

Parallel SURF Algorithm for 3D Reconstruction

Suping Wu* and Bing Feng

School of Information Engineering, Ningxia University, Yinchuan, 750021

*Corresponding author

Abstract—In this paper, we proposed a parallel SURF algorithm for 3d reconstruction to solve the problem of rapid feature extraction and matching in multi-view 3d reconstruction. SURF algorithm is an effective algorithm for feature extraction. Compared with the classic SIFT algorithm for feature extraction, it has improved somewhat in speed. However, SURF algorithm is still a time-consuming process. In this paper, we improved SURF algorithm. Aiming at the time-consuming problem of feature detection and description, we proposed a parallel algorithm based on multi-core OpenMP and CUDA architecture, and apply it to three-dimensional reconstruction. The experimental results show that the proposed algorithm achieves a certain acceleration ratio under the condition of 100% accuracy compared with the original algorithm.

Keywords—SURF; parallel algorithm; feature extraction; OpenMP; CUDA; 3D reconstruction

I. INTRODUCTION

Feature detection is widely used in visual fields such as target recognition and tracking, 3d reconstruction and image mosaic^[1-5]. Many algorithms for detecting and matching local invariant features of images have achieved good results. Harris^[6]corner point detection is the most famous feature point extraction method based on image gray value, but it cannot keep the scale unchanged. Difference-of-Gaussian (DoG) algorithm detects local extremum points as feature points in both two-dimensional plane space and scale space, and these feature points have good stability and satisfy the invariant scale. Harris algorithm tends to extract points with sharp gradient changes, while DoG algorithm generally extracts the center points of uniform region, also known as spots. Scale-invariant Feature Transform (SIFT)^[7]has obtained the scale and rotation in-variance. SIFT descriptor uses 3d histogram to conduct statistics on the direction of pixel gradient and modulus in a certain field of features, which has a strong anti-noise ability, but due to the high dimension, the speed of image matching is slow. In order to solve the timeliness of SIFT algorithm, Bay^[8] proposed a Speeded Up Robust Features (SURF) operator with a low description dimension and a fast speed.

Because the computational performance of GPU far exceeds that of CPU. For Harris algorithm, xiao han proposed a parallel algorithm based on the design idea of open computing language (OpenCL), with an acceleration ratio of 77x^[9]. For SURF algorithm, Terribery proposed a new GPU construction method of integrated image, discussed how to accelerate Gaussian template and Haar wavelet response, and optimized the Octave setting^[10]. POSIX thread^[11]was used to reconstruct the original algorithm. Some studies improved the running speed based on feature analysis^[12]and obtained the acceleration ratio of 2.5x+. Tang bin et al. proposed a Canny

parallel algorithm based on GPU and CPU heterogeneous computing platform, and the acceleration ratio could be 5.39x^[13]. Zhang jie et al. proposed a gpu-based HOG feature extraction and description algorithm, which achieved an acceleration of about 40x compared with CPU^[14]. Jie yuexin et al. proposed a blocking filtering algorithm based on DAGS+GPU, which achieved an average decoding acceleration ratio of 11-24x compared with sequential algorithm^[15].

3d reconstruction based on PMVS (Patch based multi-view stereo) method, Harris algorithm and DOG algorithm are adopted for feature extraction. SURF^[16]improved on the SIFT^[17] algorithm and was about three times faster^[18]. However, SURF still had a high time cost in 3d reconstruction of sequential images.

In addition, the deep learning method^[19-26] is also adopted in the 3d reconstruction problem, and good results have been achieved. However, the reconstruction accuracy based on feature extraction and matching method is higher. In multi-view reconstruction, there is a large amount of computation. In this paper, a hybrid parallel SURF algorithm was proposed to solve the problem of rapid feature extraction and matching in multi-view 3d reconstruction.

