



HAL
open science

Automatic detection of repeated objects in images

Mariano Rodríguez, Jean-Michel Morel, Julie Delon

► **To cite this version:**

Mariano Rodríguez, Jean-Michel Morel, Julie Delon. Automatic detection of repeated objects in images. IEEE International Conference on Image Processing, Sep 2021, Anchorage, Alaska, United States. 10.1109/icip42928.2021.9506799 . hal-03126917

HAL Id: hal-03126917

<https://hal.science/hal-03126917>

Submitted on 1 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AUTOMATIC DETECTION OF REPEATED OBJECTS IN IMAGES

M. Rodríguez,[†] J.-M. Morel[†], and J. Delon,[‡]

[†] Centre Borelli, ENS Paris-Saclay, Université Paris-Saclay, CNRS, France

[‡] Université de Paris, CNRS, MAP5 and Institut Universitaire de France

ABSTRACT

The definition of an "object" through the presentation of several of its instances is certainly one of the most efficient ways for humans and machines to learn. An object can be "learned" from a single image, just because it is repeating. In this paper, we explore a three step algorithm to detect repeated objects in images. Starting from a graph of auto-correspondences inside an image, we first extract sub-graphs composed of repetitions of unbreakable pieces of objects, that we call *atoms*. Then, these graphs of atoms are grouped into initial propositions of object instances. Finally, geometry inconsistencies are filtered out to end up with the final repeated object. The meaningfulness of object repetitions is measured by their Number of False Alarms (NFA), which provides a natural order among repeated objects in images; a very low NFA being a strong proof of existence of the discovered object. Source codes are available at <https://rdguez-mariano.github.io/pages/autosim>.

Index Terms— image comparison, SIFT, autosimilarities, symmetry detection, RANSAC, NFA.

1. INTRODUCTION

Everyday images often contain repeated objects, such as roof tiles, windows on buildings or chairs in a classroom. Humans not only identify these repetitions but also extract meaningful information from them. Sometimes, repetitions follow regularity patterns (e.g. directional, grids, symmetries, etc), but in other cases, no apparent structure is perceived.

Reliably detecting repetitions is challenging. Most approaches of the literature require strong assumptions. For example, in [1] a 2D repetition grid is required. In [2], and similar to [3], authors assume that large repetitive structures repeat along the horizontal direction in facade images. More specific relationships for repeated elements have also been investigated in [4–7]. Yet another active field imposing patterns to repetitions is symmetry detection [8–13].

Some state-of-the-art matching methods are compatible with the detection of repeated structures [14–19]. Others sacrifice this feature in order to be distinctive in practical applications [20]. Ideally, the best suited matching method for repetitions should return fully connected groups of similar key points whenever repeated objects are present in images; also, it should not connect to unrelated objects.

This paper proposes a method free from any *a priori* geometric assumption on the repetition patterns. Figure 1 describes the main steps of our proposal. Our input data is a graph of auto-correspondences in the considered image, obtained by applying a reformulation of SIFT [21] presented in [19] (based on the *a-contrario* procedure [22]) between the image and itself. Our bottom up method starts by identifying sub-graphs corresponding to repetitions of small object pieces (called atoms), then groups these atoms to form repeated objects. An atom in our approach is an image patch

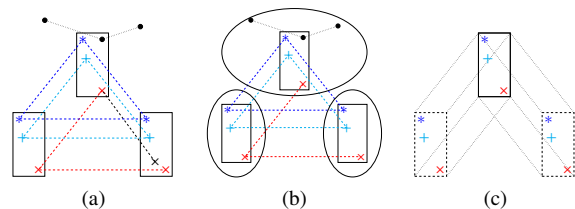


Fig. 1: Proposed steps for automatic detection of repeated objects. (a): Creating graphs of repeated atoms and filtering out mutations (black \times); (b): Grouping of key points for graphs of equal cardinality; (c): A repeated object is identified if geometry consistency is assessed among groups.

