"Clustering by Composition"—Unsupervised Discovery of Image Categories

Alon Faktor, *Student Member, IEEE* and Michal Irani, *Member, IEEE*

Abstract—We define a "good image cluster" as one in which images can be easily composed (like a puzzle) using pieces from each other, while are difficult to compose from images outside the cluster. The larger and more statistically significant the pieces are, the stronger the affinity between the images. This gives rise to unsupervised discovery of very challenging image categories. We further show how multiple images can be composed from each other simultaneously and efficiently using a collaborative randomized search algorithm. This collaborative process exploits the "wisdom of crowds of images", to obtain a sparse yet meaningful set of image affinities, and in time which is almost linear in the size of the image collection. "Clustering-by-Composition" yields state-of-the-art results on current benchmark data sets. It further yields promising results on new challenging data sets, such as data sets with very few images (where a 'cluster model' cannot be 'learned' by current methods), and a subset of the PASCAL VOC data set (with huge variability in scale and appearance).

Index Terms—Image clustering, image affinities, category discovery, unsupervised object recognition

1 Introduction

As the amount of visual information in the web grows, there is an increasing need for methods to organize it and search in it. In many cases, the images do not contain any labels or annotations, so we can rely only on their visual content. Moreover, these images may contain objects or scenes of any possible category in the world. Thus, it would be unrealistic to use supervised techniques to automatically annotate each and every one of them. An alternative approach is to use unsupervised techniques, such as mining or clustering, to discover patterns and similarities within an image collection.

Great progress has been made in previous years in unsupervised mining and clustering of images which are instances of the same object (e.g., the Notre Dame church), but taken from different viewing points with perhaps large scale differences or occlusions. Example of such works are those of [5], [14], [19], which are based on matching SIFT descriptors around interest points across two images, followed by a geometric verification phase. The reason for their success is mainly due to the fact that these descriptors indeed have good repeatedness across different instances of the same object. However, when dealing with a more general data set of images, where images contain objects of the same semantic category, and not instances of the same object, SIFTs around interest points typically do not perform well.

In this work, we deal with the problem of unsupervised discovery of visual categories within an image collection. The goal here is to group the images into meaningful clusters of images which belong to the same semantic category. Existing work on this problem can be broadly classified to two main families of approaches.

The first family of approaches is based on computing pairwise affinities between images. An example for this is the Pyramid Match Kernel of [7], which measures similarity between images according to the subset of matching local features which is discovered across the images. Other examples of commonly used pairwise affinities can be found in the comparison made by [20]. These affinities are typically based on a global "Bag of Words" representation of the images.

The second family of approaches is based on unsupervised model discovery. This approach iterates between finding clusters of similar images and learning a model which is common to each cluster. Such common cluster models can be common segments [15], common contours [12], [13], common distribution of descriptors [18], [20], representative cluster descriptors [9], [11], etc. Many of these methods require an initialization of the clusters and this is typically done by using pairwise affinities (e.g., [11] uses the pairwise affinities of [7]).

Let us consider the following image collection of Ballet and Yoga images which appears in Fig. 2. Observing these images, there seems to be no single (nor even few) common model(s) shared by all images of the same category. The poses within each category vary significantly from one image to another, there is a lot of foreground clutter (different clothes, multiple people, occlusions, etc.), as well as distracting backgrounds. Therefore, taking an unsupervised model discovery approach in this case will most probably not be beneficial. In the absence of an emerging common cluster model, the performance of unsupervised model discovery methods will be dominated by their initial pairwise affinities. This stresses the need for 'good' image affinities.

In this paper we suggest to perform clustering of image collections by computing sophisticated images affinities

- *The authors are with the Department of Computer Science and Applied Math, Ziskind Building, and The Weizmann Institute of Science. Rehovot, POB 26, Rehovot 76100, Israel.*
E-mail: {alon.faktor, michal.irani}@weizmann.ac.il.

Manuscript received 26 Dec. 2012; revised 28 May 2013; accepted 11 Oct. 2013. Date of publication 15 Dec. 2013; date of current version 12 May 2014.
Recommended for acceptance by T. Tuytelaars.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2013.251

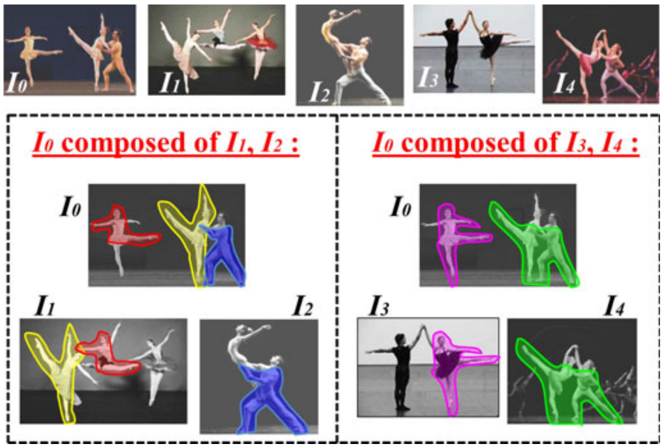


Fig. 1. *Compositions used for computing image affinities.* The affinity between images is high if they can be composed from each other like a simple "puzzle" using large non-trivial regions. Note that these regions are typically NOT 'good image segments' and therefore cannot be extracted ahead of time by image segmentation. What makes them 'good regions' for the composition is the fact that they co-occur across images, yet are statistically significant.

based on "Similarity by Composition" [3]. These kind of affinities are able to handle image collections like the one which appears in Fig. 2. Although the ballet poses differ from each other, one ballet pose can be easily composed from pieces of other ballet poses (Fig. 1). Our approach detects statistically significant regions which co-occur between a small subset of the images. Statistically significant regions are regions which have a low chance of occurring at random. The reoccurrence of such regions across images induces strong and meaningful affinities, even if they do not appear in many images (and thus cannot be identified as a common model).

We define a "good image cluster" as one in which each image can be easily composed using statistically significant pieces from other images in the cluster, while is difficult to compose from images outside the cluster. We refer to this as "clustering by composition". We further show how multiple images can be composed from each other simultaneously and efficiently using a collaborative randomized search algorithm. Each image 'suggests' to other images where to search for similar regions within the image collection. This collaborative process exploits the "wisdom of crowds of images", to obtain a sparse yet meaningful set of image affinities, and in time which is almost linear in the size of the image collection. "Clustering by composition" can be applied to very few images, as well as to larger data sets, and yields state-of-the-art results.

The rest of this paper is organized as follows: In Section 2 we provide a high-level overview of our approach, which is then detailed in Sections 3, 4, and 5. Experimental results can be found in Section 6.

2 Overview of the Approach

Our approach to unsupervised category discovery is based on computing sophisticated affinities between images. We consider two images to be similar if they can be easily composed from meaningful pieces of each other—i.e., share large non-trivial regions. These shared regions are detected using an efficient randomized search algorithm, which is further boosted by using collaborative search between the different images within the collection. This collaborative randomized search generates a sparse set of meaningful affinities in time which is linear in size of the collection and without having to compute all the pairwise affinities. The three main components of our approach are overviewed below.

1. *Image affinities by composition:* Our image affinities are based on "Similarity by Composition" [3]. The notion of composition is illustrated in Fig. 1. The Ballet image I_0 is composed of a few large (irregularly shaped) regions from the ballet images I_1 and I_2. This induces strong affinities between I_0 and I_1, I_2. The larger and more statistically

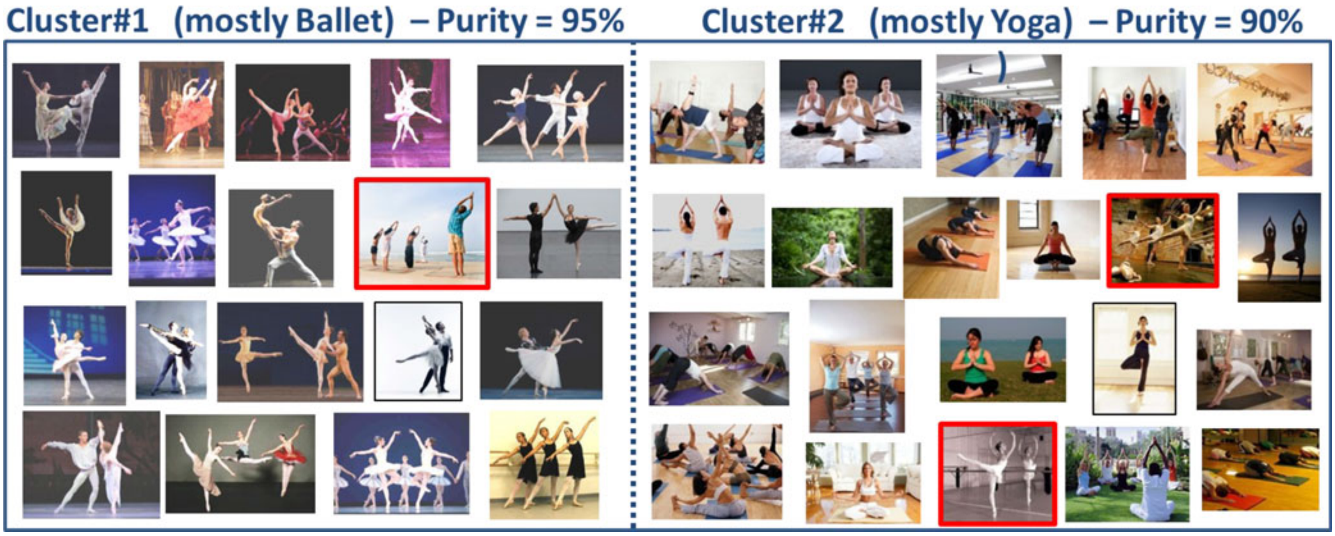

Cluster#1 (mostly Ballet) – Purity = 95%
Cluster#2 (mostly Yoga) – Purity = 90%

Fig. 2. *Clustering Results on our Ballet-Yoga data set.* This data set contains 20 Ballet and 20 Yoga images (all shown here). Images assigned to the wrong cluster are marked in red. We obtain mean purity of 92.5 percent (37 out of 40 images are correctly clustered). Note there seems to be no single (nor even few) 'common model(s)' (e.g., common shapes or segments) shared by all images of the same category. Therefore, methods for unsupervised 'learning' of a shared 'cluster model' will most likely fail (not only due to the large variability within each category, but also due to the small number of images per category).

significant those regions are (i.e., have low chance of occurring at random), the stronger the affinities. The ballet image I_0 could probably be composed of Yoga images as well. However, while the composition of I_0 from other ballet images is very simple (a 'toddler puzzle' with few large pieces), the composition of I_0 from yoga images is more complicated (a complex 'adult puzzle' with many tiny pieces), resulting in low affinities. These affinities are quantified in Section 3 in terms of the "number of bits saved" by describing an image using the composition, as opposed to generating it 'from scratch' at random. To obtain reliable clustering, each image should have 'good compositions' from multiple images in its cluster, resulting in high affinity to many images in the cluster. Fig. 1 illustrates two different 'good compositions' of I_0.

