Published in Image Processing On Line on 2018–09–24.
Submitted on 2018–03–27, accepted on 2018–08–01.
ISSN 2105–1232 © 2018 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2018.225

Fast Affine Invariant Image Matching

Mariano Rodríguez[1], Julie Delon[2], Jean-Michel Morel[1]

[1] CMLA, ENS Paris-Saclay, France (rdguez.mariano@gmail.com, moreljeanmichel@gmail.com)
[2] MAP5, University Paris Descartes, France (julie.delon@parisdescartes.fr)

Abstract

Methods performing Image Matching by Affine Simulation (IMAS) attain affine invariance by applying a finite set of affine transforms to the images before comparing them with a Scale Invariant Image Matching (SIIM) method like SIFT or SURF. We describe here how to optimize IMAS methods. First, we detail an algorithm computing a minimal discrete set of affine transforms to be applied to each image before comparison. It yields a full practical affine invariance at the lowest computational cost. The matching complexity of current IMAS algorithms is divided by about 4. Our approach also associates to each image an affine invariant set of descriptors, which is twice smaller than the set of descriptors usually used in IMAS methods, and only 6.4 times larger than the set of similarity invariant descriptors of SIIM methods. In order to reduce the number of false matches, which are inherently more frequent in IMAS approaches than in SIIM, we introduce the notion of hyper-descriptor, which groups descriptors whose keypoints are spatially close. Finally, we also propose a matching criterion allowing each keypoint of the query image to be matched with several keypoints of the target image, in order to deal with situations where an object is repeated several times in the target image.

Source Code

The source code to reproduce the same results as the demo can be found on the IPOL web page of this article[1]. Compilation and usage instructions are included in the `README.md` file of the archive. Complementary information is available at https://rdguez-mariano.github.io/pages/hyperdescriptors.

Keywords: affine invariance; IMAS; ASIFT; FAIR-SURF; SIFT; SURF

Disclaimer

The present work publishes: near optimal versions of IMAS methods [23]; IMAS with hyper-descriptors; and two structural and computational improvements. It does not publish the ORSA method [16, 14, 15] nor the USAC method [22]. They may be updated or replaced by other subroutines.

[1] https://doi.org/10.5201/ipol.2018.225

1 Introduction

Image matching, which consists in deciding whether or not several images represent some common or similar objects, is a problem recognized as difficult, especially because of the viewpoint changes between images. The apparent deformations of (smooth) objects caused by changes of the camera position can be locally approximated by affine maps. This observation has motivated the development of comparison methods robust to local affine transformations. These methods usually compare local image descriptors, which they try to make as invariant as possible to affine transformations.

The best established image comparison method is SIFT [10]. This method was shown in [17] to perform recognition invariant to image rotations, translations, and camera zoom-outs. SIFT has inspired numerous variations over the past 15 years [7, 2, 1]. In this paper, we refer to these methods as Scale Invariant Matching Methods (SIIM). Several attempts have also been made to create local image descriptors invariant to affine transformations [11, 19, 12]. Yet, it was shown in [17] that none of these approaches is truly invariant to local affine transformations, due to the fact that optical blur and affine transformations do not commute. As a result, these methods cannot handle angle viewpoint differences larger than 60° for planar objects [18, 13], and lose quickly efficiency for angles larger than 45° [6].

A more pragmatic approach, proposed a few years ago with the ASIFT Algorithm [17] and adopted by several authors ever since [21, 13], consists in applying a pre-determined set of affine transformations to each compared image, in order to simulate the transformations induced by the viewpoint changes. Instead of comparing two images, the resulting algorithm therefore compares all the pairs of simulated images. In this article, we refer to these simulation algorithms as IMAS, for Image Matching by Affine Simulation. In favorable cases, IMAS can capture changes of point of view up to an impressive 88°.

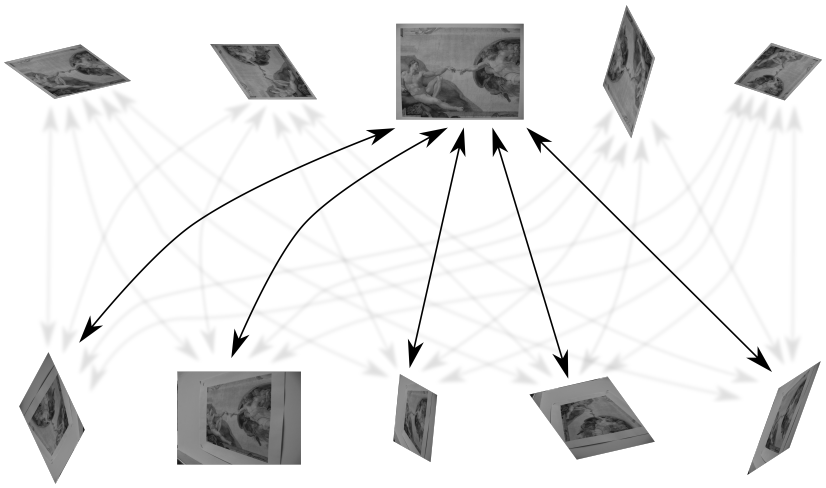


Figure 1: IMAS algorithms start by applying a finite set of optical affine simulations to u and v, followed by pairwise comparisons.

The first IMAS method provided with a mathematical proof of affine invariance is ASIFT [18, 24]. As its name indicates, ASIFT is an affine invariant extension of SIFT, that actually operates on SIFT. It can operate on any Scale Invariant Image Matching (SIIM) method like SURF [2] as well, provided its descriptor shows some robustness to angle viewpoint changes like SIFT descriptors do. Unlike MSER [11], LLD [19], Harris-Affine and Hessian-Affine [12], which attempt at normalizing all of the six affine parameters, ASIFT only simulates the two camera axis angles, and then applies SIFT which

simulates the scale and normalizes the rotation and the translation. Similarly, FAIR-SURF [21] is an IMAS method replacing SIFT by SURF in ASIFT. MODS [13] is another IMAS using heuristics to test fewer affine transforms and also combining different SIIMs. Other IMAS approaches do not involve local descriptors: FAsT-Match [9] delivers affine invariance by assuming that the template (a patch in the query image) can be recovered inside the target image by a *unique* affine map. The six affine parameters are simulated instead of the three involved in ASIFT.

Our goal is in this paper is to provide a generic IMAS method that cumulates three new improvements, all aimed at acceleration and robustness:

- it minimizes the number of camera axis angles to be simulated;
- it defines local hyper-descriptors grouping descriptors at the same location to accelerate matching;
- it resolves the problem of Lowe's matching thresholds that inhibits matching in presence of multiple similar objects in the same image.

In IMAS methods, the viewpoint change between different views of the same planar scene is measured by the so-called *relative transition tilts* [18, 24]. Transition tilts to be simulated to match two images can be much larger than absolute tilts. We describe here our optimal solution to the key question of IMAS methods: how to choose the list of tilts applied to both images to test these large transition tilts before comparison? In [23] we treated this question by finding optimal coverings of the space of affine tilts. In Section 2 we recall these results and give implementation details to construct nearly optimal coverings. Section 3 gives the construction of hyper-descriptors. In Section 4 we describe the two structural and computational improvements of the method. One is the replacement of Lowe's acceptance criterion by an *a contrario* criterion, and the other one is the elimination of useless "flat descriptors". Section 5 contains a short experimental assessment. Much more can be done on line!

2 Image Matching by Affine Simulation

In this section, we recall the definition of the space of tilts for planar affine transforms and the derivation of optimal coverings of this space. The affine transforms considered here can be interpreted as different viewpoints of a planar image from a remote camera, or more generally as the transition between two such oblique views. Indeed, given a frontal remote snapshot of a planar object $u(\mathbf{x}) = u(x, y)$, we can transition from any other remote oblique (affine) view Bu of the same object to any other oblique view Au through the affine transformation AB^{-1}.

Here, for any linear invertible map $A \in GL^+(2)$, we denote the affine transform A of a continuous image $u(\mathbf{x})$ by $Au(\mathbf{x}) = u(A\mathbf{x})$. We recall classic notation for three subsets of the general linear group $GL(2)$ of invertible linear maps of the plane,

$$\begin{aligned} GL^+(2) &= \{A \in GL(2) \mid \det(A) > 0\}, \\ GO^+(2) &= \left\{A \in GL^+(2) \mid A \text{ is a similarity}\right\}, \\ GL^+_*(2) &= GL^+(2) \setminus GO^+(2), \end{aligned}$$

where we call similarity any combination of a rotation and a zoom, and the symbol $\setminus$ denotes the set difference operator. Our central notion in the discussion is the *tilt* of an affine transform, which we now define. The proofs of the following propositions can be found in [23].

2.1 Absolute tilts

Proposition 1 (Morel, Yu [18]). *Every $A \in GL_*^+(2)$ is uniquely decomposed as*

$$A = \lambda R_1(\psi)\, T_t R_2(\phi), \tag{1}$$

where R_1, R_2 are rotations and $T_t = \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix}$, with $t > 1$, $\lambda > 0$, $\phi \in [0, \pi[$ and $\psi \in [0, 2\pi[$.

Remark. *It follows from this proposition that any affine map $A \in GL^+(2)$ is either uniquely decomposed as in (1) or is directly expressed as a similarity λR_1.*

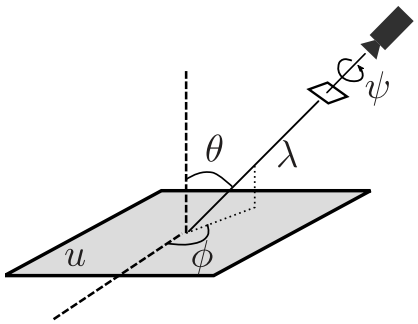


Figure 2: Geometric Interpretation of (1).

Figure 2 shows a camera viewpoint interpretation of this affine decomposition where the longitude ϕ and latitude $\theta = arccos\, \frac{1}{t}$ characterize the camera's viewpoint angles, ψ parameterizes the camera spin and λ corresponds to the zoom. In the ideal affine model, the camera is supposed to stand at infinite distance from a flat image u, so that the deformation of u induced by the camera indeed is an affine map. But the above approximation is still valid provided the image's size is small with respect to the camera distance. In other terms the affine model is locally valid for each small and approximately flat patch of a physical surface photographed by a camera at some distance. Yet, the affine deformation of the object's aspect will be different for each of its patches. This explains why affine invariant recognition methods deal with local descriptors. The parameter t defined above measures the so-called *absolute tilt* between the frontal view and a slanted view. The uniqueness of the decomposition in (1) justifies the next definition.

Definition 1. *We call* absolute tilt *of A the real number $\tau(A)$ defined by*

$$\begin{cases} GL^+(2) & \to & [1, \infty[\\ A & \mapsto & \begin{cases} 1 & \text{if } A \in GO^+(2) \\ t & \text{if } A \in GL_*^+(2) \end{cases} \end{cases},$$

where t is the parameter found when applying Proposition 1 to A.

We call longitude of A the angle $\phi(A)$ defined by

$$\begin{cases} GL^+(2) & \to & [0, \pi[\\ A & \mapsto & \begin{cases} 0 & \text{if } A \in GO^+(2) \\ \phi & \text{if } A \in GL_*^+(2) \end{cases} \end{cases},$$

where ϕ is the angle of the rotation R_2 found when applying Proposition 1 to A.

2.2 Transition Tilts

Image descriptors like those proposed in the SIFT method are invariant to translations, rotations and Gaussian zooms, which in terms of the camera position interpretation (see Figure 2) correspond to a fronto-parallel motion of the camera, a spin of the camera and to an optical zoom. We shall focus on the last part $T_t R_2$ of the decomposition (1) because it is the one that is imperfectly dealt with by SIIMs. SIIMs are instead able to detect objects *up to a similarity*. This leads us to the next definition.

Definition 2. *Let $A, B \in GL^+(2)$. Then we define the right equivalence relation $\sim$ as*

$$A \sim B \quad \Leftrightarrow \quad AB^{-1} \in GO^+(2)\,.$$

Definition 3. *Let $A, B \in GL^+(2)$. We call* transition tilt *between A and B the absolute tilt of AB^{-1}, i.e.*

$$\tau\left(AB^{-1}\right).$$

The transition tilt has an agreeable visual interpretation appearing in Figure 3. By Formula (1) applied to AB^{-1}, passing from an image Bu to an image Au comprises a single non-Euclidean transformation, namely the central tilt matrix $T_{\tau(AB^{-1})}$ which squeezes the image in a fixed direction, namely the x horizontal direction, after having rotated it. Thus the transition tilt measures the amount of image distortion caused by a change of view angle. We now recall the formal properties of the transition tilt stated in [18].