(a local element), equipped with a key point (the center's coordinates) and a local description. Repetitions are identified regardless of any *a priori* pattern, but the presence of several atoms in an object is required to validate the geometric transformation between instances of the object, and therefore identify this object. At this point, two atoms may be connected by a path of strong similarity scores, while not corresponding to the exact same physical part of an object (see Figure 2). We call these, mutations. Unfortunately, these mutations often occur between two spatially close atoms, and if not treated, they may harm the detection process. To avoid this situation, we treat these mutations by splitting the sub-graphs containing overlapping atoms.

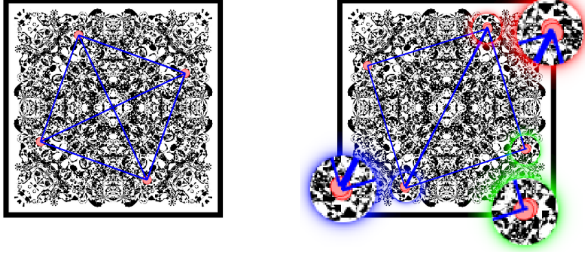
2. GRAPHS OF AUTO-CORRESPONDENCES IN IMAGES

In this section we introduce the mathematical objects that will be useful in the paper. Let $V = \{v_1, \dots, v_n\}$ be a set of atoms in an image \mathbf{u} , and let w_{ij} be the similarity score between the local descriptors from atoms v_i and v_j . We define $G = (V, E)$, the non-directed graph with set of nodes V and set of edges E . Edges represent the detected auto-correspondences between atoms. An edge between nodes v_i and v_j , denoted by $v_i \leftrightarrow v_j$, is weighted by w_{ij} .

Definition 1. Let $\tilde{G}_1 = (\tilde{V}_1, \tilde{E}_1)$ and $\tilde{G}_2 = (\tilde{V}_2, \tilde{E}_2)$ be two sub-graphs of $G = (V, E)$. We define the merging operation between two graphs, \tilde{G}_1 and \tilde{G}_2 , as the graph

$$\tilde{G}_1 \oplus \tilde{G}_2 := (\tilde{V}_1 \cup \tilde{V}_2, \tilde{E}_1 \cup \tilde{E}_2) \subset G.$$

Definition 2. Let $A \subset V$, we say A is a connected component if and only if any two nodes in A are connected by a path of positive edges in E and there is no path of positive edges connecting A and A^c .



(a) (Ideal behavior) A fully connected component (4 nodes, 6 edges) from the similarity graph, presenting one and only one atom.
 (b) (Undesired behavior) A connected component (7 nodes, 7 edges) from the similarity graph, presenting three atom mutations; see the three pairs of overlapping atoms (i.e. spatially close atoms).

Fig. 2: SIFT connected components.

Definition 3. Let $\tilde{G} = (\tilde{V}, \tilde{E})$ be a sub-graph of $G = (V, E)$. We denote

$$\text{Int}(\tilde{G}) := \{m \in E \mid m \text{ connects two nodes of } \tilde{V}\} = \tilde{E},$$

and

$$\text{Bd}(\tilde{G}) := \{m \in E \mid m \text{ connects a node of } \tilde{V} \text{ to a node of } \tilde{V}^c\},$$

the interior and boundary sets of \tilde{G} , respectively.

Remark 1. Both interior and boundary sets of a graph defined above are composed of edges, not nodes.

Definition 4. We say that \mathcal{G} is a partition of $G = (V, E)$ if and only if

$$\tilde{G} \in \mathcal{G} \Rightarrow \tilde{G} \subset G; \quad \bigcup_{\tilde{G} \in \mathcal{G}} \text{Int}(\tilde{G}) = E;$$

and

$$\tilde{G}_1, \tilde{G}_2 \in \mathcal{G} \Rightarrow \text{Int}(\tilde{G}_1) \cap \text{Int}(\tilde{G}_2) = \emptyset.$$

Definition 5. Let $G = (V, E)$ be a graph. Let $\{x \leftrightarrow y\} = m \in E$. We define the singleton graph as: $S(m) := (\{x, y\}, m) \subset G$.

