International Journal of Computer Vision 74(1), 17–31, 2007

DOI: 10.1007/s11263-006-0009-9

Detecting Irregularities in Images and in Video*

OREN BOIMAN AND MICHAL IRANI

Department of Computer Science and Applied Math, The Weizmann Institute of Science, 76100 Rehovot, Israel

Received March 7, 2006; Accepted November 14, 2006

First online version published in January, 2007

Abstract. We address the problem of detecting irregularities in visual data, e.g., detecting suspicious behaviors in video sequences, or identifying salient patterns in images. The term "irregular" depends on the context in which the "regular" or "valid" are defined. Yet, it is not realistic to expect explicit definition of all possible valid configurations for a given context. We pose the problem of determining the validity of visual data as a process of constructing a puzzle: We try to compose a new observed image region or a new video segment ("the query") using chunks of data ("pieces of puzzle") extracted from previous visual examples ("the database"). Regions in the observed data which can be composed using large contiguous chunks of data from the database are considered very likely, whereas regions in the observed data which cannot be composed from the database (or can be composed, but only using small fragmented pieces) are regarded as unlikely/suspicious. The problem is posed as an inference process in a probabilistic graphical model. We show applications of this approach to identifying saliency in images and video, for detecting suspicious behaviors and for automatic visual inspection for quality assurance.

Keywords: detecting suspicious behaviors, saliency, detecting irregularities, novelty detection, anomaly detection, action recognition, automatic visual inspection

1. Introduction

Detection of irregular visual patterns in images and in video sequences is useful for a variety of tasks. Detecting *suspicious behaviors* or *unusual objects* is important for surveillance and monitoring. Identifying *spatial saliency* in images is useful for quality control and automatic inspection. *Behavioral saliency* in video is useful for drawing the viewer's *attention*.

Previous approaches to recognition of suspicious behaviors or activities can broadly be classified into two classes of approaches: *rule-based methods* (e.g., Ivanov and Bobick (1999)) and *statistical methods* without predefined rules (e.g., Stauffer and Grimson (2000) and Zhong et al. (2004)). The statistical methods are more appealing, since they do not assume a predefined set of rules for all valid configurations. Instead, they try to automatically learn the notion of regularity from the data, and thus infer about the suspicious. Nevertheless, the representations employed in previous methods have been either very restrictive (e.g., trajectories of moving objects (Stauffer and Grimson, 2000)), or else too global (e.g., a single small descriptor vector for an entire frame (Zhong et al., 2004)).

In this paper we formulate the problem of detecting regularities and irregularities as the problem of composing (explaining) the new observed visual data (an image or a video sequence, referred to below as "query") using spatio-temporal patches extracted from previous visual examples (the "database"). Regions in the query which can be composed using large contiguous chunks of data from the example database are considered likely. The larger those regions are, the greater the likelihood is. Regions in the query which cannot be composed from the example database (or can be composed, but only using small fragmented pieces) are regarded as unlikely/suspicious. Our approach can thus infer and generalize from just a few examples, about the validity of a much larger context of image patterns and behaviors, even if those particular configurations have never been seen before. Local descriptors are extracted from small image or video patches (composed together to large ensembles of patches), thus

*Patent Pending

allowing to quickly and efficiently infer about subtle but important local changes in behavior (e.g., a man walking vs. a man walking while pointing a gun). Moreover, our approach is capable of simultaneously identifying a valid behavior in one portion of the field of view, and a suspicious behavior in a different portion the field of view, thus highlighting only the detected suspicious regions within the frame, and not the entire frame. Such examples are shown in Section 6.

Inference from image patches or fragments has been previously employed in the task of class-based object recognition (e.g. Bart and Ullman (2004), Felzenszwalb and Huttenlocher (2005) and Fergus et al.). A small number of informative fragments have been learned and preselected for a small number of pre-defined classes of objects. However, class-based representations cannot capture the overwhelming number of possibilities of composing unknown objects or behaviors in a scene, and are therefore not suitable for our underlying task of detecting irregularities.

Our approach can also be applied for detecting *saliency* in images and in video sequences. For example, given a single image *with no prior information*, we can measure the "validity" of each image region (the "query") relative to the remaining portions of the same image (the "database" used for this particular query). An image region will be detected as salient if it cannot be explained by anything similar in other portions of the image. Similarly, given a single video sequence (with no prior knowledge of what is a normal behavior), we can detect "salient behaviors" as behaviors which cannot be supported by any other dynamic phenomena occurring at the same time in the video.

Previous approaches for detecting image saliency (e.g., Itti et al. (1998)) proposed measuring the degree of dissimilarity between an image location and its immediate surrounding region. Thus, for example, image regions which exhibit large changes in contrast are detected as salient image regions. Their definition of "visual attention" is derived from the same reasoning. Nevertheless, we believe that the notion of saliency is not necessarily determined by the immediate surrounding image regions. For example, a single yellow spot on a black paper may be salient. However, if there are many yellow spots spread all over the black paper, then a single spot will no longer draw our attention, even though it still induces a large change in contrast relative to its surrounding vicinity. Our approach therefore suggests a new and more intuitive interpretation of the term "saliency", which stems from the inner statistics of the entire image. Our approach to spatial image saliency is more closely related to that of Honda and Nayar (2001). However, (Honda and Nayar, 2001) is restricted to repetitive structured image patterns and is highly dependent on the local surrounding image properties, whereas our approach is not. Examples of detected spatial saliency in images and behavioral saliency in video sequences using our approach are shown in Section 6.

Our paper therefore offers four main contributions:

1. We propose an approach for inferring and generalizing from just a few examples, about the validity of a much larger context of image patterns and behaviors, even if those particular configurations have never been seen before.
2. We present a new graph-based Bayesian inference algorithm which allows to efficiently detect *large ensembles of patches* (e.g., hundreds of patches), at multiple spatio-temporal scales. It simultaneously imposes constraints on the relative geometric arrangement of these patches in the ensemble as well as on their descriptors.
3. We propose a new interpretation to the term "saliency" and "visual attention" in images and in video sequences.
4. We present a single unified framework for treating several different problems in Computer Vision, which have been treated separately in the past. These include: attention in images, attention in video, recognition of suspicious behaviors, recognition of unusual objects, automatic visual inspection (e.g., for quality assurance), and more.

A shorter version of this paper appeared in ICCV 2005 (Boiman and Irani, 2005).

2. Inference by Composition

Given only a few examples, we (humans) have a notion of what is regular/valid, and what is irregular/suspicious, even when we see new configurations that we never saw before. We do not require explicit definition of all possible valid configurations for a given context. The notion of "regularity"/"validity" is learned and generalized from just a few examples of valid patterns (of behavior in video, or of appearance in images), and all other configurations are automatically inferred from those.

Figure 1illustrates the basic concept underlying this idea in the paper. Given a new image (a query—Fig. 1(a)), we check whether each image region can be explained by a large enough contiguous region of support in the database (see Figs. 1(b) and (c)). Although we have never seen a man sitting with both arms raised, we can infer the validity of this pose from the three database images of Fig. 1(c).

Thus, regions in the new observed data/query (an image or a video sequence) which can be explained by large contiguous chunks of data from the database are considered very likely, whereas regions in the query which

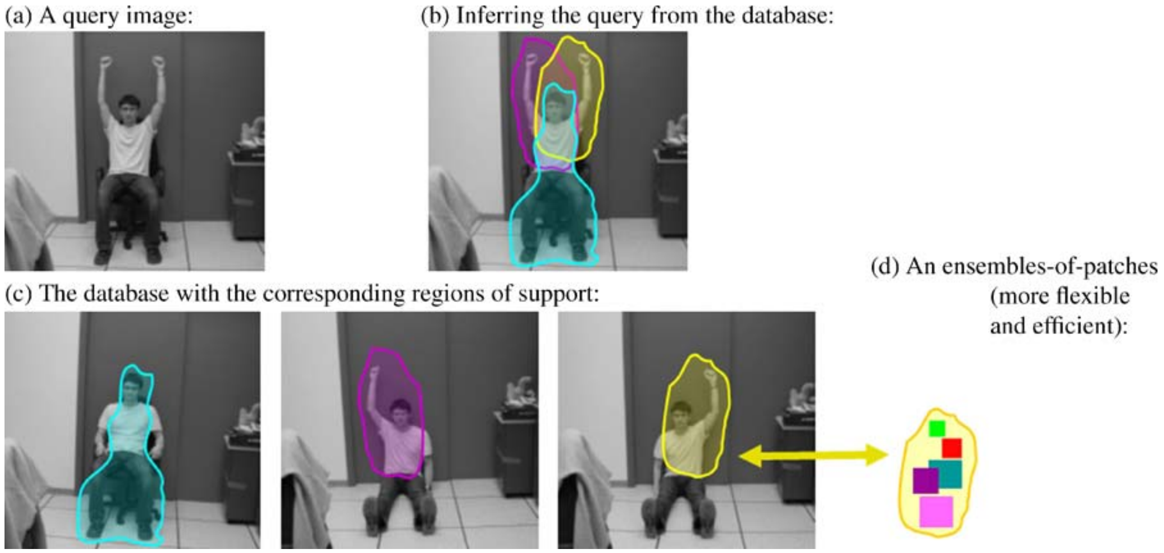

(a) A query image:
(b) Inferring the query from the database:
(c) The database with the corresponding regions of support:
(d) An ensembles-of-patches (more flexible and efficient):

Figure 1. *The basic concept—Inference by Composition*. A region in the query image is considered likely if it has a large enough contiguous region of support in the database. New valid image configurations can thus be inferred from the database, even though they have never been seen before.

