Universidade Federal Fluminense, Niterói, Rio de Janeiro, Brazil.

Fast and Robust Feature Matching

Ana Caroline Gomes Vargas, Cristina Nader Vasconcelos
ac_gomes@id.uff.br, crisnv@ic.uff.br

Introduction

From Machine Learning definition a feature is an individual measurable property of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial for effective Computer Vision algorithms, thus, many of them are being developed by different research groups. The result of a feature detection procedure over an image is a set of keypoints commonly matched to other sets extracted over different images. The crucial task that follows it in many pipelines is keypoints matching. Traditionally, the matching is obtained using a *k*-Nearest Neighbors modeling (*k*-NN). However, such approach does not model matching unicity restrictions, that is, it allows a keypoint to be associated more than once. We explore a parallel Bipartite Graph Matching (BGM) entirely in GPU for fast and robust matching and present its comparison against the k-NN. The results show that our proposal outperforms both in computing time and in statistics measurements strongly related to the unicity enforced by the bipartide graph: precision (BGM: 38% *k*-NN: 32%) and specificity (BGM: 41% *k*-NN: 24%).
Public code is available.*

Feature Dataset

Conceptually, any feature describing a set of keypoints in an application that demands unicity of keypoints on resulting pairs would be benefited by our matching model. As a replicable case study we explored the database proposed by Hauagge and Snavely[1] and their keypoint detection by local symmetry features, named SYMI, together with SIFT descriptors[2]. The database is composed by 48 image pairs and its corresponding homography, allowing a ground truth evaluation of the matching results once that given one point in the first image it is possible to transform its coordinates into the second image. It is a challenging dataset as images exhibit a range of dramatic variations in lighting, age, and rendering style.

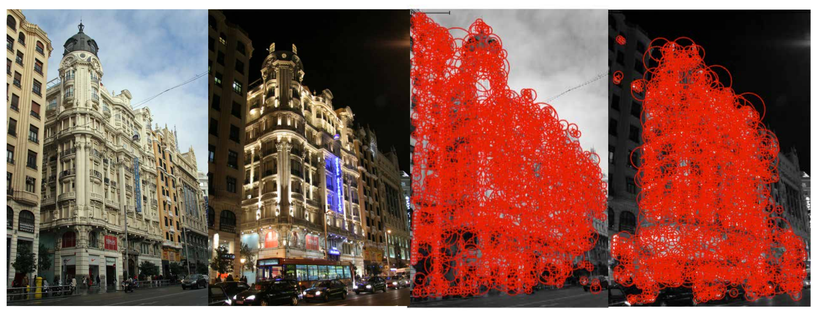

Figure 1 - Image pair from dataset [1] illustrating high degree of lighting variation and the result of keypoints detection.

Bipartite Graph Matching Definition

A bipartite graph is a graph *G* such that its set of vertices can be subdivided into two disjoint sets (*U* and *V*) and its edges connects a vertex in *U* to one in *V*. A matching *M* in G is a subset of its edges such that no node in G appears in M more than once.
In weighted graphs, each edge *(u, v)* is associated to a weight *w(u, v)* and the resulting matching weight is defined as the sum of the weights of edges in M.

The algorithm

The Auction Algorithm [3] solves the bipartite graph matching in parallel and has been originally adapted to GPU in [4]. It is semantically described as a real auction where persons compete for objects by raising their prices through competitive bidding.
In order to compute the feature matching, we assume that each of the n keypoints from the first image instantiate a person during the auction, while the keypoints of the second image are disputed as *m* objects. The order of the images is not relevant. During the auction each person is assigned to at most a single object and each object matched to at most a single person.
The goal of the auction is to assign persons to objects so as to maximize the total benefit.
We initially associate the desire of a person *i* for the object *j* with a constant value plus the negative of the distance between the descriptors corresponding to *i* and j features.
Differing to [4], our auction loops entirely in GPU until convergence through the bidding kernel followed by the assignment kernel and a third one responsible for evaluating convergence to decide if the auction is over or, in negative cases, it triggers a new auction round from within the GPU exploring modern GPU resources.

