OATM: Occlusion Aware Template Matching by Consensus Set Maximization

Simon Korman[1,3]
[1]Weizmann Institute of Science

Mark Milam[2]
[2]Northrop Grumman

Stefano Soatto[1,3]
[3]UCLA Vision Lab

Abstract

We present a novel approach to template matching that is efficient, can handle partial occlusions, and comes with provable performance guarantees. A key component of the method is a reduction that transforms the problem of searching a nearest neighbor among N high-dimensional vectors, to searching neighbors among two sets of order $\sqrt{N}$ vectors, which can be found efficiently using range search techniques. This allows for a quadratic improvement in search complexity, and makes the method scalable in handling large search spaces. The second contribution is a hashing scheme based on consensus set maximization, which allows us to handle occlusions. The resulting scheme can be seen as a randomized hypothesize-and-test algorithm, which is equipped with guarantees regarding the number of iterations required for obtaining an optimal solution with high probability. The predicted matching rates are validated empirically and the algorithm shows a significant improvement over the state-of-the-art in both speed and robustness to occlusions.

1. Introduction

Matching a template T (a small image) to a target I (a larger image) can be trivial to impossible depending on the relation between the two. In the classical setup, when I is a digital image and T is a subset of it, this amounts to a search over the set of N discrete 2D-translations, where N would be the number of pixels in I. When T and I are images of the same scene taken from different vantage points, their relation can be described by a complex deformation of their domain, depending on the shape of the underlying scene, and of their range, depending on its reflectance and illumination. For a sufficiently small template, such deformations can be approximated by an *affine* transformation of the domain ("warping"), and an affine ("contrast") transformation of the range ... *except for occlusions:* An arbitrarily large portion of the template, including all of it, may be occluded and therefore have no correspondent in the target image.

This poses a fundamental problem to many low-level tasks: To establish local correspondence (co-visibility), the template should be large, so as to be discriminative. But increasing the area increases the probability that its correspondent in the target image will be occluded, which causes the correspondence to fail, *unless occlusion phenomena are explicitly taken into account.*

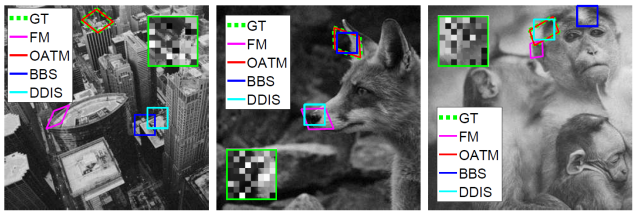


Figure 1. **Instances of the occlusion experiment (Sec. 4.2)** A template (overlaid in green) that is 60% occluded by random blocks is searched for in an image. **OATM** shows the best results in dealing with significant deformation and occlusion (use zoom for detail).

In this work we model occlusions explicitly as part of a robust template matching process where the co-visible region is assumed to undergo affine deformations of the domain and range, up to additive noise. We search for transformations that maximize *consensus,* that is the size of the co-visible set, in a manner that is efficient and comes with provable convergence guarantees.

Efficiency comes from the first contribution - a *reduction method* whereby the linear search of nearest neighbors for the d-dimensional template T through N versions in the target image is converted to a search among two *sets* of vectors, *with each set of size* $O(\sqrt{N})$ (Sect. 2.2). This reduces the search complexity from $O(N)$ to $O(\sqrt{N})$, which is practical even for very large search spaces, such as the discretized space of *affine transformations.*

For this method to work, we need a *hashing scheme that is compatible with occlusions*, which we achieve by adapting the scheme of Aiger *et al.* [2], leading to our second contribution: Rather than reporting close neighbors under the Euclidean ℓ_2 norm, we are interested in reporting pairs of vectors that are compatible, up to a threshold, on a maximum (co-visibility) consensus set. Our hashing scheme is akin to a random consensus (RANSAC-type) procedure under the ℓ_∞ norm (Sect. 2.3).

Finally, our third contribution is an analysis of the algorithm (Sect. 2.4), specifically regarding guarantees on the number of candidate hypotheses required for obtaining the

optimal solution, in the sense of maximal inlier rate, within a certain probability.

While for many low-level vision tasks *speed*, not convergence guarantee, is the key, there are applications where being able to issue a certificate is important, such as high-assurance visual pose estimation for satellite maneuvering. In our case, we achieve both speed and assurance, all the while being able to handle occlusions, which allows using larger, and therefore more discriminative, templates.

The algorithm is rather generic and is presented for a general geometric transformation of the domain space, while possible explicit decompositions are given for the 2D-translation and 2D-affine groups. In the experimental section, our algorithm is shown empirically to outperform the state-of-the-art in affine template matching [17] both in terms of efficiency and robustness to occlusion. In addition, it shows some clear advantages over some modern image descriptors on the recent HPatches [4] benchmark.

1.1. Related work

Research in template matching algorithms has focused heavily on efficiency, a natural requirement from a low level component in vision systems. This was largely achieved in the limited scope of 2D-translation and ℓ_p similarity, where full-search-equivalent algorithms accelerate naive full-search schemes by orders of magnitude [22]. unlike in real-time applications, such as robotic navigation and augmented reality, there are applications where accuracy and performance guarantees are important, such as high-assurance pose estimation for high-value assets, such as satellites or industrial equipment. This requires extending the scope of research in several aspects.

One line of works focuses on *geometric deformations* due to camera or object motion. Early works such as [11, 26] extend the sliding window approaches to handle rotation and scale. The Fast-Match algorithm [17] was designed to handle 2D-Affine transformations. It minimizes the sum-of-absolute-differences using branch-and-bound, providing probabilistic global guarantees. [29] uses a genetic algorithm to sample the 2D-affine space.

