A 3D Deformable Model-Based Framework for the Retrieval of Near-Isometric Flattenable Objects using Bag-of-Visual-Words

Rindra Rantoson **Adrien Bartoli**

Université d'Auvergne, Clermont-Ferrand, France

Corresponding author: Rindra Rantoson

rindra.sanders@udamail.fr

September 8, 2017

Abstract

We introduce a 3D deformable model-based framework for the retrieval of near-isometric flattenable objects using keypoints and BoVW (Bag-of-Visual-Words). By 3D deformable model we mean a texturemapped 3D shape which may deform isometrically. We assume that such a model is available for each object in the database. We exploit the 3D deformable models at the training and the retrieval phases. For our first contribution, we exploit the possibility of generating synthetic data from the 3D deformable models to define a new BoVW model for the database object representation. Our model chooses an optimal per-object representation by maximizing each object's mean average precision. The maximization is done over multiple candidate representations which are generated using the criteria of keypoint repeatability, weight discriminance and stability. Our second contribution is the use of SfT (Shape-from-Template) to facilitate geometric verification at the retrieval phase, for a few objects hypothesized using the new BoVW model. Existing methods use a rigid model, such as the fundamental matrix, or a simple deformable model based on semi-local constraints. SfT however is a physics-based method which uses an object's 3D deformable model to reconstruct its isometric 3D deformation from a single input image. The output of SfT thus directly provides a geometric verification score. A byproduct of our work is to extend the scope of SfT. The proposed object retrieval framework is used to provide SfT with a few object hypotheses which may be quickly tested for the 3D deformable object selection. Performance evaluation on synthetic and real images reveals the benefits of our retrieval framework using a database with size varying between 20 and 1,000 objects. The use of the new BoVW model and SfT versus the BoVW baseline and a rigid model improves the retrieval performance by 4.2% and 11.3% with p-values of 5×10^{-6} and 7×10^{-30} respectively.

Contents

1 Introduction

Object retrieval aims at finding which objects of a database are present in a query. It has been studied for various data types, typically regular 2D images, RGB-D images and full 3D models. The object database contains object descriptors, which originate from one data type. Similarly, the query has one data type, possibly different from the database's. Object retrieval methods thus address several combinations of object database and query image data types. Extensive studies and important progress have been recently made in the 3D-3D case [Feng et al., 2016; Pickup et al., 2016; Sahillioğlu and Kavan, 2016] and in the 2D-2D case [Arandjelović and Zisserman, 2012b; Sivic and Zisserman, 2003; Veltkamp et al., 2013]. However, the 3D-2D case with deformable objects has received significantly less attention [Alcantarilla and Bartoli, 2012; Blanz and Vetter, 2003; Magnenat et al., 2015; Ricard et al., 2005]. In this case, the object database is a collection of 3D deformable models, and the query is a simple, plain RGB image. By 3D deformable model, we mean a texturemapped 3D shape which may deform isometrically. One can think of augmented reality applications such as animated and interactive books like in [Magnenat et al., 2015] and tourism magazines/maps. For these applications, a page is considered as an object, so the pages of the book form the database. In [Magnenat et al., 2015], the query image shows the page containing the colored character. The changes are immediately visible on the 3-D model of the character as the child colors.

Existing works that build on 3D deformable models do not exploit their full potential [Alcantarilla and Bartoli, 2012; Magnenat et al., 2015]. More precisely, they use keypoints and count the number of correspondences between a model and the query image to decide the presence of an object. [Alcantarilla and Bartoli, 2012] registers each 3D deformable model from the database to the query image in turn, while [Magnenat et al., 2015] uses a voting scheme, which we hereafter call the *voting template scheme*, followed by geometric verification. These strategies do not allow the object database to scale in size.

We propose an integrated system to deformable object retrieval in the 3D-2D case. Our system is designed to handle flattenable objects. These are objects which can be made physically flat without causing significant stretching, such as pieces of paper, cloth and garment. The principle of our system however could be applied to non flattenable objects. Similarly to previous works [Alcantarilla and Bartoli, 2012; Magnenat et al., 2015], we use keypoints and a correspondence counting criterion to decide the presence of an object. Our two main contributions lie in how we embed the 3D deformable models in a retrieval framework, namely BoVW (Bag-of-Visual-Words) combined with the TF-IDF (Term Frequency-Inverse Document Frequency) weighting scheme. In the baseline BoVW framework, features are extracted from images and classified into visual words. Each object is then represented as a vector of visual words weighted using the TF-IDF statistic. We call such a vector the *TD (Template Descriptor)* and the set of such vectors for all the database objects

a *TDS (Template Descriptor Set)*. The retrieval phase then follows three main stages: initial objectwise hypotheses generation, global object hypotheses ranking and geometric verification. At the first stage, a similarity score is computed between each object and the query using the TDS. At the second stage, the objects are ranked according to their similarity score. At the third stage, a geometric verification which further reranks the strongest object hypotheses is carried out, using their spatial consistency. Since geometric verification may be computationally expensive, only a small subset of the objects are selected as candidates for re-ranking, by requiring that the initial BoVW score be strong enough.

Our first contribution exploits the power of synthetic data generation at the training phase to learn a novel representation model called *CTDS* (*Combined TDS*). CTDS chooses an optimal per-object representation among multiple representations named *extended TDS*, which are generated from three *base TDS* by applying repeatability, discriminance and stability criteria, as illustrated in figure 1. The three base TDS are respectively named T_1^{base}, T_2^{base} and T_3^{base}. We show experimentally that CTDS, compared to the baseline using a single TDS, significantly improves performance by 4.2% with a p-value of 5×10^{-6}. Compared to the voting template scheme inspired from [Magnenat et al., 2015], our BoVW-based retrieval framework provides a better trade-off between performance and runtime. Unexpectedly, our framework gives even better performance for a database with fewer than 100 objects using less than half of the number of descriptors as dictionary size. We obtained a MeanAP of 89.1% with a dictionary of 4,000 words using our framework versus a MeanAP of 82.9% with 8,529 descriptors using the voting template scheme, for a complexity in $\mathcal{O}(m \log n)$, where m and n are respectively the number of descriptors in the query image and in the object's texturemap for the voting template scheme while n is the dictionary size in our framework.

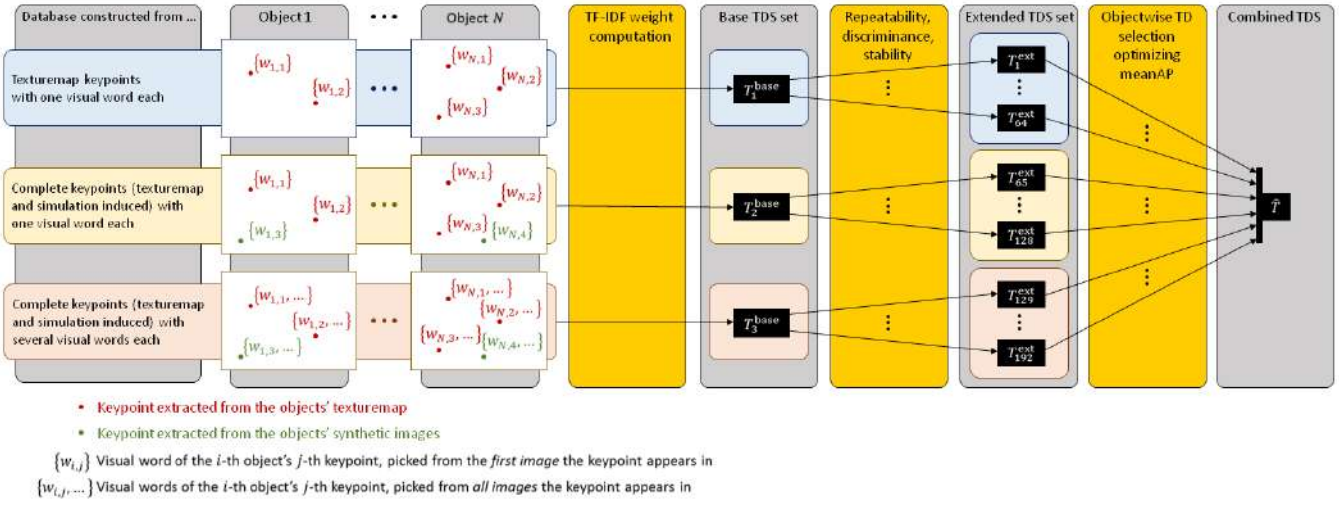


Figure 1: Learning strategy for the CTDS.

Our second contribution achieves physics-based geometric verification using SfT (Shape-from-Template). The goal is to select, from the few hypotheses generated from the first retrieval phase, which ones comply with a physics-based explanation of the image. For that purpose, we match each 3D deformable model (also called template in the SfT literature) to the image, by computing a 3D deformation satisfying the object's deformation law. We consider near-isometric deformations because their use in SfT is well-understood [Bartoli et al., 2015; Salzmann and Fua, 2011], though this is not a hard limit of our system. In contrast, most existing retrieval methods use a physics-based model only for rigid objects [Pilet et al., 2008; Sivic and Zisserman, 2003] and image-based semi-local constraints [Hartley, 1997; Lazebnik et al., 2006; Schmid and Mohr, 1997] or statistical models such as the DPM (Deformable Part-based Models) [Blanz and Vetter, 2003; Felzenszwalb et al., 2010] otherwise. Although flexible models such as graph and hyper-graph exist, they are time consuming and handle partially deformations [Collins et al., 2014; Lee et al., 2011; Leordeanu and Hebert, 2005; Yan et al., 2015]. We show experimentally that using SfT for geometric verification improves the retrieval performance significantly compared to two rigid models or to the absence of geometric verification. The use of SfT versus the rigid model based on the fundamental matrix and the graph matching in [Collins et al., 2014], respectively improves retrieval by 11.3% and 7.9% with p-values of 7×10^{-30} and 5×10^{-17}. Its absence reduces performance by approximately 20%.

2 Related Work

Existing approaches to retrieval follow one of two major models. The first model uses image color directly. The second model uses keypoints. The CNN (Convolutional Neural Network) [Girshick, 2015; He et al., 2014; Simonyan and Zisserman, 2014] and BoVW are the most prominent and popular representatives of each model. We have chosen the latter as baseline to combine retrieval with SfT. The reason is that current effective SfT methods use keypoints. We show that they thus naturally integrate with BoVW, which also uses keypoints. By exploiting the 3D deformable model, we propose improvements of the BoVW model and the geometric verification, which strengthen the BoVW framework against difficult imaging conditions such as strong occlusion, clutter and illumination variation.