Note that the regions employed in our composition are not the standard image segments commonly used as image regions (as in [8], [15]). They are not confined by image edges, may be a part of a segment, or may contain multiple segments. Such regions are not extracted ahead of time via image segmentation, but are rather determined by their co-occurrence in another image. In other words, what makes them 'good regions' is NOT them being 'good segments', but rather the fact that they co-occur across images, yet, are statistically significant (non-trivial).

The regions are image-specific, and not cluster-specific. A region may co-occur only once within an image collection. However, since it has a low chance of occurring at random, the fact that it was found in another image provides high evidence to the affinity between those two images. Such an infrequent region cannot be 'discovered' as a 'common cluster shape' from the collection (as in [12], [13]). Employing the co-occurrence of non-trivial large regions, allows to take advantage of high-order statistics and geometry, even if infrequent, and without the necessity to 'model' it. Our approach can therefore handle also very small data sets with very large diversity in appearance (as in Figs. 2 and 4). These notions are explained in detail in Section 3.

2. Randomized detection of shared regions: When describing our "affinity by composition", we assumed the shared regions are known, but in practice these shared regions have to be automatically detected between the different images. However, since the regions can be of arbitrary size and shape, the region detection is in principle a hard problem even between a pair of images (let alone in a large image collection). Therefore, we propose a randomized search algorithm which ensures that shared regions between two images will be detected efficiently with a high probability.

Our randomized search algorithm is inspired by "PatchMatch" [1], [2], but searches for similar regions (as opposed to similar patches or descriptors). We represent an image by computing N patches (or descriptors) at some dense image grid and consider a region in the image to be an ensemble of patches (or descriptors) along with their relative positions within the region. We show that when randomly sampling descriptors across a pair of images, and propagating good matches between neighboring descriptors, large shared regions can be detected in *linear time* $O(N)$. In fact, the larger the region, the faster it will be found, and with higher probability. We refer to this collaboration between descriptors as exploiting the "wisdom of crowds of pixels" for efficient detection of shared regions between two images. Section 4 explains the randomized region search and provides analysis of its complexity. Examples of detected shared regions can be found in Figs. 7 and 10.

3. Efficient "collaborative" multi-image composition: Clustering a collection of M images, should in principle require computing "affinity by composition" between all pairs of images—i.e. a complexity of $O(NM^2)$, where N is the number of densely sampled patches (or descriptors) in each image. However, we show that when all the images in the collection are composed simultaneously from each other, they can collaborate to iteratively generate with very high probability the most statistically significant compositions in the image collection. Moreover this can be achieved in runtime almost *linear* in the size of the collection (without having to go over all the image pairs).

Images collaborate by 'giving advice' to each other where to search in the collection according to their current matches. For example, looking at Fig. 1, image I_0 has strong affinity to images $I_1, \ldots, I_4$. Therefore, in the next iteration, I_0 can 'encourage' $I_1, \ldots, I_4$ to search for matching regions in each other. Thus, e.g., I_3 will be 'encouraged' to sample more in I_1 in the next iteration. Note that the shared regions between I_1 and I_3 need not be the same as those they share with I_0. For example, the entire upper body of the standing man in I_3 is similar to that of the jumping lady in the center of I_1.

More precisely, we suggest the following collaborative randomized multi-image composition algorithm (see Fig. 3). Our randomized search is applied to the entire collection of images in an iterative fashion. At the first iteration, each descriptor in each image, samples descriptors uniformly from the image collection. After propagating matches between neighboring descriptors, we obtain region matches which are used to compute initial affinities between images. Then in the next iterations, instead of using a uniform sampling, each descriptor samples in a non-uniform way according to suggestions made by other images, to which it had high affinities in the previous iteration.

This process produces within a few iterations a *sparse* set of reliable affinities (corresponding to the most significant compositions). Such sparsity is essential for good image clustering, and is obtained here via 'collective decisions' made by all the images. The collaboration reduces the computational complexity of the overall composition dramatically, to $O(NM)$. In other words, the average complexity per image remains very small—practically linear in the size of the image $O(N)$, *regardless of the number of images M in the collection*! We refer to this as exploiting the "wisdom of crowds of images" for efficient image clustering. These ideas are described in detail in Section 5.

3 COMPUTING IMAGE AFFINITIES BY COMPOSITION

'Similarity by Composition' [3] defines a similarity measure between a 'Query image' Q and a 'Reference image' Ref, according to the 'ease' of composing Q from pieces of Ref. Computing this measure requires finding regions which are shared by the two images. For now, we will assume those regions are given to us, but later in Section 4 we will show how to automatically detect them efficiently. Below

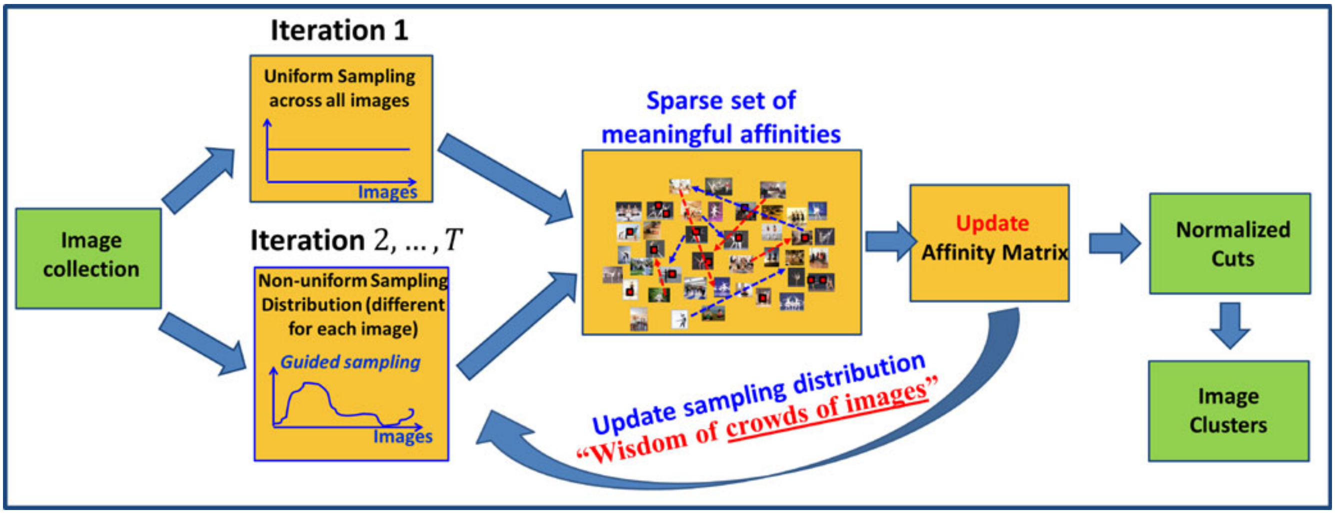

Iteration 1
Uniform Sampling across all images
Images
Image collection
Iteration 2, ..., T
Non-uniform Sampling Distribution (different for each image)
Guided sampling
Images
Sparse set of meaningful affinities
Update Affinity Matrix
Normalized Cuts
Image Clusters
Update sampling distribution
“Wisdom of crowds of images”

Fig. 3. *The full scheme of our collaborative clustering algorithm.* The algorithm starts with uniform random sampling across the entire collection. The connections created between images induce affinities between images. At each iteration, the sampling density distribution of each image is re-estimated according to ‘suggestions’ made by other images. Finally, the resulting affinities are fed to the N-Cut algorithm to obtain the desired clusters.

we review the main concepts of the ‘Similarity by Composition’ framework, as well as describe our own method for computing it.

Estimating the likelihood of a shared region R: A region R is represented as an *ensemble of densely sampled descriptors* $\{d_i\}$, with their relative positions $\{l_i\}$ within R. Let $p(R|Ref,T)$ denote the likelihood to find the region $R \subset Q$ in another image Ref at a location/transformation denoted by T. This likelihood is estimated by the similarity between the descriptors of R and the corresponding descriptors (according to T) in Ref:

$$p(R|Ref,T) = \frac{1}{Z}\prod_i \exp - \frac{|\Delta d_i(Ref,T)|^2}{2\sigma^2}, \tag{1}$$

where $\Delta d_i(Ref,T)$ is the error between the descriptor $d_i \in R \subset Q$ and its corresponding descriptor (via T) in Ref and Z is a normalization factor. We use the following approximation of the likelihood of R, $p(R|Ref)$ according to its best match in Ref:

$$p(R|Ref) \triangleq \max_T p(R|Ref,T)p(T). \tag{2}$$

(This forms a lower bound on the true likelihood).

In our current implementation, the descriptors $\{d_i\}$ were chosen to be two types of descriptors (estimated densely in the image, every other pixel): HOG [6] and local self-similarity (LSS) [16]. These two descriptors have complementary properties: the first captures local texture information, whereas the second captures local shape information while being invariant to texture (e.g., different clothing). We assume for each region R a uniform prior $p(T)$ on the transformations T over all pure shifts. We further allow small local non-rigid deformations of R (slight deviations from the expected (relative) positions $\{l_i\}$ of $\{d_i\}$). Scale invariance is introduced separately (see Section 5.2).

