CS6320: 3D Computer Vision
Final Project
Multiple Baseline Stereo

Arthur Coste: *coste.arthur@gmail.com*

April 2013

Contents

1 Introduction

In this final project, we are going to discuss multiple baseline stereoscopy as a central topic but other subjects will also be discussed.

In the second project, we studied and discuss the disparity computation based on a set of two images. But, we noticed that there could be uncertainties and ambiguities in this computation. So we thought it could be a good idea to explore this aspect and look for other methods to improve this process.

We decided to study and explore the Multiple Baseline Stereo technique introduced and developed by Okutomi and Kanade in the early years 1990. So, it's an old technique which is widely used and can be extended to robotics applications.

This project is a great occasion to explore some connected techniques such as rectification and the link between mathematical functions, hardware capabilities and application.

In this project, the implementation is made using MATLAB, the functions we developed are included with this report and their implementation presented in this document.

The following functions are associated with this work :

- $multibaseline.m$: $multibaseline()$ Stand alone function
- $rectify3.m$: $rectify3()$ Stand alone function

Warning : The implementation we made is not fully optimized and takes up to a dozen minutes to compute 4 disparity map and the reconstructed surface. So it's a time consuming process.

Note : All the images and drawings were realized for this project so there is no Copyright infringement in this work.

2 Theoretical presentation

2.1 Stereoscopy

Stereoscopy is a technique for creating or enhancing the perception of depth. The basic stereoscopy model intends to mimic the human vision system by using a set of two images from which depth is estimated and computed using different kinds of methods. Here is an illustration of our horizontal camera set up :

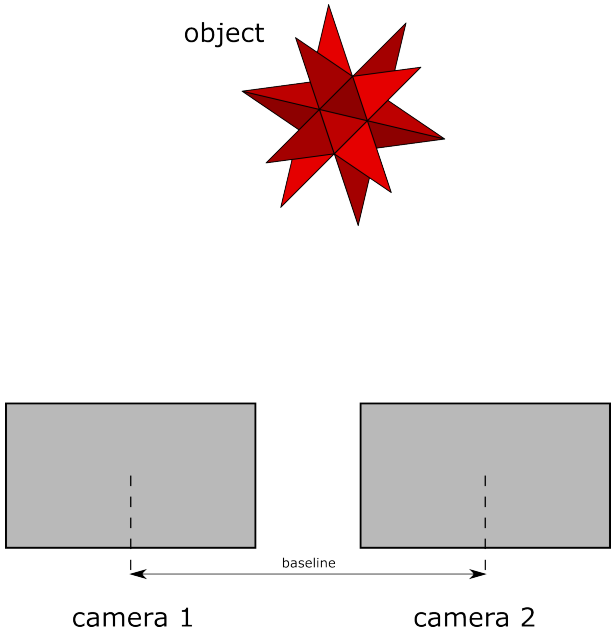

Figure 1: Horizontal stereographic acquisition

Then we can simplify this model to a geometric model with our known and unknown parameters:

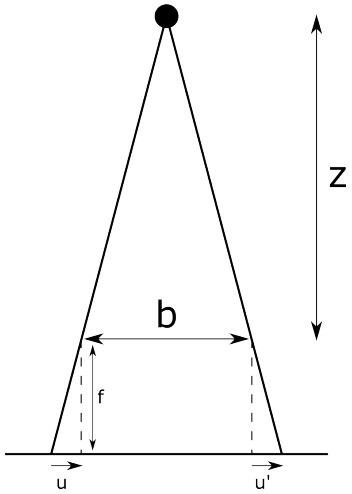

Figure 2: Geometric model

Based on two images and the fact that we know the baseline and the focal length, we are going to measure disparity on the two images and reconstruct the depth map.
With this set up, the disparity is given by:

$$d = |u' - u| \tag{1}$$

Then if we apply the relationships associated to similar triangles we can write :

$$\frac{B}{z} = \frac{d}{f} \tag{2}$$

And finally we obtain:

$$z = f\frac{B}{d} \qquad \text{or} \qquad d = Bf\frac{1}{z} \tag{3}$$

2.2 Similarity measurement

In this section, we are going to introduce and discuss the issue of similarity measurement. In fact, this is an essential point stereoscopy and depth reconstruction because it provides us with the disparity map which is the depth map up to scale. So, this is a very important step and a lot of possible mathematical measurement exists, some more complex, longer to compute, with different levels of complexity. We are going to present some of them with some results because in the Computer Vision literature a wide number of them are used and it appeared to be interesting to know a bit more about them.

