Kato *et al. IPSJ Transactions on Computer Vision and Applications* (2017) 9:12
DOI s41074-017-0021-8

IPSJ Transactions on Computer Vision and Applications

EXPRESS PAPER

Open Access

Selecting image pairs for SfM by introducing Jaccard Similarity

Takaharu Kato[1*†], Ikuko Shimizu[1†] and Tomas Pajdla[2]

Abstract

We present a new approach for selecting image pairs that are more likely to match in Structure from Motion (SfM). We propose to use Jaccard Similarity (JacS) which shows how many different visual words is shared by an image pair. In our method, the similarity between images is evaluated using JacS of bag-of-visual-words in addition to tf-idf (term frequency-inverse document frequency), which is popular for this purpose. To evaluate the efficiency of our method, we carry out experiments on our original datasets as well as on "Pantheon" dataset, which is derived from Flickr. The result of our method using both JacS and tf-idf is better than the results of a standard method using tf-idf only.

Keywords: Structure from motion, tf-idf weighting, Jaccard Similarity

1 Introduction

Image matching, i.e., finding coincident points in several images, is one of the most important topics in computer vision. It is used in the field like object recognition, image stitching, and 3D reconstruction. In all of these fields, detecting features in each image and matching those features to find coincident image points are needed.

The Structure from Motion (SfM), which is one of the 3D reconstruction techniques, is an important application of image matching. SfM reconstructs 3D structure and camera positions from 2D image sequences (Fig. 1).

Recently, many applications in computer vision have been utilizing large datasets of photos on the Internet [1]. With the development of social networking service (SNS), such as Flickr and Facebook, every day, every minute, thousands of photos are uploaded to online databases. And those photos would cover large parts of the Earth. Several techniques utilizing photos on the Web for SfM have appeared [2–7].

Photo Tourism [8] was the first system which worked on the large photo collections on the Web for SfM. In the Photo Tourism system, large image collections from either personal photo collections or the photo collections on the Web are used as an input. First, camera positions are computed and a sparse 3D model of the scene is reconstructed. Then, features detected by using SIFT keypoint detector [9] are then matched exhaustively between all image pairs to accomplish reconstruction. However, a typical image has several thousand keypoints, so exhaustive matching of all image pairs in image collections requires too much computational time and resources to be practical.

To make matching feasible, it is required to find out which images may see the same scene without doing full expensive matching. Often, expensive image matching is replaced by much more efficient search based on meta-information, e.g., keywords or GPS location, or by visual search [10].

Most of the images in the image collections on the Web do not have the information of the locations of the cameras, although some of the images are available with GPS orientation information stored in Exif tags. Even when the GPS positions of images are available, it is not sure that nearby images see the same scene. The same holds true for matching images by keywords attached. Hence, an efficient image-based similarity hinting on seeing common scene is always useful.

Figure 2 shows an example of Flicker images tagged by "Notre Dame" keyword. No two images among the five images have a significant overlap to be worth matching. For example, the bottom right image shows the interior of Notre Dame while others show it from outside.

*Correspondence: kato@m2.tuat.ac.jp
†Equal contributors
[1]Tokyo University of Agriculture and Technology, Tokyo, Japan
Full list of author information is available at the end of the article

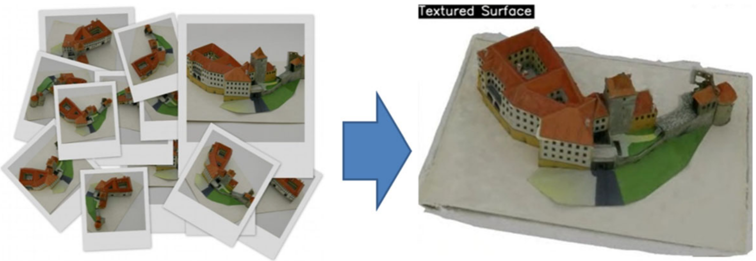


Fig. 1 Structure from Motion [19]

Speeding up the matching can be cast as image search where efficient form of image similarity is constructed. A classical example is the tf-idf (term frequency - inverse document frequency) document similarity used in document search. MatchMiner [11] selects image pairs for matching using bag-of-visual-words and tf-idf weighting to assess image pair similarity. Some other researches use bag-of-visual-words and tf-idf weighting as well [12–14].