To achieve *photometric invariance*, [13] introduced a fast scheme for matching under non-linear tone mappings, while [10] used the Generalized Laplacian distance, which can handle multi-modal matching. Our method can provide affine photometric invariance, i.e., up to global brightness and contrast changes.

In this work we propose a *quadratic* improvement upon the runtime complexity of these methods, which depends linearly on the size of the search-space (*i.e.,* exponential in its dimension). More recently we are seeing attempts at matching under 2D-homographies using deep neural networks [9, 20], although these methods do not provide any guarantees and like the previously mentioned methods - they were not designed to handle *partial occlusion*.

Two recent works can handle *both* geometric deformations and partial occlusion through similarity measures between rectangular patches: the Best Buddies Similarity (BBS) measure [8], based on maximizing the number of mutual nearest-neighbor pixel pairs, and Deformable Diversity Similarity (DDIS) [25], that examines the nearest neighbor field between the patches. DDIS dramatically improves the heavy runtime complexity of BBS, but is limited in the extent of deformation it can handle, since it penalizes large deformations. Also, the sliding window nature of these methods limits the extent of occlusion they can handle. While OATM is limited to handling *rigid* transformations, it is provably able to efficiently handle high levels of deformation and occlusion.

Another relevant and very active area of research is learning discriminative descriptors for image patches (natural patches or those extracted by feature detectors), from the earlier SIFT [19] and variants [7, 23] to the more recent [24, 5, 12]. We show OATM to be superior in its ability to match under significant deformation and occlusion.

Lastly, the problem of occlusion handling was addressed in many other areas of computer vision, including tracking [31, 30, 15], segmentation [28], image matching [27], multi-target detection [6], flow [14] and recognition [21].

Within a landscape of "X-with-deep-learning" research, our work is counter-tendence: We find that the need to provide provable guarantees in matching, albeit relevant to niche applications, is underserved, and data-driven machine learning tools are not ideally suited to this task.

2. Method

2.1. Problem Definition

In template matching, one assumes that a template T and an image I are related by a geometric transformation of the domain $F = \{f : \mathbb{R}^2 \to \mathbb{R}^2\}$ and a photometric transformation of the range space. The goal is to determine the transformation of the domain, despite transformations of the range. Here we assume that both T and I are discretized, real valued, square images, and hence can be written as $T : \{1, \ldots, n\}^2 \to \mathbb{R}$ (and similarly $I : \{1, \ldots, m\}^2 \to \mathbb{R}$), where T and I are $n \times n$ and $m \times m$ images, respectively. The set of transformations F can be approximated by a discrete set of size N, possibly large, up to a desired tolerance. For example, in the standard 2D-translation setup, the set F contains all possible placements of the template over the image at single pixel offsets, and hence $N = |F| \approx (m - n)^2$ with a tolerance of one pixel. Moreover, in our analysis we will assume nearest-neighbor interpolation (rounding) which allows us to simplify the discussion to fully discretized transformations of the form $f : \{1, \ldots, n\}^2 \to \mathbb{Z}^2$.

With a slight abuse of notation we indicate with $p \in T$ (and likewise $q \in I$) a pixel p in the template domain $\{1,\ldots,n\}^2$ and $T(p)$ will denote its real valued intensity.

For a given transformation f, we define the (photometric) residual, or reprojection error, at pixel $p \in T$ by $res_f(p) = |T(p) - I(f(p))|$. The known "brightness constancy constraint" guarantees that the residual can be made small (to within a threshold) by at least one transformation f. However, it is only valid for portions of the scene that are Lambertian, seen under constant illumination and most importantly: co-visible (unoccluded).

We are now ready to pose Occlusion-Aware Template Matching (OATM) as a Consensus Set Maximization (CSM) problem, where we search for a transformation under which a maximal number of pixels are co-visible, *i.e.,* mapped with a residual that is within the a threshold.

Definition 1. [Occlusion-Aware Template Matching (OATM)] *For a given error threshold t, find a transformation f^* given by:*

$$f^* = \operatorname*{argmax}_{f \in F} \sum_{p \in T} [res_f(p) \leq t] \tag{1}$$

where $[\cdot]$ represents the indicator function.

Our reduction to a product space relies extensively on a distance notion between geometric transformations (which depends on the source domain - the template T).

Definition 2. [Distance Δ between transformations] *Let $f_1, f_2 \in F$. We define the distance $\Delta(f_1, f_2) = \max_{p \in T} \|f_1(p) - f_2(p)\|$ where $\|\cdot\|$ represents the Euclidean distance in the (target) domain of the image I.*

2.2. Reduction to a Product Space

Recall (Equation (1)) that our goal is to find an optimal transformation f^*, one whose residual

$$res_{f^*}(p) = |T(p) - I(f^*(p))| \tag{2}$$

is below a threshold t at as many pixels $p \in T$ as possible. In order to optimize (1) we would need to compare T to N possible target vectors $I(f(T))$ (all possible transformed templates in the target image).

The main idea here will be to enumerate the search space in a very different way. On the source image side we define a set U of templates (vectors) obtained by local perturbations of the template T, while on the target side we define a set V of templates that "covers" the target image I in a sense that every target template location will be close to one of those in V. In such a way, if a copy of the template appears in the image, there must be a pair of similar templates (vectors) $u \in U$ and $v \in V$. Refer to Figure 2 to get the intuition for the 2D-translation case.

Formally, for a given tolerance $\epsilon > 0$, let $f \in F$ be a transformation such that $\Delta(f, f^*) < \epsilon$. For an arbitrary $p' \in T$, if we assume the existence of some $p \in T$ such that $f(p) = f^*(p')$, which is the case in our model under the assumption of co-visibility, by substituting $p' = f^{*-1}(f(p))$ in Equation (2), we get:

$$res_{f^*}(p') = |T(f^{*-1}(f(p))) - I(f(p))| \,. \tag{3}$$

If we set $h = f^{*-1} \circ f$, we can write:

$$res_{f^*}(p') = |T(h(p)) - I(f(p))| \tag{4}$$

for pixels p in the sub-template $T_h = \{p \in T \,:\, h(p) \in T\}$, for which $h(p) = p' \in T$.