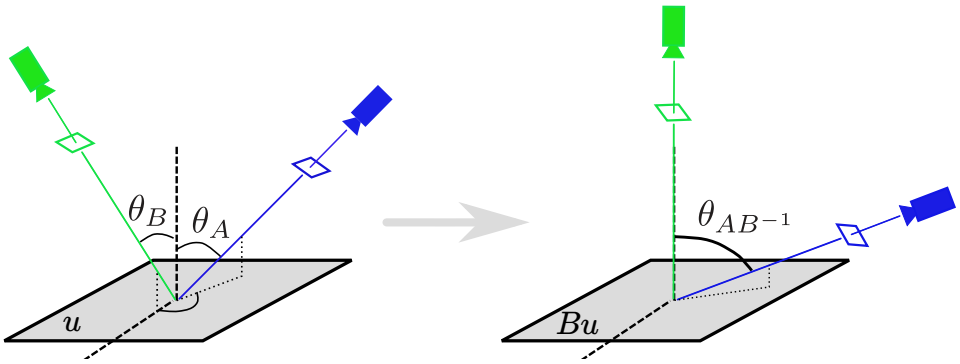


Figure 3: Passage from transition tilts (left side) to absolute tilts (right side).

Proposition 2. *For $A, B \in GL^+(2)$ we have*

1. $\tau(AB^{-1}) = 1 \Leftrightarrow A \sim B$;
2. $\tau(A) = \tau(A^{-1})$;
3. $\tau(AB^{-1}) = \tau(BA^{-1})$;
4. $\tau(AB^{-1}) \leq \tau(A)\,\tau(B)$;
5. $max\left\{\frac{\tau(A)}{\tau(B)}, \frac{\tau(B)}{\tau(A)}\right\} \leq \tau(AB^{-1})$.

Definition 4. *We call* Space of Tilts, denoted by Ω, *the quotient $GL^+(2)\,/\sim$, where the equivalence relation $\sim$ has been given in Definition 2.*

This definition completes Definition 2 and clarifies the geometrical interpretation of the space of tilts: an element in the space of tilts represents the set of all the camera spins and zooms associated with a certain tilt in a certain direction.

Notation. *Let $A \in GL^+(2)$. We denote by $[A]$ the equivalence class in the space of tilts associated to A i.e.*

$$[A] = \left\{ B \in GL^+(2) \mid A \sim B \right\}.$$

Definition 5. *We denote by i the canonical injection from the space of tilts to $GL^+(2)$ defined by*

$$i : \left\{ \begin{array}{ccc} \Omega & \rightarrow & GL^+(2) \\ [A] & \mapsto & T_{\tau(A)} R_{\phi(A)} \end{array} \right. .$$

This injection filters out the canonical representative from each class which is a mere tilt in the x direction.

Remark. *Clearly, the function i satisfies*

$$[A] = [i([A])],$$

and the space of tilts can be parameterized by picking these representative elements in each class as

$$\Omega = [Id] \bigcup \left\{ \bigcup_{(t,\phi) \in]1,\infty[\times [0,\pi[} [T_t R_\phi] \right\}.$$

The notion of transition tilt is helpful for measuring the affine distortion from a fixed affine viewpoint to surrounding affine viewpoints. Proposition 3 gives an adequate measuring tool. As argued in [23, 6], transition tilt tolerances (determining visible viewpoints) are SIIM dependent. Most SIIMs are able to identify viewpoint changes under 45° for image sizes around 700×550.

Transition tilts do not depend on the class representative of A or B, so they can be defined directly on the quotient $GL^+(2) / \sim$.

Proposition 3. *The function d*

$$d : \left\{ \begin{array}{ccc} \Omega \times \Omega & \rightarrow & \mathbb{R}_+ \\ ([A], [B]) & \mapsto & \log \tau (B A^{-1}) \end{array} \right. ,$$

is a metric acting on the space of tilts.

Let us now recall disks formulas in tilt space with respect to the metric d in Proposition 3.

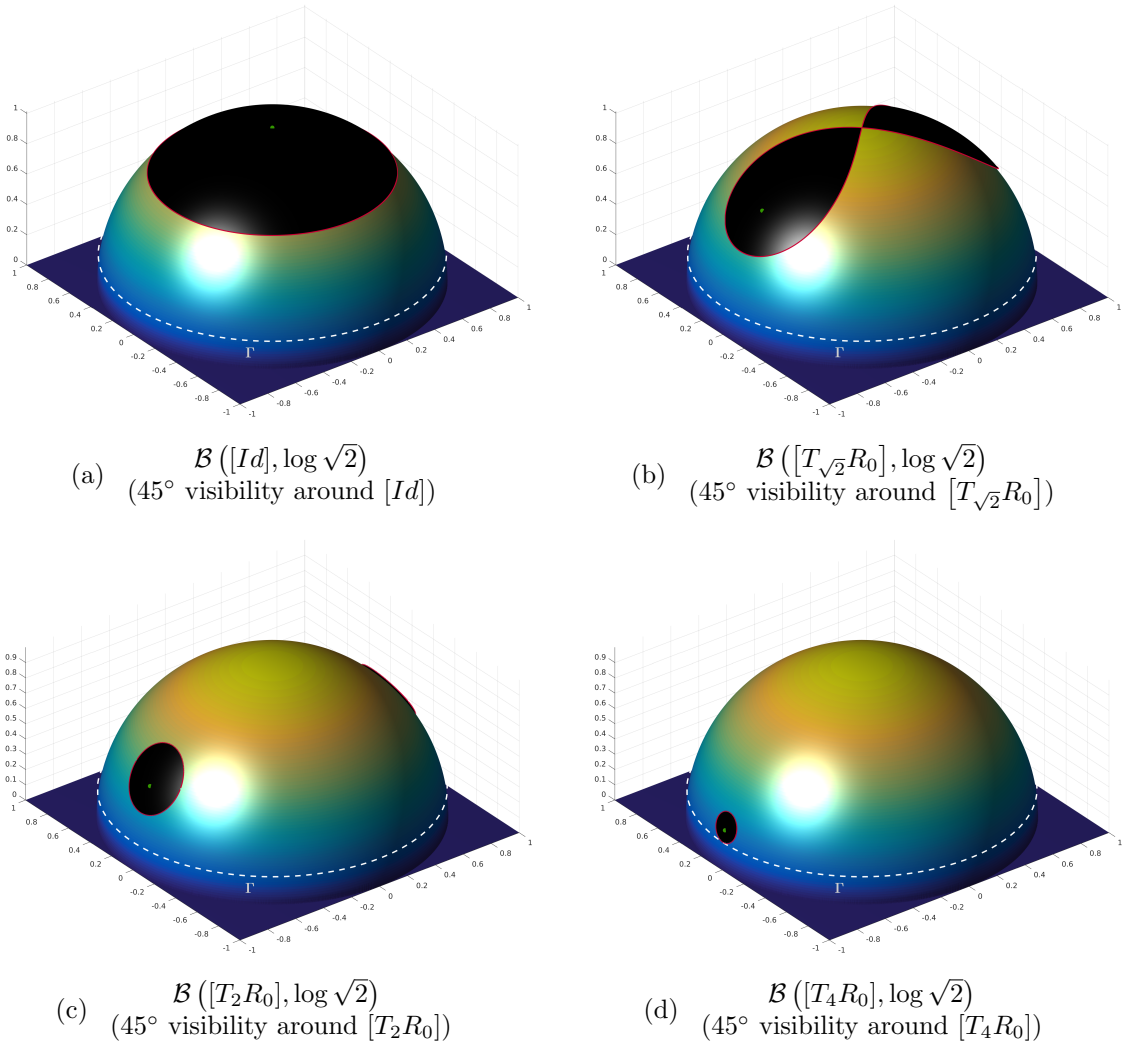
Notation. *Let $S \in \Omega$ and $r > 0$. We denote either by $\mathcal{B}(S, r)$ or by $\mathcal{B}_S^r$, the disk in the space of tilts centered at S and with radius r.*

Theorem 1 (Rodríguez, Delon, Morel [23])**.** *Given an element of the space of tilts in canonical form $[T_t R(\phi_1)]$, the disk $\mathcal{B}([T_t R(\phi_1)], r)$ in the space of tilts corresponds to the following set*

$$\left\{ [T_s R(\phi_2)] \mid G(t, s, \phi_1, \phi_2) \leq \frac{e^{2r} + 1}{2e^r} \right\},$$

where

$$G(t, s, \phi_1, \phi_2) = \left(\frac{\frac{t}{s} + \frac{s}{t}}{2} \right) \cos^2(\phi_1 - \phi_2) + \left(\frac{\frac{1}{st} + st}{2} \right) \sin^2(\phi_1 - \phi_2).$$



(a) $\mathcal{B}\left([Id], \log\sqrt{2}\right)$ (45° visibility around $[Id]$)

(b) $\mathcal{B}\left([T_{\sqrt{2}}R_0], \log\sqrt{2}\right)$ (45° visibility around $[T_{\sqrt{2}}R_0]$)

(c) $\mathcal{B}\left([T_2R_0], \log\sqrt{2}\right)$ (45° visibility around $[T_2R_0]$)

(d) $\mathcal{B}\left([T_4R_0], \log\sqrt{2}\right)$ (45° visibility around $[T_4R_0]$)

Figure 4: (Perspective views) Green point - Affine transformation in question; Dashed line - $\partial\mathcal{B}\left([Id], \log 4\sqrt{2}\right)$; Black surface - Disk in question

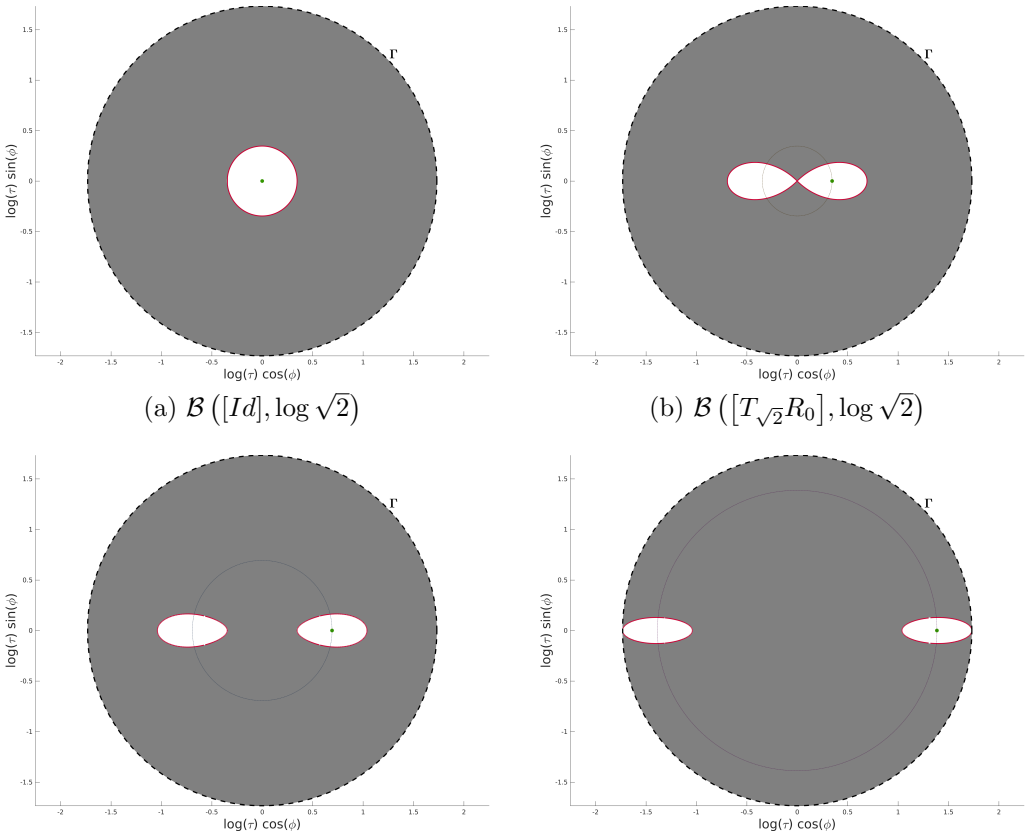
Figures 4 and 5 show, in a perspective view and in polar coordinates respectively, four disks in the space of tilts centered at four reference tilts. The radius of these disks corresponds to a maximal change of angle view of 45° with respect to the disk's center. The larger the tilt, the smaller the disk with that radius, which means that we need more and more disks to cover the high tilt regions. Notice that all disks appear duplicated by symmetry. Indeed, a perspective visualization of Ω is impossible in $\mathbb{R}^3$: Ω is the quotient of the half sphere by a central symmetry.