2.1 Baseline BoVW and Variants

Long used in text and document retrieval, the bag-of-words model was first used as BoVW in image-based retrieval in [Sivic and Zisserman, 2003]. In BoVW, a visual vocabulary is constructed offline. SIFT descriptors [Lowe, 2004] extracted from the whole image dataset are clustered in visual words using k-means [Nister and Stewenius, 2006]. At retrieval, each image is represented by quantizing its local descriptors into

the predefined visual vocabulary, resulting in frequency vectors combined with TF-IDF weights. The latter down-weights the words that commonly occur in the database. Objects are retrieved based on the normalized scalar product between the query and each database object's TF-IDF weight vector. However, retrieval may fail due to information loss occurring in descriptor quantization (causing corresponding descriptors to be assigned to different visual words), feature drop-out, substantial noise in the descriptors and the use of an inappropriate metric for descriptor comparison. To overcome quantization problems, the initial object representation was enhanced using soft-assignment, where descriptors are assigned to multiple visual words [Gemert et al., 2008; Jégou et al., 2009; Philbin et al., 2008; Zheng et al., 2014] or by introducing a novel IDF expression using an Lp-norm pooling technique [Zheng et al., 2013]. Other representation models, FV (Fisher-Vector) [Perronnin and Dance, 2007] and VLAD (Vector of Locally Aggregated Descriptors) [Jégou and Chum, 2012; Jégou et al., 2010b] were proposed for efficiency and accuracy. FV encodes an image by pooling local features based on the general Fisher kernel. VLAD, which may be viewed as a simplification of the Fisher kernel representation, aggregates the differences between the descriptors of the same distribution and its assigned visual word. Hamming embedding [Jégou et al., 2008, 2010a] improved the approximation of the original descriptors. Learning was used to select different quantization parameters, such as the quantization variability [Makadia, 2010; Mikulik et al., 2013], the metrics [Philbin et al., 2010] and the descriptors [Winder et al., 2009] which here are different than SIFT. [Arandjelović, 2012; Arandjelović and Zisserman, 2011] also use different descriptors, the boundary descriptors, to retrieve so-called smooth objects. Hierarchical k-means [Nister and Stewenius, 2006], approximate k-means [Philbin et al., 2007] and recursive k-means tree [Pilet and Saito, 2010] were used for efficiency. To mitigate the effect of feature loss and to deal with noisy descriptors, the initial pipeline was strengthened with query expansion [Arandjelović and Zisserman, 2012b; Chum et al., 2007, 2011; Jégou et al., 2008; Philbin et al., 2008; Z. Yongwei et al., 2012], feature augmentation [Arandjelović and Zisserman, 2012b; Turcot and Lowe, 2009], multiple queries [Arandjelović and Zisserman, 2012b; Chen et al., 2012; Qi and Luo, 2016] and other re-ranking variations [Jégou et al., 2009; Qin et al., 2011]. These methods are used to improve recall. Query expansion issues new queries by averaging the weight vectors from spatially verified regions. In [Arandjelović and Zisserman, 2012b], a discriminative query expansion is carried out using machine learning to learn a weight vector for re-querying. An SVM classifier is trained from the spatially consistent and the most dissimilar results as, respectively, the positive and the negative samples, that is further used for ranking in terms of the distance away from the decision boundary. Since all the query expansion methods rely strongly on the spatially consistent proposals validated by the baseline approach, improvement is hardly gained. Feature augmentation consists in augmenting offline the images in the database with all features of images containing

the same view of the object. The BoVW representation of each image is augmented with the visual words of its neighbors based on an image graph [Arandjelović and Zisserman, 2012b; Turcot and Lowe, 2009] representing the relationship between dataset images. The concern is on the expanded low-level features while the relationships between dataset images contained in the image graph, which are more reliable in exploring relevant images, are relatively ignored. Multiple-query search is generally proposed to cope with the single-query problem by using multiple images of an object for retrieval. For that purpose, [Arandjelović and Zisserman, 2012a] introduced several simple rules to merge the results of each single query while [Chen et al., 2012] combined a discriminative query expansion with certain effective learning methods. Nevertheless for this last one, the classifier training phases are overly time-consuming for real-time retrieval.

We have adopted feature augmentation as well as soft-assignment by exploiting synthetic data unlike the existing approaches. They use a limited number of real images [Arandjelović and Zisserman, 2012b; Turcot and Lowe, 2009], to obtain a good trade-off between speed and efficiency while optimizing the retrieval performance without any additional query expansion method. We perform feature augmentation when defining the second and the third base TDS, for which a subset of synthetic data features are back-projected to the texturemap. We perform soft-assignment when defining the third base TDS by taking into account all the variations of the visual words assigned to the corresponding descriptors through the synthetic data, instead of using a limited number of visual visual words assigned to the descriptor with $(r = 3)$-nearest neighbors in [Philbin et al., 2008]. Again, the advantage of having synthetic data, generated thanks to the 3D deformable model, facilitates the traceability of keypoints which enables us to know with better accuracy the possible variations of the visual words of the corresponding descriptors, rather than assuming a fixed number of the closest visual words as in [Philbin et al., 2008]. The soft-assignment affects the calculation of weights in the TDS but not the query weight vector, allowing its easy manipulation for any subsequent use. Unlike the query expansion method in [Arandjelović and Zisserman, 2012b] for which a weight vector is learned for re-querying, our method learns a weight vector for each database object at the offline training phase to define the proposed CTDS representation without an additional query expansion phase in the standard pipeline. The specificity of our learning approach consists in the selection of the extended weight vector, generated from the base TDS set using discriminance, repeatability and stability criteria, to maximize the per-object retrieval performance. We use SIFT features and filter their assignment to the closest visual words using Lowe's ratio test [Lowe, 2004]. Since SfT crucially relies on keypoints, boundary descriptors [Arandjelović, 2012; Arandjelović and Zisserman, 2011] which are adapted to objects with limited texture, are not directly relevant in our framework.

2.2 Geometric Verification

A range of geometric verification methods has been proposed to boost the performance of object-based image retrieval at large scale. They discard unreliable correspondences in a given pair of images and are carried out by evaluating the consistency of spatial transformations between the image regions containing the correspondences. Most of the spatial matching strategies estimate a geometric model (such as a plane homography, affine transformation and similarity) using a RANSAC-based method [Philbin et al., 2007; Sivic and Zisserman, 2003] or the Hough transform [Avrithis and Tolias, 2014; Grauman and Darrell, 2007; Jégou et al., 2008, 2010a; Leibe et al., 2008; Lowe, 2004; Shen et al., 2012; Zhang et al., 2011]. Spatial consistency is verified if the number of correspondences which fit the model, the inliers, is greater than a predefined threshold. A fixed geometric model related to the local, semi-local or global spatial constraints is generally assumed. For instance, the spatial constraint between the query and the retrieved image is modeled by an affine transformation between neighbouring matches in [Sivic and Zisserman, 2003] while [Philbin et al., 2007] exploit the local shape of features (local scale, orientation, affine parameters) to generate relative transformation hypotheses from single correspondences. The matching process of the latter is made deterministic by enumerating all hypotheses. [Jégou et al., 2008, 2010a] use a weaker geometric model based on a 2D affine transformation which filters matching descriptors that are not consistent in terms of angle and scale. Combined with the Hamming Embedding, this geometric constraint is integrated within an inverted file used during the first retrieval phase. Since the constraints are weak, a global geometric re-ranking is finally required. [Lowe, 2004] attempts to find a group of correspondences using the generalized Hough transform which vote for the same pose of an object: model location, orientation and scale. Local shape is exploited using single correspondences as in [Philbin et al., 2007]. [Shen et al., 2012] handle object rotation, scaling, viewpoint change and appearance deformation using a spatially constrained similarity measure which tolerates moderate object deformation. Similarly to [Zhang et al., 2011] and [Avrithis and Tolias, 2014], votes arise from single feature correspondences. Several scale and rotation transformations are applied to the query features and produce a 2D translation voting map for each database image. For all these methods, the model supports translation invariance only. Furthermore, if some approaches are parameter-free [Raguram and Frahm, 2011], others use the flexible models which are typically used in recognition such as [Carneiro and Jepson, 2007], graph and hyper-graph matching-based approaches [Collins et al., 2014; Leordeanu and Hebert, 2005] and [Lee et al., 2011; Leordeanu et al., 2011; Yan et al., 2015; Zass and Shashua, 2008]. [Carneiro and Jepson, 2007] use a flexible semi-local model for the identification of multiple groups of consistent correspondences using pairwise relationships between correspondences. This handles some non-rigid deformations. Graph matching approaches attempt to find correpondences between two feature

sets with a technique which depends on the considered order of relations between features. First order methods consider the node-wise unary compatibility between two point sets. For second order methods, the feature set is expressed as a graph involving nodes which represent features and edge weights which measure the similarity between nodes. The correspondence is ascertained when the structure similarity across two graphs is preserved [Collins et al., 2014; Leordeanu and Hebert, 2005]. The hyper-graph methods such as in [Lee et al., 2011; Leordeanu et al., 2011; Yan et al., 2015; Zass and Shashua, 2008], are characterized by higher-order relations between graphs. They are more robust to outliers and noise while presenting a better scale, rotation and deformation invariance. They come at a higher computational cost however. In [Avrithis and Tolias, 2014; Grauman and Darrell, 2007; Lazebnik et al., 2006; Vedaldi and Soatto, 2008], the Spatial Pyramid Matching model employs correspondence distributions over hierarchical partitions of the transformation space instead of pairwise computation. [Grauman and Darrell, 2007] map features to a histogram pyramid in descriptor space, and then match them in a bottom-up process. The approximation of similarities by bin size constitutes the benefit of the approach in terms of complexity. [Lazebnik et al., 2006] apply the same idea to the image space but in such a way that geometric invariance is lost. Recent works have combined the approach with the Hough transform [Avrithis and Tolias, 2014] using the local feature shape to generate votes. This endows invariance to similarity and the involved one-to-one mapping facilitates some flexibility to handle non-rigid motion and multiple matching surfaces or objects.

Unlike existing geometric verification methods elaborated for the rigid model [Philbin et al., 2007; Sivic and Zisserman, 2003] and statistical models adaptable to mildly non-rigid objects [Avrithis and Tolias, 2014; Carneiro and Jepson, 2007], we introduce a physics-based model, SfT, making use of 3D deformable models. State of the art single-object SfT methods handle noise and erroneous correspondences. For instance, erroneous correspondences can be removed by assuming that the surface is locally smooth and that its local topology must thus be preserved [Pizarro and Bartoli, 2012].

3 Framework Overview

3.1 General Points

The 3D deformable model is a cornerstone of our framework. We specifically address the training phase as well as geometric verification in the retrieval phase, as illustrated by figure 2. On the one hand, the availability of the 3D deformable model allows us to synthetically generate learning, validation and test datasets. We took advantage of this potential to introduce CTDS and to learn its components. The idea is to learn weights and a detection threshold for each object so as to maximise the performance of the first

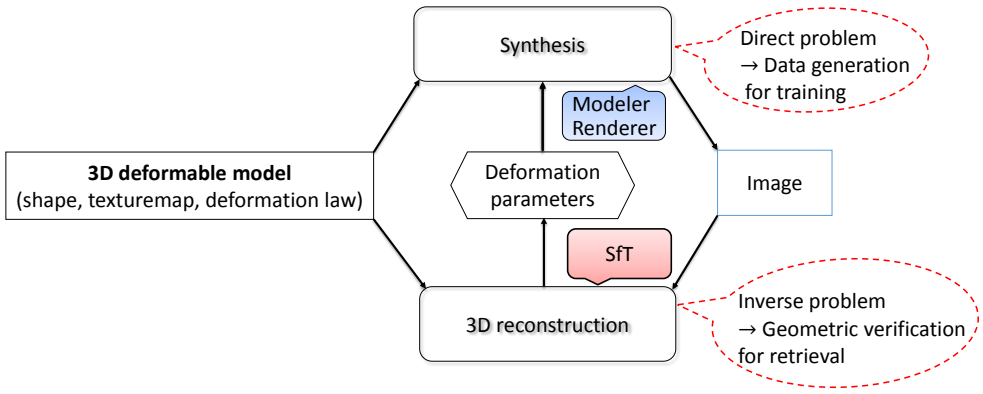


Figure 2: The 3D deformable model plays a keyrole in our framework, at the training and retrieval phases.

retrieval stage. On the other hand, the 3D deformable model is exploited for geometric verification using SfT. The 3D deformable model is matched to the query image by computing a 3D deformation satisfying the object's deformation law. The number of correspondences defines the geometric verification score.

