Matching Local Self-Similarities across Images and Videos

Eli Shechtman Michal Irani
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
76100 Rehovot, Israel

Abstract

We present an approach for measuring similarity between visual entities (images or videos) based on matching internal self-similarities. What is correlated across images (or across video sequences) is the internal layout of local self-similarities (up to some distortions), even though the patterns generating those local self-similarities are quite different in each of the images/videos. These internal self-similarities are efficiently captured by a compact local "self-similarity descriptor", measured densely throughout the image/video, at multiple scales, while accounting for local and global geometric distortions. This gives rise to matching capabilities of complex visual data, including detection of objects in real cluttered images using only rough hand-sketches, handling textured objects with no clear boundaries, and detecting complex actions in cluttered video data with no prior learning. We compare our measure to commonly used image-based and video-based similarity measures, and demonstrate its applicability to object detection, retrieval, and action detection.

1. Introduction

Determining similarity between visual data is necessary in many computer vision tasks, including object detection and recognition, action recognition, texture classification, data retrieval, tracking, image alignment, etc. Methods for performing these tasks are usually based on representing an image using some global or local image properties, and comparing them using some similarity measure.

The relevant representations and the corresponding similarity measures can vary significantly. Images are often represented using dense photometric pixel-based properties or by compact region descriptors (features) often used with interest point detectors. Dense properties include raw pixel intensity or color values (of the entire image, of small patches [25, 3] or fragments [22]), texture filters [15] or other filter responses [18]. Common compact region descriptors include distribution based descriptors (e.g., SIFT [13]), differential descriptors (e.g., local derivatives [12]), shape-based descriptors using extracted edges (e.g. Shape Context [1]), and others. For a comprehensive comparison of many region descriptors for image matching see [16].

Figure 1. *These images of the same object (a heart) do NOT share common image properties (colors, textures, edges), but DO share a similar geometric layout of local internal self-similarities.*

Although these representations and their corresponding measures vary significantly, they all share the same basic assumption – that there exists a common underlying visual property (i.e., pixels colors, intensities, edges, gradients or other filter responses) which is shared by the two images (or sequences), and can therefore be extracted and compared across images/sequences. This assumption, however, may be too restrictive, as illustrated in Fig. 1. There is no obvious image property shared between those images. Nevertheless, we can clearly notice that these are instances of the same object (a heart). What makes these images similar is the fact that their local intensity patterns (in each image) are repeated in nearby image locations in a similar relative geometric layout. In other words, *the local internal layouts of self-similarities are shared by these images, even though the patterns generating those self-similarities are not shared by those images*. The notion of self similarity in video sequences is even stronger than in images. E.g., people wear the same clothes in consecutive frames and backgrounds tend to change gradually, resulting in strong self-similar patterns in local space-time video regions.

In this paper we present a "local self-similarity descriptor" which captures internal geometric layouts of local self-similarities within images/videos, while accounting for small local affine deformations. It captures self-similarity of color, edges, repetitive patterns (e.g., the right image in Fig. 1) and complex textures in a single unified way. A textured region in one image can be matched with a uniformly colored region in the other image as long as they have a similar spatial layout. These self-similarity descriptors are estimated on a dense grid of points in image/video data, at multiple scales. A good match between a pair of images (or a pair of video sequences), corresponds to finding a matching ensemble of such descriptors – with similar descriptor values at similar relative geometric positions, up to small non-rigid deformations. This allows to match a wide vari-

1-4244-1180-7/07/$25.00 ©2007 IEEE

ety of image and video types which are difficult to match otherwise: Complex objects in cluttered images are shown to be detected with only *rough hand sketches*; Differently textured instances of the same object are detected even if there are no clear boundaries; Complex actions performed by different people wearing different clothes with different backgrounds, are detected with no prior learning, based on a single example clip.

Self-similarity is closely related to the notion of statistical co-occurrence of pixel intensities across images, captured by Mutual Information (MI) [23]. Alternatively, internal joint pixel statistics are often computed and extracted from individual images and then compared across images (e.g., [8, 21, 11]). Most of these methods are restricted to measuring statistical co-occurrence of *pixel-wise measures* (intensities, color, or simple local texture properties), and are not easily extendable to co-occurrence of larger more meaningful patterns such as image patches (in some cases, such as MI, this limitation is due to the *curse of dimensionality*). Moreover, statistical co-occurrence is assumed to be global (within the entire image) – a very strong assumption which is often invalid. Some of these methods further require a prior learning phase with many examples [21, 11].