Regarding h, since we know that $\Delta(f, f^*) < \epsilon$, it is easy to see that $\Delta(h, id) < \epsilon / s(f^*)$, where id is the identity transformation and $s(f^*)$ is the minimal scale of f^*, defined by $s(f) = \min_{p \in T} \|f(p)\| / \|p\|$.

If we call $\epsilon' = \epsilon / s(f^*)$ we can now define the restricted subset of functions (which is a *ball* of radius ϵ' around the identity, in the function space F):

$$F_{\epsilon'} = \{h \in F \,:\, \Delta(h, id) < \epsilon'\} \tag{5}$$

Let $Net_\epsilon(F)$ be an arbitrary ϵ-net over the space F, with respect to the distance Δ. Namely, for any $f \in F$ there exists some $f' \in Net_\epsilon(F)$ such that $\Delta(f, f') < \epsilon$.

The result is that we have decomposed the search for an optimal $f^* \in F$ in Eq. (1), to the search of the equivalent (recall that $h = f^{*-1} \circ f$) optimal pair (h, f) in the product space $F_{\epsilon'} \times Net_\epsilon(F)$. Namely, we can reformulate the OATM problem (Equation (1)) as:

$$f^* = \operatorname*{argmax}_{\substack{h \in F_{\epsilon'} \\ f \in Net_\epsilon(F)}} \sum_{p \in T_h} \frac{1}{|T_h|} \Big[|T(h(p)) - I(f(p))| \leq t \Big] \tag{6}$$

For simplicity of description and implementation we can work with a fixed subtemplate T' of T, defined by the intersection of all sub-templates $\{T_h\}_{h \in F_{\epsilon'}}$, which results in:

$$f^* = \operatorname*{argmax}_{\substack{h \in F_{\epsilon'} \\ f \in Net_\epsilon(F)}} \sum_{p \in T'} \Big[|T(h(p)) - I(f(p))| \leq t \Big] \tag{7}$$

It may appear that, up to this point, we stand to gain nothing, since under any reasonable discretization of the transformation sets $Net_\epsilon(F)$ and $F_{\epsilon'}$, it holds that $|F| \approx |Net_\epsilon(F)| \cdot |F_{\epsilon'}|$, i.e. that the size of the search space remains unchaged. However, this decomposition allows us to design preprocessing schemes for two sets of vectors[1]

$$U = \{T(h(T'))\}_{h \in F_{\epsilon'}} \tag{8}$$

$$V = \{I(f(T'))\}_{f \in Net_\epsilon(F)} \tag{9}$$

in a manner that enables an efficient search over the terms $|T(h(p)) - I(f(p))|$ from (7) for all $(h, f) \in F_{\epsilon'} \times Net_\epsilon(F)$.

[1] $h(T')$ and $f(T')$ are shorthands for $\{h(p)\}_{p \in T'}$ and $\{f(p)\}_{p \in T'}$

Efficiency comes from designing the product space in a way that the sets U and V have approximately equal size ($\sqrt{N}$) and from using a search algorithm whose complexity depends on the *sum* of the space sizes (order $\sqrt{N}$), and not on their *product* (of size N). We provide explicit decompositions for the 2D-translation and 2D-affine spaces.

2.3. Search by Random Grid based Hashing

We have transformed the problem of matching between a single vector and N target vectors to that of finding matching vectors between two sets of $\sim\sqrt{N}$ vectors. Matching between a pair of high-dimensional point sets is a classical problem in the search literature, clearly related to the problem of finding all close neighbors in a single point-set. Our approach is based on random grid hashing [1] - an algorithm that is straightforward to implement and which has been shown to work well in practice [2].

In [1], to hash a collection of d dimensional points, the space is divided into cells by laying a randomly shifted uniform grid (each cell is an axis-parallel cube with side-length c). The points are arranged accordingly in a hash table and then all pairs of points that share an entry in the hash table are inspected, reporting those whose distance is below the specified threshold. The process is then repeated a suitable number of times in order to guarantee, with high probability, that all or most pairs of close points are reported.

Unlike the work of Aiger *et al.* [1, 2] that uses the ℓ_2 norm to measure the similarity between vectors, we use the number of coordinates whose absolute difference is below a threshold. Furthermore, we replace the dimensionality reduction in [2] (a Johnson–Lindenstrauss transform) by a random choice of a small number of coordinates (pixels), in order to enable matching under occlusions. These changes require a different analysis of the algorithm. Refer to Algorithm 1 for a summary of our basic hashing module.

2.4. Analysis

The main result needed for a high-assurance template matcher is a guarantee on the success probability of Algorithm 1. The following term will be used in our claims:

$$P(\alpha, d, \hat{d}) = \frac{\binom{\alpha d}{\hat{d}}}{\binom{d}{\hat{d}}} = \frac{\alpha d \cdot (\alpha d - 1) \cdot \ldots \cdot (\alpha d - \hat{d} + 1)}{d \cdot (d-1) \cdot \ldots \cdot (d - \hat{d} + 1)}$$

