Efficient and Scalable 4-th order Match Propagation

David Ok, Renaud Marlet, Jean-Yves Audibert

Efficient and Scalable
4th-order Match Propagation

David Ok, Renaud Marlet, and Jean-Yves Audibert

Université Paris-Est, LIGM (UMR CNRS), Center for Visual Computing
École des Ponts ParisTech, 6-8 av. Blaise Pascal, 77455 Marne-la-Vallée, France

Abstract. We propose a robust method to match image feature points taking into account geometric consistency. It is a careful adaptation of the match propagation principle to 4th-order geometric constraints (match quadruple consistency). With our method, a set of matches is explained by a network of locally-similar affinities. This approach is useful when simple descriptor-based matching strategies fail, in particular for highly ambiguous data, e.g., with repetitive patterns or where texture is lacking. As it scales easily to hundreds of thousands of matches, it is also useful when denser point distributions are sought, e.g., for high-precision rigid model estimation. Experiments show that our method is competitive (efficient, scalable, accurate, robust) against state-of-the-art methods in deformable object matching, camera calibration and pattern detection.

1   Introduction

Establishing correspondences between sets of features detected in images arises in many vision tasks, e.g., object matching, camera calibration and pattern detection. In many cases, distinctive feature descriptors and simple matching strategies [1, 2] successfully produce a reasonably good set of matches w.r.t. the task requirements: large enough to carry meaningful information and with a large enough proportion of true positives (little contamination by false positives). But in ambiguous settings, e.g., when similar objects occur several times (e.g., windows on a facade, rocks in a landscape) or when distinctive textures are lacking, these matching strategies may fail and jeopardize the whole task. Yet, more robust correspondences can be found using the geometric consistency of feature location. Methods that try to address this issue fall into three main categories.

The first category includes RANSAC-based methods [3, 4], possibly in conjunction with Hough-based clustering [5]. They are fast and robust if the noise in the data can be estimated and if the percentage of inliers (i.e., true correspondences) among the set of candidate correspondences is of order 10% or greater. But this is hardly the case for pattern detection and ambiguous feature matching, where true correspondences can be less than 5%. Besides, the correspondences are explained by a number of independent homographies, i.e., disjoint planar facets [4, 5]. There is no relation among homographies other than looking for a totally new homography at the periphery of a previous one, which is inappropriate for curved surfaces and deformations.

In constrast, the second kind of approaches explicitly handles such cases. Correspondence selection is formulated there as a hypergraph matching problem that exploits geometric cues [6–11]. However, the algorithmic complexity can be prohibitive in practice: given n points, time $O(n^d \log n)$ has been reported for d-order potentials and after a number of approximations [9]. Such methods hardly scale to thousands of interest points, which would correspond to huge (gigabytes) affinity tensors, even after sparsification. Moreover, not all of them define how to discriminate inliers from outliers. Many hypergraph matchers only look for a bijection: a match is always found for any point, although dummy points can be added to attract outliers [12]. Some authors also use a threshold on the computed match confidence [12], but the confidence value is relative and cannot be easily associated to a geometric, understandable measure, leaving the user clueless for setting a sensible threshold value. Besides, looking for a single explanation of all correspondences may be an issue for scenes with moving objects.

Methods in the third category solve many local correspondence problems through simultaneous match propagation [13–15]: different seeds are grown and adapt to different transformations. However, these approaches basically exploit 2nd-order constraints and heavily depend on affine shape adaptation. They are thus not or poorly applicable to features that are not affine-covariant, such as DoG-SIFT [1]. Moreover, as shown by our experiments, affine shape determination is not very precise and shape adaptation can thus be significantly noisy. Even if optimized during propagation [14], affine shapes lack robustness. Some approaches also require the images to be available [13, 14], as opposed to only working on the set of abstract feature points. In addition, these methods cope with a reasonable amount of matching ambiguity, but fail to limit detection when the set of possible correspondences is strongly contaminated by outliers.

Our method tries to overcome the above drawbacks. It is a careful adaptation of the match propagation principle to 4th-order geometric constraints (macth quadruple consistency). Our framework explains a set of matches by a continuous network of locally-similar affinities which are determined from neighboring matches rather than by the affine shape of a single match. Our approach enjoys many good properties. It works on any kind of feature point (not only affine-covariant) and different types of features can even be freely mixed, for denser, more uniform or more precise correspondences. Besides, it does not require the image pixels after detection, contrary to most propagation based methods. Although it has no global view of all correspondences (contrary to non-approximating hypergraph matchers), it produces very reliable matches. It can tell inliers from outliers and it is robust to high outlier contamination rates. It adapts to scenes that have to be explained by different, separate models or by continuous model deformation. Last, it scales to hundreds of thousands of matches, both in time and space.

The rest of the paper is organized as follows. Section 2 states the optimization problem we try to solve. Section 3 and 4 present our algorithm and our pattern matcher. Section 5 evaluates our method for deformable object matching, camera calibration and pattern detection (accurate localization). Section 6 concludes.
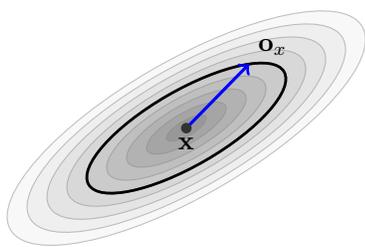
(1) the position $\mathbf{x} \in \mathbb{R}^2$ of the feature in the image, which we note by a font change only for readability,
(2) a shape $\mathcal{S}_x$ representing the (possibly anisotropic) scale of the feature, encoded by a scale matrix $\mathbf{\Sigma}_x \in \mathbb{R}^{2\times 2}$,
(3) an orientation $\mathbf{o}_x$,
(4) a feature descriptor $\mathbf{v}_x$.