Several families of similarity measurement are available : Classical method which include Distance measurement, Zero mean distances, Normalized distances or locally scaled distance. We will explore the previous kind on measurement because they are quite easy to implement. Then there is a derivative family based on gradients and Sobel or Kirsch operator or other kinds of transformation. Then

there is the Cross family based on Cross Correlation measurements. An other family is the non parametric family using Hamming distance, Jeffrey measure or other kind of measurements based on local transforms. And finally, there is a Robust family with other methods based on estimators or statistical measurements.

2.2.1 Set up and parameters

To perform this analysis and description of similarity measurement we are going to use the same data set and parameters for all the measurements to produce comparable images. We are going to use one of the Tsukuba University data set with a size 5 square window (kernel) and we are going to impose a boundary condition of maximum disparity set to 20 pixels. Here are the original two images and the ground truth disparity map for these images.

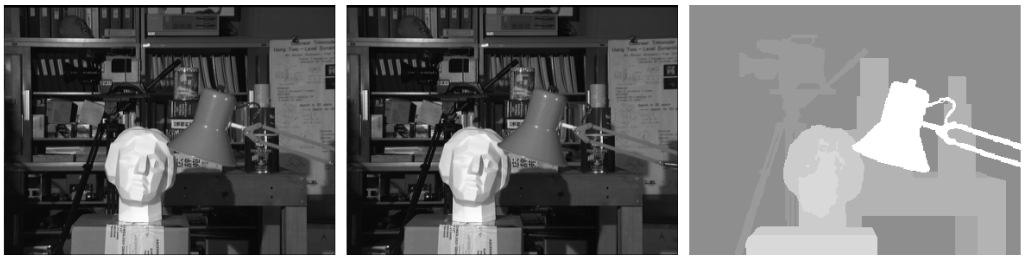

Figure 3: Left and right images of the data set and disparity ground truth

2.2.2 Sum of Absolute Differences

The Sum of Absolute Differences (SAD) is defined by :

$$SAD(i,j) = \sum_{(i,j)\in W} |I_1(i,j) - I_2(x+i, y+j)| \tag{4}$$

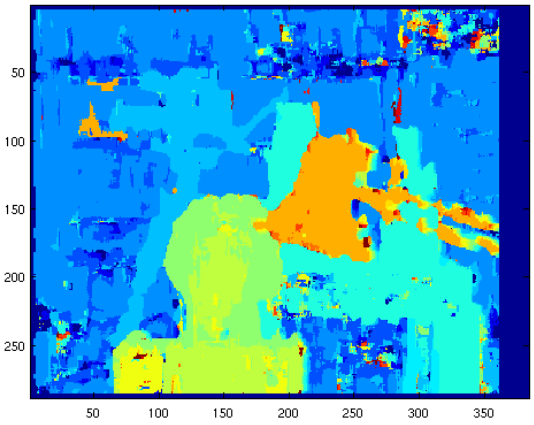

Figure 4: Disparity map computed with SAD disparity measurement and a size 5 kernel

The Zero Mean Sum of Absolute Differences (ZMSAD) is defined by :

$$ZMSAD(i,j) = \sum_{(i,j)\in W} |I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i, y+j) + \bar{I}_2(x+i, y+j)| \tag{5}$$