Definition 6. Let \mathcal{G} be a partition of G . We denote by \mathcal{G}_m , $m \in \text{Int}(G)$, the only element $\tilde{G} \in \mathcal{G}$ such that $m \in \text{Int}(\tilde{G})$.

3. RETRIEVING ATOMS IN IMAGES

Small local elements, called atoms, are the unbreakable pieces that compose objects. Ideally, a perfect matching method should return fully connected components in the similarity graph whenever atom repetitions are observed (see Figure 2a); in those cases, any node can act as the class representative for the atom repetition. Unfortunately, it is often the case with existing matching methods that more than one atom repetition is present in a connected component. This is due to mutations, see Figure 2b, which are often observed with spatially close atoms. Also, atom repetitions do not always appear as fully connected components, due to missed matches. Nevertheless, the intuition supports that all ‘copies’ of an atom should belong to a unique connected component.

If not treated properly, mutations from spatially close atoms are a nuisance to the detection process, for they may point to a bigger

(and misleading) number of repetitions, see Figure 2b. In order to avoid the aforementioned nuisance, we encourage the splitting of connected components harboring overlapping atoms. Other types of mutation not harming the inferred number of repetitions are more easily handled as geometry consistency will be imposed in the next steps.

In this section we want to identify atoms to the interior of some similarity graph. Each interior should be free from mutations, so it can be considered to harbor several instances of only one atom. For that, a pre-processing of the raw similarity graph is required. Usually, clustering algorithms, related to the minimization of a functional (i.e. RatioCut, Ncut, etc), are used for this purpose. However, the main drawback of best-performing clustering algorithms is the added time in computations. Furthermore, it is not straightforward for classic clustering algorithms to add a free-from-mutation constraint to the functional. With this in mind, we define a simple functional, well suited to our problem, and propose a heuristic to optimize it.

Let $\mathcal{G} = \{G_1, \dots, G_k\}$ be a partition of graphs. Starting from the thinnest partition \mathcal{G} (i.e. composed of isolated matches), the merging procedure will be guided by the following functional:

$$f(\mathcal{G}) := \sum_{G \in \mathcal{G}} [f_I(G) + f_O(G)], \quad (1)$$

where

$$f_I(G) = \frac{1}{\sum_{e \in E} w_e},$$

and

$$f_O(G) = \begin{cases} 0, & G \text{ has no overlapping atoms} \\ \infty, & G \text{ has overlapping atoms} \end{cases}$$

The functional f will encourage the merging of graphs resulting in higher cumulative similarities and no overlapping atoms. Algorithm 1 proposes an heuristic to minimize the functional appearing in Equation 1. It ensures that sub-graphs in the resulting partition will not exhibit the undesired behavior of Figure 2b. In order to minimize the proposed functional, Algorithm 1 keeps track of the interior and boundary sets after each merge with the following formulas:

$$\begin{aligned} \text{Int}(\tilde{G}_1 \oplus \tilde{G}_2) &= \text{Int}(\tilde{G}_1) \cup \text{Int}(\tilde{G}_2), \\ \text{Bd}(\tilde{G}_1 \oplus \tilde{G}_2) &= \text{Bd}(\tilde{G}_1) \cup \text{Bd}(\tilde{G}_2) \setminus \text{Int}(\tilde{G}_1 \oplus \tilde{G}_2). \end{aligned}$$

Algorithm 1 Greedy heuristic to minimize Equation 1

input: M - set of Matches. It includes similarity measurements.

start:

