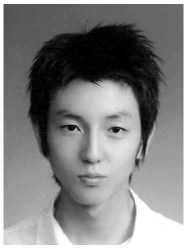


Sunghwan Choi (S'10) received the B.S. degree in electronic engineering and avionics from Korea Aerospace University, Gyeonggi-do, Korea, in 2009, and the Ph.D. degree in electrical and electronic engineering from Yonsei University, Seoul, Korea, in 2015. He is currently a Senior Research Engineer of the Smart Car R&D Laboratory, LG Electronics, Seoul, Korea.

His research interests include computer vision, 2D/3D video processing, computational photography, augmented reality, computational aspects of human vision, and image-based modeling and rendering.

Dongbo Min (M'09) received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, in 2003, 2005, and 2009, respectively. From 2009 to 2010, he was with Mitsubishi Electric Research Laboratories as a Post-Doctoral Researcher, where he developed a prototype of 3D video system. From 2010 to 2015, he was with the Advanced Digital Sciences Center, Singapore, which was jointly founded by the University of Illinois at Urbana-Champaign and the Agency for Science, Technology and Research, a Singapore Government Agency. Since 2015, he has been an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, Korea.

His research interests include computer vision, 2D/3D video processing, computational photography, augmented reality, and continuous/discrete optimization.



Bumsub Ham (M'14) received the B.S. and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea, in 2008 and 2013, respectively. He is currently a Post-Doctoral Research Fellow with Willow Team of INRIA Rocquencourt, École Normale Supérieure de Paris, and Centre National de la Recherche Scientifique.

He was a recipient of the Honor Prize in the 17th Samsung Human-Tech Prize in 2011 and the Grand Prize in Qualcomm Innovation Fellowship in 2012.

His current research interests include computer vision, computational photography, and machine learning, in particular, regularization, graph matching, and super-resolution, both in theory and applications.

Kwanghoon Sohn (M'92-SM'12) received the B.E. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1983, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1985, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1992.

He was a Senior Member of the Research Staff with the Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, Korea, from 1992 to 1993, and a Post–Doctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. He was a Visiting Professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently a Professor with the School of Electrical and Electronic Engineering, Yonsei University.

His research interests include 3D image processing, computer vision, and image communication. He is a member of the International Society for Optics and Photonics.