Texture Instance Similarity via Dense Correspondences

Tal Hassner[1,2] Gilad Saban[1] Lior Wolf[3]

[1] The open University of Israel [2] USC / Information Sciences Institute [3] Tel-Aviv University

Abstract

This paper concerns the task of evaluating the similarity of textures instances: Rather than discriminating between different texture classes, our goal is to identify when two images display the same texture instance. To address this problem, we propose an approach inspired by alignment based recognition theories. We offer a pixel-based method, employing a robust, dense correspondence estimation engine, applied to an efficient, novel representation, to match the pixels of two texture photos. We describe means for quantifying the quality of these matches, considering in particular the quality of the flow established between the two images. These quality measures are effectively combined into similarity scores by using standard linear SVM classifiers. By relying on a general, alignment based approach our method can be applied to different problem domains (different texture classes) with little modification. We demonstrate this by reporting state-of-the-art results on benchmarks for fingerprint recognition and two new benchmarks for texture-based animal identification.

1. Introduction

Nature is abundant in examples of texture classes. These can be as large as a zebra's stripes or as small as the intricate details of our fingerprints. The differences between different texture instances, from one individual to the next, can, and often are, used in order to distinguish between them. This fact has long since been exploited in biometrics, in order to identify humans using their fingerprints, retinal patterns, and more. Not surprisingly, many animals use similar means to distinguish between members of their own species. One such example are zebra finches (Taeniopygia guttata). The males of this monogamous bird species present detailed textures on their chests (Fig. 1). Studies show that female finches are capable of recognizing their life-long mates based on such appearances alone. In both examples, it is the unique features of texture instances that capture the differences between individuals.

This problem is challenging, mostly because texture classes often exhibit *small* intra-class variations, and it is these small variations that must be identified in order to distinguish between different instances of the same texture. Over the years, it has received considerable attention, but mostly from the perspective of fingerprint recognition, likely due to its many applications in biometrics and security. Typical fingerprint recognition methods, however, use man-tailored features, i.e., *minutiae*, extracted from print images. These features include various structures unique to human fingers, such as "whorls" and "loops," which can be compared between different prints in order to determine the identity of the fingerprint's owner (e.g, [4, 5, 32], and many others). Of course, such features are often not defined (or do not exist) for other texture classes.

Our own approach instead considers how well two images can be aligned as its underlying measure of similarity.

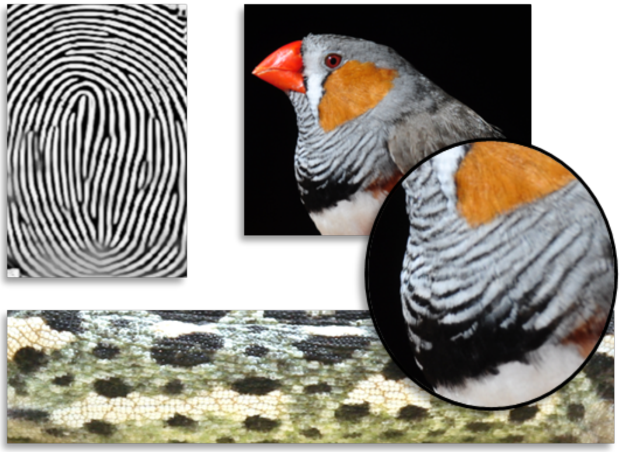


Figure 1: **Examples of texture instance recognition problems.** Fingerprints (top left) are one example of the problem of distinguishing between textures from the same class using fine details. Male Zebra finches (top right) are identified by their female partners using the texture patterns on their chests. Chameleons change their textural patterns as a method of communication (bottom). Here, we present a method which is agnostic to the class of textures being compared, yet effective enough to obtain state-of-the-art performance on standard fingerprint recognition challenges.

The notion that alignment and recognition are fundamentally related has been suggested in the past, yet almost entirely by using rigid alignment and structured objects. Here, we turn to methods previously reserved mostly for optical flow estimation in stereo and motion analysis tasks in order to address non-rigid, unstructured, texture classes.

Our key observation is that dense, non-parametric, matching of two texture photos provides a rich source of information on their similarity. This information is expressed by multiple measures of the quality of these matches: Intuitively, if the two photos present the same texture instance, we expect that it would be easy to match every pixel in one, to every pixel in the other. Here, “easy” is expressed by the similarity of the warped photos, the smoothness of the flow, and more. These measures, when considered in unison, provide an effective novel means of determining if the same texture instance appears in both photos.

To realize our approach, we offer these contributions:

- **Efficient novel representation.** We describe a dense descriptor produced by *exploding* [34] LBP codes [31]. The resulting representation is half the size of the SIFT descriptor (128D) [27], often used for similar tasks, yet proves more effective when applied to the task at hand.

- **Flow-based similarity.** We evaluate image similarity by explicitly considering the quality of the per-pixel flow between two images. To our knowledge, this is the first principled attempt to use dense correspondences and flow quality as a similarity measure [16]. We show this information to be highly informative.