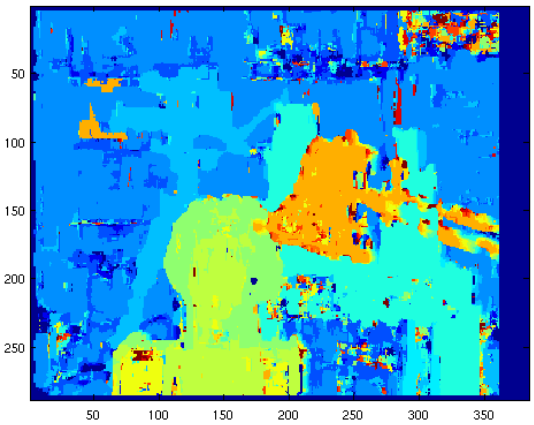

Figure 5: Disparity map computed with ZMSAD disparity measurement and a size 5 kernel

2.2.3 Sum of Square Differences

The Sum of Square Differences (SSD) is defined by :

$$SSD(i,j) = \sum_{(i,j)\in W} (I_1(i,j) - I_2(x+i, y+j))^2 \tag{6}$$

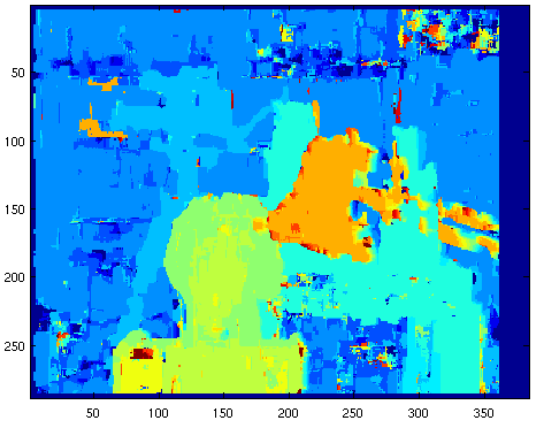

Figure 6: Disparity map computed with SSD disparity measurement and a size 5 kernel

The Zero Mean Sum of Square Differences (ZMSSD) is defined by :

$$ZMSSD(i,j) = \sum_{(i,j)\in W} (I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i, y+j) + \bar{I}_2(x+i, y+j))^2 \quad (7)$$

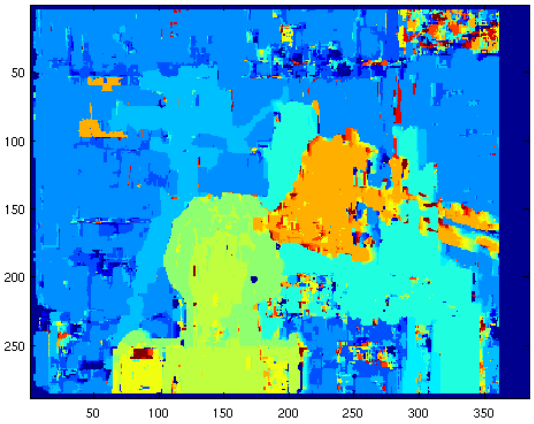

Figure 7: Disparity map computed with ZMSSD disparity measurement and a size 5 kernel

2.2.4 Normalized Cross Correlation

The Normalized Cross Correlation (NCC) is defined by :

$$NCC(i,j) = \frac{\sum_{(i,j)\in W} I_1(i,j) I_2(x+i, y+j)}{\sqrt{\sum_{(i,j)\in W} I_1(i,j)^2 \sum_{(i,j)\in W} I_2(x+i, y+j)^2}} \tag{8}$$

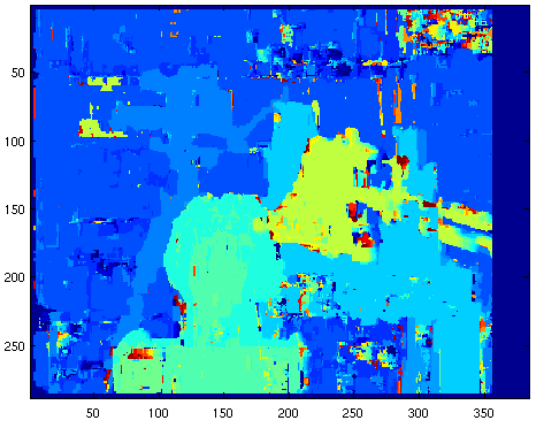

Figure 8: Disparity map computed with NCC disparity measurement and a size 5 kernel