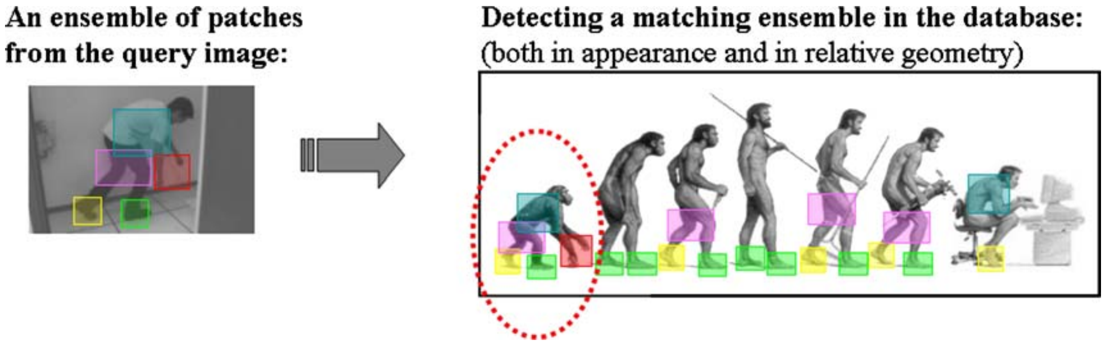

An ensemble of patches from the query image:
Detecting a matching ensemble in the database: (both in appearance and in relative geometry)

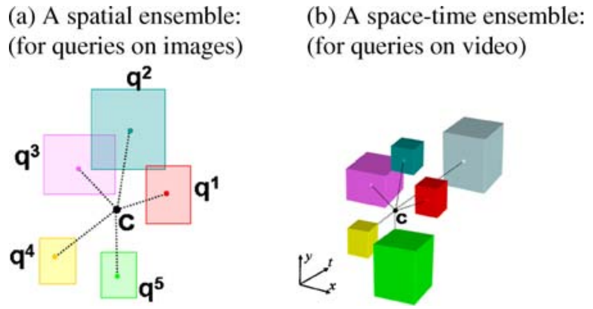
Figure 2. Detecting a matching ensemble of patches.

cannot be explained by large enough database pieces are considered unlikely or suspicious. When the visual query is an image, then those chunks of data have only a spatial extent. When the visual query is a video sequence, then those chunks of data have both a spatial and a temporal extent.

3. Ensembles of Patches

Human behaviors and natural spatial structures never repeat identically. For example, no two people walk in the same manner. One may raise his arms higher than the other, or may just walk faster. We therefore want to allow for small non-rigid deformations (in space and in time) in our "pieces of puzzle" (chunks of data). This is particularly true for large chunks of data. To account for such local non-rigid deformations, large chunks are broken down to an *ensemble of lots of small patches* at multiple scales with their relative geometric positions. This is illustrated in Fig. 1(d). In the inference process, we search for a similar geometric configuration of patches with similar properties (of behavior, or of appearance), while allowing for small local misalignments in the relative geometric arrangement. This concept is illustrated in Fig. 2. When the visual query is an image, then an ensemble of patches is composed of spatial patches (see Fig. 3(a)). When the visual query is a video sequence, then the ensemble of patches is composed of spatio-temporal patches (see Fig. 3(b)), which allows to capture information about dynamic behaviors. In our current implementation, a single ensemble typically contains *hundreds* of patches, simultaneously from multiple scales (multiple spatial scales in the case of image patches, and multiple space-time scales in the case of spatio-temporal patches).

While the idea of composing new data from example patches was previously proven useful for a variety of tasks (e.g., Efros and Leung (1999), Freeman et al. (2000) and Wexler et al. (2004)), these methods did not impose any geometric restriction on the example patches


(a) A spatial ensemble: (for queries on images)

(b) A space-time ensemble: (for queries on video)

Figure 3. Ensembles of patches in images and video.

used for construction, i.e., their relative positions and distances in the database. This was not necessary for their purpose. It is however crucial here, for the purpose of detecting irregularities. Often, the only real cue of information for distinguishing between a likely and an unlikely phenomenon is the degree of fragmentation of its support in the database. For example, the stretched arm of a man holding a gun is similar to an instantaneous stretching of the arm while walking, but its region of support is very limited in time.

Capturing the geometric relations of patches was identified as being important for the task of class-based object recognition (Bart and Ullman, 2004; Felzenszwalb and Huttenlocher, 2005; Fergus et al.,; Leibe et al.) Those approaches are not suitable for our objective for two reasons: (i) Their geometric configurations are restricted to a relatively small number of patches, thus cannot capture subtle differences which are crucial for detection of irregularities. (ii) Those configurations were pre-learned for a small number of pre-defined classes of objects, whereas our framework is applicable to any type of visual data. While the geometric constraints of Leibe et al. are more flexible, thus allowing to recognize new object configurations from just a few examples, their method is still limited to a set of predefined object classes with *predefined object centers*. This is not suitable for detecting irregularities, where there is no notion of object classes.

"Video Google" (Sivic and Zisserman, 2003) imposes geometric constraints on large collections of non class-based descriptors, and searches for them very efficiently. However, those descriptors are spatial in nature and the search is restricted to individual image frames, thus not allowing to capture behaviors.

In order for the inference to be performed in reasonable times, information about the small patches and their relative arrangement must be efficiently stored in and extracted from the database. For each small patch extracted from the examples, a descriptor vector is computed and stored (see below), along with the *absolute* coordinates of the patch (spatial or spatio-temporal coordinates). Thus, the relative arrangement of all patches in the image/video database is implicitly available. Later, our inference algorithm takes an ensemble of patches from the visual query and searches the database for a similar configuration of patches (both in the descriptors and in their relative geometric arrangement). To allow for fast search and retrieval, those patches are stored in a multi-scale data structure. Using a probabilistic graphical model (Section 4), we present an efficient inference algorithm (Section 4.2) for the ensemble search problem.

3.1. Patch Descriptors

Patch descriptors are generated for each query patch and for each database patch. The descriptors capture local information about appearance/behavior. Our current implementation uses very simple descriptors, which could easily be replaced by more sophisticated descriptors:

The Spatial Image Descriptor of a small (e.g., 7 × 7) spatial patch is constructed as follows: The spatial gradient magnitude is computed for each pixel in the patch. These values are then stacked in a vector, which is normalized to a unit length. Such descriptors are densely extracted for each point in the image. This descriptor extraction process is repeated in several spatial scales of the spatial Gaussian pyramid of the image. Thus, a 7 × 7 patch extracted from a coarse scale has a larger spatial support in the input image (i.e., in the fine scale). In some applications an RGB/intensity-based descriptor may be more appropriate than a gradient-based one. In general our overall framework is not restricted to those particular descriptors. Those could be easily replaced by more sophisticated spatial descriptors such as SIFT (Lowe, 2004) etc.

The Spatio-Temporal Video Descriptor of a small (e.g., $7 \times 7 \times 4$) spatio-temporal video patch is constructed from the absolute values of the temporal derivatives in all pixels of the patch. These values are stacked in a vector and normalized to a unit length. This descriptor extraction process is repeated in several spatial and temporal scales of a space-time video pyramid. Thus, a 7 × 7 × 4 patch extracted from a coarse scale has a larger spatial and larger temporal support in the input sequence.