- **General pipeline.** Out system provides a novel approach to fingerprint recognition, very different from the many traditional, minutia-based approaches which dominate this field, and which we outperform. Importantly, unlike these earlier fingerprint recognition methods, we show that our method can be effectively applied to other texture recognition tasks.

2. Related work

This paper touches upon several domains, typically treated separately. Due to the vast literature in each of these fields, only a brief survey is provided here.

Texture classification. Classifying textures has been considered separately from the general problem of image classification due to the unique properties of textures [14]. Textures are assumed to be products of statistical processes, both local (the value of a particular pixel depends on those of its spatial neighbors) and stationary (this relationship does not depend on the spatial position of the pixel). Textures often exhibit small inter-class variability; different texture classes can appear similar, making their classification challenging. Recently, there has been a renewed interest in this problem, due to both its many applications (e.g., material classification in [25]) as well as the development of robust and invariant texture descriptors, including [13, 36].

We believe that the assumptions that are at the heart of texture classification make existing methods ill-suited for the problem at hand. This, as they are all designed with a specific goal of being invariant to precisely the small differences that distinguish individual texture instances.

Fingerprint recognition. Fingerprints are a particular class of textures [20] which has been studied extensively for very obvious reasons. For a comprehensive overview of the field, we refer the reader to [28]. Automatic methods for fingerprint recognition can roughly be classified as either of the three general approaches: Finger-ridge analysis methods [12] which consider features such as ridge counts, local ridge frequencies etc.; Feature-based methods seek particular fingerprint structures (i.e., minutia), represent and then match them between print images [5, 41]; finally, correlation based methods which compare the two images, as a whole, by superimposing the two images and measuring the differences between them [1]. The method proposed here can be viewed as an example of the latter method type. Despite minutia based methods currently being considered the state-of-the-art [4, 38], we surpass these results with a global approach by using a non-linear alignment method between the two images.

Alignment and recognition. Our approach follows what may be considered a “recognition by alignment” approach. Such methods include the correlation based methods for fingerprint recognition discussed above, but also methods ranging from the early work of Huttenlocher and Ullman [18] to the more recent object recognition by scene alignment of [33]. These methods, however, were designed for structured object classes and scenes, and rely on global image representations and parametric transformations. Moreover, these methods do not explicitly consider properties of the transformation itself (in our case, the per-pixel flow) as a means for determining image similarity.

3. Overview of our approach

Problem definition: We are given two texture images from the same class, I and I'. Our goal is to determine if they are both images of the same texture instance, or not. For simplicity, we assume these are grayscale images, leaving treatment of color for future work, and both are of the same dimension $n \times m$. As mentioned before, we assume that if the photos are both of the same texture instance then matching each pixel $p \in I$ to each pixel $q \in I'$ (and vice versa), would be *easy*. Here, we consider in particular the *warp* between the two images; that is, we assume that

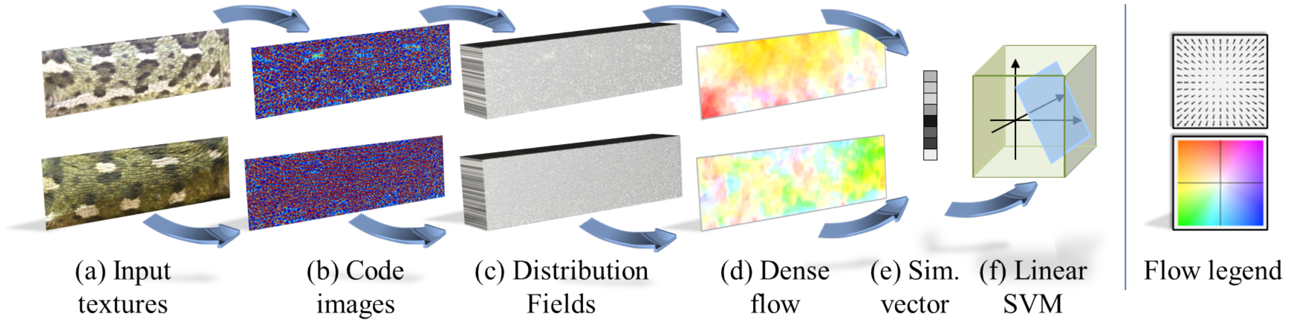


Figure 2: **Method overview.** (a) Input textures; (b) their color-coded code images; (c) *exploding* the code images results in a novel, 59D, per-pixel representation; (d) dense flow established between the two images; (e) 8D similarity vector computed from flow fields and warped images; (f) linear SVM is applied to classify the similarity vectors as "same" or "not-same". Right: color legend for the flow images in (d).

two images of the same subject can be matched with minimal and smooth pixel displacements, and would produce warped images very similar to each other. To accurately form these correspondences and exploit our assumptions, we take the following steps, illustrated in Fig. 2 and detailed in the next sections.

We begin by converting both images to LBP code images [31] C and C' (Fig. 2 (b)). This step assigns each pixel in both images a single integer code representing the micro-texture around that pixel (Sec. 4.1). Next, we produce our novel, per-pixel representation by *exploding* [34] the two code images (Fig. 2 (c)), obtaining distribution fields D and D', where each pixel is now represented as a 59D vector (Sec. 4.2). These per-pixel representations are compact, allowing them to be efficiently matched from one image to the other, but are nevertheless well suited for representing local texture information. Note that LBP or modern variants and [34] by themselves perform poorly on the task discussed here, as shown in Sec. 5.