II. PARALLEL SURF ALGORITHM

The detection of the characteristic point for the SURF is based on the Hessian approximation matrix, and the characteristic points are detected using the maximum value of the determinant of the Hessian matrix. For Hessian matrix calculation, the second order Gaussian differential template is convolved with the image. In order to simplify the second order Gaussian differential template, the convolution of the template and the image is transformed into a box filter^[27] operation, so that the simplified template is only composed of several rectangular regions

For the response value of the box filter, it can be simply

solved by integrating the image^[28]. The specific integral image is defined as follows:

$$I_{\Sigma}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (1)$$

By integrating the image, it is to easily solve the grey value to a rectangular area of any size, i.e. to solve the response value of the box filtering. Therefore, the determinant value of the approximate Hessian matrix is calculated as follows:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (2)$$

Where $\det(H_{approx})$ is the determinant value of approximate Hessian matrix.

By approximating the determinant value of Hessian matrix

to represent the response value of feature points at a certain point x in the image, and traversing all pixel points in the image, the response image of feature points detection at a certain scale is formed. The pyramid image of multi-scale feature points response is formed by using the template size of different box filtering^[29]. 3D non-maximum suppression is carried out through pyramid image, and finally accurate coordinates of actual extreme points are obtained through sub-pixel interpolation.

$$\hat{x} = -\frac{\partial^2 H^{-1}}{\partial x^2} \frac{\partial H}{\partial x} \quad (3)$$

Feature description, SURF uses integral images to generate feature vector descriptions. Before feature description, it determines the principal direction.

In the circular neighborhood with the radius of $6s$, the Haar wavelet response values in the x and y directions of the image are calculated, and the sampling step is set as s , the size of the wavelet is set as $4s$, where s refers to the scale of the feature points to be detected.

When the wavelet response value is calculated, use the Gaussian weighting function of $\sigma=2s$ to weight it. To obtain the principal direction value, a fan-shaped sliding window of a size of $\pi/3$, centered on the feature point, is rotated to slide the window at a step length of 0.2 radian. The horizontal and vertical directions of the response values of the image Haar wavelet in the sliding window are summed respectively. The two responses sums generate a new vector, and the corresponding direction of the longest vector is the principal direction. The $20s * 20s$ image was divided into $4 * 4$ sub-blocks along the principal direction, and the Haar template with size $2s$ was used for the response value calculation for each sub-block.

For each sub-block, Haar template of size $2s$ is used to calculate the response value, and the response value is counted, and the 4-dimensional vector $(\sum dx, \sum |dx|, \sum dy, \sum |dy|)$, dx and

dy are the response value of the Haar wavelet in the x and y directions. Therefore, SURF feature points are described as $4*4*4 = 64$ -dimensional vectors. Finally, the response was summed and normalized.

SURF algorithm needs to determine the principal direction of the detected feature points. The determination of the principal direction needs to calculate the Haar wavelet response and then move through the sliding window to calculate the response sums. The step length of each move is 0.2 radian. After the principal direction is determined, the image is divided into blocks along the principal direction, and the Haar wavelet is conducted again for response and statistics, and Gaussian

weighting is carried out for each response. Although the response time of each Haar wavelet is very short, the statistical time of multiple responses is relatively large.

A. Multi-core Parallel

Since the time-consuming operations are all related to integral image, Row-first prefix sum of integration image may be calculated first, On this basis, the prefix sum^[30] of the column are calculated. Because the prefix sums have data correlation, it is not been to directly for calculation.

On multi-core shared storage machines, the method of Callahan D^[31] has certain advantages, which is implemented through OpenMP in this paper. In order to solve the problem of evaluating the first row and first column of the integral image, add 1 to both the column and the row of the integral image. For the prefix sums of each row and column, due to the data competition, the critical operation in OpenMP may be used to ensure their independent execution.

Due to the resource consumption of allocation thread in parallel processing, therefore, the judgment of the number of cores and threads before the parallelization operation is conducted, and the parallel operation is conducted when the number is greater than 2.