As a consequence, affine simulation is now a reliable way of extending the initial visibility range of a SIIM. The idea is to place affine simulations in a way that they render all elements in region $\gamma \subset \Omega$ perfectly visible for at least one of them. When that happens we call that set of affine simulations a *covering* of the region in question.

Definition 6. *We call $\mathbb{S} \subset \Omega$ an α°-covering of a region $\Gamma \subset \Omega$ if and only if*

$$\Gamma \subset \bigcup_{S \in \mathbb{S}} \mathcal{B}\left(S, \log \frac{1}{\cos(\alpha^\circ)}\right).$$

Remark. *In Definition 6, $\mathbb{S} \subset \Omega$ actually corresponds to the centers of the balls constituting the covering. For our scopes it will be a finite set, that determines the set of affine transformations used*



(a) $\mathcal{B}\left([Id], \log\sqrt{2}\right)$ (b) $\mathcal{B}\left([T_{\sqrt{2}}R_0], \log\sqrt{2}\right)$

(c) $\mathcal{B}\left([T_2R_0], \log\sqrt{2}\right)$ (d) $\mathcal{B}\left([T_4R_0], \log\sqrt{2}\right)$

Figure 5: (Polar coordinates)
Green point - Affine transformation in question
Dashed line - $\partial\mathcal{B}\left([Id], \log 4\sqrt{2}\right)$
Dotted line - Equal tilts
Red line - Disk's boundary

to simulate affine viewpoints in IMAS algorithms, i.e.

$$\{i(S) \,|\, S \in \mathbb{S}\}.$$

The region $\Gamma \subset \Omega$ of Definition 6 usually denotes a circular region representing all viewpoints within a certain angle θ. The following definition gives a name to these sets.

Definition 7. *The set $\Gamma \subset \Omega$ is called a γ°-region if and only if*

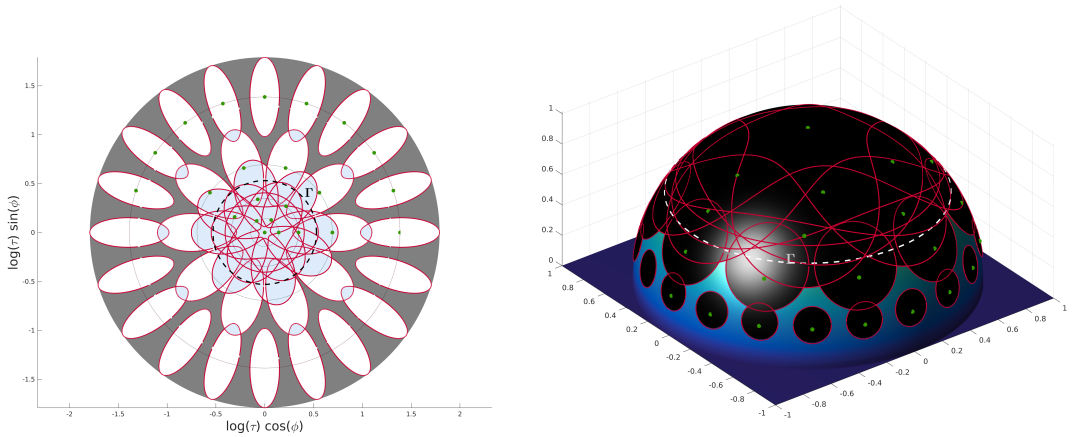
$$\Gamma = \left\{ [T_t R_\phi] \,|\, t \leq \frac{1}{\cos(\gamma^\circ)} \right\}.$$

Several α°-coverings of γ°-regions have been proposed in [23, 18, 24, 21, 13] for SIFT and SURF; among them those in Figure 6. It is easily seen that they are far from optimality: some of these coverings do not really cover the region they were meant to, except for ASIFT [18, 24] which instead is visually redundant. The following section describes the near optimal coverings proposed in [23].

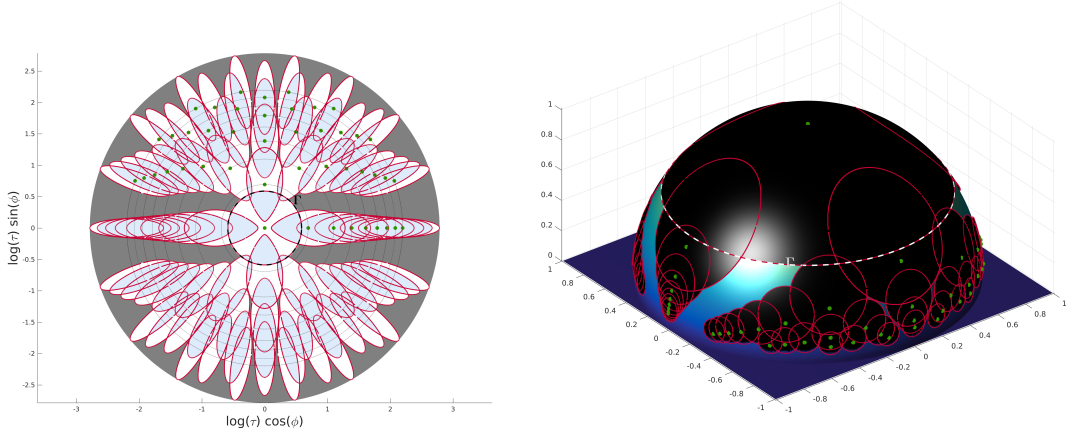
2.3 Near Optimal α°-Coverings

Near optimal coverings in [23] ensure minimal complexity for IMAS algorithms. One example of these α°-coverings is represented in Figure 9a.

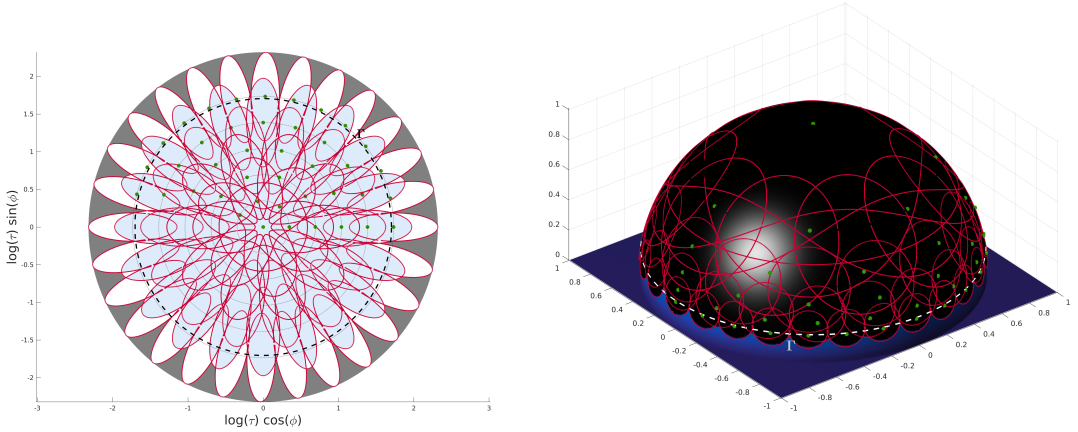
The optimization problem is to minimize the overall number of descriptor comparisons while maintaining the same detection efficiency. This minimization *is not* equivalent to a minimization of



(a) FAIR-SURF fixed tilts [21], a set of 23 affine simulations with an area ratio of 11.42. It would represent a 48°-covering of a 53°-region. Highly redundant in the central part, it does not cover the 80°-region.



(b) MEDIUM DoG-SIFT [13], a set of 45 affine simulations with an area ratio of 9. It would represent a 56°-covering of a 56°-region. Although highly redundant, it does not cover the 80°-region.



(c) ASIFT [18, 24], a set of 41 affine simulations with an area ratio of 13.77. It would represent a 56°-covering of a 80°-region. Highly redundant, it does cover the region it was meant to.

Figure 6: Non optimal coverings

Green points - Affine camera simulations
Red lines - Visibility tolerance from each affine simulation
White/Black surfaces - Visible viewpoints regions
Dashed line - Covered regions

the number of simulated versions being used. Their efficiency criterion is based on two straightforward remarks. The first one is that if a digital image suffers a tilt t in any direction, its area gets modified

by a factor $\frac{1}{t}$. The second one is that the expected number of keypoints in a digital image is proportional to its area. Both remarks imply that the complexity of an IMAS algorithm will be given by the overall area of the simulated images being ultimately compared. This justifies the next definition.

Definition 8. *We call* area ratio *of* $\mathbb{S}$ *(a finite set of elements in* Ω*) the real number*

$$\sum_{S\in\mathbb{S}} \frac{1}{\tau\left(S\right)}.$$

The area ratio fixes the factor (larger than 1) by which the image area is being multiplied when summing the areas of all of its tilted versions. Then, as the ultimate goal is to reduce the number of key point comparisons, it is natural to look for a set $\mathbb{S}$ whose area ratio is close to the infimum among all $\log r$-coverings of Γ. It is difficult to find an optimal solution for this NP hard problem. Fortunately, our search space in the set of $\log r$-coverings can be drastically reduced by imposing practical and theoretical constraints to $\mathbb{S}$. Those constraints follow from simple requirements for an image matching method.

Definition 9. *We shall say that a set* $\mathbb{S}\in\Omega$ *is feasible if and only if:*

1. $[Id]\in\mathbb{S}$.
2. *There exist* $n\in\mathbb{N}^+$ *and*

$$(t_1,\ldots,t_n,\phi_1,\ldots,\phi_n)\in[1,\infty[^n\times]0,\pi]^n$$

such that

$$\mathbb{S}\setminus\{[Id]\}=\bigcup_{i=1}^{n}\left\{[T_{t_i}R_{k\phi_i}]\in\Omega\,|\,k=0,\ldots,\left\lfloor\frac{\pi}{\phi_i}\right\rfloor\right\},$$

where $\lfloor a\rfloor$ *denotes the nearest integer less than or equal to a real number* a.

The intuition behind Definition 9 is as follows: 1) avoids an image resolution loss before comparison, an obvious requirement; 2) imposes groups of concentric equidistant tilts which is a sound isotropy requirement.

Definition 10. *Set* $\Gamma=\mathcal{B}\left([Id],\log\Lambda\right)$. *A feasible set* $\mathbb{S}\in\Omega$ *with parameters*

$$(n,(t_1,\ldots,t_n,\phi_1,\ldots,\phi_n))\in\mathbb{N}^+\times[1,\infty[^n\times]0,\pi]^n$$

is said to be optimal among feasible sets if and only if it realizes the minimal area ratio. In other words, optimal feasible sets are solutions of

$$\begin{aligned}&\underset{(n,(t_1,\ldots t_n,\phi_1,\ldots\phi_n))\in\mathbb{N}^+\times[1,\infty[^n\times]0,\pi]^n}{\arg\min}\ 1+\sum_{i=1}^{n}\frac{|J_{t_i,\phi_i}|}{t_i},\\ &\text{subject to:}\quad \Gamma\subset\mathcal{B}^{\log r}_{[Id]}\cup\left\{\bigcup_{1\le i\le n}\ \bigcup_{S\in J_{t_i,\phi_i}}\mathcal{B}^{\log r}_{[S]}\right\},\end{aligned}\tag{2}$$

where J_{t_i,ϕ_i} *is the set of transformations of the form*

$$T_{t_i}R_{\phi_i},\ T_{t_i}R_{2\phi_i}\ \ldots,\ T_{t_i}R_{\left\lfloor\frac{\pi}{\phi_i}\right\rfloor\phi_i},$$

$|J_{t_i,\phi_i}|$ *is the cardinal of* J_{t_i,ϕ_i} *and* $\mathcal{B}^{\log r}_{[S]}$ *is denoting* $\mathcal{B}\left([S],\log r\right)$.

Some conditions have been proposed in [23] in order to verify that a γ° region is truly covered by a feasible set with a fixed number of concentric equidistant tilts. Let the intersection of disks boundaries, which are composed at most of two elements for non identical disks, be denoted by

$$\Sigma_i \coloneqq \partial\mathcal{B}^{\log r}_{\left[T_{t_i}\right]} \cap \partial\mathcal{B}^{\log r}_{\left[T_{t_i R_{\phi_i}}\right]}, \tag{3}$$

and their respective closest and farthest elements be denoted by

$$\min \Sigma_i \coloneqq \arg\min_{S\in\Sigma_i} d\left(S,[Id]\right), \qquad \max \Sigma_i \coloneqq \arg\max_{S\in\Sigma_i} d\left(S,[Id]\right).$$

Then Algorithm 1 summarizes the aforementioned conditions in a function, called IsGammaCovered, that is to be called for querying if a feasible set covers a γ° region.