Fig. 1: Geometric information of a feature.

2 Problem Statement

Before formulating the problem, we lay down a few definitions and notations. Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets of features extracted respectively from two images, and let $\mathcal{M} \subseteq \mathcal{X} \times \mathcal{Y}$ be a given set of possible matches. In the following, we denote a match by $m = (x, y)$. It is typically a pair of features whose descriptors are close, or close enough compared to other close descriptors. Note that $\mathcal{M}$ may include numerous ambiguities, i.e., any number of matches with the same feature x or y. ($\mathcal{M}$ can even be $\mathcal{X} \times \mathcal{Y}$.) A set of matches $R \subset \mathcal{M}$ is called a *region*.

2.1 Feature Information

The sets of features and matches can freely mix detectors and descriptors of different kinds, e.g., Harris-affine or Hessian-affine interest points [16], DoG-SIFT blobs and descriptors [1], MSER regions [17]. But a meaningful match can only involve a detector-descriptor pair of the same kind.

For each kind f of feature (a detector-descriptor pair), and each feature x of kind f, we assume that the information illustrated in Fig. 1 is available. Note that, while affine-covariant keypoint scales are elliptic, e.g., with a Harris-affine detector, others such as DoG-SIFT scales are isotropic, i.e., circular. The orientation is typically given by the dominant gradient direction around x at some appropriate scale. The feature descriptor abstracts the image around x, also at some appropriate scale, for comparison with other detected features.

2.2 Match Consistency Under Affinity Constraint

Feature information, besides descriptors, provides the ground for assessing the geometric consistency of a set of features. If x and y match, and if ϕ is a local affinity relating image 1 around x to image 2 then: the position $\phi(\mathbf{x})$ should be close to $\mathbf{y}$, taking scale into account; shape $\phi(\mathcal{S}_x)$ should be close to shape $\mathcal{S}_y$; and orientation $\phi(\mathbf{o}_x)$ should be close to orientation $\mathbf{o}_y$. Symmetrically, this should also be true of (y, x) for the inverse affinity ϕ^{-1}. We elaborate these notions.

"Being close" hinges on the specific characteristics of the kind of feature. We assume that each detector for a feature kind f comes with its *associated repeatability expectations*, that depend, e.g., on the detector precision and parameters
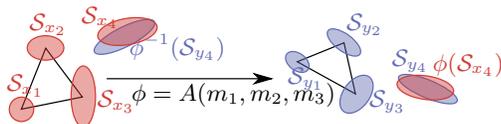
Fig. 2: Affine consistency of match m_4 w.r.t. $\phi = A(m_1, m_2, m_3)$.

or on the maximum expected change in images (viewpoint, illumination, etc.). Based on this knowledge, we can test position, shape and orientation consistency between two features $\phi(x)$ and y. Standard definitions include a threshold on the distance between the positions, possibly taking scale into account based on $\phi(\mathcal{S}_x)$ and/or $\mathcal{S}_y$. The *scale-sensitive distance* to x (resp. to y) is defined as:

$$d_x(\mathbf{x}') = (\mathbf{x}' - \mathbf{x})^T \boldsymbol{\Sigma}_x (\mathbf{x}' - \mathbf{x}) \tag{1}$$

Joint with a threshold on the Jaccard distance (overlap) between co-centered $\phi(\mathcal{S}_x)$ and $\mathcal{S}_y$, it is used to estimate detector repeatability [16].

It can be combined with an orientation angle threshold. Note that a typical threshold value for the Jaccard distance is 0.4 [18]. However, because we use a local affine approximation (see below), this threshold can be loosened, e.g., to 0.6. This prevents the unwanted, premature rejection of possible matches.

Assuming we know the expected precision of position δ_p, the expected precision of shape δ_s and the expected precision of orientation angle δ_o, we define predicate a $\mathbb{P}_f$ to test simultaneously position, shape and orientation consistency:

$$\mathbb{P}_f(x_1, x_2) = (d_{x_1}(\mathbf{x}_2) < \delta_\mathrm{p}) \wedge \left(1 - \frac{\mathrm{area}(\mathcal{S}_{x_1} \cap \mathcal{S}_{x_2})}{\mathrm{area}(\mathcal{S}_{x_1} \cup \mathcal{S}_{x_2})} < \delta_\mathrm{s}\right) \wedge (\mathbf{o}_{x_1}.\mathbf{o}_{x_2} > \cos(\delta_\mathrm{o})) \tag{2}$$

Predicate $\mathbb{P}_\phi$ assesses the consistency of a match under an affinity constraint ϕ:

$$\mathbb{P}_\phi(x, y) \stackrel{\mathrm{def}}{=} \mathbb{P}_f(\phi(x), y)) \wedge \mathbb{P}_f(x, \phi^{-1}(y)) \tag{3}$$

Namely, an f-match $m = (x, y)$ is *position-, shape- and orientation consistent w.r.t. affinity ϕ* iff $\mathbb{P}_f$ holds both for $(\phi(x), y)$ and $(x, \phi^{-1}(y))$.