Claim 1. [analysis of Algorithm 1] *Algorithm 1 succeeds (reports a pair $u, v \in U \times V$ with maximal possible inlier rate of α) with probability at least*

$$P(\alpha, d, \hat{d}) \cdot \left(1 - \frac{t}{c}\right)^{\hat{d}} \tag{10}$$

Proof. The derivation is straightforward, since the algorithm succeeds if a pair of optimal matching vectors u, v collide in the hash table. A collision is guaranteed to occur, given a combination of two events. First, the event that the set of the $\hat{d}$ sampled dimensions is a subset of the αd inlier dimensions. This occurs with probability $P(\alpha, d, \hat{d})$, since this is a hyper-geometric distribution with αd success items among a population of d, with $\hat{d}$ samples all required to be success items. Second, we need to multiply by the probability that a collision occurred subject to the randomness in the grid offset. In this case, the $\hat{d}$-dimensional $\hat{u}$ and $\hat{v}$ differ by at most t in each coordinate. Therefore, and since the offset is uniform and independent between coordinates, $\hat{u}$ and $\hat{v}$ are mapped into the same cell (and hence collide in the hash table) with probability at least $(\frac{c-t}{c})^{\hat{d}} = (1 - \frac{t}{c})^{\hat{d}}$. □

---

input: Sets U and V of vectors in $\mathbb{R}^d$; threshold t;
output: A vector pair $(u, v) \in U \times V$ with maximal found consensus set (inlier rate)
parameters: Sample dimension $\hat{d}$; cell dimension c;

---

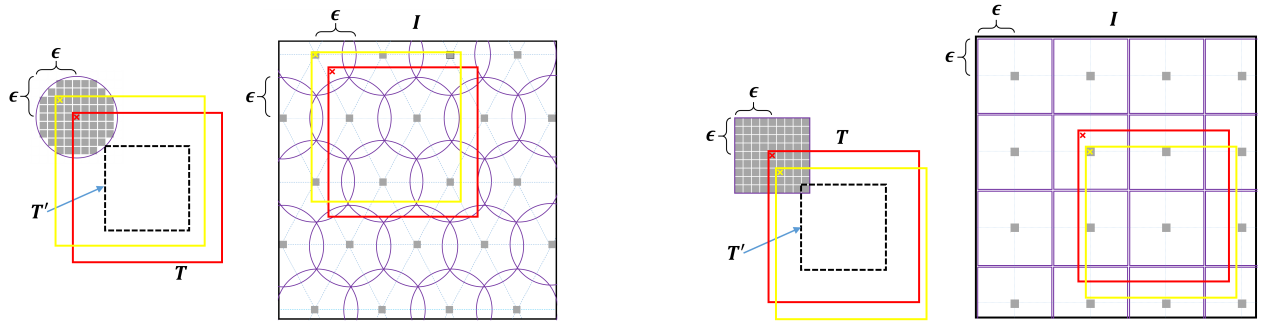
1. Pick $\hat{d}$ random dimensions out of $1, \ldots, d$.
2. Let $\hat{U}$ and $\hat{V}$ be the vector sets U and V reduced to the $\hat{d}$ random dimensions.
3. Generate a random $\hat{d}$-dimensional offset vector o in $[0, c]^{\hat{d}}$.
4. Map each vector in $\hat{U}$ and $\hat{V}$ into a $\hat{d}$-dimensional integer, according to $Map(\hat{v}) = \lfloor (\hat{v} + o)/c \rfloor$.
5. Arrange the resulting integers into a hash table using any hash function from $\mathbb{N}^{\hat{d}}$ to $\{1, \ldots, |U|\}$.
6. Scan the hash table sequentially, where for each pair of vectors $\hat{u}$ and $\hat{v}$ that share a hash value, count the number of inlier coordinates in $i \in \{1, \ldots, d\}$ (those for which $|u(i) - v(i)| \leq t$).
7. Return a pair u, v with maximal found inlier rate

Algorithm 1: Consensus Set Maximization in vector sets.

Claim 2. [analysis of Algorithm 1 - stronger version] *Assume there exists a pair $u, v \in U \times V$ which are identical up to a zero-mean Gaussian noise with standard deviation σ at an α-fraction of their coordinates. Algorithm 1 succeeds (reports a pair $u, v \in U \times V$ with inlier rate at least α) with probability at least*

$$P(\alpha, d, \hat{d}) \cdot \left(\int_0^c (1 - \frac{x}{c}) \cdot \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \cdot exp(-\frac{x^2}{2\sigma^2}) dx \right)^{\hat{d}} \tag{11}$$

Proof. The only difference here compared to the previous claim is regarding the probability of vectors of inlier coordinates falling into a single cell. The difference is in the definition of inliers, where here we not only assume a maximal absolute difference of t at each coordinate but we rather make the stronger (but realistic) assumption that the vectors at inlier coordinates differ only due to Gaussian noise of



(a) ϵ-net construction (b) simple construction

Figure 2. **Illustration of two possible decompositions for 2D-translation**. In each of (a) and (b) the sets of sampled vectors (templates) U (from T) and V (from I) are represented by gray pixels which denote the top left corner position of the sampled templates. If the Template (red) appears in the target image, there will be a respective pair of matching samples in U and V (shown in yellow). The parameter ϵ is taken such that the number of samples (number of gray squares) on both sides is approximately equal (both approximately $\sqrt{N}$).

a known standard deviation. In such a case, the absolute difference per coordinate follows a folded Gaussian distribution (see e.g. [18]), and therefore we integrate over the possible absolute differences x in the range $[0, c]$. □

2.5. Occlusion-Aware Template Matching

Given Algorithm 1 and its performance guarantees, we can now specify our complete OATM template matching algorithm. The template matcher will run Algorithm 1 a certain number of times and return the target location in the image, which corresponds to the overall best pair of vectors found. As a reminder, Algorithm 1 returns a pair of vectors which are of the form $\{T(h(p))\}_{p\in T'}$ and $\{I(f(p))\}_{p\in T'}$, which suggests the pair of transformations (h, f) as a candidate solution, from which a single transformation $f^* = f \circ h^{-1}$ can be extracted.