Algorithm 1: IsGammaCovered

input:
The initial α° visibility (fixes $r = \frac{1}{\cos\alpha^\circ}$).
The γ° region to cover (fixes $t_\gamma = \frac{1}{\cos\gamma^\circ}$).
parameters:
n - Number of concentric equidistant tilts for the feasible set (as in Definition 9). This also fixes the amount of sets Σ_i, defined in (3).
ε - Number of uniformly discretized elements in each dimension with respect to the metric d of Proposition 3.

1. covered_portion $= r$ *// A feasible set always has the disk* $\mathcal{B}^{\log r}_{[Id]}$
2. **if** $\Sigma_1 = \emptyset$ **then**
   return(false)
3. **foreach** $i = 1, \cdots, n-1$ **do**
   if $\tau\left(\min \Sigma_i\right) >$*covered_portion* **then**
   return(false)
   covered_portion $= \max \Sigma_i$ *// the annulus* $\mathcal{B}^{\log\tau(\max\Sigma_i)}_{[Id]} \setminus \mathcal{B}^{\log\tau(\min\Sigma_i)}_{[Id]}$ *is already covered*
   if *covered_portion* $> t_\gamma$ **then**
   return(true)
   if $\Sigma_{i+1} = \emptyset$ **then**
   return(false)
   foreach $[T_t R_\phi] \in \mathbb{F}_\varepsilon$ *// $\mathbb{F}_\varepsilon$ is the finite ε-dense set appearing in (4).*
   do
   if $[T_t R_\phi] \notin \bigcup_{j=i}^{i+1} \mathcal{B}^{\log r-\varepsilon}_{\left[T_{t_j} R_{\left\lfloor \frac{\phi}{\phi_j}\right\rfloor \phi_j}\right]} \cup \mathcal{B}^{\log r-\varepsilon}_{\left[T_{t_j} R_{\left\lceil \frac{\phi}{\phi_j}\right\rceil \phi_j}\right]}$ **then**
   return(false) *// $[T_t R_\phi]$ must lie inside one of the four nearest disks (Theorem 1 is used)*
   // at this point the annulus $\mathcal{B}^{\log\tau(\min\Sigma_{i+1})}_{[Id]} \setminus \mathcal{B}^{\log\tau(\max\Sigma_i)}_{[Id]}$ *has been proved to be covered*
   covered_portion $= \min \Sigma_{i+1}$
4. **if** $\tau\left(\max \Sigma_n\right) > t_\gamma$ **then**
   return(true)
else
   return(false)

Figure 7 illustrates the iterative process described in Algorithm 1. One crucial step in Algorithm 1 is the creation of an ε-dense set of a given annulus. As explained in [23], this set is a helping hand

to ensure a $\log(r-\varepsilon)$-covering up to an error of ε and so, by dilating back disks radius to r one ensures $\log r$-coverings. Of course, there exists an infinite number of ε-dense sets. For annulus like

$$\mathcal{B}_{[Id]}^{\log t_{i+1}} \setminus \mathcal{B}_{[Id]}^{\log t_i}$$

we propose to build the following set, which is proven to be a ε-dense set by the mere application of the triangle inequality of the metric d,

$$\mathbb{F}_\varepsilon := \left\{ [T_{e^{n\varepsilon}t_i} R_{k\beta_i}] \in \Omega | n, k \in \mathbb{N}_+, e^{n\varepsilon} t_i < t_{i+1}, k\,\beta_\varepsilon\left(e^{n\varepsilon} t_i\right) < \pi \right\}, \tag{4}$$

where the function β_ε, appearing in Definition 11, determines the angle step for equal distances over the same tilt.

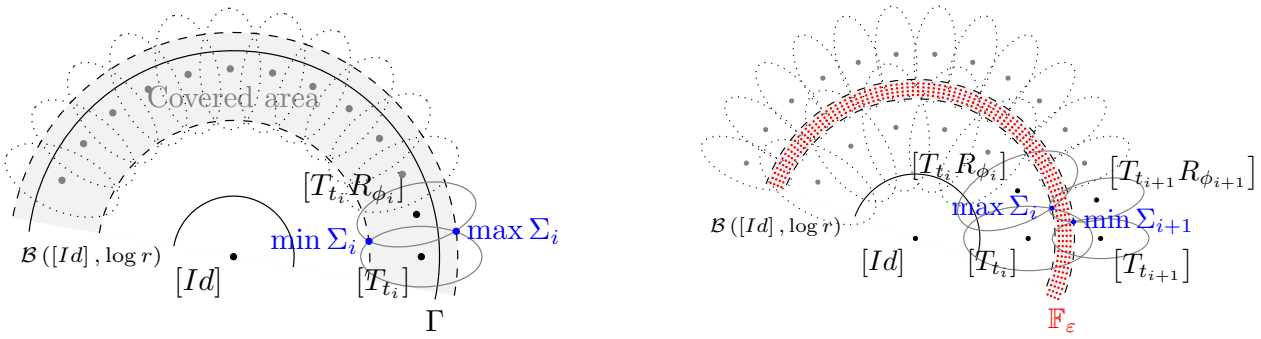


Figure 7: Verifying covering conditions for feasible sets in Algorithm 1. Left : covered annulus determined by $\min \Sigma_i$ and $\max \Sigma_i$. Right : all elements in $\mathbb{F}_\varepsilon$ must lie inside at least one disk.

Indeed, let $z \in \mathcal{B}_{[Id]}^{\log t_{i+1}} \setminus \mathcal{B}_{[Id]}^{\log t_i}$ and its surrounding four points $y_i \in \mathbb{F}_\varepsilon$, $i \in \mathbb{Z}/4\mathbb{Z}$ satisfying $d(y_i, y_{i+1}) = \varepsilon$. Four auxiliary points, x_i $i \in \mathbb{Z}/4\mathbb{Z}$, are defined as projections of z on arcs (see Figure 8). In that case, we always have either $d(y_i, x_i) \leq \frac{\varepsilon}{2}$, either $d(y_{i+1}, x_i) \leq \frac{\varepsilon}{2}$. This implies that there exists at least one pair (j, k) with $k = j$ or $k = j+1$ for which $d(x_j, z) \leq \frac{\varepsilon}{2}$ and such that

$$d(y_k, z) \leq d(y_k, x_j) + d(x_j, z) \leq \varepsilon.$$

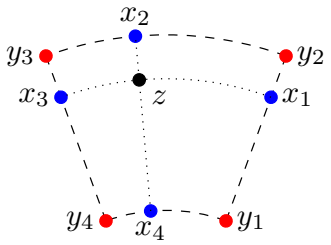


Figure 8: Proving the ε-density of $\mathbb{F}_\varepsilon$.

Definition 11. *We denote by β_ε, the function defined by*

$$\beta_\varepsilon : \begin{cases}]1, \infty] & \to & [0, \pi[\\ t & \mapsto & \phi(t) \end{cases},$$

where $\phi(t)$ is such that

$$d\left([T_t], \left[T_t R_{\phi(t)}\right]\right) = \varepsilon.$$

The procedure in the proof of Proposition 3.16 in [23] can also be applied to find all kinds of near optimal coverings depending on the initial visibility α° and the γ°-region to be covered. Algorithm 2 delivers a way of finding near optimal α°-coverings by fixing n, the number of concentric equidistant tilts, and then optimizing over $2n$ dimensions. By means of the canonical injection in Definition 5, a given α°-covering determines the set of affine maps to be simulated in Algorithm 4. Some examples of these kind of coverings can be found in Table 1.

Algorithm 2: Finding near optimal coverings

input:
The initial α° visibility and the γ° region to cover.
parameters:
n - Number of concentric equidistant tilts (as in Definition 9).
κ - Number of uniformly discretized elements in each dimension with respect to the metric d of Proposition 3.
inner definitions:
$]t_1,t_2]_\kappa = \left\{\frac{t_2}{e^{n\varepsilon_\kappa}} \,|\, n\in\mathbb{N}, t_1 < \frac{t_2}{e^{n\varepsilon_\kappa}} \le t_2\right\}$, $]0,\beta]_\kappa^t = \{n\beta_{\varepsilon_\kappa}(t) \,|\, n\in\mathbb{N}, 0 < n\beta_{\varepsilon_\kappa}(t) \le \beta\}$
where ε_κ is such that $\left|]t_1,t_2]_\kappa\right| = \left|]0,\beta]_\kappa^t\right| = \kappa$, and $\beta_{r^2}(t_i)$ appears in Definition 11.

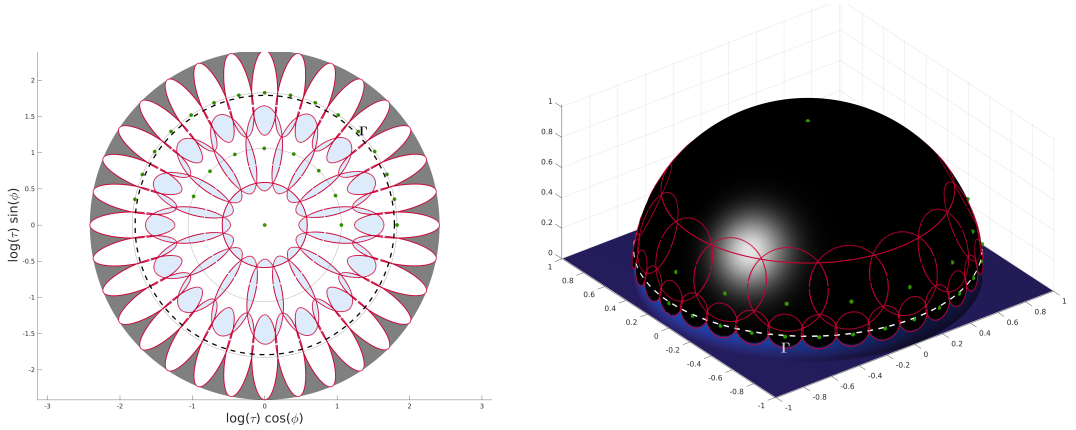
1. $r = \log\left(\frac{1}{\cos(\alpha^\circ)}\right)$ // α° *equivalent transition tilt*
2. $ar = \infty$ // *current minimal area ratio*
3. **foreach** $(t_1,t_2,\cdots,t_n) \in]r,r^2]_\kappa \times]t_1 r, t_1 r^2]_\kappa \times \cdots \times]t_{n-1} r, t_{n-1} r^2]_\kappa$ **do**
   foreach $(\phi_1,\phi_2,\cdots,\phi_n) \in]0,\beta_{r^2}(t_1)]_\kappa^{t_1} \times]0,\beta_{r^2}(t_2)]_\kappa^{t_2} \times \cdots \times]0,\beta_{r^2}(t_n)]_\kappa^{t_n}$ **do**
      if $\left(\sum_{i=1}^n t_i \left\lfloor \frac{\pi}{\phi_i} \right\rfloor < ar\right)$ & IsGammaCovered $(\gamma, t_1,\cdots,t_n,\phi_1,\cdots,\phi_n)$ **then**
         $ar = \sum_{i=1}^n t_i \left\lfloor \frac{\pi}{\phi_i} \right\rfloor$
         $\left(t_1^{\text{opt}},\cdots,t_n^{\text{opt}}\right) = (t_1,\cdots,t_n)$
         $\left(\phi_1^{\text{opt}},\cdots,\phi_n^{\text{opt}}\right) = (\phi_1,\cdots,\phi_n)$

return $\mathcal{S} = \{t_k^{opt}, \phi_k^{opt}\}_{k=1,\ldots,n}$

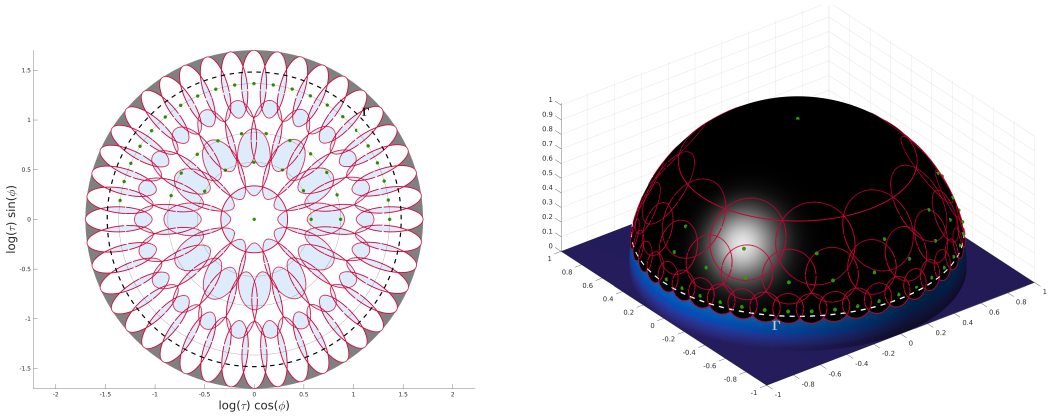
α°	γ°	**EV**	**AR**	t_1^{opt}	ϕ_1^{opt}	t_2^{opt}	ϕ_2^{opt}	t_3^{opt}	ϕ_3^{opt}
45°	80°	87°	15.889	1.84641	0.459445	2.68973	0.234551	4.58177	0.116774
54°	80°	87°	7.354	2.54902	0.450362	4.71215	0.18624	-	-
54°	81°	88°	7.548	2.67673	0.350162	5.65043	0.175859	-	-
56°	80°	87°	6.290	2.89419	0.396183	6.33474	0.198091	-	-
56°	83°	88°	7.221	2.89419	0.397562	6.07477	0.150497	-	-
56°	84°	89°	9.014	2.79309	0.461217	4.61946	0.24717	9.65081	0.123523
58°	82°	88°	5.971	3.01682	0.450814	6.03598	0.200202	-	-
58°	84°	89°	7.979	3.02483	0.448874	5.09033	0.261983	10.4035	0.131014
60°	84°	89°	6.126	3.2948	0.396543	7.78261	0.156965	-	-