Note that this descriptor is nearly *invariant to a static background*, since the temporal derivative is always zero in any static background. Therefore, using this spatio-temporal descriptor, we can detect irregular actions in a new query sequence, regardless of the background. However, this simple descriptor is dependent on spatial texture, which may pose a problem with people wearing highly textured clothes. Our approach, however, is not restricted to the particular choice of these simple descriptors. Those descriptors could be easily replaced by more sophisticated space-time descriptors (which are action-sensitive and more invariant to appearance), such as Shechtman and Irani (2005) or Laptev and Lindeberg (2003).

4. The Basic Algorithm

Given a new visual query (an image or a video sequence), we would like to estimate the likelihood of each and every point in it. This is done by checking the validity of a large region (e.g., 50 × 50 region in an image, and 50 × 50 × 50 region in a video sequence) surrounding *every* pixel. The large surrounding region is broken into lots (hundreds) of small patches at multiple scales (spatial or spatio-temporal), and is represented by a single ensemble of patches corresponding to that particular image/video point. Let $q^1, q^2, \ldots, q^n$ denote the patches in the ensemble (see Fig. 3(a)). Each patch q^i is associated with two types of attributes: (i) its descriptor vector d^i, and (ii) its location in absolute coordinates l^i. We choose an *arbitrary* reference point c (e.g., the center of the ensemble—see Fig. 3(a)), which serves as the "origin" of the local coordinate system (thus defining the relative positions of the patches within the ensemble).

4.1. Statistical Formulation

Let an observed ensemble of patches within the query be denoted by y. We would like to compute the joint likelihood $P(x, y)$ that the observed ensemble y in the query is similar to some hidden ensemble x in the database (similar both in its descriptor values of the patches, as well as in their relative positions). We can factor the joint likelihood as: $P(x, y) = P(y \mid x)P(x)$. Our modelling of $P(y \mid x)$ resembles the probabilistic modelling of the "star graph" of Felzenszwalb and Huttenlocher (2005). However, in the class-based setting of Felzenszwalb and Huttenlocher (2005) what is computed is $P(y;\theta)$, where θ is a prelearned set of parameters of a given patch-constellation of an object-class. In our case, however, there is no notion of objects, i.e., there is no prior parametric modelling of the database ensemble x. Thus, θ is undefined, and $P(x)$ must be estimated non-parametrically directly from the database of examples.

Let d_y^i denote the descriptor vector of the i-th observed patch in y, and l_y^i denote its location (in absolute coordinates). Similarly, d_x^i denotes the descriptor vector of the i-th hidden (database) patch in x, and l_x^i denotes its location. Let c_y and c_x denote the "origin" points of the observed and hidden ensembles. The similarity between any such pair of ensembles y and x is captured by the following likelihood:

$$P(x, y) = P\left(c_x, d_x^1, \ldots, l_x^1, \ldots, \quad c_y, d_y^1, \ldots, l_y^1, \ldots\right) \tag{1}$$

In order to make the computation of the likelihood in Eq. (1) tractable, we make some simplifying statistical assumptions. Given a hidden database patch and its descriptor d_x^i, the corresponding observed descriptor d_y^i is assumed to be independent of the other patch descriptors. (This is a standard Markovian assumption, e.g., Freeman et al. (2000), which is obviously not valid in case of overlapping patches, but is a useful approximation). We model the similarity between descriptors using a Gaussian distribution:

$$P\left(d_y^i \mid d_x^i\right) = \alpha_1 \exp\left(-\frac{1}{2}\left(d_y^i - d_x^i\right)^T S_D^{-1}\left(d_y^i - d_x^i\right)\right) \tag{2}$$

where α_1 is a constant, and S_D is a constant covariance matrix, which determines the allowable deviation in the descriptor values. Other distributions can be plugged in the model, corresponding to other descriptor similarity functions. Given the *relative* location of the hidden database patch $(l_x^i - c_x)$, the relative location of the corresponding observed patch $(l_y^i - c_y)$ is assumed to be independent of all other patch locations. This assumption enables to compare the geometric arrangement of two ensembles of patches with enough flexibility to accommodate for small changes in viewing angle, scale, pose and behavior. Thus:

$$\begin{aligned} P\left(l_y^i \mid l_x^i, c_x, c_y\right) = \alpha_2 \cdot \exp\Big(&-\frac{1}{2}\left(\left(l_y^i - c_y\right) - \left(l_x^i - c_x\right)\right)^T \\ &\times S_L^{-1}\left(\left(l_y^i - c_y\right) - \left(l_x^i - c_x\right)\right)\Big) \end{aligned} \tag{3}$$

where α_2 is a constant, and S_L is a constant covariance matrix, which captures the allowed deviations in the relative patch locations. (In this case the dependency in relative locations was modelled using a Gaussian, however the model is not restricted to that).

So far we modelled the relations between attributes *across ensembles* (descriptors: d_y^i, d_x^i, and relative locations: $l_y^i - c_y, l_x^i - c_x$). We still need to model the relations *within the hidden ensemble*, namely, the relations between a patch descriptor d_x^i to its location l_x^i. In the general case, this relation is highly non-analytic, and hence cannot be modelled parametrically (in contrast to class-based approaches, e.g. Felzenszwalb and Huttenlocher (2005) and Fergus et al. (2003)). Therefore, we model it *non-parametrically* using examples from the database:

$$P\left(d_x \mid l_x\right) = \begin{cases} 1 & (d_x, l_x) \in \textit{Database} \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

where d_x and l_x are an arbitrary descriptor and location.

We assume a uniform prior distribution for c_x and c_y (local origin points), i.e., no prior preference for the location of the ensemble in the database or in the query. The relation between all the above-mentioned variables is depicted in the Bayesian network in Fig. 4.

Thus, for an observed ensemble y and a hidden database ensemble x, we can factor the joint likelihood

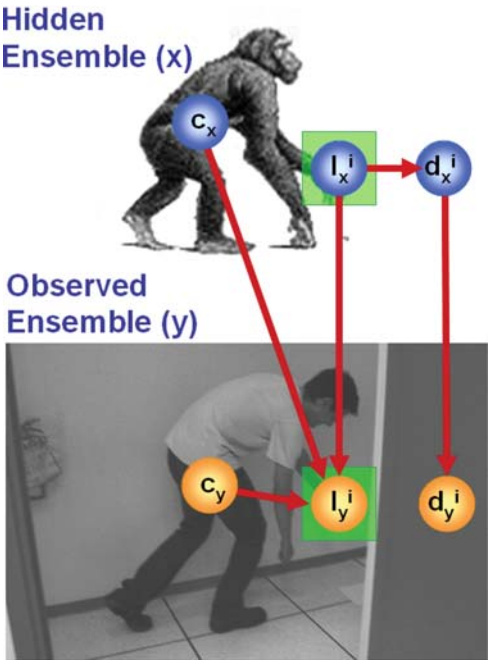


Figure 4. The probabilistic graphical model. The Bayesian dependencies are illustrated using the arrows between variables. The dependencies are illustrated only for one patch in the ensemble (the i-th patch). Observed variables are marked in "orange"; Hidden variables are marked in "blue". c_x and c_y are the "origin" of the hidden and observed ensembles, respectively. l_x^i and l_y^i are the locations (in absolute coordinates) of the i-th patch in the hidden and observed ensembles; d_x^i and d_y^i are the descriptor vectors of the i-th patch in each ensemble.

$P(x, y)$ of Eq. (1) using Eqs. (2)–(4) as follows:

$$P\left(c_x, d_x^1, \ldots, l_x^1, \ldots, c_y, d_y^1, \ldots, l_y^1\right) = \alpha \prod_i P\left(l_y^i \mid l_x^i, c_x, c_y\right) P\left(d_y^i \mid d_x^i\right) P\left(d_x^i \mid l_x^i\right) \quad (5)$$

For any hidden ensemble assignment with non-zero likelihood, we define the *composition cost* as the minus log likelihood function:

$$-\log P\left(c_x, d_x^1, \ldots, l_x^1, \ldots, c_y, d_y^1, \ldots, l_y^1\right) = \sum_i -\log P\left(l_y^i \mid l_x^i, c_x, c_y\right) + \sum_i -\log P\left(d_y^i \mid d_x^i\right) + \alpha_1 \quad (6)$$

Where $\alpha_1 = \log(\alpha)$ is a constant. The first term is the overall cost of local geometric deformations in the ensemble, while the second term is the overall cost of appearance/descriptor deformations in the ensemble.