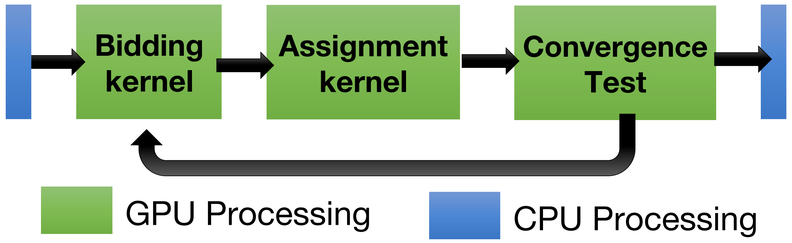

Figure 2 - Auction cycle

The bipartite graph model previously presented [4] do not deal with our case study natural restriction that some keypoints may be left unmatched even if there is still some free keypoints in the opposite keypoint set.
We model this variation creating a virtual node representing an object that is not associated to any real keypoint and connect it to a single person. Such model design also enforces a threshold for the descriptors distance, as above the virtual node value, real objects are no longer economically competitive.
We did not create virtual nodes for people, as objects that receive an offer at any time, never became free again according to the auction rules, thus, feature keypoints that may be considered free at the end of the auction have never received any bid.
Varying the virtual node value produces the following precision-recall curve. Assuming the virtual node price as 0, no keypoint is considered interesting for a matching thus all of them will be associated to the virtual node, while augmenting it more and more keypoints are turn into possible pairs, until all pairs may be considered interesting in the limit case. A virtual node may be matched even increasing its price, in cases where the number of people and objects is unbalanced. This is a practical issue, as the number of keypoints produced by the feature detector in each image of the dataset pairs vary.

Results

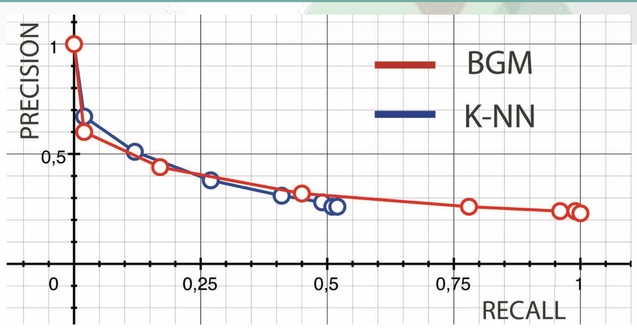

Figure 3: Precision-Recall curves

	Precision	*Accuracy*	*Sensitivity*	*Specificity*
BGM	37.86%	32.82%	26.96%	40.68%
k-NN	31.75%	33.33%	44.77%	24.34%

Table 1: Comparison Statistics.

In order to consider a keypoints pair as a truth positive, we adopted the intersection-over-union (IoU) criteria as in [1]. Such criteria broke the unicity assumption of the bipartite model, as more than one pair containing an specific keypoint may satisfy it. Such criteria favored the true positive statistics of the k-NN algorithm, as the bipartite matching results is limited to counting unique pairs, and k-NN is free to produce more pairs. It is expected that enforcing unique pairs to the ground truth dataset would enable a natural increase of true positive ratio obtained by our solution, but our choice was to replicate the matching criteria from the dataset [1].
Despite of the fact that the dataset used does not enforces unicity, the statics of the proposed matching remained stable under large variations of the virtual node value, and competitive to the *k-nn* solution, as illustrated in Table 1. It is expected that in a dataset that enforces unicity, the true negative ratio would be maintained, while the false negative ratio would diminish naturally, as its unique pair could be available for matching.
Timing comparisons were made against two implementation of the k-NN algorithm available in OpenCV library named Flann *k*-NN and brute force *k*-NN. The tests were computed using a Titan X **. The Table 2 shows that our solution results obtained a speedyup of approximately 12 times the Flann solution and 4 times the brute force *k*-NN.

# keypoints:	***BGM***	***Flann***	***Brute f. k-NN***
488-1999	8	82	28
2000-3999	17	179	61
4000-5999	71	829	276
6000-8512	149	1765	593

Table 2: Timing comparison (in ms*) for image pairs with increasing number of keypoints*

References

[1] D. C. Hauagge, N. Snavely. 2012. Image matching using local symmetry features. In Proc. of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12). 206-213. 2012.
[2] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91-110, 2004.
[3] D. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. Computational Optimization and Applications, 1(1):7-66, 1992.
[4] C. Vasconcelos, B. Rosenhahn. Bipartite Graph Matching Computation on GPU. In Proc. of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR '09). 42-55. 2009

**Public code: http://www.ic.uff.br/~crisnv/bgm2*
***We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X used for this research.*