In our approach, *self-similarities are measured locally* (within a surrounding image region), and not globally. Our framework *explicitly models local and global geometric deformations* of self-similarities. Furthermore, we use *patches* (at different scales) as the basic unit for measuring internal self-similarities (these capture more meaningful image patterns than individual pixels). Local self-similarities of image patterns have also been employed for the purpose of texture edge detection [25], for detecting symmetries [14], and for other applications. The use of global self-similarity (between entire pre-aligned video frames) has also been proposed in video [2] for gait recognition.

Finally, we compare our measure to several commonly used image-based and video-based similarity measures, and demonstrate its applicability to object detection, retrieval, and action detection.

2. Overview of our approach

We would like to compare a "template" image $F(x, y)$ (or a video clip $F(x, y, t)$) to another image $G(x, y)$ (or video $G(x, y, t)$). F and G need not be of the same size. In fact, in most of our examples, F is a small template (of an object or action of interest), which is searched within a larger G (a larger image, a longer video sequence, or a collection of images/videos).

"Corresponding" points in F and G can look very different (e.g., see Fig. 3). While measuring similarity *across images* can be quite complex, the similarity *within each image* can be easily revealed with very simple similarity measures, such as a simple SSD (Sum of Square Differences),

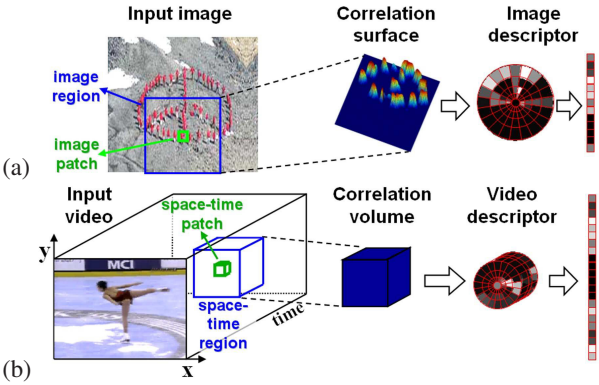


Figure 2. **Extracting the local "self-similarity" descriptor.** *(a) at an image pixel. (b) at a video pixel.*

resulting in local self-similarity descriptors which can now be matched across images (see Fig. 3).

We associate a "local self-similarity" descriptor d_q with every pixel q. This is done by correlating the image patch centered at q with a larger surrounding image region (e.g., of radius 40 pixels), resulting in a local internal "correlation surface" (Fig. 2.a). We use the term "local" to denote a small portion of the image (e.g., 5%) as opposed to the entire image. The correlation surface is then transformed into a *binned log-polar representation* [1] (Fig 2.a). This representation has two important benefits: (i) It results in a *compact* descriptor d_q for every pixel q. (ii) This descriptor accounts for increasing positional uncertainty with distance from the pixel q, thus accounting for *local spatial affine deformations* [1] (i.e., small variations in scale, orientation, and shear). On top of that, our descriptor accounts for additional *local non-rigid deformations* (see Sec. 3.1).

When matching video data – the patches, regions, correlation surfaces (volumes), and the self-similarity descriptors, are all *space-time entities* (Fig 2.b). The space-time video descriptor accounts for local affine deformations both in space and in time (thus accommodating also small differences in speed of action). Sec. 3 describes in detail how the local self-similarity descriptors are computed for images and for video data.

In order to match an entire image/video F to G, we compute the local self-similarity descriptors d_q densely throughout F and G . All the local descriptors in F form together a single global "ensemble of descriptors", which maintains their relative geometric positions. A good match of F in G corresponds to finding a similar ensemble of descriptors in G – *similar both in the descriptor values, and in their relative geometric positions* (up to small local shifts, to account for small deformations). This is described in Sec. 4.

We show results of applying this approach to detection of complex objects in cluttered images (Sec. 5), to image retrieval with simple hand sketches (Sec. 6), and to action detection in complex videos (Sec. 7).