In our formulation, the covariance matrices S_D and S_L are constant. The reason for this is that our approach is *non-parametric and data-driven*. Parametric approaches to object recognition (such as Felzenszwalb and Huttenlocher (2005), and Fergus et al.) allow for multiple learnt covariance matrices, each associated with a different part in the model. This is a good approach when the recognition task is restricted to a few known predefined classes, each with its pre-defined parts and parameters. This, however, is not the setting in our case, where there is no predefined notion of what we are looking for, yet, we want to be able to detect subtle irregularities compared to the examples. Our model is therefore non-parametric and its generalization capabilities do not rely on parameter tuning, but rather on the diversity of the examples in the database. In that sense, our non-parametric modelling bears resemblance to the non-parametric treatment of Leibe et al.

In our implementation we have set the covariance matrices S_D and S_L to simple scalar variance determined empirically. This simple setting was satisfactory for our experiments. Note that in this setting, the sole purpose of these two parameters is to properly weight the costs of geometric deformations and appearance/descriptor deformations. Moreover, note that these are the only parameters in the model, and therefore requires very little parameter tuning.

4.2. Belief Propagation Inference

Given an observed ensemble, we seek a hidden database ensemble ,which maximizes its MAP (maximum a-posterior probability) assignment. This is done using the above statistical model, which has a simple and exact Belief Propagation algorithm (Yedidia et al., 2003). According to Eq. (5) the MAP assignment can be written as:

$$\max_X P\left(c_x, d_x^1, \ldots, l_x^1, \ldots, c_y, d_y^1, \ldots, l_y^1\right) = \alpha \prod_i \max_{l_x^i} P\left(l_y^i \mid l_x^i, c_x, c_y\right) \max_{d_x^i} P\left(d_y^i \mid d_x^i\right) P\left(d_x^i \mid l_x^i\right) \quad (7)$$

This expression can be phrased as a message passing algorithm in the graph of Fig. 4. First we compute for each patch the message m_{dl}^i passed from node d_x^i to node l_x^i regarding its belief in the location l_x^i:

$$m_{dl}^i\left(l_x^i\right) = \max_{d_x^i} P\left(d_y^i \mid d_x^i\right) P\left(d_x^i \mid l_x^i\right) \quad (8)$$

Namely, for each observed patch, compute all the candidate database locations l_x^i with high descriptor similarity. Next, for each of these candidate database locations, we pass a message about the induced possible origin locations c_x in the database:

$$m_{lc}^i(c_x) = \max_{l_x^i} P\left(l_y^i \mid l_x^i, c_x, c_y\right) m_{dl}\left(l_x^i\right) \quad (9)$$

At this point, we have a candidate list of origins suggested by each individual patch. To compute the likelihood of an entire ensemble assignment, we multiply the beliefs from all the individual patches in the ensemble:

$$m_c(c_x) = \prod_i m_{lc}^i(c_x) \tag{10}$$

The inference performed by this algorithm is a MAP inference. Therefore, something that occurred once in the examples database is equally likely as something that occurred many times. This formulation is useful in many applications, however, there may be applications where we would like the frequency of occurrence in the database to affect the likelihood of an ensemble. A simple modification of the above algorithm allows to compute likelihood instead of MAP, by transforming the inference algorithm from a max-product to a sum-product.

4.3. Estimating the Likelihood of a Query Point

For each point in the query, we try to compose a large region around it. This is done by checking the validity of a large region surrounding every point, using the above inference process (by computing a *query region* likelihood). This point participates in many query regions. We define the likelihood of a *query point* as the maximal likelihood of a region containing that point. Therefore, a point in the query will have a high likelihood, if there exists a large region containing it, with a corresponding similar database region. This way, we can compose queries with partial occlusion of objects, since points which are near the boundary are contained in a large region inside the object. However, partial occlusions might create small contiguous regions of objects, which cannot be composed with high likelihood using our current inference algorithm.

We would like the region that we compose around every point to be as large as possible, because the larger the region is, the higher the evidence that the point is not irregular. However, there are cases in which a "regular" observed ensemble cannot be fully composed by a single database ensemble (e.g., due to partial occlusion). In those cases (which are not very frequent), we reduce the size of the observed region (e.g., by 25%) and repeat the inference process without the discarded patches. We penalize the overall ensemble likelihood score for each patch we discard. In terms of Eq. (6) we add a constant cost penalty for every patch we discard. The magnitude of the penalty term, reflects the importance we attribute to the composition region size.

Handling Ensembles of Different Sizes: In order to detect irregular regions in an entire observation, we can simply threshold the composition cost in Eq. (6). However, there may be cases where the size of the observed ensemble would be different (e.g., because of non-informative regions, regions excluded from analysis, data boundaries, etc.). In order to compare composition cost of ensembles of different sizes, a normalization is required. We use a normalization based on the statistical significance of the composition cost. We define the null-hypothesis H_0 such that each observed ensemble was generated using the statistical model defined above. Therefore, the statistical significance of a composition cost C_0 can be measured by the *pvalue* $Pr(C > C_0 \mid H_0)$. Assuming the null-hypothesis, and given the hidden ensemble, each term in the composition cost in Eq. (6) is distributed χ^2 and the overall cost is also distributed χ^2. These distributions can be used to compute the pvalue which "normalizes" the composition cost for ensembles of different sizes.

5. An Efficient Inference Algorithm

A naive implementation of the message passing algorithm presented in Section 4.2 is very inefficient, since independent descriptor queries are performed for each patch in the observation ensemble, regardless of answers to previous queries performed by other patches. This results in a complexity of $O(Nk)$ where N is the number of patches in the database (e.g., 100,000 patches for a one-minute video database) and k is the number of patches in the ensemble (e.g., 256). Moreover, we should scan the entire query (the new image or new video sequence), which results in a total complexity of $O(Nkq)$, where q is the number of patches in the query. The complexity is prohibitive for real applications, because each of the terms (N, k and q) is not negligible. In this section we show how to significantly reduce the complexity without sacrificing accuracy.

5.1. The Progressive Elimination Process

The patches in the observed ensemble are related by a certain geometric arrangement. We can use this knowledge for an efficient search by *progressive elimination* of the search space in the database: We compute the message m_{dl}^i for a small number of patches (e.g., 1). The resulting list of possible candidate origins induces a very restricted search space for the next patch. The next patch, in turn, eliminates additional origins from the already short list of candidates, etc. This process is illustrated in Fig. 5. In order to speed-up the progressive elimination, we use *truncated* Gaussian distributions (truncated after 4σ), in Eqs. (2) and (3). Therefore, these distribution give a likelihood of zero to high patch deformations in terms of geometry or appearance/descriptor. The search of the first patch costs $O(N)$. We keep only the best c candi-

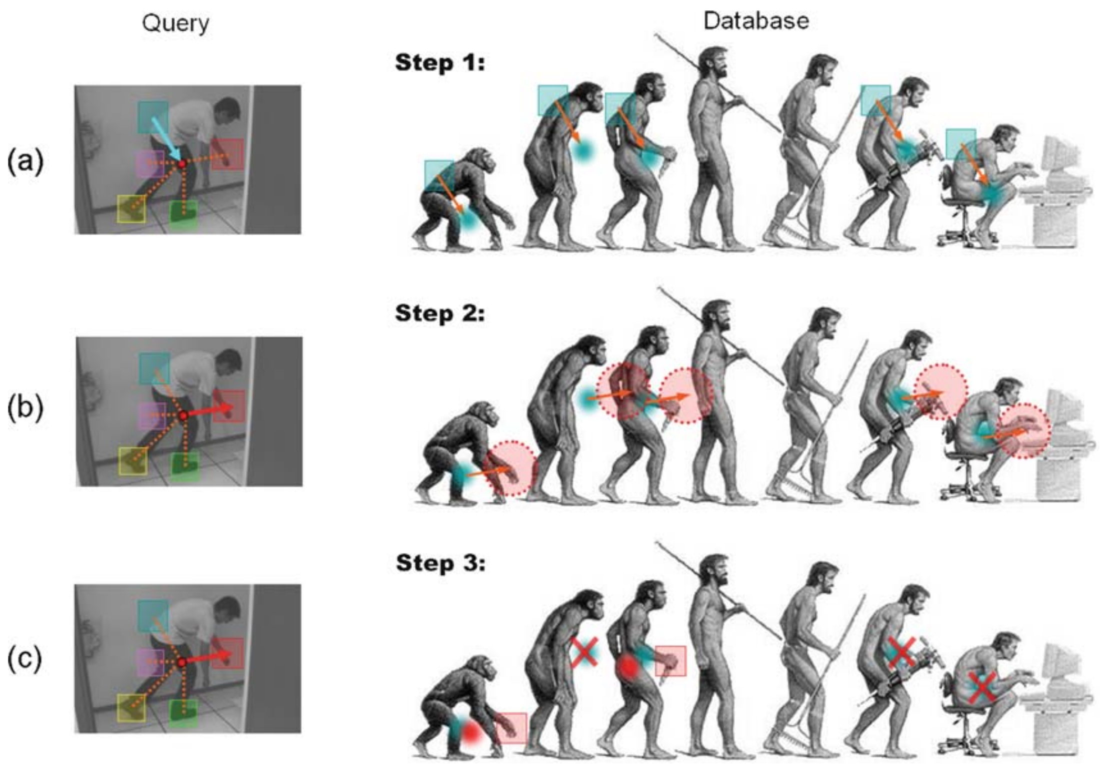


Figure 5. The progressive elimination process. Step 1: Searching for the first patch (blue square) in the database yields several possible locations. Each location induces probability for the location of the hidden ensemble center (blue circles). Step 2: We use the PDF of the hidden ensemble center induced by the first (blue) patch in order to search for the second (red) patch only in non-zero likelihood regions (dotted red circles). (c) Step 3: Each detected location of the red patch induces another PDF for the hidden ensemble center. Locations where one of the PDFs is zero are eliminated. We proceed to process all the other patches in the ensemble in the same way.