The considered object keypoints[1] include the texturemap keypoints and the deformation-induced keypoints which both define *the complete keypoints* set. The *texturemap keypoints* are the set of keypoints detected in the set of texturemaps, and the *deformation-induced keypoints* are the set of keypoints which do not exist in the texturemaps and are extracted from synthetically generated learning images. The object is declared present if the score is higher than a learnt detection threshold. This threshold is specific to each object, due to the variation of the number and behaviour of keypoints per object. It is learnt at the training phase to satisfy a required minimum true positive rate or a maximum false negative rate, according to the application requirements. In practice, we have used a minimum true positive rate $TP = 90\%$.

3.2 Object Database

We consider six databases of different size: B_1 is a database of 20 objects, B_2 and B_3 are databases of 50 objects, B_4 is a database of 100 objects, B_5 is a database of 500 objects and finally, B_6 is a database of 1000 objects. The six databases are subsets of each other: $B_1 \subset B_2 \subset B_4 \subset B_5 \subset B_6$ and $B_3 \subset B_4$. B_1, B_2 and B_4 are defined from cover pages of various magazines as illustrated in figure 3 while B_5 and B_6 are additionally composed of some entire comics as illustrated in figures 4 and 5. Furthermore, B_3 is composed of 50 pages of the comics illustrated in figure 5. B_6 is the largest database which is small compared to the existing databases such as ImageNet containing more than a million images. This is because the applications and the context are different. Here, we study a 3D-2D retrieval framework which involves 3D models and images, not just images as in ImageNet. It was attempted to run SfT on an object database [Alcantarilla

[1]We use 'keypoint' as an image location, 'descriptor' as a vector describing a keypoint's local neighborhood, and 'feature' as a keypoint and its descriptor.

Figure 3: B_2 or the first fifty objects of B_4, B_5 and B_6.

and Bartoli, 2012]. However, this was a mere execution of SfT on all objects in the database, limiting the database size to between $5 - 10$ objects. The reason is that even though SfT runs in real-time for a single object [Collins and Bartoli, 2015; Östlund et al., 2012], it does not scale well with the number of objects being tested. Thus, the use of a database reaching the size of 1000 is a serious challenge for SfT and sufficient for many applications. For these applications, the training time is typically unimportant as the database is known a priori and fixed, but the testing time is critical.

The use of each database is explained as follows. The database B_4 of 100 objects is used to illustrate the CTDS generation and its comparison with T_1^{base}, CNN, FV and VLAD. B_4 is also used for SfT comparison with a graph matching approach and the rigid model using the fundamental matrix. Furthermore, the performance of our retrieval framework procedure stages is evaluated on $B4$. The other databases are used for performance comparison between T_1^{base} and VT which is the only existing retrieval approach in the context of SfT.

Figure 4: Some pages of a comics composing B_5 and B_6.

Figure 5: Some pages of a comics composing B_3, B_5 and B_6.

3.3 Training Phase

For a vocabulary of k words, we define $V_i = (t_1, ..., t_j, ..., t_k)^\top \in \mathbb{R}^k$ as the TD representing the i^{th} object in the database B. This is the TF-IDF weight vector whose component t_j is:

$$t_j = \frac{n_{ji}}{n_i} \log \frac{N}{n_j}, \tag{1}$$

with n_{ji} the number of occurrences of the j^{th} word in the i^{th} object, n_i the total number of words in the i^{th} object, N the number of objects in B and n_j the number of objects in B containing the j^{th} word. The set of TD for all objects in B defines the TDS, denoted $T = \{V_1, ..., V_N\} \in \mathbb{R}^{k \times N}$.

Our framework generates multiple such TDS and combines them into CTDS as follows (figure 1):

- **L1: Generation of the base TDS set (section 4.2).** We generate three base TDS denoted $T_1^{\text{base}}, T_2^{\text{base}}, T_3^{\text{base}} \in \mathbb{R}^{k \times N}$. The goal is to exploit keypoints detected in the texturemap and in synthetic views of the objects.

- **L2: Generation of the extended TDS set (section 4.3).** We generate the extended TDS denoted $T_1^{\text{ext}}, ..., T_{192}^{\text{ext}} \in \mathbb{R}^{k \times N}$ by applying keypoint discriminance, repeatability and stability criteria on the

base TDS set with various parameters.

- **L3: Definition of CTDS (section 4.4).** We select the extended TD that maximises the retrieval performance for each object. The set of selected TD for all objects forms CTDS, denoted $\hat{T} = \{\hat{V}_1, ..., \hat{V}_N\} \in \mathbb{R}^{k \times N}$.

- **L4: Learning the detection thresholds (section 4.4).** We learn two sets of detection thresholds. The first set is denoted $\xi = \{\xi_1, \xi_2, ..., \xi_N\}$ and is used at the first and second retrieval stages. The second set is denoted $\zeta = \{\zeta_1, \zeta_2, ..., \zeta_N\}$ and is used at the third retrieval stage.

3.4 Retrieval Phase

An image used to query the database follows the word assignment process whilst its TF-IDF weight vector V_q is extracted. The query image may contain none, one or several objects. For a given TDS T, which may be the optimal $\hat{T}$ or any other TDS, the three stages of our retrieval phase are:

- **D1: Initial objectwise hypotheses generation.** The presence of the i^{th} object is hypothesized if the similarity score s_i satisfies:

$$s_i \geq \xi_i \quad \text{with} \quad s_i = \frac{{V_i}^T V_q}{||V_i||||V_q||} \quad \text{and} \quad V_i \in T. \tag{2}$$

- **D2: Global object hypotheses ranking**. The hypothesized objects are ranked according to their respective normalized similarity score:

$$\hat{s}_i = s_i - \xi_i. \tag{3}$$

  This criterion contributes to the ranking performance by ensuring the homogeneity of scores issued of different TD, which may have inhomogeneous ranges of values.

- **D3: Geometric verification.** The R strongest object hypotheses are verified using SfT. We limit R to 10 and in practice R can be lower than 10 if D1 hypothesized less than 10 objects. Each object hypothesis' features are matched to the query image. The correspondences are then fed in a robust SfT estimator [Pizarro and Bartoli, 2012] which filters out the miscorrespondences. Denoting γ_i the number of correspondences related to the i^{th} object, we validate the hypothesis if:

$$\gamma_i \geq \zeta_i. \tag{4}$$

(a) (b) (c) (d)

Figure 6: (a)(b) Two simulated 3D deformations. (c)(d) A validation datum and a test datum for the two 3D deformations in (a) and (b) respectively combined with two different camera projections and backgrounds.

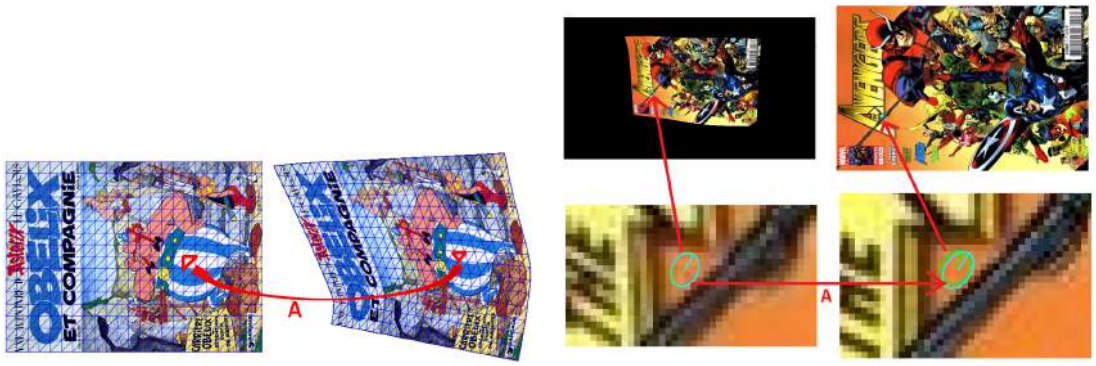
3.5 Data Generation

We use, without loss of generality, the database B_4 of $N = 100$ objects, to give figures on the size of the generated datasets. Various simulated 3D deformations with random parameters such as the ones in figures 6a and 6b are generated using the model from [Perriollat and Bartoli, 2013] for three types of datasets:

- The learning dataset (D_L) is defined from 1,200 deformations per object, without background, for a total number of 120,000 images. They result of the combination of 80 3D deformations with 15 camera projections. Illustrations are provided in the top left image of figure 7b and in figures 9c, 9d, 9e, 9f.
- The validation dataset (D_V) is defined from 1,200 deformations per object with background for a total number of 120,000 images. They result of the combination of 80 3D deformations with 15 camera projections both combined with 20 backgrounds per object. An illustration is provided in figure 6c.
- The test dataset (D_T) is defined from 300 deformations per object with background for a total number of 30,000 images. They are defined from 3D 20 deformations combined with 15 camera projections and 20 new backgrounds. An illustration is provided in figure 6d.

The output images are RGB images of size 720×1280 pixels. The learning dataset is used for building the vocabulary, generating the base TDS set, defining the deformation-induced keypoints and for the implementation of the repeatability and stability criteria. The validation dataset is used to build the CTDS and to learn the two sets of detection thresholds ξ and ζ.

Finally, the test dataset is dedicated to retrieval performance evaluation. We have initially used a visual vocabulary of 1,000 words built from the descriptors of the complete keypoints. We recall that the complete keypoints include the texturemap keypoints and the deformation-induced keypoints. For an object, the deformation-induced keypoints are defined by searching correspondences between a deformation image from the learning dataset and the associated texturemap. The keypoints of the deformation image are warped back to the texturemap assuming that the inverse of the warp function is locally approximated by an affine transformation, as illustrated by figure 7a. Redundant keypoints are then discarded using the overlap



(a) (b)

Figure 7: (a) A local affine transformation A between the isometrically deformed object image and its 3D deformable model; (b) Illustration of the overlap criterion [Mikolajczyk et al., 2005] to compare the geometry of keypoints.

criterion [Mikolajczyk et al., 2005], which consists in comparing the local regions of the two keypoints, as illustrated by figure 7b. Redundancy is found if the overlap area of the two regions is higher than a predefined threshold, which we choose here as 45%. The search space of redundant keypoints is limited by half of the minor axis of the back projected region. The deformation-induced keypoints form features by using the initial descriptors calculated from the generated image it first appears in.

4 Learning the Combined Template Descriptor Set

4.1 Strategy

Once a dictionary of k words $W = \{w_j\}$, $j \in [1, k]$, is constructed, the CTDS building procedure illustrated by figure 1 is carried out. It starts with the generation of three base TDS from which the extended TDS (figure 8) and CTDS (equation (6)) are constructed. More precisely, our approach is based on finding a per-object TD by maximizing its resulting mean average precision. For the i^{th} object, this is represented by the following maximization problem:

$$\max_{V \in \mathbb{R}^k} \text{MeanAP}_i^+(V). \tag{5}$$

We cannot solve this problem exactly because of the form of the objective function. Our approach is based on sampling the configuration space and relaxes the original problem (5) at two levels. Sampling is where the first level of relaxation occurs. Our sampling strategy is guided by priors known to be beneficial to feature selection from the literature, namely the three criteria of discriminance, repeatability and stability [Faheema and Rakshit, 2010; Gehler and Nowozin, 2009; Tirilly et al., 2009; Tsai, 2012]. Each of these criteria has a

free parameters representing a threshold, whose optimal selection leads to a complex combinatorial problem. We have discretized these thresholds, which represents the second level of relaxation in our approach. The discretization is based on experimental observations. We finally combine all parameter values to obtain a set of 192 candidate TD per object and select the optimal one by choosing the highest mean average precision. Importantly, our approach eventually uses the original objective function to find the TD. Assembling these TD of different types for all the N objects forms CTDS.