There are two reasons to evaluate directly the inlier rate $P^* = \frac{1}{|T|}\sum_{p\in T}\left[|T(p) - I(f^*(p))| \leq t\right]$ instead of the proxy $\frac{1}{|T'|}\sum_{p\in T'}\left[|T(h(p)) - I(f(p))| \leq t\right]$. One is to avoid interpolation errors by applying the concatenated transformation $f^* = f \circ h^{-1}$ directly. The second and more important one is that the detected inlier rate reflects only pixels of T' in a sub-template of T.

---

input: template T and image I; threshold t;
family of transformations F (of size N);
output: $f \in F$ with maximum consensus (Eq. (1))

---

1. Decompose F into the product $F_{\epsilon'} \times Net_\epsilon(F)$ choosing an ϵ s.t. $|F_{\epsilon'}| \approx |Net_\epsilon(F)| \approx \sqrt{N}$.
2. Construct the vector sets U and V (Eqs. (8)-(9)).
3. **repeat** Algorithm 1 for k times (with U, V and t) to obtain transformations $\{f_i\}_{i=1}^k$.
4. **return** the transformation f_i with largest consensus set (Eq. (1)).

Algorithm 2: OATM: Occlusion Aware Template Matching

Occlusion-Aware Template Matching (OATM) is summarized in Algorithm 2. It consists of running Algorithm 1 for k iteration. If we denote by P_α the success probability of Algorithm 1, given in Equation (11) of Claim 2, it holds that the success probability of Algorithm 2 is at least:

$$1 - (1 - P_\alpha)^k \tag{12}$$

and conversely, the number of iterations k needed in order to succeed with a predefined probability p_0 (e.g. 0.99) is: $\log(1 - p_0)/\log(1 - P_\alpha)$.

It is important to note that the number of iterations k can be determined *adaptively*, based on the findings of previous rounds. As is common in the RANSAC pipeline, every time the best maximal consensus (inlier rate) is updated, the number of required iterations is decreased accordingly.

Notice that the algorithm is generic with respect to the underlying transformation space F. It does however require the knowledge of how to efficiently decompose it into a product space (Step 1). We next describe two such constructions for 2D-translations and provide a construction for the 2D-affine group in the supplementary material [16].

2.6. 2D-translation constructions

Recall that at the basis of our algorithm is the decomposition of the transformation search space F into a product of spaces $F_{\epsilon'} \times Net_\epsilon(F)$, controlled by a parameter ϵ. Depending on the structure of the space F ($|F| = N$), we will pick a value of ϵ (and ϵ') for which $|F_{\epsilon'}| \approx |Net_\epsilon(F)| \approx \sqrt{N}$, in order to minimize the complexity which depends on the sum of the sizes of the product spaces. We make the decomposition explicit for the case of 2D-translations.

Since no scale is involved, $s(f^*)=1$ and hence $\epsilon'=\epsilon$. Given a square template T and image I of dimensions $m \times m$ and $n \times n$, the subspaces F_ϵ and $Net_\epsilon(F)$ can be constructed using a hexagonal cover of a square by circles of radius ϵ, as is depicted in Figure 2(a). The sizes of the resulting subspaces F_ϵ and $Net_\epsilon(F)$: $\pi\epsilon^2$ and $(n-m+1)^2/(1.5\sqrt{3}\epsilon^2)$, can be made equal by tuning ϵ.

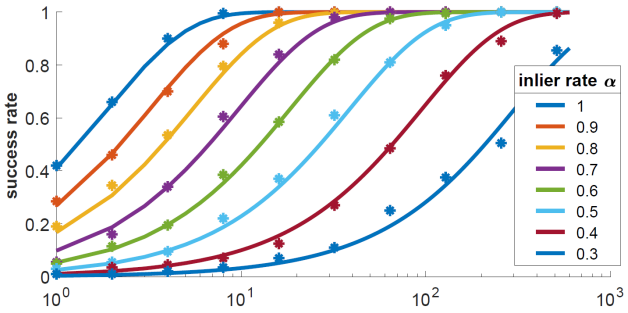


Figure 3. **Empirical validation of Algorithm 2's guarantees**. The theoretical success probabilities of OATM as a function of the number of iterations k (solid curves) for different inlier rates α (notice the log-scale x-axis) can be seen to match the algorithm success rates (markers) measured in a large-scale experiment.

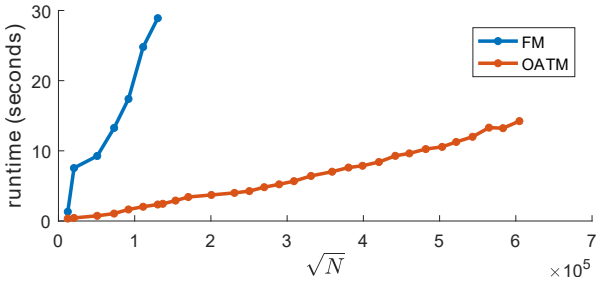


Figure 4. **Scalability experiment**. OATM is compared empirically to FM, over a 2D-affine search space of size N. As expected, the runtime of OATM grows linearly in $\sqrt{N}$, while that of FM is linear in N (notice the $\sqrt{N}$ x-axis).

However, this covering is sub-optimal by a multiplicative factor of $1.5\sqrt{3}$ due to the overlap of circles. We can actually get a practically optimal decomposition (while not strictly following the ϵ-net definition), as is depicted in Figure 2(b). We take the product of the sets: $F_\epsilon = \{i,j \ : \ i,j \in -\epsilon,\dots,\epsilon\}$ and $Net_\epsilon(F) = \{i,j \ : \ i,j \in \{\epsilon+2k\epsilon\}$ for $k = 1,\dots,\lfloor (n-m+1)/2\epsilon) \rfloor\}$. This results in $|F_\epsilon| = 4\epsilon^2$ and $|Net_\epsilon(F)| = (n-m+1)^2/(4\epsilon^2)$. Taking $\epsilon = 0.5\sqrt{n-m+1}$ yields $|F_\epsilon| = |Net_\epsilon(F)| = n-m+1$.

