Filtering out nondiscriminative keypoints by geometry based keypoint constellations

Domen Rački, Matej Kristan

Faculty of Computer and Information Science, University of Ljubljana
{domen.racki, matej.kristan}@*fri.uni-lj.si*

Abstract

Keypoint-based object detection typically utilizes the nearest neighbour matching technique in order to mach discriminative and reject nondiscriminative keypoints. A detected keypoint is found to be nondiscriminative if it is similar enough to more than one model keypoint. This strategy does not always prove efficient, especially in cases where objects consist of repeating patterns, such as letters in logotypes, where potentially useful keypoints can get rejected. In this paper we propose a geometry-based approach for filtering out nondiscriminative keypoints. Our approach is not affected by repeating patterns and filters out non discriminative keypoints by means of prelearned geometry constraints. We evaluate our proposed method on a challenging dataset depicting logotypes in real-world environments under strong illumination and viewpoint changes.

1 Introduction

Object detection has a wide range of applications in specialized as well as everyday computer vision tasks, like image retrieval systems [2, 11], everyday object recognition [5, 3] and automated pick and place systems [6]. Before a specific object can be detected, an abstract representation of the object, i.e., a visual model is constructed utilizing features, extracted from an image depicting the object of interest. The visual model is used to perform object detection by matching the model features to features extracted from a test image, containing or not containing the object of interest. Feature matching is typically done in a nearest neighbour manner, meaning that every extracted feature is matched to the most similar model feature.

As illustrated in Figure 1, using only the nearest-neighbour matching technique, noisy mismatches, where a potentially non-discriminative detected feature is matched to a random model feature cannot be avoided. In order to address noisy mismatches and reject non-discriminative features, a similarity threshold θ is proposed in [4]. A potential feature match $\boldsymbol{k}_i \rightarrow \boldsymbol{m}_i$ is rejected, if the similarity threshold between the detected feature $\boldsymbol{k}_i$ and two model features $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ is higher than θ, i.e., $\frac{s_1}{s_2} > \theta$, where $s_1 = \boldsymbol{k}_i \sim \boldsymbol{m}_i$ and $s_2 = \boldsymbol{k}_i \sim \boldsymbol{m}_j$. Although this approach rejects non-discriminative features it also presents a potential problem. Consider that our object

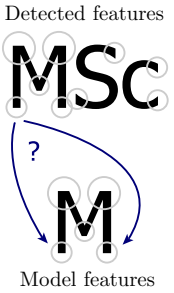

Figure 1: Illustration of a case where a detected feature, represented by a gray circle, cannot be correctly matched to a model feature, also represented by a gray circle, due to its non-discriminative nature.

is the letter "M", as illustrated in Figure 1. The features representing the bottom-left and the bottom-right "leg" of the letter M are identical. If we now present a test image on which we would like to detect our object, the letter M, the following happens. When we try to match detected features to model features, the proposed threshold rejects potentially useful matches, since the feature detected on the bottom-left "leg" of the letter M in the presented image is very similar to two model features, representing the bottom-left and the bottom-right "leg" of the letter M object. If, however, we ignore the similarity threshold and match using only the nearest-neighbour technique, we see that it is not clear which of the two features is the better match for the feature detected on the bottom-left "leg" of the letter M in the proposed image.

Existing approaches on preserving spatial relations between local features can be found in [9, 8], where, for a given object, authors present an approach towards encoding triplets of neighbouring features and locating the encoded triplets across test images. Although the mass pipeline detection process shows promising results on the FlickrLogos-32 dataset [10], the whole approach requires a set of diverse training images, whereby the problem of object deformations is addressed by warping either training images, or database images in order to achieve robustness against object deformations, i.e., deformations of spatial relations between features. We argue that the spatial layout encoding technique could be generalized to incorporate an arbitrary number of neighbouring features and spatial deformations of features, whereby requiring

a single planar training image, i.e., a one-shot learning approach utilizing pre-learned geometry restrictions. We evaluate our proposed approach on a challenging real-world dataset dubbed FlickrLogos-32 [10], consisting of 2240 images, depicting logotypes in real-world environments under strong illumination and viewpoint changes.