The Zero Mean Normalized Cross Correlation (ZMNCC) defined by :

$$ZMNCC(i,j) = \frac{\sum_{(i,j)\in W} (I_1(i,j) - \bar{I}_1(i,j))(I_2(x+i, y+j) - \bar{I}_2(x+i, y+j))}{\sqrt{\sum_{(i,j)\in W} (I_1(i,j) - \bar{I}_1(i,j))^2 \sum_{(i,j)\in W} (I_2(x+i, y+j) - \bar{I}_2(x+i, y+j))^2}} \tag{9}$$

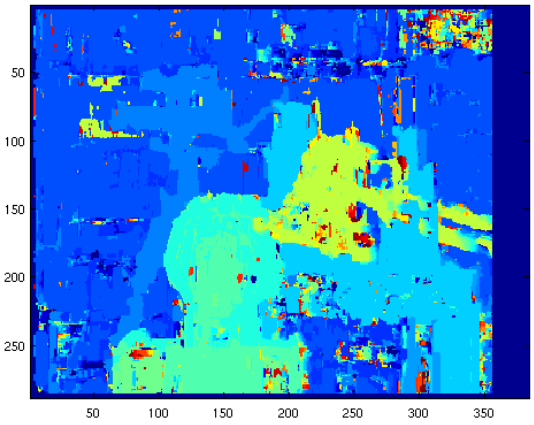

Figure 9: Disparity map computed with ZMNCC disparity measurement and a size 5 kernel

The Robust Zero Mean Normalized Cross Correlation ($ZMNCC_R$) defined by :

$$ZMNCC_R(i,j) = \frac{\sum_{(i,j)\in W}(I_1(i,j) - median(I_1(i,j)))(I_2(x+i,y+j) - median(I_2(x+i,y+j)))}{||I_1(i,j) - median(I_1(i,j))||_1||I_2(x+i,y+j) - median(I_2(x+i,y+j))||_1} \tag{10}$$

2.2.5 Sum of Hamming Distances

The use of Hamming Distance (HD) is an interesting concept which comes from information theory and which is widely used in computer science and signal processing. Hamming distance plays a great role in error correction codes to quantify the difference between strings of symbols. It's mathematically defined by:

$$d(a,b) = \sum_{i=0}^{n-1}(a_i \oplus b_i) \tag{11}$$

Where a and b are symbols and $\oplus$ is defining the bitwise logical exclusive or operator. In our case it leads to the following similarity measurement SHD which belongs to the family of non parametric similarity measurements.

$$SHD(i,j) = \sum_{i,j\in W} I_1(i,j) \oplus I_2(x+i,y+j) \tag{12}$$

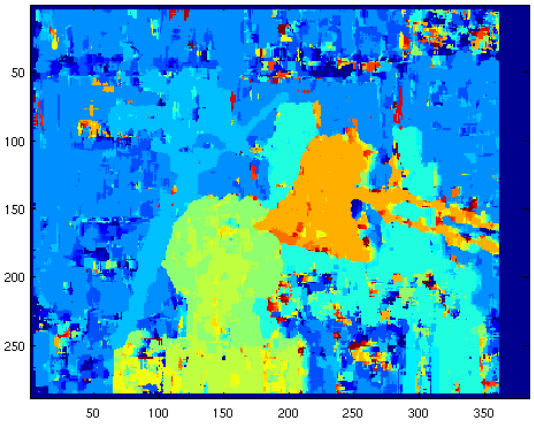

Figure 10: Disparity map computed with SHD disparity measurement and a size 5 kernel

2.2.6 Discussion on similarity measurement

The point of this section was to review briefly some techniques to compute disparity and discuss sources of improvement. Indeed, in class we discussed the Zero Mean Normalized Cross Correlation a lot and used it as exclusive technique in our projects. But, when we looked at the literature of Computer Vision and especially Okutomi and Kanade paper where they used the Sum of Squared Differences we decided that exploring this aspect could be interesting. So, we looked at the "state of the art" and found a nice summary paper of similarity measurement from Chambon and Crouzil. We implemented some of them, we presented the results before and we are now going to discuss them and conclude about this section about similarity measurements.