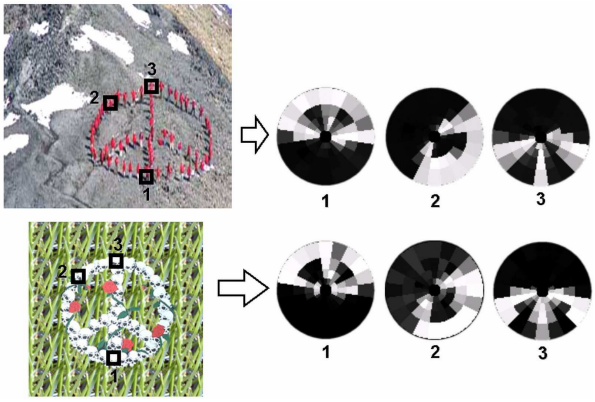


Figure 3. **Corresponding "Self-similarity descriptors".** *We show a few corresponding points (1,2,3) across two images of the same object, with their "self-similarity"descriptors. Despite the large difference in photometric properties between the two images, their corresponding "self-similarity" descriptors are quite similar.*

3. The local "self-similarity descriptor"

3.1. The Spatial image descriptor

Fig 2.a illustrates the procedure for generating the self-similarity descriptor d_q associated with an image pixel q. The surrounding image patch (typically 5×5) is compared with a larger surrounding image region centered at q (typically of radius 40), using simple *sum of square differences* (SSD) between patch colors (we used CIE L*a*b* space). The resulting *distance surface* $SSD_q(x, y)$ is normalized and transformed into a "correlation surface" $S_q(x, y)$:

$$S_q(x, y) = \exp\left(-\frac{SSD_q(x, y)}{\max\left(var_{noise}, var_{auto}(q)\right)}\right) \quad (1)$$

where var_{noise} is a constant that corresponds to acceptable photometric variations (in color, illumination or due to noise), and $var_{auto}(q)$ takes into account the patch contrast and its pattern structure, such that sharp edges are more tolerable to pattern variations than smooth patches. In our implementation $var_{auto}(q)$ is the maximal variance of the difference of all patches within a very small neighborhood of q (of radius 1) relative to the patch centered at q.

The correlation surface $S_q(x, y)$ is then transformed into log-polar coordinates centered at q , and partitioned into 80 bins (20 angles, 4 radial intervals). We select the maximal correlation value in each bin. The maximal values in those bins form the 80 entries of our local "self-similarity descriptor" vector d_q associated with the pixel q. Finally, this descriptor vector is normalized by linearly stretching its values to the range [0..1], in order to be invariant to the differences in pattern and color distribution of different patches and their surrounding image regions.

Fig. 3 displays the local self-similarity descriptor computed at several corresponding image locations in two differently looking images of the same object. Note that despite the large difference in photometric properties between the two images, their local self-similarity descriptors at corresponding image points (computed separately within each image) are quite similar.

Properties & benefits of the "self-similarity descriptor":

(i) Self-similarities are treated as a local image property, and are accordingly measured *locally* (within a surrounding image region), and not globally (within the entire image). This extends the applicability of the descriptor to a wide range of challenging images.

(ii) The log-polar representation accounts for *local affine deformations* in the self-similarities.

(iii) By choosing the maximal correlation value in each bin, the descriptor becomes insensitive to the exact position of the best matching patch within that bin (similar to the observation used for brain signal modelling, e.g. in [19]). Since the bins increase in size with the radius, this allows for addition radially increasing *non-rigid deformations*.

(iv) The use of *patches* (at different scales) as the basic unit for measuring internal self-similarities captures more meaningful image patterns than individual pixels. It treats colored regions, edges, lines and complex textures in a single unified way. A textured region in one image can be matched with a uniformly colored region or a differently textured region in the other image, as long as they have a similar spatial layout, i.e., those regions have similar *shapes*. Note that this is done without any explicit segmentation or edge detection, and can thus also handle regions (textured or uniform) with unclear boundaries.

3.2. The Space-time video descriptor

The notion of self similarity in video sequences is even stronger than it is in images. People tend to wear the same clothes in consecutive video frames and background scenes tend to change gradually, resulting in strong self-similar patterns in local space-time video regions.