$M.\text{SORT}()$ ▷ Order matches with respect to similarity measurements

$\mathcal{G} = \bigcup_{m \in M} S(m)$ ▷ Initial partition of singleton-graphs

$f^* = \infty$ ▷ f^* serves to identify when \mathcal{G} is not being modified

while $f(\mathcal{G}) < f^*$ **do**

$f^* = f(\mathcal{G})$

foreach $m \in M$ **do**

▷ Matches with higher similarity scores are analyzed first

$m_1^* \in \arg \min_{m_1 \in \text{Bd}(\mathcal{G}_m)} f(\mathcal{G}_m \oplus \mathcal{G}_{m_1}) - f(\mathcal{G}_m) - f(\mathcal{G}_{m_1})$

if $f(\mathcal{G}_m \oplus \mathcal{G}_{m_1^*}) < f(\mathcal{G}_m) + f(\mathcal{G}_{m_1^*})$ **then**

$\mathcal{G} = \mathcal{G}_m \oplus \mathcal{G}_{m_1^*} \cup \mathcal{G} \setminus \{\mathcal{G}_m, \mathcal{G}_{m_1^*}\}$

▷ Update the graph partition

return: \mathcal{G}

4. FROM ATOMS TO OBJECTS

In the previous section we have addressed the problem of identifying repetitions of unbreakable pieces of objects, atoms. These atom repetitions are provided by Algorithm 1 as a partition of graphs, \mathcal{G} . In this section, we pass from repeated atoms to the detection of repeated objects.

4.1. Pre-labeling object instances

Intuitively, an object atom is supposed to be present in all instances of the object in question, i.e., the number of object instances equals the number of instances of any of its atoms. Let Υ_{N_G} be the set of all graphs in \mathcal{G} with fixed cardinality N_G , representing all atoms that repeat N_G times in the image. For a fixed number of instances N_G , we define the most meaningful object repetition as the one that possesses the more atoms in Υ_{N_G} . We assume that no image zone should belong to two instances of an object at the same time.

In this section, we look for a function L that assigns to each node of a graph in Υ_{N_G} , a tentative instance of the most meaningful N_G -times-repeated object. Let us define such a function, called labeling function, as

$$L: \Upsilon_{N_G} \times \{1, \dots, N_G\} \rightarrow \{1, \dots, N_G\}, \quad (2)$$

$$(G, i) \mapsto c$$

where the i -th node of G is assigned to the object instance c . We say that G is *unconsensual* if and only if $\exists i, j, i \neq j$, such that $L(G, i) = L(G, j)$. We also define the consensual score as:

$$L_{score}(L) := \sum_{G \in \Upsilon_{N_G}} \prod_{c=1}^{N_G} |\{i : L(G, i) = c\}|, \quad (3)$$

where the inner expression in the sum equals zero if two or more nodes of a graph are assigned to the same cluster by L ; and one otherwise.

In practice, the labeling function can be provided by one of the following clustering algorithms: Spectral Clustering on KNN graphs, Gaussian Mixture Models clustering, or even the simple KNN clustering algorithm. We argue that the set of all nodes is not difficult to cluster, as our graphs are based on information provided by a distinctive matching method. Indeed, a simple KNN algorithm already provides fair enough results, see Figure 3.

Algorithm 2 presents how a generic clustering method is used to create labeling functions L in the presence of graph outliers. In the following we describe this process for a simple clustering based on Voronoi regions. First, we select one graph G at random from Υ_{N_G} and propose it as center of each N_G cluster. We then count the number of graphs from Υ_{N_G} whose nodes belong to one and only one Voronoi region. Finally, we iterate this process and keep the clusters (with centers at G) providing the biggest number of graphs in consensus with them. As depicted in Figure 3, each node from each graph is then labeled with a reference to the cluster to which it belongs. As one can imagine, these labels might not yet represent repetitions of objects due to spurious graphs. They will be filtered out in the next section.

