Relevance Assessment for Visual Video Re-ranking

Javier Aldana-Iuit(✉), Ondřej Chum, and Jiři Matas

Department of Cybernetics, Faculty of Electrical Engineering, Center for Machine Perception, Czech Technical University in Prague, Karlovo nam. 13, 121 35 Prague 2, Czech Republic
{aldanjav,chum,matas}@cmp.felk.cvut.cz

Abstract. The following problem is considered: Given a name or phrase specifying an object, collect images and videos from the internet possibly depicting the object using a textual query on their name or annotation. A visual model from the images is built and used to rank the videos by relevance to the object of interest. Shot relevance is defined as the duration of the visibility of the object of interest. The model is based on local image features. The relevant shot detection builds on wide baseline stereo matching. The method is tested on 10 text phrases corresponding to 10 landmarks. The pool of 100 videos collected querying YouTube with includes seven relevant videos for each landmark. The implementation runs faster than real-time at 208 frames per second. Averaged over the set of landmarks, at recall 0.95 the method has mean precision of 0.65, and the mean Average Precision (mAP) of 0.92.

Keywords: Video re-ranking · Object detection · Wide-baseline stereo matching

1 Introduction

In this paper we address an application of acquiring videos containing a user specified object. The user provides a text identification of the object of interest, possibly also an image – for example from a Wikipedia page. The text description is used to query some external image and video sharing sites. From the relevant images of additional views of the object, a visual model is built. The model is then used to efficiently identify shots in videos depicting the object of interest and consequently to re-rank the videos.

An example of the use of our solution is the following: During a holiday trip in Paris we took a set of images from the Notre Dame cathedral from different points of view, then we may want to search on YouTube videos related to the same landmark, in order to learn more about it or getting tour guide videos with information about surrounding venues. It would be annoying to check manually the retrieved videos for finding the shots where the landmark appears in case it does so.

In the proposed method, we are not interested in indexing a fixed corpus of videos, but we relay on text based search capabilities provided by, for example, You Tube. Through the text search, possibly relevant, but likely noisy, a short-list of videos is obtained. An efficient visual content based matching is applied to verify and re-rank the initial short-list. The paper focuses on the object model building from a set of images and on efficient online detection of the object in videos. The method is summarized in Fig. 1.

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part I, LNCS 8814, pp. 421–430, 2014.
DOI: 10.1007/978-3-319-11758-4_46

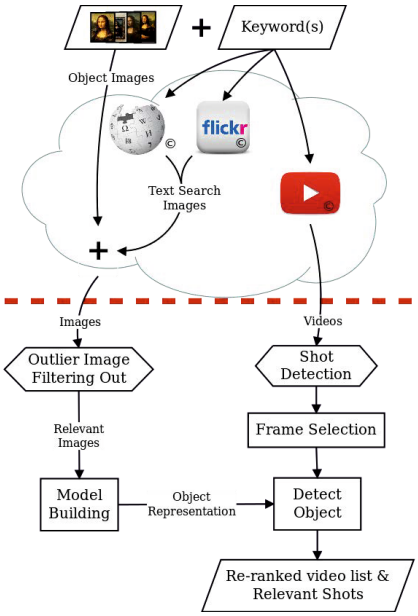


Fig. 1. The workflow of relevant shot detection. The part below the dashed red line is automatic and the focus of the paper, the text-based search has been done manually.

The applicability of the proposed system ranges from individual user searches for relevant videos to systematic augmentation of Wikipedia (or similar) pages with relevant video documents.

Relevant work. Visual content based searching of videos and large image collection has become very popular with Video Google [16] by Sivic and Zisserman. In this work, as well as in other image retrieval publications [6,11,14], it is assumed that the video or image collection is going to be sought repeatedly for different query objects. Therefore an offline stage of indexing of the videos or image collection takes place. On the contrary, we assume that each video is unlikely to be needed multiple times. In fact, most of the videos will never be accessed, and therefore we leave the initial retrieval of the short list on text search facilities of the video sharing site, YouTube in our experiments.