We propose using Jaccard Similarity (JacS), which is also known as Jaccard Similarity Coefficient, for calculating image pair similarity in addition to using tf-idf. JacS is originally used for information retrieval [15], and when it is employed for estimating image pair similarity, it shows how many different visual words do image pairs have in common. The min-Hash, which is a locality-sensitive hashing of JacS, is used for image retrieval [16]. In our experiment with an image collection with ground truth, we show that the accuracy of JacS alone is sometimes better than the accuracy when using tf-idf alone and that the accuracy of JacS used together with tf-idf is always much better than using JacS or tf-idf alone.

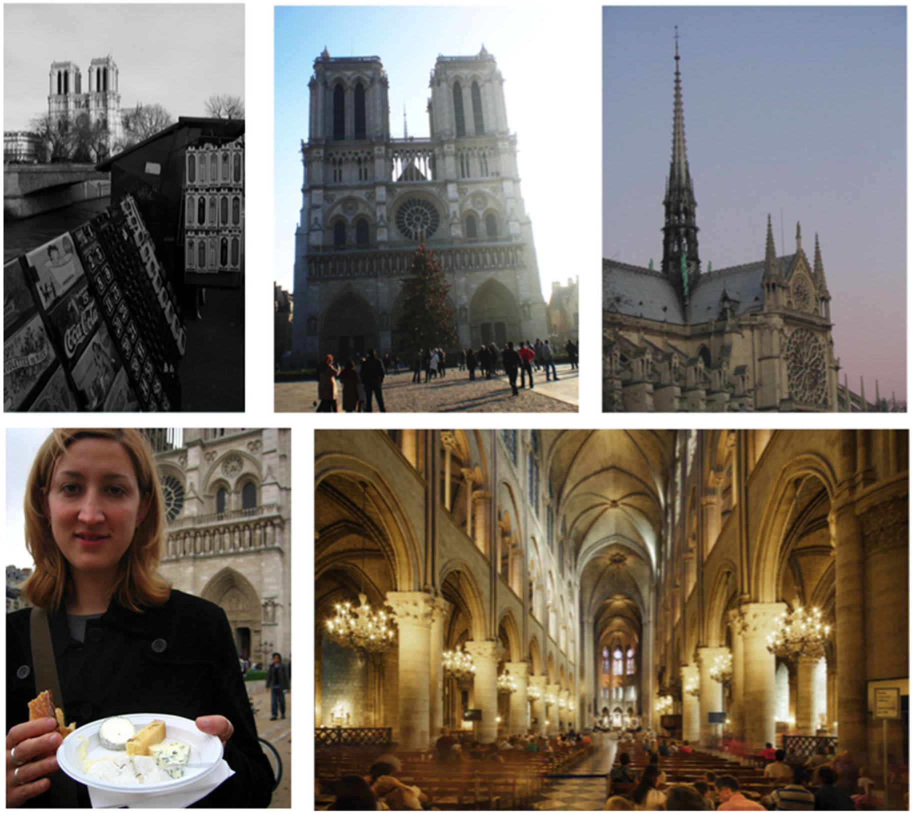


Fig. 2 Photos of "Notre Dame" from Flickr. *Top left* photo shows Notre Dame from a distance. *Top middle* and *right* show typical outside appearances of Notre Dame. *Bottom left* shows an example of a photo full with people. *Bottom right* shows the inside of Notre Dame

2 Related work

For exploring image connectivity in large image collections, several techniques have been proposed. In this section, we will explain previous techniques for discovering image pairs for matching.

As we mentioned briefly in Chapter 1, Photo Tourism [8] proposed utilizing large photo collections on the Web. With using exhaustive pairwise image matching, this approach obtains image connectivity graph. However, it takes too much time and computational effort to match every pair of images.