The self-similarity descriptor presented in Sec. 3.1 is extended into space-time. Patches, regions, correlation surfaces (volumes), and the self-similarity descriptors, become space-time entities (see Fig 2.b). In our implementation we used $5 \times 5 \times 1$ patches, correlated against a surrounding $60 \times 60 \times 5$ space-time video region. The resulting "correlation volume" is transformed to a log-log-polar representation (logarithmic intervals both in space and in time, but polar only in space, resulting in a cylindrically shaped volume – see Fig 2.b). The resulting self-similarity descriptor vector is of size 182.

4. Matching global ensembles of local descriptors

In order to match an entire image/video F to G, we compute the local self-similarity descriptors d_q densely throughout F and G. These descriptors are computed 5 pixels apart

from each other (in every image or video frame). All the local descriptors in F form together a global "ensemble of descriptors". A good match of F in G corresponds to finding a similar ensemble of descriptors in G – *similar both in the descriptor values, and in their relative geometric positions* (up to small local shifts, to account for small global non-rigid deformations).

However, not all descriptors in the ensemble are informative. We first *filter out non-informative descriptors*, namely: (i) descriptors that do not capture any local self-similarity (i.e., whose center patch is *salient*, not similar to any of the other patches in its surrounding image/video region), and (ii) descriptors that contain high self-similarity everywhere in their surrounding region (corresponding to a *large homogeneous region*, i.e., a large uniformly colored or uniformly-textured region). The former type of non-informative descriptors (*saliency*) are detected as descriptors whose entries are all below some threshold (before normalizing the descriptor vector – see Sec 3.1). The latter type of non-informative descriptors (*homogeneity*) are detected by employing the sparseness measure of [9]. Discarding non-informative descriptors is important, as these may lead to ambiguous matches later in the matching phase. Note that the remaining descriptors still form a dense collection (much denser than sparse interest points [13, 16, 12]). Moreover, typical interest points locations would not necessarily correspond to locations of informative self-similarity descriptors, whereas a uniform patch or an edge-like patch may form an informative one.

To find a good match for the "ensemble of descriptors" of F within G, we use a modified version of the efficient "ensemble matching" algorithm of [3]. This algorithm employs a simple probabilistic "star graph" model to capture the relative geometric relations of a large number of local descriptors. In our applications, we connect all the descriptors in the template F into a *single* such ensemble of descriptors, and employ their search method for detecting a similar ensemble of descriptors within G (which allows for some local flexibility in descriptor positions and values). We used a sigmoid function on the L_1 distance to measure the similarity between descriptors. The ensemble search algorithm generates a dense likelihood map in the size of G, which corresponds to the likelihood of detecting F at each and every point in G (i.e., according to its degree of match). *Locations with high likelihood values are regarded as detected locations of F within G.*

Since self-similarity may appear at various scales and in different region sizes, we extract self-similarity descriptors at multiple scales. In the case of images we use a Gaussian image pyramid for generating those scales; in case of video data we use a space-time video pyramid. We use the same parameters (patch size, surrounding region, etc.) for all scales. Thus, the physical extent of a small 5×5 patch in a coarse scale, corresponds to the extent of a large image patch at a fine scale. An ensemble of descriptors is generated and searched for each scale independently, generating its own likelihood map. To combine information from multiple scales, we first normalize each log-likelihood map by the number of descriptors in its scale (these numbers may vary significantly from scale to scale). The normalized log-likelihood surfaces are then combined by a weighted average with weights corresponding to the degree of sparseness [9] of these log-likelihood surfaces.

5. Object detection in images

We applied the approach presented in the previous sections to detect objects of interest in cluttered images. Given *a single example image* of an object of interest (the "template image" – e.g., the flower in Fig. 4.a), we densely computed its local image descriptors of Sec. 3.1 to generate an "ensemble of descriptors". We search for this template-ensemble in several cluttered images (e.g., Fig. 4.b) using the algorithm of Sec. 4. Image locations with high likelihood values were regarded as detections of the template image and are overlayed on top of the dark gray image. We used the *same* threshold for all examples in each of the figures (but varied it for different templates). *No prior image segmentation* was involved, *nor any prior learning*.