Correspondences from D to D' and vice versa are then established, using the robust SIFT flow algorithm of [26] (Fig. 2 (d)). This allows examining the flow between the images, its quality, and the similarity of the warped images. This similarity is quantified using a number of measures designed to reflect the quality of the obtained two-sided flows (Fig. 2 (e) and Sec. 4.3). These measures are used to classify the original image pair, I and I', as belonging to the same instance (label of "same") or not ("not-same") using a linear SVM [7] classifier (Fig. 2 (f)).

Global alignment. As with methods designed for other recognition tasks (e.g., face recognition [15]), we rely on an initial global alignment of the two images. Here, we assume only coarse alignment and use standard tools to obtain it. The alignment process applied to the images of different problem domains and the sensitivity of our method to alignment errors are discussed in Sec. 5.

There may presumably be texture classes where this assumption would be less justified; images of some texture classes may be harder to align than others. Here, we limit ourselves to problem domains where this is not the case, and good-enough alignment can be obtained by considering either the texture itself (e.g., fingerprints) or additional contextual information (e.g., zebra finches, chameleons). This class of texture based identification problems is diverse and general enough, yet, as far as we know, there is no alternative general solution that is suitable.

4. Detailed description of our method

4.1. From images to codes

Local binary patterns (LBP) [31] are a successful texture image representation. They assign each pixel with single integer codes, reflecting the micro-texture around it. Their successful application in many texture classification tasks is the primary reason for our choosing it here, though we use it very differently than previous work. For completeness, we next briefly describe these codes.

The LBP code assigned to each pixel is produced by sampling s values g_r, $r = 1..s$, evenly distributed on a ring centered on the pixel, and thresholding their values by the central pixel's value, g_c. This provides s bits, signifying, for each sample, if $g_r \geq g_c$ (1) not (0). This s-bit string is taken as the binary representation of an LBP code and assigned to that pixel. For $s = 8$ samples, LBP would therefore be an 8-bit binary code, or an integer value from $[0..255]$.

In practice, it was observed that only few of the possible code strings capture meaningful appearance information, useful for classification [31]. These were termed "uniform" codes, and defined as codes for which the binary string has only few transitions from '1' to '0' bits, or back. In a typical implementation, and the one used here, no more than two such transitions are allowed (i.e., LBP^{U2}), resulting in a code range of 58 values produced by uniform codes, and an additional value representing all other (non-uniform) bit strings (59 values). Our system uses the original LBP implementation from [31], with all its parameters kept unchanged (in particular, $s = 8$).

Following the success of the original LBP codes, quite a few similar codes were proposed, with different advantages over LBP in different tasks. In our scheme, these other codes can serve as substitutes for LBP, to different effects. We have experimented with several such alternatives, but chose LBP as it was consistently better than these others. One such comparison is provided in Sec. 5.

4.2. From codes to dense descriptors

We define our novel, per-pixel descriptors, the *distribution fields LBP* (DFLBP) codes, by computing a Distribution Field (DF) over the codes assigned to each pixel. DF generalize many representations used in computer vision. The SIFT [27] and the HOG descriptors [9], for example, can both be considered DFs, wherein each pixel is represented by distributions of gradients, or their orientations, around that pixel. Our own work is somewhat more related to the definition of DF given by [34]for the purpose of tracking objects though a sequence of images.

The DF D and D' are two matrices of dimension $n \times m \times d$, where $n \times m$ is the spatial dimension of the two code images C and C' (also I, I'), and d, the number of bins in the distribution. This last value, d, represents the number of possible values each pixel in C abd C' can take. D is produced from C (and D' from C') by *exploding* it: Each pixel is replaced by a single-spike distribution, a Kronecker delta with d bins. That is, DF D is produced from code image C, (similarly, D' from C') as follows

$$D(i,j,k) = \begin{cases} 1, & \text{if } C(i,j) == k \\ 0, & \text{Otherwise} \end{cases} . \tag{1}$$

Here, k represents the LBP code value at pixel $C(i,j)$, and can take a value from $[0..58]$. Following this step, 2D Gaussian smoothing with standard deviation σ is applied along the spatial dimensions of the image ($n \times m$) for each of the d bins. This smoothing is performed for two reasons: first, to reflect the uncertainty due to small, local motion; second, to infuse the representation of each pixel with information regarding the codes assigned to its surrounding neighbors. In all the experiments reported in Sec. 5, we used $\sigma = 1$, without changing this value.

We note that despite being an extremely popular image and texture representation for over a decade, to our knowledge, extracting dense, per-pixel representations from LBP codes, in the manner described above, is novel. Sec. 5 demonstrates the effectiveness of this representation by comparing it with the more widely used Dense SIFT [40] representation.