The remainder of the paper is structured as follows. In Section 2 we provide a short description of our proposed keypoint model, followed by the description of how geometry restriction are learned in Section 3. Object detection by constellations is described in Section 4. Experiments and results are described and presented in Section 5. We conclude with the discussion in Section 6.

2 The keypoint constellation model

A keypoint-based visual model of an object typically consists of encoded keypoints and their corresponding descriptors. In our model, we additionally encode geometric restrictions based on the locations of neighbouring keypoints and learned keypoint variances, which we call keypoint constellations. A keypoint constellation thus consists of a set of keypoints, which model local object structures.

Given a set of extracted keypoints and corresponding region descriptors, for example obtained with [4], where each keypoint is represented by a location vector $[x, y]^T$ and the corresponding orientation $[s, \theta]^T$, an object is encoded by a visual model consisting of a set of keypoints $\{\boldsymbol{k}_j^{(l)}\}_{j=1:N_k}$, where N_k corresponds to the number of extracted keypoints, and a set of corresponding descriptors $\{\boldsymbol{d}_j^{(l)}\}_{j=1:N_k}$, where $\boldsymbol{d} = [a_1, ..., a_n]^T$. In our setup, the extracted keypoints are represented by position, scale and orientation. In the polar coordinate system a keypoint is represented by a position vector $[x, y]^T$, which encodes the center of the polar coordinate system, i.e., the absolute location of the keypoint in an image, the scale s and the angular coordinate θ, i.e., $\boldsymbol{k}^{\mathcal{P}} = [x, y, s, \theta]^T$. Thus, in the Cartesian coordinate system, a keypoint can be encoded by a pair of points $\boldsymbol{k}^{\mathcal{C}} = [\boldsymbol{c}, \boldsymbol{o}]^T$, whereby, $\boldsymbol{c} = [x, y]^T$, representing the position of the keypoint and $\boldsymbol{o} = [s \cdot \cos(\theta), s \cdot \sin(\theta)]^T$, representing the scale and angle.

2.1 Encoding constellations of keypoints

A "pair" keypoint $\boldsymbol{p}$ is considered to be in the neighbourhood of a given "root" keypoint $\boldsymbol{r}$, if the distance between $\boldsymbol{r}$ and $\boldsymbol{p}$ is less than ϵ_{max}. In order to prevent cases where $\boldsymbol{p}$ is arbitrary close to $\boldsymbol{r}$, an additional constraint is that the distance between $\boldsymbol{r}$ and $\boldsymbol{p}$ must be greater than ϵ_{min}. The region between ϵ_{min} and ϵ_{max} is dubbed *the ϵ region*. It follows that a keypoint is considered to be a pair keypoint $\boldsymbol{p}$ of a given root keypoint $\boldsymbol{r}$, if $\boldsymbol{p}$ lies within the ϵ region around $\boldsymbol{r}$, as illustrated in Figure 2. The ϵ region is determined by considering the scale s of the root keypoint. Any detected keypoint, which lies within the two thresholds $\epsilon_{min} = s\alpha$ and $\epsilon_{max} = s\beta$ is considered to be in the neighborhood of $\boldsymbol{r}$, i.e., a pair keypoint of $\boldsymbol{r}$.

Since $\boldsymbol{r}$ represents the root of the constellation, all of the corresponding pairs are encoded relative to $\boldsymbol{r}$. The encoding is performed by computing a similarity transformation $\boldsymbol{T}_{\mathcal{U}} : \boldsymbol{r} \rightarrow \boldsymbol{u}$, which maps $\boldsymbol{r}$ from the image space $\mathcal{I}$, to the so called unit keypoint $\boldsymbol{u}$, which lies in the unit space, i.e., $\boldsymbol{u} \in \mathcal{U}$. In the unit space, the unit keypoint lies in the cartesian coordinate system origin with a scale of one and a zero angle, i.e., $\boldsymbol{u} = [0, 0, 1, 0]^T$. Each pair keypoint $\boldsymbol{p}$ is mapped to the unit space of the root keypoint by the transformation $\boldsymbol{T}_{\mathcal{U}}$. All pair keypoints are thus encoded relative to the root keypoint, encoding a keypoint constellation.