For simplicity, we assume that all unconsensual graphs (0-labeled by Algorithm 2) are removed from Υ_{N_G} and that all nodes are rearranged so as to

$$L(G, i) = i, \quad \forall i \in \{1, \dots, N_G\}.$$

Algorithm 2 Generic pre-labeling of nodes

input: Υ_{N_G} - All graphs of matches having N_G nodes; N_{iters} - Number of iterations; FINDCLUSTERS - Clustering method.

start:

for $i \in \{1, \dots, N_{iters}\}$ **do**

 Select G at random from Υ_{N_G} .

 Define L_i with the help of FINDCLUSTERS, initialized at G .

$i^* = \arg \max_i L_{score}(L_i)$.

Set L_{i^*} labels of non consensual graphs to 0.

▷ The 0 label acts as an unconsensual flag.

return: L_{i^*} .

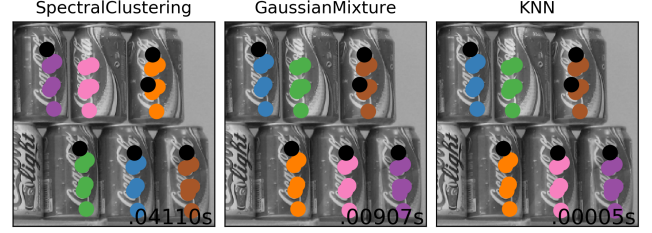


Fig. 3: Pre-labeling graphs nodes for Υ_6 . Labeling provided by spectral clustering, gaussian mixture models and KNN. Random colors indicates appartenance to clusters. Black nodes belong to a non consensual graph, as there are two nodes belonging to the same cluster (the top right cluster).

4.2. Geometry consistency

The standard RANSAC (RANDOM SAMPLE CONSENSUS) algorithm [23] computes the parameters fitting a mathematical model from observed data in the presence of outliers. Numerous improvements have been proposed in the literature for RANSAC, see [24–28], but the core idea behind them remains.

In the case of homography estimation, the classic RANSAC algorithm returns the homography η_j computed in iteration j having the largest consensus of inliers among all iterations. The j -iteration can be briefly described in two steps:

1. (Model fitting) Randomly select s matches $(x_i \leftrightarrow y_i)_{i=1, \dots, s}$ from the set of all matches (M_T) and compute the homography η_j that yields the best fit.
2. (Model consensus) Count the number of matches from M_T that are within a distance threshold of σ (i.e. counting inliers).

We propose to detect a repeated object from pre-labeled graphs by finding graphs in consensus with respect to multiple homographies. These homographies describe the transformations of a query object to all other present instances of it. The labeling function L in Equation 2 assigns to each node (i.e. atom) from a fixed graph, a label. These labels indicate an initial proposition of object instances, but without geometric validation. We describe in Algorithm 3 how to identify homography consistent copies of an object from graphs in Υ_{N_G} and a pre-labeling function. For simplicity, we select the query object to be among nodes assigned to the first label ($c = 1$) and all other labels relate to different instances of this object. The homography fitting function HFITTING $_{N_s}$ of Algorithm 3 can be based on raw matches ($N_s = 4$) or on the affine information ($N_s = 2$). Notice that $\alpha_{max} = (\infty, \infty, \infty, \infty)$ implies the classic score for homography consensus in Equation 4.

Algorithm 3 REP-RANSAC

input: Υ_{N_G} - All graphs of matches having N_G nodes; HFITTING_{N_s} - a homography model estimator from N_s matches; N_{iters} - Number of iterations; α_{max} - vector of maximal affine discordanace; σ - maximal distance error threshold; N_s - Number of samples needed for Homography fitting.