4.2 The Base TDS Set

For a database B, the three base TDS differ from each other according to the considered keypoints:

- T_1^{base} involves the texturemap keypoints. For each object, the descriptors of its texturemap keypoints are extracted and matched to visual words. The TF-IDF weights are then computed based on these visual words to form T_1^{base}.
- T_2^{base} involves the complete keypoints. For each object, the descriptors of its complete keypoints are extracted and matched to visual words. The TF-IDF weights are then computed based on these visual words to form T_2^{base}.
- T_3^{base} involves the complete keypoints and all their corresponding points in the learning dataset. For each object, the descriptors of its complete keypoints and the descriptors of all their corresponding points in the learning dataset are extracted and matched to visual words. All obtained visual words are then used to define the TF-IDF weights which thus take into account the variation of words for a given object keypoint.

4.3 The Extended TDS Set

Several extended TDS are generated from the base TDS set using three criteria. These include keypoint discriminance, repeatability and stability with respect to the learning dataset. They are applied separately or combined, as shown in figure 8. The application of a criterion consists of filtering the weights of the base TDS set with different parameters. Novel representations of the objects hence result from this filtering. The goal is to point out the specific characteristics of the objects through the considered criteria while representing and recognizing them accordingly. Several parameters could be chosen and would lead to the generation of several extended TDS, depending on the allowed computation and memory resources. The validation stage subsequently ensures the final selection of the optimal extended TD subset. In our implementation, we generate 192 extended TDS.

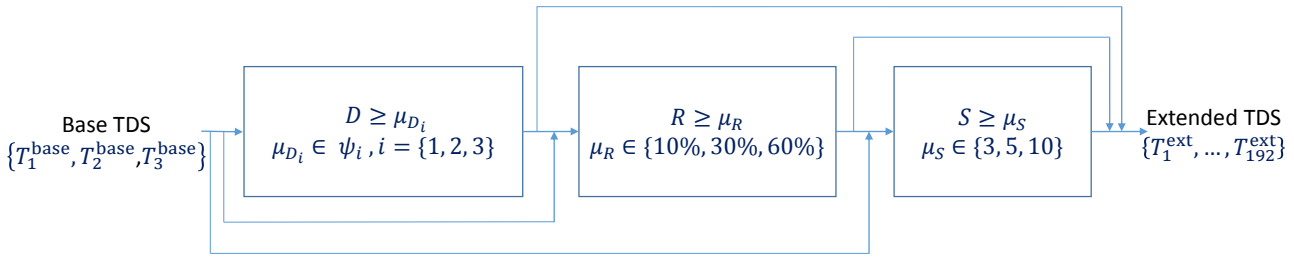


Figure 8: Generation of 192 extended TDS by applying keypoint discriminance D, repeatability R and stability S on the base TDS set $\{T_1^{\text{base}}, T_2^{\text{base}}, T_3^{\text{base}}\}$ with their respective thresholds μ_{D_i} (for which $\psi_1 = \{0.002, 0.005, 0.01\}, \psi_2 = \{0.002, 0.0035, 0.005\}, \psi_3 = \{0.001, 0.002, 0.003\}$), μ_R and μ_S.

Discriminance

The discriminance criterion directly selects high TF-IDF weights from the base TDS set to create an extended TDS for a given threshold μ_D. We recall that an object's visual word weight decreases with the visual word occurrences in the object and through all objects in the database. Since many low weights can perturb an object characterization, by keeping only visual words of high weights, objects endowed with discriminant visual words are expected to be pointed out by the generated TDS at retrieval. Different weight thresholds are set empirically based on the considered base TDS histogram which gives the weight distribution, as shown in figure 10a. For that case, three thresholds were chosen for each TDS of the base TDS set: $\mu_{D1} \in \{0.002, 0.005, 0.01\}$ for T_1^{base}, $\mu_{D2} \in \{0.002, 0.0035, 0.005\}$ for T_2^{base}, $\mu_{D3} \in \{0.001, 0.002, 0.003\}$ for T_3^{base}.

Repeatability

The repeatability criterion aims at characterizing an object by the visual words of its most repeatable keypoints. The repeatability of a keypoint is quantified by counting the number of times the keypoint appears through the learning dataset. The keypoints of the 50^{th} object (figure 9a) and associated words are illustrated by figures 9b, 9c, 9d, 9e and 9f. Keypoints with high repeatability are selected by applying a threshold μ_R. The corresponding words are selected from the base TDS to create a new extended TDS. Each object has its own point repeatability histogram, as shown in figure 10b and global thresholds are chosen for the three base TDS to generate an extended TDS set with repeatability $\mu_R \in \{10\%, 30\%, 60\%\}$.

Stability

The stability criterion allows one to generate an extended TDS set which exhibits objects of highly stable visual word points. The visual word associated to the same keypoint found in different images may vary

as illustrated by figures 9a, 9b, 9c, 9d, 9e and 9f. This may be due to the word assignment method, the used metric, the deformation and noise. For each object keypoint, the word variation of its corresponding keypoint in the learning dataset is counted. High stability occurs when the variability of the collected visual words is lower than a threshold μ_S. The words corresponding to stable keypoints are selected from the base TDS to create a new extended TDS. No significant change in the retrieval performance is directly observed by using different threshold values. However the combination of the stability criterion with the two previous ones is beneficial. Three global thresholds, $\mu_S \in \{3, 5, 10\}$ are thus used on the base TDS set. An example of a visual word stability histogram related to an object is provided in figure 10c.

4.4 The Combined TDS

The objective of this stage is to identify, for the i^{th} object, the extended TD which maximizes its retrieval performance. We use the mean average precision $\text{MeanAP}_i^+(V)$, as follows:

$$\hat{V}_i = \arg \max_{V \in \{V_{i,1}^{\text{ext}}, \ldots, V_{i,192}^{\text{ext}}\}} \text{MeanAP}_i^+(V) \quad \text{with} \quad i \in [1, N], \tag{6}$$

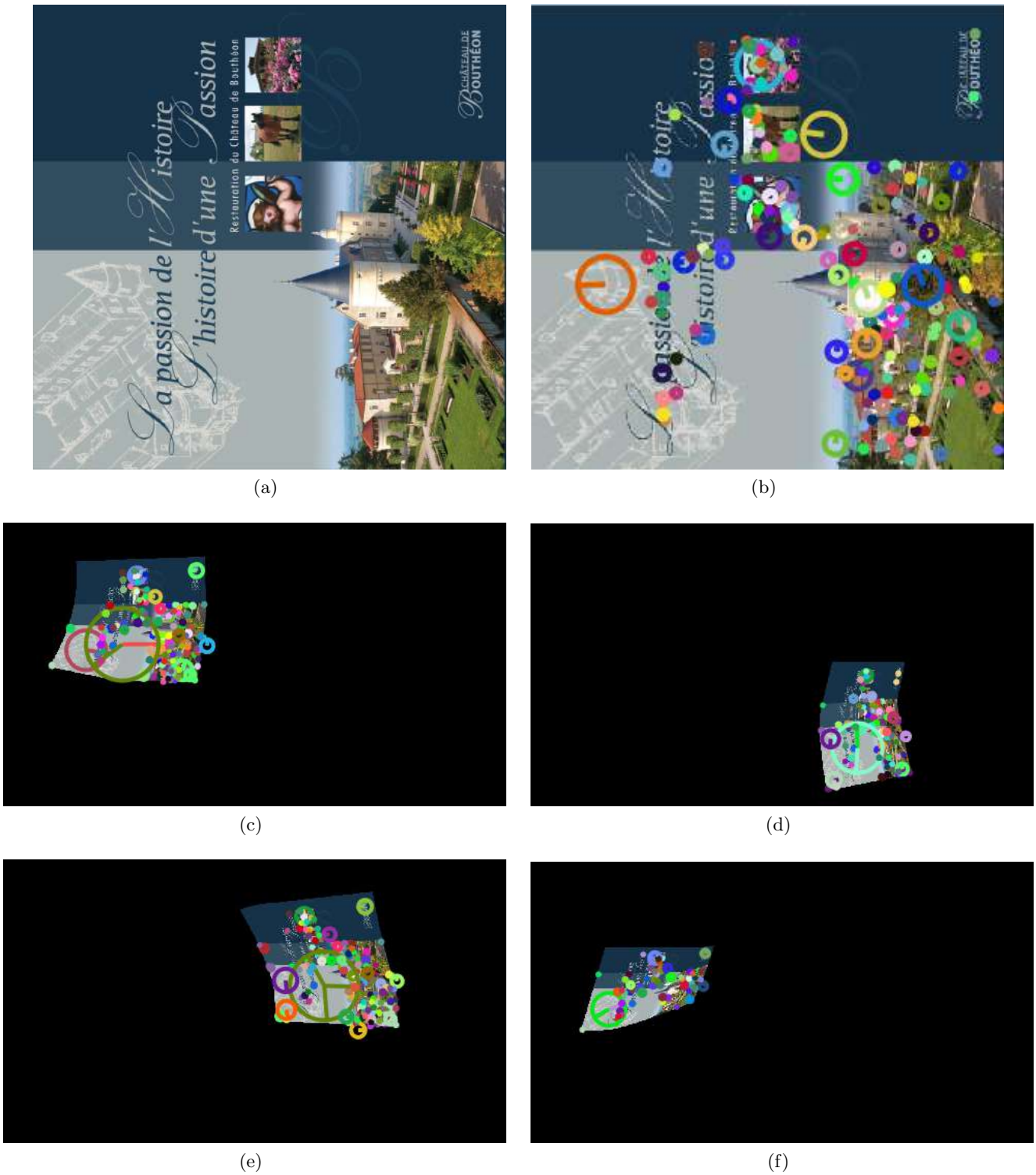
and $V_{i,l}^{\text{ext}} \in \mathbb{R}^k$ is the TD of the i^{th} object from the extended TDS T_l^{ext}, $l \in [1, 192]$. $\text{MeanAP}_i^+(V)$ is computed as follows. For each query image[2], the precision-recall curve is created and the average precision AP is computed as the mean of the precision across all recall rates. It can be visually interpreted as the area under the precision-recall curve. MeanAP_i^+ is computed by averaging AP related to the object's positive images. The set of selected extended TD forms CTDS. For each $\hat{V}_i$, we also learn the object detection threshold ξ_i which satisfies the required minimum true positive rate, based on the corresponding ROC curve. Figure 11b gives an overview of the selected ROC points of all objects for $TP = 90\%$.

