Matching Pixels using Co-Occurrence Statistics

Rotal Kat
School of EE
Tel-Aviv University
rotalkat@mail.tau.ac.il

Roy Jevnisek
School of EE
Tel-Aviv University
jernisek@post.tau.ac.il

Shai Avidan
School of EE
Tel-Aviv University
avidan@eng.tau.ac.il

Abstract

We propose a new error measure for matching pixels that is based on co-occurrence statistics. The measure relies on a co-occurrence matrix that counts the number of times pairs of pixel values co-occur within a window. The error incurred by matching a pair of pixels is inversely proportional to the probability that their values co-occur together, and not their color difference. This measure also works with features other than color, e.g. deep features. We show that this improves the state-of-the-art performance of template matching on standard benchmarks.

We then propose an embedding scheme that maps the input image to an embedded image such that the Euclidean distance between pixel values in the embedded space resembles the co-occurrence statistics in the original space. This lets us run existing vision algorithms on the embedded images and enjoy the power of co-occurrence statistics for free. We demonstrate this on two algorithms, the Lucas-Kanade image registration and the Kernelized Correlation Filter (KCF) tracker. Experiments show that performance of each algorithm improves by about 10%.

1. Introduction

Measuring similarity between pixels is a basic task in computer vision. Stereo matching algorithms, for example, use template matching to measure the similarity of potential matches. Texture synthesis algorithms rely on patch similarity to fill in holes, and tracking algorithms need to match the appearance of the object from one frame to the next.

Let us focus on template matching as a canonical application that relies on a pixel similarity measure. Arguably the most popular measure is the Sum-of-Squared-Differences (SSD) that is based on the Euclidean distance between corresponding pixel values in the template and the candidate window.

But SSD is very sensitive to small deformations. To deal with this problem one often use patch level representations such as SIFT [18], HOG [3], or the first layers of a deep network [26]. These representations use a small neighborhood to collect local statistics that increase the robustness of pixel representation to small misalignment and deformations, at the cost of losing precise pixel localization. The metric used to compare these features often remains the Euclidean metric.

The key contribution of this paper is the introduction of a new similarity measure between pixel values that is based on co-occurrence statistics. Co-occurrence statistics are collected over the entire image plane and measure the probability of a pair of pixel values to co-occur within a small window. We take the cost of matching pixels to be inversely proportional to the probability of their values co-occurring. Why?

Because co-occurrence statistics has long been used to capture texture. Pixel values that co-occur frequently in the image are probably a part of textured region. Therefore, this measure implicitly captures some notion of texture similarity. This has nothing to do with the actual pixel values, only their co-occurrence statistics. In other words, we learn pixel similarity from data instead of imposing the Euclidean distance on it.

Co-occurrence statistics differ from the patch based representations mentioned earlier. Patch based methods collect *local* statistics whereas co-occurrence collects *global* statistics. The two approaches complement each other and we can collect co-occurrence statistics of RGB values, as well as other, more involved features, such as deep features. Experiments show that combining both approaches greatly enhances the performance of template matching on standard template matching benchmarks.

We then propose an embedding scheme that maps the pixel values of the input image to a new space. The embedding maps pixel values that co-occur frequently to nearby points in the embedded space, where proximity is based on the Euclidean distance. There are several reasons for doing that. First, it allows us to run *existing* template matching implementations on the embedded images without any modifications. Second, because existing template matching algorithms achieve sub-pixel accuracy, we get this accuracy for free. The alternative, of achieving sub-pixel accuracy by working directly with co-occurrence statistics, is not straightforward to achieve. Third, working with sub-pixel accuracy lets us extend our template matching algorithm to work with more general transformations that do not fall on integer pixel coordinates (i.e., rotations, 2D affine). On the downside, we find that working in the embedded space degrades the accuracy of template matching, compared to working directly with co-occurrence statistics. Still, working with embedded images yields results that are substantially better than working with SSD.

Finally, there is no need to limit ourselves to template matching. We can run any vision algorithm on the embedded images. We demonstrate this on two algorithms. The Lucas-Kanade (LK) image registration algorithm [19] and the Kernelized Correlation Filter (KCF) tracker [11].

The LK algorithm performs gradient descent, a step that is easy to do in Euclidean space, but not as easy when working with co-occurrence statistics directly. The KCF tracker treats tracking as a binary classification problem and solves it efficiently by working in the frequency domain. Again, it is easy to perform FFT on a Euclidean space but it is not clear how to compute the Fourier transform of a space endowed with a co-occurrence error measure.

These problems go away once we embed the images. Experiments show that both algorithms enjoy a 10% boost in performance just by working on the embedded images, with no modification to the actual algorithms themselves.

To summarize, we introduce a new error measure that is based on co-occurrence statistics. The new measure is robust to misalignment and deformations, fast to compute, and can work with different pixel values such as RGB color or deep features. We then suggest an embedding scheme and show that other vision algorithms can benefit from the co-occurrence error measure. Results of extensive experiments on several data sets demonstrate the potential of our method.

2. Background

We use template matching to demonstrate the power of co-occurrence statistics as a similarity measure. Because template matching is a vast topic, we cover here only the relevant work related to ours. We focus on the simple case of 2D translation, but the principles presented here can be extended to other parametric transformations. For an overview see [23].