A quick note regarding a difference between Cross Correlation related measurements and difference based measurements. Indeed, on the first case we want to maximize the cross correlation while on the other hand we want to minimize the differences between patches. So, these two approaches are opposed on an optimization point of view.

Thanks to the various similarity measurement we presented before, we can see that the overall result is quite consistent from one method to the other. It would appear on the Zero Mean method could introduce a more noisy result than the other methods. This section was just a quick presentation of easy other possible similarity measurement. It definitely makes some sense when studying depth measurement and disparity computation to discuss the other available methods and explore possible ways of improvement.

Another important criterion is the choice of the disparity scanning length. Indeed, for rectified images, we need to constrain the search for a correspondences to a certain length to avoid spending to much computational resources and also to reduce as much as possible the ambiguity issue which will be presented later.

To conclude, we can summarize that to chose a good similarity measurement we have to take in account many factors. Firstly, we need to consider the application and if it requires good quality results versus real time performances. Then, we need to identify the hardware capabilities versus the computational time. So, once we know how to design the trade off on these aspects we can pick up the most suitable similarity metric. In real time applications, fast computational metrics should be preferred such as Absolute Difference, but the drawback is the sensitivity to noise and low performances with matching discrimination or adaptation to camera small defaults such as blurring effects. By introducing Zero Mean (ZM) regularisation we introduce a bit more complexity but increase a bit the discrimination. The locally averaging computation allow to reduce noise influence with smoothing. Those two techniques require more computational resources and capacities. A more robust metric to intensity issues and discrimination is the use of Normalized Cross Correlation and optimally the Zero Mean Normalized Cross Correlation, but it requires more computational time and resources. On small embedded systems, the use of Hamming Distance based on a bitwise XOR operation is one of the fastest available metric, but it's one of the least robust but depending on the application it could be enough.
So, there is a great trade off in this aspect regarding application and available hardware.

2.3 Epipolar Geometry and Rectification

With two or more image acquisitions we have to deal with a special kind of geometry : the epipolar geometry. As we know, with an image, we are lying in the perspective projection model due to the loss of the depth dimension. The whole goal of this project is to reconstruct this 3rd lost dimension. So it appears to be important to discuss some concepts related to epipolar geometry. As mentioned before, the stereoscopic system is trying to mimic the human perception system using two cameras with a converging angle. So we need two cameras to acquire two slightly different images on which we can find correspondences and estimate some physical properties.
The following picture illustrates the set up of the system.

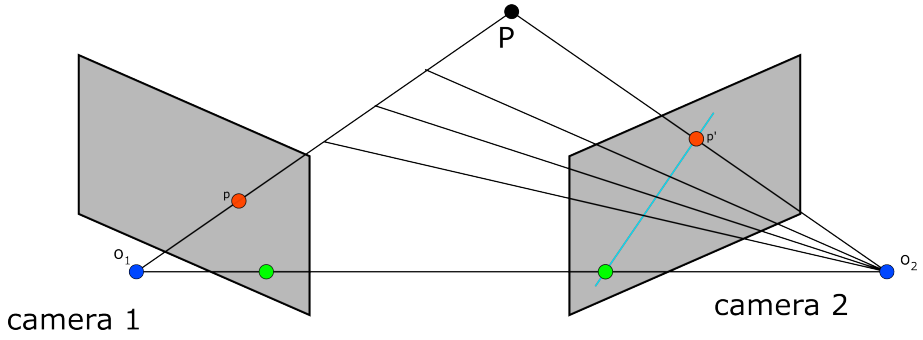

Figure 11: Epipolar geometry in the case of two camera

According to the epipolar constraint we should have all the vectors : $\overrightarrow{O_1p}, \overrightarrow{O_2p'}$ and $\overrightarrow{O_1O_2}$ coplanar.
This leads to formulate the epipolar constraint with this equation:

$$\overrightarrow{O_1p}.[\overrightarrow{O_1O_2} \times \overrightarrow{O_2p}] = 0 \tag{13}$$