Finally, we define a predicate $\mathbb{P}_d$ (used for match propagation) that tests relative, scaled distance consistency: two f-matches $m = (x, y)$ and $m' = (x', y')$ are *(scaled-)distance-consistent* iff the distance of x' to x, relative to the scale of x, is close enough to the distance of y' to y, relative to the scale of y:

$$\mathbb{P}_d(m, m') \stackrel{\mathrm{def}}{=} \frac{\min(d_x(\mathbf{x}'), d_y(\mathbf{y}'))}{\max(d_x(\mathbf{x}'), d_y(\mathbf{y}'))} > 1/2 \tag{4}$$

2.3 Region Consistency

The feature correspondence problem relies on two kinds of assumptions. First, if (x, y) is a good match, the image around x should be similar to the image around y. This photometric criterion translates into features having "close enough" descriptors. Second, given a set of matches $(x_i, y_i)_{1 \leq i \leq n}$, the relative

position of feature x_i w.r.t. others features $(x_j)_{j\neq i}$ is expected to be similar to the relative position of y_i w.r.t. other features $(y_j)_{j\neq i}$. This criterion is mainly geometric, i.e., based on the relative coordinates of the features. But it also has an indirect, photometric flavor as the feature shapes and orientations also have to agree when relating x_i and y_i in the context of $(x_j, y_j)_{j\neq i}$.

The geometric assumption only holds locally. Geometric consistency is thus only expected in independent image regions, i.e., separate sets of features. Accordingly, we define the consistency of a given single region as well as the region-wise consistency of a set of separate regions. We will actually be looking for a subpartition $\mathcal{R} = (R_i)_{1\leq i\leq n}$ of $\mathcal{M}$ such that each region R_i is affine-consistent. (A set of regions $\mathcal{R}$ is a *subpartition* of $\mathcal{M}$ iff it is a partition of a subset of $\mathcal{M}$.)

Local Affinity. A consistent region can be defined as a set of matches locally related by an affine homography [16]. We actually do not define a consistent region by a single affinity but by many. This particular setting provides a valuable flexibility allowing a region to adapt to substantial non-affine transformations (cf. §5.1). Given a triple of matches $(m_i)_{1\leq i\leq 3} = (x_i, y_i)_{1\leq i\leq 3}$, one can construct a unique affine transformation $\phi = A((m_i)_{1\leq i\leq 3})$ between images 1 and 2 that maps $\mathbf{x}_i$ to $\mathbf{y}_i$ for all $1 \leq i \leq 3$. This only makes sense if the positions are not *degenerate*, i.e., if the points are not aligned, and more generally if the triangles corresponding to the feature triples in both images do not have too sharp angles. Now the affine-consistency of a given match m can be defined as the conjunction of the position, shape and orientation consistency of m w.r.t. affinity ϕ. Specifically, we say that m is *affine-consistent* with matches $(m_i)_{1\leq i\leq 3}$ iff $(m_i)_{1\leq i\leq 3}$ is not degenerate and $\mathbb{P}_\phi(m)$ holds for $\phi = A((m_i)_{1\leq i\leq 3})$. Fig. 2 illustrates this concept.

Region Affine-Consistency. As 3 matches can always be related by an affinity, region affine-consistency makes sense for at least 4 matches. It is our 4th-order constraint. A quadruple of matches $(m_i)_{1\leq i\leq 4}$ is *affine-consistent* iff for all $1 \leq i \leq 4$, m_i is affine-consistent with $(m_j)_{1\leq j\leq 4, j\neq i}$. This is extended to a region using chains of affine-consistent quadruples. First, given a region $R \subset \mathcal{M}$, a pair of matches $m, m' \in R$ are said *affine-consistent in R* iff there exists a sequence $(m_{1,i}, m_{2,i}, m_{3,i}, m_{4,i})_{1\leq i\leq n}$ of affine-consistent quadruples in R from m to m', i.e., s.t. $m = m_{1,1}$, $m_{4,i} = m_{1,i+1}$ for $1 \leq i < n$, and $m_{4,n} = m'$. Then, a region R is said *affine-consistent* iff any different matches $m, m' \in R$ are affine-consistent. By extension, a subpartition $\mathcal{R} = (R_i)_{1\leq i\leq r}$ of $\mathcal{M}$ is said affine-consistent iff each region R_i is affine-consistent. An important property is that, given two affine-consistent regions R, R' such that $R \cap R' \neq \varnothing$, $R \cup R'$ is affine-consistent.

Maximal Consistency. Finally, we are interested in finding a maximum number of meaningful matches; the actual number of underlying regions does not matter.

Given a set of regions $\mathcal{R}$, we thus define the *size of $\mathcal{R}$*, noted $\|\mathcal{R}\|$, as the number of matches occurring in $\mathcal{R}$, i.e., $\|\mathcal{R}\| = |\bigcup_{R\in\mathcal{R}} R|$. If $\mathcal{R}$ is a subpartition of $\mathcal{M}$, this reduces to $\|\mathcal{R}\| = \sum_{R\in\mathcal{R}} |R|$. If we can additionally impose that

---

Algorithm 1 Region growing from a seed match m_1.

Notations:
- $\mathcal{N}_K(m)$: K nearest matches of m that are scaled-distance consistent.
- C: matches that are scaled-distance consistent w.r.t. at least 1 match in R.
  When C is modified, it is always kept sorted by increasing distrust score.

