Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction

Cheng Guan

Jun 24, 2018

Abstract

As we know,Image matching is one of the most challenging stages in 3D reconstruction. It usually takes up half of the computation cost, and inaccurate matching may lead to failure. Therefore, we need fast and accurate image matching for 3D reconstruction. The authors of this paper proposes a cascaded hash algorithm to speed up image matching. In order to speed up image matching, the cascaded hash method proposed is designed to be a three level structure, which is hash lookup, hash remapping and hash ranking.

1. Introduction

In the field of computer vision, 3D reconstruction is a classic and challenging problem. It should be used in different fields. In recent years, large-scale 3D reconstruction from community photo collection has become a new research topic, attracting more and more researchers. Researchers at home and abroad. However, 3D reconstruction is a huge computational cost. For example, it takes more than a day to rebuild an object with only one thousand pictures on a machine. In the Structure from Motion (SfM) model[1, 3].The 3D reconstruction pipeline can be divided into several steps: feature extraction, image matching, trajectory generation and geometric estimation. Among them, image matching occupies the major computational cost, and even exceeds half of it in some cases. In addition, inaccurate matching results may lead to reconstruction failure. Therefore, fast and accurate image matching is the key to 3D reconstruction.

Image matching techniques can be roughly divided into three categories: point matching, line matching and region matching. Because of its robustness to illumination change, affine transformation and view change, point matching has been widely paid attention, and many affective algorithms have been paid more attention over the past few decades [2, 5]. However,point matching is usually very time consuming. For pairwise image matching ,the computational complex of exhaustively comparing all feature points in two images is $O\left(N^2\right)$, where N is the average number of feature points in each image.

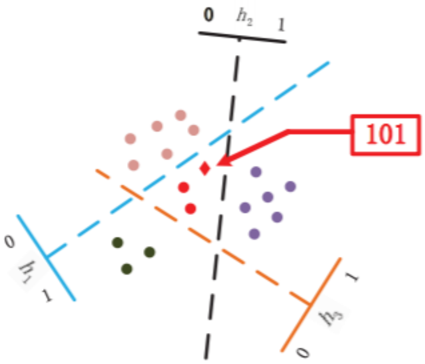

Figure 1. Locality Sensitive Hashing

2. The Proposed Approach

In this paper, the authors propose a Cascade Hashing structure, named CasHash, to speed up image matching for 3D reconstruction. In their method, a simple hashing algorithm, Locality Sensitive Hashing (LSH), is adopted to generate binary code (refer to Fig. 1). Consequently, the following is a brief introduction to LSH algorithm.

2.1. Locality Sensitive Hashing

Let $X = \{x_1, x_2, ..., x_n\}$ be a set of data points,where $x_i \in \mathbb{R}^d$. Given a query vector q, In order to find the most similar items in X to the query. LSH is perhaps the most well known hashing based ANN search scheme, which relies on the existence of locality sensitive hashing functions. Assume γ be a family of hashing functions mapping $\mathbb{R}^d$ to Hamming space $\mathbb{B}$. For any two points x and y, it chooses a function h from γ uniformly at random and is confined to the probability $h\left(x\right) = h\left(y\right)$. The function family γ is locality sensitive if it satisfies the following conditions: A family γ of functions from $\mathbb{R}^d$ to $\mathbb{B}$ is called $\left(R, cR, P_1, P_2\right)$-sensitive for $D\left(.,.\right)$ if for any $x, y \in \mathbb{R}^d$ in Eq.1 and Eq.2.

$$Pr_{h\in\gamma}\left(h\left(x\right)\right) \geq P_1, D\left(x,y\right) \leq R \tag{1}$$

$$Pr_{h\in\gamma}\left(h\left(x\right)\right)\leq P_2, D\left(x,y\right)\geq cR \tag{2}$$

where $D\left(.,.\right)$ is a distance function in the original space $\mathbb{R}^d$. Obviously, a family γ is valid only when $c > 1$, and $P_1 > P_2$. Given valid LSH functions, Gionis proved that the query time for retrieving $(1+\epsilon)$-near neighbors is bounded by $O\left(n^{\frac{1}{1+\epsilon}}\right)$ for Hamming distance [4].

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2011. 1

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features. *Computer Vision and Image Understanding*, 2008. 1

[3] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011. 1

[4] A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, 1999. 2

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