Parallel algorithms for stereo vision – shape from stereo

XingJie Ruan, Xenophon

Institute of Information
& Mathematical Sciences
Massey University at Albany,
Auckland, New Zealand
`Xingjie.Ruan.1@uni.massey.ac.nz`

Parallel algorithms for stereo vision – shape from stereo

XingJie Ruan, Xenophon

Institute of Information
& Mathematical Sciences
Massey University at Albany,
Auckland, New Zealand
Xingjie.Ruan.1@uni.massey.ac.nz

This paper proposes the parallel algorithms for stereo vision. The author present the concept of stereo vision, the existing stereo vision system in the world, the advantage of using parallel in stereo vision and the parallel algorithms for stereo vision

Keywords: parallel; stereo vision; algorithms; correspondence.

1 Introduction

Almost all the human beings got the stereo vision. People know objects are moving in the recognition of stereo vision. Parallel processing has been used in computer vision over the past two decades. And also, nowadays, stereo vision becomes an important part of computer vision. Stereo vision used in many areas of computer vision. For example , stereo vision can be used in robots, the robot can use their stereo vision to detect the distance between itself and the target object. Stereo vision can be used in the security system, to join two or more images together, so that can increase the range of vision. In the another hand, how to build stereo vision in real-time becomes a big problem. Whatever in the case of robot or the case of security system, speed becomes more and more important. With paralleling the algorithms for stereo vision, we can get 2-3 times even 20 times speed up.

2 Stereo vision

2.1 Concept of Stereo vision

Stereo vision (Stereopsis) is the process in visual perception leading to the sensation of depth from the two slightly different projections of the world onto the retinas of the two eyes. [1]

2.2 Human Stereo vision

Human has two eyes and one head, each eye can capture one image in a really short time. And
the eyes send this two images to out brain. After the computation in the brain, the brain will show us what we saw. There is one way to confirm the images captured by eyes are different: close left eye to capture the image from the right eye , then close the right eye to capture the image from left eye. According to the contradistinction, there are some slightly differences between two images. By the result of the contradistinction, human knows that the brain is not just only mix two images together, but mix them by some methods.

Human are able to deduce depth information from differences of a scene as seen by left and right eye. For example, if object one is moving on object two, object two is seen by human but some parts of object is not seen by human (if object two is bigger then object one). Then human know that: object one is closer.

2.3 Stereo vision in computer vision

Stereo vision in computer vision is almost the same as stereo vision in human. Stereo vision in computer vision and stereo vision in human both work on the same principle. We can set up two camera as people's eyes, one computer as people's brain. Each camera can capture one image and send them to the “brain” , which is the computer. Then the computer need some methods to mix the images together. Then the result will present on the screen.

2.4 Contradistinction between human stereo vision and computer stereo vision

Human stereo vision and computer stereo vision is almost the same. That's why stereo vision can be presented by computer. But they are not totally the same. Because the computer do not know what people 's brain do. So it is necessary to “tell” the computer, which methods human used.

3 Parallel in stereo vision

Nowadays, stereo vision is not only use in mixing two images together but also used in many areas. For example, we use stereo vision on a robot, to simulate what human can see by their eyes. With this vision, robot can do some tasks for human. Such as pick up one object in the labyrinth, clean the inside of the pipe in some dangerous places. Speed is very important aspect for stereo vision process in real-time. It is necessary to parallel it.

3.1 History

There are two ways to speed up the real-time stereo vision, the first way is use some hardware to speed up. The other way is to parallel the algorithms for stereo vision.

3.1.1 Hardware

There are some existing hardware architectures in the world. One of the hardware architectures is created by J.Falcou, J.serot, T.Chateau and F.Jurie in 2005 [2].