The ‘Statistical Significance’ of a region R: Recall that we wish to detect *large non-trivial* recurring regions across images. However, the *larger* the region, the *smaller* its likelihood according to Eq. (2). In fact, tiny uniform regions have the highest likelihood (since they have lots of good matches in Ref). Therefore, it is not enough for a region to match well, but should also have a low probability to occur at random. This is obtained by:

$$Likelihood\ Ratio(R) = \frac{p(R|Ref)}{p(R|H_0)}. \tag{3}$$

This is the likelihood ratio between the probability of generating R from Ref, versus the probability of generating R at

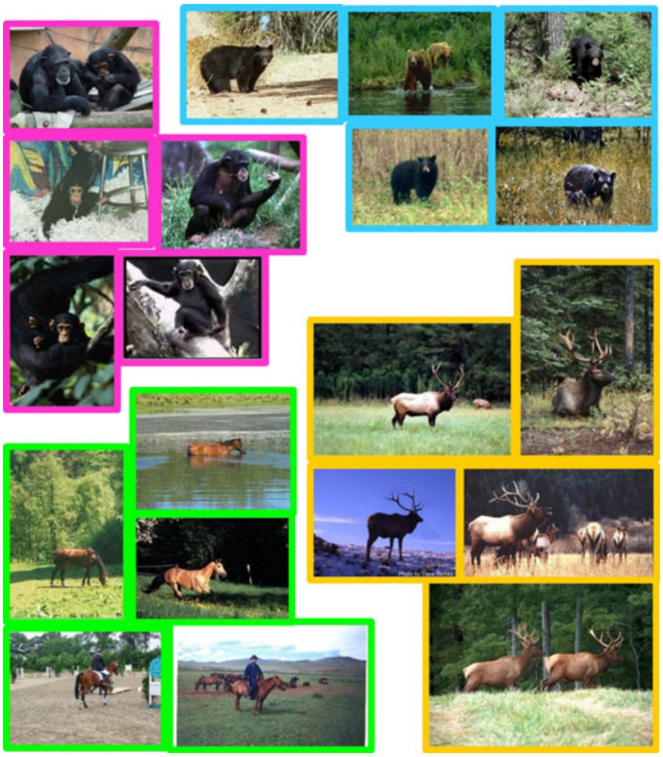


Fig. 4. *Clustering Results on our Animal data set (horses, elks, chimps, bears).* Note how much variability there is within each class, yet how much confusion there are across different classes—for example, the background of the horse and elk is very similar to each other. Our algorithm, however, obtains 100 percent purity.

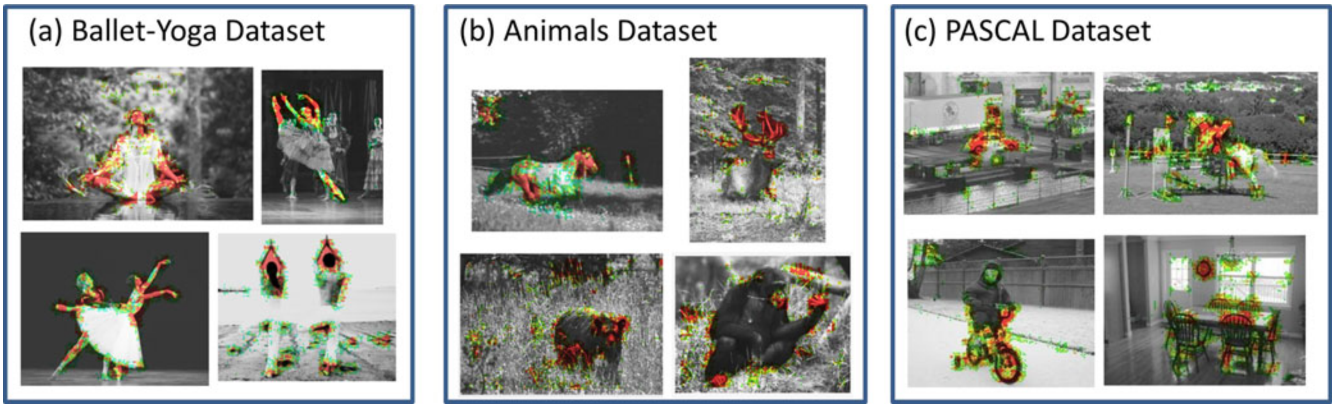


Fig. 5. *Statistical significance of descriptors.* (a) Images from the Ballet-Yoga data set. (b) Images from the Animal data set. (c) Images from the PASCAL data set. Red signifies descriptors (HOG of size 15X15) with the highest statistical significance (descriptors that rarely appear). Green—lower significance; Grayscale—much lower. Note that only the center point of each descriptor is shown here. Statistically significant regions coincide with body gestures (Ballet-Yoga) or object parts (Animals, PASCAL) which are unique and informative to the separation between the different classes in each data set.

random (from a "random process" H_0). $p(R|H_0)$ measures the statistical in significance of a region (high probability = low significance). If a region matches well, but is trivial, then its likelihood ratio will be low (inducing a low affinity). On the other hand, if a region is non-trivial, yet has a good match in another image, its likelihood ratio will be high (inducing a high affinity).

We next present an approach we developed for efficiently estimating $p(R|H_0)$, i.e. the chance of a region R to be generated at random. Assuming descriptors $d_i \in R$ are independent: $p(R|H_0) = \prod_i p(d_i|H_0)$. Given a set of images (the images we wish to cluster, or a general set of natural images), we define D to be the collection of all the descriptors extracted from those images. We define $p(d|H_0)$ to be the probability of randomly sampling the descriptor d from the collection D (or its frequency in D). This can be estimated using Parzen density estimation, but is too time consuming. Instead, we quantize D into a small rough 'codebook' $\hat{D}$ of a few hundred codewords (e.g., using k-means or even just uniform sampling). Frequent descriptors in D will be represented well in $\hat{D}$ (have low quantization error relative to their nearest codeword), whereas rare descriptors will have high quantization error. This leads to the following rough approximation, which suffices for our purpose: $p(d|H_0) = \exp{-\frac{|\Delta d(H_0)|^2}{2\sigma^2}}$, where $\Delta d(H_0)$ is the error between d and its most similar codeword in $\hat{D}$.

Fig. 5 displays $\Delta d(H_0) \propto -\log p(d|H_0)$ for a few images of the Ballet/Yoga, Animals and PASCAL data sets. Red marks descriptors (HOG of size 15X15) with high error $\Delta d(H_0)$, i.e., high statistical significance. Image regions R containing many such descriptors have high statistical significance (low $p(R|H_0)$). Statistically significant regions in Fig. 5a appear to coincide with body gestures that are unique and informative to the separation between Ballet and Yoga. Recurrence of such regions across images will induce strong and reliable affinities for clustering. Observe also that the long horizontal edges (between the ground and sky in the Yoga image, or between the floor and wall in the Ballet images) are not statistically significant, since they are composed of short horizontal edges which occur abundantly in many images. Similarly, statistically significant regions in Figs. 5b and 5c coincide with parts of the animals/objects that are unique and informative for their separation (e.g., the Monkey's face and hands, the Elk's horns, the bicycle's wheels, etc.). This is similar to the observation of [4] that the most informative descriptors for classification tend to have the highest quantization error.

Unlike the common use of codebooks ("bags of descriptors") in recognition, here the codebook is NOT used for representing the images. On the contrary, a descriptor which appears frequently in the codebook is "ignored" or gets very low weight, since it is very frequently found in the image collection and thus not informative.

The "Saving in Bits" obtained by a region R: According to Shannon, the number of bits required to 'code' a random variable x is $-\log p(x)$. Taking the $\log$ of Eq. (3) and using the quantized codebook $\hat{D}$ yields (disregarding global constants):

$$\log \frac{p(R|Ref)}{p(R|H_0)} = \sum_i |\Delta d_i(H_0)|^2 - |\Delta d_i(Ref)|.^2 \tag{4}$$

This is the number of bits saved by generating R from Ref, as opposed to generating it 'from scratch' at random (using H_0)—"savings in bits"$(R|Ref)$. Therefore, if a region R is composed of statistically significant descriptors (with high $\Delta d_i(H_0)$), and has a good match in Ref (low $\Delta d_i(Ref)$), then R will obtain very high 'savings in bits' (because the difference between the two errors is large). In contrast, a large recurring uniform region or a long edge will hardly yield any 'savings in bits', since both errors $\Delta d_i(H_0)$ and $\Delta d_i(Ref)$ will be low, resulting in a small difference.

So far we discussed a single region R. When the query image Q is composed of multiple (non-overlapping) regions $R_1, \ldots, R_r$ from Ref, we approximate the total 'savings in bits' of Q given Ref, by summing up the 'savings in bits' of the individual regions. This forms the affinity between Q and Ref:

$$\text{affinity(Q, Ref)} = savings(Q|Ref) = \sum_{i=1}^{r} savings(R_i|Ref). \tag{5}$$

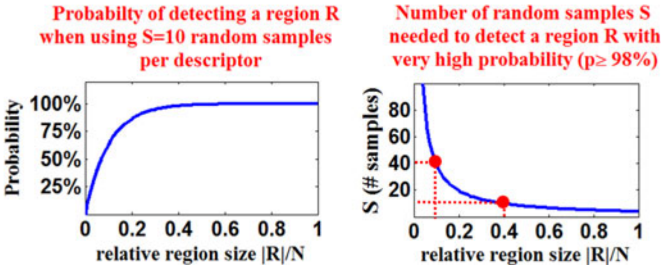


Fig. 6. *Illustration of Claim 1.* Analytic graphs for (a) the probability detection p of a region R (for a given number of random samples S), (b) the required number of random samples S required for the region detection (for a given probability detection p). Those are analyzed as a function of the relative size of the region R/N. For example, to detect a shared region of relative size 10 percent with probability $p \geq 98\%$ requires $S = 40$ random samples.

4 Randomized Detection of Shared Regions

We propose a randomized search algorithm for detecting large unknown (irregularly shaped) regions which are shared across images. We represent an image by computing patches (or descriptors) at some dense image grid and consider a region in the image to be an ordered ensemble of patches (or descriptors) along with their relative positions within the region. Inspired by "PatchMatch" [2], we exploit the power of random sampling and the coherence of neighboring pixels (descriptors) to quickly propagate information. However, unlike PatchMatch, we search for 'matching *regions*', as opposed to matching patches/descriptors.