Table 1: Near optimal α°-coverings of γ°-regions. As proved in [23], these α°-coverings will ensure for a generic IMAS an extended visibility in column **EV** for a near-minimal area ratio in column **AR**.

Remark. *The global optimization proposed in Algorithm 2 should be followed by some iterations of a refined search on the neighboring subsets containing the current optimum.*



(a) 56°-covering with a set of 28 affine simulations that renders viewpoints visible in a 82°-region.



(b) 45°-covering with a set of 43 affine simulations that renders viewpoints visible in an 82°-region.

Figure 9: Near optimal coverings as in [23]

Green points - Affine camera simulations
Red lines - Visibility tolerance from each affine simulation
White/Black surfaces - Visible viewpoints regions
Dashed line - Covered regions

2.4 Simulating Digital Tilts

While the former sections only dealt with affine geometry, we now must introduce the formalism of the camera blur, as we shall deal with digital image recognition. Our goal is to define rigorously affine invariant recognition for digital images.

We now introduce a compact notation for convolution with a Gaussian. We shall denote by $\star_x$ the 1-D convolution convolution operator in the x-direction, i.e.

$$G \star_x u\,(x,y) = \int_{\mathbb{R}} G\,(z)\,u\,(x-z,y)\,dz.$$

We denote by $\mathbb{G}^x_\sigma$ the 1D convolution operator in the x direction with

$$G^x_{c\sigma}\,(x) \;:=\; \frac{1}{\sqrt{2\pi}c\sigma} e^{-\frac{x^2}{2(c\sigma)^2}},$$

namely

$$\mathbb{G}^x_\sigma u \;:=\; G^x_{c\sigma} \star_x u.$$

Here the constant $c \geq 0.7$ is large enough to ensure that all convolved images, initially sampled at 1 distance, can be sub-sampled at Nyquist distance σ without causing significant aliasing.

We now formalize the notion of tilt. There are actually three different notions of tilt, that we must carefully distinguish. In all that follows, u_0 denotes the (theoretical) infinite resolution image that would be obtained by a frontal snapshot of a plane object with infinitely many pixels.

Definition 12. *Given $t > 1$, the tilt factor, define:*

- *Geometric tilts*

$$T_t u_0(x, y) =: u_0(tx, y).$$

- *Simulated tilts (taking into account camera blur)*

$$\mathbb{T}_t v =: T_t \mathbb{G}^x_{\sqrt{t^2-1}} \star_x v.$$

- *Digital tilts (transforming a digital image u into a digital image)*

$$\mathbf{u} \to \mathbf{S}_1 \mathbb{T}_t I \mathbf{u}.$$

where $\mathbf{S}_1$ and I denote respectively the image sampling operator defined on the grid $\mathbb{Z}^2$ and the Shannon-Whittaker interpolator of a digital image on $\mathbb{Z}^2$.

Digital tilts are the ones used in practice. It all adds up because the simulated tilt yields a blur permitting $\mathbf{S}_1$-sampling. Algorithm 3 digitally simulates tilts in any direction, i.e simulates affine viewpoints from any place of the sphere.

Algorithm 3: Generating digital tilts in any direction ($\mathbf{u} \rightsquigarrow \mathbf{S}_1 \mathbb{T}_t R_\phi I \mathbf{u}$)

input: the digital image $\mathbf{u}$ and parameters (t, ϕ)

1. $\mathbf{u} = \text{Rotate}(\mathbf{u}, \phi)$ *// with bilinear interpolation*
   // Rotate will frame the rotated version of $\mathbf{u}$ in a minimal rectangular image
2. $\mathbf{u} = \text{GaussianBlur1D}\left(\mathbf{u}, \sigma = 0.8\sqrt{t^2 - 1}\right)$ *// Gaussian blur in the x-direction*
3. $\mathbf{u} = \text{Subsample}(\mathbf{u}, t)$ *// Subsamples the image along the x-direction by a factor of t*

return u

2.5 From SIIM to IMAS

We have seen in the previous sections how to compute a near optimal discrete set of affine transformations in order to cover a given region with a visibility α. The core idea of the IMAS approach, described in Algorithm 4, is to apply this optimal set of transformations to images before comparing them with a SIIM method. We showed in Proposition 3.6 of [23] that this algorithm offers an affine-invariant version of the associated SIIM method. Indeed, the optimal covering ensures that there is at least one pair of simulated images whose transition tilt is visible for the SIIM. Table 1 gives some examples of ensured visibility, depending on the assumed initial visibility α° of the SIIM and the γ°-region to be covered.

The core idea of the IMAS algorithm is illustrated by Figure 1.

Choosing the right α-covering is fundamental. It depends on the SIIM's transition tilt tolerance (with respect to a given image size) and the wanted extended visibility. Clearly, the parameter α should be less than the initial visibility (determined by the transition tilt tolerance) of the SIIM.

Algorithm 4: Formal IMAS (Image Matching by Affine Simulation)

input: query and target images: $\mathbf{u}$ and $\mathbf{v}$.
parameters:
a routine SIIM-Detector, two sets of optimal coverings $\mathcal{S}_1 = \{t_k^1, \phi_k^1\}_{k=1,\dots,n_1}$ and $\mathcal{S}_2 = \{t_k^2, \phi_k^2\}_{k=1,\dots,n_2}$, provided by Algorithm 2
// $\zeta(D, \mathbf{u}, t, \phi)$ is a simple routine that filters out those descriptors in D which after back projection with $(T_t R_\phi)^{-1}$ are not fully inside the domain of $\mathbf{u}$

1. **foreach** $k = 1, \dots, n_1$ **do**
   $D_k^u = \text{SIIM-Detector}\left(\mathbf{S}_1 \mathbb{T}_{t_k^1} R_{\phi_k^1} \mathbf{u}\right)$ *// Descriptors on each simulated image from* $\mathbf{u}$
   $D_k^u = \zeta\left(D_k^u, \mathbf{u}, t_k^1, \phi_k^1\right)$
2. **foreach** $k = 1, \dots, n_2$ **do**
   $D_k^v = \text{SIIM-Detector}\left(\mathbf{S}_1 \mathbb{T}_{t_k^2} R_{\phi_k^2} \mathbf{v}\right)$ *// Descriptors on each simulated image from* $\mathbf{v}$
   $D_k^v = \zeta\left(D_k^v, \mathbf{v}, t_k^2, \phi_k^2\right)$
3. **foreach** $(k_1, k_2) \in \{1, \dots, n_1\} \times \{1, \dots, n_2\}$ **do**
   $M_{k_1,k_2} = \text{SIIM-Matcher}\left(D_{k_1}^u, D_{k_2}^v\right)$ *// Set of matches between* $\mathbf{u}$ *and* $\mathbf{v}$

return $M = \bigcup_{(k_1,k_2)\in\{1,\dots,n_1\}\times\{1,\dots,n_2\}} M_{k_1,k_2}$

3 Hyper-Descriptors in IMAS

IMAS algorithms implementations have to deal with various types of spurious or redundant matches that do not appear in the corresponding SIIM approaches. In this section, we describe the reasons behind the occurrence of these aberrant matches. Since false matches make it hard in practice to identify the underlying image transformation, we propose a simple idea to eliminate most of them without significantly reducing the number of good matches. To do this, we rely on the notion of hyper-descriptors. A hyper-descriptor is a group of several local descriptors of the different simulated images whose corresponding keypoints are close when back-projected in the original image plane.

3.1 Identifying Aberrant Matches

We identify in the following paragraphs three kinds of spurious or aberrant matches inherently produced by IMAS algorithms.

First, these approaches naturally favor repetitive matches, since the same keypoints can be matched in several pairs of simulated images. These matches are often good but provide redundant information and should be considered as a single match.

Second, IMAS algorithms are prone to yield what we call *multiple-to-one* matches, where several keypoints from the query image match one and the same keypoint from the target image. These matches appear frequently in IMAS algorithms when they use Lowe's second nearest neighbor acceptance criterion[2]. Since simulated target images with high absolute tilts are smaller and contain fewer keypoints, the number of second nearest neighbors they provide is also smaller for these images than for lower absolute tilts. In consequence, keypoints in these images are more likely to match those in the query image and have a tendency to be involved in *multiple-to-one* matches. These matches can be considered as false matches.

Third, IMAS algorithms have a strong tendency to give *one-to-multiple* matches, where a single keypoint from the query image matches multiple keypoints from the target image. Such matches,

[2]In Lowe's acceptance criterion, the ratio between the distance to the nearest neighbor and the distance to the second nearest neighbor is thresholded to decide if a match is accepted or rejected.

which can also be taken as indications of false matches, are naturally eliminated from SIIM approaches by Lowe's acceptance criterion. The multiplicity of keypoints comparisons for a given structure in IMAS algorithms favors the apparition of such *one-to-multiple* matches.

In order to handle all the above spurious matches generated by Algorithm 4, a post-processing routine was proposed in [18, 24] and adopted by all subsequent IMAS approaches. This post-processing is usually composed of three successive filters designed to remove repetitive matches, *multiple-to-one* matches and *one-to-multiple* matches. Such a post-processing is described in Algorithm 5 and can be applied after Algorithm 4 to identify the underlying transformation between the two images. Unfortunately, many good matches also get eliminated by this hack. For example, if only one false match comes to meet one end of a true match, then both matches get eliminated. Conversely, truly repetitive objects create various *multiple-to-one / one-to-multiple* matches that will always get discarded by this post-processing, and should not.

To avoid the loss of such correct matches, while eliminating the spurious and aberrant matches described above, we introduce in the next section the notion of hyper-descriptor.

3.2 Hyper-Descriptors Matching

Definition 13. *Let* $\mathbf{u}$ *be an image and* $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ *the optimal set of affine transformed versions of* $\mathbf{u}$ *obtained with an IMAS algorithm. We call* ρ-hyper-descriptor *of* $\mathbf{u}$ *a group of SIIM descriptors of this set of images, whose keypoints, once reprojected in* $\mathbf{u}$*, are all contained in a ball of radius* ρ*. The corresponding group of keypoints is called* ρ*-hyper-keypoint.*

In practice, we choose the radius ρ between 3 and 6 pixels, and we keep this parameter fixed. In the following, for the sake of simplicity, we will assume that ρ is given and speak directly about hyper-descriptors. Figure 10 shows an exemple of hyper-descriptor composed of three SIIM descriptors extracted from three different affine simulations of the same image $\mathbf{u}$. This example illustrates an ideal case in which the center of three SIIM descriptors coincide. In practice the keypoints are not detected at exactly the same location. This ideal case would appear if no numerical errors were involved when simulating tilts and obtaining descriptors around an infinitely accurate corresponding keypoint.