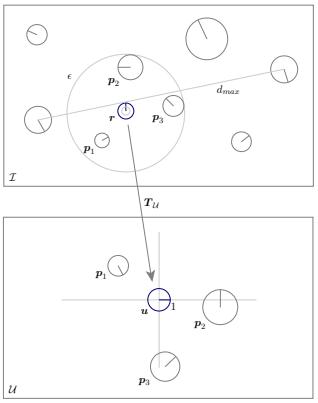

Figure 2: Keypoints in the image space $\mathcal{I}$, which lie within an ϵ region around a root keypoint $\boldsymbol{r}$ are considered to be the pair keypoints. The transformation $\boldsymbol{T}_{\mathcal{U}} : \boldsymbol{r} \rightarrow \boldsymbol{u}$ maps all pair keypoints to the normalized unit space $\mathcal{U}$, encoding a local keypoint neighborhood of the root keypoint.

3 Learning keypoint variance

When an object is deformed, local structures retain their spatial relations depending on the degree of deformation. A keypoint constellation consisting of pair keypoints, encoded in the root keypoint unit space, rigidly models the spatial relations between the encoded keypoints. When an object deforms, keypoint locations deforme as well. It follows that a deformable constellation can be obtained by considering the variance of the encoded keypoints.

In order to model the expected variance of a keypoint within a constellation, we consider two different approaches. The first approach is to measure the expected variance of a keypoint by taking a set of images and subjecting them to a series of projective transformations i.e., warping the images. The expected variance of a keypoint is measured by considering each detected keypoint on each warped image. The second approach is to model the expected variance by considering the maximal allowed deformation of an image, under which a keypoint detection method is capable of detecting a keypoint. The expected keypoint variance can thus be modelled by a function, obtained by considering the maximal allowed image deformation. The former and latter approaches, essentially modelling expected keypoint variance with a Gaussian, are briefly explained in the following subsections.

3.1 Empirical keypoint variance

Given a set of planar images, depicting different objects, keypoint variance can be measured by warping these im-

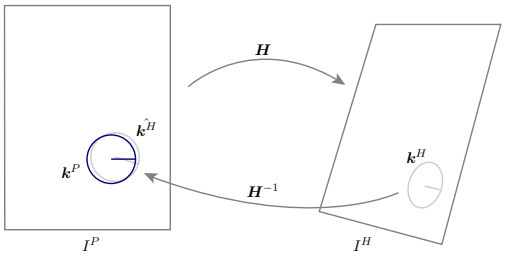

Figure 3: Give a transformation $\boldsymbol{H}$, the planar image I^P can be warped, i.e., subjected to projective transformation. Since $\boldsymbol{H}$ is known, a keypoint $\boldsymbol{k}^H$, depicted in gray, detected on a warped image can be backprojected by $\boldsymbol{H}^{-1}$, and it's variation determined relative to it's reference keypoint $\boldsymbol{k}^P$, depicted in blue.

ages and considering the variance of each keypoint detected on a warped image, relative to a reference keypoint, detected on a planar image, as illustrated in Figure 3. Given a reference keypoint $\boldsymbol{k}^P$, detected on a planar image, it's variance is measured by considering it's warped instance $\boldsymbol{k}^H$. If this is repeated for a set of keypoints, and a set of transformations, the measured data yields the expected variance.

3.2 Numerical keypoint variance

Assume that a feature detection method is capable of detecting keypoints under affine transformations of γ degrees of the image. Keypoint localization will slightly vary, depending on γ, and will result in a variance of the keypoint location. The expected location variance can be modelled as follows. Imagine a plane Π with a unit keypoint $\boldsymbol{u}$, and a point $\boldsymbol{q}$ on the x-axis. By transforming Π with a transformation $\boldsymbol{T}_\gamma : \Pi \rightarrow \Pi'$, we simulate the maximal affine transformation by γ degrees, under which a keypoint can be detected, as shown in Figure 4.