4.3. Measuring texture similarity

Following global alignment, we assume that images of the same texture should be easy to match at the per-pixel level. Underlying this assumption is the observation that textures can change in appearance – but only to a limited extent. Thus, if the same texture appears in two images, only limited, local deformations are required in order to match the features of the texture in one image to those of the other. To make this observation concrete, we next describe similarity measures which seek to determine just how much a texture image deformed in the process of matching its pixels to another texture image.

Following our encoding, we match pixels of one image to those of another using the SIFT flow of [26]. SIFT flow seeks a flow (a warp),

$$W = \{w_{i,j}\}_{i=1..n,j=1..m} = \{[u_{i,j}, v_{i,j}]\}_{i=1..n,j=1..m},$$

assigning each pixel in the $n \times m$ image I a matching pixel with a similar representation in I', while preserving spatial discontinuities. SIFT flow originally matched Dense-SIFT descriptors as per-pixel representations. Here we instead use our own, novel representation of Sec. 4.2. The advantages of using our representation over the original Dense-SIFT are demonstrated empirically in Sec. 5.

The warp W from I to I', allows transfer of intensities from I' back to I, by the "unwarp" process, defined as:

$${I'}^{w}_{i,j} = I'_{i+u_{i,j},j+v_{i,j}}.$$

Following this step, the image ${I'}^{w}$ holds the intensities of image I', warped to spatial locations of their matching pixels in I. We extract our similarity measures by considering both this warped image ${I'}^{w}$, and the warp itself W, as well as the warped image $I^{w'}$ and its warp W', obtained by computing the flow from I' back to I.

Table 1 summarizes our similarities. The warped image ${I'}^{w}$ (I^{w}) is compared with the target image I (I') using the cosine similarity (Table 1, row 1) and L2 norm (row 2). These values reflect how well flow succeeds in matching the appearance of one texture to the other, (illustrated in Fig. 3). In addition, we consider the smoothness and the magnitude of the flow displacements. These are measured by the L2 norms of the flow vectors (Table 1 row 3) and the L2 norm of the flow-vector gradients (row 4).

We compute these distances for the warp from I to I' and the warp obtained from I' to I. This, as the flow returned by SIFT flow is not guaranteed to be symmetric. Thus, we represent the similarity of two textures using an 8D, real-valued vector. In order to ensure symmetry of

Table 1: **Appearance and flow-based similarity measures.** Here, $\mathrm{v}(A)$ denotes the vectorized form of matrix A and $||A||_F$ its Frobenious norm. U and V are the decompositions of W into two, $n \times m$ flow matrices, storing for each image pixel $p_{i,j}$ the displacements $u_{i,j}$, and $v_{i,j}$, resp.

Appearance:	
1 Cosine sim.	$\frac{\mathrm{v}(I'^{w})^T \mathrm{v}(I)}{\sqrt{\mathrm{v}(I'^{w})^T \mathrm{v}(I'^{w})}\sqrt{\mathrm{v}(I)^T \mathrm{v}(I)}}$
2 L2 appearance	$\|\|I'^{w} - I\|\|_F^2$
Flow:	
3 Small disp.	$\|\|U\|\|_F^2 + \|\|V\|\|_F^2$
4 Smooth flow	$\|\|\frac{\delta U}{\partial x}\|\|_F^2 + \|\|\frac{\delta U}{\partial y}\|\|_F^2 + \|\|\frac{\delta V}{\partial x}\|\|_F^2 + \|\|\frac{\delta V}{\partial y}]\|\|_F^2$

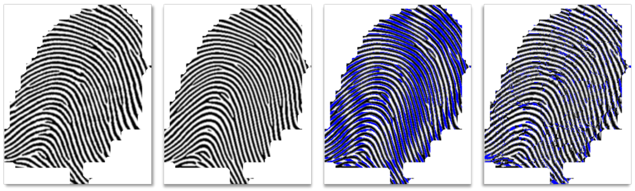


Figure 3: **Effect of warping fingerprints.** (a-b) Aligned, pre-processed prints from the same subject; (c) their overlay, showing severe misalignments at the pixel level (in blue); (d) overlay after warp (Sec. 4.3) almost completely removes these mismatches.

comparisons, each pair of values, from I' to I and vice versa, is sorted smaller-value-first in the 8D vector. A final similarity score is then obtained from a linear SVM [7], trained on labeled same/not-same pairs.

Relation to existing work. It is worth noting that SIFT flow itself, has been used in the past to compute image similarities [26]. Here, we compare our approach to theirs, and show that our system is substantially more accurate (Sec. 5).

Also noteworthy is the relation to [3]. Like us, they compare two images by considering how easy it is to produce one image from the other. Our method, however, explicitly considers the coefficients of the warp between the two images, not just appearance similarities. Also, their method is not designed to distinguish between different texture instances as it relies on structured shapes.

Finally, the use of multiple features for classification has recently been explored by others in the related task of material classification [35]. Here, however, we use multiple similarity scores extracted by considering the same features, with a novel emphasis on the flow between the two images as the key to determining their similarity.