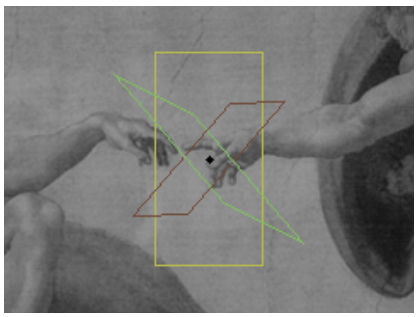


Figure 10: A ρ-hyper-descriptor $d = \{\omega_1, \omega_2, \omega_3\}$
Black point - Center of d and of each ω_i
Yellow parallelogram - SIIM descriptor ω_1 on $\mathbf{S}_1\mathbb{T}_2\mathbf{u}$
Brown parallelogram - SIIM descriptor ω_2 on $\mathbf{S}_1\mathbb{T}_4R_{\frac{\pi}{4}}\mathbf{u}$
Green parallelogram - SIIM descriptor ω_3 on $\mathbf{S}_1\mathbb{T}_6R_{\frac{3\pi}{2}}\mathbf{u}$

Now that we have defined hyper-descriptors, Algorithm 7 describes a greedy approach to extract them from an image $\mathbf{u}$. First, a SIIM detector (SIFT or SURF for instance) is applied to all affine simulated versions of $\mathbf{u}$ (the set of affine transformations is provided as a parameter of the algorithm). Then, each detected SIIM descriptor gets assigned to an existing hyper-descriptor or a new one is created. When assigned to an existing group, the center of the hyper-keypoint needs to be recomputed

Algorithm 5: Post-Processing of Algorithm 4

input:
M - List of output Matches of Algorithm 4.
parameters:
ρ_1, ρ_2, ρ_3 - Distance thresholds
GeometricFilter - An algorithm detecting a geometric consensus among a list of matches (e.g. RANSAC, ORSA Homography [14], ORSA Fundamental [15] or USAC [22]).
output:
Filtered list of Matches

// a match m is composed by $m.k_q$ and $m.k_t$ which are respectively the associated query key-point and target key-point.

// The spatial distance between any two key-points is denoted by $\Lambda(k_1, k_2) = \left| \begin{pmatrix} k_1.x - k_2.x \\ k_1.y - k_2.y \end{pmatrix} \right|$

// unique filter

$M_u = \emptyset$
1. **foreach** $m \in M$ **do**
    flag-unique = true
    foreach $m_u \in M_u$ **do**
        if $\Lambda(m.k_q, m_u.k_q) \leq \rho_1$ ***and*** $\Lambda(m.k_t, m_u.k_t) \leq \rho_1$ **then**
            flag-unique = false
    if *flag-unique == true* **then**
        $M_u = M_u \cup m$
$M = M_u$

// multiple2one filter

$M_m = \emptyset$
2. **foreach** $m \in M$ **do**
    flag-multiple2one = false
    foreach $m_m \in M \setminus \{m\}$ **do**
        if $\Lambda(m.k_q, m_m.k_q) \geq \rho_3$ ***and*** $\Lambda(m.k_t, m_m.k_t) \leq \rho_2$ **then**
            flag-multiple2one = true
    if *flag-multiple2one == false* **then**
        $M_m = M_m \cup m$
$M = M_m$

// one2multiple filter

$M_m = \emptyset$
3. **foreach** $m \in M$ **do**
    flag-one2multiple = false
    foreach $m_m \in M \setminus \{m\}$ **do**
        if $\Lambda(m.k_q, m_m.k_q) \leq \rho_2$ ***and*** $\Lambda(m.k_t, m_m.k_t) \geq \rho_3$ **then**
            flag-one2multiple = true
    if *flag-one2multiple == false* **then**
        $M_m = M_m \cup m$
$M = M_m$

// Filtering matches agreeing with geometric consensus

4. $M = \text{GeometricFilter}(M)$
return M

and the hyper-descriptor can be merged with a neighboring group. Each of the above computations can be done with an $O(1)$ complexity, implying that the descriptor extraction parts of Algorithm 4 and Algorithm 6 have about the same complexity.

Algorithm 6: IMAS (with hyper-descriptors)

input: query and target images: $\mathbf{u}$ and $\mathbf{v}$.

1. $D_1 = \text{IMAS-Detector}(\mathbf{u})$ *// as in Algorithm 7*
2. $D_2 = \text{IMAS-Detector}(\mathbf{v})$ *// as in Algorithm 7*
3. $M = \text{IMAS-Matcher}(D_1, D_2)$ *// as in Algorithm 8*

return M

Algorithm 7: IMAS-Detector

input: image $\mathbf{u}$

parameters:

a routine SIIM-Detector, one set of optimal coverings $\mathcal{S} = \{t_k, \phi_k\}_{k=1,\dots,n}$, provided by Algorithm 2

// The spatial distance between any two descriptors is denoted by $\Lambda(d_1, d_2) = \left| \begin{pmatrix} d_1.x - d_2.x \\ d_1.y - d_2.y \end{pmatrix} \right|$

1. $D = \emptyset$ *// Storage of hyper-descriptors*
2. **foreach** $k = 1, \dots, n$ **do**
   - **foreach** $s \in \text{SIIM-Detector}(\mathbf{S}_1 \mathbb{T}_{t_k} R_{\phi_k} \mathbf{u})$ **do**
     - **if** $(\mathbb{T}_{t_k} R_{\phi_k})^{-1}(s)$ *is fully inside the domain of* $\mathbf{u}$ **then**
       - **if** *it exists* $d \in \arg\min_{b \in D} \Lambda(b, s)$ *such that* $\Lambda(d, s) \leq \rho$ **then**
         - *// Add the descriptor* s *to the* ρ*-hyper-descriptor* d
         - $d = d \bigcup \{s\}$
         - $d.x = \frac{\sum_{s \in d} s.x}{|d|}$, $d.y = \frac{\sum_{s \in d} s.y}{|d|}$
         - **foreach** $d_1 \in D$ *such that* $\Lambda(d_1, d) \leq \rho$ **do**
           - $d = d \bigcup d_1$
           - $d.x = \frac{\sum_{s \in d} s.x}{|d|}$, $d.y = \frac{\sum_{s \in d} s.y}{|d|}$
         - **else**
           - *// Create a new* ρ*-hyper-descriptor from* s
           - $D = D \bigcup \{\{s\}\}$

return D

The distance between hyper-descriptors is defined as the minimal distance between the descriptors they are composed of.

Definition 14. *Let*

$$\begin{aligned} d_1 &= \{\alpha_1, \cdots, \alpha_{n_1}\} \\ d_2 &= \{\beta_1, \cdots, \beta_{n_2}\} \end{aligned}$$

be two hyper-descriptors where α_i *and* β_j *are denoting SIIM descriptors. Let also* δ *be a distance for SIIM descriptors. We call distance between* d_1 *and* d_2 *the positive number*

$$\Delta(d_1, d_2) = \min_{\substack{1 \leq i \leq n_1 \\ 1 \leq j \leq n_2}} \delta(\alpha_i, \beta_j).$$

Remark. *Usually* $\delta(\alpha, \beta)$ *is either* $\|\alpha - \beta\|_{L_1}, \|\alpha - \beta\|_{L_2}$ *or the Hamming distance*[3].

We can now derive an IMAS Matcher algorithm between hyper-descriptors (see Algorithm 8). We use here a straightforward generalization of Lowe's criteria to the previous distance. If two hyper-descriptors (d_1, d_2) define a match, then the left and right positions of a match are refined to

$$\arg \min_{(\alpha,\beta)\in d_1 \times d_2} \delta(\alpha, \beta).$$

Algorithm 8: IMAS-Matcher

input: two sets of hyper-descriptors D_1, D_2.
parameters: the match ratio $\lambda \in]0, 1[$.
1. $M = \emptyset$ // *Storage of Matches*
2. **foreach** $d \in D_1$ **do**
   - $a \in \arg\min_{c\in D_2} \Delta(d, c)$ // $\Delta(x, y) = \min_{(\alpha,\beta)\in x\times y} \delta(\alpha, \beta)$
   - $b \in \arg\min_{c\in D_2\setminus a} \Delta(d, c)$
   - **if** $\frac{\Delta(d,a)}{\Delta(d,b)} \leq \lambda$ **then**
     - $M = M \bigcup \arg\min_{(\zeta,\eta)\in d\times a} \delta(\zeta, \eta)$

return D

In order to get similar performances, the parameter *match ratio* in Algorithm 8 should be greater than its homologous in the SIIM-Matcher of Algorithm 4. Indeed, the second nearest neighbour applied on each simulated target image is less restrictive than just one application of it on all simulated target images at the same time.

In practice, for SIFT based descriptors, the match ratio (λ) of Algorithm 8 is set to 0.8. This configuration seems to correspond to a match ratio of 0.6 for the SIIM-Matcher of Algorithm 4.

Note that hyper-descriptors are not associated with a given affine transformation but rather group descriptors from several simulated versions of u. Comparing all the hyper-descriptors of two images **u** and **v** is therefore faster than comparing the descriptors of all their simulated versions. Indeed, when computing the distance between an hyper-descriptor of **u** and an hyper-descriptor of **v**, the computation can be stopped as soon as this distance exceeds the second smallest distance already calculated for this point. This step saves much more time with hyper-descriptors than with conventional descriptors.

The use of hyper-descriptors allows to completely remove the classical filters used in standard IMAS algorithms to avoid problematic matches. Indeed, all post-processing filters that were usually applied in standard IMAS algorithms are now pointless:

1. Filtering repetitive matches (the *unique* filter step of Algorithm 5) is no longer useful. Algorithm 7 considers groups of close descriptors as one single hyper-descriptor holding all the information. A match between two hyper-descriptors is considered as a single match.

2. Filtering *one-to-multiple* matches is now naturally included by the fact that the IMAS-Matcher of Algorithm 8 generalizes Lowe's criterion [10] to hyper-descriptors.

3. *Multiple-to-one* matches are not forbidden with hyper-descriptors but do not appear any more in practice.

[3]The Hamming distance is mostly used with binary descriptors.

Algorithms 7 and 8 finally give birth to Algorithm 6, an IMAS algorithm based on hyper-descriptors. As simple as it is, this algorithm increases radically the quality of matches. No post-processing is needed in order to extract the underlying meaningful transformation. Thus, any parameter estimation approach like RANSAC [5], LO-RANSAC [4], ORSA [14, 15] or USAC [22], can be applied right after Algorithm 6. We then propose that any IMAS based on Algorithm 4 should evolve into Algorithm 6.

4 Two Structural and Computational Improvements

We describe in this section two computational tricks. The first one slightly modifies Lowe's acceptance criterion in order to enable multiple matches for each hyper-descriptor. The second one is purely used for speed-up considerations, and consists in filtering flat descriptors in the early stages of the whole matching algorithm.

4.1 *A Contrario* Matching Revisited

In Lowe's acceptance criterion, the ratio between the distance to the nearest neighbor and the distance to the second nearest neighbor is thresholded to decide if a match is accepted or rejected. The second nearest neighbor is taken in the target image. This has several drawbacks. First, it introduces a bias for small target images, which contain less descriptors and therefore pass the threshold more easily. A second structural bias is that this threshold also eliminates matches with repeated regions in the target images. One way of allowing one-to-multiple matches that are truly present in the target image is to create an *a contrario* model from an independent base of keypoints. We therefore propose to take a third image as background model, as it was first proposed in [3]. Instead of selecting the second nearest descriptors among those of the target image, Algorithm 9 uses the nearest hyper-keypoint among those of this third image.

Algorithm 9: IMAS-Acontrario-Matcher

input: three sets of ρ-hyper-descriptors D_1, D_2, D_a.
parameters: the match ratio $\lambda \in]0, 1[$.
1. $M = \emptyset$ *// Storage of Matches*
2. **foreach** $d \in D_1$ **do**
   $a \in \arg\min_{c \in D_2} \Delta(d, c)$ *// $\Delta(x, y) = \min_{(\alpha,\beta)\in x\times y} \delta(\alpha, \beta)$*
   $b \in \arg\min_{c \in D_a} \Delta(d, c)$
   if $\frac{\Delta(d,a)}{\Delta(d,b)} \leq \lambda$ **then**
      $M = M \bigcup \arg\min_{(\zeta,\eta)\in d\times a} \delta(\zeta, \eta)$
return D

The idea behind Algorithm 9 is consistent with Lowe's justification. It evaluates on the *a contrario* image how likely it is that a descriptor matches so well just by chance.

By equipping Algorithm 6 with the IMAS-Matcher of Algorithm 9 we allow repetitions in an image to be recognized. In this way, more reliable information is passed on. For example, keypoints lying on repetitive windows in a building will not be removed, they will rather match with each other and add up when the meaningful transformation is queried while post-processing.