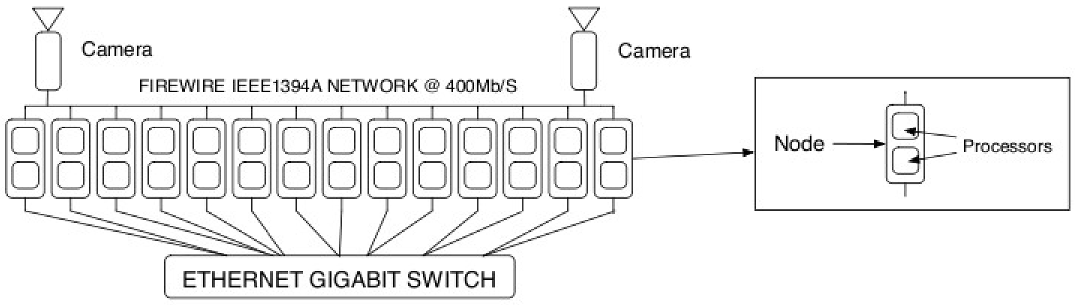


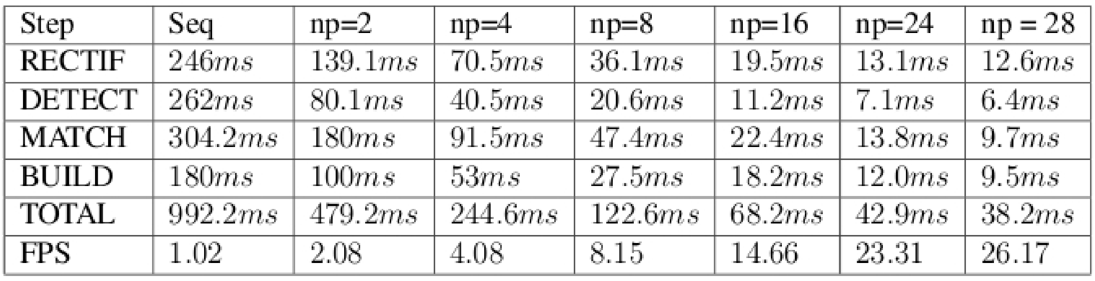
Figure 1: Cluster Architecture.

The Figure 1 (from [2]) shows the cluster architecture. It includes 14 computing nodes. Each node has a dual-processor Apple G5 running at 2GHz and 1Gb memory. All nodes connected to a gigabit ethernet switch. And the video streams is provided by two cameras via the FIREWIRE IEEE1394 NETWORK. It has a maximum-speed at 400Mb per second.

The cluster is doing the job what people's brain do. The camera can capture images and provide the video streams enough for the processing. After each node received the images, they can work on the image, get the result and then send back to the master node. The result will present to the user on the output drive such as screen or printer.

But there is a synchronization problem: each node must be work on same frame in the video stream. On the other word, it must wait until each node finished the task on the current frame. In OpenMPI library we can use MPI_Barrier to make sure all the nodes received the images and work on the right images.

Here's the result of the cluster Architecture:



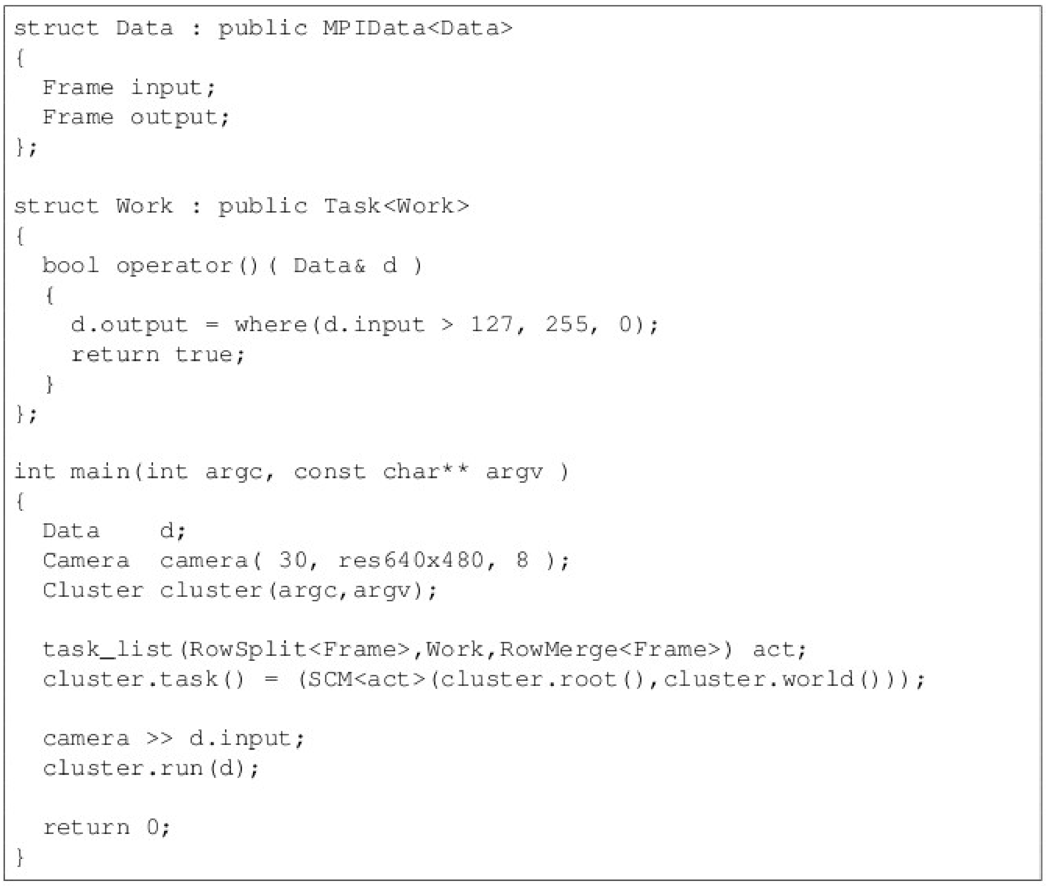
Step	Seq	np=2	np=4	np=8	np=16	np=24	np = 28
RECTIF	246ms	139.1ms	70.5ms	36.1ms	19.5ms	13.1ms	12.6ms
DETECT	262ms	80.1ms	40.5ms	20.6ms	11.2ms	7.1ms	6.4ms
MATCH	304.2ms	180ms	91.5ms	47.4ms	22.4ms	13.8ms	9.7ms
BUILD	180ms	100ms	53ms	27.5ms	18.2ms	12.0ms	9.5ms
TOTAL	992.2ms	479.2ms	244.6ms	122.6ms	68.2ms	42.9ms	38.2ms
FPS	1.02	2.08	4.08	8.15	14.66	23.31	26.17