5. Experimental results

Our system was implemented in MATLAB, using libSVM [6], the LBP code from [31], and the original SIFT flow code of [26]. All parameters were left at their default values, unchanged. Thus, better results may possibly be obtained by parameter optimization. To efficiently produce distribution fields, we employed the fast, approximate Gaussian smoothing of [22], which applies multiple, multiscale mean filters computed using integral images. The distribution fields were then linearly normalized to the range of [0..255] and stored as 8-bit integer values. The resulting implementation is fast, requiring about a second to process and compare two images in all the tests reported here, on a standard PC with no optimization or specialized hardware.

We present results on three tasks, representing three very different problem domains. For fingerprint recognition we use the FVC2004-DB1 benchmark [29]. In answer to problems independently raised by two separate groups of biologists, we perform tests on images of chameleons and zebra finchs. Chameleon images represent 13 different behavior patterns, and the task is to determine which one was being broadcast by the chameleon males. Zebra finch images include twenty male individuals, and the task is to identify each male based on the zebra-like pattern on its chest. Both these sets are smaller then some of the large contemporary computer vision benchmarks, yet they reflect real problems and real data, painstakingly collected by domain experts.

In all our tests, we report the error rate (ERR), False Match Rate (FMR), lowest false rejection rate (FRR) for FMR$\leq 1\%$ (FMR100), and same with FMR$= 0\%$ (FMR0). In all cases, lower values are better.

The following methods are compared in all our tests.

1. **SIFT flow (min):** Original SIFT flow [26], with the minimal energy of its cost function, computed for two warps W and W', used as a similarity score.

2. **SIFT flow (mean):** Same as (1), using the mean energy between the two flows instead on the minimum. SIFT flow results, using both the mean and min of its energy, are presented in order to show the improvement of our combined representation and similarity score over its original energy function.

3. **CLBP**: A recent successful LBP variant, developed for texture classification [13]. We use their full method for comparing textures and making same/not-same decisions: Images are represented as histograms of CLBP codes, and classification is performed according to the distances between these histograms. We include this method to show how texture classification methods perform in texture instance recognition tasks.

We additionally tested the following variants of our method.

4. **Us + SIFT:** Our similarity of Sec. 4.3, using Dense-SIFT (DSIFT) descriptors. That is, steps (b-c) in Fig. 2 are replaced with extraction of standard DSIFT features [40]. These results are provided in order to evaluate the contribution of our novel representation to the overall recognition performance.

5. **Us + CLBP:** Our full scheme using CLBP codes [13] (step (b) in Fig. 2 replaced by computing CLBP codes instead of LBP codes). We produce two separate distribution fields for their two code images, $CLBP\_S$ and $CLBP\_M$, and concatenate them into a single $n \times m \times d$, $d = d_s + d_m$ representation. We tested several of their variants, and show the one which provided the best space-performance trade-off – the rotationally invariant, uniform CLBP codes – requiring an $n \times m \times 20$ dimensional representation.

6. **Us + DFLBP:** Our full scheme using exploded LBP codes (Sec. 4).

Tested, but not shown is our system using the following two alternative per pixel representations: non-smoothed DF (i.e., sparse, binary vector representations) and vectorized, 7×7 neighborhoods of LBP codes. Both achieved very low results and were therefore excluded from our results.

5.1. Fingerprint recognition

We test our method on the FVC2004-DB1 benchmark, still considered today the most challenging of its kind [24], using the protocol defined by that contest. Training is limited to ten individual fingers, each one represented by eight prints, in set “B.” The test set, “A,” includes 100 individuals, again, with eight prints each. All images were first pre-processed using standard fingerprint enhancement routines [17], and publicly available code [23]. Prints were then aligned using the shape-context based method of [1], using the authors’ own code[1]. The same alignment was used in all tests, excluding results reported by others (e.g. [2, 12, 38]), which used their own, newer and presumably better alignment methods. Four example fingerprint images are provided in Fig. 4 (left).

Training was performed using all available pairs of same images in set B; 280 image pairs. For the not-same pairs, we used only the first image of each individual, coupling it with the others, for 360 not-same pairs. Following training with SVM, testing was performed by evaluating the performance on set A. Here, again, all possible same pairs were used (2,800), and again, only the first image of each individual, was coupled with the first image of all other individuals to assemble not-same test pairs.

Results are presented in Table 2, and ROC curves in Fig. 5 (left). We provide also performances of the following recent methods, developed exclusively for this task.

7. **Cao et al.** [4] who proposed an improved minutia-based representation and classification scheme.

8. **Gottschlich** [12] who build on a commercial fingerprint recognition engine by proposing better print enhancement. Only equal error rates (ERR) were published for this method.

9. **Bartunek et al.** [2], who proposed their own enhancement method. Recognition was performed using the NBIS software for fingerprint recognition developed by NIST [30].

10. **Sutthiwichaiporn and Areekul [38]**, who presented yet another print enhancement technique, classifying using a previously published method [19]. Here too, only equal error rates (ERR) were reported.

11. **Peralta et al.** [32] where a method for pre-filtering minutia features is described and shown to improve both accuracy and speed of print recognition. No FMR0 scores are reported for their method.