4.5 Evaluation

The outcome of the selection stage uses 25 extended TD over 192 extended TD per object. Almost all criteria are beneficial. The effective thresholds are empirically determined by the optimization algorithm. The total training time is less than 12 days which includes time for:

- Synthetic image (learning and validation dataset) generation ($\approx$ 5 days);
- Dictionnary generation ($\leq$ 1h);
- Feature extraction and their assignement to visual words ($\approx$ 2h50);

[2]There is a total number of 120,000 query images defined from the validation dataset, which includes 1,200 positive images and $1,200 \times 99 = 118,800$ negative images per object.

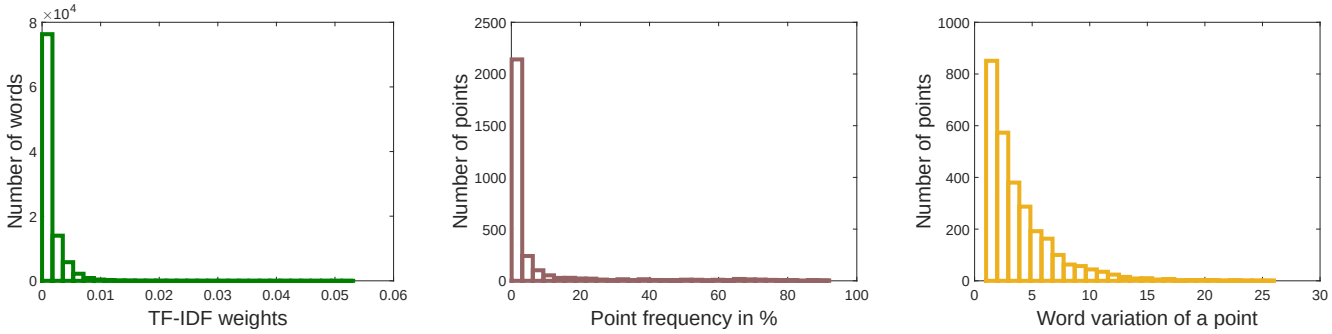


(a) (b)

(c) (d)

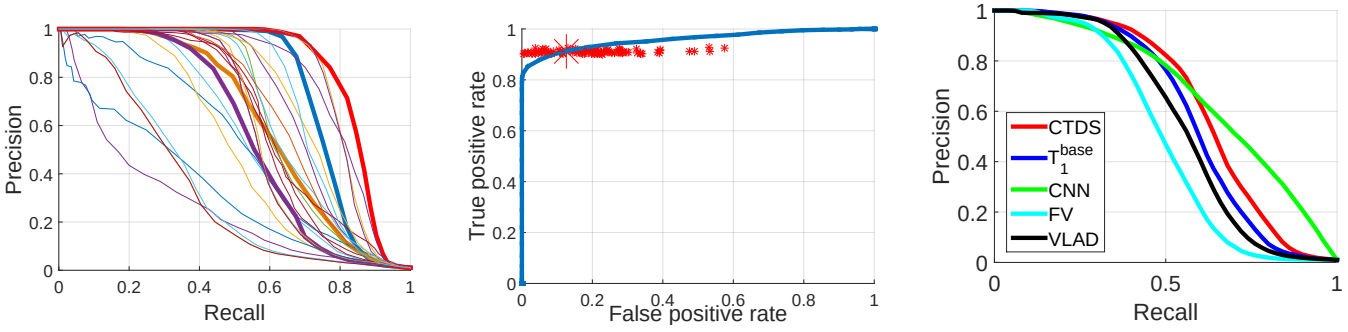
(e) (f)

Figure 9: (a) The 50^{th} object, its keypoints and the associated words (in different colored circles) extracted from the texturemap (b) and extracted from some of its learning data (c)(d)(e)(f) to illustrate the identification of repeatable and stable keypoints.



(a) Discriminance (b) Repeatability (c) Stability

Figure 10: (a) Histogram related to T_1^{base} showing the word distribution as a function of the TF-IDF weights. (b) Histogram showing the distribution of the 50^{th} object's points according to their frequency. (c) Histogram showing the distribution of the 50^{th} object's point according to their word variability.



(a) (b) (c)

Figure 11: (a) Precision-recall curves related to the 50^{th} object for some extended TD, with the selected extended TD in thick red, $V_{50,1}^{\text{base}}$ in thick blue (the baseline), $V_{50,2}^{\text{base}}$ in thick orange and $V_{50,3}^{\text{base}}$ in thick purple. (b) ROC curve (in blue) of the selected extended TD for the 50^{th} object and ROC points of all objects for the learnt thresholds for $TP = 90\%$. (c) Precision-recall curve comparison on D_T for all objects.

- Corresponding keypoints and their visual word tracks for defining T_3^{base} ($\leq$ 3 days);
- Base TDS and extended TDS generation ($\leq$ 5 minutes);
- Indexation of all images in the validation dataset ($\leq$ 5 minutes);
- MeanAP calculation and TD selection for each object ($\leq$ 1 minute);
- Threshold learning ($\leq$ 3 days).

Some parts such as synthetic image generation could be made drastically faster with the use of GPU based rendering. Note that for most applications, the training time is not critical.

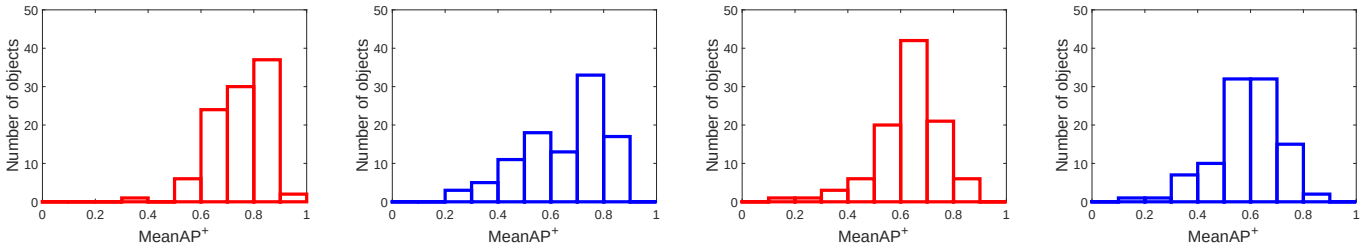
The performance of CTDS versus a unique TDS is first evaluated on the whole dataset. To this end, the mean and the median of the AP related to the whole dataset, MeanAP and MedAP, are computed on the validation dataset D_V and the test dataset D_T. A synthesis of the result is reported in table 1. Since the performance of T_2^{base} and T_3^{base} is low on D_V, the comparison is focused on T_1^{base} and CTDS on D_T. Concerning MeanAP and MedAP, an enhancement of 4.2% and 4.5% is observed for CTDS with a p-value of 5×10^{-6} for MeanAP. Figure 11c illustrates this difference for each detector. The histograms in figures 12a, 12b, 12c and 12d report the distribution of $\text{MeanAP}^+(V)$ per object on D_V and on D_T. Secondly, a comparison of CTDS to CNN using the VGG16 network [Simonyan and Zisserman, 2014], FV and VLAD, both using $k = 16$, has been carried out on D_T. CTDS outperforms these approaches with a MeanAP difference of 4% to CNN, 7.7% to VLAD and 14.5% to FV for p-values lower than 5×10^{-6}.

CTDS represents each object by its specificity thanks to the training phase which uses visual word discriminance, keypoint repeatability and stability criteria, and for which weights are learned independently to find the optimal per-object representation. This endows CTDS with the capability to handle deformation, viewpoint change and occlusion better than T_1^{base}, FV, VLAD and CNN. This also gives CTDS the capability to handle illumination variations and blur when working on real images, see §5.3.

A thorough analysis of CTDS' evaluation has been carried out for the first three cases defined in §5.1, involving several combinations of our retrieval framework stages as illustrated in figures 13a, 13b and 13c. For all cases, the use of CTDS enhances performance with an improvement of respectively 4%, 6.1%, 6.4% for MeanAP (see table 2) with p-values lower than 1×10^{-5} for all cases.

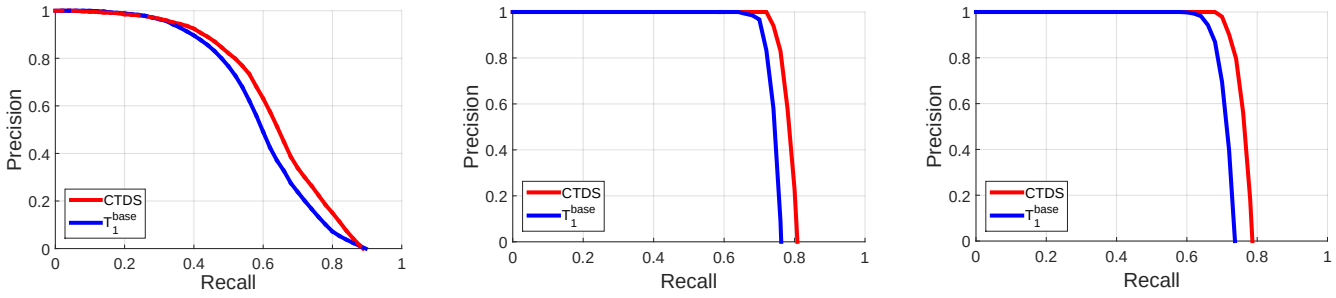
Statistics	CTDS	T_1^{base}	T_2^{base}	T_3^{base}	CNN	FV	VLAD
MeanAP							
on D_V	**74.6**	65.2	37.2	25.5	–	–	–
on D_T	**63.1**	58.9	–	–	59.1	48.6	55.4
MedAP							
on D_V	**76.6**	70.1	33.8	19.9	–	–	–
on D_T	**64.3**	59.8	–	–	60.4	48.2	57.6

Table 1: Statistics on retrieval performance for different approaches, in percents (%).



(a) CTDS (b) T_1^{base} (c) CTDS (d) T_1^{base}

Figure 12: Histograms showing the object distribution according to their MeanAP$^+$ on D_V (a,b) and on D_T (c,d).



(a) Case 1 (b) Case 2 (c) Case 3

Figure 13: Precision-recall curves obtained using CTDS and T_1^{base} for three cases.

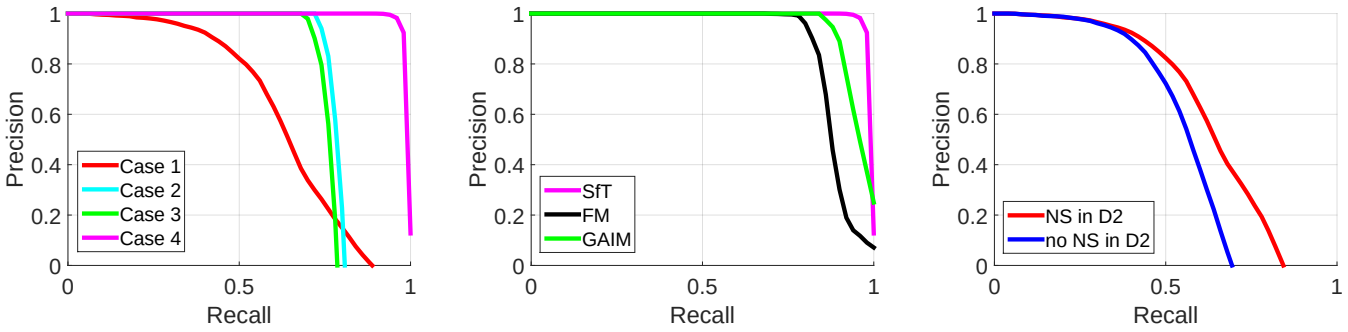
5 Experimental Results

5.1 Evaluation of CTDS-based BoVW

The performance of our retrieval framework is evaluated on D_T for several combinations of the three stages of the retrieval phase, in order to understand the influence of each of them:

- Case 1 : D1. Application of the first retrieval stage to all objects of the database, with $R = N$.
- Case 2 : D1 + D3. Application of the first retrieval stage combined with the third one. This last stage is applied on R object hypotheses from the first stage, with $R \leq N$.
- Case 3 : D1+D2+D3. Application of the whole retrieval procedure. The last stage is applied on the R strongest object hypotheses from the first and the second stages, with $R \leq 10$.
- Case 4 : D3. Direct application of SfT on all objects of the database, with $R = N$ [Alcantarilla and Bartoli, 2012]. This case also represents the direct application of the geometric verification method on all objects of the database, with $R = N$, used for comparison with SfT.