Template matching seeks to find a candidate window in a target image that matches a given template. This requires the definition of a similarity measure between the window and the template, such as the Sum-of-Squared-Differences (SSD) or Sum-of-Absolute-Differences (SAD). To deal with illumination changes one might use Normalized Cross-Correlation (NCC) or the more elaborate Generalized Laplacian Distance [8]. To handle noise and outliers one might use robust measures such as M-estimators [1].

When tracking a deformable object it might be better to represent the template as a histogram and use an appropriate similarity measure between histograms [2].

In the medical image literature, the use of information theoretic criteria is very popular. For example, two images of different modality are aligned by maximizing their mutual information [20, 29]. However, it is important to point out that mutual information used in these cases is between the different modalities, whereas we are dealing with images of the same modality. See [24] for a recent survey.

Egnal [7] proposed to use Mutual Information (MI) as a stereo correspondence measure to handle illumination changes. Each patch in the source image is matched to several candidate patches in the target image and the match that maximizes the MI is selected. There is no global information sharing in the process. Each patch is processed independently.

Kim *et al.* [17] later extended the idea to work on a Markov Random Field (MRF). The basic idea is to use Graph-Cuts to find a disparity field that maximizes the MI between the warped source image and the target image, instead of trying to minimize a SSD or SAD error measure. See also the follow up work by Hirschmuller [13]. These works differ from ours because they measure the MI between the entire warped source and target images. We, on the other hand, focus on learning the co-occurrence statistics at the pixel level and from the entire image.

The Lucas-Kanade algorithm was adopted to work with MI by Dowson and Bowden [6] by changing the equations to maximize the MI between the two images instead of minimizing the standard SSD error. As with previous work, they demonstrate their algorithm on images with different modalities or different illumination.

Co-occurrence statistics was first introduced by Haralick *et al.* [10] for the purpose of texture analysis. Recently, co-occurrence data (termed mutual pointwise information) was used for crisp boundary detection by Isola *et al.* [15]. They use Normalized Cuts [25] to find edges. However, instead of using pixel differences when constructing the Affinity matrix, they use co-occurrence statistics. Co-occurrence statistics was also used to extend the Bilateral Filter to deal with texture [16]. The core idea there was to replace the range Gaussian of the bilateral filter with a co-occurrence statistics measure, thus capturing texture properties instead of differences in pixel value.

The idea of using co-occurrence statistics for image matching was also suggested by Hseu *et al.* [14]. However, they only considered the case of gray scale images and the simple case of 2D translation. There is no discussion of color, or patch based features, and no discussion of the embedding idea presented here. The only experiment they present is a synthetic one on the image of *Lena*.

There has also been a considerable amount of work on embedding using co-occurrence data. Globerson *et al.* discuss Euclidean embedding of co-occurring data of different types, such as text and images [9]. This is a generalization of the Stochastic Neighborhood Embedding (SNE) of Hinton and Roweis [12] and its extension tSNE by Van Der Maaten and Hinton [28]. Common to these works is that they attempt to preserve neighborhood structure such that data that co-occur in the original space should co-occur in the embedding space. This is in contrast with our goal where we wish to embed the data such that Euclidean distance in the embedded space will match the co-occurrence statistics in the original space. We therefore use Multi-Dimensional-Scaling (MDS) for our embedding.

Most recently, Dekel *et al.* [4] and Talmi *et al.* [27] have been working on the same problem of Template Matching.

Dekel *et al.* proposed the Best-Buddies Similarity (BBS) measure. BBS maps the pixels of the template and the candidate window into a spatial-appearance space and then defines a similarity measure between the two point sets. In particular, they compute the mutual nearest neighbors between the two point sets and show that this measure is robust to outliers. This comes at a higher computational cost.

This work was later extended by Talmi *et al.* [27] that introduced two key ideas: The first is to enforce diversity in the mutual nearest-neighbor matching and the second is to explicitly consider the deformation of the nearest-neighbor field. To reduce the computational burden they use Approximate Nearest Neighbor.

Both methods do not achieve sub-pixel accuracy and do not generalize to other parametric transformations such as rotations. Our method, in contrast, is simpler, fits with existing template matching pipeline, and can generalize to other parametric transformations. In addition, our embedding scheme allows any other vision algorithm to benefit from co-occurrence error measure.

3. Method

SSD based template matching minimizes the following objective function: $\sum_p (T_p - R_p)^2$ where T is the template, $R \subseteq I$ is a region in image I, with the same size as T, and p is pixel location.

Co-occurrence based template matching (CoTM) maximizes the following objective function instead: $\sum_p M(T_p, R_p)$, where M is a (normalized) co-occurrence matrix that is learned from the image data. Once we have computed M, we can use it to give the cost of matching pixel value T_p with pixel value R_p. In case of multi-channel images (i.e., color or deep features), we quantize the image to a fixed number of k clusters using k-means. In what follows we define the co-occurrence matrix and discuss its properties.