The concept of matching multiple views of a single object to obtain a visual model with stable local features has been used in a number of applications. In query expansion [4], a generative model of the object is built from a small number of geometrically consistent retrieved result images. In [17], the database features are reduced by matching the images within the dataset, resulting in more compact representation without hurting the search performance.

Text-based search works as mechanism for collecting input data, likewise [1]. In video analysis, features that are repeatedly detected over a number of consecutive frames are reliable [15] and are kept for further computation.

2 The Method

The paper focuses on two aspects of the problem: the object model building and efficient online detection of the object in videos.

The object model is built from a set of pre-filtered, but still possibly contaminated, images of an object of interest. We take such a collection of images as an input and call it *the pool of images*. To pre-filter the images obtained from image sharing site by a text query, user provided images or Wikipedia images are used. All images are used jointly to build a representation of the object based on local features. Detailed description follows in Sec. 2.1.

Another input to our method is a short-list of videos, retrieved by a text query to You Tube. Videos from the short-list are represented as sequences of shots, each shot is represented by its key frames. A relevance of a key-frame to the object is given by the number of geometrically consistent image features found after a wide-baseline stereo matching to the object model. The videos are finally ranked w.r.t. the number of relevant frames. Detailed description of the object detection is given in Sec. 2.3.

2.1 Object Model

In this section, the process of the object model construction is described. The model is a collection of local affine covariant image features localized in an image coordinate frame. Rather than using a single image to obtain the model, we use a small set of images (sets of 7 images were used in our experiments). Using multiple images provides richer description, as some parts of the object may not be well represented in a single image due to noise, (self-)occlusion, etc.

Local features. Local affine covariant features are extracted in images from the pool of images, using *Hessian Affine* detector [10]. The image features are described with the SIFT [8] descriptor.

Model coordinate system. We identify the model coordinate system with one of the images. The Iconoid shift [18] is applied to select the reference image, which is used to define the coordinate system of the model. The Iconoid shift is seeded from each image in turn and the image selected as a mode the most often is selected as the reference image. Unrelated images are filtered out from the pool of images preserving the top K images from the mode support scored by the Homography Overlap Distance (HOD) defined in [18] only. We used $K = 7$ in our experiments. Fig. 2 (a) shows three pools of images, the green rectangles indicates the reference images.

Features from other images are back-projected to the model based on image-to-image homography robustly estimated [3] between the images and the reference image.

Feature selection. In the local feature matching state, the descriptors of the features are compared. Pairs of features (one feature from each image) with similar descriptors are considered as tentative correspondences. It has been suggested in [8] that a distance between descriptor of the same surface patch in different images also depends on the

(a) (b)

Fig. 2. (a) Three examples of set of images used for building the object models, each set is known as the *Pool of images*. (b) Local features found in multiple images in the image pool, called *salient* features. The color scale indicates the number of images where the feature were recognized, and the ellipses indicate the shape of the feature.

appearance of the patch itself, therefore it is better to use the ratio of the distance to the nearest and the second nearest descriptor in the other image.

Since our model is a collection of back-projected features from multiple images, one physical patch can be represented by a number of descriptors. If the distance ratio approach [8] is adopted, the distance ratio can be close to 1 even for a good tentative correspondence, because the first and second nearest descriptor may belong to two different instances of the same physical scene patch. We compare two approaches avoiding this phenomenon.

The first approach is based on a recent idea from [12], called *1st Geometrical Inconsistent* strategy. Some detectors, especially those using synthetic image warping to improve feature detection, have multiple detections of very similar features. To avoid dropping correct tentative correspondences, authors of [12] suggest to compute the distance ratio to the nearest descriptor that comes from feature that is sufficiently far away (i.e. geometrically inconsistent) from the tentatively corresponding one.