4.2 Filtering Flat Descriptors for Faster Computations

In order to speed-up the matching part of the algorithm, we can identify and eliminate unidirectional descriptors in the early stages of the algorithm. Flat descriptors are more likely to match each other and create too many false matches. These flat descriptors are identified with two internal filters in SIIM:

1. On-edge keypoints are considered as unstable. To detect an edge response, the ratio of smallest to largest principal curvatures of the DOG function (eigenvalues of the Hessian) is to be below a threshold. In our case, we set the threshold to 0.08 for octave scales less than 1 and to 0.06 otherwise.

2. Strongly biased descriptors towards a particular direction can also be eliminated by the means of the structure tensor. The eigenvalues of the structure tensor effectively summarize the predominant directions of the gradient around the keypoint. In our case, the ratio between the smallest and largest eigenvalues is set to be less than 0.06.

This filter is nonetheless optional and does not change significantly the final result. However, it often reduces the total computing time.

5 Numerical Results

Our IMAS method can be tested as an IPOL demo[4] for two of the most popular state-of-the-art SIIMs, namely SIFT and SURF[5]. The IMAS versions of SIFT and SURF are now ensured to have minimal complexity thanks to the near optimal coverings described in Section 2.

Several versions of SIFT can also be tested in our IPOL demo. HalfSIFT [8] and RootSIFT [1] have been successfully applied in Computer Vision. They yield small modifications of SIFT descriptors but improve the quality of the results. By only taking the square root of a SIFT descriptor after a normalization, RootSIFT is known to outperform SIFT in terms of transition tilt tolerance [23]. Unfortunately, most SIIMs fail in the case of non monotone intensity variations. HalfSIFT attempts to handle this by generating mod π-oriented descriptors. Indeed, this property makes HalfSIFT robust to contrast inversions. It improves the comparison of day/night images of the same objects, or images of the same objects taken in different wavelengths. Finally, a third descriptor, called HalfRootSIFT, cumulates the effects of RootSIFT and HalfSIFT. It is also available in the demo and improves over HalfSIFT.

5.1 Using Optimal Coverings

We first illustrate the gain obtained by using our near optimal coverings (see [23]) instead of the classical coverings proposed in [18, 21]. We refer the reader to [23] for a more rigorous approach. Table 2 shows a brief comparison between classical and optimal coverings for two SIIMs: SIFT + L1 norm and Root-SIFT + L2 norm.

Retrieved homographies in Table 2 were visually the same for all fours methods. The mean homogeneous homography matrix H_{mean} and its standard deviation are shown in (5). As those statistics are difficult to interpret, we then compute in (6) the maximal distance of mapped points for all four homographies h_i with respect to the mean homography h_{mean}. Equation (6) also indicates

[4] https://doi.org/10.5201/ipol.2018.225

[5] We use the SURF version developed in [20] which improves its former version in transition tilt tolerance.

	M	ar	ar^2	Keypoints (seconds)	Matching (seconds)	Filters (seconds)
ASIFT + L1	801	13.7	189.6	6	36	0
Optimal ASIFT + L1	401	8.9	79.2	3	14	0
ARoot-SIFT + L2	821	13.7	189.6	6	10	1
Optimal ARoot-SIFT + L2	503	7.34	53.8	3	3	0

Table 2: Matching methods performance over query and target images from Figure 11. Computations were performed on an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz with 4 cores.
M - Matches.
ar - area ratio.

Figure 11: Graffiti.

that the time saved in computations when using optimal coverings does not affect the accuracy of the retrieved homographies.

$$H_{\text{mean}} = \begin{bmatrix} 0.4356 & -0.6739 & 455.6987 \\ 0.4460 & 1.0201 & -51.2587 \\ 0.0005 & -0.0001 & 1.0000 \end{bmatrix}, \text{ std} = \begin{bmatrix} 0.0011 & 0.0052 & 0.9886 \\ 0.0016 & 0.0061 & 0.3213 \\ 0.0000 & 0.0000 & 0 \end{bmatrix} \tag{5}$$

$$\max_{v \in \text{ query image domain}} \max_{i \in 1,\cdots,4} \|h_{\text{mean}}(v) - h_i(v)\|_{L^2} = 3.2994. \tag{6}$$

5.2 Using Hyper-Descriptors

Using hyper-descriptors, introduced in Section 3, usually yields more quality matches than using standard descriptors. Descriptors that once were eliminated by the *multiple-to-one / one-to-multiple* filter are now kept without causing a burst of false matches. This means that no post-processing is needed after Algorithm 6. In order to identify the underlying meaningful transformations, this demo relies on four versions of the RANSAC Algorithm [5]: ORSA Homography [14], ORSA Fundamental [15] and USAC (Homography and Fundamental) [22].

We first highlight the need of all filters in the case of usual descriptors in Algorithm 4 and the advantage of Algorithm 6. Table 3 gives detailed information on how filters perform when applied sequentially from left to right. Optimal Affine RootSIFT was selected to perform all comparisons in this section.

It is usually required to remove repetitive matches when using RANSAC. Surprisingly, Table 3 (rows 3 and 4) shows an example in which applying the unique filter results at the end in a smaller quantity of true matches than not applying it. In practice, it is usually not the case and repetitive

matches might produce a degenerate case for RANSAC, yielding at the end an inconsistent transformation. Table 3 (rows 1 and 2) shows that the application of multiple-to-one and one-to-multiple filters can be a more crucial step.

The last row of Table 3 shows that Algorithm 6 is already giving a clean set of matches and that most of the post-processing (Algorithm 5) is now pointless, except for the geometric filter. Figure 13 shows the output matches from Algorithm 4 and Algorithm 6 corresponding respectively to the second and last rows of Table 3. Figure 13 visually shows the improvement in terms of quality of Algorithm 6 over Algorithm 4.

IMAS Algorithm	Output Matches	*Unique* Filter	*Multiple-to-one* and *one-to-multiple* Filters	ORSA Homography Filter [14]
Algorithm 4	10986	-	-	6
Algorithm 4	10986	8864	-	0
Algorithm 4	10986	-	122	42
Algorithm 4	10986	8864	117	38
Algorithm 6	309	-	-	98

Table 3: The first and second columns represent an IMAS algorithm and its output. From the third to the fifth columns one reads the sequence of filters appearing in Algorithm 5 applied (or not applied) to the output of the IMAS algorithm in question (in all cases configured for Optimal Affine-RootSIFT) which was run on the images from Figure 12. The final number of matches is coloured in blue if they are in accordance with the underlying homography; red on the contrary.

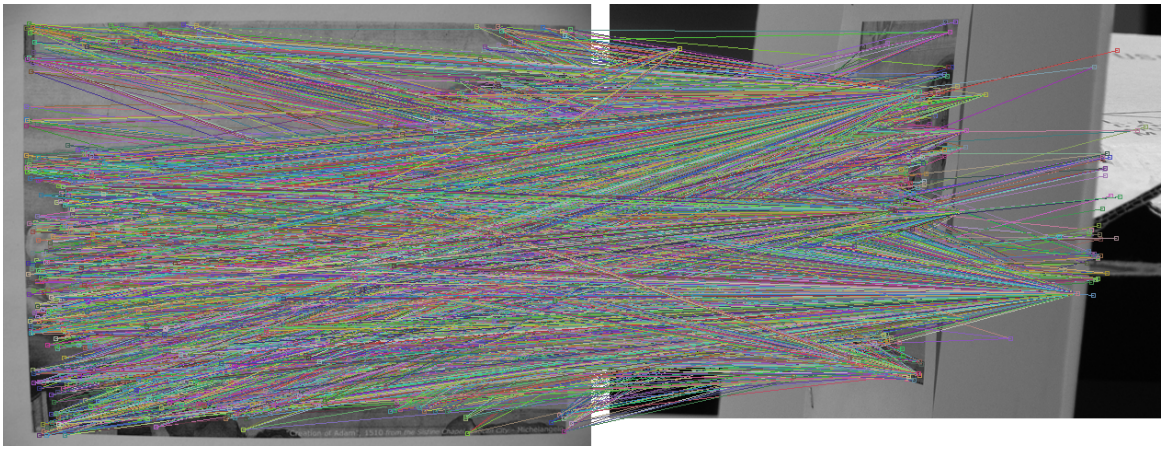
Figure 12: Adam.

5.3 Using the *A Contrario* Version of Lowe's Ratio Criterion

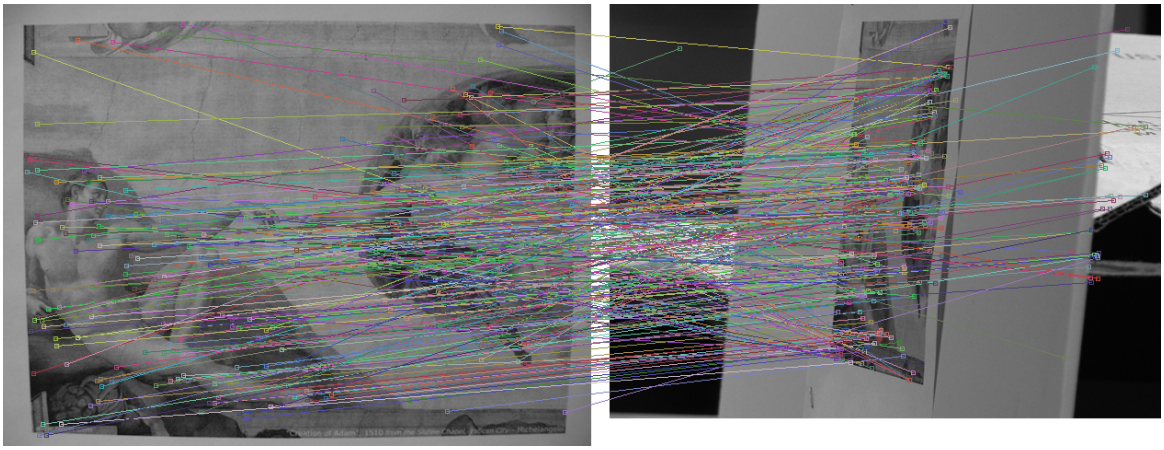
The *a contrario* matching in IMAS often incurs in a larger quantity of valid matches being accepted. However, the whole process relies on the hypothesis that the *a contrario* keypoints database represents an acceptable background model. This means that we should pay attention to the choice of the *a contrario* image. Usually, it is desired for this image to contain a wide variety of descriptors; natural images containing vegetation and water seem to go along with this property. Figure 14 has been selected for our experiments.

The interest of this *a contrario* version of Lowe's acceptance criterion lies in two properties:

1. First, it has a tendency to increase the number of matches, without increasing the number of false matches. This is illustrated in the following by a panorama stitching experiment.



(a) Algorithm 4 followed only by the unique filter of Algorithm 5, corresponding to the second row of Table 3. Too many *multiple-to-one* and *one-to-multiple* matches make it impossible for ORSA to determine the underlying homography.



(b) The raw output of Algorithm 6, corresponding to the last row of Table 3.

Figure 13: Analyzing Algorithms 4 and 6

Figure 14: Selected *a contrario* image for our demo.

2. Second, it authorizes multiple matches per descriptor, hence the detection of structure even if it is repeated multiple times in the target images. This is illustrated in the following by the detection of repeated structures.

Panorama stitching Panorama stitching is the process of combining two or more images with overlapping regions from different viewpoints to produce a single panorama. If the homography relating two images is perfectly known, then each point in one image can be located with respect to the other image's coordinates.

ORSA Homography [14] can be applied to assess if a homography explains the output matches of Algorithm 6. If it exists, that homography contains all the information needed for retrieving the query image around the target one. Figure 15 shows 474 matches in accordance with the homography retrieved by ORSA [14] applied right after Algorithm 6. Figure 16 shows a panorama stitching using this homography.

A slightly improved version of the above panorama stitching is obtained by introducing the *a contrario* matcher of Algorithm 9. The *a contrario* keypoints database was extracted from the *a contrario* image in Figure 14 by applying Algorithm 7. Resulting in 521 matches explained by the retrieved homography. Figure 17 shows the stitching result.