3.1. Co-occurrence Matrix

A co-occurrence matrix $C(a, b)$ counts the number of times that the two pixel values a and b appear together in an image. Each pair contributes to C relative to their distance in the image plane. Formally:

$$C(a,b) = \frac{1}{Z} \sum_{p,q} exp(\frac{-d(p,q)^2}{2\sigma^2})[I_p = a][I_q = b] \quad (1)$$

where p and q are pixel location, I_p is the value of pixel p in image I, and Z is a normalization factor. σ is a user specified parameter and $[\cdot]$ equals 1 if the value inside the bracket is true and 0 otherwise. The use of a Gaussian weight captures our belief that pixel pairs that are close in the image plane matter more. In practice, we only consider pixels within a window proportional to σ.

Co-occurrence as described by Eq. 1 promotes pixel values that occur often in the image. To preserve pixel values that rarely occur in the image (and therefore we believe are important) we divide C by their prior probabilities to get the Pointwise Mutual Information (PMI) matrix:

$$M(a,b) = \frac{C(a,b)}{h(a)h(b)} \quad (2)$$

where $h(a)$ is the probability of observing pixel value a in the image (i.e., h is a normalized histogram of pixel values). While co-occurrence promotes pixel values that frequently appear in the image, PMI penalizes them.

Fig. 1 shows a query image and its PMI matrix M. For better visualization we show only the meaningful rows/columns of the matrix. The color patches along the axis of the PMI matrix indicate the cluster's color. The entries of the matrix are given in inverse grayscale, so bright colors mean a low PMI score and dark colors mean a high score. M(A) specifies the PMI of brown and blue colors. Since brown and blue rarely co-occur, their PMI is low. On the other hand, orange and white co-occur frequently, hence their PMI value, M(B), is high. This has nothing to do with the intensity differences between brown and blue vs orange and white. The only factor that affects M is how often pixel values co-occur. Another interesting entry in the matrix is the one of light and dark green, M(C). Even though they co-occur frequently, their PMI value is low. This is because the prior probabilities of light and dark green are fairly high in the image.

This property of M will come in handy when trying to match a template with a lot of background pixels in it (see Fig. 2). In this case we match two templates with different size to the same image. The result shows that in both cases only pixels that belong to the object have a high weight and the matching result is almost the same.

3.2. Template Matching

Given a template T and a region $R \subseteq I$, what is the probability that R matches T? Assuming Gaussian independent

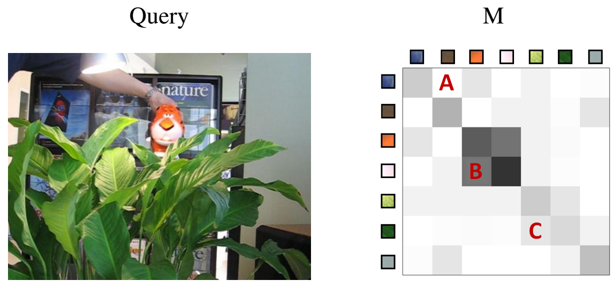

Figure 1. Co-occurrence Statistics: (Left) Query image. (Right) It's corresponding PMI matrix M. For better visualization we show only the important rows/columns of M. We collect co-occurrence statistics from the query image according to Eq. 2. M(A) has a low score because brown and blue rarely co-occur in the image. On the other hand, white and orange co-occur frequently, therefore their corresponding entry, M(B), is high. Light and dark green co-occur frequently but their score M(C) is low because each of them appear frequently in the image.

pixel noise, the (log) probability is:

$$\begin{aligned} &logPr(R|T;\Theta)\\ &= \textstyle\sum_p logPr(R_p|T_p;\Theta)\\ &= \textstyle\sum_p logG(T_p - R_p|0;\sigma)\\ &= -\frac{1}{2}|T|log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\textstyle\sum_p ||T_p - R_p||^2 \end{aligned} \tag{3}$$

where $G(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma}}exp(-\frac{1}{2\sigma^2}(x-\mu)^2)$ is the Gaussian density function. The last expression is the sum-of-squared-differences (SSD). Minimizing it maximizes the probability of region R matching T.

The Gaussian noise assumption is very strong. It assumes that the geometric transformation used to warp the template to the image is sufficient and hence all noise is due to intensity errors (that are modeled with a Gaussian). In practice, the transformation model we use might not be sufficient to capture the true deformations of the template.

Another problem with the Gaussian noise assumption is that it is very sensitive to outliers. For example, if some of the pixels in T or R belong to the background or are occluded then their error will have a very strong and negative effect on the outcome of the match.

We use co-occurrence statistics to address these issues. Specifically, we maximize the same objective function, but assume a different noise model. Assuming that pixels move locally and independently, we have that:

$$\begin{aligned} &logPr(R|T)\\ &= \textstyle\sum_p log(Pr(R_p|T_p))\\ &= \textstyle\sum_p log(Pr(R_p,T_p)) - \textstyle\sum_p log(Pr(T_p))\\ &= \textstyle\sum_p log(Pr(R_p,T_p)) \end{aligned} \tag{4}$$

where we drop $\sum_p log(Pr(T_p))$ because it depends only on the template which is fixed. As can be seen, in the Gaussian model we minimize the sum of squared distances, while in Eq. 4 we maximize the sum of joint probabilities. An outline of the algorithm is given in Algorithm 1.