start:

for $i \in \{1, \dots, N_{iters}\}$ **do**

 Randomly select G_1, \dots, G_{N_s} from Υ_{N_G}

for $c = 2, \dots, N_G$ **do**

$$\left[\begin{array}{l} \eta_c^i := \text{HFITTING}_{N_s}(\mathbf{x}_{G_1}^1 \leftrightarrow \mathbf{x}_{G_1}^c, \dots, \mathbf{x}_{G_{N_s}}^1 \leftrightarrow \mathbf{x}_{G_{N_s}}^c) \\ \mathbb{A}_c := \mathbb{1} \left\{ \alpha(A_E(\mathbf{x}_{G_1}^1 \leftrightarrow \mathbf{x}_{G_1}^c), A_{\eta_c^i}(\mathbf{x}_{G_1}^1)) < \alpha_{max} \right\} \end{array} \right]$$

 ▷ where $\mathbf{x}_{G_c}^c$ correspond to

 the G node pre-labeled to cluster c , α is the affine discordanace measure appearing in [28], $\alpha(A_E(\mathbf{x}_{G_1}^1 \leftrightarrow \mathbf{x}_{G_1}^c))$ are the estimated affine maps from $\mathbf{x}_{G_1}^1$ to $\mathbf{x}_{G_1}^c$, and $A_{\eta_c^i}(\mathbf{x}_{G_1}^1)$ are the first order Taylor approximations of η_c^i around $\mathbf{x}_{G_1}^1$.

$$S_i \left(\left\{ \eta_c^i \right\}_c \right) = \sum_{G \in \Upsilon_{N_G}} \prod_{c=2}^{N_G} \mathbb{1} \left\{ \|\eta_c^i(\mathbf{x}_G^1) - \mathbf{x}_G^c\|_{L^2} \leq \sigma \right\} \times \mathbb{A}_c \quad (4)$$

$i^* = \arg \max_i S_i$

return: $\left\{ \eta_c^{i^*} \right\}_{c \in \{2, \dots, N_G\}}$

4.3. Detecting meaningful objects across cardinalities

Up until now and for each set Υ_{N_G} , a most significant repeated object is proposed and validated. In this section, we order these propositions by their Number of False Alarms (NFA), based on the *a-contrario* procedure [22]. Notice that we do not use the NFA to automatically set up parameters as done in [24]; instead we use it to measure meaningfulness across cardinalities.

For each possible cardinality N_G , we define the associated NFA of its most meaningful object (computed in Algorithm 3) as,

$$NFA \left(\left\{ \eta_c \right\}_c \right) = (n - N_s) \binom{n}{k} \binom{k}{N_s} P(k, n) \quad (5)$$

where $n = |\Upsilon_{N_G}|$, $k = S \left(\left\{ \eta_c \right\}_c \right)$ (the function S appears in Equation 4) and the probability of a random graph to be an inlier of the model is expressed as

$$P(k, n) = \mathbb{P} \left(\left\{ \|\mathbf{X}\|_{L^2} \leq \sigma \right\} \cap \left\{ \mathbf{A} < \alpha_{max} \right\} \right)^{(N_G-1)(k-N_s)}.$$

Indeed, if there are k graphs inliers, potentially all N_s out of them yield the correct configuration. Each one has a probability

$$\mathbb{P} \left(\left\{ \|\mathbf{X}\|_{L^2} \leq \sigma \right\} \cap \left\{ \mathbf{A} < \alpha_{max} \right\} \right)^{(N_G-1)}$$

of being inlier according to the background model, see Equation 4. Through the assumption of independence among graphs, the probability that all k of them are inliers is the above quantity powered by $k - N_s$, since N_s ones were used for estimation are automatically inliers (they have error 0). The number k of inliers is usually not known in advance, so all values of k are tested (from $N_s + 1$ to n), which explains the factor $(n - N_s)$ in Equation 5.