3. Empirical validation of the analysis

Algorithm success rate (2D-translation)

We begin with a large-scale validation of the theoretical guarantees of the algorithm (shown for the 2D-translation case), with each of the number k of iterations in the set $\{1,2,4,8,16,32,64,128,256,512\}$, while the other parameters are kept fixed.

We run 200 template matching trials for each inlier rate α in the set $\{0.3,0.4,0.5,0.6,0.7,0.8,0.9,1\}$. The success rate reported is the relative number of trials for which an exact match was found. For each trial we created a template matching instance, by first extracting a 100×100 template T from a 500×500 image I with grayscale intensities in [0,1], taken (scaled) at random from the Unplash data-set[2]. A random α-fraction of the template pixels are labeled as inlier pixels, and the intensity $T(p)$ of each outlier pixel p is replaced with the intensity that is 0.5 away from it in absolute difference. This setting guarantees that the resulting inlier rate is exactly α, and the algorithm succeeds only if it samples a pure set of inliers. Finally, we add to the image I white Gaussian noise with std equivalent of 5 greylevels.

The results are shown in Figure 3, where the empirical success rates per α (markers) can be seen to match the theoretical success rates from Equation (12) (solid curves). It is important to mention that these are minimal success rates guaranteed for finding the perfect match, which strictly hold, irrespective of the template and image contents, while in practice we often observe significantly better rates.

[2]A set of 65 high-res images we collected from https://unsplash.com/, which we present in the supplementary material [16].

Algorithm scalability (2D-affine)

In this experiment (result shown in Figure 4) we verify the argued $O(\sqrt{N})$ runtime of our algorithm. A simple way of doing so is by creating a sequence of affine matching instances (see the experiment in Section 4.1 for the technical details), were square templates of a fixed side length of 32 pixels are searched in square images with varying side lengths in the set $\{100,200,300,\dots,3200\}$, while keeping other affine search limits fixed - scales in the range $[2/3,3/2]$ and rotations in the range $[-\pi/4,\pi/4]$. This leads to a sequence of configuration sizes N that grows quadratically (hence the markers are distributed roughly linearly in the $\sqrt{N}$ x-axis). As can be seen, the runtime of OATM grows linearly with $\sqrt{N}$, and can handle in reasonable time a ratio of up to 100 between template and image dimensions. For reference, the complexity of the Fast-Match (FM) algorithm [17], representing the state-of-the-art in affine template matching, depends on a standard parameterization of the 2D-affine space (whose size grows linearly in N - see [17] for details). As can be seen, it cannot cope with template-image side length ratio of over 20.

4. Results

In this section we demonstrate the advantages of the proposed algorithm through several controlled and uncontrolled experiments on real data.

Implementation details The parameters used in our implementation were chosen by simple coordinate descent over a small set of random synthetic instances (generated as described in Sec. 4.1). For the random grid, we use sample dimension $\hat{d} = 9$; cell dimension $c = 2.5t$; where we take the threshold $t = 2\sigma\sqrt{2/\pi}$ (twice the mean of a zero-mean folded-normal-distribution), given a noise level of σ, or $t = 10$ greylevels when it is unknown. Our method can provide affine photometric invariance, i.e., global bright-

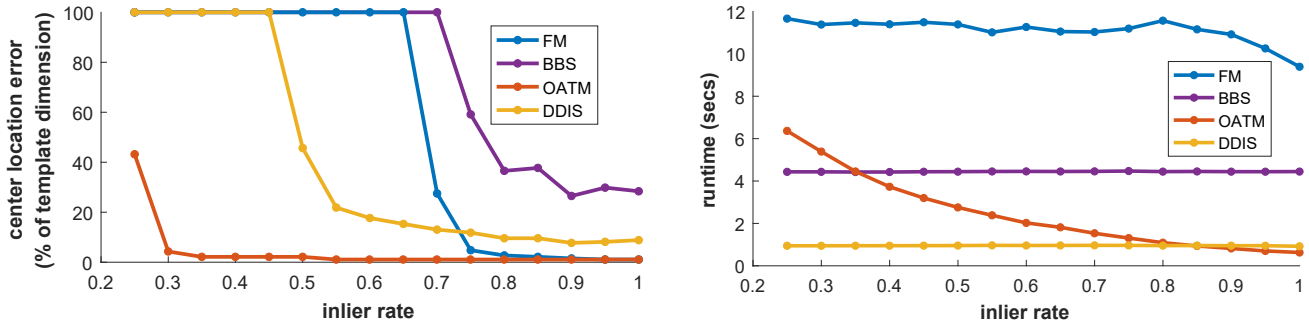


Figure 5. **Results of the occlusion experiment (Sec. 4.2):** median center location errors (**left**) and average run-times (**right**).

ness and contrast changes, by standardizing the vector sets U and V (in step 2 of Algorithm 2) to have the mean and standard deviation of the template.

4.1. Template matching evaluation

We test our algorithm in a standard template matching evaluation, not involving occlusions, in order to compare to other algorithms, such as Fast-Match (FM) [17] representing state-of-the-art in affine template matching. We run a large-scale comparison, using different combinations of template and image sizes (a larger gap between their sizes implies a larger size N of the search space). We will use the following shorthands for template and image dimensions: T1 for 16×16, T2 for 32×32 and T3 for 64×64. Likewise: I1 for 160×160, I2 for 320×320 and I3 for 640×640.