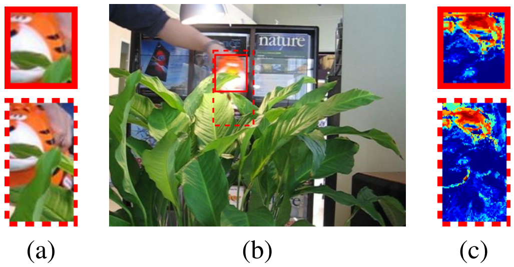

Figure 2. Background Influence: (a) small (top) and large (bottom) templates. (b) query image, we mark by solid (dashed) line the patch that best matches the small (large) template. (c) The per-pixel score $M(T_p, Q_p)$. Notice that adding more background pixels doesn't changes the overall score significantly.

Algorithm 1: Co-Occurrence Template Matching (CoTM)

```
Input  : Template ($T$), Query Image ($I$)
Output: Matching Region ($\hat{R}$)
$I_{idx}$ ← Quantize( $I$ );   $T_{idx}$ ← Quantize( $T$ )
$C$ ← Collect Co-occurrence($I_{idx}, T_{idx}$)        % Eq. 1
$M$ ← Normalize($C$)                                    % Eq. 2
Compute $S_R = \sum_p M(T_{idx}(p), R_{idx}(p))\ \ \forall R_{idx} \subseteq I_{idx}$
Return $\hat{R} = \arg\max_R (S_R)$
```

3.3. Embedding

In Sec. 3.2 we have shown how to use co-occurrence statistics to match a template to an image. We now extend this approach to address some of its limitations. First, it is not clear how to use this scheme to match with sub-pixel accuracy. Naively, one might suggest interpolating the input image and use the interpolated values in the co-occurrence matrix. Since co-occurrence is not a linear operation, this is clearly wrong. Second, sub-pixel accuracy will allow us to extend template matching to deal with more general transformations that do not work on integer pixel coordinates (i.e., rotations, 2D affine). Third, we would like to make use of existing template matching algorithms and not have to modify them.

On top of that, we would like other vision applications to take advantage of the co-occurrence measure. For example, Lucas-Kanade [19] uses a first order Taylor approximation to derive a gradient descent process to register a pair of images. This assumes the images are differentiable. Unfortunately, the matrix M is not differentiable which complicates things.

Another example is the Kernelized Correlation Filter (KCF) tracker [11]. KCF treats tracking as a binary classification problem that is solved efficiently in the frequency domain. However, it is not clear how to apply the Fourier transform to a space endowed with a co-occurrence similarity measure, and not a Euclidean distance.

To address these problems we propose to embed the pixel values in a new space that is endowed with a regular Euclidean metric. We can then perform Lucas-Kanade, KCF tracking, or any other vision algorithm, for that matter, in the new space. To do that, we assume that the co-occurrence matrix is an affinity matrix. Our goal is to map points with high affinity (i.e., high co-occurrence value) to nearby points in the embedded Euclidean space.

We use Multi-dimensional scaling (MDS) for the embedding. MDS takes a distance matrix as an input. It then uses eigenvalues decomposition to find a mapping to a given d-dimensional space such that L_2 distance in this space produces a distance matrix that is as close as possible to the input distance matrix. Formally, we look for points $\{y_1, ..., y_k\}$, $y_i \in R^d$ such that:

$$\arg\min_{\{\underline{y}\}} \sum_{a,b} (D(a,b) - ||\underline{y}_a - \underline{y}_b||_2)^2 \quad (5)$$

where the distance matrix is:

$$D(a,b) = -log\left(\frac{C(a,b)}{\sqrt{C(a,a)\cdot C(b,b)}}\right) \quad (6)$$

Manipulating C in this fashion ensures that D is symmetric with zeros across its diagonal [1].

Given the distance matrix D defined in Eq. 6 we use MDS to embed it in a d-dimensional space. Each pixel is now assigned the corresponding vector. Fig. 3 illustrates the embedding process. We show the embedding results to 1D (i.e., grayscale) and 3D (i.e., RGB images). Observe how textured regions in the input image are mapped to constant colors in the embedded space. In particular, in the 3D case, the different textures are mapped to Red, Green and Blue colors, which are far apart in color space.

Any vision algorithm can now operate on the embedded images. We demonstrate this using Template matching, Lucas-Kanade, and KCF tracking. The advantage of the embedding is that existing vision pipelines remain intact.

4. Results

We evaluated CoTM on two public benchmark datasets. The first, created by Dekel *et al.* [5] from 35 annotated *color* video sequences of the OTB dataset [31]. Those videos are challenging because of occlusions, nonrigid deformations, in-plane/out-plane rotation, luminance changes, scale differences and more. The dataset contains 105 template-image pairs. Each image pair consist of frames f and $f+20$, where f was randomly chosen. For each pair of frames, the template is the annotated ground-truth bounding box in frame f and the query image is frame $f + 20$.