Figure 2: Result of Cluster Architecture.

From the result of the cluster Architecture (Figure2 form [2]), if the task process by using single processor. It will take 992.2ms. With 2 nodes it return the result 479.2ms. And it get the 38.2ms when using 28 nodes. That means it almost got 30 times speed up. A normal video got 30fps , and with the 28 nodes on processing the task, it almost got the quality of the real-time video.

3.1.2 Software

Paralleling in programming takes a very important part in stereo vision. Here are some of the example codes which are working on the Cluster Architecture.



```
struct Data : public MPIData<Data>
{
  Frame input;
  Frame output;
};

struct Work : public Task<Work>
{
  bool operator()( Data& d )
  {
    d.output = where(d.input > 127, 255, 0);
    return true;
  }
};

int main(int argc, const char** argv )
{
  Data    d;
  Camera  camera( 30, res640x480, 8 );
  Cluster cluster(argc,argv);

  task_list(RowSplit<Frame>,Work,RowMerge<Frame>) act;
  cluster.task() = (SCM<act>(cluster.root(),cluster.world()));

  camera >> d.input;
  cluster.run(d);

  return 0;
}
```

Figure 3: Software Architecture,A simple binary thresholding application using in cluster architecture[2]

According to the Figure.3 [2], the software architecture contains a data structure for the input and output. The example code received the frame from the cameras via FIREWIRE Network, create a task on the cluster world then process the task on the cluster. After the node finish processing the data, the nodes will send back the data to the root which means the master, and the master will output the data.

4 Parallel algorithms for stereo vision

Parallel algorithms for stereo vision is based on the stereo vision algorithms. It does the same thing what stereo vision algorithms do. Only different is we will separate the whole task into some small independence task in the master processor, send those small tasks to each slave node and process in the node, get the result back from the slave node, then join them together. Build up the hardware is quite expensive, so parallel the algorithms is a good way to speed up.

4.1 Analysis

There are four steps to get the stereo vision in computer vision: Set up the environment, Calibrate on the camera, finding correspondence and compute depth.

4.1.1 Environment

To get the stereo vision, it's necessary to have two cameras and one computer. There are some variables in the stereo vision. Distance of two cameras, the angel from the cameras to the object and any variables related to the cameras and object.

4.1.2 Calibrate on the camera

Nothing is perfect in the world. Same as the cameras. Because of the mirror in the camera are not the same, in some case, a beeline in the real-world will present as the curve in the image. In this situation, it is necessary to calibrate the cameras. From the result of calibrating the cameras, they should present the same beeline in two images.

4.1.3 Finding correspondence

The correspondence problem means how to conjugate points in the 3-d in two images. One point in the image captured by left camera may be anywhere in the image captured by right camera. It can be paralleled. Because the point which is the pixel from left camera is not depend on any pixel in the left camera.