Although efficient, the theoretical complexity of PatchMatch is $O(N \log N)$, where N is the number of densely sampled patches or descriptors. This is because it spends most of its time seeking good matches for the spurious and isolated descriptors. However, in our application, we wish to find only *large* matching regions across images (and ignore the small spurious distracting ones). We show that this can be done in linear time $O(N)$. In fact, *the larger the region, the faster it will be found* (with fewer random samples, and with higher probability). In other words, and quite surprisingly: *region-matching is 'easier' than descriptor-matching!* We refer to this as the "wisdom of crowds of pixels".

The Region Growing (Detection) Algorithm: Let R be a shared region (of unknown shape, size, or position) between images I_1 and I_2, each consisting of N densely computed descriptors. Let R_1 and R_2 denote its instances in I_1 and I_2, respectively. The goal is to find for each descriptor $d_1 \in R_1$ its matching descriptor $d_2 \in R_2$.

(i) Sampling: Each descriptor $d \in I_1$ randomly samples S locations in I_2, and chooses the best one. The complexity of this step is $O(SN)$. The chance that one of the S samples of a single descriptor d will accidently fall on its correct match in I_2 is very small. However, the chance that *at least one* of the samples of all the descriptors from R_1 will accidently fall on its correct match in R_2 is very high if R is large enough (see Claim 1 and Fig. 6a.). Therefore, once a descriptor from R_1 finds a good match in R_2, it propagates this information to all the other descriptors in R_1—as described in the next step.

(ii) Repeat several times:

a. *Propagation:* Each descriptor chooses between its best match so far, and the match proposed by its spatial neighbors (with appropriate shift)—whichever has the lower matching error. For example, each descriptor suggests to its neighbor on the right the location which is just on the right from the location of its own match. The propagation through the entire image is achieved quickly via two image sweeps (once from top down, and once from bottom up). The complexity of this step is $O(N)$.

b. *Local search:* Each descriptor $d \in I_1$ *randomly* samples η (typically a small number) locations in a small neighborhood around its current best match so far, and checks if one of the new locations improves its best match. This allows the regions to grow in a non-rigid fashion. The complexity of this step is $O(N)$.

Complexity: The overall runtime is $O(SN)$. In Claim 1, we prove that for large enough regions, the required S is a small constant, yielding a overall linear complexity, $O(N)$.

Next we provide a series of claims (Claims 1-4) which quantify the number of random samples per descriptor S required to detect shared regions R across images (*pairs* and *collections* of images) at high probability. This is analyzed as a function of the relative region size in the image $|R|/N$, the desired detection probability p, and the number of images M in the image collection. *All proofs appear in Section 8.*

4.1 Shared Regions between Two Images

For the purpose of theoretical analysis only, we assume that all images consist of N densely sampled descriptors, and that the transformation between shared regions is a pure rigid shift (disregarding the local non-rigid search step in the region growing algorithm).

Claim 1 (A single shared region between two images). *Let R be a region which is shared by two images, I_1 and I_2 of size N. Then:*

(a) Using S random samples per descriptor, guarantees to detect the region R with probability $p \geq (1 - e^{-S\,|R|/N})$.

(b) To guarantee the detection of the region R with probability $p \geq (1 - \delta)$, requires $S = \frac{N}{|R|}\log(\frac{1}{\delta})$ samples per descriptor.

Proof. See Section 8. □

Implication: Figs. 6a and 6b graphically illustrates the terms in claim 1.a, b. For example, to detect a shared region of relative size 10 percent with probability $p \geq 98\%$ requires $S = 40$. Thus, an complexity of $O(40N)$ – linear in N.

Claim 2 (Multiple shared regions between two images). *Let $R_1, \ldots, R_L$ be L shared non overlapping regions between two images I_1 and I_2. If $|R_1| + |R_2| + \cdots |R_L| = |R|$, then it is guaranteed to detect at least one of the regions R_i with the same probability $p \geq (1 - \delta)$ and using the same number of random samples per descriptor S as in the case of a single shared region of size $|R|$.*

Proof. See Section 8. □

Implication: Consider the case where at least 40 percent of one image can be composed using several (smaller) pieces of another image. Then according to Fig. 6b, when using only $S = 10$, we are guaranteed to detect at least one of the shared regions with probability $p \geq 98\%$. Moreover, as shown in Fig. 6a, this region will most likely be one of the

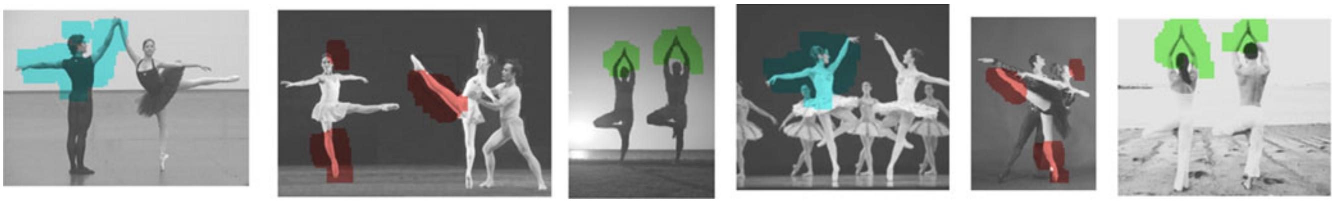


Fig. 7. *Examples of shared regions detected by our algorithm.* Detected connecting regions across images are marked by the same color.

largest regions in the composition, since small regions have very low detection probability with $S = 10$ (e.g., a region of size 1 percent has only 10 percent chance of detection).

4.2 Shared Regions within an Image Collection

We now consider the case of detecting a shared region between a query image and at least one other image in a large collection of M images. For simplicity, let us first examine the case where all the images in the collection are "partially similar" to the query image. We say that two images are *"partial similar"* if they share at least one large region (say, at least 10 percent of the image size). The shared regions R_i between the query image and each image I_i in the collection may be possibly different ($R_i \neq R_j$).

Claim 3 (Shared regions within an image collection). *Let I_0 be a query image, and let $I_1 \ldots, I_M$ be images of size N which are "partially similar" to I_0. Let $R_1, \ldots, R_M$ be regions of size $|R_i| \geq \alpha N$ such that R_i is shared by I_0 and I_i (the regions R_i may overlap in I_0). Using $S = \frac{1}{\alpha}\log(\frac{1}{\delta})$ samples per descriptor in I_0, distributed randomly across $I_1, .., I_M$, guarantees with probability $p \geq (1-\delta)$ to detect at least one of the regions R_i.*

Proof. See Section 8 . □

Implication: The above claim entails that using the *same number of samples S* per descriptor as in the case of two images, but now *scattered randomly across the entire image collection*, we are still guaranteed to detect at least one of the shared regions with high probability. This is *regardless of the number of images M in the collection!* For example, if the regions are at least 10 percent of the image size (i.e., $\alpha = 0.1$), then $S = 40$ random samples per descriptor in I_0, distributed randomly across $I_1, \ldots, I_M$, suffice to detect at least one region R_i with probability 98 percent.

In practice, however, only a portion of the images in the collection are "partially similar" to I_0, and those are 'buried' among many other non-similar images. Let the number of "partially similar" images be $\frac{M}{C}$, where $\frac{1}{C}$ is their portion in the collection. It is easy to show that in this case we need to use C times more samples, than in the case where all the images in the collection were "partially similar", in order to find at least one shared region between I_0 and one of the "partially similar" images. For example, assuming there are four clusters and assuming all images in the cluster are "partially similar" (which is not always the case) then $C = 4$. Note that typically, the number of clusters is much smaller than the number of images, i.e., $C \ll M$.

Our clustering algorithm (Section 5) applies this region search process *simultaneously* to all images against each other. Each descriptor in each image randomly samples a total of S descriptors from the entire collection. We wish to guarantee that in a single 'simultaneous iteration', *almost all the images* in the collection will generate at least one strong connection (large shared region) with at least one other image in the collection.

Claim 4 (Multiple images versus Multiple images). *Assume: (i) Each image in the collection is "partially similar" to at least $\frac{M}{C}$ images. (ii) The shared regions are at least 10 percent of the image size. (iii) We use $S = 40C$ random samples per descriptor (sampled in the entire collection). Then at least 95 percent of the images in the collection are guaranteed to generate at least one strong connection (find at least one large shared region) with at least one other image in the collection with* extremely *high probability. This probability rapidly grows with the number of images, and is practically 100 percent for $M \geq 500$.*

Proof. See Section 8. □

Implication: Claim 4 implies that after one iteration, 95 percent of the images will generate strong connections to other images in the collection. Very few iterations thus suffice to guarantee that *all* images have at least one such connection. Figs. 7 and 10 show examples of such connecting regions detected by our algorithm in the Ballet/Yoga data set and the PASCAL data set.

5 The Collaborative Image Clustering Algorithm

So far, each image independently detected its own shared regions within the collection, using only its 'internal wisdom' (the "wisdom of crowds of pixels"). We next show how collaboration between images can significantly improve this process. Each image can further make *'scholarly suggestions'* to other images where they should sample and search within the collection. For example, looking at Fig. 1, image I_0 has strong affinity to images $I_1, \ldots, I_4$. Therefore, in the next iteration, I_0 can 'encourage' $I_1, \ldots, I_4$ to search for matching regions in each other. Thus, e.g., I_3 will be 'encouraged' to sample more in I_1 in the next iteration. The guided sampling process via multi-image collaboration significantly speeds up the process, reducing the required number of random samples and iterations. Within few iterations, strong connections are generated among images belonging to the same cluster. We refer to this as the "wisdom of crowds of images".

In a nut-shell, our algorithm starts with uniform random sampling across the entire collection. The connections created between images (via detected shared regions) induce affinities between images (see Section 3). At each iteration, the sampling density distribution of each image is re-estimated according to 'suggestions' made by other images (guiding it where to sample in the next iteration).