Some SfM methods detect candidate image pairs beforehand in order to avoid exhaustive matching. MatchMiner [11] is one of the methods used for selecting image pairs. It adopts the bag-of-visual-words and tf-idf weighting to estimate image pair similarity. For constructing bag-of-visual-words, MatchMiner trains a vocabulary tree on 50,000 images of a single city, which are not used in experiments for selecting image pairs, to yield one million visual words. This bag-of-visual-words is used for all experiments. Those visual features are extracted by SIFT descriptor, and approximately the closest visual word is assigned to every keypoint of each image. Images are represented by histograms of visual words, with the standard normalized tf-idf weighting applied to each histogram. Image pair similarity is evaluated by the dot product of their normalized tf-idf weighted histograms. In MatchMiner, a modified version of Rocchio's relevance feedback [17] is applied on the Top k most similar images for each query images.

Using the bag-of-visual-words and standard tf-idf weighting is a common method for estimating image similarity. Near Duplicate Image Detection [16] uses a bag-of-visual-words with tf-idf weighting method into image similarity measures as well. Originally, bag-of-visual-words and tf-idf weighting method is used in text retrieval. Before MatchMiner or Near Duplicate Image Detection, Video Google [18] applied bag-of-visual-words and tf-idf weighting method to image retrieval . In Video Google, visual vocabulary is constructed by using Mahalanobis distance and K-means clustering on SIFT descriptors extracted from a video.

A bag-of-visual-words with tf-idf weighting is known as a successful approach for image and particular object retrieval. The tf index indicates that the visual words which appear frequent in an image are important, while the idf index indicates that the visual words which appear among several images are less important. However, SfM system requires an image pair which shares same points, and thus the concept of the idf does not seem to be suitable for the purpose.

In this paper, we propose to exploit Jaccard Similarity, which shows how many visual vocabularies do image pairs have in common, for calculating image pair similarity in addition to tf-idf method. We demonstrate the superiority of our method on our original datasets, as well as on the dataset which was used for testing in MatchMiner.

3 Proposed method overview

We propose to introduce Jaccard Similarity in addition to the similarity based on tf-idf weighting. The outline of our method is as follows: (1) constructing a bag-of-visual-words from image collections, (2) estimating similarities using dot product of tf-idf, (3) estimating similarities using JacS, and (4) selecting the image pairs which are selected in both tf-idf and JacS. In this section, the algorithm for each step is explained.

3.1 Constructing a bag-of-visual-words

We build a bag-of-visual-words by random sampling from each image collection for each experiment. For each image, we pick 10% of features, which are extracted by SIFT descriptor, randomly for building visual vocabularies. Every descriptor is then represented as the most appropriate visual word by the nearest neighbor clustering. In this paper, we assume 10% of the features from one image are enough to express images approximately.

3.2 Similarity using tf-idf weighting

In text as well as in image retrieval [18], tf-idf weighting is commonly used to weight histograms of word frequencies bag-of-words. The weighting by tf-idf is computed by the following formula,

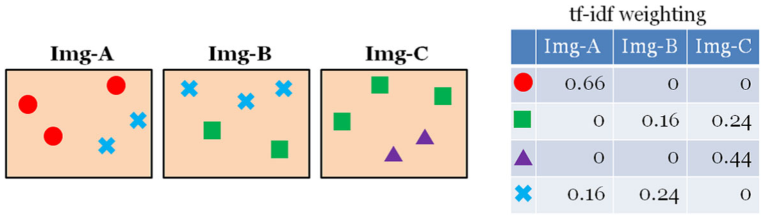
$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \tag{1}$$

where n_{id} represents the number of occurrences of the visual word i in the image d, n_d represents the total number of visual words which appear in the image d, N represents the number of images in the image collection, and n_i represents the number of images which include the visual word i in the image collection. Figure 3 illustrates an example of tf-idf weighting.