For each template-image size combination, we ran 100 random template matching trials. Each trial (following [17]) involves selecting a random image (here, from the Unplash data-set) and a random affine transformation (parallelogram in the image). The template is created by inverse-warping the parallelogram and white gaussian noise with 5 graylevels equivalent std is added to the image.

For each trial we report average overlap errors and run-times. The overlap error is a scalar in $[0,1]$ given by 1 minus the ratio between the intersection and union of the detected and true target parallelograms.

The results are summarized in Table 1. OATM is typically an order of magnitude faster than FM, at similar low error levels. FM cannot deal with the setting T1-I3, due to the large number of configurations N (the image edge length is 40 times the template edge length), while OATM deals with a more tolerable size of $\sqrt{N}$.

		template-image sizes								
		T1-I1	T1-I2	T1-I3	T2-I1	T2-I2	T2-I3	T3-I1	T3-I2	T3-I3
FM	err.	0.09	0.13	NA	0.05	0.05	0.09	0.02	0.01	0.03
	time	12.22	25.37	NA	4.35	7.78	32.07	1.33	1.90	11.61
OATM	err.	0.07	0.10	0.13	0.02	0.04	0.04	0.01	0.02	0.13
	time	0.15	0.18	0.39	0.53	0.76	1.73	0.51	0.64	1.01

Table 1. **Template matching evaluation** for different template image sizes, including average runtime (seconds) and overlap error.

4.2. Robustness to occlusions

In this experiment, we evaluate how well OATM and several other methods deal with occlusion. We repeat the protocol from the previous experiment (Section 4.1), except that we take a fixed template-image size (T2-I2) and we synthetically introduce a controlled amount of outlier pixels. One way of doing so (see examples in Figure 1) is by introducing random 4×4 blocks. We repeated the experiment with two other ways of introducing occlusion, resulting in similar results, which we provide in the supplementary material [16]. These come to show that our method is robust to the spatial arrangement of the occlusion mask.

In addition to Fast-Match (FM) [17], we compare with two additional template matching methods - Best Buddies Similarity (BBS) [8] and Deformable Diversity Similarity (DDIS) [25], both specialized in handling complex geometric deformations and high levels of occlusion. For a fair comparison, since BBS and DDIS match the template in a sliding window fashion (and account for deformation within the window), we measure center location errors (rather than overlap error) - the distance between the center of the target window and the true target center location, as a percentage of the template dimension (clipped at 100%).

The plots in Figure 5 summarize the experiment. OATM can be seen to provide the most accurate detections at a very wide range of inlier rates, starting from around 0.25. DDIS can handle inlier rates of above 0.5, but is slightly less accurate in localization due to its sliding window search. FM was not designed to handle occlusions explicitly and fails to do so for inlier rates under 0.75. BBS does not handle inlier rates under 0.75 and its localization is suboptimal when dealing with the affine deformations in this setting.

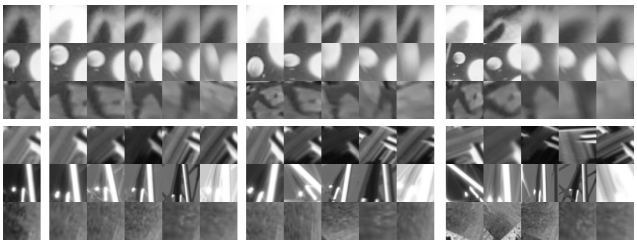
In terms of speed, DDIS is clearly the most efficient. DDIS and BBS are agnostic of the inlier rate, while the runtime of OATM is inverse proportional to the inlier rate, due to its RANSAC-like adaptive stopping criterion.

4.3. Matching partially occluded deformed patches

In this experiment we use the recent HPatches [4] data-set, which was designed for benchmarking modern local

image descriptors. The patches were extracted from 116 sequences (59 with changing viewpoint, 57 with changing illumination), each containing 6 images of a planar scene with known geometric correspondence given by a 2D homography. Approximately 1300 square 65×65 reference patches (rectified state-of-the-art affine detected regions) are extracted from the first image in each sequence. The exact set of corresponding patches were then extracted from the 5 other sequence images, using the ground-truth projection, while introducing 3 levels (Easy, Hard, Tough) of controlled geometric perturbation (rotation, anisotropic scaling and translation), to simulate the location inaccuracies of current feature detectors.

These perturbations introduce significant geometric deformations (e.g. rotation of up to $10°/20°/30°$) as well as increasing levels of occlusion (average overlap of $78\%/63\%/51\%$) for the Easy/Hard/Tough cases. Figure 6 shows several examples of extracted reference patches and their matching patches at the different levels of difficulty.



ref E1 E2 E3 E4 E5 H1 H2 H3 H4 H5 T1 T2 T3 T4 T5

Figure 6. **Samples from the HPatches [4] dataset.** *viewpoint* sequences (rows 1-3) and *illumination* sequences (rows 4-6).

This data is useful in showing the capabilities of our method in handling such challenges, in comparison with the common practice of matching features by their descriptors. We focus on the proposed 'matching' task [4], in which each reference patch needs to be located among each of the patches of each sequence image. A template matching algorithm cannot strictly follow the suggested task protocol, which was defined for matching patches by their descriptors. Instead, we pack all the ($\sim$1300) square target patches into a single image in which we search for the template using the photometric invariant version of OATM. The target patch chosen is the one which contains the center location of the warped template patch. For mean-Average-Precision (mAP) calculation, since our method only produces a single target patch we assign a weight of 1 to the detected target patch and 0 to the rest.