Equation 5 allow us to assign a natural order between all detected objects by Algorithm 3. Figure 4 shows these ordered detections across cardinalities. Clearly, a strong NFA (i.e. $\log \text{NFA} \ll 0$)

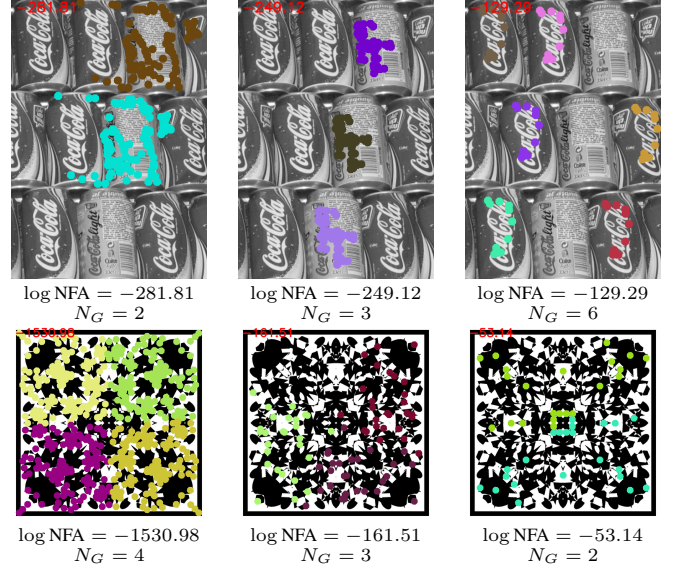


Fig. 4: In descending order from left to right, the three most significant detections. Each instance of an object is randomly colored.

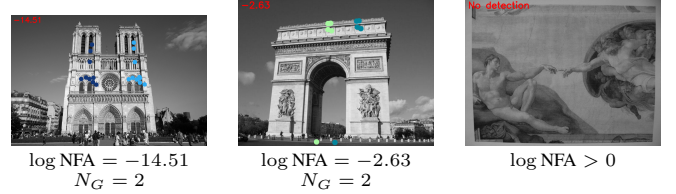


Fig. 5: Most significant repeated objects with weak NFAs.

supports the evidence of a repeated object. On the other hand, weak NFA signatures (i.e. $\log \text{NFA} \approx 0$) shows poor evidence of repeated objects. If for a fixed image all $\log \text{NFA} > 0$, then a no detection is reported. Examples of most significant repeated objects with weak NFAs can be found in Figure 5.

5. CONCLUSIONS

We have presented a methodology to identify object repetitions based on graphs derived from multiple SIFT matches. The SIFT method can be replaced by any other local matching method. The underlying idea is simple: if several local descriptions of a repeated object stand out (for the similarity score), then a coherence between their associated graphs points out repetitions of a same object. The pipeline for detecting object repetitions starts by analysing each detected similarity graph, associated with the repetition of a local atom. Then, an initial proposition of the most significant repeated object is deduced from graphs of equal cardinality, and its geometric inconsistencies filtered out to provide a final proposition of "repeated object". The validity of the object's repetitions is determined by computing an NFA. Clearly, in order to keep detecting more repeated objects, a simple workaround would consist in removing all geometrically consistent graphs and iterate the pipeline over the remaining graphs till no object repetition is detected. This will be explored in the future.