The second approach tries to reduce the number of features in the model by joint clustering in the SIFT and image domain. For each feature back-projected into the model, 130D SIFT-XY descriptor is created by concatenating the SIFT descriptor with the feature coordinates (multiplied by a normalizing constant). The features are clustered by applying DBSCAN [5] algorithm to the SIFT-XY descriptors. In order to drop randomly detected features that are not repeatable, features from singleton clusters are dropped. An average feature (in SIFT and XY) is kept in the model for each of the feature clusters. The average model features for different landmarks are shown in Fig. 2 (b). A similar approach for computing mid-level features is proposed in [7]. The full algorithm to compute the set of salient features is summarized in Alg. 1.

2.2 Video Representation

The set of videos collected from the text retrieval are represented by a subset of keyframes (Intra-coded frames or I-frames) concerning the CODEC. Local affine covariant features are detected and described on every selected key-frame. This stage avoids the wide-baseline stereo matching over all frames of the video, rather than that, we match the object model against up to 1% of the total number of frames. For shot boundary detection, we apply a simple detector [2], that thresholds the sum of pixel-wise

Algorithm 1. Salient features

Require: Pool of images (P), reference image (I_{ref})
Ensure: Set of salient features (SF)
  $N \leftarrow |P|$
  // Detect and describe image features
  for $i = 1$ to N **do**
    $f_i \leftarrow$ `hessian_affine_detection`(p_i)
    $d_i \leftarrow$ `SIFT_description`(f_i)
  end for
  $D = \{d_1, ..., d_N\}$
  // Features in images of the pool without the reference
  $C \leftarrow D \setminus \{d_{ref}\}$
  $c_i \in C, i = 1, ..., N-1$
  // Set of reprojected features (RF)
  $RF \leftarrow \{f_{ref}\}$
  for $j = 1$ to $N-1$ **do**
    $H_j \leftarrow$ `wbs_match`(d_{ref}, c_j)
    $RF \leftarrow \{RF \cup$ `reproject_features`$(H_j, c_j)\}$
  end for
  $CL \leftarrow$ `DBSCAN_clustering`(RF)
  // Salient features are described by average SIFT
  $SF \leftarrow$ `average_SIFT`(CL, RF)
  return SF

absolute differences. To reduce the number of selected key-frames, we drop key-frames close to the shot boundary, as these are typically corrupted by the shot transition.

2.3 Object Detection in Video Frames

A shot is regarded as relevant if the object or landmark appears on at least one of its selected frames. The object recognition is addressed as a *Wide-Baseline Stereo Matching* problem, as proposed in e.g. [9]. To efficiently detect the nearest neighbor SIFT descriptors, approximate nearest neighbor search is used [13]. Global geometric model and supporting tentative correspondences are robustly estimated using LO-RANSAC [3]. The geometric model of homography or affine transformation are compared.

The relevance of the video the object model is given by the number of relevant frames that appear in the video.

3 The Dataset

The relevant shot detection algorithm was applied to a dataset of images and videos collected from 10 different queries: *Petra city* in Jordan, *Notre Dame cathedral* in France, *Taj Mahal palace* in India, *The Mona Lisa* painting in France, *The Merida's Monumento a la Patria* in Mexico, *Christ The Redeemer* in Brazil, the Coca-Cola logo, the Lola perfume container, Starbucks logo and Virgin Mary painting.

The image pools contain 7 images (top 7 images in the mode support ranked by the HOD) per query object with a fixed width of 640 pixels and keeping the aspect ratio. All images were stored in JPEG format. The number of local affine covariant features detected on the images are presented on Tab. 1.

The video set contains 100 videos downloaded from You Tube. Every object of interest has 10 videos, 7 of them actually depict the object and 3 of them works as confusers (videos were retrieved by querying You-Tube with the same text search but the object never appears on scene).

The videos have an average duration of 3 minutes, the frame rate is fixed 25 fps, the size of the frames is 640x480 pixels. All videos are stored with the codec H.264, which inserts a keyframe (Intra-coded picture) every 60 frames. Notice that only keyframes are processed.

4 Experiments

4.1 Object Model Construction

The effectiveness of the object representation is tested in the experiments comparing the results using 2 types of representation. The first one is called *Union* which is the set of reprojected features on the reference image with no filtering stage. The second one is the set of salient features (described in Sec. 2.1) and it is called *Salient*.