After every image is expressed as a vector of weighted visual words, the vectors are normalized and all pairwise image similarities are obtained by the dot product of the vectors. In Fig. 3, the similarity between Img-A and Img-B equals 0.20 while the similarity between Img-B and Img-C equals 0.27. Although Img-B has two features in common with both Img-A and Img-C, tf-idf-based similarity shows Img-B is similar to Img-C more than Img-A.

3.3 Jaccard similarity

We introduce Jaccard Similarity of images. JacS calculates the similarity of an image pair as the fraction of distinct visual words, which are common to an image pair.



tf-idf weighting

	Img-A	Img-B	Img-C
●	0.66	0	0
■	0	0.16	0.24
▲	0	0	0.44
✖	0.16	0.24	0

Fig. 3 An example of tf-idf weighting. Each visual word is weighted by their occurrences. The tf-idf weighting indicates *red circles* in *Img-A* and *purple triangles* in *Img-C* are important, while *blue crosses* and *green squares* which appear among the two images are less important

The image pair similarity by JacS is computed by the following formula,

$$s_{ij} = \frac{n_{ij}}{n_i + n_j - n_{ij}} \tag{2}$$

where n_i is the number of visual words, which appear in image i; n_j is the number of visual words, which appear in image j; and n_{ij} is the number of visual words, which appear in both image i and j. Figure 4 illustrates an example of JacS-based similarity. In this figure, visual words occurrences are the same as in Fig. 3. By using JacS-based similarity, the image pair A and B gets the same similarity as the image pair B and C.

3.4 Top *k* most similar images

With the similarity matrix obtained by the tf-idf-based similarity and JacS, we select image pairs and evaluate those methods. For each image as a query image, top k most relevant images are selected according to the similarity matrix and are assumed as the true image pairs for matching. Then, the accuracy of selecting image pairs is obtained by comparing the selected image pairs to the ground truth image graph. We show in the next section that false image pairs obtained by the tf-idf-based similarity differ from the false image pairs obtained by the JacS. Therefore, we propose to use the intersection of two image pair sets obtained by both similarity methods: tf-idf and JacS.

4 Experimental evaluation

4.1 Our original datasets

We first demonstrate tf-idf-based method and JacS on our original datasets: "Bear" and "Dolls," to show the difference of behaviors between the two methods.

4.1.1 Dataset "Bear"

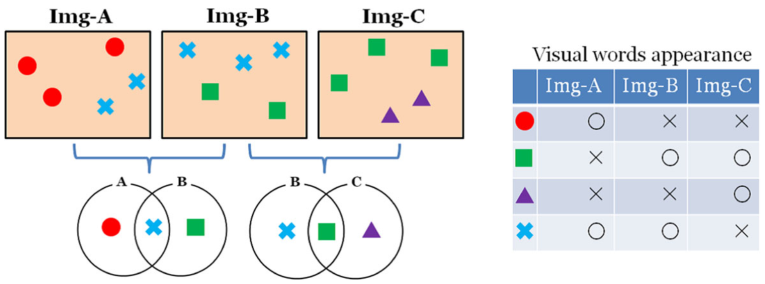
To compare the behaviors of tf-idf-based similarity and JacS, we first prepared photos of a figure taken from eight directions (Fig. 5).

The connected graph of dataset "Bear" is shown in Fig. 6. Each number on the vertex in the graph is corresponding to the number above the photos in Fig. 5. In this figure, tf-idf-based similarity shows better precision. The bear figure has similar texture on almost the entire surface. It appears that idf performs better here, while JacS tends to make more mistakes.

4.1.2 Dataset "Dolls"

Secondly, unlike the dataset "Bear," we prepared several images with different types of texture (Fig. 7).