Remarkably, although our method (row 6) was not specifically designed for fingerprint recognition (as those in rows 7,8), it outperforms specialized methods with a noticeable gap. This gap is far greater when comparing our method with the texture classification method of [13] (row 3). Despite being designed for the seemingly related task of texture classification, it does poorly here due to the pooling of codes over image regions, a process which masks results in a loss of information valuable for discriminating between different texture instances. Finally, our suggested representation (Sec. 4), though half the size of the SIFT descriptor, achieves better performance in these tests (row 6 vs. row 4).

Also notable is the advantage of the proposed descriptors over the more traditional dense SIFT. We believe the reason for this is that SIFT low-pass filters the image intensities directly, hence potentially losing high frequency information. This information is particularly vital for textured images where it is the high frequency details which often distinguish between different texture instances. Our DFLBP avoids this by smoothing distribution fields rather than images directly.

Effect of global alignment. It is interesting to note the effect global alignment has on the system’s performance. Specifically, our errors were due entirely to extreme errors in the alignment step. This is significant, as the same or more recent alignment [2, 12, 38] was used by others, but with lower accuracy.

[1] www.mathworks.com/matlabcentral/fileexchange/29280-ngerprint-matching-algorithm-using-shape-context-and-orientation-descriptors

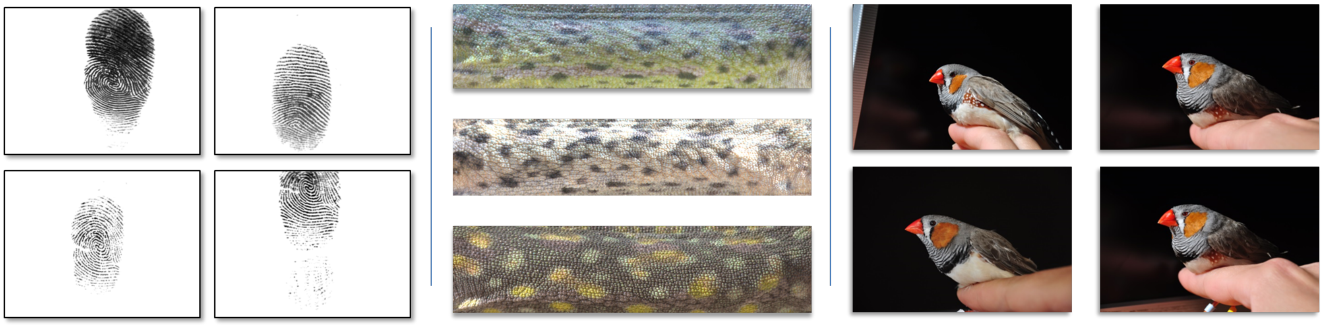


Figure 4: **Example images from our benchmarks.** Left: Fingerprints from the FVC2004-DB1 benchmark [29]; mid: male Chameleon body textures; right: male Zebra finches. In each of these sets, one of the images belongs to a different class than the others. Despite being very similar, forensic experts and specialist Biologists can typically spot the odd one in each set.

Table 2: **FVC2004-DB1 results.**

Method	ERR	FMR100	FMR0
1. SIFT flow (min)	36.5	97.6	99.7
2. SIFT flow (mean)	35.8	97.5	99.7
3. CLBP	44.8	96.5	99.7
4. Us + SIFT	4.9	9.8	20.5
5. Us + CLBP	9.5	27.8	66.4
6. Us + DFLBP	4.4	9.2	20.5
7. [4], 2012	8.2	16.7	38.6
8. [12], 2012	8.9	N/A	N/A
9. [2], 2013	9.6	18.9	30.9
10. [38], 2013	5.4	N/A	N/A
11. [32], 2014	7.0	13.0	N/A

Table 3: **Chameleon results.**

Method	ERR	FMR100	FMR0
1. SIFT flow (mean)	51.0	97.5	99.5
2. SIFT flow (min)	48.0	97.5	99.5
3. CLBP	46.7	98.5	99.0
4. Us + SIFT	44.0	99.0	99.5
5. Us + CLBP	40.5	98.5	99.5
6. Us + DFLBP	36.2	97.5	99.0

Table 4: **Zebra Finch results.**

Method	ERR	FMR100	FMR0
1. SIFT flow (min)	28.5	81.8	90.0
2. SIFT flow (mean)	28.5	82.5	90.0
3. CLBP	44.2	76.2	96.8
4. Us + SIFT	27.2	71.2	87.5
5. Us + CLBP	28.0	76.2	96.2
6. Us + DFLBP	21.4	65.0	90.0

In all three tables, lower values are better, shaded cells are best scores. Please see text for more details.