1: **procedure** GROWREGION(m_1, K)
2:     Pick matches $m_2, m_3 \in \mathcal{N}_K(m_1)$
3:     $R \leftarrow \{m_i\}_{1 \le i \le 3}$ ▷ *Initialize R with seed*
4:     $C \leftarrow \bigcup_{1 \le i \le 3} \mathcal{N}_K(m_i) \setminus R$ ▷ *Initialize C*
5:     **while** $\exists\, (m, m', m'', m''') \in C \times R^3$ affine-consistent **do** ▷ LOCALSEARCH *(algo 2)*
6:         $R \leftarrow R \cup \{m\}$ ▷ *Grow R*
7:         $C \leftarrow C \cup (\mathcal{N}_K(m) \setminus R)$ ▷ *Ensure that matches in R are excluded*
8:     **end while**
9:     Return R
10: **end procedure**

---

the number $|\mathcal{R}|$ of underlying regions be minimal, the subpartition of $\mathcal{M}$ into affine-consistent regions with maximum size is actually unique (if any).

Our feature matching problem can now be stated: *Find the affine-consistent subpartition $\mathcal{R}$ of $\mathcal{M}$ of maximum size, then minimum cardinality.*

Ambiguity Freedom. Different tasks have different requirements regarding match ambiguity. For instance, whereas repeated pattern detection overtly calls for ambiguous matches, scene tracks used for estimating camera calibration parameters require unambiguous matches. A variant of our feature matching problem additionally require ambiguity-freedom, at the region or subpartition level. Note that uniqueness is not guaranteed in this case. More formally, a match (x, y) is *unambiguous in M* iff for all $(x', y') \in M \setminus \{(x, y)\}$, $x \neq x'$ and $y \neq y'$; a region R is *ambiguity-free* iff for any match $m \in R$, m is unambiguous in R; and a set of regions $\mathcal{R}$ is *ambiguity-free* iff R is ambiguity-free for all $R \in \mathcal{R}$.

3   Match Propagation Procedure

Due to the highly combinatorial nature of this optimization problem, we propose an algorithm and a set of heuristics that efficiently determine an affine-consistent subpartition of $\mathcal{M}$ of large size, and small cardinality, possibly ambiguity-free. Although we do not satisfy the global extremality constraints, our experiments show that our local maxima yield very good sets of matches (cf. §5).

The algorithm follows a region growing scheme. Given an initial region consisting of a triple of potential matches, we iteratively add more matches into the region provided they are geometrically consistent with some triple of matches already in the region. When no more match can be added, the region is considered as valid iff it is large enough. More regions can be grown by re-running the algorithm on the remaining potential matches. See Algorithm 1 for details.

Besides, if unambiguity is required, any match (x, y) is checked for ambiguity before being added to a growing region R. If there already is a match (x, y') or (x', y) in R, then (x, y) is removed from the remaining potential matches and associated to R, but without contributing to $|R|$.

The key ingredients of the algorithm are additional heuristics for growing the regions, that prevent a combinatorial explosion and only explore a limited number of pertinent cases, most likely matches being tried first. They enable a selective evaluation of concistency checks, in particular the shape consistency which can be computationally intensive. They are presented in the following.

Ordering and Limiting Potential Matches. Matches (x, y) are ordered by increasing distrust score, defined as follows. Let D be a distance in the descriptor space, e.g., Euclidean distance for SIFT. For a descriptor $\mathbf{v}_x \in \mathcal{V}_\mathcal{X}$, let $\mathbf{v}_y^1, \mathbf{v}_y^2 \in \mathcal{V}_\mathcal{Y}$ be respectively its nearest neighbor (1-NN) and its second nearest neighbor (2-NN). The distrust score (or Lowe score [1]) of match $m = (x, y)$ is defined as

$$L_{\mathcal{X} \to \mathcal{Y}}(x, y) = \frac{D(\mathbf{v}_x, \mathbf{v}_y^1)}{D(\mathbf{v}_x, \mathbf{v}_y^2)} \quad \leq 1 \tag{5}$$

The smaller the score $L_{\mathcal{X} \to \mathcal{Y}}(m)$ is, the less ambiguous match m is. Usually, a set of reliable matches is obtained with matches m such that $L_{\mathcal{X} \to \mathcal{Y}}(m) \leq \ell$. Typically, ℓ ranges in $[0.6; 0.8]$. However, doing so discards ambiguous matches. To avoid it, the distrust score is extended as follows:

$$L_{\mathcal{X} \to \mathcal{Y}}(x, y) = \begin{cases} \frac{D(\mathbf{v}_x, \mathbf{v}_y)}{D(\mathbf{v}_x, \mathbf{v}_y^2)} & \text{if } \mathbf{v}_y = \mathbf{v}_y^1 \quad \leq 1 \\ \frac{D(\mathbf{v}_x, \mathbf{v}_y)}{D(\mathbf{v}_x, \mathbf{v}_y^1)} & \text{if } \mathbf{v}_y \neq \mathbf{v}_y^1 \quad \geq 1 \end{cases} \tag{6}$$

$$L(m) = \min \left(L_{\mathcal{X} \to \mathcal{Y}}(m) , L_{\mathcal{Y} \to \mathcal{X}}(m) \right). \tag{7}$$

$L_{\mathcal{X} \to \mathcal{Y}}(x, y)$ quantifies an ambiguous match (x, y) by the relative proximity of $\mathbf{v}_y$ with respect to its 1-NN. $L(m)$ makes the distrust score symmetric. Note that using max rather than min would delay too much the analysis of 1-to-many ambiguities. In our work, $\mathcal{M}$ is the set of matches m such that $L(m) \leq \ell$, where ℓ can be greater than 1. Consequently, $\mathcal{M}$ is much more ubiquitous than with the usual Lowe criterion, for a better support of repetitive patterns.