The localization of feature points is mainly divided into three steps: feature point detection, non-maximum suppression, and accurate location of feature points. Firstly, feature point detection is carried out for images at all scales. For the construction of scale space, feature point detection for each group and each layer is independent of each other and can be processed in parallel. 26 neighborhood response points in the scale space (scale up and down all the 9 neighborhood, the scale of 8 neighborhood) as candidate points, then the nonmaximal suppression is Carried out, Although involving up and down two neighborhood, but whether relate only to detect the feature points of discarded, so as long as each thread only needs to read 26 response points from three adjacent scales for comparison, there is no data correlation and can be processed in parallel. Finally, the accurate positions of feature points are calculated by interpolation. Due to the data competition, the critical operation in OpenMP is used to complete.

DoH response scale space is constructed in parallel, and the pixels in the scale group are judged as extreme points by using the adjacent 26 neighborhood of scale space, and the precise coordinates of feature points are calculated by interpolation.

Considering that the problem of data competition is extremely easy to occur when the determination of the principal direction of a single feature point is parallel with the description of the feature point internally, it needs to be completed by the critical operation, which increases the resource consumption. Moreover, the determination of the principal direction of a single feature point and the description of the feature point take less time, since the allocation of threads will cause resource consumption and slow down the calculation, only the detected feature points are used. The principal direction of feature points and the description of feature points are obtained by detection of all feature points.

B. Hybrid Parallel

Feature point extraction and matching is the first step of 3d reconstruction. In this method, SURF algorithm is used as the feature point algorithm, and a series of initial candidate points were obtained through initial matching with polar line constraints.

The Image under CUDA is constructed as follows:

```

1. struct cudaImage
2. {int    width;    /* Image width */
3. int    height;   /* height of image */
4. char*  data;     /* image data pointer */
5. size_t widthStep; /*The size (in bytes) of the rows
in the image */
};

```

In order to consider more tasks to the GPU, the matrix transpose and the RGBA image into grayscale image are uploaded to the GPU to processing, in addition, for the two prefix sums of integral image are solved with the cudpp library.

The solution of feature points is the same as that performed on the CPU. Calculate the approximate Hessian matrix determinant value, calculate the 3*3*3 non-maximum suppression on the device, download the feature point to host, and write the feature point group.

The feature points processed in CUDA are reused, so as to build a separate structure

```

typedef struct
{ float x;
float y;
float scale;
int laplacian;
} fasthessian_cudaPoint;

```

The feature detection of each Octave is independent of each other. GridDim is used to correspond to the current processing layer, that is, each layer is processed by a GPU function. In the feature detection process of each layer, integral images are mainly constructed. The box filter is calculated, the response value of box filter is normalized, symbol of Laplace operator and the determinant of the Hessian matrix are calculated. the solution of the integral image is mainly two prefix sums using CUDA cudpp library function to solve, box filter solution is by building a good integral image to do add and subtract, then box filter response values are conduct normalization processing, and save the Laplace's symbol, the Hessian determinant calculation, corresponding to a blockDim, through the use of share memory instead of using a global memory.

The processing of blocks is allocated to a streaming multiprocessor for processing. For the experimental environment GPU is Fermi architecture, each SM has no more than 8 blocks and no more than 1536 threads. In this way, if the size of blocks is 16 times 16, so that a SM can load 6 blocks, so

that the GPU performance can be fully played. Another thread is wrapped as a Wrap to execute, Each Wrap is usually 32 threads, so if the number of threads per line is 16, it is a good way to incorporate global memory access and also to have good control over instruction divergence.

For character description, it is due to build good integral image often read, Texture memory can process some images and has a cache, which is faster than global memory. Store the created integral image in the texture memory. Calculating the principal direction of the direction of the longest vector is to synchronize operations.

The cumulative operation of computing the feature point description also needs to wait until all the gradient calculations are completed, which also needs to be synchronized.