5.2. Chameleon behavior recognition

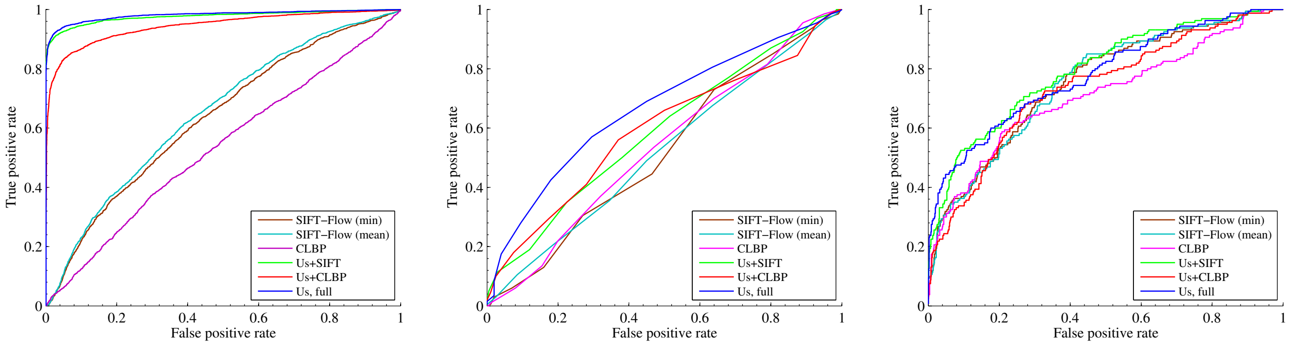
Males of the common chameleon species (chamaeleo chamaeleon) are known to employ different mating tactics, varying between large dominant males and small subordinate males, males mimicking female behavior, and more. Often, alternative tactics are characterized by distinctive color patterns [8, 37]. Here, we test our system's capability, by determining if image-pairs of these patterns represent the same behavior or not. Hence, unlike our other tests, small texture differences signify not different individuals, but rather different behavioral signals. In our tests we use only grayscale images, leaving treatment of color for future work. Color cues, which contribute much discriminative information, are therefor not used; recognition is determined solely by the texture patterns.

For our tests, we use a database of 109, adult chameleon body, grayscale photos (see Fig. 4 (mid)). These were classified by experts into 13 color patterns, based on similar color, stripes, and skin patches. Patterns represent different behavioral tactics, and include labels such as: small-size brown males and females during breeding season; medium-size brown males and females during breeding season; small-size brown males during courtship; medium-size brown males during courtship; and large-size green males during courtship. Our system was applied directly to these images, without prior-alignment or any other preprocessing.

In our tests, 2,000 image pairs are used, half are pairs from the same category and half with different texture patterns. Of these pairs, 1,600 are used for training, and the remaining 400 used for testing. Results are reported in Table 3 and Fig. 5 (mid). Evidently, this task is harder than fingerprint recognition, with performance scores dropping to near chance for some methods. This is not surprising, as the expert labeling of these categories often relied on color to class discrimination, but this information was not used our tests. Still, our full approach does far better than the others, succeeding in classifying a great deal of the images, despite the absence of color, with the CLBP variant of our method coming in a distant second.

5.3. Zebra finch male recognition

Zebra finchs mate for life, and effort has been made by Biologists to understand how partners recognize each other



FVC2004-DB1 result ROC curves

Chameleon result ROC curves

Zebra Finch result ROC curves

Figure 5: **ROC curves for the three benchmarks.** Please see text for more on the methods tested.

in, often very large, flocks. One possibility is that different individuals have unique songs or behaviors. In one recent study [11], female finches were trained to peck at a digital screen whenever a still-image of their mate was presented, thus demonstrating that females recognize their mates based on appearance only.

We test whether recognition is possible solely based on the zebra patterns on the males' chests (Fig. 4 (right)). These patterns are unique to each individual, and so can conceivably be used for recognition. This task is exceedingly difficult as the differences between these patterns are small, but they were often not clearly visible in our images.

Our data consists of a total of 138, high-resolution photos, of 16 finch males. We preprocessed these photos by aligning them using SIFT [27] keypoints extracted from the entire bird, and used to compute affine transformations from all bird photos to a single, manually selected reference image. A mask was then manually drawn around the texture of the reference bird, and applied to all aligned images, at the same coordinates, to produce the input for our method. In our test protocol, we created 1,920 pairs of such images, of which 640 were used for training and the remaining pairs for testing. Both test and train sets consisted of half same pairs and half not-same pairs.

Results are reported in Table 4. ROC curves are also provided, in Fig. 5 (right). Here too, the same pattern is evident, whereby our method, using LBP codes, outperforms both the larger SIFT descriptor, as well as the CLBP variant. The gap between these three methods and all others remains substantial. We note that the alignment step accounted to very few of the errors made by our system (less than 15% of the errors); errors were almost entirely due to the challenging viewing conditions in this set.

6. Conclusions

Methods specialized to specific imaging domains often utilize domain knowledge in order to extract explicit information. While this level of information is unavailable to more general methods, these enjoy a broader interest and often a more rapid evolution. Here, we show that a recent but widely used optical flow computation method can be adapted to texture identification domains by replacing the underlying descriptors and by a principled evaluation of the flow established between the two images, as well as the similarity of the warped images themselves.

We show that the proposed system outperforms recent published results in the domain of fingerprint recognition, yet is able to outperform alternative methods in two other biological identification tasks. As progress is made to the various components of this system (e.g., descriptor invariance properties [10, 39], matching engine quality [21] etc.) we expect the capabilities of this approach to likewise improve. Notwithstanding this potential, we would also like to incorporate our proposed generic system more closely with specific problem domains. Employing our proposed generic system in concert with domain information based techniques, for examples, performing minutia analysis for fingerprint recognition may provide addition gains.