Tab. 1 contains the number of features in the two object representations and the reference image itself. The average size of the *salient* representation is 3% of the whole features detected on the pool of images (union) and 18% of the features detected on the reference image. The significant reduction in the cardinality of the feature sets is reflected in memory allocation and the complexity of matching task. The average time for building a *salient* model is 4.1 sec. for a single image pool. Construction time of Union models (1.27 sec) is obtained subtracting the mean-shift clustering step. The percentages of processing time for each step of the model computation are shown in Fig. 5.

4.2 Comparison of Different Approaches

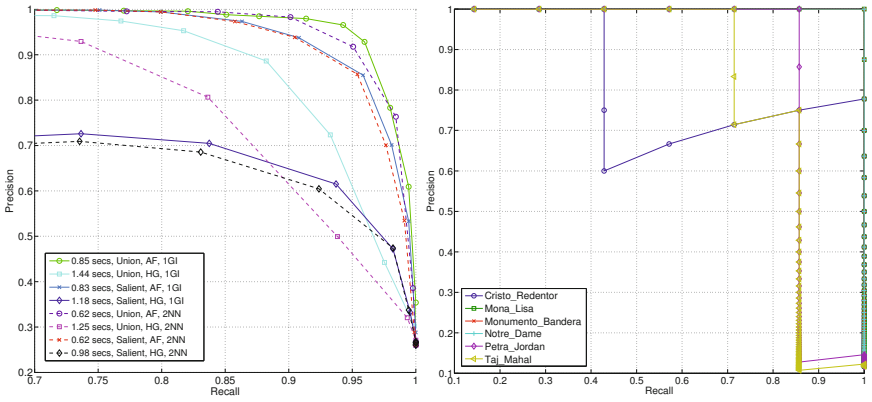
In this section we compare different combination of choices of model construction (Union vs. Salient), tentative correspondence establishment (2NN vs. 1GI) and global geometry model type used in RANSAC (homography vs. affine transformation).

For this task, one video per landmark (6 videos) were annotated manually fixing the subsampling factor $s = 50$ (0.5fps). The Fig. 4.2 (a) shows the recall/precision curves obtained from 6 different method combinations. A number of observation can be made from the plot: (1) better matching results are obtained with more restrictive affine transformation than with full planar homography model; (2) the union representation of the object is slightly more accurate than the salient representation; (3) the 1GI brings almost no advantage for the salient model since similar features have been locally unified in the clustering step.

Two single-frame examples of the matching results with the object model are shown in Fig. 4. Figure 4 (a) corresponds to a frame with the object of interest, the system

Table 1. The *number of features* in the object representations is shown: *Ref. image* column for features on the reference images, *Union* column for features detected in the whole pool of images and *Salient* column for selected features only. In addition, Kendall tau rank correlation coefficients between the ground truth video ranked list and both retrieved ranked lists, regarding the *Text search* list and the *Re-ranked* list by relevance assessment, are shown as *Ranking Quality*. Best ranked lists are highlighted with bold font.

Query object	Number of features			Ranking Quality	
	Ref. image	Union	Salient	Text Search	Re-ranked
Taj Mahal	1368	11363	585	**0.78**	0.47
Petra city	3484	28109	1002	0.60	**0.78**
Notre Dame	5981	30611	2962	0.56	**0.60**
Monumento Patria	2764	14758	739	0.47	**0.60**
Mona Lisa	2303	17243	2449	0.47	**0.73**
Christ Reedemer	3771	10965	477	0.51	**0.69**
Coca Cola	834	8466	315	0.51	**0.51**
Starbucks	1408	12345	1017	0.33	**0.56**
Virgin Mary	6594	66675	5589	0.69	**0.73**

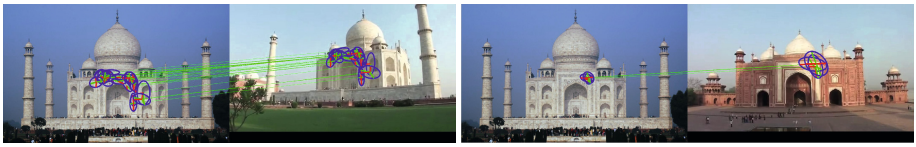


(a) (b)