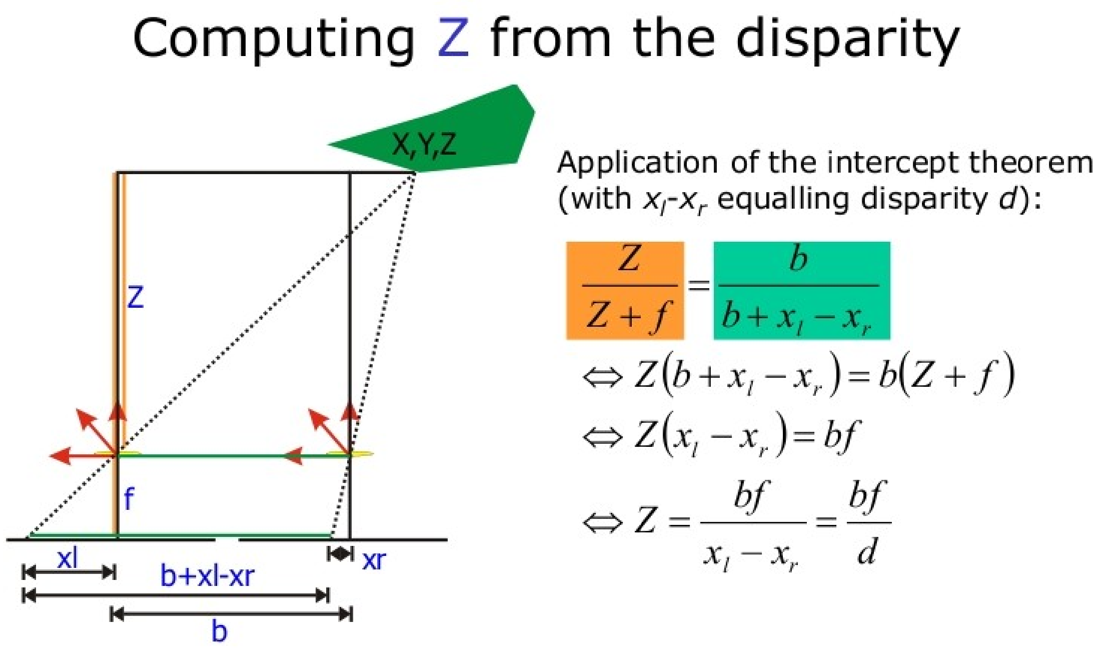
4.1.4 Compute depth

We can know how close is the object because our brain can compute depth of the image. The part is related to the computation by triangulation[3].

4.2 Methodology

4.2.1 Compute depth (Triangulation)

The methodology used for compute depth is show as below.



Application of the intercept theorem (with x_l-x_r equalling disparity d):

$$\frac{Z}{Z+f} = \frac{b}{b+x_l-x_r}$$

$$\Leftrightarrow Z(b+x_l-x_r) = b(Z+f)$$

$$\Leftrightarrow Z(x_l-x_r) = bf$$

$$\Leftrightarrow Z = \frac{bf}{x_l-x_r} = \frac{bf}{d}$$

Figure 4: Compute depth [3]

From the Figure4 [3] , all the variables are known when build up the environment. We can get depth by using the formula in the Figure4.

4.2.2 Camera Calibration

The common method to calibrate a camera is to take a image of a chessboard. First to do is Canny edge detection , after the detection, fitting the straight line to detect linked edges, then intersecting the lines to obtain the image corners, match image corners and 3-D target chessboard corners which visible in image. Then we can get the pairs of match point.

4.3 Implementation

From the section.3 in this report. It tells us all the works are almost independent. Each task is not related to another task. It is possible to do that in parallel. For example, in finding correspondence, the master processor first get a pair of images, broadcast the left image to the slave node. Then separate the right image into pieces. Send those pieces to the slave nodes. The task of each slave node is to find out the pixels in the left image which match their own data (pixels).

5 Conclusion

This paper show that the architecture of the cluster for real-time stereo vision, the concept and algorithms for the stereo vision. In the result of the research, we know that it is possible to implement parallel in stereo vision and obvious that there is speed up. Real-time stereo vision problems can be solved by parallel algorithms for stereo vision.

References

[1] Retrieved 15 MAY 2009. “Stereopsis,” in *Wikipedia,The free Encyclopedia*,

From http://en.wikipedia.org/wiki/Stereo_vision

[2] J. Falcou, J. Serot, T. Chateau, F. Jurie, “A Parallel Implementation of a 3D Reconstruction Algorithm for Real-Time Vision.” in *Parallel Computing 2005* ,2005

[3] Klaus D.Toennies., “5.Stereo Vision (Introduction),” in *3D Computer Vision*, University Magdeburg

[4] R.Hartley and A. Zisserman, “Multiple View Geometry,” in *Computer Vision*,Cambridge University

Press, 2000, pp. 138-183

[5] O. Faugeras, “Three-Dimensional Computer Vision: A Geometric Approach”, MIT Press,1996, pp. 33-68

[6] R. Y. Tsai,“A Versatile Camera Calibration Technique for 3D Machine Vision”, IEEE J. Robotics

& Automation, RA-3, No. 4, August 1987, pp. 323-344