We have applied our algorithm to real image templates as well as to *rough hand-sketched templates* – see Figs. 4, 5, 6, 7. Note that in the case of sketched templates, although the sketch is uniform in color, such a global constraint is not imposed on the searched objects. This is because the self-similarity descriptor tends to be more local, imposing self-similarity only within smaller object re-

Figure 4. **Object detection.** *(a) A single template image (a flower). (b) The images against which it was compared with the corresponding detections. The continuous likelihood values above a threshold (same threshold for all images) are shown superimposed on the gray-scale images, displaying detections of the template at correct locations (red corresponds to the highest values).*

Figure 5. **Detection using a sketch.** *(a) A hand-sketched template. (b) Detected locations in other images.*

gions. (For example, the objects in our examples were typically $150-200$ pixels in each dimension, whereas the self-similarity descriptor is constrained to a radius of 40 pixels around each pixel). Our method is therefore capable of detecting similarly shaped objects with global photometric variability (e.g., people with pants and shirts of different colors/textures, etc.)

Comparison to other descriptors and measures:

We further compared the matching performance of our descriptors with some state-of-the-art local descriptors evaluated in [16]. We selected a subset of local descriptors which ranked highest (and used the implementation in [16]). These included: *gradient location-orientation histogram (GLOH)* [16] – a log-polar extension of SIFT [13] that was shown to be more robust and distinctive, local *Shape Context* [1] (an extended version with orientations [16]), and four other descriptors. For a sound comparison of these descriptors with ours, they were extracted densely on edges (at multiple scales) to avoid homogeneous regions and lack of interest points, and combined using the same "ensemble matching" method described in Sec. 4. In addition, we compared our method against Mutual Information, applied globally to the template (we tried both on color and on gray-scale representations). We compared our method to the above methods on many challenging pairs of images (more than 60 pairs with templates such as flowers, hearts, peace symbols, etc.; each template was compared against multiple images). Correct detection of the template was declared if there was a unique high peak detected within the other image, at the correct object. All of the above-mentioned methods failed in the majority of the cases to find the template in the correct position in the other image, whereas our method found it correctly in 86% of them. A few (representative) examples are displayed in Fig. 7.

Figure 6. **Detection using a sketch.** *(a) A hand-sketched template. (b) The images against which it was compared with the corresponding detections.*

6. Image retrieval by "sketching"

We further examined the feasibility of our method for image retrieval from a database of images using rough *hand-sketched queries*. We generated 8 rough hand-sketches corresponding to various *complex human poses*. We also generated a database of 72 images (downloaded from the internet), with the demand that each of the 8 poses appears within at least 9 of the database images . Fig. 8 shows the hand-sketched queries (templates) and the 9 top-ranked images retrieved for each query. The score for each database image given a query image was computed as the highest likelihood value obtained for that query image within that database image. On top of that we have verified that the peak values were obtained at the correct locations. False detections are marked by a red frame. Note the high detection rate: in all cases the 5 top-ranked images were correct, and for most pose-queries it found 8 out of the 9 database images within its top 9 matches. Note the cluttered backgrounds and the high geometric and photometric variability between different instances of each pose. Moreover, note the high variability of the different instances of the same pose (different images within a column), vs. the small differences across different poses (e.g., pose of column 3 vs. 4, and 5 vs. 6; serving a role similar to that of distractors).

Previous methods for image retrieval using image sketches (e.g., [10, 7]) assume that the sketched query image and the database image share similar low-resolution photometric properties (colors, textures, low-level wavelet coefficients, etc.). This assumption is not valid for the database of Fig. 8. Other methods assume similarity of the contour of objects (e.g., [6]) or proximity of corresponding extracted edges (e.g., [24]). While many objects are well