Local Search for Region Growing When trying to grow a region R with a match $m = (x, y) \in C$ (line 5 of Algorithm 1), we prune the search of a triple of matches (m', m'', m''') by considering only close matches, i.e., matches $(x', y') \in R$ such that x' is among the k nearest neighbors of x, or y' is among the k nearest neighbors of y. Specifically, line 5 of Algorithm 1 actually calls Algorithm 2 to find a triple of matches that provides affine consistency to candidate match m.

Sidedness Constraint. We also introduce a sidedness constraint that, experimentally, is very efficient in pruning the search and more efficient than the one in [14].

Algorithm 2 Local search for region growing.

Notation: $\mathcal{W}_k(m)$: neighborhood of m containing k nearest matches in R.

```
 1: procedure LOCALSEARCH(R, C)
 2:     for m = (x, y) ∈ C do
 3:         find its nearest match m' ∈ R
 4:         for (m'', m''') ∈ Wₖ(m') × Wₖ(m') do
 5:             return (m, m', m'', m''') if affine-consistent
 6:         end for
 7:     end for
 8:     return ∅
 9: end procedure
```

The general idea is that if $m_1 = (x_1, y_1)$ and $m_2 = (x_2, y_2)$ are good matches, then the directed lines $\overrightarrow{\mathbf{x_1 x_2}}$ and $\overrightarrow{\mathbf{y_1 y_2}}$ should define corresponding half spaces. More formally, given two points $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, the half space on the left of $\overrightarrow{\mathbf{uv}}$ is $E(\mathbf{u}, \mathbf{v}) = \{\mathbf{w} \in \mathbb{R}^2 \mid \det(\mathbf{v} - \mathbf{u}, \mathbf{w} - \mathbf{u}) > 0\}$. A match (x, y) is *side-consistent* w.r.t. matches $(x_1, y_1), (x_2, y_2)$ iff $\mathbf{x} \in E(\mathbf{x_1}, \mathbf{x_2}) \Leftrightarrow \mathbf{y} \in E(\mathbf{y_1}, \mathbf{y_2})$. When evaluating a match candidate m for growing a region R, m can be excluded if there are $m_1, m_2 \in R$ such that m is not side-consistent w.r.t. matches m_1, m_2.

For robustness, the sidedness consistency applies only to matches (x, y) such that $\mathbf{x}$ (resp. $\mathbf{y}$) is not to close to line $\overrightarrow{\mathbf{x_1 x_2}}$ (resp. $\overrightarrow{\mathbf{y_1 y_2}}$). This prevents spurious match rejections caused by non-affine transformations or due to the imprecision of feature localization. For efficiency, we limit consistency checks for a region $R = (x_i, y_i)_{1 \le i \le n}$ to the contour edges of the convex hulls associated respectively to $(\mathbf{x}_i)_{1 \le i \le n}$ and $(\mathbf{y}_i)_{1 \le i \le n}$. We also impose that at any step of the region growing, the contour vertices of the convex hull of the points already matched in $\mathcal{X}$ should correspond to the contour vertices of the convex hull of the points already matched in $\mathcal{Y}$. The sidedness-checking procedure in [14] operates over all pairs of matches in a given region R, and thus performs $O(|R|^2)$ line checks. Our sidedness check operates only on the perimeter of R, rather than the whole area. The number of line checks is thus linear in the number of vertices on the contour of the convex hull, which is in practice $O(\sqrt{|R|})$.

4   Repetitive Pattern Search

Our feature matching algorithm can easily be turned into a pattern matcher. Given a object model M_0 defined by a geometric region I_0 in some input image I, the goal is to retrieve all objects that are similar to M_0 in some image J (possibly equal to I), i.e., to find image regions in J that are similar to I_0. We consider the case where I_0 is defined as the interior of a polygon P_0.

For this, we define $\mathcal{X}_0$ as the set of features inside polygon P_0 in I and $\mathcal{Y}$ as the set of features in J not in $\mathcal{X}_0$. We then grow regions of $\mathcal{M} \subset \mathcal{X}_0 \times \mathcal{Y}$ as described above, allowing ambiguity on $\mathcal{X}_0$. The resulting set of regions $\mathcal{R} = (R_i)_{1 \le i \le n}$ corresponds to discovered pattern instances. The image region in J

corresponding to a set of matches R_i can be retrieved by assuming local affinity transformations from I to J. More formally, given a vertex $\mathbf{u} \in \mathbb{R}^2$ of polygon P_0 in I, let x_1, x_2, x_3 be the geometrically closest 3 features in I such that there are matches $(m_j)_{1 \le j \le 3} = (x_j, y_j)_{1 \le j \le 3} \in R_i$. Then the corresponding polygon vertex in image J is $A(m_1, m_2, m_3)(\mathbf{u})$. The polygon P_i formed by such vertices defines an image region J_i of J that delineates the matched object M_i.

More pattern instances can be found by removing features in $\mathcal{R}$ from $\mathcal{Y}$ and reusing recursively image regions $(J_i)_{1 \le i \le n}$ as new input patterns, until no new pattern instance is found. To reduce the risk of pattern drifting, the patterns have to be explored in breadth-first search.