6. REFERENCES

- [1] Y. Liu, R. T. Collins, and Y. Tsin, "A computational model for periodic pattern perception based on frieze and wallpaper groups," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 3, pp. 354–371, 2004.
- [2] C. Wu, J.-M. Frahm, and M. Pollefeys, "Detecting large repetitive structures with salient boundaries," in *European conference on computer vision*. Springer, 2010, pp. 142–155.
- [3] P. Müller, G. Zeng, P. Wonka, and L. Van Gool, "Image-based procedural modeling of facades," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 85, 2007.
- [4] T. Leung and J. Malik, "Detecting, localizing and grouping repeated scene elements from an image," in *European Conference on Computer Vision*. Springer, 1996, pp. 546–555.
- [5] J. Liu, J. Mundy, and A. Zisserman, "Grouping and structure recovery for images of objects with finite rotational symmetry," in *Proc. Asian Conf. on Computer Vision*, 1995, vol. 1, pp. 379–382.
- [6] L. Van Gool, T. Moons, M. Proesmans, and A. Oosterlinck, "Groups, fixed sets, symmetries, and invariants," in *Proceedings, International Conference on Image Processing*. IEEE, 1995, vol. 3, pp. 356–359.
- [7] F. Schaffalitzky and A. Zisserman, "Geometric grouping of repeated elements within images," in *Shape, contour and grouping in computer vision*, pp. 165–181. Springer, 1999.
- [8] A. Turina, T. Tuytelaars, and L. Van Gool, "Efficient grouping under perspective skew," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE, 2001, vol. 1, pp. I–I.
- [9] S. Wenzel, M. Drauschke, and W. Förstner, "Detection of repeated structures in facade images," *Pattern Recognition and Image Analysis*, vol. 18, no. 3, pp. 406–411, 2008.
- [10] V. Patraucean, R. Grompone von Gioi, and M. Ovsjanikov, "Detection of mirror-symmetric image patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 211–216.
- [11] G. Loy and J.-O. Eklundh, "Detecting symmetry and symmetric constellations of features," in *European Conference on Computer Vision*. Springer, 2006, pp. 508–521.
- [12] H. Cornelius and G. Loy, "Detecting bilateral symmetry in perspective," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE, 2006, pp. 191–191.
- [13] H. Cornelius, M. Perdoch, J. Matas, and G. Loy, "Efficient symmetry detection using local affine frames," *Image Analysis*, pp. 152–161, 2007.
- [14] F. Cao, J. L. Lisani, J. M. Morel, P. Musé, and F. Sur, *A Theory of Shape Identification*. Springer, 2008.
- [15] J. Rabin, J. Delon, and Y. Gousseau, "A statistical approach to the matching of local features," *SIIMS*, vol. 2, no. 3, pp. 931–958, 2009.
- [16] R. Grompone von Gioi and V. Pătrăucean, "A contrario patch matching, with an application to keypoint matches validation," in *ICIP*, pp. 946–950. 2015.
- [17] M. Rodríguez, G. Facciolo, R. G. von Gioi, P. Musé, J.-M. Morel, and J. Delon, "Sift-aid: boosting sift with an affine invariant descriptor based on convolutional neural networks," in *2019 26th IEEE International Conference on Image Processing (ICIP)*, Sep 2019.
- [18] M. Rodríguez and R. G. V. Gioi, "Affine invariant image comparison under repetitive structures," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct 2018, pp. 1203–1207.
- [19] M. Rodríguez, J. Delon, and J.-M. Morel, "Fast affine invariant image matching," *IPOL*, vol. 8, pp. 251–281, 2018.
- [20] P. Doubek, J. Matas, M. Perdoch, and O. Chum, "Image matching and retrieval by repetitive patterns," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 3195–3198.
- [21] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.
- [22] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt Theory to Image Analysis*. Springer, 2008.
- [23] 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*, vol. 24, no. 6, pp. 381–395, 1981.
- [24] L. Moisan, P. Moulon, and P. Monasse, "Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers," *IPOL*, vol. 2, pp. 56–73, 2012.
- [25] L. Moisan, P. Moulon, and P. Monasse, "Fundamental Matrix of a Stereo Pair, with A Contrario Elimination of Outliers," *IPOL*, vol. 6, pp. 89–113, 2016.
- [26] 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*, vol. 35, no. 8, pp. 2022–2038, 2013.
- [27] M. Rais, G. Facciolo, E. Meinhardt-Llopis, M. J.-M., B. A., and C. B., "Accurate motion estimation through random sample aggregated consensus," *CoRR*, vol. abs/1701.05268, 2017.
- [28] M. Rodríguez, G. Facciolo, R. Grompone von Gioi, P. Musé, and J. Delon, "Robust estimation of local affine maps and its applications to image matching," in *WACV*, 2020.