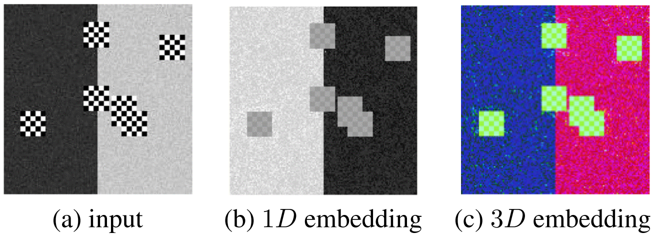
(a) input (b) $1D$ embedding (c) $3D$ embedding

Figure 3. Embedding: The input image (a) is embedded either to 1D (b) or 3D (c) space. Observe how the checkerboard texture is mapped to an almost constant color both in (b) and in (c). Vision algorithms that assume images are piecewise constant will benefit from working on the embedded images.

We also evaluate our method on a similar but larger dataset due to Oron *et al.* [21]. This benchmark was generated from the OTB dataset and includes both color and grayscale videos. The dataset consists of three data sets. Each dataset includes 270 template-image pairs and each image pair consist of frames f and $f + \Delta f$, where f was randomly chosen and $\Delta f \in \{25, 50, 100\}$[2].

The evaluation metric is based on the standard Intersection over Union (IoU). The area-under-curve (AUC) is used to compare between the different methods.

We use pre-trained VGG network [26] to generate deep features in a way similar to [21] and [27]. Specifically, we concatenate the 64 features of *conv*1_2 with the 256 features of *conv*3_4, which amounts to 320 features per pixel. *conv*3_4's size is 1/4 of the original image, in both dimensions. We used bilinear interpolation to resize it back to the image size.

4.1. Evaluation

We compare CoTM, on both color and deep feature images, to two state-of-the-art measures for template matching: Deformable Diversity Similarity [27] (DDIS) and Best-Buddies Similarity [5] (BBS). In addition, we compare our method to SSD. The success plots for all methods on the 105 template-image pairs benchmark are presented in Fig. 4.

Some comments are in order. The AUC score of template matching using color pixel values and standard SSD measure is quite poor at 0.43. Replacing color features with deep features, but keeping the SSD error measure, increases the score to 0.55. However, replacing the SSD similarity measure with co-occurrence statistics, while keeping the color features, increases the score to 0.62. In other words, using co-occurrence statistics of simple RGB values leads to better results than using deep features with standard SSD measure. Combining deep features and co-occurrence similarity measure brings the score to 0.67.

Examples of CoTM are shown in Fig. 5. As can be seen,

[1] The matrix D is not guaranteed to be a distance matrix because the triangle inequality is not guaranteed to hold. In practice, we did not observe any problems with the embedding.

[2] The dataset for $\Delta f = 100$ consists of only 252 video sequences.

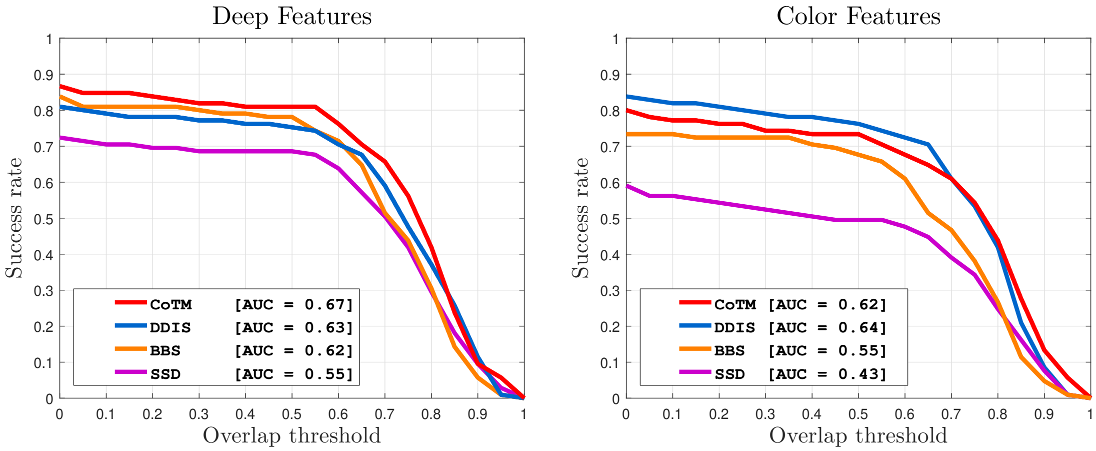

Figure 4. Accuracy: Evaluation on the benchmark of [5]: 105 template-image pairs. Left: evaluation on deep features. Right: evaluation on color features. AUC is shown in the legend.