5   Results

We used the same parameters for *all* our experiments, which indicates the stability of our method. The region growing parameters defined in §3 are defined as $K = 80$ and $k = 10$. A region R is deemed valid iff $|R| \ge 7$. In the reported experiments, we processed on average $N = 5000$ points per image (sometimes tens of thousands) and 15 matches per point, i.e., $|\mathcal{M}| = 75,000$ on average. The number of matches per point, up to 650 in our examples, depends on the ambiguity of the descriptor value. A complete region-growing trial can take up to 4 seconds, for a very large and dense region. For deformable object matching and calibration, we performed 1000 attempts to grow regions; for pattern detection, all possible seeds were explored.

5.1   Deformable Object Matching

We evaluated our method on deformable object matching using the ETHZ Toys dataset (40 images of 9 models, and 23 test images), testing each model image against each test image. We compared with Ferrari et al. [14], Kannala et al. [13] and Cho et al. [15], as reported in their papers. For a fair comparison, we used MSER and Harris-affine features with SIFT descriptors, like [15]. Methods [13, 14] additionally use color information and dense photometric information.

Performance is reported in the ROC curve on the left part of Fig. 3, which depicts the detection rate versus false positive rate, letting a detection threshold
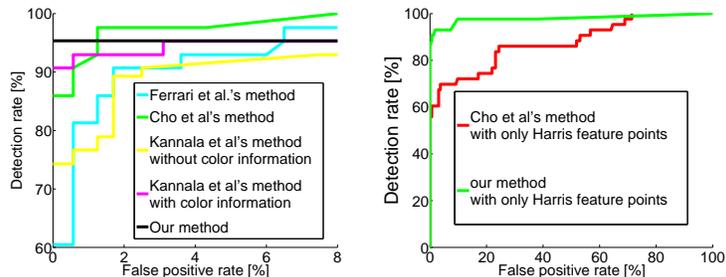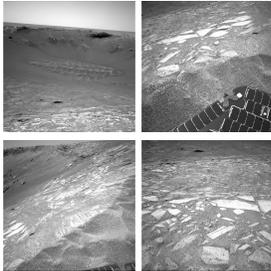

Fig. 3: ROC curves on the ETHZ Toys dataset.

Methods	Cameras	Match Time
Ours	**20/31**	60 mn
[1] + [3]	5/31	**5** mn
[15]	7/31	2880 mn
[14]	2/31	540 mn

Table 1: Some images of the *Books* dataset and calibration results.

Methods	# Cams	MSRE	# Tracks
Ours	**60/60**	5.00×10^{-2}	**75, 966**
[1] ($\ell = 0.3$) + [3]	22/60	**2.00×10^{-2}**	3, 266
[1] ($\ell = 0.4$) + [3]	30/60	3.13×10^{-2}	5, 598
[1] ($\ell = 0.5$) + [3]	**33/60**	47.50×10^{-2}	1, 131
[1] ($\ell = 0.6$) + [3]	28/60	5.68×10^{-2}	6, 378
[1] ($\ell = 0.7$) + [3]	28/60	6.47×10^{-2}	6, 533
[1] ($\ell = 0.8$) + [3]	28/60	8.27×10^{-2}	**6, 667**
[1] ($\ell = 0.9$) + [3]	28/60	8.84×10^{-2}	6, 564

Table 2: Some images of the *Mars* dataset and calibration results.

vary. (An object is considered as detected if the number of produced matches, summed over all its model views, exceeds this threshold.) Our method outperforms others, except for high false positive rate. This makes our method attractive for object matching tasks that tolerate only few wrong detections.

We performed a second experiment with the same dataset and the same parameters as Cho et al. [15], but only considering Harris affine features, which are reported to be among the most ambiguous affine-covariant features [18]. The right part of Fig. 3 confirms that our method is less prone to false detection, as it outperforms Cho et al.'s method both for low and high false positive rates.

5.2   Accurate and Scalable Matching for Camera Calibration

We tested a calibration task using Bundler [19] as a black-box taking as input a set of matches. We used two pathological datasets, which are hard to calibrate: *Books* (31 images) and *Mars* (60 images)[1]. In *Books* (cf. Table 1), matching ambiguities arises from the uniform background and the chair, as well as the repeated letters on the covers. We calibrate (here with Harris-affine features) many more cameras than Ferrari et al.'s [14], Cho et al.'s [15], and a baseline consisting in a Lowe criterion [1] followed by a RANSAC filter [3] estimating the fundamental matrix. In *Mars* (cf. Table 2), the landscape is very flat and the numerous rocks create ambiguous matches. Yet all 60 cameras are calibrated successfully with our method (with DoG features), contrary to RANSAC, which only calibrates half of the cameras. The mean squared reprojection error (MSRE, in pixels) and the number of consistent scene tracks also compare favorably. Our implementation has actually been used in the 3D-reconstruction chain of the

---

[1] PRoVisG Mars 3D Challenge, http://cmp.felk.cvut.cz/mars/

$|\mathcal{M}|$	3,000		10,000		30,000		100,000	
DoG	2,676	0.21 s	5,342	0.42 s	7,027	0.70 s	7,027	1.36 s
MSER	1,585	0.84 s	2,283	1.11 s	2,283	1.46 s	2,283	1.83 s
Hessian	2,190	1.71 s	5,054	3.02 s	5,922	3.35 s	5,922	3.99 s
Harris	2,178	1.59 s	6,250	3.62 s	10,273	3.58 s	10,623	4.01 s

Table 3: For a given number $|\mathcal{M}|$ of potential matches, number N of corresponding features and average running time, on all image pairs of [18]'s dataset.

winners of the *PRoVisG Mars 3D Challenge 2011*, from which this dataset is extracted. For this dataset, all 1770 possible image pairs are considered in 3.5 hours using parallelization on a 8-core CPU Xeon 2.8GHz machine.