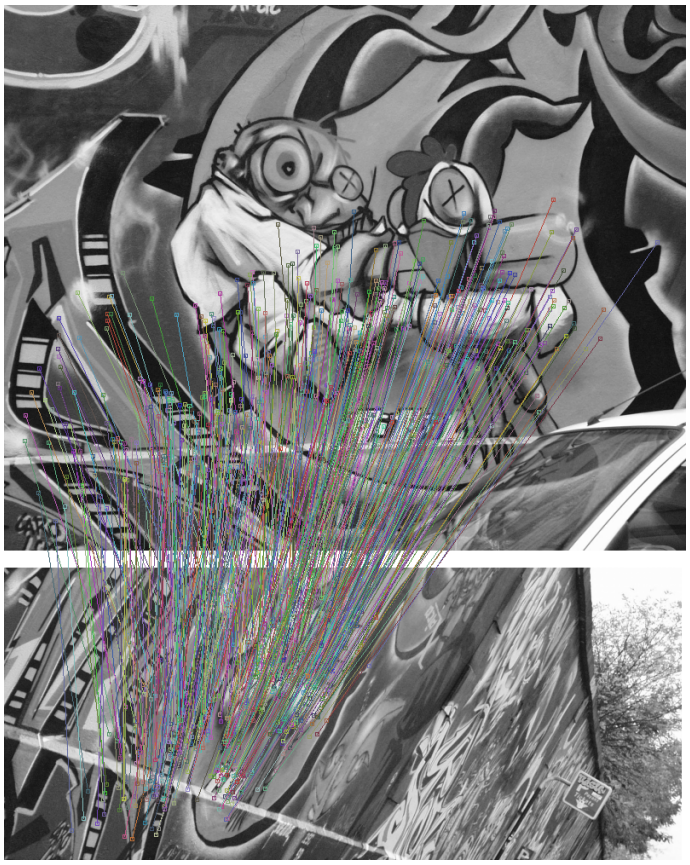


Figure 15: 474 matches among the output of Algorithm 6 were explained by ORSA Homography [14].

Detection of repeated structures As explained in Section 4.1, true repeated descriptors will annihilate each other if Algorithm 8 is applied. Figure 18 shows an example where most descriptors in the target image find a repeated copy somewhere in this very image. Only 22 matches were found in this scenario!

Figure 19 highlights the full potential of the *a contrario* matcher of Algorithm 9. This optional *a contrario* matcher is indeed well adapted when many repetitions are involved in the target image. Obviously its success depends on the choice of the *a contrario* image. On the other hand, this dependence is proved to be weak in practice (see Table 4).

Figure 16: Panorama stitching on Graffiti using the retrieved homography found by ORSA [14].

Figure 17: Panorama stitching on Graffiti with *a contrario* Matching. In this case, 521 matches among the output of Algorithm 6 with *a contrario* Matcher were explained by an homography retrieved by ORSA [14].

6 Conclusion

Three concepts to improve affine invariant image matching have been presented in this work. First, optimized set of simulations were presented in Section 2 for generic IMAS algorithms depending on the SIIM and the targeted viewpoint tolerance. In Section 3 a more robust framework for IMAS algorithms was presented based on generalizations of standard keypoints and the second closest neighbor criteria introduced by Lowe [10]. Finally, the *a contrario* matching of Section 4.1 is an optional way of keeping true repetitive matches in images. However, the success of the *a contrario*

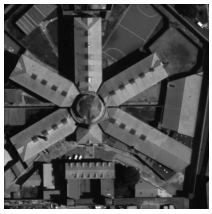
A contrario image	Size	Number of Hyper-Descriptors	Found Matches
	640 × 480	1181	227
	512 × 512	2601	212
	668 × 498	2891	198

Table 4: Weak dependence on the *a contrario* image. The *a contrario* version of Algorithm 6 was applied on the images from Figure 19 with three different *a contrario* images shown in this table. The resulting output matches are varying around the 222 found matches in Figure 19, visually depending on the number of hyper-descriptors.

Figure 18: A total of 26 matches were found by Algorithm 6 with the matcher of Algorithm 8.

matcher relies strongly on the assumption that our *a contrario* hyper-descriptors are able to capture the density of the space of natural hyper-descriptors.

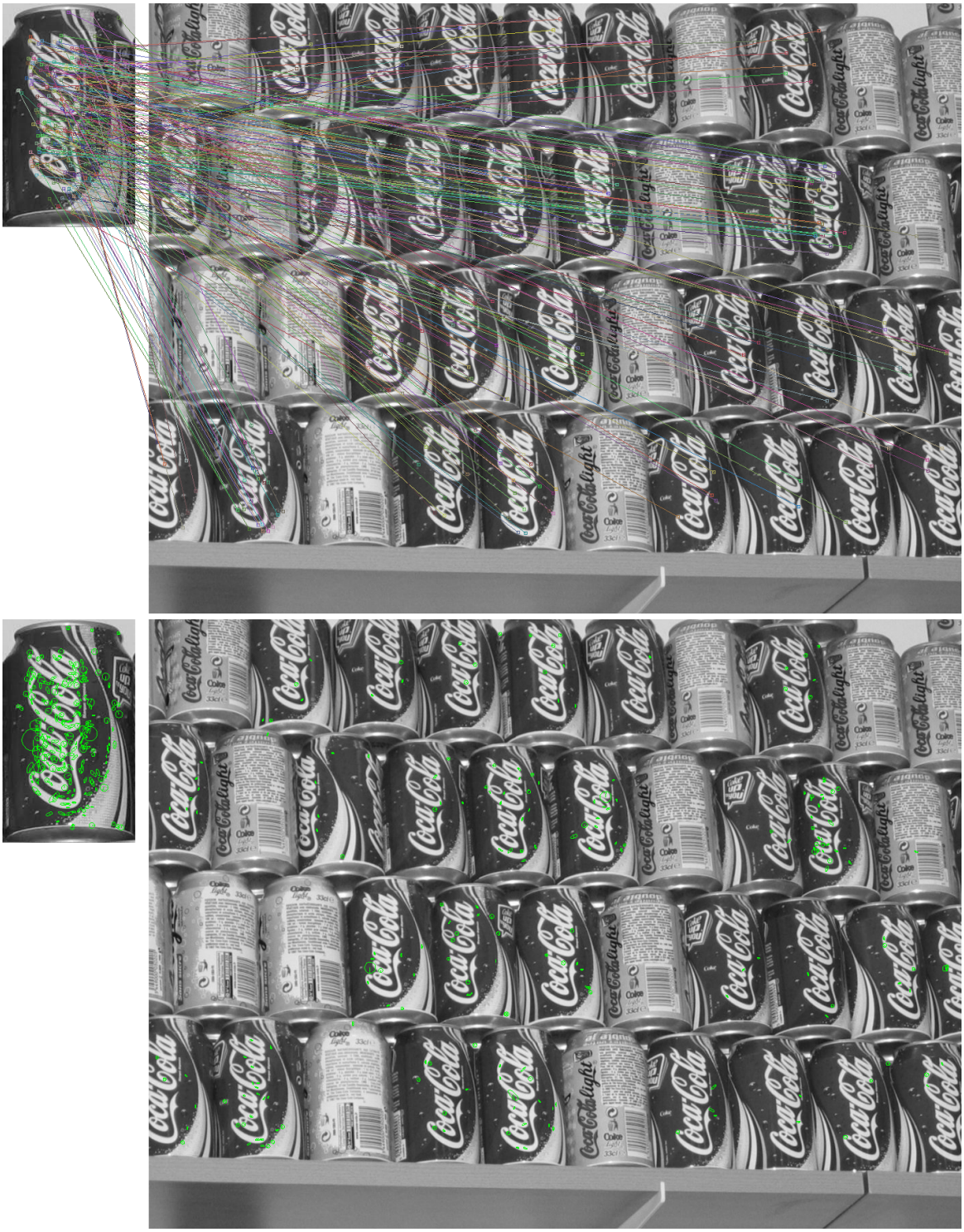


Figure 19: A total of 222 matches were found by Algorithm 6 (with the *a contrario* matcher of Algorithm 9). Comparing these results to those of Figure 18 shows the interest of the *a-contrario* matcher.

7 Image Credits

 belong to the Extreme View Dataset[6].

 was taken from http://wallpaperhdpark.com/awesome-nature-pictures

 were taken from the IPOL demo *Anatomy of SIFT*[7].

Images appearing in Figures 18-19 © Julie Delon.

References

[1] R. Arandjelovic and A. Zisserman, *Three things everyone should know to improve object retrieval*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 2911–2918. https://doi.org/10.1109/CVPR.2012.6248018.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded up robust features*, European Conference on Computer Vision, 1 (2006), pp. 404–417. https://doi.org/10.1007/11744023_32.

[3] F Cao, J.-L. Lisani, J.-M. Morel, P Musé, and F Sur, *A Theory of Shape Identification*, Springer Verlag, 2008. ISBN 978-3-540-68481-7.

[4] O. Chum, J. Matas, and J. Kittler, *Locally Optimized RANSAC*, Proceedings of the DAGM, 2781 (2003), pp. 236–243. https://doi.org/10.1007/978-3-540-45243-0_31.

[5] M.A. Fischler and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, 24 (1981), pp. 381–395. https://doi.org/10.1145/358669.358692.

[6] M. Karpushin, *Local features for RGBD image matching under viewpoint changes*, PhD thesis, Télécom ParisTech, 2016. https://tel.archives-ouvertes.fr/tel-01483314.

[7] Y Ke and R Sukthankar, *PCA-SIFT: A more distinctive representation for local image descriptors*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2 (2004), pp. 506–513. https://doi.org/10.1109/CVPR.2004.1315206.

[8] A. Kelman, M. Sofka, and C.V. Stewart, *Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007. https://doi.org/10.1109/CVPR.2007.383426.

[9] S. Korman, D. Reichman, G. Tsur, and S. Avidan, *Fast-Match: Fast Affine Template Matching*, in IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 1940–1947. https://doi.org/10.1007/s11263-016-0926-1.

[10] D.G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60 (2004), pp. 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[6] http://cmp.felk.cvut.cz/wbs

[7] http://demo.ipol.im/demo/82/

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla, *Robust wide-baseline stereo from maximally stable extremal regions*, Image and Vision Computing, 22 (2004), pp. 761–767. https://doi.org/10.1016/j.imavis.2004.02.006.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool, *A Comparison of Affine Region Detectors*, International Journal of Computer Vision, 65 (2005), pp. 43–72. https://doi.org/10.1007/s11263-005-3848-x.

[13] D. Mishkin, J. Matas, and M. Perdoch, *MODS: Fast and robust method for two-view matching*, Computer Vision and Image Understanding, 141 (2015), pp. 81 – 93. https://doi.org/10.1016/j.cviu.2015.08.005.

[14] L. Moisan, P. Moulon, and P. Monasse, *Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers*, Image Processing On Line, 2 (2012), pp. 56–73. https://doi.org/10.5201/ipol.2012.mmm-oh.

[15] ——, *Fundamental Matrix of a Stereo Pair, with A Contrario Elimination of Outliers*, Image Processing On Line, 6 (2016), pp. 89–113. https://doi.org/10.5201/ipol.2016.147.

[16] L. Moisan and B. Stival, *A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix*, International Journal of Computer Vision, 57 (2004), pp. 201–218. https://doi.org/10.1023/B:VISI.0000013094.38752.54.

[17] J-M. Morel and G. Yu, *On the consistency of the SIFT method*, tech. report, 2008. Citeseer.

[18] ——, *ASIFT: A new framework for fully affine invariant image comparison*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 438–469. https://doi.org/10.1137/080732730.

[19] P. Musé, F. Sur, F. Cao, and Y. Gousseau, *Unsupervised thresholds for shape matching*, in Proceedings of the International Conference on Image Processing, vol. 2, 2003, pp. 647–650. https://doi.org/10.1109/ICIP.2003.1246763.

[20] E. Oyallon and J. Rabin, *An Analysis of the SURF Method*, Image Processing On Line, 5 (2015), pp. 176–218. https://doi.org/10.5201/ipol.2015.69.

[21] Y. Pang, W. Li, Y. Yuan, and J. Pan, *Fully affine invariant SURF for image matching*, Neurocomputing, 85 (2012), pp. 6–10. https://doi.org/10.1016/j.neucom.2011.12.006.

[22] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J-M. Frahm, *USAC: a universal framework for random sample consensus*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (2013), pp. 2022–2038. https://doi.org/10.1109/TPAMI.2012.257.

[23] M. Rodriguez, J. Delon, and J-M. Morel, *Covering the space of tilts. application to affine invariant image comparison*, SIAM Journal on Imaging Sciences, 11 (2018), pp. 1230–1267. https://doi.org/10.1137/17M1140509.

[24] G. Yu and J-M. Morel, *ASIFT: An Algorithm for Fully Affine Invariant Comparison*, Image Processing On Line, 1 (2011), pp. 1–28. https://doi.org/10.5201/ipol.2011.my-asift.