This results in a "guided" random walk through the image collection. Finally, the resulting affinities (from all iterations) are fed to the N-Cut algorithm [17], to obtain the desired clusters. A diagram describing our collaborative image clustering algorithm is shown in Fig. 3.

Note that N-Cut algorithm (and other graph partitioning algorithms) implicitly rely on two assumptions: (i) that there are enough strong affinities within each cluster, and (ii) that the affinity matrix is relatively sparse (with the hope that there are not too many connections across clusters). The sparsity assumption is important both for computational reasons, as well as to guarantee the quality of the clustering. This is often obtained by sparsifying the affinity matrix (e.g., by keeping only the top $10\log_{10}M$ values in each row [9]). The advantage of our algorithm is that it implicitly achieves both conditions via the 'scholarly' multi-image collaborative search. The 'suggestions' made by images to each other quickly generate (within a few iterations) strong intra-cluster connections, and very few inter-cluster connections.

```
The algorithm;
Initiate affinity matrix to zero: $A \equiv 0$ ;
Initiate sampling distributions to uniform: $P = U$ ;
for iteration $t = 1, \ldots, T$ do
    for image $i = 1, \ldots, M$ do
        1. Randomly sample according to distribution $P_i$;
        2. 'Grow' regions – update the mapping $F_i$ (Sec. IV);
        3. Update row $i$ of "Bit-Saving" matrix $B$ using $F_i$;
    end
    Update affinity matrix: $A = \max(A, B)$;
    Update $P$ (using the "wisdom of crowds of images");
end
Impose symmetry on $A$ by $A = max(A, A^T)$;
Apply N-cut [17] on $A$ to obtain K image clusters (we assume
K is known);
```

5.1 Algorithmic Details

Notations:

- F_i ($i = 1, \ldots, M$) denotes the mapping between the descriptors of image I_i to their matching descriptors in the image collection. It contains for each descriptor the index of the image of its match (different descriptors can map to different images) and its spatial displacement in that image. The induced mapping is constrained by the region growing algorithm of Section 4 and therefore, it tends to be piece-wise smooth in areas where matching regions were detected and quite chaotic elsewhere.
- A denotes the affinity matrix of the image collection. Our algorithm constructs this matrix using the information obtained by composing images from each other.
- B denotes the "Bit-Saving" matrix at each iteration. The value B_{ij} is the "Saving in Bits" contributed by image I_j to the composition of image I_i, at a specific iteration.
- P denotes the "sampling distribution" matrix at each iteration. P_{ij} is the prior probability of a descriptor in image I_i to randomly sample descriptors in image I_j when searching for a new candidate match. P_i (the ith row of P) determines how image I_i will distribute its samples across all other images in the collection in the next iteration.
- U denotes the matrix corresponding to a *uniform* sampling distribution across images. (i.e., all its entries equal $\frac{1}{M-1}$, except for zeros on the diagonal).

Explanations:

- *Randomly sample according to distribution P_i:* Each descriptor in image I_i samples descriptors at S random locations in the image collection. Each of the S samples is sampled in 2 steps: (i) an image index $j = 1, \ldots, M$ is sampled according to distribution P_i (ii) a candidate location in I_j is sampled uniformly. The candidate match, among the S samples, with the best matching score is set to be the current best match. We use $S = 40$ random samples per descriptor, which is the number suggested by our theoretical analysis in Section 4 for guaranteeing region detection between images which share regions of total size of at least 10 percent of the image size.
- *Update row i of "Bit-Saving" matrix B using F_i:* The detected ("grown") regions need *not* be explicitly segmented in order to compute the "Savings-in-bits". Instead, for each image I_i, we first disregard all the descriptors which are spuriously mapped by F_i (i.e., descriptors that are mapped to different images and/or to different image locations than their surrounding descriptors). This is done as follows: for each descriptor d, we count how many descriptors d_i within its surrounding neighborhood of radius 5 grid points (with grid distance of two pixels between adjacent descriptors), are mapped to the same relative image location (up to small deviations of three grid points). A descriptor with less than 15 consistently mapped local descriptors is considered "spuriously mapped".

  Let χ_i denote all remaining descriptors in image I_i (the descriptors mapped consistently with their surrounding descriptors). These descriptors are part of larger regions grown in I_i. B_{ij} is estimated using the individual pixel-wise "Savings-in-bits" induced by the mapping F_i, summed over all the descriptors in χ_i which are mapped to image I_j:

  $$B_{ij} = \sum_{k\in\chi, F_i(k)\mapsto I_j} \left|\Delta d_k(H_0)\right|^2 - \left|\Delta d_k(I_j)\right|.^2$$

  $\Delta d_k(I_j)$ is the error between descriptor d_k in I_i and its match in I_j (induced by F_i).
- *Update P (using the "wisdom of crowds of images")*: For the development of the update rule of P, let us consider a Markov chain (a "Random Walk") on the graph whose nodes are the images in the collection. We set the transition probability matrix between nodes (images) $\hat{B}$ to be equal to the "Bit-Savings" matrix B, after normalizing each row to 1. $\hat{B}_{ij}$ reflects the *relative* contribution of each image to the current composition of image I_i. If we start from state i

TABLE 1
Performance Evaluation on Benchmark Data Sets

Benchmark	# of classes	Measure	[12]	[13]	[11]	Our Method (full search range) HOG	Our Method (restricted search range) HOG	Our Method (restricted search range) LSS	Our Method (restricted search range) HOG+LSS	Relative Improvement over current state-of-the-art
Caltech	4	F-measure	-	-	0.87	**0.905**	**0.935**	**0.895**	**0.975**	+12.1%
Caltech	10	F-measure	-	-	0.68	**0.825**	**0.9**	**0.81**	**0.935**	+37.5%
Caltech	7	Purity	-	-	78.9	**89.6**	**90.9**	**85.5**	**91.6**	+16.1%
Caltech	20	Purity	-	-	65.6	**75.1**	**80.7**	**81.5**	**86.3**	+31.5%
ETHZ	5	Purity	76.5	87.3	-	**89.8**	**94.9**	**94.1**	**96.5**	+10.5%

Our results show significant improvement over state-of-the-art methods. For each data set (each row in the table), our result is compared against the state-of-the-art method to-date on that data set. The last column shows the relative improvement obtained by our method.

(image I_i) and go one step in the graph, we will get a distribution equal to $\hat{B}_i$ (the image own "wisdom"). Similarly, if we go two steps we get a distribution $\hat{B}_i^2$ (the neighbors' "wisdom"). Using these facts, we update the sampling distributions in P as follows:

$$P = \frac{1}{3}(\hat{B} + \hat{B}^2 + U).$$

The first term, $\hat{B}$, encourages each image to keep sampling in those images where it already found initial good regions. The second term, $\hat{B}^2$, contains the 'scholarly' suggestions that images make to each other. For example, if image I_i found a good region in image I_j (high $\hat{B}_{ij}$), and image I_j found a good region in image I_k (high $\hat{B}_{jk}$), then $\hat{B}_{ik}^2$ will be high, suggesting that I_i should sample more densely in image I_k in the next iteration. The third term, U, promotes searching uniformly in the collection, to avoid getting 'stuck' in local minima.

5.2 Incorporating Scale Invariance

In order to handle scale invariance, we generate from each image a cascade of multi-scale images, with relative scales $\{(\sqrt{0.8})^l\}_{l=0}^5$ $-$images of size $\{1, 0.8, 0.64, 0.51, 0.41, 0.33\}$ relative to the original image size (in each dimension). The region detection algorithm is applied to the entire multi-scale collection of images, allowing region growing also across different scales between images. The multi-scale cascade of images originating from the same input image are associated with the same entity in the affinity matrix A.

5.3 Complexity (Time and Memory)

All matrix computations and updates ($max(A, B)$, $\hat{B}^2$, update P, etc.) are efficient, both in terms of memory and computation, since the matrix B is *sparse*. Its only non-zero entries correspond to the image connections generated in the current iteration.

We set the number of iterations in our algorithm to be $T = 10 \log_{10} M$, which is the recommended sparsity of the affinity matrix by [9]. Note that our algorithm directly estimates a good set of sparse affinities (as opposed to computing a full affinity matrix and then sparsifying it). T is typically a small number (e.g., for $M = 1{,}000$ images $T = 30$; for $M = 10{,}000$ images $T = 40$). The complexity of each iteration is $O(NM)$ (see Section 4). Therefore, the overall complexity of our clustering algorithm is $O(NMlog_{10}(M))$—almost linear in the size of the image collection (NM).

6 Experimental Results

We tested our algorithm on various data sets, ranging from benchmark evaluation data sets (Caltech, ETHZ), on which we compared results to others, to more difficult data sets (PASCAL), on which to-date *no* results were reported for *purely unsupervised* category discovery. Finally, we also show the power of our algorithm on *tiny* data sets. Tiny data sets are challenging for unsupervised learning, since there are very few images to 'learn' from.

In all the experiments, we set the number of iterations to be $T = 10 \log_{10} M$, where M is the number of images we wish to cluster. After performing the T iterations, we kept the highest $10 \log_{10} M$ values in each row of the affinity matrix. We experimented with two types of densely-sampled descriptors: $15X15$ HOG descriptors [6] and $25X25$ local self-similarity descriptors [16]. These were computed densely—every two pixels—with high overlaps.

We also compared the performance of each of these descriptors with a combination of the affinity matrices produced by the two descriptor types. The combination was done simply by first normalizing the row of each matrix to 1 and then summing the matrices. Finally, we experimented with restricting the spatial search range of each descriptor to no more than 25 percent of the image size (around each descriptor). This restriction enforces a weak prior on the rough geometric arrangement within the image (similarly to [4] and [10]).

6.1 Experiments on Benchmark Evaluation Data Sets

We used existing benchmark data sets (Caltech, ETHZ-shape) to compare results against [11], [12], [13] using their experimental setting and measures. Results are reported in Table 1. The four data sets generated by [11] consist of difficult classes from Caltech-101 with non-rigid objects and cluttery background (such as leopards and hedgehogs), from four classes (189 images) up to 20 classes (1,230 images). Example images are shown in Fig. 8. The ETHZ-shape data sets consists of five classes: Applelogos, Bottles, Giraffes, Mugs and Swans. For the ETHZ data set, we followed the experimental setting of [12] (which crops the images so that the objects are 25 percent of the image size). For both Benchmarks, our algorithm obtains state-of-the-art results (see Table 1). Note that for the case of 10 and 20 Caltech classes, our algorithm obtains more than 30 percent relative improvement over current state-of-the-art.