For each case, MeanAP is shown in figure 14a with the performance synthesis (MeanAP and MedAP), the respective induced costs (average time per image using a bi-processor Intel Xeon E5-2670 v3) and the average number of hypothesized objects R are given in table 2. The direct application of SfT in case 4 leads to the best retrieval performance with the highest cost. The number of hypothesized objects obtained from D1 varies between query images with a mean of 18.79 over 30,000 query images with a MeanAP of 81.9% when applying SfT (case 2) and a MeanAP of 80.1% when applying subsequently D2 and D3 (case 3). This is a good performance even though recalls are limited due to the thresholds respectively used in D1 and in D3. A slight retrieval performance difference is noticed for cases 2 and 3 with considerable difference in computation time and number of hypotheses R. The importance of the normalized score used in D2 is illustrated by figure 14c. We compared SfT with FM, the rigid model based on the fundamental matrix [Hartley, 1997] and with GAIM (Graph-based Affine Invariant Matching), the graph matching approach in [Collins et al., 2014]. A significant performance difference is observed in figure 14b and quantified by calculating the difference of MeanAP (as reported in table 2) which is equal to 11.3% for SfT compared to FM and 7.9% for SfT compared to GAIM with respectively p-values 7×10^{-30} and 5×10^{-17} in favour of SfT. In terms of speed, SfT is faster than FM and GAIM which was verified to be time consuming. For one image, SfT requires less than 5 seconds while GAIM needs more than a minute on average.



(a) (b) (c)

Figure 14: (a) Precision-recall curves obtained using CTDS for the four cases. (b) Precision-recall curves for the case 4 obtained using SfT, FM and GAIM. (c) Precision-recall curves obtained for D1+D2 according to the used criterion in D2 (NS stands for Normalized Scores).

Cases	MeanAP (%)	MedAP (%)	R	Time (s)
Case 1				
Using CTDS	**62.7**	**63.8**	100	0.07
Using T_1^{base}	58.7	59.4		
Case 2				
Using CTDS	**81.9**	**81.7**	≈ 19	0.28
Using T_1^{base}	75.8	75.5		
Case 3				
Using CTDS	**80.1**	**79.8**	≈ 9	0.20
Using T_1^{base}	73.7	73.5		
Case 4				
Using SfT	**99.0**	**99.3**	100	4.92
Using FM	87.7	87.7	100	5.04
Using GAIM	91.1	91.2	100	60.11

Table 2: Performance evaluation for 100 objects with a dictionary size of 1,000.

5.2 Comparison of T_1^{base} with the Voting Template Scheme

The BoVW baseline T_1^{base} has been compared with the voting template scheme (VT) adapted from [Magnenat et al., 2015] without geometric verification for both. The comparison is achieved for different sizes of the database while varying the dictionary for each. The idea is to quickly obtain performance result behaviour under different conditions between the baseline of BoVW T_1^{base} and VT, assuming that CTDS would give better performance than T_1^{base} based on the previous evaluation on CTDS. Thus, it helps to identify in which conditions CTDS outperfoms VT.

For VT, each descriptor in the input image votes for the template which has the closest descriptor, assuming that each template roughly has the same number of descriptors. The similarity score is then based on the number of collected template votes and objects are thus ranked accordingly. The comparison requires some adaptations. Here, SIFT descriptors are used instead of BRISK, and the vote is normalized by the number of template descriptors since it differs between templates.

The results are reported in figure 15a and table 3 for the first 20 objects, in figure 15b and table 4 for the first 50 objects and additionally in figure 15c and table 5 for a comics of 50 pages, in figure 16a and table 6 for the first 100 objects, in figure 16b and table 7 for the first 500 objects and finally in figure 16c and table 8 for 1,000 objects. We notice that the retrieval performance using T_1^{base} is higher than using VT for a database size lower than 100 with a dictionary size which is at least half of the total number of descriptors. However, for a database size greater than l00 objects, the retrieval performance using VT is the highest for any dictionary. This is probably due to the spread of the false votes. Indeed, the lower the database size, the higher the false positive rate induced by the concentration of the false votes onto an object. Furthemore, although VT has the best results from a certain database size, T_1^{base} gives a better compromise between performance and cost. Indeed, as illustrated by figures 16b and 16c, a dictionary size which is ten times lower than the total number of descriptors provides a great performance, very close to VT, recalling that the matching algorithm's complexity is approximately in $\mathcal{O}(m \log n)$, where m and n are respectively the number of the descriptors in the query image and in the object's texturemap for the voting template scheme while n is the dictionary size in our framework.

	T_1^{base}				**VT**
#VW	1,000W	2,000W	4,000W	5,000W	
MeanAP (%)	75.4	82.4	89.1	**90.0**	82.9
MedAP (%)	73.7	81.6	88.5	**89.2**	85.9

Table 3: Results for 20 different objects (8,529 descriptors).

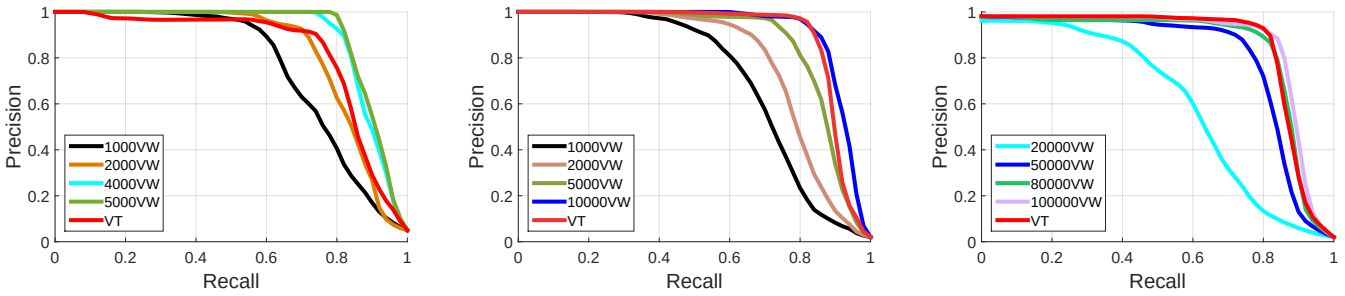


Figure 15: (a) Precision-recall curves related to 20 objects (8,529 descriptors); (b) Precision-recall curves related to 50 objects (19,944 descriptors); (c) Precision-recall curves related to 50 pages of a comics (19,6665 descriptors).

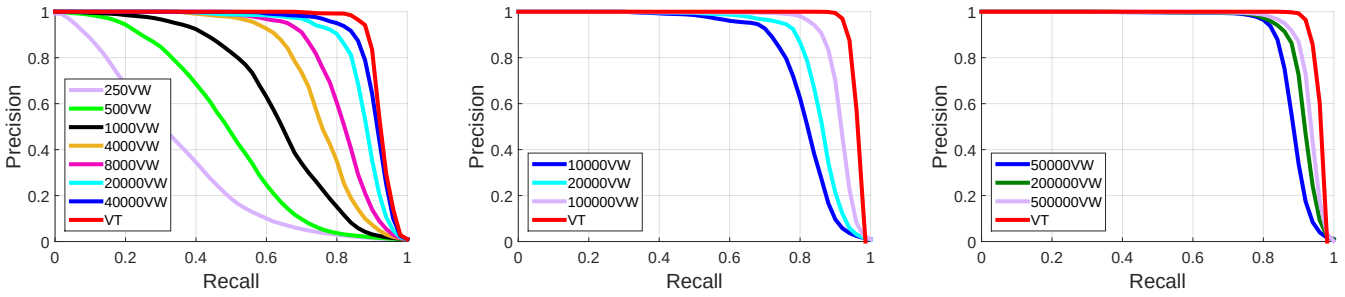


Figure 16: (a) Precision-recall curves related to 100 objects (45,063 descriptors); (b) Precision-recall curves related to 500 objects (1,060,395 descriptors); (c) Precision-recall curves related to 1,000 objects (2,697,181 descriptors).

	$T_1^{\mathbf{base}}$				**VT**
#VW	1,000	2,000	5,000	10,000	
MeanAP (%)	70.6	78.2	85.8	**91.2**	89.0
MedAP (%)	72.2	79.2	88.1	**92.5**	90.0

Table 4: Results for 50 different objects (19,944 descriptors).

	$T_1^{\mathbf{base}}$				**VT**
#VW	20,000	50,000	80,000	100,000	
MeanAP (%)	60.2	80.0	84.3	**86.6**	85.3
MedAP (%)	64.2	83.7	87.9	**89.0**	87.4

Table 5: Results for 50 objects from 50 pages of a comics (196,665 descriptors).

	$T_1^{\mathbf{base}}$							**VT**
#VW	250	500	1,000	4,000	8,000	20,000	40,000	
MeanAP (%)	32.6	47.9	58.9	75.0	80.7	86.7	90.0	**91.8**
MeanAP (%)	33.9	48.3	59.8	75.9	81.8	88.0	90.9	**91.9**

Table 6: Results for 100 objects (45,063 descriptors).

	$T_1^{\mathbf{base}}$			**VT**
#VW	10,000	20,000	100,000	
MeanAP (%)	80.5	85.0	90.7	**95.5**
MedAP (%)	81.9	86.1	91.3	**95.7**

Table 7: Results for 500 objects (1,060,395 descriptors).

	$T_1^{\mathbf{base}}$			**VT**
#VW	50,000	200,000	500,000	
MeanAP (%)	87.9	90.9	92.6	**95.5**
MedAP (%)	88.3	91.3	93.2	**95.7**

Table 8: Results for 1,000 objects (2,697,181 descriptors).

5.3 Results on Real Images

To evaluate the perfomance of our CTDS detector (the proposed BoVW model) on real images, we have compared it with T_1^{base} (the BoVW baseline) and VT (the voting baseline) using two different database sizes with respectively two different dictionaries:

- A database[3] size of 100 with a dictionary size of 1,000 for the first example.
- A database size of 50 with a dictionary size of 10,000 for the remaining examples.

Thus, for each example, three different proposals obtained using CTDS, T_1^{base} and VT, are presented, respectively followed by SfT filtered results. The considered real images are obtained from single shots for the first five examples and from video frames for the remaining examples for which the acquisition time is mentioned under each figure.