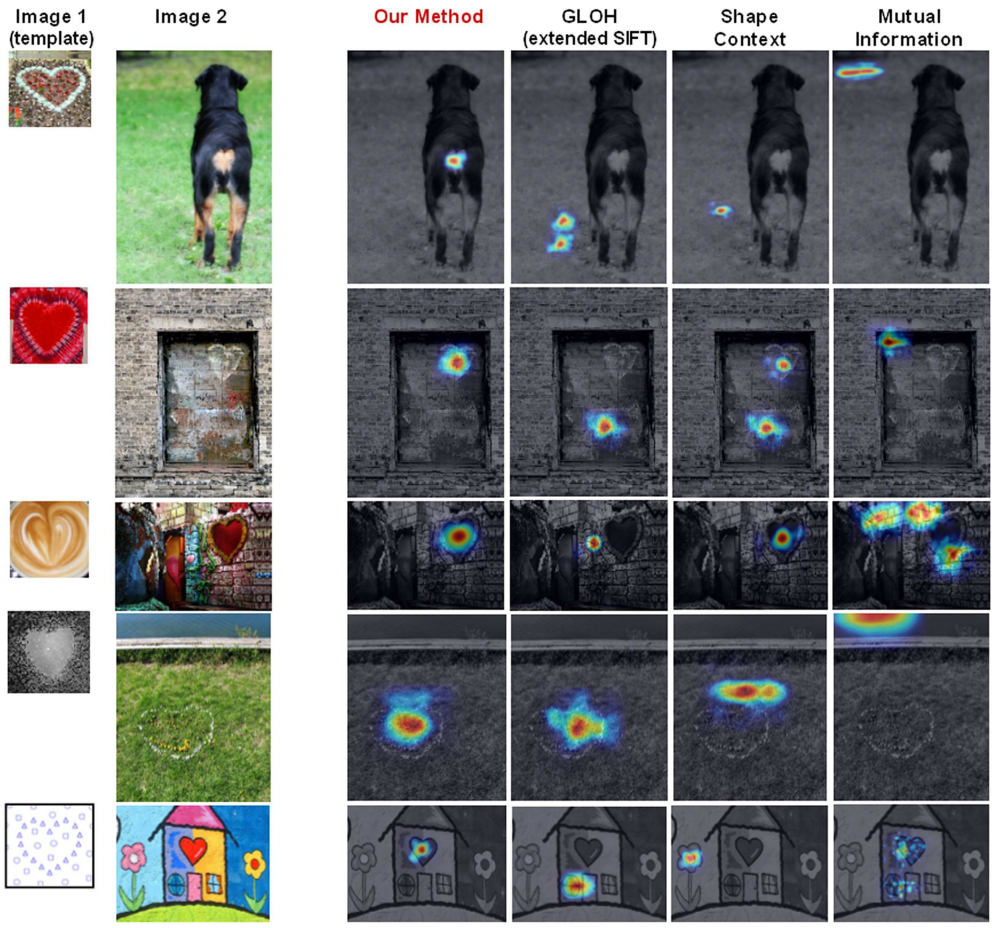


Figure 7. **Comparison to other descriptors and match measures.** *We compared our method with several other state-of-the-art local descriptors and matching methods on more than 60 challenging image pairs. All these methods failed to find the template in Image 2 in the majority of the pairs, whereas our method found the objects in the correct location in* 86% *of them. Each method was used to generate a likelihood surface, and peaks above* 90% *of its highest value are displayed. Displayed are a few such examples (these are representative results, each template was compared against multiple images not shown in the figure) – see text for more details. The object in each pair of images was of similar size (up to +/-* 20% *in scale), but is often displayed larger, for visibility purposes.*

characterized by their edges, there are many other objects that do not have such clear edges, e.g. see Fig. 1. Our descriptor captures both edge properties as well as region properties and therefore can address such cases.

7. Action detection in video

Applying the ensemble matching algorithm of Sec. 4 with the space-time self-similarity descriptors (Sec. 3.2) gives rise to simultaneous detection of multiple complex actions in video sequences of different people wearing different clothes with different backgrounds, without requiring any prior learning (i.e., based on a single example clip). Our approach can be applied to complex video sequences without requiring any foreground background segmentation [26], nor any motion estimation [5] or tracking. Unlike [4, 17, 12], our method requires no prior learning (nor multiple examples of each action), nor assumes existence of common space-time features across sequences.

Our approach is probably most closely related to the capabilities presented in our previous work [20]. However, unlike [20], our new method can handle highly aliased video sequences with *non-instantaneous motions*. Furthermore, our method can match both stationary and moving objects (as opposed to only moving objects in [20]).