Fig. 3. (a) Recall/Precision curves on training data for the "Union of feature sets" and "the set of salient features". The mean processing time per frame is shown in the legend. (b) Recall/Precision curves for the "Salient, AF, 1GI" method applied to the 10 landmarks.

found 39 correct feature matches with an inlier ratio of 40%. The Fig. 4 (b) shows the result of matching a frame without the object, the system found 6 inliers with a ratio of 11%, even though all matches are actually incorrect. The number of inliers of matched features is significantly higher when the object is present in the frame.

The best performance concerns to the *Union* model, AF geometric model for RANSAC and 1GI as matching strategy. The later configuration has the second shortest mean processing time per frame. The fastest results comes from the *Salient* model, AF and 2NN, the counter part is a 8% lower precision. The mean processing time per frame in this configuration is 83.1 sec, see Fig. 5 for percentages of time per processing



(a) (b)

Fig. 4. Matching of the object model against a frame containing the object (a) and a frame where the object does not appear (b)

stage. The processing time for 1GI and 2NN are not significantly different because of the previous *SIFT-XY* filtering that suppresses multiple instances of the same feature which hurts the 2NN matching strategy.

For verification, an additional set of experiments were performed with the parameter setting: AF and 1GI and the Union features representation. Based on the recall/precision curve for the training stage, we fixed the detection threshold to 6 inliers which corresponds to a recall of 0.96 and precision of 0.93. The relevant frames are detected with an average recall of 0.88 and average precision of 0.94 over 20 prelabeled videos. The mean processing time per frame is 0.49 secs. The precision and recall fall 0.05 and 0.02, respectively, from training to testing stage. In application such as determining whether the object is present in the video sequence is enough to tune the system for a high ($> 95\%$) recall even that the precision is lower than $30\% - 20\%$, since with only one frame detected correctly, the whole video would be classified as positive.

Relevance-Shot and Re-ranking

In the shot-level detection, *salient* models and video representation are used for ranking the list of retrieved videos. For the experiments with short-lists of 10 videos and 30% of confusers for each landmark, we obtained for recall 1, a precision of 1. Then, we propose to measure the improvement of the re-ranking method over the text search ranked lists by means of the Kendall tau rank correlation coefficients wrt the ground truth of video ranked list by relevant content. The re-ranking method improved the quality of the retrieved ranked lists in 90% of the landmarks (see Table 1).

A set of more challenging experiments were done over all landmarks with short-list of 60 videos and 88% of confusers. The performance of the algorithm on precision and recall is shown in the graphs of Fig. 4.2 (b). Querying the *salient* model at recall 0.95, the average precision is 0.67 and the mean Average-Precision (mAP) is 0.92. For the *Union* model, at recall 0.95, the average precision is 0.64 and the mAP is 0.9. In our experiments, the landmark *Christ, The redeemer* (Cristo Redentor in Portuguese) gets the lowest performance because the image features depends in the light conditions since the statue has the same color everywhere so the shape of the features is strongly dependent of shadows and shadings. Besides, most of the related videos capture the object under extreme point of view (from helicopter) hard to recognize even for the human eye.

The frame selection (video subsampling) is performed in 454.5 fps, and the percentage of time spent in each step of this task is shown in Fig. 5.

Once the object representation is built and the videos are subsampled, the relevance shot detection is done on 208 fps (faster than real-time). The matching task for building the model is the most expensive stage in time and computation resources but it is independent on the length of the short-list and frame selection, moreover the geometric relationship between views (pool of images) are computed during the Iconoid shifting for finding the reference image.