We also compared with a method for tensor-based, 3rd-order hypergraph matching [9], with image 1 and 4 of the *graffiti* dataset used in [18], where the ground truth homography $\mathbf{H}$ is known. DoG features were detected and described with the SIFT descriptor. We evaluated the accuracy a, i.e., the proportion of actually correct matches among produced ones, as a function of the number N_f of features to match. To enable comparison, we experimented with various feature sets such that $|\mathcal{X}| = |\mathcal{Y}| = N_f$ and there is a bijection between $\mathcal{X}$ and $\mathcal{Y}$ such that for each $x \in \mathcal{X}$, there is a unique $y \in \mathcal{Y}$ satisfying $\|\mathbf{Hx} - \mathbf{y}\| \leq 5$ pixels, and likewise when permuting $\mathcal{X}$ and $\mathcal{Y}$. For bare TM (3rd-order affinities only), performance is poor: $a \leq 0.80$ for $N_f \leq 20$ and $a \leq 0.05$ for $N_f \geq 30$. Adding 1st-order SIFT descriptors to 3rd-order affinities improves it: $0.75 \leq a \leq 0.85$ for $N_f \leq 200$. But our method achieves better results: $0.95 \leq a$ for $N_f \leq 200$.

Although our theoretical complexity is $O(BN^2 \log N + \log |\mathcal{M}|)$, where N is the number of features and B the maximum degree of ambiguity of matches in $\mathcal{M}$, it is less than quadratic in N in practice, as illustrated in Table 3 on [18]'s dataset. (DoG is faster as it requires no ellipse intersection computation.) It is better, e.g., than tensor-based matching [9], which would be here $O(N^3 \log N)$ or $O(N^4 \log N)$, or agglomerative clustering [15], which is at least $O(|\mathcal{M}|^2)$.

5.3 Accurate Pattern Localization: Window Detection

We experimented with pattern detection, looking for windows in building facades. Although this problem has already been attacked [20–23], *accurate* localization has been treated very little for unrectified images. Window localization is challenging because of the wide range of appearance variety, the lack of texture, and the illumination variations. Unrectified images adds up to these challenges. Windows are then related by homographies or affinities: they may vary in size and shape, and it is difficult to detect small windows with almost no texture.

We used *eTRIMS* [24] for evaluation, which displays many different architectural and building styles, with annotations for windows. We selected the 45 images having at least 6 windows. For each image, we indicated seed windows manually and we generated rectified images for comparison purposes. In the case where images are rectified, they are indicated either manually or by a trained cascade classifier (CC) [25]. Then our pattern search retrieves missing windows.

		Rectified images		Unrectified Images	
Methods	TPR	TNR	TPR	TNR
Manual+ours	75%	96%	71%	98%
CC+ours	60%	93%	N/A	N/A
CC	46%	96%	N/A	N/A

Table 4: Example image and results on the *eTRIMS* dataset. (CC) is the cascade classifier run solely to detect windows. (Manual+ours) and (CC+ours) are methods where input window quadrilaterals are respectively provided manually and by the classifier (CC) combined with our method to retrieves missing windows.

To apply our repetitive pattern search, DoG, Harris-Affine and MSER features are extracted in each image and described by the SIFT descriptor. We only keep matches whose distrust score is less than 1.2, i.e., matches within 20% of the best match (description-wise). The bounding box of the pattern windows is dilated by 15% before search, to include some surrounding information, and shrunk back when instances are found to estimate the window region accurately.

The methods are compared in terms of mean true positive rate ($\overline{\text{TPR}}$) and mean true negative rate ($\overline{\text{TNR}}$), which should ideally be close to 1. Results are reported in Table 4. In case the image is not rectified, the $\overline{\text{TPR}}$, loses 4 points w.r.t. the rectified case. This slight degradation is chiefly due to estimation errors of the geometric transformation between the matched patterns. Shift and size errors between the geometric region of the detected pattern and the estimated image region also accumulates. Still, our method achieves a very low $\overline{\text{FPR}}$ of 2%.

5.4   Affinity Estimation: Triple vs Single Match

Finally, as discussed in the introduction, we evaluate the interest of match triples (m, m', m'') to construct accurate and robust affinities vs resorting to single matches $m = (x, y)$, using the shapes $(\mathcal{S}_x, \mathcal{S}_y)$ and orientation $(\mathbf{o}_x, \mathbf{o}_y)$. Experiments with Mikolajczyk et al.'s dataset [18] demonstrate that our region growing process performs consistently and significantly better when affinities ϕ are estimated with match triples. Each dataset consists of 6 images. For each dataset and for a given kind of feature f, we extract all feature points of type f. We match image 1 to images 2–5. Initial f-matches are obtained and ranked with Lowe's criterion. The distrust threshold is set to $\ell = 1$. On average, our region growing deals with $7,000$ to $28,000$ f-matches with an outlier proportion of at least 75%. We compare the performance of our region growing in terms of precision for both variants: triples and single matches (see Fig. 4).