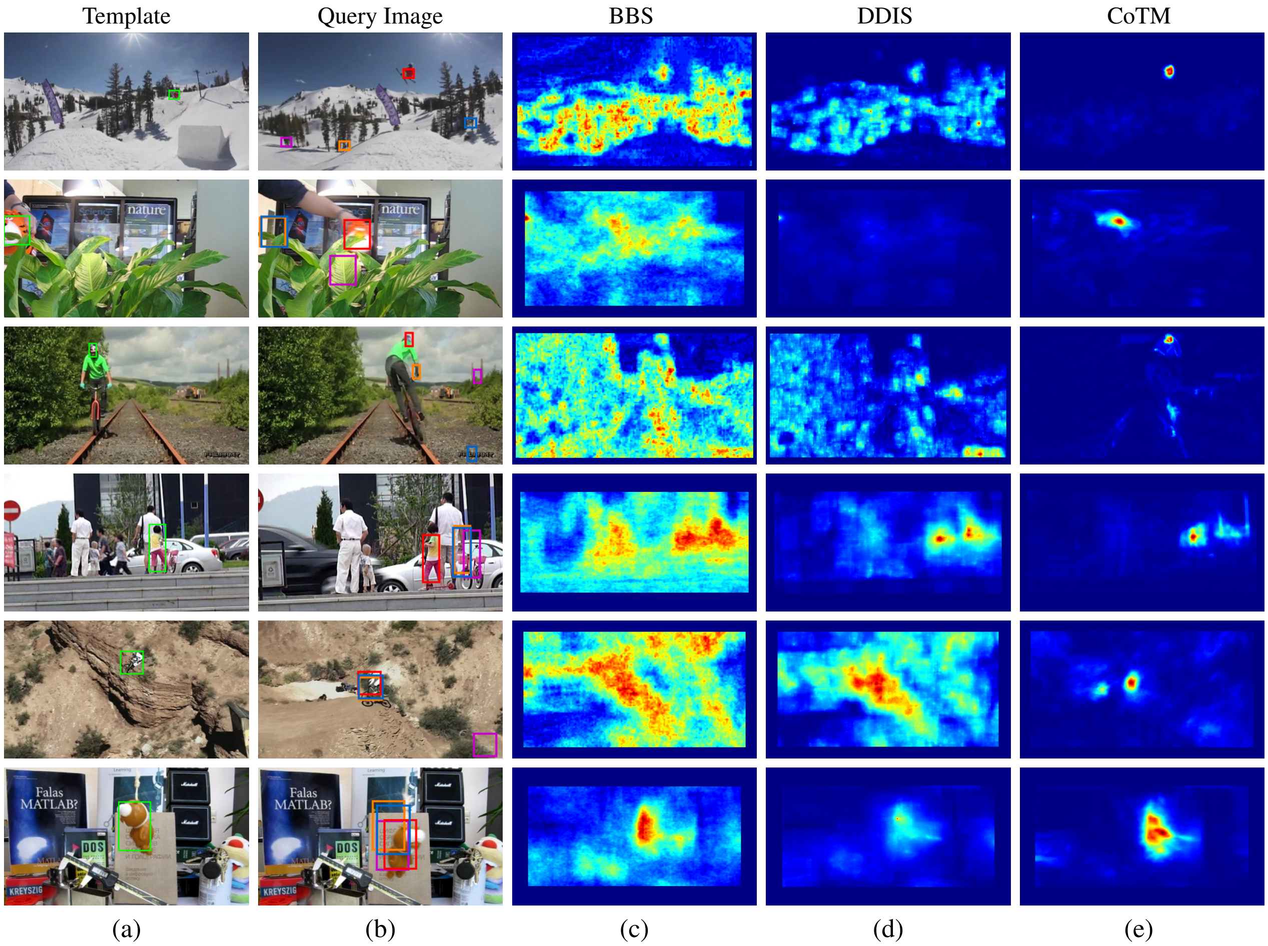

Figure 5. Results on real data using color features: (a) The template marked in green (b) Detection results in the query image of 6 different methods: CoTM, DDIS, BBS, SSD. (c-e) The corresponding likelihood maps of BBS, DDIS and CoTM respectively. Observe how sharp and localized are our heat maps.

Method	25	50	100	Mean
CoTM-DC	0.69	0.61	0.59	0.63
CoTM-D	0.68	0.60	0.59	0.62
DDIS-D	0.68	0.60	0.58	0.62
BBS-D	0.60	0.51	0.53	0.55
SSD-D	0.58	0.51	0.52	0.54
DDIS-C	0.65	0.59	0.54	0.59
CoTM-C	0.60	0.54	0.45	0.53
BBS-C	0.59	0.50	0.45	0.51
SSD-C	0.43	0.36	0.31	0.37
CoTM-DU	0.63	0.54	0.51	0.56
CoTM-CU	0.61	0.53	0.46	0.53

Table 1. Results on [22]: 270 pairs with $\Delta frame \in \{25, 50, 100\}$. We compare our method (CoTM) to that of [27] (DDIS). "-C" denote color features, "-D" denotes deep features. We also run our method on a concatenated feature vector of color and deep features (denoted "-DC"). For the last two rows (-DU, -CU) we have computed the k-means prototypes on an external image set, instead of computing them per-image. As can be seen, performance does not change much.

the heat map of CoTM is usually clean with a very strong and localized peak at the correct location.

We repeated our experiment on the (larger) data set of [22] and report results in Table 1. As can be seen, Talmi *et al.* [27] outperforms us on color features and we outperform them on deep features. Concatenating color and deep features per pixel and using co-occurrence statistics we achieve an AUC score of 0.69 which is the highest reported score on this benchmark.

We have also evaluated the importance of prototypes (i.e., the cluster centers of the k-means quantization step) on performance. To this end, we have computed a *universal* set of k-means prototypes from some external image dataset, and used them instead of running k-means on each image. Results are reported in Table 1 as CoTM-DU and CoTM-CU. As can be seen, the accuracy does not change much.