Notice that restricting the spatial search range of descriptors improves the results (see Table 1). Furthermore,

Fig. 8. *Example images from the Caltech Subsets.* Images include non-rigid objects (e.g., Leopards) and significant background clutter. Our method provides a significant leap in results on these kinds of data sets (see Table 1).

combining between the affinity matrices produced by the HOG and the LSS descriptors induces a further improvement compared to using each of the descriptors separately.

6.2 Experiments on a Subset of Pascal-VOC 2010 Data Set

The Pascal data set is a very challenging data set, due to the large variability in object scale, appearance, and due to the large amount of distracting background clutter. Unsupervised category discovery is a much more difficult and ill-posed problem than classification, therefore to-date, *no* results were reported on PASCAL for *purely unsupervised* category discovery. We make a first such attempt, restricting ourselves at this point to four categories: Car, Bicycle, Horse and Chair. We generated a subset of 100 images per category restricting ourselves to images labeled "side view" and removing images which simultaneously contain objects from 2 or more of the above categories (otherwise the clustering problem is not well-defined). The PASCAL-VOC subset used in our experiments can be found in *www.wisdom.weizmann.ac.il/~vision/ClusterByComposition.html*. Fig. 9 show a few example images from the Car and Horse categories, demonstrating how challenging this data set is (huge variability in scale, lots of foreground and background clutter, etc.).

We tested our algorithm on this subset, clustering it to four clusters, and obtained a mean purity of 68.5 percent. In this case, restricting the search range did not yield better results since the object locations are scattered across the entire image. Fig. 11 shows example images which were clustered correctly along with example images which were mis-clustered. Notice that mis-clustered images can be conceptually 'confusing', like a horse with a carriage (which is confused for a car or bicycle due to the wheels). Fig. 10 show examples of shared regions detected by our algorithm in this PASCAL subset.

To our best knowledge, nobody has run totally unsupervised clustering on this data set before, so we had no one to compare to. However, we ran a baseline experiment with spatial pyramid match kernel (SPM) [10] and N-Cut and our algorithm gave 20 percent relative improvement over this baseline.

To understand the sources for confusion in our clustering results, we computed a confusion matrix of the generated

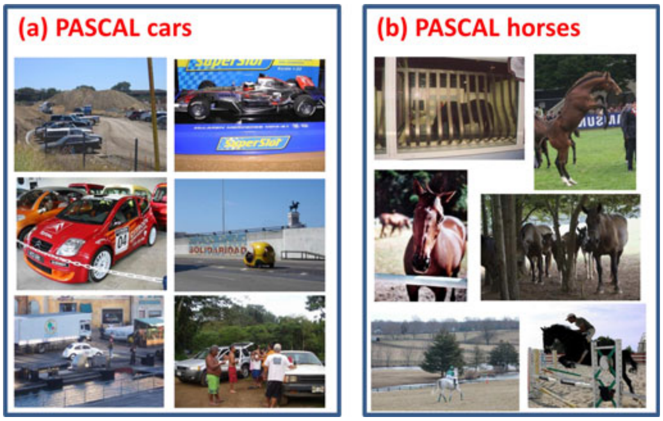


Fig. 9. *Example images from:* (a) PASCAL car category, (b) PASCAL horse category. The objects in each category can be tiny or huge, have non-rigidities and may have lots of clutter and occlusions.

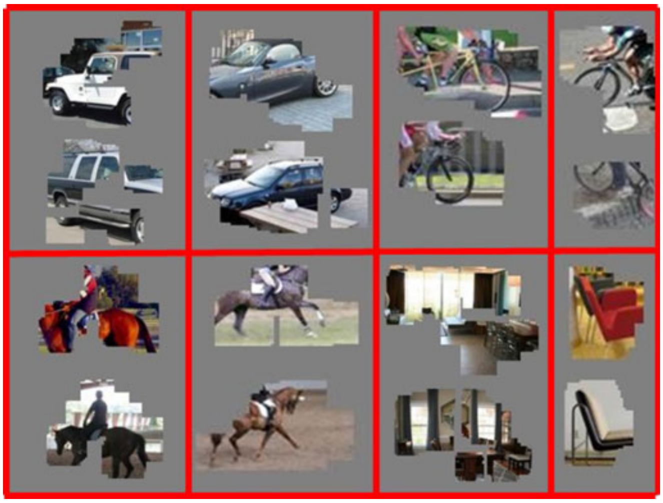


Fig. 10. *Examples of shared regions detected by our algorithm for the PASCAL data set.* Each box contains a shared region. Note that the regions do not necessarily appear in their true scale and may capture only a small portion of the image which they came from (less than 10 percent).

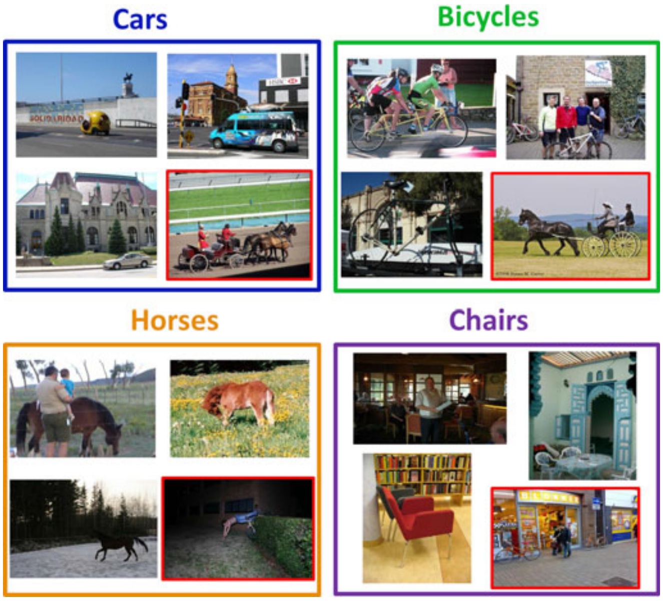


Fig. 11. *Clustering of PASCAL data set.* Four categories (Car, Bicycle, Horse, Chair), 100 images per category. Examples of correct/incorrect clustering (mis-clustered images marked in red).

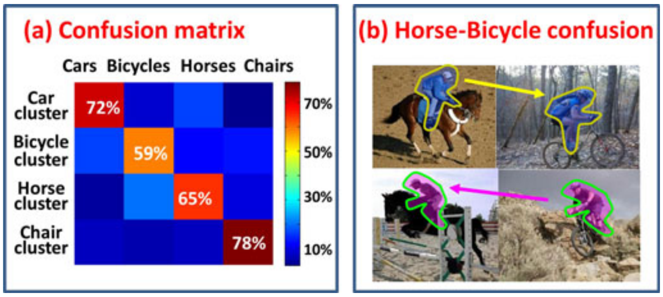


Fig. 12. *Typical confusions in the PASCAL data set.* (a) Confusion matrix of the clusters generated by our algorithm. (b) Illustration why horse and bicycle images tend to be confused—they share an almost identical pose of a human rider, which is a large and non trivial region (inducing strong affinity between those images).

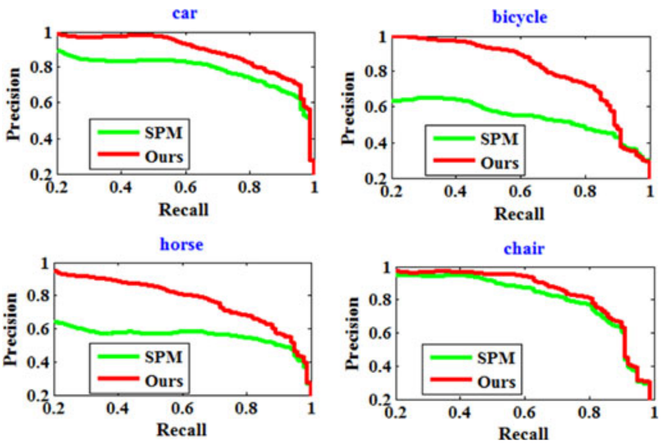


Fig. 13. *Precision-Recall of our affinity matrix for the PASCAL data set.* We use our affinity matrix to generate precision-recall curves and compare them to the curves obtained by the spatial pyramid match affinities.

clusters (see Fig. 12a). The different values in each row represent the distribution of images within that cluster. For example, the car cluster contains 72 percent cars, 8 percent bicycles, 17 percent horses and 3 percent chairs. The identity of each cluster was determined by the category which got the most images in the cluster. Ideally, we would like the values on the diagonal to be 100 percent and the off-diagonal values to be 0 percent.

As can be seen, the car and chair cluster have relatively good purity. However, there seems to be strong confusion between Horses and Bicycles. This is surprising since horses and bicycles have quite different appearances. However, the reason for this confusion is the very unique yet almost identical pose of the human riding these two types of objects (see examples in Fig. 12b). Such a similar non-trivial pose of the rider induces strong affinities between those images, resulting in confusion between those two categories.

Precision-Recall of our affinity matrix: Finally, to measure the quality of the affinity matrix generated by our unsupervised algorithm, we conduct the following experiment. For each image we compute its resulting average affinity to the images within each of the classes (using the ground truth labels of the other images). We define the following classification confidence for each image I_i per class c : $score(i,c) = \frac{\Sigma_{j\in c, j\neq i} A(i,j)}{\Sigma_{j\neq i} A(i,j)}$, where A denotes our affinity matrix. Namely, $score(i,c)$ is the affinity of I_i to class c divided by the total affinity of I_i to all other images. We then compute precision-recall curves using the scores of each of the classes. For a given class and a given score threshold, the precision measures the percentage of class images among all the images which passed the threshold, and the recall counts the percentage of these class images with respect to the total number of class images. We then compared our precision-recall curves to that obtained using the affinities of the spatial pyramid match kernel.