Figure 8 shows the connected graph of dataset "Dolls." Black lines show false image pairs, while the other colors are corresponding to the frame colors in Fig. 7. In the figure, green vertexes have many connections with other colored vertexes in the tf-idf connected graph, while they does not have any connections with other colors in the JacS connected graph. The figure of green vertex has many similar feature descriptors, and thus JacS which simply



Visual words appearance

	Img-A	Img-B	Img-C
●	○	×	×
■	×	○	○
▲	×	×	○
✖	○	○	×

Fig. 4 An example of JacS. For the image pair A and B, *blue crosses* exist in common, while Img-A has *red circles* of it's own and Img-B has *green squares*. Totally, Img-A and Img-B has three visual vocabularies, and one of them appears in common. In this case, the similarity between Img-A and Img-B is 0.33

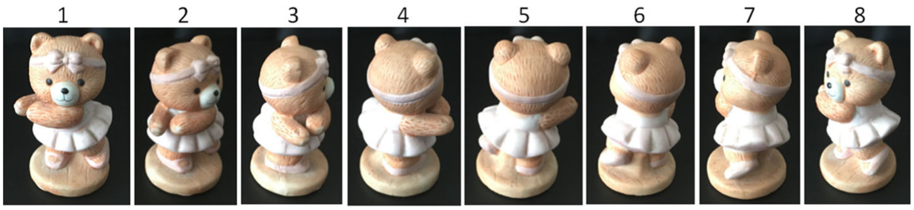


Fig. 5 Dataset "Bear". This dataset totally has eight images

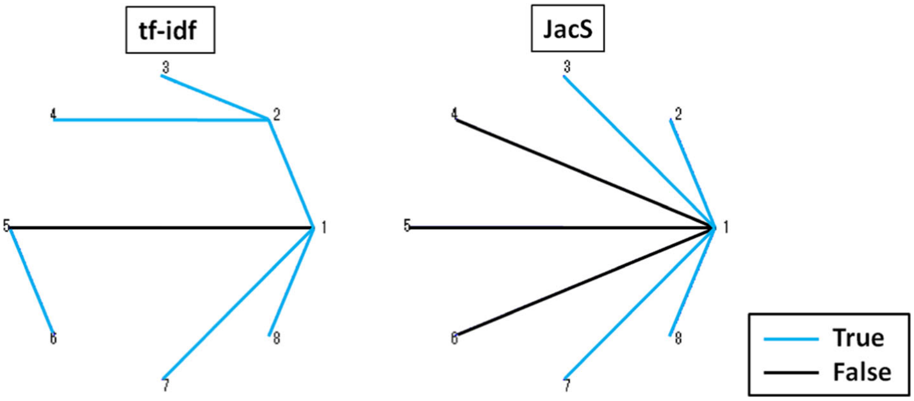


Fig. 6 Dataset "Bear" connected graph. Top $k = 1$. In this dataset, the next and two images away from a query image are assumed as true image pairs

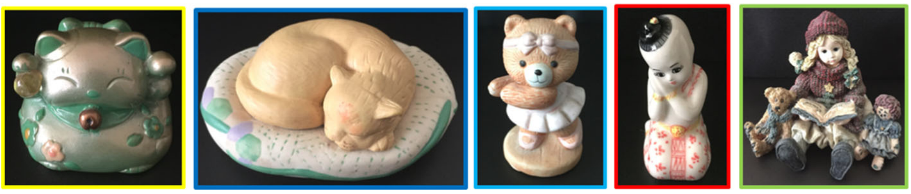


Fig. 7 Dataset "Dolls." Each figure is taken photos from eight directions. This dataset totally has 40 images