The transformation $\boldsymbol{T}_u : \boldsymbol{u}' \rightarrow \boldsymbol{u}$ maps the transformed unit keypoint $\boldsymbol{u}'$ back to its initial position. By using $\boldsymbol{T}_u$ to back-project the transformed point $\boldsymbol{q}'$, the back-projection error between $\boldsymbol{q}$ and $\boldsymbol{q}''$ can be computed, i.e., $e = ||\boldsymbol{q} - \boldsymbol{q}''||_2$. Since the error depends on the distance of the point $\boldsymbol{q}$ from the coordinate system origin, the back-projection error has to be computed for n different points $\boldsymbol{q}$, at different distances. The measured back-projection error data yields a second degree polynomial function, from which we can extract the analytical relation between γ and the variance of a point by solving the polynomial fit equation $\boldsymbol{e} = \boldsymbol{D}\boldsymbol{c}$, where the vector $\boldsymbol{e} \in \mathbb{R}^n$ corresponds to the backprojected errors of each point, the matrix $\boldsymbol{D}_{n\times 3}$ to the distances of each point from the coordinate system origin and the vector $\boldsymbol{c} \in \mathbb{R}^n$ to the second degree polynomial coefficients.

4 Object detection by constellations

Object detection by constellation models proceeds by first extracting keypoints from an image. The extracted features are matched to the model features by nearest-neighbour matching, i.e.,

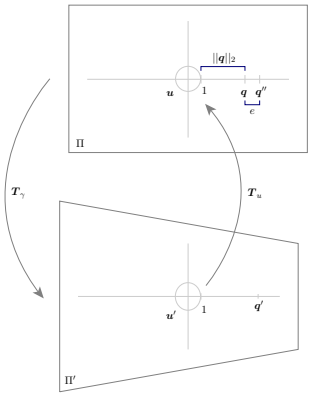

Figure 4: The plane Π is subjected to the maximal allowed affine transformation of γ degrees, under which a keypoint detection method is capable of detecting a keypoint. By reprojecting the transformed point $\boldsymbol{q}'$ back to the original plane Π, the reprojection error, relative to $\boldsymbol{q}$, can be measured.

$$\boldsymbol{k}_i \rightarrow \boldsymbol{m}_i \quad \text{if} \quad \boldsymbol{d}_i^m = \arg\min_{\boldsymbol{d}_i^m}(||\boldsymbol{d}_i^k - \boldsymbol{d}_i^m||_2).$$

For each matched feature, a constellation search is performed utilizing the encoded geometry restrictions to distinguish features located on the object from features located elsewhere. For each encoded variance in the constellation around a given matched keypoint, a check is performed whether any of the neighbouring keypoints fall into the encoded geometry restrictions. If no neighbouring keypoints fall into the constellation, the given matched keypoint is found to be non-discriminative. This step acts as a filtering process for non-discriminative features. A generalized Hough transform [1] is applied to determine a cluster of features voting for the same image position and the hypothesis is refined by robustly fitting an object model to the cluster [7].

5 Experiments and results

The proposed constellation model is designed to merely enhance basic feature-based object detection methods, as constellations are computed and utilized "on top" of extracted features. Considering that the proposed similarity threshold in [4] does not always prove efficient, the experimental evaluation consists of comparing the detection performance of six different models, summarized in Table 1. Essentially we compare the performance of the constellations vs. the performance of the similarity threshold as a filtering technique, on the challenging real-world dataset FlickrLogos-32 [10].

Figure 5 depicts the the average number of generated hypotheses per tested model, i.e., μ_{total} and the number of hypotheses exceeding the threshold τ_{GHT} in the Generalised Hough transform accumulator array, i.e., $\mu_{\tau_{GHT}}$. The proposed constellation models $\mathcal{C}^E$, $\mathcal{C}^E_\tau$, $\mathcal{C}^N$ and $\mathcal{C}^N_\tau$ significantly reduce μ_{total} and $\mu_{\tau_{GHT}}$ when compared to the basic models $\mathcal{B}$ and $\mathcal{B}_\tau$. Although the proposed constellation models significantly reduce the number of hypotheses μ_{total} and $\mu_{\tau_{GHT}}$, the occurring question is how

Table 1: Tested models.