As can be seen in Fig. 13, the precision obtained by our method grows dramatically as the recall decreases, obtaining for all classes more than 90 percent precision at low recall values (the highest ranked images). Our precision-recall is consistently better than that of SPM for all classes, with a very significant gap at the bicycle and horse classes. The reason for the large gap at the horse and bicycle classes, is that many horse and bicycle images have similar scenes (e.g., forest or field). Therefore, since SPM captures mostly the global geometric arrangement within each image, it got confused by the two classes. We, on the other hand, are less sensitive to such scene similarity.

Overall, we obtain an average precision (averaged over all classes) of 79 percent for the top 75 percent ranked images in each class, 93 percent average precision for the top 50 percent ranked images and 96 percent average precision for the top 25 percent ranked images. SPM, on the other hand, obtained an average precision of 61, 71 and 77 percent for the top 75, 50 and 25 percent ranked images.

6.3 Experiments on *Tiny* Data Sets

Existing methods for unsupervised category discovery require a *large* number of images per category (especially for complex non-rigid objects), in order to 'learn' shared 'cluster models'. To further show the power of our algorithm, we generated two *tiny* data sets: the Ballet-Yoga data set (Fig. 2) and the Animal data set (Fig. 4). These tiny data sets are *very challenging* for unsupervised category discovery methods, because of their *large* variability in appearance versus their *small* number of images.

Our algorithm obtains excellent clustering results for both data sets, even though each category contains different poses, occlusions, foreground clutter (e.g., different clothes), and confusing background clutter (e.g., in the animal data set). The success of our algorithm can be understood from Figs. 5 and 7: Fig. 5 shows that the descriptors with the highest statistical significance are indeed the most informative ones in each category (e.g., the Monkey's face and hands, the Elk's horns, etc.). Fig. 7 shows that meaningful shared regions were detected between images of the same category.

6.4 Convergence of our Algorithm

In order to test the convergence of our algorithm, we performed the following experiment on the Caltech-20 classes benchmark. We ran our algorithm for 120 iterations and after each iteration we applied N-cut on the affinity matrix computed so far and computed the mean purity of the resulting clusters. Before applying N-cut we always kept the highest $10 \log_{10} M$ values in each row of the affinity matrix. Results are shown in Fig. 14.

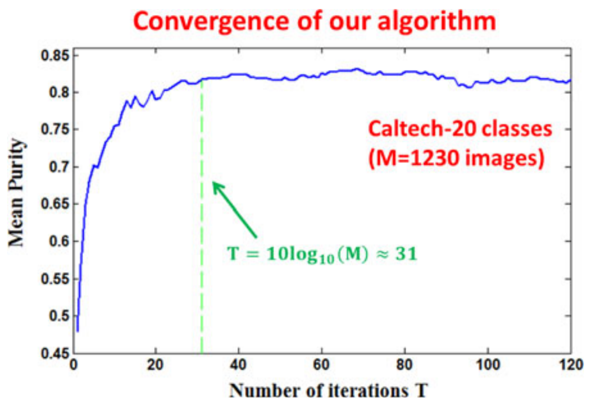


Fig. 14. *Testing the convergence of our algorithm.* Our algorithm converges to its best performance after $T = 10\log_{10} M \approx 31$ iterations. This implies that our algorithm indeed generates a sparse set of meaningful affinities very efficiently, without having to compute all the pairwise affinities.

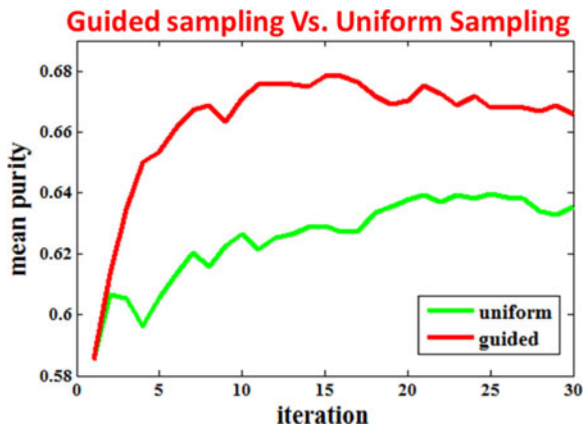


Fig. 15. *Guided sampling versus uniform sampling for the PASCAL data set.* The guided sampling obtains a relative improvement of 6 percent over uniform sampling, as well as converges significantly faster.

Note that after $10\log_{10} M \approx 31$ iterations, our algorithm has almost converged to its best performance—i.e., examining new connections in the collection will not improve the performance. These results show that our algorithm is indeed able to generate a sparse set of meaningful affinities very efficiently, without having to compute all the pairwise affinities. Moreover, we can see that even after a single iteration, our algorithm is able to obtain mean purity of almost 50 percent, meaning that most images already found good connections in the collection (this empirically verifies claim 4).

6.5 Analyzing Different Components of Our Algorithm

To better understand the behavior of our algorithm we conducted the following analysis:

1. *Sparse guided sampling versus exhaustively computing all pairwise “affinities by composition”:* We compared our results to those obtained by computing all the pairwise “affinities by composition”, followed by sparsifying the affinity matrix and then applying N-cuts (See Table 2). For each pair of images, we used our region detection algorithm (Section 4) with the same number of samples that was used in each iteration of our sparse multi-image composition algorithm ($S = 40$). Table 2 shows how efficient our sparse multi-image composition algorithm is in reducing the time complexity as the number of images increases. For example, in PASCAL we had 2,400 images (400 images at six scales) and we used 15 iterations—yielding a time saving by a factor of $2,400/15 = 160$. Moreover, our algorithm obtains slightly better results in clustering performance. This shows that we do not lose any significant information by not computing all the pairwise affinities, and even gain a small improvement. The reason for this is perhaps, as suggested in Section 5, that we directly obtain a sparse set of meaningful affinities, as opposed to ‘blindly’ sparsifying the affinity matrix after computing the full matrix. The sparse guided sampling does not get distracted by spurious feature matches, whereas the dense (exhaustive) one does.

TABLE 2
Sparse Guided Sampling versus Exhaustively Computing All Pairwise “Affinities by Composition”

Dataset	Cal-4	ETHZ	Cal-7	Cal-10	Cal-20	PASCAL
# of images	189	255	441	489	1230	2400
Clustering improvement	-2%	$+3\%$	$+2.5\%$	$+6.5\%$	$+0.5\%$	$+2\%$
Runtime (minutes)	15	21	40	47	135	126
Time saving	$\times 8$	$\times 10$	$\times 17$	$\times 18$	$\times 40$	$\times 160$

2. *Guided sampling versus uniform sampling:* We further compared our results to those obtained by using a sparse uniform sampling (instead of a sparse *guided* sampling). For all data sets besides PASCAL, moving to uniform sampling cause only a slight decrease in performance. The reason for this is that our “affinities by composition” are discriminative enough between different classes. Therefore, even if we sample at the wrong (non-class) images, our affinities will compensate for this and give these images lower values as compared to the values obtained when sampling at the class images. However, on the PASCAL data set, our guided sampling yielded an improvement over uniform sampling (as can be seen in Fig. 15). The guided sampling obtains a relative improvement of 6 percent as well as converges significantly faster.

7 Conclusion

In this paper we suggest a new approach to image clustering—“Clustering-by-Composition”. Our approach is based on composing an image, like a simple ‘puzzle’, from large non-trivial pieces of other images. Similar images will be ‘easy’ to compose from each other and non-similar images will be a lot ‘harder’ to compose from each other. We show that using this approach we can capture complex visual similarity between images. This enables us to discover clusters of images which belong to very challenging image categories, that do not have any simple model that is common to them (such as common shape, segments, etc.).

Our contributions are of three folds:

i. We define a “good” region which induces high affinity between images as one that is rare (has a low chance of occurring at random) yet shared with another image.

ii. We demonstrate how using the quantization error of descriptors with a respect to a codebook, we can estimate very efficiently how rare a region is. This is very different from how people commonly use codebooks ("bags of descriptors") in recognition and classification. Usually a codebook is used in order to represent the image. Here we use the codebook to detect the descriptors that are not represented well in the codebook and those are the most informative for us.

iii. We suggest a randomized search process, with good theoretical guarantees, which enables the efficient detection of shared regions across images. We further incorporate the "wisdom of crowds of images" into this randomized search to obtain a collaborative clustering algorithm. This algorithm generates a sparse set of meaningful affinities at time which is almost linear in the size of the collection, without having to compute all the pairwise affinities. This sparsity is essential for good clustering.

Finally, we obtain state-of-the-art results on benchmark data sets and got very encouraging results on new challenging data sets. These include data sets with very few images (where a 'cluster model' cannot be 'learned' by current methods), and a subset of the challenging PASCAL VOC data set.

8 Proofs of Claims

Below we provide the proofs for the claims of Section 4.

Proof of claim 1. Let R_1 and R_2 denote the instances of a region R in I_1 and I_2. In order to detect the entire region R, at least one descriptor $d_1 \in R_1$ has to randomly sample its correct match $d_2 \in R_2$ (following which the entire region will be 'grown' due to the propagation phase of the Region Growing Algorithm described in Section 4). So, the probability of detecting a region is equal to the probability that at least one of the descriptors $d_1 \in R_1$ will randomly sample its correct match $d_2 \in R_2$.