Fig. 9 shows results of applying our action detection algorithm for detecting a ballet turn in a video clip of [20]. We also compare our results to their results on this video data. They had 2 miss detections and 4 false alarms (see figure and video clip). Our new algorithm has detected all instances of the action correctly, with no false alarms or missed detections. The superior results are because we can now handle strong temporal aliasing, and explicitly account

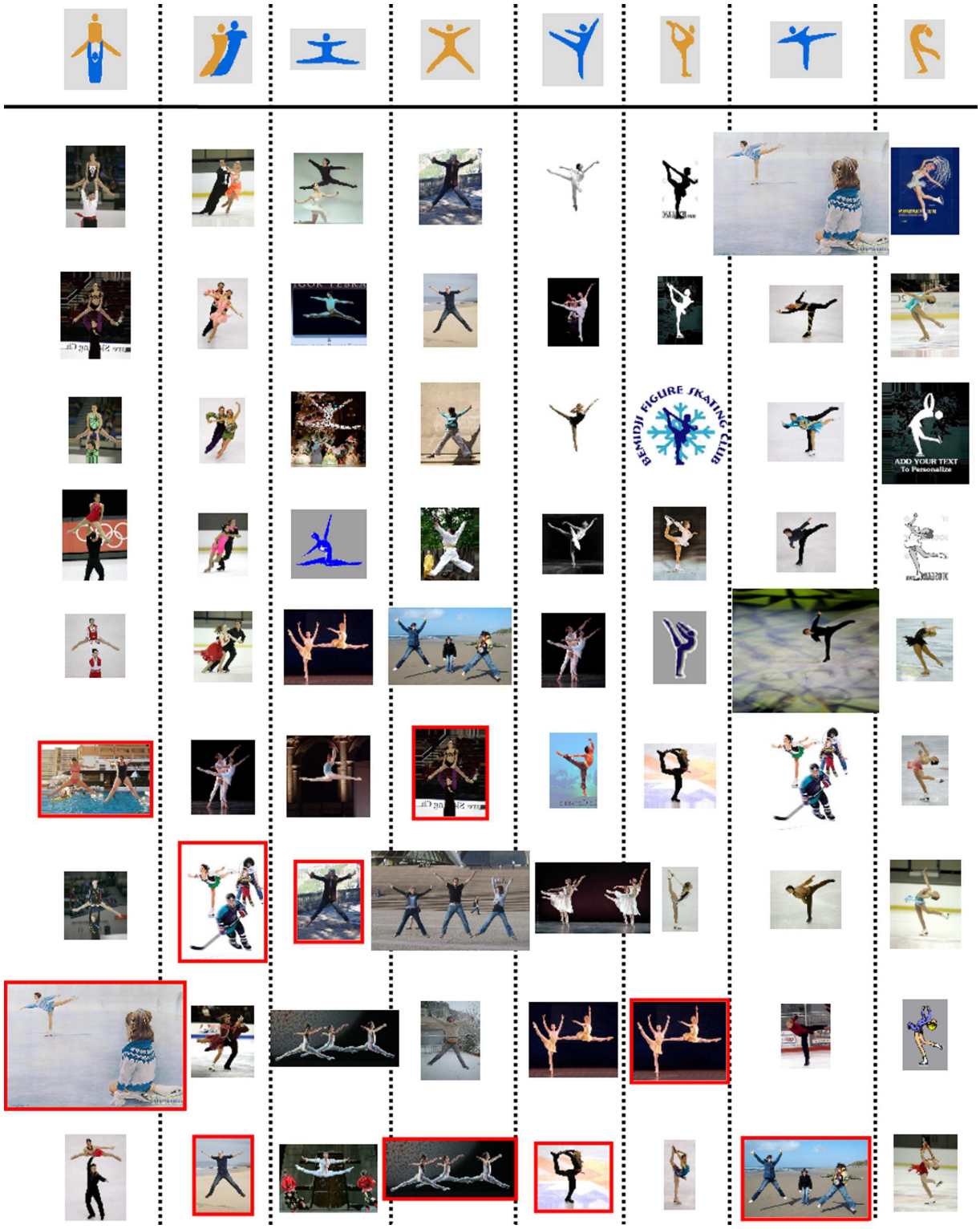


Figure 8. **Image retrieval by sketching.** *8 hand-sketched template queries (top row) were used to retrieve images from a database of* 72 *images. Each of these poses appears within* 9 *database images (see text). The* 9 *top-ranked images retrieved by each template query are presented in the column below the template (in decreasing match score). False detections are marked by a red frame.*

(a) The action clip: 

(b) 

(c) 

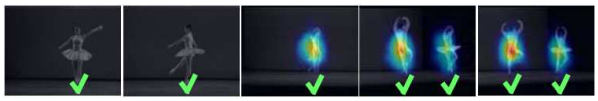
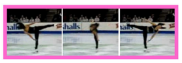
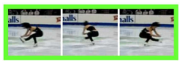
(d) 