Our method is fast, straightforward to implement and does not require the use of Approximate Nearest Neighbor packages. Our un-optimized MATLAB code takes on average 2.7 seconds to process a single template-image pair using color features on an $i7$ Windows machine with 32GB of memory. This excludes the k-means step that takes a couple of seconds.

4.2. Evaluation of CoTM Embedding

Next, we evaluated the MDS embedding scheme for template matching using Eq. 6 on the 105 data set. In particular we evaluate embedding into a 3 as well as 256 dimensional space. Once we embed the template and image we use the standard SSD error measure for template matching. Detection results are summarized in Fig 6. We found that Co-occurrence Embedding Template Matching (CoETM) works better than the Best-Buddies-Similarity measure of Dekel *et al.* [5]. Our method is simpler to implement and faster to compute. The embedding can be done as a preprocessing stage and the embedded images can be used in existing template matching software.

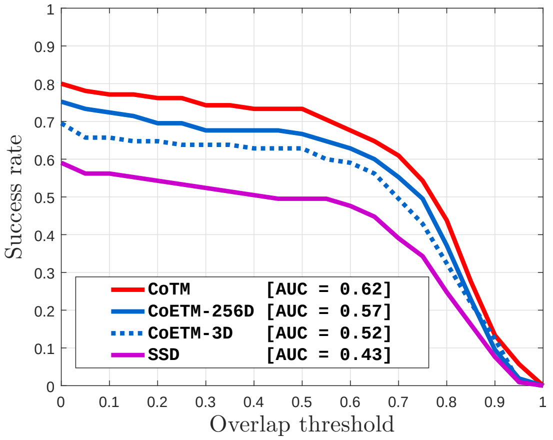

Figure 6. CoTM Embedding (CoETM): evaluation on [5] shows that CoETM performs simillarly to CoTM.

To demonstrate the power of embedding we have coupled it with the Lucas-Kanade registration algorithm and evaluated it on the 105 data set. For each pair, we generated an initial guess within half bounding box distance from the ground truth. We used this guess to initialize a 4-level pyramid LK algorithm. The exact same algorithm was tested on color as well as embedded images. We used the same IoU metric to measure success. For the embedding we use MDS scheme of dimension 3.

Fig. 7 shows that LK with CoE (i.e., CoTM on embedded images) converges to final bounding boxes that are better than regular LK. Some example are shown in Fig. 8. In particular, observe that the last example in the figure shows LK with a 2D Translation+Rotation. It is not obvious how to extend the work of Dekel *et al.* [5] or Talmi *et al.* [27] to support such motion models.

We also run an out-of-the-box KCF tracker [11] on the OTB dataset [30] and report results in Fig. 9. As can be seen, using Co-occurrence embedding improves results by about 10% with no modifications to the original KCF algorithm. To accelerate run-time, we use only the first frame in each sequence to compute the co-occurrence embedding and apply it to the rest of the frames in that sequence.

4.3. Limitations

CoTM suffers from a number of limitations. First, we found that co-occurrence on gray pixel values does not work well. We also found that performance degrades when the pixel values of the template occur frequently in the background. This is because in such cases background pixels are

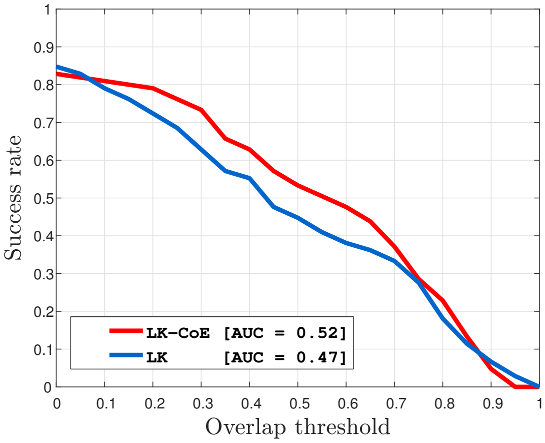

Figure 7. Template matching accuracy using LK: Evaluation on [5]: 105 template-image pairs. LK with Co-occurrence embedding (LK-CoE) outperforms regular LK.

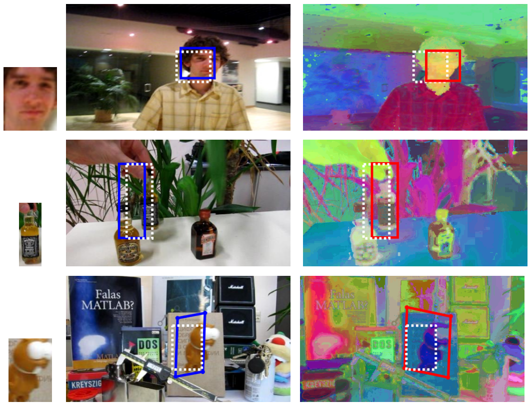

Figure 8. Co-Occurrance Luckas Kanade: Left: template. Center: result of regular LK. Dashed white rectangle is initial guess. Blue rectangle is final result. Right: result of LK on embedded images. Dashed white rectangle is initial guess. Red rectangle is final result. Our result is far from the initial rectangle, indicating that the basin of attraction is larger and the convergence is better.

not down-weighted. Finally, we have not addressed illumination changes and leave this for future research. Failure examples are shown in Fig. 10. Many of these failure cases can be mitigated by working on deep features.