Finally, the vector normalization of feature description is carried out, and the feature vector is 64 dimensions. A block is used to process a feature vectors, that is, a block is used to process a feature point. First, 16 sub-blocks are calculated through 4 threads, and then the results of 4 threads are cumulative to calculate the reciprocal of its square root. Finally, feature vectors are normalized through 64 threads.

In the process of 3d reconstruction based on multiple views, the processing of multiple images is completed by CPU, and the calculation of feature points and feature description of each image is completed by GPU.

III. EXPERIMENTS

In this paper, the experimental platforms includes a laptop Intel dual-core four-thread processor i5, the main frequency 2.5ghz and a desktop Intel quad-core quad-thread processor Q9400, with the main frequency of 2.67ghz. The system memory is 4G, the operating system Windows7, the development environment is Visual Studio 2010, and the OpenMP environment is the environment configured in Visual Studio. This article uses the test images from CMVS^[8]algorithm of test image(<http://www.di.ens.fr/cm-vs/>), order image size is 640*480,1400*1300.

The running time unit of the algorithm in the table is milliseconds (ms). The test time is the calculation time of the algorithm. The input and output time is not compared. Each time value is the average of 10 measurements. In the experiment, the acceleration ratio is a measure of the efficiency of the algorithm.

In order to make the algorithm able to extract features of multiple images quickly and effectively, feature extraction of image sequences in 3d reconstruction is solved, and one or several images are allocated to a thread for processing.

In this paper, after parallel processing of multiple images, the average execution time of each image was measured and compared with the execution time of one image.

A. Multi-core Parallel Experiments

The parallel algorithm was applied to 30 images for multi-core experiments, and the comparison diagram of laptop and desktop acceleration ratio was presented as shown in fig.1 and fig.2. experimental results show that the acceleration ratio is increasing with the increase of the number of cores. In 3d

reconstruction based on image sequences, the parallel algorithm is stable in both laptop and desktop, and has certain scalability and robustness.

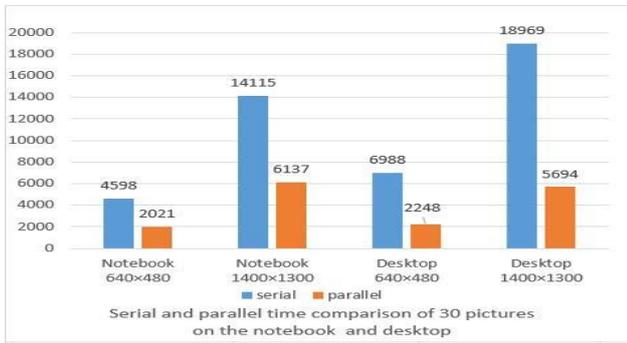


FIGURE I. SERIAL AND PARALLEL TIME COMPARISON OF 30 PICTURES ON THE NOTEBOOK AND DESKTOP ON MULTI-CORE OPENMP

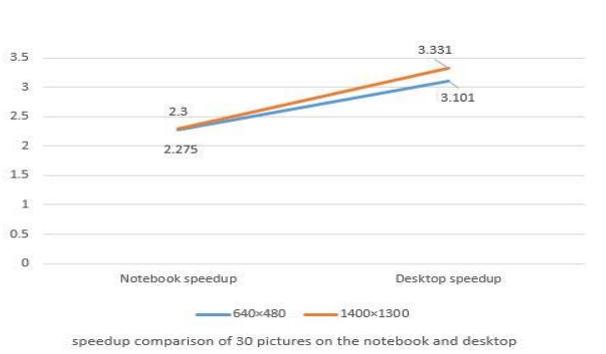


FIGURE II. Speedup comparison of 30 pictures on the notebook and desktop based on Multi-core OpenMP

B. Hybrid Parallel Experiments

According to the above process, 30 images for hybrid (based on Multi-core and CUDA) parallel computing, the experiment results show in fig.3. and fig.4., which show that the acceleration ratio is increasing with the increase of the size of images. In addition, in 3d reconstruction based on image sequences, the parallel algorithm has certain scalability and robustness. For hybrid parallel, the acceleration ratio has been improved to some extent, which also gives full play to the computing power of CPU and GPU.