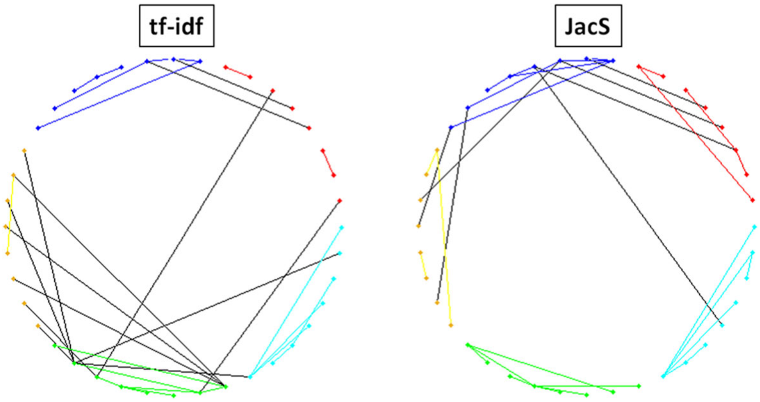


Fig. 8 Dataset "Dolls" connected graph. Top $k = 1$. In this dataset, we assume image pairs which show the same figure as true pairs

Fig. 9 Dataset "Pantheon". The images from Flickr tagged by "Pantheon" keyword. This dataset totally has 1123 images

estimate the number of shared visual words works better. It appears that the two methods obtain different false image pairs.

4.2 Dataset "Pantheon"

We finally evaluated the three methods: tf-idf-based method, JacS-based method, and our proposed method, with a dataset from MatchMiner [11]. In MatchMiner, the ground truth image graphs are computed by exhaustive geometric verification on all image pairs. We use this ground truth for evaluating our method. Figure 9 shows an example of the images in dataset "Pantheon".

With changing *k* value from 1 to 30, we select image pairs by each method. Then, selected image pairs are compared with the ground truth, and the average precision is computed for each *k* value (Fig. 10). The method tf-idf with JacS showed the best precision in every *k* value.

5 Conclusion

We have introduced Jaccard Similarity (JacS) for selecting image pairs for matching in the Structure from Motion (SfM). JacS considers occurrences of visual words in two images for selecting image pairs while the previous method based on tf-idf considers occurrences of visual words in the whole database. To confirm the differences of behaviors between the JacS and the method based on tf-idf, we have tested with two datasets: "Bear" and "Dolls." As a result of the two experiments, it appears that JacS and tf-idf-based method obtain different false image pairs.

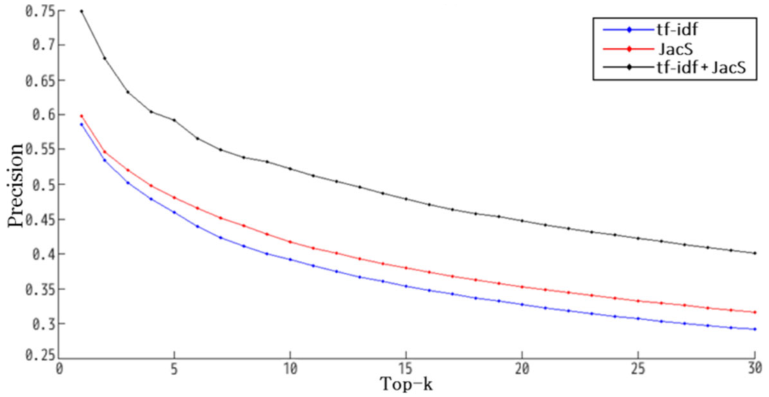


Fig. 10 Average precision at *k*. Each *line* represents as follows: tf-idf (tf-idf + JacS)

We have estimated image pair similarities by JacS and tf-idf-based method and have selected image pairs which are selected in both methods to make the accuracy higher. With the dataset "Pantheon," which is tested in MatchMiner [11] as well, our method has improved precision by 15%. We are now trying to extract connected components of high image similarities.

Acknowledgements
This work was partly supported by JSPS KAKENHI Grant Number 15X00445 and EU H2020-ICT-2016-2017 LADIO - Live Action Data Input and Output.

Authors' contributions
TK designed and carried out the experiments and wrote the manuscript. IS supervised the work and edited the manuscript. TP helped in the implementation and advised TK on the concept and experiments and edited the manuscript. All authors reviewed and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]Tokyo University of Agriculture and Technology, Tokyo, Japan. [2]CTU in Prague, FEE, Prague, Czech Republic.