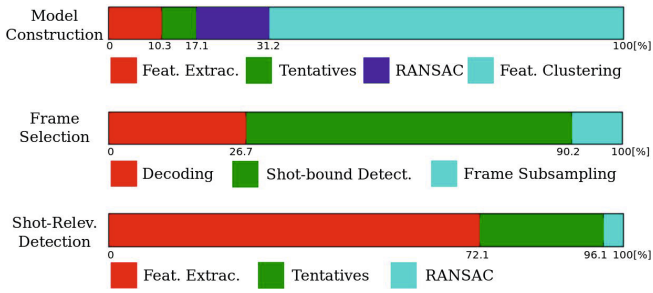


Fig. 5. The fraction of the time spent in the main steps of the relevance-shot detection (in %): building the object representation (top), the frame selection (middle) and the detection task (bottom)

5 Conclusions

In the paper, we have considered the following problem. Given a set of images that includes images of an object of interest and possibly outliers and a pool of videos, re-rank the videos by relevance to the object of interest. Further, the videos are augmented with a list of shots depicting the object of interest. The proposed approach first builds a visual model of the object of interest based on local image features. The relevant shot detection builds on wide baseline stereo matching. Shot relevance is defined as the recording time spent capturing the object of interest reflected in the number of frames depicting it. A number of algorithmic options have been experimentally evaluated. The experiments were carried out on a set of 100 videos collected querying You-Tube with 10 different text phrases.

The best performing method builds the model as a union of features from all example images and constructed the tentative correspondences using the 1^{st} geometrically inconsistent rule. Averaged over the 10 landmarks, mAP is 0.92 querying the object model based on salient features that turns out to outperforms the union model by 2% on mAP. The implementation runs faster than real-time at 208 fps.

Acknowledgments. Javier Aldana-Iuit was supported by CONCIYTEY-CONACYT-Mexico PhD scholarship 216786, DGRI-SEP and Project SGS13/142/OHK3/2T/13. Ondřej Chum was supported by the project GACR P103/12/2310. Jiři Matas was supported by the Czech Science Foundation project GACR P103/12/G084.

References

1. Arandjelović, R., Zisserman, A.: Multiple queries for large scale specific object retrieval. In: British Machine Vision Conference (2012)
2. Boreczky, J.S., Rowe, L.A.: Comparison of video shot boundary detection techniques. In: Storage and Retrieval for Still Image and Video Databases IV, pp. 170–179 (1996)
3. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: Michaelis, B., Krell, G. (eds.) DAGM 2003. LNCS, vol. 2781, pp. 236–243. Springer, Heidelberg (2003). http://dx.doi.org/10.1007/978-3-540-45243-0_31
4. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
5. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, pp. 226–231. AAAI Press (1996)
6. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
7. Koniusz, P., Yan, F., Mikolajczyk, K.: Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. Computer Vision and Image Understanding **117**(5), 479–492 (2013). http://www.sciencedirect.com/science/article/pii/S1077314212001725
8. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
9. Matas, J., Obdrzlek, S., Chum, O.: Local affine frames for wide-baseline stereo. In: ICPR (4), pp. 363–366 (2002), http://dblp.uni-trier.de/db/conf/icpr/icpr2002-4.html#MatasOC02
10. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. Int. J. Comput. Vision **65**(1–2), 43–72 (2005). http://dx.doi.org/10.1007/s11263-005-3848-x
11. Mikulík, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 1–14. Springer, Heidelberg (2010)
12. Mishkin, D., Perdoch, M., Matas, J.: Two-view matching with view synthesis revisited. In: IVCNZ, pp. 436–441 (2013)
13. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Application (VISSAPP 2009), pp. 331–340. INSTICC Press (2009)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
15. Sivic, J., Schaffalitzky, F., Zisserman, A.: Object level grouping for video shots. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3022, pp. 85–98. Springer, Heidelberg (2004)
16. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
17. Turcot, P., Lowe, D.G.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: ICCV Workshop LAVD (2009)
18. Weyand, T., Leibe, B.: Discovering favorite views of popular places with iconoid shift. In: Metaxas, D.N., Quan, L., Sanfeliu, A., Gool, L.J.V. (eds.) ICCV, pp. 1132–1139. IEEE (2011), http://dblp.uni-trier.de/db/conf/iccv/iccv2011.html#WeyandL11