Results using the first database are almost similar for the three compared retrieval frameworks, figure 17a, with a slightly better performance in term of ranking for VT, which was expected from the performance evaluation on synthetic data shown in figure 16a and statistically quantified in table 6. Illustrations of feature extraction and matching are also provided for the first example, respectively in figures 17b, 17c and 17d, as well as the 3D reconstruction by SfT of the two detected objects in figures 18a and 18b. The use of the second database highlights the contributions of our framework while emphasizing the limitations of VT and T_1^{base} as expected according to the performance evaluation on synthetic data shown in figure 15b and quantified in table 4. Summary statistics are provided in table 9 for the three frameworks and the retrieval performance per object is shown in table 10. VT hardly detects object 5 as observed in figures 19a, 19d and in table 10, probably due to the lack of (distinctive) features, figures 19b and 19c. Furthemore, only

[3]The second fifty objects are presented in figure 23.

our CTDS based retrieval framework detects object 5 either in the presence of multiple objects in figure 20a or when strong warping occurs in figure 20c. However, for this last case, SfT does not validate object 5 hypothesized in the first retrieval proposals due to the insufficient number of matched features. This situation occurs also for object 8 in figures 22a and 22c, where only CTDS proposals contain the object despite the strong occlusion, although SfT does not retain it. CTDS indeed deals well with the occlusion of some objects, like the detection of object 43 in figures 20c and 21a and object 21 in figure 21b. For these cases, the occluded objects are then validated by SfT. BoVW can also deal with light occlusion depending on the object and the acquisition conditions such as in figure 20b while VT is the only approach which fails to detect the occluded object 43. CTDS also copes with blur as illustrated in figure 22c where the less blurred object 21 with the occluded object 8 are detected. Nevertheless, all frameworks fail when objects are strongly distorted, represented in low resolution, and/or strongly blurred such as object 36 in figures 21a, 21c and 22c. The retrieval performance of object 36 is the lowest in this video. We have noticed that in most cases of the video frames, CTDS provides the best performance on average (table 9), as the only framework which detects objects in figures 20c, 21c, 22a and 22c and the highest number of objects in figures 20a, 21a, 21b and 22b. Through these examples, the contribution of SfT is also pointed out.

Statistics	CTDS	T_1^{base}	VT
MeanAP (%)	**70.0**	57.3	45.8
MedAP (%)	**78.9**	58.5	44.2

Table 9: Statistics in percents (%) on retrieval performance on database B_2 of 50 objects.

Object Id	6	5	43	36	10	8	21
CTDS	94.5	**79.0**	**78.9**	35.9	**90.6**	**64.3**	**47.1**
T_1^{base}	**99.1**	58.4	40.6	**64.9**	63.7	37.3	37.1
VT	93.8	0	57.0	36.1	72.3	44.2	17.2

Table 10: MeanAP$^+$ in percents (%) per object.

6 Conclusion

We have presented a retrieval framework for quasi-isometric flattenable objects and for the applicability of 3D reconstruction by SfT with a large database. The availability of the 3D deformable model constitutes a cornerstone of our framework for the training and retrieval phases. Indeed, the prior knowledge endowed by the 3D deformable models motivated our choice to implement a learning strategy from synthetic data. The

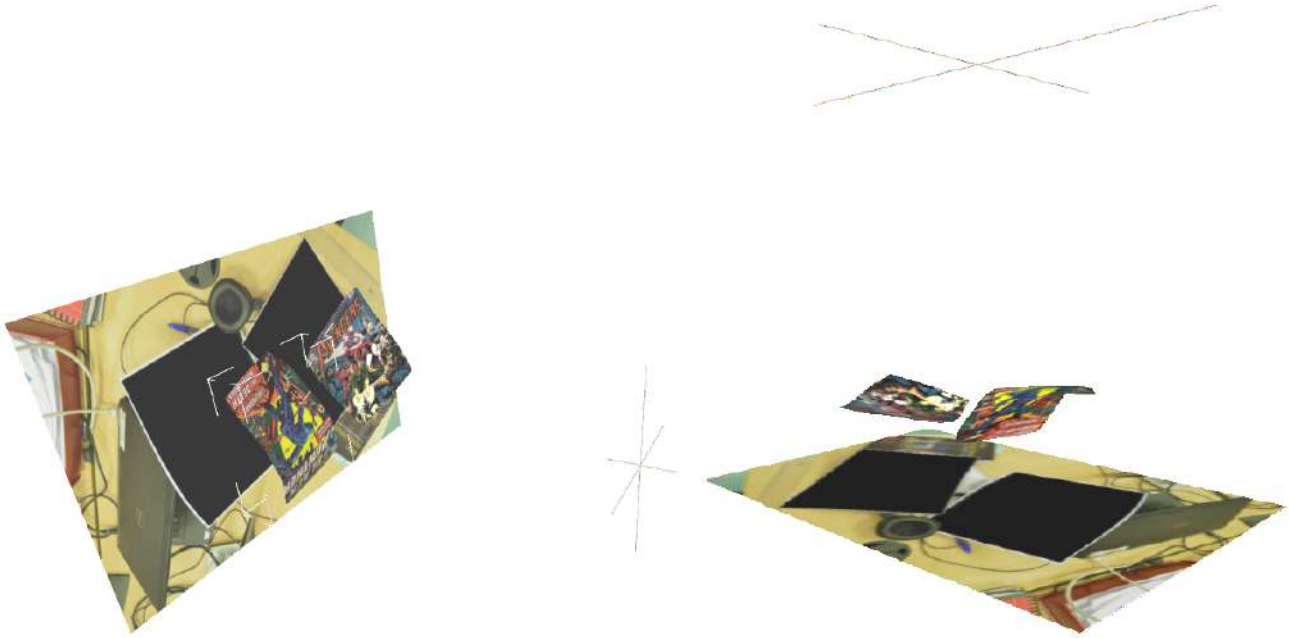
(a)

(b)

(c)

(d)

Figure 17: (a) Query image containing the 73^{th} and the 79^{th} object with the retrieval results obtained using three frameworks. (b) Feature extraction. (c)(d) Feature matches.



(a) (b)

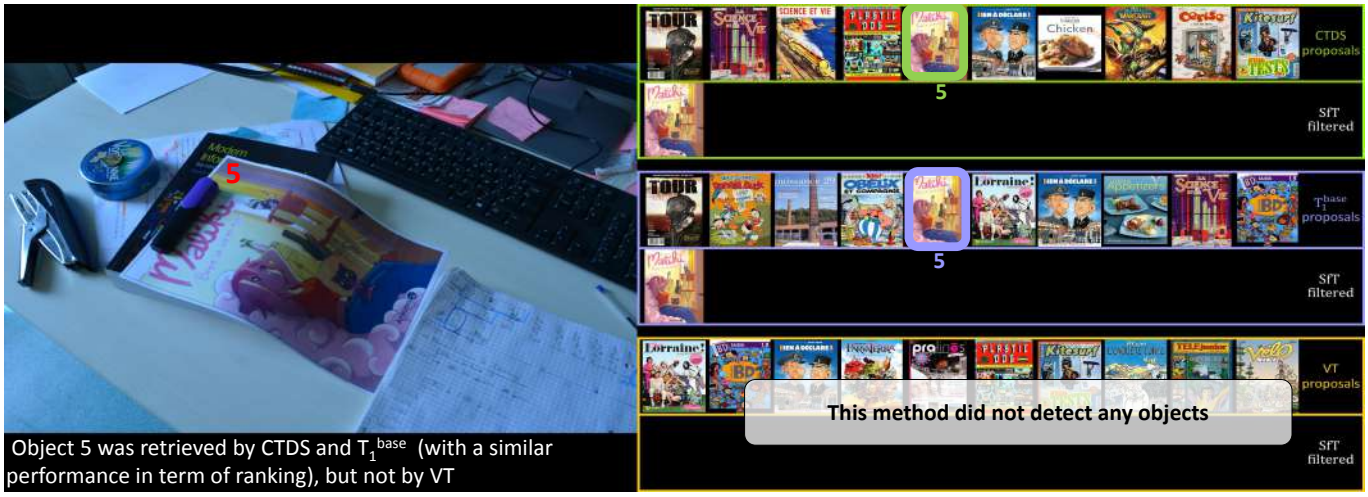
Figure 18: (a)(b) 3D reconstruction by SfT for two viewpoints.

idea is to learn an optimal representation for each object, leading to CTDS, a novel model of the database objects. The availability of the 3D deformable models also allows us to use SfT to implement physic-based geometric verification. On the one hand, we have shown that the use of SfT is more appropriate than the classical rigid model. On the other hand, we have shown the benefit of using CTDS instead of the existing single TDS. Moreover, compared to the voting template scheme, the introduction of an elaborated retrieval procedure provides a better compromise between performance and cost. Our retrieval framework which aims to serve SfT has also been evaluated on its interwoven cores. This has provided an overview on the expected results in terms of precision and the induced cost. The applicability of SfT via the proposed complete retrieval framework provides the best performance in terms of precision and time. To improve the geometric verification by SfT, a weighted vector associated with each feature could be considered, rather than simply counting the number of inlier correspondences. Consequently, an additionnal threshold related to the sum of weights could be trained.

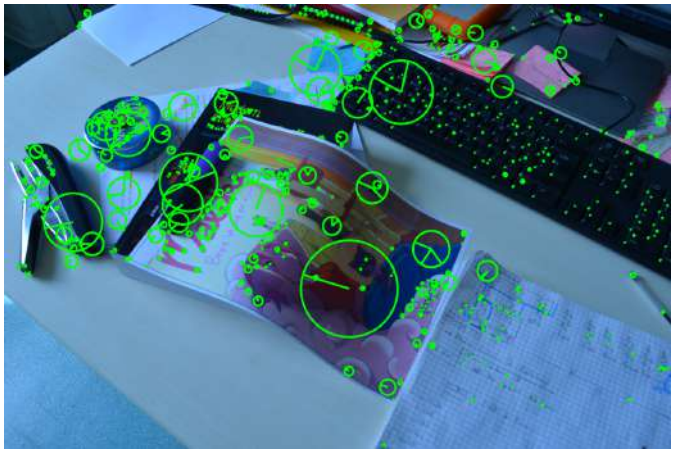
Acknowledgements. This research has received funding from the EU's FP7 through the ERC research grant 307483 FLEXABLE.

References

P. F. Alcantarilla and A. Bartoli. Deformable 3D reconstruction with an object database. In *BMVC*, 2012.



(a)



(b)

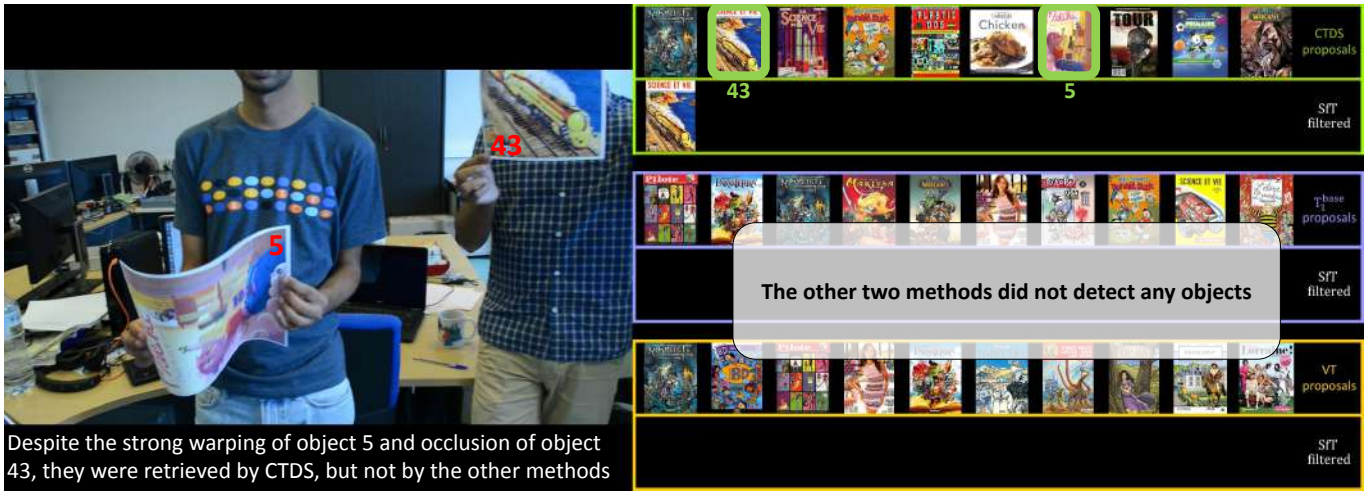
(c)

(d)

Figure 19: (a)(d) Query image containing object 5 from two points of view providing different proposals for each considered method. (b)(c) Features extraction for both.