Received: 21 February 2017 Accepted: 9 March 2017
Published online: 04 April 2017

References

1. Schaffalitzky F, Zisserman A (2002) Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In: European conference on computer vision. Springer Berlin Heidelberg. pp 414–431. http://link.springer.com/chapter/10.1007/3-540-47969-4_28
2. Havlena M, Torii A, Pajdla T (2010) Efficient structure from motion by graph optimization. In: European Conference on Computer Vision. Springer Berlin Heidelberg. pp 100–113. http://link.springer.com/chapter/10.1007%2F978-3-642-15552-9_8
3. Havlena M, Torii A, Knopp J, Pajdla T (2009) Randomized structure from motion based on atomic 3d models from camera triplets. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE. pp 2874–2881. http://ieeexplore.ieee.org/document/5206677/
4. Frahm JM, Fite-Georgel P, Gallup D, Johnson T, Raguram R, Wu C, Pollefeys M (2010) Building rome on a cloudless day. In: European Conference on Computer Vision. Springer Berlin Heidelberg. pp 368–381. http://link.springer.com/chapter/10.1007/978-3-642-15561-1_27
5. Wilson K, Snavely N (2014) Robust global translations with 1dsfm. In: European Conference on Computer Vision. Springer International Publishing. pp 61–75. http://link.springer.com/chapter/10.1007/978-3-319-10578-9_5
6. Cui Z, Tan P (2015) Global structure-from-motion by similarity averaging. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE. pp 864–872. http://ieeexplore.ieee.org/document/7410462/
7. Schonberger JL, Radenovic F, Chum O, Frahm JM (2015) From single image query to detailed 3D reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE. pp 5126–5134. http://ieeexplore.ieee.org/document/7299148/
8. Snavely N, Seitz SM, Szeliski R (2006) Photo tourism: exploring photo collections in 3D. ACM Trans Graph (TOG) 25(3):835–846. ACM
9. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
10. Heath K, Gelfand N, Ovsjanikov M, Aanjaneya M, Guibas LJ (2010) Image webs: Computing and exploiting connectivity in image collections. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE. pp 3432–3439. http://ieeexplore.ieee.org/abstract/document/5539991/
11. Lou Y, Snavely N, Gehrke J (2012) Matchminer: Efficient spanning structure mining in large image collections. In: Computer Vision–ECCV 2012. Springer Berlin Heidelberg. pp 45–58. http://link.springer.com/chapter/10.1007/978-3-642-33709-3_4
12. Agarwal S, Furukawa Y, Snavely N, Simon I, Curless B, Seitz SM, Szeliski R (2011) Building Rome in a day. Commun ACM 54(10):105–112
13. Chum O, Philbin J, Sivic J, Isard M, Zisserman A (2007) Total recall: automatic query expansion with a generative feature model for object retrieval. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE. pp 1–8. http://ieeexplore.ieee.org/document/4408891/
14. Chum O, Mikulik A, Perdoch M, Matas J (2011) Total recall II: query expansion revisited. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE. pp 889–896. http://ieeexplore.ieee.org/document/5995601/
15. Chahal M (2016) Information retrieval using Jaccard Similarity Coefficient. In: International Journal of Computer Trends and Technology (IJCTT)—Volume 36 Number 3. Seventh Sense Research Group. http://www.ijcttjournal.org/archives/ijctt-v36p124
16. Chum O, Philbin J, Zisserman A (2008) Near duplicate image detection: min-Hash and tf-idf weighting. In: BMVC. BMVA Press Vol. 810. pp 812–815. http://www.bmva.org/bmvc/2008/papers/119.html
17. Rocchio JJ (1971) Relevance feedback in information retrieval. In: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall Inc. pp 313–323
18. Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. ICCV 2(1470):1470–1477
19. Heller J, Havlena M, Jancosek M, Torii A, Pajdla T (2015) 3D reconstruction from photographs by CMP SfM web service. In: Machine Vision Applications (MVA), 2015 14th IAPR International Conference on. IEEE. pp 30–34. http://ieeexplore.ieee.org/document/7153126/