Figure 9. **Action detection and comparison to [20].** *(a) A few sample frames from a template video clip (ballet turn). (b) A few sample frames from a long ballet video sequence (25 seconds) with two dancers, against which the template was compared. (c) In the results obtained by [20] there are 2 missed detections and 4 false alarms. A few frames from those erroneous video segments are shown here (marked by "X"). (d) OUR NEW RESULTS: All instances of the action were detected correctly with no false alarms or missed detections.* **These results are easier to grasp in video. Please see: *www.wisdom.weizmann.ac.il/∼vision/SelfSimilarities.html*.**

(a) Turn 1:  Turn 2: 

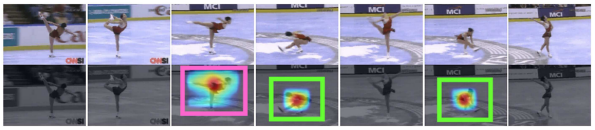
(b) 

Figure 10. **Action detection.** *(a) A few sample frames from two different action templates of different ice-skating turns – turn1 (in pink) and turn2 (in green). (b) A few sample frames from a long ballet video sequence (30 seconds) of a different ice-skater, and the corresponding detection results below. All instances of the two actions were detected correctly with no false alarms or missed detections, despite the strong temporal aliasing.* **Please see: *www.wisdom.weizmann.ac.il/∼vision/SelfSimilarities.html*.**

for some non-rigid deformations (both spatial and temporal) between the template action clip and the matched video.

Fig. 10 shows detection of two different types of ice-skating movements. The template clips (one example clip for each movement type) were obtained from one ice-skater, and were used to detect these actions in a long ice-skating sequence of a different ice-skater wearing different clothes, at a different place. These sequences contain very strong temporal aliasing and therefore cannot be handled well by methods like [20]. Unlike [17] (who also showed results on ice-skating video sequences), our approach requires no prior learning from multiple examples of each action, and does not rely on having common extractable features across action sequences. Moreover our sequences are characterized by a much higher motion aliasing.

Acknowledgments: The authors would like to thanks Y. Caspi and O. Boiman for their helpful comments. This work was supported in part by the Alberto Moscona Foundation. The research was conducted at the Moross Laboratory for Vision and Motor Control at the Weizmann Institute of science.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 2002.

[2] C. BenAbdelkader, R. G. Cutler, and L. S. Davis. Gait recognition using image self-similarity. *EURASIP Journal on Applied Signal Processing*, 2004(4), 2004.

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *ICCV*, Beijing, October 2005.

[4] P. Dolla'r, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *ICCV workshop VS-PETS*, 2005.

[5] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, October 2003.

[6] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. In *ECCV*, May 2006.

[7] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance. *PAMI*, 17(7), 1995.

[8] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE T-SMC*, 1973.

[9] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *JMLR*, 5:1457–1469, 2004.

[10] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. In *SIGGRAPH*, 1995.

[11] N. Jojic and Y. Caspi. Capturing image structure with probabilistic index maps. In *CVPR*, 2004.

[12] I. Laptev and T. Lindeberg. Space-time interest points. *ICCV*, 2003.

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

[14] G. Loy and J.-O. Eklundh. Detecting symmetry and symmetric constellations of features. In *ECCV*, 2006.

[15] J. Malik, S. Belongie, J. Shi, and T. K. Leung. Textons, contours and regions: Cue integration in image segmentation. In *ICCV*, 1999.

[16] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.

[17] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *British Machine Vision Conference*, 2006.

[18] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. In *IJCV*, 2000.

[19] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, , and T. Poggio. Robust object recognition with cortex-like mechanisms. *to appear in PAMI*, 2006.

[20] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR*, 2005.

[21] C. Stauffer and W. E. L. Grimson. Similarity templates for detection and recognition. In *CVPR*, 2001.

[22] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. *Proc. 4th International Workshop on Visual Form*, 2001.

[23] P. Viola and W. W. III. Alignment by maximization of mutual information. In *ICCV*, pages 16–23, 1995.

[24] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006.

[25] L. Wolf, X. Huang, I. Martin, and D. Metaxas. Patch-based texture edges and segmentation. In *ECCV*, 2006.

[26] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.