We can simplify it, if we introduce the coordinate independent framework associated to the first camera.

$$p.[t \times (\mathcal{R}p')] = 0 \tag{14}$$

We can finally introduce the essential matrix ϵ defined by :

$$\epsilon = [t_x]\mathcal{R} \qquad \Rightarrow \qquad p^T \epsilon p' = 0 \tag{15}$$

With t_x being the skew symetric matrix resulting in the cross product $t \times x$:

$$[t_x] = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = [t_x]X = T \times X \tag{16}$$

As we discussed in the previous project, an important problem still needs to be considered which is the internal and external camera calibration. This can be integrated in our framework using the two following equations : $p = \mathcal{K}\hat{p}$ and $p' = \mathcal{K}'\hat{p}'$ with $\mathcal{K}$ and $\mathcal{K}'$ are the calibration matrices associated

to each camera.
According to Longuet-Higgins relation we can now use them in the previous equation providing us with:

$$p^T \mathcal{F} p' = 0 \tag{17}$$

We will rely on this equation in the implementation to compute epipolar lines and determine epipoles locations.

Two interesting points are associated with this special geometry : epipoles. An epipole is the projection on the other image of the center of the other camera. The left and right epipoles are determined by the left and right null spaces of the Fundamental matrix $\mathcal{F}$. Once we know the location of these points, we can compute the epipolar lines which allow us to define the match from one point in one image to an associated epipolar line in the other image.

In this project, our input data needs to be images acquired with an exact horizontal translation to perform horizontal matching between images. So thanks to what we have said before it implies that the epipolar lines of each image are parallel lines and the epipoles are rejected to infinity.

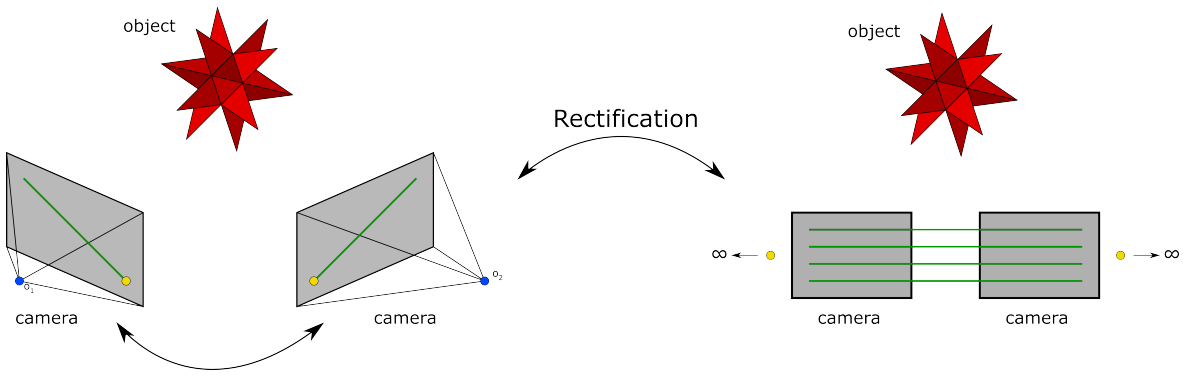

Figure 12: Theoretical presentation of rectification technique

As we can see on the previous picture, our goal is to geometrically transform images acquired with "random" displacement between each other to align their epipolar lines. By doing so, we will be able to simplify and optimize the search for match in disparity computation by only scanning a single horizontal line. As we presented before, the Fundamental matrix is giving the transport from a feature in the first image into the second image. In the special case of rectified images, the fundamental matrix can be reduced to a simpler form:

$$\mathcal{F} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{18}$$

It exists several technique to perform rectification according to the information we have on the input images and kind of results we want to obtain and achieve.