Model notation	Similarity threshold	Empiric variance	Numeric variance
$\mathcal{B}$	×	×	×
$\mathcal{B}_\tau$	✓	×	×
$\mathcal{C}^E$	×	✓	×
$\mathcal{C}^E_\tau$	✓	✓	×
$\mathcal{C}^N$	×	×	✓
$\mathcal{C}^N_\tau$	✓	×	✓

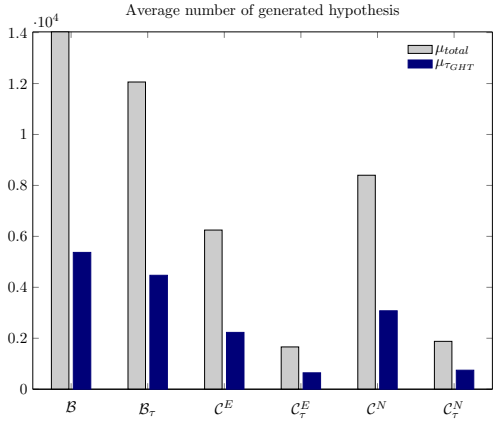

Figure 5: Hypothesis reduction.

detection performance is affected. Figure 6 depicts the Precision-Recall curves for the tested models. In general, there is no noticeable difference in the performance of the models, implying that the proposed constellation models do not corrupt detection by removing true positives.

We find that the best variation of the constellation model is the model with empirically determined feature variance, which significantly reduces the number of mismatched features, without significantly affecting detection performance.

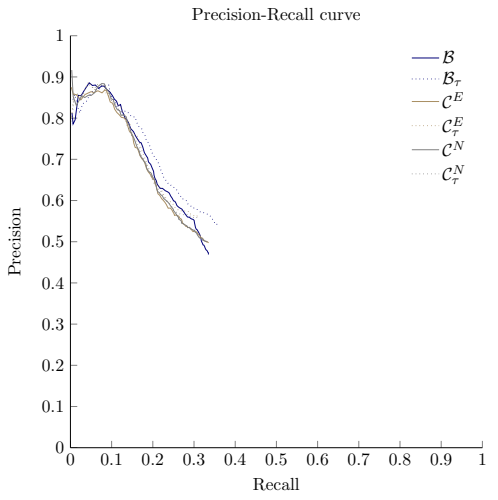

Figure 6: Model detection performance.

6 Conclusion

We propose a learned constellation model for the construction of object models as an enhancement of basic feature-based object detection methods. In contrast to the use of the similarity threshold, the proposed constellation model aims in filtering out mismatched features and producing clearer object location hypotheses by the use of geometry. Two variations of the proposed constellation model, with empirically and numerically modelled variance, and the basic feature model, all with and without the similarity threshold were evaluated on the challenging real-world dataset FlickrLogos-32 [10], depicting logotypes in real-world environments. Overall, the proposed constellation models reduce the number of mismatched features, without significantly affecting detection performance. By reducing the number of mismatched features, the constellation model reduces the number of potential object location hypotheses which need to be verified, achieving a substantial noise reduction compared to the similarity threshold, even in highly cluttered environments. The constellations based solely on geometry are computed and utilized ”on top” of detected features, so any feature extraction method whose keypoints are represented by a center location, scale and orientation can be enhanced with the proposed constellation model.

References

[1] D. H. Ballard. Readings in computer vision: Issues, problems, principles, and paradigms. chapter Generalizing the Hough Transform to Detect Arbitrary Shapes, pages 714–725. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8, Oct 2007.

[3] E. Hsiao, A Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *CVPR*, pages 2653–2660, June 2010.

[4] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV - Volume 2*, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[5] M. Merler, C. Galleguillos, and S. Belongie. Recognizing groceries in situ using in vitro training data. In *CVPR*, pages 1–8, June 2007.

[6] Paolo Piccinini, Andrea Prati, and Rita Cucchiara. Real-time object detection and localization with sift-based clustering. *IVC*, 30(8):573–587, August 2012.

[7] Domen Rački, Luka Čehovin, and Matej Kristan. Detection of 3d objects with a multiple-view keypoint model from item images. In *ERK*, volume 8, pages 100–103, 2014.

[8] Stefan Romberg and Rainer Lienhart. Bundle min-hashing. In *IJMIR*, volume 2, pages 243–259, 2013.

[9] Stefan Romberg and Rainer Lienhart. Bundle min-hashing for logo recognition. In *ICMR*, pages 113–120, 2013.

[10] Stefan Romberg, Lluis Garcia Pueyo, Rainer Lienhart, and Roelof van Zwol. Scalable logo recognition in real-world images. In *ICMR*, pages 25:1–25:8, 2011.

[11] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, CVPR ’11, pages 809–816, Washington, DC, USA, 2011. IEEE Computer Society.