date origins from the list proposed by the first patch (in our implementation, $c = 50$). The second patch is now restricted to the neighborhoods of c locations. The third will be restricted to a much smaller number of neighborhoods. Thus, in the worst case scenario, our complexity is $O(N + kc) \approx O(N)$. In contrast, the complexity of the inference process in Felzenszwalb and Huttenlocher (2005) and Leibe et al. is $O(Nk)$, while the complexity of the "constellation model" (Fergus et al.) is *exponential* in the number of patches. The above proposed reduction in complexity is extremely important for enabling video inference with ensembles containing hundreds of patches. Note that limiting the number of candidate origin points to c candidates might be problematic: For instance, if the first patch we choose is non-informative (i.e., single edge), then choosing the best c candidates is arbitrary and we might discard the globally optimal ensemble. In practice, other components of our inference algorithm (multi-scale strategy, predictive search, and scanning the observation) eliminate this risk. Note that if we assume truncated Gaussian distributions (or other finite support distributions), and if searching for the first few patches yields less than c candidate locations, then the progressive elimination process guarantees an exact solution, because we only discard candidates with zero likelihood. Note that this entails that under such conditions we can offer an exact inference which is equivalent to Belief Propagation with reduced complexity. Moreover, we know during the inference if the result is exact (optimal) or if it is only an approximation.

5.2. Multi-Scale Search

To further speedup the elimination process, we use a coarse-to-fine strategy (both in space and in time). We choose the first searched patches from a coarse scale, for two reasons: (i) There is a much smaller number of coarse patches in the database than fine patches (thus decreasing the effective N in the first most intensive step), and (ii) coarse patches are more discriminative because they capture information from large regions. This eliminates candidate origins of database ensembles very quickly. We proceed until we process all the coarse scale patches in the observed ensemble. Then we project the candidate origin points to the next finer scale and continue to process patches in the finer scale (both in space and in time). We proceed in this multi-scale manner to process all the patches in the observed ensemble. The complexity of the multi-scale search is $O(N_0 + kc)$, where N_0 is the number of patches in the coarsest scale of the space-time pyramid.

5.3. *Efficient Database Storage and Retrieval*

A simple implementation of the database would be to use an array of patch descriptors and search it linearly. However, time and space complexity can be improved significantly for database retrieval and storage, respectively. Storage space can be reduced significantly by keeping approximations of the descriptor vectors. For instance, all the descriptor vectors can be projected on a low dimensional linear space using standard techniques such as PCA and ICA. In addition, vector quantization techniques (such as Kmeans, or Jurie and Triggs (2005)) can be used to cluster groups of descriptors. The result of projection and quantization is that there are less descriptor types to store, and each descriptor vector is shorter. Another benefit is that database retrieval time is reduced. Note that projection and quantization introduce errors in the descriptor vectors. We can eliminate the error if each 'compressed' descriptor contains a link to the original descriptor. In this case, storage space would not be reduced, but the retrieval time would be reduced. A closely related approach to reduce database retrieval time is to use better data structures for storing the descriptor vectors, such as KD-trees and hash-tables for finding approximate nearest neighbors. These data-structures enable fast range queries (finding all elements in the database in a certain range around a given element). The resulting time complexity is $O(Range(N_0) + kc)$, where $Range(N_0) \ll N_0$ is the cost of a range query in the database data structure with N_0 elements (patches) in it.

5.4. *Using Predictive Search*

So far we assumed that the composition algorithm described above is applied to all the points in the observation, independently of each other. This is usually wasteful as neighboring observed ensembles tend to have neighboring hidden ensembles in the database. We utilize this fact to speed up the composition by predicting the values of hidden ensemble variables in space and in time. By using all the previously composed ensembles in the vicinity of the current ensemble (in space and in time), we predict the location of the hidden ensemble center and the identity of the hidden patches in the database, using knowledge obtained for the overlapping observed patches. We use the simplest prediction: Given a neighboring observed ensemble ($\tilde{\mathbf{y}}$ and its corresponding detected database ensemble $\tilde{\mathbf{x}}$) we predict some of the hidden variables in hidden ensemble $\mathbf{x}$ corresponding to a new observed ensemble $\mathbf{y}$. We predict the hidden ensemble center c_x using:

$$c_x = \tilde{c_x} + c_y - \tilde{c_y} \tag{11}$$

Moreover, for each observed patch (l_y^i, d_y^i), which participated in the predicting ensemble $(\tilde{l_y^i}, \tilde{d_y^i}) = (l_y^j, d_y^j)$ we predict the corresponding hidden variables $(l_x^i, d_x^i) = (\tilde{l_x^j}, \tilde{d_x^j})$. The rest of the hidden variables, which are not predicted, can be inferred very quickly using the progressive elimination process. Note that for neighboring ensembles, most of the observed patches overlap, therefore the complexity of composing a new ensemble is very low.

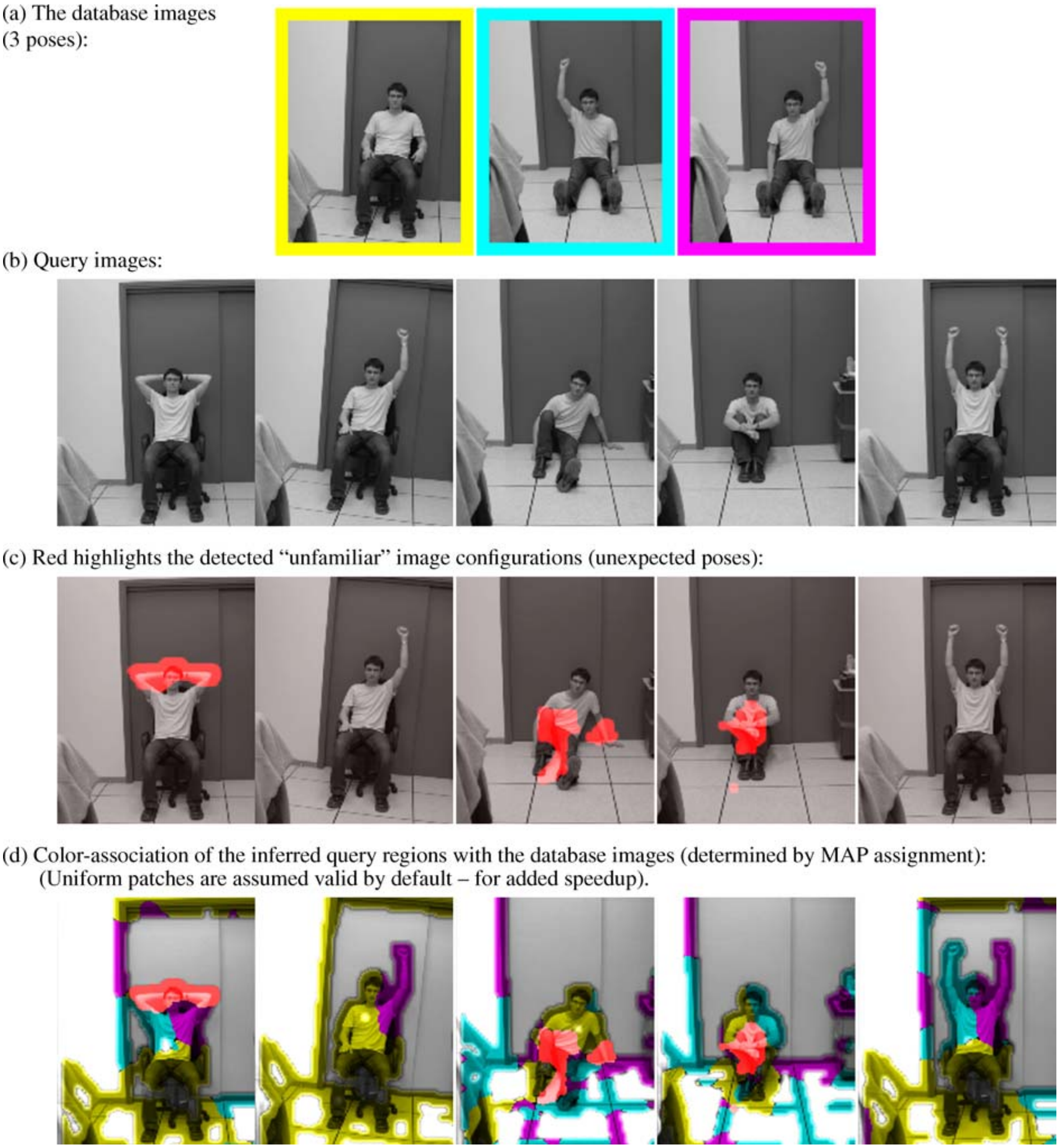
In cases where the prediction is bad and hence results in a low quality composition (i.e., low likelihood of the observed region), we discard the prediction results and use the usual inference over the entire database. Thus, the predictive search does not prevent detection elsewhere in the database. However, in most cases the predictive search is quite accurate and reduces the inference time considerably. Assume that there is a 'chain' of valid predictions of length r. The cost of predicting an ensemble in this chain is $O(k)$. Therefore the total complexity of such a chain is $O(Range(N_0) + kc + kr)$ instead of $O(Range(N_0)r + krc)$ without prediction. Besides significantly reducing total inference time, prediction actually improves the accuracy of inference. This is because regions where the composition was accurate, propagate information to regions with less certainty (e.g., the leg of a standing person has less certainty than the upper part of the body).