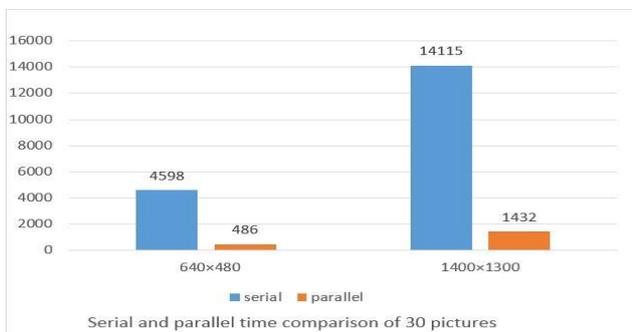


FIGURE III. SERIAL AND PARALLEL TIME COMPARISON OF 30 PICTURES ON THE NOTEBOOK AND DESKTOP BASED ON MULTI-CORE AND CUDA

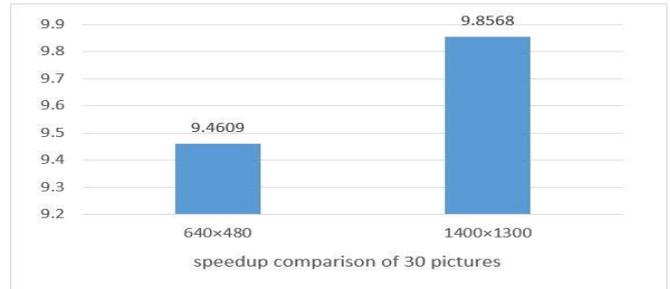


FIGURE IV. SPEEDUP COMPARISON OF 30 PICTURES ON THE NOTEBOOK AND DESKTOP BASED ON MULTI-CORE AND CUDA

In addition, fig.5. and fig.6. show the feature extraction of the serial and parallel algorithms in the original experimental picture. The green line represents the direction and the blue circle represents the light background. The parallel algorithm can guarantee 100% accuracy compared with the original algorithm.



FIGURE V. THE FIRST PICTURE SERIAL SURF ALGORITHM FEATURE EXTRACTION



FIGURE VI. THE FIRST PICTURE PARALLEL ALGORITHM FEATURE EXTRACTION

IV. CONCLUSION

In this paper, we have proposed a parallel SURF algorithm for 3d reconstruction to solve the problem of rapid feature extraction and matching in multi-view 3d reconstruction. In view of feature detection and feature description time-consuming parts, our parallel algorithm uses multi-core OpenMP and CUDA architectures, and the algorithm was applied to 3d reconstruction. The experimental results have demonstrated the effectiveness of the proposed approach on images from CMVS^[32] algorithm of test images. One desirable direction of this work is to optimize parallel strategy of the determination of the principal direction of the feature point and the accumulation part of the feature point description.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 61662059.