Precision rates for triples are consistently better. We give two explanations. First, orientation estimation is often unstable; it remains sensitive to illuminations changes, blurring and compression. Second, local affinities estimated from the shape of DoG features are unsurprisingly inaccurate and consequently produces worse precision rates in general. Even when elliptic features are used, affinities estimated from triples still produce much better results in many cases.
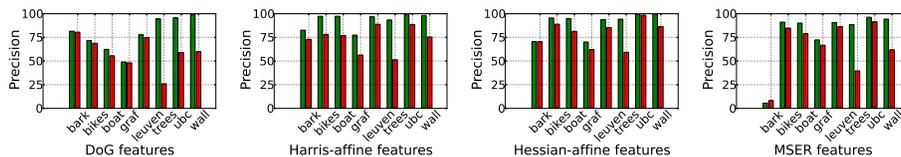
Fig. 4: Precision (%) of region growing on Mikolajczyk et al's. dataset (images 1-3). Affinities are computed from match triples (green) or single matches (red).

6 Conclusion

We have proposed a feature matching method that enforces photometric and geometric consistency. As illustrated by our experiments, it is efficient, scalable, accurate and robust, even in the presence of high ambiguity, improving over other existing methods. This allows applications in repetitive pattern detection.

Our approach belongs to the region-growing/match-propagation family [26]. Although it uses known ideas for matching under affinity constraint [18], it includes original ingredients and, as a whole, provides a unique blend. Our propagation is based on local affinities like [13–15], not pixel adjacency [26, 27], flow [26] or similitudes [28]. Our affinities are computed from match triples (any kind of feature points, possibly in combination), not necessarily affine correspondences [14, 15], 2nd moment matrix plus gradient orientation [13], or patch transformations [26, 28]. Our affinity constraint is 4th-order and sensitive to feature scale, not 2nd-order [6, 8, 15], 3rd-order [10] and photometric [9], or 4th-order reduced to points [7, 10]. For precision and robustness, each match of our growing regions selects nearby scale-consistent candidates; each candidate (best first) then looks for a nearby consistent triple in the region. It is simpler that the expansion-contraction phases of [14]. In [13], a region point only defines a single affinity to select admissible candidates, while in [15], growing is via agglomerative clustering. Our propagation is isotropic, image-order insensitive, scale-invariant and adapts to varying detection density like [15], contrary to fixed-size grid in model image [14], fixed-size pixel neighborhood [13, 26, 27] or reference image [28]. We are purely based on features, like [15], rather than photometric similarity. We do not require images (pixels) after feature detection, unlike [13, 14, 26–28], nor a regular flow of images [27] or epipolarly rectified image pairs [27]. All these characteristics are crucial for robustness and precision in difficult settings.

Acknowledgements. This work is part of IMAGINE, a joint research project between Ecole des Ponts ParisTech (ENPC) and CSTB.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
2. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010) 815–830
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24** (1981) 381–395

4. Pritchett, P., Zisserman, A., Zisserman, A.: Wide baseline stereo matching. In: ICCV. (1998) 754–760
5. Brown, M., Lowe, D.: Invariant features from interest point groups. In: BMVC. (2002) 656–665
6. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: ICCV. (2005) 1482–1489
7. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR. (2008)
8. Choi, O., Kweon, I.S.: Robust feature point matching by preserving local geometric consistency. Comput. Vis. Image Underst. **113** (2009) 726–742
9. Duchenne, O., Bach, F., Kweon, I.S., Ponce, J.: A tensor-based algorithm for high-order graph matching. IEEE Trans. PAMI **33** (2011) 2383–2395
10. Chertok, M., Keller, Y.: Efficient high order matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 2205–2215
11. Cho, M., Lee, J., Lee, K.: Reweighted random walks for graph matching. In: ECCV. (2010) V: 492–505
12. Zheng, Y., Doermann, D.: Robust point matching for nonrigid shapes by preserving local neighborhood structures. Tr. PAMI **28** (2006)
13. Kannala, J., Rahtu, E., Brandt, S., Heikkila, J.: Object recognition and segmentation by non-rigid quasi-dense matching. In: CVPR. (2008) 1–8
14. Ferrari, V., Tuytelaars, T., Gool, L.J.V.: Simultaneous object recognition and segmentation by image exploration. In: ECCV (1). (2004) 40–54
15. Cho, M., Lee, J., Lee, K.M.: Feature correspondence and deformable object matching via agglomerative correspondence clustering. In: ICCV. (2009)
16. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: ECCV (I). (2002) 128–142
17. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC. (2002) 384–393
18. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV **65** (2005) 43–72 . Dataset: http://www.robots.ox.ac.uk/~vgg/research/affine/.
19. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from internet photo collections. Int. J. Comput. Vision **80** (2008) 189–210
20. Lee, S.C., Nevatia, R.: Extraction and integration of window in a 3D building model from ground view image. In: CVPR (2). (2004) 113–120
21. Ali, H., Seifert, C., Jindal, N., Paletta, L., Paar, G.: Window detection in facades. In: ICIAP. (2007) 837–842
22. Haugeard, J.E., Philipp-Foliguet, S., Precioso, F.: Windows and facades retrieval using similarity on graph of contours. In: ICIP. (2009) 269–272
23. Recky, M., Leberl, F.: Windows detection using k-means in cie-lab color space. In: ICPR. (2010) 356–359
24. Korč, F., Förstner, W.: eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, University of Bonn (2009)
25. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV **57** (2004) 137–154
26. Lhuillier, M., Quan, L.: Match propagation for image-based modeling and rendering. Tr. PAMI **24** (2002) 1140–1146
27. Cech, J., Sanchez-Riera, J., Horaud, R.: Scene flow estimation by growing correspondence seeds. In: CVPR. (2011) 3129–3136
28. HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D.: Non-rigid dense correspondence with applications for image enhancement. SIGGRAPH **30** (2011)