6. Applications

The approach presented in this paper gives rise to a variety of applications which involve detection of irregularities in images and in videos:

6.1. *Detecting Unusual Image Configurations*

Given a database of example images, we can detect unusual things in a new observed image (such as objects never seen before, new image patterns, etc.) An example is shown in Fig. 6. Images of three different poses are provided as a database (Fig. 6(a)). Images of other poses are provided as queries (Fig. 6(b)). New valid poses (e.g., a man sitting on the chair with both arms up, a man sitting on a chair with one arm up) are automatically inferred from the database, even though they have never been seen before. New pose parts which cannot be inferred from the three database images are highlighted in red as being "unfamiliar" (Fig. 6(c)). Figure 6(d) visually indicates the database image which provided most



(a) The database images (3 poses):

(b) Query images:

(c) Red highlights the detected "unfamiliar" image configurations (unexpected poses):

(d) Color-association of the inferred query regions with the database images (determined by MAP assignment): (Uniform patches are assumed valid by default – for added speedup).

Figure 6. *Detection of irregular image configurations.* New valid poses are automatically inferred from the database (e.g., a man sitting on the chair with both arms up, a man sitting on a chair with one arm up), even though they have never been seen before. New pose parts which cannot be inferred from the three database images are highlighted in red as being "unfamiliar".

evidence for each pixel in the query images (i.e., it tells which database image contains the largest most probable region of support for that pixel. Note, however, that these are *not* the regions of support themselves). Uniform patches (with negligible image gradients) are assumed valid by default and discarded from the inference process (for added speedup).

6.2. *Spatial Saliency in a Single Image*

Given a single image (i.e., no database), salient image regions can be detected, i.e., image regions which stand out as being different than the rest of the image. This is achieved by measuring the likelihood of each image region (the "query") relative to the remaining portions of

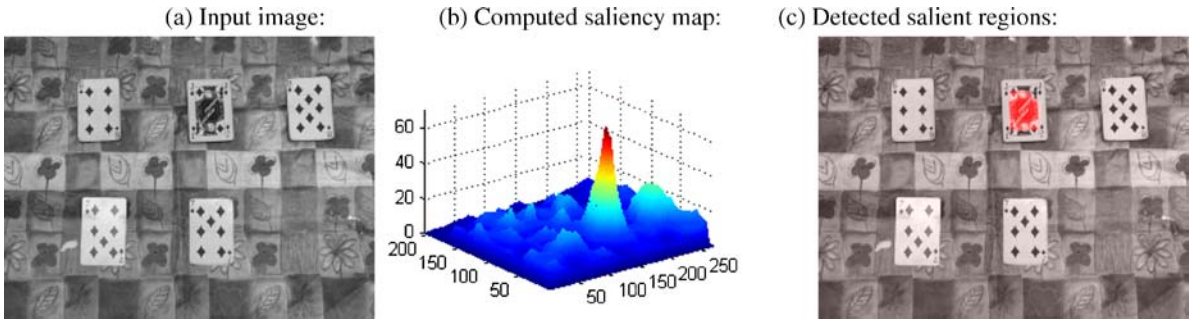


Figure 7. Identifying salient regions in a single image (no database; no prior information). The Jack card was detected as salient. Note that even though the diamond cards are different from each other, none of them is identified as salient.

the same image (the "database" used for inferring this particular region). This process is repeated for each image region. (This process can be performed efficiently by adaptively adding and removing the appropriate descriptors from the "database" when proceeding from the analysis of one image region to the next). Such an example is shown in Fig. 7. This approach can be applied to problems in automatic visual inspection (inspection of computer chips, goods, etc.).

6.3. *Detecting Suspicious Behaviors*

Given a small database of sequences showing a few examples of valid behaviors, we can detect suspicious behaviors in a new long video sequence. This is despite the fact that we have never seen all possible combinations of valid behaviors in the past, and have no prior knowledge of what kind of suspicious behaviors may occur in the scene. These are automatically composed and inferred from space-time regions in the database sequence. An example is shown in Fig. 8, which shows a few sample frames from a 2-minute-long video clip, along with detected suspicious behaviors. For full videos see `www.wisdom.weizmann.ac.il/~vision/Irregularities.html`. The result of our algorithm is a dense likelihood map. In our video examples, a *single* threshold was selected for an entire video sequence query. More sophisticated thresholding methods (hysteresis, adaptive threshold, etc.) can be used.

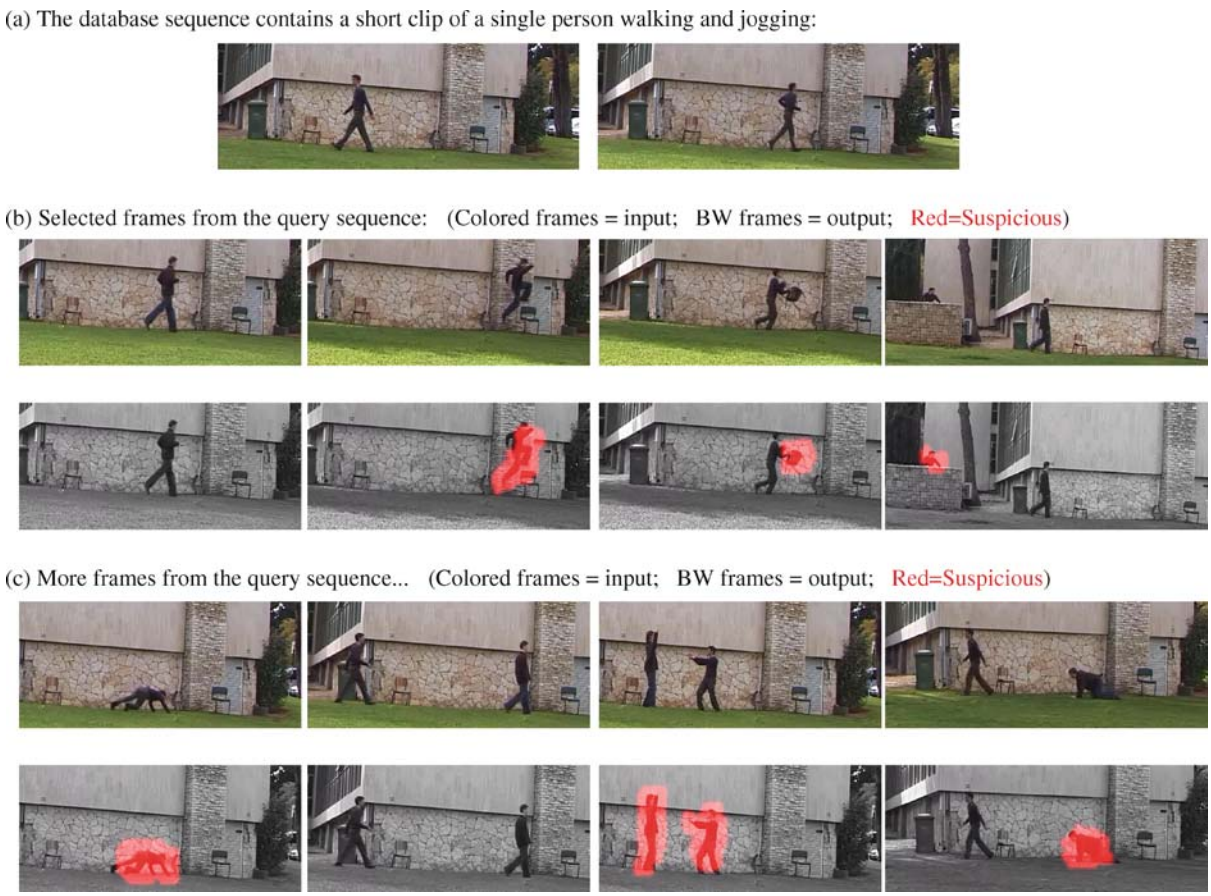
Note that because our space-time patch descriptors were based on temporal derivatives (see Section 3.1), the detection results are invariant to different *static* backgrounds in the query and example database sequences. (In fact, because we detect suspicious dynamic behaviors, we do not process the static regions, which reduces run-time considerably.) An important property of our approach is that we can incrementally and adaptively update the database when new regular/valid examples are provided, simply by appending their raw descriptors and locations to the database. No "relearning" process is needed. This is essential in the context of detecting suspicious behaviors, should a detected suspicious behavior be identified as a false alarm. In such cases, the database can be updated by appending the new example, and the process can continue.