5. Conclusions

We presented a new measure for pixel similarity that is based on the co-occurrence statistics. Instead of measuring the intensity difference between pixel values, we measure their co-occurrence score. Pixel values that co-occur often are penalized less than pixel values that co-occur frequently. This is because co-occurrence captures texture to some degree. Hence, pixel values that come from the same textured region probably have a high co-occurrence score.

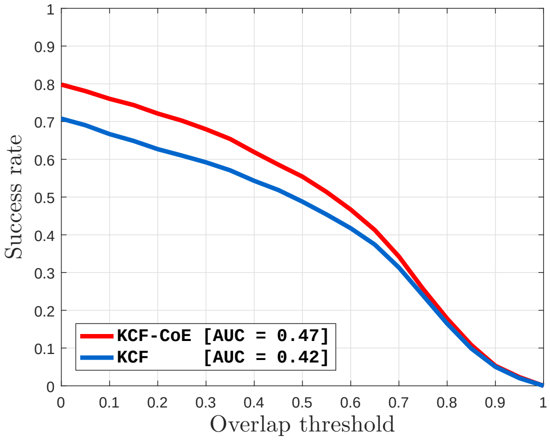

Figure 9. KCF Tracking: Evaluation on [31]: KCF with Co-occurrence embedding (KCF-CoE) outperforms regular KCF.

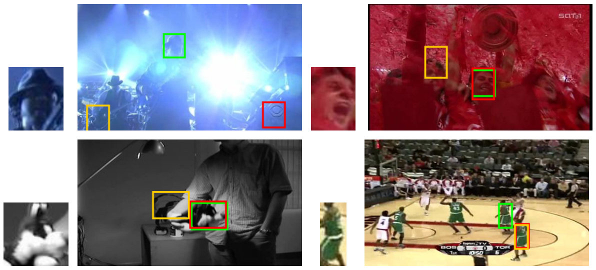

Figure 10. Limitations: shows our failure cases. Left: template. Right: query image. On the image we mark the ground truth location in green and our detection results using color features and deep features in yellow and red respectively.

Co-occurrence statistics captures global image statistics, as opposed to local image statistics that are captured by various patch representations. Combining co-occurrence statistics (that capture global statistics) with deep features (that capture local statistics) leads to state of the art results in template matching on standard datasets.

We then suggest an embedding scheme that maps pixel values in the input space to a new space such that pixel values that co-occur often are mapped to nearby points in the embedded space. This allows any vision algorithm to enjoy the power of co-occurrence statistics by working on the embedded images, instead of the original ones. We demonstrate the power of this embedding on the Lucas-Kanade image registration algorithm and the Kernelized Correlation Filter (KCF) tracker. Both algorithms enjoy a 10% boost in performance just by working on the embedded images instead of the original ones.

Acknowledgments: Parts of this research were supported by ISF grant 1917/15.

References

[1] J.-H. Chen, C.-S. Chen, and Y.-S. Chen. Fast algorithm for robust template matching with m-estimators. *IEEE Transactions on Signal Processing*, 51(1):230–243, Jan 2003.

[2] C. D., R. V., and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 2, pages 142–149, 2000.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[4] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman. Best-buddies similarity for robust template matching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2021–2029, 2015.

[5] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman. Best-buddies similarity for robust template matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2021–2029. IEEE, 2015.

[6] N. Dowson and R. Bowden. Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation. *IEEE transactions on pattern analysis and machine intelligence*, 30(1):180–185, 2008.

[7] G. Egnal. Mutual Information as a Stereo Correspondence Measure. Technical report, University of Pennsylvania, Jan. 2000.

[8] E. Elboher, M. Werman, and Y. Hel-Or. The generalized laplacian distance and its applications for visual matching. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2315–2322, 2013.

[9] A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 1973.

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):583–596, 2015.

[12] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 833–840, 2002.

[13] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):328–341, Feb. 2008.

[14] H.-W. Hseu, A. Bhalerao, R. Wilson, and C. C. Al. Image matching based on the co-occurrence matrix, 1999.

[15] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Crisp boundary detection using pointwise mutual information. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pages 799–814, Cham, 2014. Springer International Publishing.

[16] R. J. Jevnisek and S. Avidan. Co-occurrence filter. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, pages 1033–1040, 2003.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004.

[19] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[20] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 16(2):187–198, 1997.

[21] S. Oron, T. Dekel, T. Xue, W. T. Freeman, and S. Avidan. Best-buddies similarity - robust template matching using mutual nearest neighbors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[22] S. Oron, T. Dekel, T. Xue, W. T. Freeman, and S. Avidan. Best-buddies similarity-robust template matching using mutual nearest neighbors. *arXiv preprint arXiv:1609.01571*, 2016.

[23] W. Ouyang, F. Tombari, S. Mattoccia, L. D. Stefano, and W.-K. Cham. Performance evaluation of full search equivalent pattern matching algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:127–143, 2011.

[24] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, Aug 2003.

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[27] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template matching with deformable diversity similarity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[28] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[29] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[30] Y. Wu, J. Lim, and M. Yang. Online object tracking: A benchmark. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2411–2418, 2013.

[31] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.