The results are summarized in Table 2. The reference descriptor methods include SIFT [19] and its variant RSIFT [3], the binary descriptors BRIEF [7] and ORB [23] and the deep descriptors DeepDesc (DDESC) [24] and TFeat ratio* (TF-R) [5]. For SIFT, TF-R, DDESC and RSIFT, results are given for the superior whitened and normalized versions of the descriptors (as reported in [4]).

	viewpoint seqs			**illumination seqs**		
method	Easy	Hard	Tough	Easy	Hard	Tough
BRIEF [7]	25.6	6.9	2.4	20.5	5.9	2.0
ORB [23]	36.4	11.1	3.7	28.9	8.8	3.2
SIFT [19]	59.4	30.6	15.3	52.6	26.1	13.3
TF-R [5]	58.9	35.5	19.0	48.5	28.6	15.6
DDESC [24]	58.6	36.0	20.2	50.7	30.0	17.0
RSIFT [3]	64.0	35.2	18.5	**57.1**	**30.2**	15.9
OATM	**72.7**	**49.2**	**32.1**	43.3	29.3	**19.7**

Table 2. **Results on the HPatches [4] Image Matching benchmark.** Results are in terms of mean-Average-Precision (mAP), where all results except that of OATM were reported in [4].

Clearly, for both viewpoint and illumination sequences - the mAP of OATM deteriorates more gracefully with the increase in geometric deformation and level of occlusion, compared to the descriptor based methods. While the state-of-the-art features and descriptors may be highly insensitive to certain local geometric deformations and different photometric variations (and hence some outperform OATM in the Easy illumination case), they are not as effective in dealing with significant deformation and occlusion, unlike OATM which explicitly explores the space of affine deformations and reasons about substantial occlusion levels.

Furthermore, the naive current application of OATM on this data suggests that performance could be further improved by: (i) finding a distribution over target locations rather than one single detection; (ii) being aware of the patch structure of the stacked target image; (iii) using advanced representations instead of the greylevel pixelwise description. That being said, unlike the descriptor based methods, the template matching nature of OATM is certainly not suitable for large-scale matching, where a large pool of patches needs to be matched against another. Nevertheless, many of the ideas presented here could be possibly adapted, e.g. to the image-to-image matching setup.

5. Conclusions

We have presented a highly efficient algorithm for 2D-affine template matching that is carefully analyzed and is shown to improve on previous methods in handling high levels of occlusion and geometric deformation.

The results on the HPatches data-set raise the question of whether descriptor based matching is able to handle the geometric deformations and high occlusion levels that are inherent in the localization noise introduced by feature detectors. This is the case even in the advent of deep learning, and the development of methods that can explicitly reason for deformation and occlusion seems to be necessary for improving the state-of-the-art in visual correspondence.

Acknowledgement Research supported by ARO W91 1NF-15-1-0564/66731-CS, ONR N00014-17-1-2072 and a gift from Northrop Grumman.

References

[1] D. Aiger, H. Kaplan, and M. Sharir. Reporting neighbors in high-dimensional euclidean space. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 784–803, 2013. 4

[2] D. Aiger, E. Kokiopoulou, and E. Rivlin. Random grids: Fast approximate nearest neighbors and range searching for image search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3471–3478, 2013. 1, 4

[3] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE, 2012. 8

[4] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 2, 7, 8

[5] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, volume 1, 2016. 2, 8

[6] P. Baque, F. Fleuret, and P. Fua. Deep occlusion reasoning for multi-camera multi-target detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010. 2, 8

[8] 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. 2, 7

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 2

[10] E. Elboer, M. Werman, and Y. Hel-Or. The generalized laplacian distance and its applications for visual matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2322, 2013. 2

[11] K. Fredriksson. *Rotation Invariant Template Matching*. PhD thesis, University of Helsinki, 2001. 2

[12] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015. 2

[13] Y. Hel-Or, H. Hel-Or, and E. David. Matching by tone mapping: Photometric invariant template matching. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):317–330, 2014. 2

[14] J. Hur and S. Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[15] N. Joshi, S. Avidan, W. Matusik, and D. J. Kriegman. Synthetic aperture tracking: tracking through occlusions. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 2

[16] S. Korman. Occlusion aware templae matching webpage. http://www.eng.tau.ac.il/~simonk/OATM. 5, 6, 7

[17] S. Korman, D. Reichman, G. Tsur, and S. Avidan. Fast-match: Fast affine template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2331–2338, 2013. 2, 6, 7

[18] F. Leone, L. Nelson, and R. Nottingham. The folded normal distribution. *Technometrics*, 3(4):543–550, 1961. 5

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 2, 8

[20] T. Nguyen, S. W. Chen, S. S. Shivakumar, C. J. Taylor, and V. Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *arXiv preprint arXiv:1709.03966*, 2017. 2

[21] E. Osherov and M. Lindenbaum. Increasing cnn robustness to occlusions by reducing filter support. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[22] W. Ouyang, F. Tombari, S. Mattoccia, L. Di Stefano, and W.-K. Cham. Performance evaluation of full search equivalent pattern matching algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):127–143, 2012. 2

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011. 2, 8

[24] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015. 2, 8

[25] 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. 2, 7

[26] D. Tsai and C. Chiang. Rotation-invariant pattern matching using wavelet decomposition. *Pattern Recognition Letters*, 23(1):191–201, 2002. 2

[27] Y. Yang, Z. Lu, and G. Sundaramoorthi. Coarse-to-fine region selection and matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5051–5059. IEEE, 2015. 2

[28] Y. Yang, G. Sundaramoorthi, and S. Soatto. Self-occlusions and disocclusions in causal video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4408–4416, 2015. 2

[29] C. Zhang and T. Akashi. Fast affine template matching over galois field. In *BMVC*, pages 121–1, 2015. 2

[30] T. Zhang, K. Jia, C. Xu, Y. Ma, and N. Ahuja. Partial occlusion handling for visual tracking via robust part matching. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1258–1265. IEEE, 2014. 2

[31] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision*, 111(2):171–190, 2015. 2