(a)

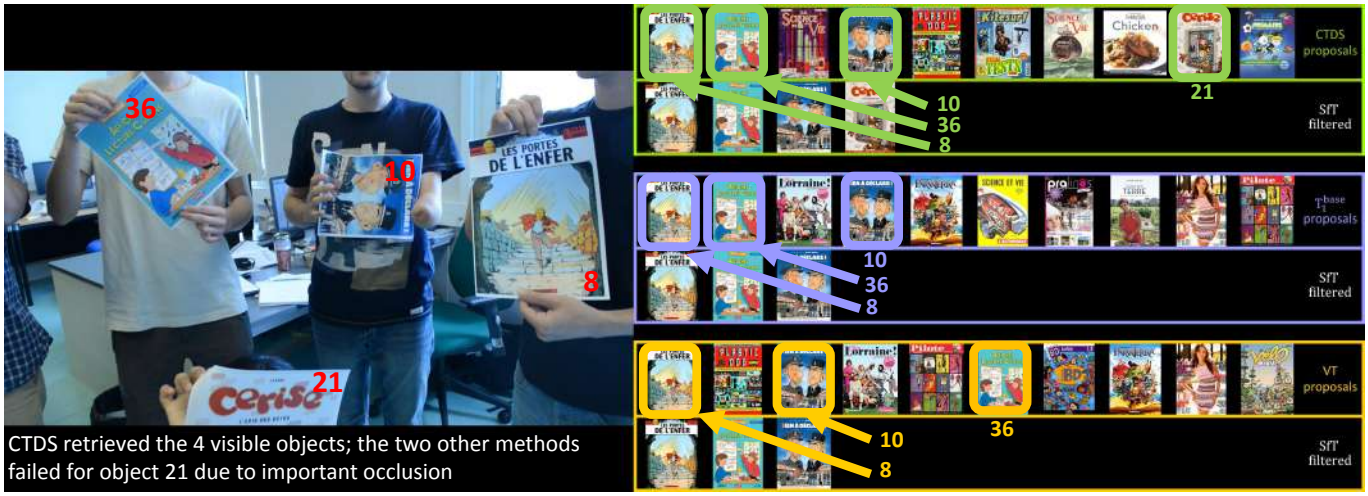
(b)



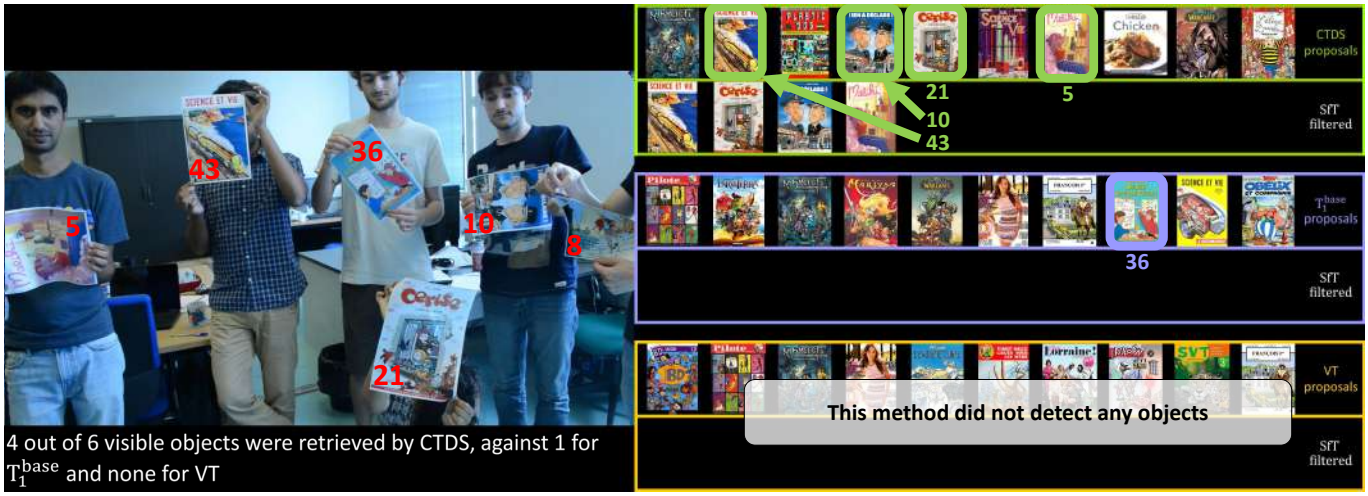
(c) 0:00:36.267

Figure 20: (a) Query image containing objects 5, 6, 36 partially and object 10 entirely.(b) Query image containing object 36 entirely and object 43 partially. (c) Query image containing object 5 entirely and object 43 partially.

(a) 0:01:13.7



(b) 0:01:00.033



(c) 0:01:47.133

Figure 21: (a) Query image containing objects $8, 10, 21, 36$ entirely and object 43 partially. (b) Query image containing objects $8, 10, 36$ entirely and object 21 partially. (c) Query image containing objects $5, 8, 10, 36, 43$.

(a) 0:01:34.933

(b) 0:01:13.433

(c) 0:01:24.600

Figure 22: (a) Query image containing objects $8, 10, 36, 43$. (b) Query image containing objects $8, 10, 21$ entirely and objects $36, 43$ partially. (c) Blurred query image containing objects $8, 10, 21, 36, 43$.

Figure 23: Second fifty objects in the database contained in B_4, B_5 and B_6.

O. Arandjelović. Object matching using boundary descriptors. In *BMVC*, 2012.

R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *ICCV*, 2011.

R. Arandjelović and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012a.

R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012b.

Y. Avrithis and G. Tolias. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International Journal of Computer Vision*, 107(1):1–19, 2014.

A. Bartoli, Y. Grard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-Template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, 2015.

V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

G. Carneiro and A. D. Jepson. Flexible spatial configuration of local image features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2089–2104, 2007.

Y. Chen, X. Li, A. Dick, and A. van den Hengel. Boosting object retrieval with group queries. *IEEE Signal Processing Letters*, 19(11):765–768, 2012.

O. Chum, J. Philbin, J. Sivic, M. M. Isard, and A. Zisserman. Total recall II: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, 2011.

T. Collins and A. Bartoli. Realtime shape-from-template: System and applications. In *ISMAR*, 2015.

T. Collins, P. Mesejo, and A. Bartoli. An analysis of errors in graph-based keypoint matching and proposed solutions. In *ECCV*, 2014.

A. G. Faheema and S. Rakshit. Feature selection using bag-of-visual-words representation. In *IACC*, 2010.

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9): 1627–1645, 2010.

J. Feng, Y. Wang, and S. Chang. 3D shape retrieval using a single depth image from low-cost sensors. In *WACV*, 2016.

P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

J. C. V. Gemert, J. Geusebroek, C. J. Veenman, and A. W. Smeulders. Kernel codebooks for scene categorization. In *ECCV*, 2008.

R. Girshick. Fast R-CNN. In *ICCV*, 2015.

K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8(Apr):725–760, 2007.

R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

H. Jégou and O. Chum. Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In *ECCV*, 2012.

H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010a.

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010b.

S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *CVPR*, 2011.

B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.

M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

M. Leordeanu, A. Zanfir, and C. Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. In *ICCV*, 2011.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

S. Magnenat, D. T. Ngo, F. R. Zund, M. Ryffel, G. Noris, G. Rothlin, A. Marra, M. Nitti, P. Fua, M. Gross, et al. Live texturing of augmented reality characters from colored drawings. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1201–1210, 2015.

A. Makadia. Feature tracking for wide-baseline image retrieval. In *ECCV*, 2010.

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(1-2):43–72, 2005.

A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning vocabularies over a fine quantization. *International Journal of Computer Vision*, 103(1):163–175, 2013.

D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

J. O. M. Östlund, A. Varol, T. D. Ngo, and P. Fua. Laplacian Meshes for Monocular 3D Shape Recovery. In *ECCV*, 2012.

M. Perriollat and A. Bartoli. A computational model of bounded developable surfaces with application to image-based three-dimensional reconstruction. *Computer Animation and Virtual Worlds*, 24(5):459–476, 2013.

F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010.

D. Pickup, X. Sun, P. L. Rosin, and R. R. Martin. Skeleton-based canonical forms for non-rigid 3D shape retrieval. *Journal of Computational Visual Media*, 2(3):231–243, 2016.

J. Pilet and H. Saito. Virtually augmenting hundreds of real pictures: An approach based on learning, retrieval, and tracking. In *VR*, 2010.

J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.

D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

S. Qi and Y. Luo. Object retrieval with image graph traversal-based re-ranking. *Signal Processing: Image Communication*, 41:101–114, 2016.

D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. V. Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 2011.

R. Raguram and J. Frahm. Recon: Scale-adaptive robust estimation via residual consensus. In *ICCV*, 2011.

J. Ricard, D. Coeurjolly, and A. Baskurt. Generalizations of angular radial transform for 2D and 3D shape retrieval. *Pattern Recognition Letters*, 26(14):2174–2186, 2005.

Y. Sahillioğlu and L. Kavan. Detail-preserving mesh unfolding for nonrigid shape retrieval. *ACM Transactions on Graphics*, 35(3):27, 2016.

M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.

C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.

X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, 2012.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CVPR*, 2014.

J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

P. Tirilly, V. Claveau, and P. Gros. A review of weighting schemes for bag of visual words image retrieval. 2009.

C.-F. Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012, 2012.

P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshops*, 2009.

A. Vedaldi and S. Soatto. Relaxed matching kernels for robust image comparison. In *CVPR*, 2008.

R. Veltkamp, H. Burkhardt, and H. Kriegel. State-of-the-art in content-based image and video retrieval. *Springer Science and Business Media*, 22, 2013.

S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *CVPR*, 2009.

J. Yan, , C. Zhang, H. Zha, W. Liu, X. Yang, and S. Chu. Discrete hyper-graph matching. In *CVPR*, 2015.

Z. Z. Yongwei, L. Bicheng, and G. Haolin. Bag-of-visual-words based object retrieval with E2LSH and query expansion. 127:713–725, 2012.

R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *CVPR*, 2008.

Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011.

L. Zheng, S. Wang, Z. Liu, and Q. Tian. Lp-norm idf for large scale image search. In *CVPR*, 2013.

L. Zheng, S. Wang, and Q. Tian. Coupled binary embedding for large-scale image retrieval. *IEEE Transactions on Image Processing*, 23(8):3368–3380, 2014.