6.4. *Spatio-Temporal Saliency in Video*

Using our approach we can identify salient behaviors within a single video sequence, without any database or prior information. For example, one person is running amongst a cheering crowd. The behavior of this person is obviously salient. In this case, saliency is measured relative to all the other behaviors observed at the same time. The "validity" of each space-time video segment (the "query") is measured relative to all the other video segments within a small window in time (the "database" for this particular video segment). This process is repeated for each video segment. Such an example is shown in Fig. 9. For full videos see `www.wisdom.weizmann.ac.il/~vision/Irregularities.html`. Video saliency can also be measured relative to other temporal windows. E.g., when the saliency is measured relative to the entire video, behaviors which occur only once will stand out. Alternatively, when the saliency is measured relative to the past (all previous frames), new behaviors which have not previously occurred will be spotted. This gives rise to a variety of applications, including video synopsis.

6.5. *Automatic Visual Inspection (Quality Assurance)*

Our approach can be used for automatic visual inspection. Automatic visual inspection is widely used for quality assurance in the manufacture of goods, electronic


(a) The database sequence contains a short clip of a single person walking and jogging:

(b) Selected frames from the query sequence: (Colored frames = input; BW frames = output; Red=Suspicious)

(c) More frames from the query sequence... (Colored frames = input; BW frames = output; Red=Suspicious)

Figure 8. *Detection of suspicious behaviors.* New valid behavior combinations are automatically inferred from the database (e.g., two men walking together, a different person running, etc.), even though they have never been seen before. behaviors which cannot be inferred from the database clips are highlighted in red as being "suspicious". For full videos see `www.wisdom.weizmann.ac.il/~vision/Irregularities.html`

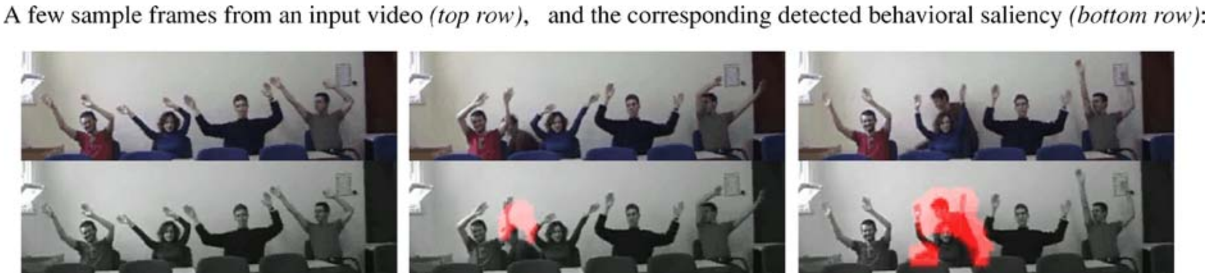

A few sample frames from an input video *(top row)*, and the corresponding detected behavioral saliency *(bottom row)*:

Figure 9. *Detecting salient behaviors in a video sequence (no database and no prior information).* Saliency is measured relative to all the other behaviors observed at the same time. In this example, all the people wave their arms, and one person behaves differently. For full videos see `www.wisdom.weizmann.ac.il/~vision/Irregularities.html`

printed boards, wafers, etc. One of the main problems in automatic inspection is describing all the possible correct patterns. In some cases, an exact reference for comparison can be supplied. In those cases automatic inspection reduces to a simple problem of pattern matching with change detection. However, there are many important complex cases where it is meaningless or impossible to provide a reference for comparison, (e.g., because of the combinatorial complexity of the space of "good" cases). We address such cases using our approach for detecting irregularities. By supplying a few examples of expected/normal patterns (for goods, printed boards, wafers, photomasks, flat panel displays, ceramic tiles, fabric, fruits, etc.) we can try to generalize from the examples and compose new observations that were never seen before. Regions with low composition likelihood

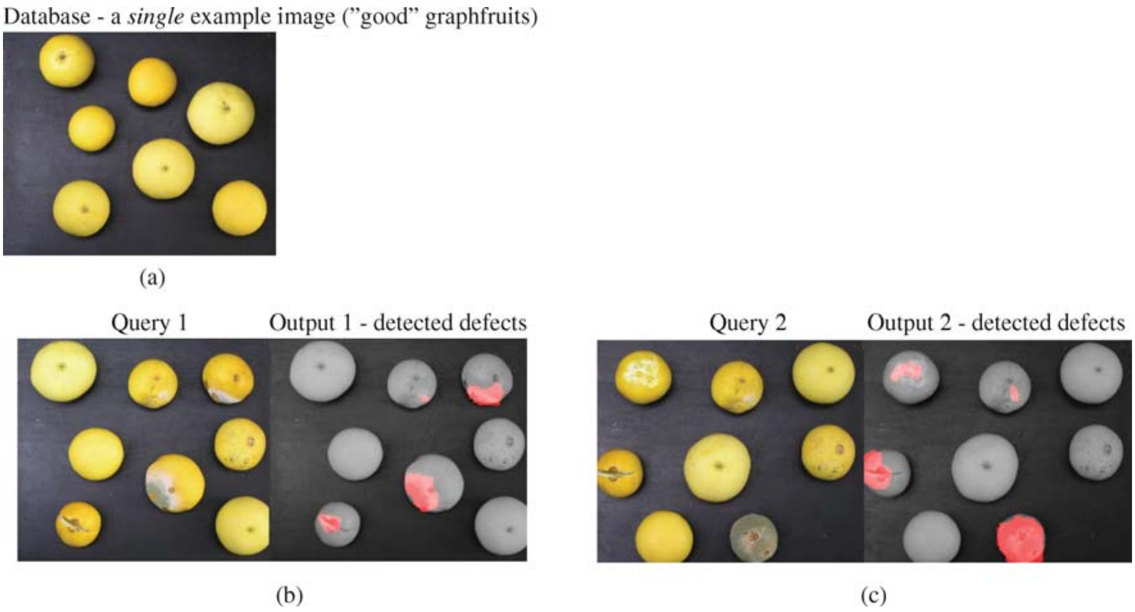

Database - a *single* example image ("good" graphfruits)
(a)
Query 1
Output 1 - detected defects
Query 2
Output 2 - detected defects
(b)
(c)

Figure 10. *Detection of defects in grapefruit images.* Using the single image (a) as a "database" of high quality grapefruits, we can detect defects in different grapefruits at different arrangements in images (b),(c). In both image pairs the input image is to the left and the output image is to the right. Detected defects are highlighted in "red".