Acknowledgments

We thank T. Keren-Rotem and E. Geffen for providing us with the behavior and photo data on the chameleons. We also thank G. Sion, S. Fleishman, J. Terkel, A. Barnea for providing the zebra finch photos.

References

[1] J. Abraham, P. Kwan, and J. Gao. Fingerprint matching using a hybrid shape and orientation descriptor. *State of the art in*

Biometrics, pages 25–56, 2011.

[2] J. S. Bartunek, M. Nilsson, B. Sallberg, and I. Claesson. Adaptive fingerprint image enhancement with emphasis on preprocessing of data. *TIP*, 22(2):644–656, 2013.

[3] O. Boiman and M. Irani. Similarity by composition. In *NIPS*, volume 19, 2007.

[4] K. Cao, X. Yang, X. Chen, X. Tao, Y. Zang, J. Liang, and J. Tian. Minutia handedness: A novel global feature for minutiae-based fingerprint matching. *PRL*, 33(10):1411–1421, 2012.

[5] R. Cappelli, M. Ferrara, and D. Maltoni. Fingerprint indexing based on minutia cylinder-code. *TPAMI*, 33(5):1051–1057, 2011.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), 2011. www.csie.ntu.edu.tw/∼cjlin/libsvm.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[8] M. Cuadrado, J. Martín, and P. López. Camouflage and escape decisions in the common chameleon chamaeleo chamaeleon. *Biological Journal of the Linnean Society*, 72(4):547–554, 2001.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1. IEEE, 2005.

[10] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *CVPR*, 2015.

[11] S. Fleischman, J. Terkel, and A. Barnea. Conspecific visual recognition in zebra finch females. In *International Behavior conference*, 2013.

[12] C. Gottschlich. Curved-region-based ridge frequency estimation and curved gabor filters for fingerprint image enhancement. *TIP*, 21(4):2220–2227, 2012.

[13] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *TIP*, 19(6):1657–1663, 2010.

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

[15] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. In *CVPR*, 2015.

[16] T. Hassner and C. Liu. *Dense Image Correspondences for Computer Vision*. Springer, 2015.

[17] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *TPAMI*, 20(8):777–789, 1998.

[18] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *ICCV*, pages 102–111, 1987.

[19] S. Jirachaweng. *Performance Improvement on Enhancement, Orientation Modeling, and Matching of Fingerprint*. PhD thesis, Kasetsart University, 2010.

[20] P. Johnson, F. Hua, and S. Schuckers. Texture modeling for synthetic fingerprint generation. In *CVPRw*, 2013.

[21] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, pages 2307–2314. IEEE, 2013.

[22] P. Kovesi. Fast almost-gaussian filtering. In *Digital Image Computing: Techniques and Applications (DICTA)*, pages 121–125. IEEE, 2010.

[23] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia.

[24] S. Li, H. Kim, C. Jin, S. Elliott, and M. Ma. Assessing the level of difficulty of fingerprint datasets based on relative quality measures. *Information Sciences*, 2013.

[25] C. Liu, G. Yang, and J. Gu. Learning discriminative illumination and filters for raw material classification with optimal projections of bidirectional texture functions. In *CVPR*, pages 1430–1437. IEEE, 2013.

[26] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *TPAMI*, 2011. Available: people.csail.mit.edu/celiu/SIFTflow/.

[27] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[28] D. Maio and A. K. Jain. *Handbook of fingerprint recognition*. Springer, 2009.

[29] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. FVC2004: Third fingerprint verification competition. In *Biometric Authentication*. Springer, 2004.

[30] NIST. Home page for NBIS. Available: www.nist.gov/itl/iad/ig/nbis.cfm, 2010.

[31] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7), 2002. Available: www.cse.oulu.fi/CMV/Downloads/LBPMatlab.

[32] D. Peralta, M. Galar, I. Triguero, O. Miguel-Hurtado, J. M. Benitez, and F. Herrera. Minutiae filtering to improve both efficacy and efficiency of fingerprint matching algorithms. *EAAI*, 32:37–53, 2014.

[33] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *NIPS*, pages 1241–1248, 2007.

[34] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012.

[35] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *IJCV*, 103(3):348–371, 2013.

[36] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, 2013.

[37] D. Stuart-Fox and A. Moussalli. Selection for social signalling drives the evolution of chameleon colour change. *PLoS biology*, 6(1):e25, 2008.

[38] P. Sutthiwichaiporn and V. Areekul. Adaptive boosted spectral filtering for progressive fingerprint enhancement. *PR*, 46(9):2465–2486, 2013.

[39] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *Trans. Pattern Anal. Mach. Intell.*, 2014. To appear.

[40] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proc. int. conf. on Multimedia*, pages 1469–1472, 2010.

[41] A. Vij and A. Namboodiri. Fingerprint indexing based on local arrangements of minutiae neighborhoods. In *CVPRw*, pages 71–76. IEEE, 2012.