The probability of a single descriptor $d_1 \in R_1$ to randomly fall on its correct match $d_2 \in R_2$ is $\frac{1}{N}$ (where N is the size of the image). Therefore, the probability that it will NOT fall on d_2 is $(1-\frac{1}{N})$. The probability that NONE of its S samples will fall on d_2 is $(1-\frac{1}{N})^S$. Therefore, the probability that NONE of the descriptors in R_1 will randomly fall on their correct match is $q \triangleq (1-\frac{1}{N})^{S|R_1|} = (1-\frac{1}{N})^{S|R|}$. Thus the probability of detecting the shared region R is $p \triangleq (1-q)$. (a) for $N \geq 1$ it holds that $(1-\frac{1}{N})^N \leq e^{-1}$. Implying that $q = (1-\frac{1}{N})^{N\frac{S|R|}{N}} \leq e^{-\frac{S|R|}{N}}$. So $p = (1-q) \geq 1 - e^{-\frac{S|R|}{N}}$. (b) We need to guarantee that $p = (1-q) \geq 1-\delta$, and ask what is minimal number of samples S required. We know from (a) that $p = (1-q) \geq 1 - e^{-\frac{S|R|}{N}}$. So if we require $1 - e^{-\frac{S|R|}{N}} \geq 1-\delta$ we will satisfy the condition. Switching sides we get: $e^{-\frac{S|R|}{N}} \leq \delta$. Applying $\log$ gives us: $-\frac{S|R|}{N} \leq \log(\delta)$. Rearranging the terms: $S \geq \frac{N}{|R|}\log(\frac{1}{\delta})$. □

Proof of claim 2. $R_1, \ldots, R_L$ are non-overlapping regions, so their probabilities of detection are statistically independent of each other. The probability that all of the regions are not detected is therefore equal to the product of the probabilities of each region not being detected: $\prod_{i=1}^{L}(1-\frac{1}{N})^{S|R_i|}$. This is equal to $(1-\frac{1}{N})^{S\Sigma_{i=1}^{L}|R_i|} = (1-\frac{1}{N})^{S|R|} = q$. So the probability of detecting at least on region is equal to $1-q$ which is identical to the term obtained in claim 1.a for the probability of detecting a single shared region with size $|R|$. Similarly, we also get the same term for the required number of samples S as was obtained in claim 1.b. □

Proof of claim 3. We will first develop a term for the probability of not detecting a specific region $R_i (i = 1, \ldots, M)$. The only change from claim 1.a is that the search space is M times larger (since there are M other images instead of only one). So this probability is equal to $(1-\frac{1}{NM})^{S|R_i|}$. If there were no overlaps between the regions, then the probability $\tilde{q}$ that none of the regions are detected (as was shown in claim 2) equals to the product of the probabilities of each region not being detected: $\tilde{q} = \prod_{i=1}^{M}(1-\frac{1}{NM})^{S|R_i|} = (1-\frac{1}{NM})^{S\sum_{i=1}^{M}|R_i|} le (1-\frac{1}{NM})^{SM\alpha N} = ((1-\frac{1}{NM})^{NM})^{S\alpha} \leq e^{-S\alpha}$.

An overlap between the regions will not change this term. This is due to the fact that on the one hand there are fewer descriptors in the union of all the regions, but on the other hand each descriptor has a higher probability of finding a good match at random. It is easy to show that these two terms cancel each other. Therefore, the probability of detecting at least one of the regions is equal to $p = (1-\tilde{q}) \geq (1-e^{-S\alpha})$. Finally, in order to guarantee detection of at least one region with probability $\geq (1-\delta)$ we need to use $S \geq \frac{1}{\alpha}\log(\frac{1}{\delta})$ samples. □

Proof of claim 4. According to claim 3, in order to guarantee with probability $p \geq 98\%$ that an image I_0 will detect at least one region which is at least 10 percent of the size of the image and is shared with another image, we are required to use $S = 40$ random samples per descriptors ($\delta = 0.02$ and $\alpha = 0.1$). This is the required number of samples S when all the M images are "partially similar" to I_0. When only $\frac{M}{C}$ of the images are "partially similar" to I_0, then S must be C times larger, i.e. $S = 40C$ (using similar derivations to those in claim 3).

This, however, was for one specific image I_0. When applying this process simultaneously to all the M images, we would like to check what percent of the images will detect with very high probability at least one shared region with another image. We will regard the event of each image trying to detect a shared region as an independent Bernoulli trial with success probability of $p = 0.98$ (the guaranteed probability of detecting a shared region per trial). We have M images, thus M Bernoulli trials, all with the same success probability p. Therefore, The number of successes, i.e., the number of images which detect a shared region, has a Binomial distribution $Bin(M, p)$. Similarly, the number of failures has also a Binomial distribution $Bin(M, 1-p)$.

When M is several hundreds ($100 \leq M \leq 1{,}000$) and $1-p = 0.02$ is quite small, the resulting product $M(1-p)$ is of an intermediate size (between 2 and 20). It is well known that in these cases, the binomial

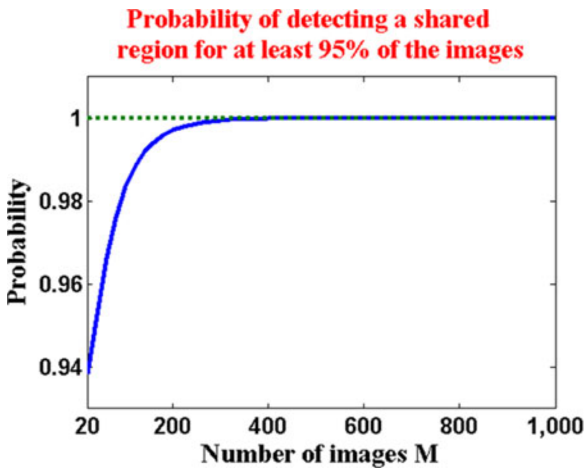


Fig. 16. *Illustration of Claim 4.* Analytic graph for the probability that most images (95 percent) will generate a strong connection (detect a shared region) with another image. This is analyzed as a function of the number of images in the collection M. This probability goes to 1 as M increases.

distribution $Bin(M, 1-p)$ can be approximated well with a Poisson distribution with parameter $\lambda = (1-p)M = 0.02M$. In other words, the probability that k images did not detect a shared region can be approximated by $\frac{e^{-\lambda}(\lambda)^k}{k!}$.

The probability that at least rM of the images detected at least one shared region is equal to the probability that all the images detected a region ($k=0$), or that all but one detected a shared region ($k=1$), ... or that all but $(1-r)M$ detected a shared region. Therefore, it can be approximated by $\sum_{k=0}^{(1-r)M} \frac{e^{-\lambda}(\lambda)^k}{k!}$. Fig. 16 shows this probability for $r = 95\%$ as function of the number of images M. We can see that the probability that at least 95 percent of the images detected a shared region is very high and goes to 1 as M increases (is practically 100 percent for $M \geq 500$). □

Acknowledgments

The authors would like to thank S. Bagon, M. Zontak, D. Glasner and O. Bartal for their helpful comments on the paper. This work was funded in part by the Israeli Science Foundation and the Israeli Ministry of Science.

References

[1] C. Barnes, "Patchmatch: A Fast Randomized Matching Algorithm with Application to Image and Video," PhD thesis, Princeton Univ., 2011.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman, "Patchmatch: A Randomized Correspondence Algorithm for Structural Image Editing," *Proc. ACM SIGGRAPH*, 2009.

[3] O. Boiman and M. Irani, "Similarity by Composition," *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2006.

[4] O. Boiman, E. Shechtman, and M. Irani, "In Defense of Nearest-Neighbor Based Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] O. Chum, M. Perdoch, and J. Matas, "Geometric Minhashing: Finding a (Thick) Needle in a Haystack," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] K. Grauman and T. Darrell, "Unsupervised Learning of Categories from Sets of Partially Matching Image Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[8] C. Gu, J.J. Lim, P. Arbelaez, and J. Malik, "Recognition Using Regions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] G. Kim, C. Faloutsos, and M Hebert, "Unsupervised Modeling of Object Categories Using Link Analysis Techniques," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[11] Y.J. Lee and K. Grauman, "Foreground Focus: Unsupervised Learning from Partially Matching Images," *Int'l J. Computer Vision*, vol. 85, pp. 143-166, 2009.

[12] Y.J. Lee and K. Grauman, "Shape Discovery from Unlabeled Image Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] N. Payet and S. Todorovic, "From a Set of Shapes to Object Discovery," *Proc. 11th European Conf. Computer Vision (ECCV)*, pp. 57-70, 2010.

[14] J. Philbin and A. Zisserman, "Object Mining Using a Matching Graph on Very Large Image Collections," *Proc. Sixth Indian Conf. Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008.

[15] B.C. Russell, A.A. Efros, J. Sivic, W.T. Freeman, and A. Zisserman, "Using Multiple Segmentations to Discover Objects and their Extent in Image Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

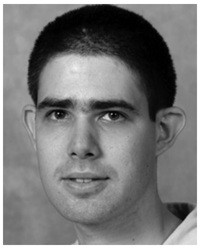
[16] E. Shechtman and M. Irani, "Matching Local Self-Similarities Across Images and Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[17] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[18] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, "Discovering Objects and Their Localization in Images," *Proc. IEEE 10th Int'l Conf. Computer Vision (ICCV)*, 2005.

[19] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. IEEE Ninth Int'l Conf. Computer Vision (ICCV)*, 2003.

[20] T. Tuytelaars, C.H. Lampert, M.B. Blaschko, and W. Buntine, "Unsupervised Object Discovery: A Comparison," *Int'l J. Computer Vision*, vol. 88, pp. 284-302, 2010.



Alon Faktor received the BSc degree in electrical engineering and physics from the Israel Institute of Technology, Technion, in 2009 and the MSc degree in mathematics and computer science from the Weizmann Institute of Science in 2011. He is currently working toward the PhD degree at the Weizmann Institute of Science. His current research focuses on areas of computer vision and video information analysis.

Michal Irani received the BSc degree in mathematics and computer science in 1985 and the MSc and PhD degrees in computer science in 1989 and 1994, respectively, all from the Hebrew University of Jerusalem. From 1993 to 1996, she was a member of the technical staff in the Vision Technologies Laboratory at the David Sarnoff Research Center in Princeton, New Jersey. She joined the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science in 1997, was promoted to an associate professor in 2002, and to a full professor in 2007. Her research interests center around computer vision, image processing, and video information analysis. Her prizes and honors include the David Sarnoff Research Center Technical Achievement Award (1994), the Yigal Allon Three-Year Fellowship for Outstanding Young Scientists (1998), and the Morris L. Levinson Prize in Mathematics (2003). She also received the best paper awards at the European Conference on Computer Vision (ECCV) 2000 and 2002, the honorable mention for the Marr Prize at the IEEE International Conference on Computer Vision (ICCV) 2001 and 2005, and a best poster award at the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2004. She served as an associate editor of the *Transactions on Pattern Analysis and Machine Intelligence* in 1999-2003. She is a member of the IEEE.