REFERENCES

- [1] Bao S.Y., et al. Dense object reconstruction with semantic priors. in *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE Conference on. 2013:IEEE.
- [2] Delaunoy A. and M. Pollefeys. Photometric Bundle Adjustment for Dense Multi-View 3D Modeling. in *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on. 2014:IEEE.
- [3] Shan Q., et al. Occluding Contours for Multi-view Stereo. in *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on. 2014: IEEE.
- [4] Dou M., Taylor J., Fuchs H., et al. 3d scanning deformable objects with a single rgbd sensor[C]//*Computer Vision and Pattern Recognition (CVPR)*, 2015 IEEE Conference on. IEEE, 2015:493-501.
- [5] Choi S., Zhou Q. Y., Koltun V. Robust reconstruction of indoor scenes[C]//*Computer Vision and Pattern Recognition (CVPR)*, 2015 IEEE Conference on. IEEE, 2015:5556-5565.
- [6] C. Harris, M. Stephens. A Combined Corner and Edge Detector. 4th Alvey Vision Conference. 1988, 147-151.
- [7] M. Grabner, H. Grabner, H. Bischof. Fast approximated SIFT[C]. *Proceedings of the 7th Asian Conference on Computer Vision, Hyderabad, India, 2006, Part I*, 918-927.
- [8] H. Bay, A. Ess, T. Tuytelaars, et al. Speeded-up robust features (SURF)[J]. *Computer vision and image understanding*, 2008, 110(3):346-359.
- [9] Xiao Han, Ma Ge, Zhou Qinglei. Harris corner detection algorithm on OpenCL architecture[J]. *Computer Science*, 2014, 41(7):306-309.
- [10] Terriberry T B, French L M, Helmsen J. GPU accelerating speeded-up robust features[J]. *3D Data Processing, Visualization and Transmission*, Georgia Tech, Atlanta. 2008.
- [11] Zhang N. Computing parallel speeded-up robust features (p-surf) via posix threads. *Emerging Intelligent Computing Technology and Applications*. 2009: 287-296.
- [12] Zhang N. Computing optimised parallel speeded-up robust features (p-surf) on multi-core processors. *International Journal of Parallel Programming*. 2010, 38(2): 138-158.
- [13] Wang J, Zhang SY, Liu T, etc. FPGA-based embedded multicore processors and parallel SUSAN algorithm. *Computer Engineering*, 2008, 31 (11) : 1995-2004 (in Chinese with English abstract).
- [14] Wang ZG. Local feature GPU-accelerated algorithm research and implementation of SURF [MS. Thesis]. Tsinghua University, 2011 (in Chinese with English abstract).
- [15] Callahan D. Recognizing and parallelizing bounded recurrences. *Languages and Compilers for Parallel Computing*, Springer, 1992, 169-185.
- [16] Lowe D G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. 2004, 60(2): 91-110.
- [17] Bay H, Tuytelaars T, Van Gool L. Surf: Speeded up robust features. *Computer Vision-ECCV 2006*. 2006: 404-417.
- [18] Bauer J, Sunderhauf N, Protzel P. Comparing several implementations of two recently published feature detectors. In: *Proc. of the International Conference on Intelligent and Autonomous Systems*. 2007.
- [19] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322-337. Springer International Publishing, 2016. 2
- [20] Liu, F., Shen, C., Lin, G.: Deep convolutional neural networks for depth estimation from a single image. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. (2015)
- [21] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3D SURF for robust three dimensional classification. In *Proc. ECCV*, 2010. 2
- [22] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. In *3D Vision (3DV)*, 2016.
- [23] [23]54. A. T. Tran, T. Hassner, I. Masi, and G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *CVPR*, 2017.
- [24] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.
- [25] P. Dou, S. K. Shah, and I. A. Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. In *IEEE Conference on CVPR*, 2017.
- [26] C. Hane, C. Zach, A. Cohen, and M. Pollefeys. Dense semantic 3d reconstruction. *IEEE TPAMI*, 39(9):1730-1743, 2017.
- [27] Simard P Y, Bottou L, Haffner P, et al. Boxlets: a fast convolution algorithm for signal processing and neural networks[J]. *Advances in Neural Information Processing Systems*. 1999: 571-577.
- [28] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2001. 511-518.
- [29] Lowe D G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. 2004, 60(2): 91-110.
- [30] Wang ZG. Local feature GPU-accelerated algorithm research and implementation of SURF [MS. Thesis]. Tsinghua University, 2011 (in Chinese with English abstract).
- [31] Callahan D. Recognizing and parallelizing bounded recurrences. *Languages and Compilers for Parallel Computing*, Springer, 1992, 169-185.
- [32] Furukawa Y, Ponce J. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on. 2010, 32(8): 1362-1376.