Cascade Hashing Structure

Cheng Guan

Jun 26, 2018

1. Cascade Hashing

In order to speed up image matching as quickly as possible, the cascade Hashing structure is composed of three layers: Hashing lookup, Hashing remapping and Hashing ranking, The flowchart of the method is shown in Fig. 1. The 3-layer hashing is designed to map the image's feature representation from coarse code to fine binary code, resulting in faster image matching. Besides, each layer of cascaded hash uses different measurement and filtering strategies to filter out some feature points noise in cross validation mode. In this sense, the proposed CasHash is not very sensitive to noise in 3D reconstruction [1].

1.1. Hashing Lookup with Multiple Tables

In the first step,the authors apply hashing lookup with short codes to do a coarse search. Especially, all the feature points in all images are embedded into m bits binary codes with LSH. For feature point p in image I, in order to find its matching point in image J, a lookup table with a set of buckets is constructed using the m bits binary codes. All the points that fall into the same bucket of point p in image J will be returned. This above procedure is called as hashing lookup.

Hashing lookup emphasizes its actual search speed, because its complexity is constant time. However, using a single hashing table with long bits, hashing lookup is often failed because Hamming space becomes thinner and thinner, and few samples fall in the same bucket. In this paper, the authors use multi-table strategy to solve this problem.In particular, the authors generate L functions $g_l\left(q\right) = \left(h_{1,q}\left(q\right), h_{2,q}\left(q\right), \cdots, h_{m,q}\left(q\right)\right), l = 1, 2, \cdots, L$, where $h_{s,l}\left(1 \leq s \leq m, 1 \leq l \leq L\right)$ are generated independently and uniformly at random from γ. In summary, the data structure is constructed with L hashing tables which have m bits, and then each point p is assigned to a bucket $g_l\left(p\right)$, for $l = 1, , L$.

1.2. Hashing Remapping

After coarse search in hashing lookup stage, the Euclidean distance between each candidate and query can be calculated accurately. However, the direct calculation of Euclidean distance requires a large amount of computation since the number of candidates is still large.

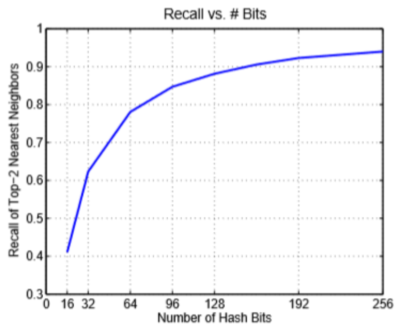

Figure 2. The recall of the two nearest neighbors when top 10 are returned while the number of bits changes

When applying LSH to the feature points, *e.g.*SIFT, the ranking performance is closely related to the number of bits. In LSH, it can be theoretically guaranteed that longer binary codes will result in more discriminative hashing expression, and thus better performance of ranking. We conduct an experiment on toy data with different number of bits. Recall of top 2 (in terms of Euclidean distance) is shown in Fig. 2 when top 10 (Hamming distance) is returned. In this paper, the authors remap the candidates into a 128-d Hamming space.

1.3. Top k Ranking via Hashing

In order to execute more precise search, in the third step, the authors expect to seek the k nearest neighbors in Hamming space, and then find the two nearest neighbors among them in Euclidean space. As the most commonly used method, the time complexity of linear search to find top k items is $O\left(kN\right)$. A tree-based data structure called min-heap was proposed to find the top k items and the time complexity is $O\left(\log\left(k\right)N\right)$ [2, 4].

After the top k candidates are found,the authors can find the two nearest neighbors among them according to the Euclidean distance, and matches that pass Lowes ratio test will be accepted [3].

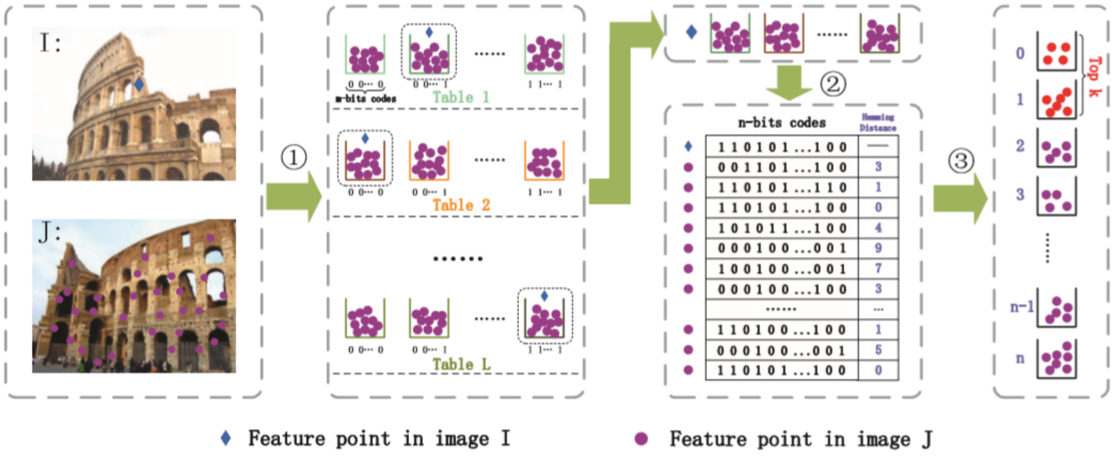

Figure 1. The flowchart of our proposed Cascade Hashing approach.

References

[1] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, 2008. 1

[2] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen. *Introduction to algorithms*. The MIT press, 2001. 1

[3] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[4] S. M. Seitz, R. Szeliski, and N. Snavely. Photo tourism: Exploring photo collections in 3D. *ACM TOG*, 2006. 1