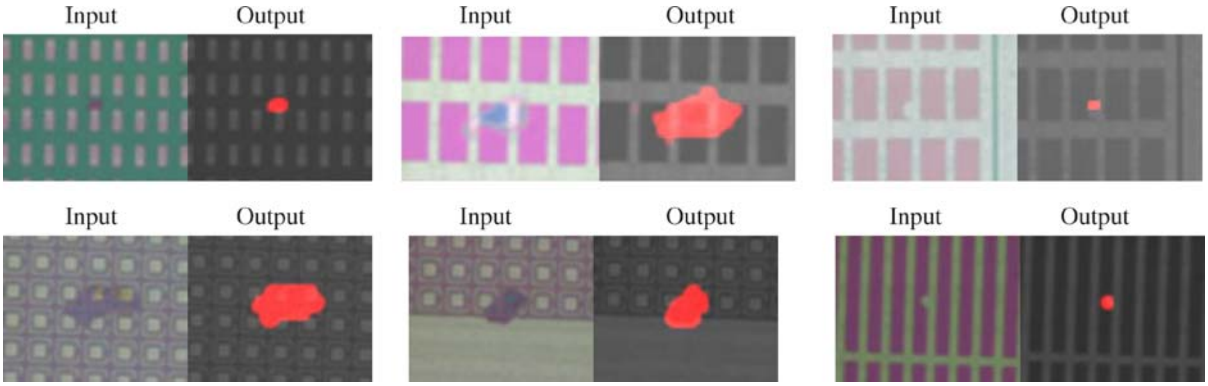


Figure 11. *Detection of defects in wafer images (No database and no prior information).* Wafers tend to exhibit repeating structures. This can be utilized using our saliency approach to detect defects without any database. In each example, the left image is the input, the right image is the output. Detected defects are highlighted in "red".

will be considered as defects. One such example is shown in Fig. 10 for fruit inspection.

Often, inspected products exhibit repeating patterns (e.g., wafers, fabric, flat panel displays). In these cases we can use our saliency approach to detect defects without any prior examples. This is illustrated in Fig. 11 for wafer inspection and in Fig. 12 for fabric inspection. For the examples shown we have used patch descriptors based on RGB or gray levels values accordingly. We have used a Gaussian distribution for modelling descriptor similarity. Our approach, however, is not restricted to this particular choice of descriptors.

7. Conclusion

We address the problem of detecting irregularities in visual data (images or video). The term "irregular" depends on the context in which the "regular" or "valid" are defined. Yet, it is not realistic to expect explicit definition of all possible valid configurations for a given context. We pose the problem of determining the validity of visual data as a process of constructing a puzzle: We try to compose a new observed image region or a new video segment ("the query") using chunks of data extracted from previous visual examples ("the database"). Regions

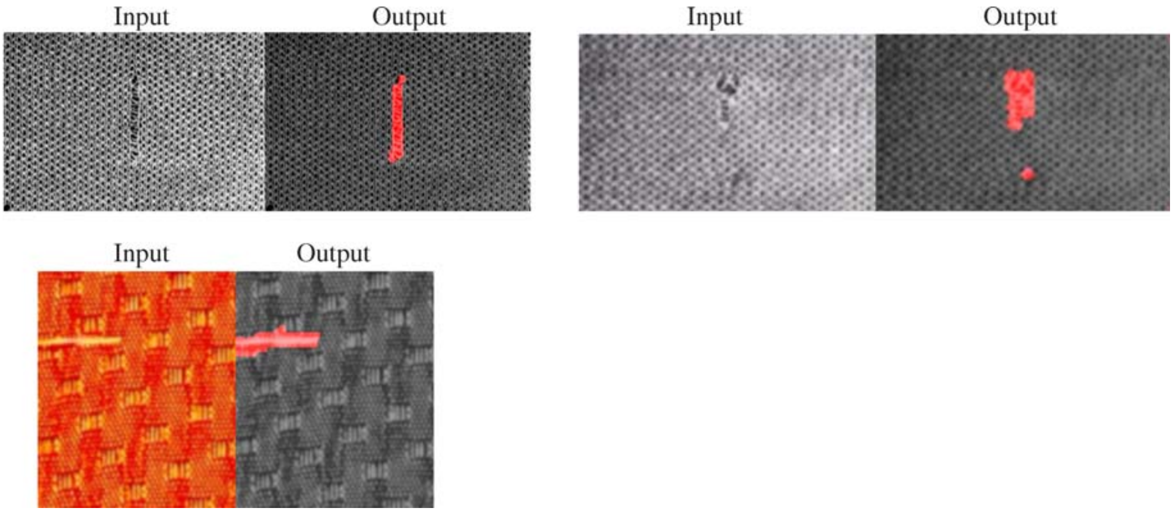


Figure 12. Detection of defects in fabric images (No database and no prior information). Fabric tend to exhibit nearly repeating textures and patterns with small non-rigid deformations. This can be utilized using our saliency approach to detect defects without any database. Detected defects are highlighted in "red".

in the observed data which can be composed using large contiguous chunks of data from the database are considered very likely, whereas regions in the observed data which cannot be composed from the database (or can be composed, but only using small fragmented pieces) are regarded as unlikely/suspicious. We refer to this process as "inference by composition". It allows to generalize from just a *few* examples as to what is regular and what is not in a much larger context. The composition process is implemented as an efficient inference algorithm in a probabilistic graphical model, which accommodates for small spatio-temporal deformations between the query and the database.

"Inference by composition" can also be used to detect saliency in visual data without any prior examples. For this purpose we regard each image region as a "query", and try to compose it using the remainder parts of the image (the "database"). This is repeated in turn for all image regions. Salient regions will be detected as such which cannot be "explained" (composed) using other parts of the image. This leads to a new definition of the term *saliency* in visual data. In the case of video data, those regions are spatio-temporal, and the salient video regions correspond to salient behaviors.

Our "inference by composition" approach is general and can therefore address a wide range of problems in a single unified framework. Its generality stems from the fact that it does not resort to any pre-learned class-based models. We demonstrated applications of this approach to detecting suspicious behaviors, salient behaviors, prominent image regions, defects in goods and products.

Our current algorithm has two main limitations: (i) Although occlusions can be handled to some extent, it cannot handle extreme occlusions (such as when only small fragmented parts of the object are visible). (ii) The time and memory complexity of our current inference algorithm is linear in the size of the example database. This is obviously problematic for very large databases. These two problems are a topic of our future research.

Acknowledgments

This work was supported in part by the Israeli Science Foundation (Grant No. 281/06) and 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

Bart, E. and Ullman, S. 2004. Class-based matching of object parts. In *Video Register04*, p. 173.

Boiman, O. and Irani, M. 2005. Detecting irregularities in images and in video. In *ICCV05*, pp. I:462–469.

Efros, A.A. and Leung, T.K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*, pp. 1033–1038.

Felzenszwalb, P. and Huttenlocher, D. 2005. Pictorial structures for object recognition. *IJCV*, 61(1):55–79.

Fergus, R., Perona, P., and Zisserman, A. 2003. Object class recognition by unsupervised scale-invariant learning. In *CVPR03*.

Freeman, W., Pasztor, E., and Carmichael, O. 2000. Learning low-level vision. *IJCV*, 40:25–47.

Honda, T. and Nayar, S. 2001. Finding 'anomalies' in an arbitrary image, pp. II:516–523.

Itti, L., Koch, C., and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *PAMI*.

Ivanov, Y. and Bobick, A. 1999. Recognition of multi-agent interaction in video surveillance. In *ICCV*.

Jurie, F. and Triggs, B. 2005. Creating efficient codebooks for visual recognition. In *ICCV05*, pp. I:604–610.

Laptev, I. and Lindeberg, T. 2003. Space-time interest points. In *ICCV03*, pp. 432–439.

Leibe, B., Leonardis, A., and Schiele, B. 2004. Combined object categorization and segmentation with an implicit shape model. In *ECCV04 Workshop on Statistical Learning in CV*.

Lowe, D. 2004. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110.

Shechtman, E. and Irani, M. 2005. Space-time behavior based correlation, pp. I:405–412.

Sivic, J. and Zisserman, A. 2003. Video google: A text retrieval approach to object matching in videos. In *ICCV*.

Stauffer, C. and Grimson, E. 2000. Learning patterns of activity using real-time tracking. *PAMI*.

Wexler, Y., Shechtman, E., and Irani, M. 2004. Space-time video completion. In *CVPR04*, pp. I:120–127.

Yedidia, J.S., Freeman, W.T., and Weiss, Y. 2003. Understanding belief propagation and its generalizations, pp. 239–269.

Zhong, H., Shi, J., and Visontai, M. 2004. Detecting unusual activity in video. In *CVPR04*, pp. II:819–826.