Here is a quick presentation of an easy algorithm from Richard Hartley:

Given the Fundamental matrix $\mathcal{F}$ we need to find two rectification matrices $\mathcal{R}$ and $\mathcal{R}' \in L(3)^2$ such as :

$$\mathcal{F} \cong \mathcal{R}'^T \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \mathcal{R} \tag{19}$$

So we need to find a Rectification matrix for each image of the stereo pair : $(\mathcal{R}_0, \mathcal{R}'_0)$ and $(\mathcal{R}_1, \mathcal{R}'_1)$. So we have :

$$\mathcal{F} \cong \mathcal{R}'^T_0 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \mathcal{R}_0 \cong \mathcal{R}'^T_1 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \mathcal{R}_1 \tag{20}$$

We know that rectification matrices are full rank matrices so we have :

$$\left(\mathcal{R}'_1 \mathcal{R}'^{-1}_0\right)^T \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \mathcal{R}_1 \mathcal{R}_0^{-1} \cong \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{21}$$

$$\mathcal{F}' \cong \mathcal{M}'^T \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \mathcal{M} \cong \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{22}$$

The goal of this method is then to estimate and find the best compatible set of matrices $\mathcal{R}$ and $\mathcal{R}'$ coming from $\mathcal{R}_0$ and $\mathcal{R}'_0$ and compatible with $\mathcal{F}$.

To perform this operation we did not implement this piece of code ourselves but relied on a tutorial code furnished here [4].
In this algorithm we need to set as input the Fundamental matrix between the left and the right image and the set of associated landmarks. We relied on a previous project implementation to perform the computation of the Fundamental matrix based on manually selected landmarks.
Here are our input images :

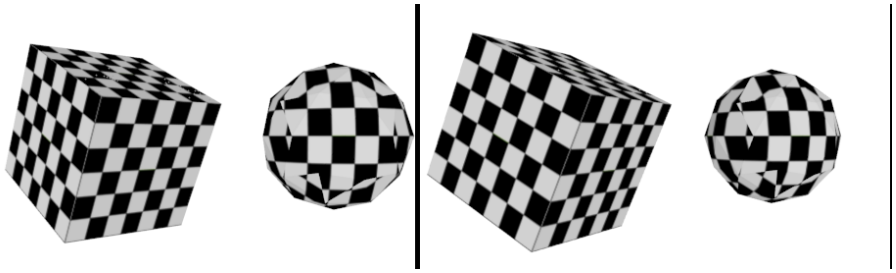

Figure 13: Input left and right images for rectification

Due to laziness in our manual selection of points, which is a time consuming operation, we only picked a dozen points in the cube, so the result is that the Fundamental matrix is computed with

this limited set of points. This leads to issues with the accuracy of this matrix because we only deal with a small set of landmarks and moreover they are all located in the same region. So we provided to the rectification algorithm an inaccurate matrix so the result of rectification is poorly constrained to few points which is presented in the following picture:

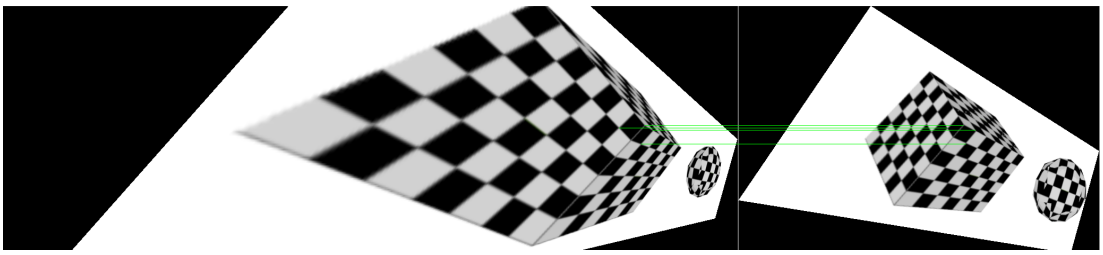

Figure 14: Rectified images with limited landmarks

So to solve this issue, as presented in the work we used for this part, the author uses a SIFT technique to have a large set of landmarks over the whole image which allow a better computation of the Fundamental matrix and then a better rectification of images.

We decided to use a test data set which is already rectified to avoid dealing with this issue. But with a larger number of landmarks we could obtain a more robust computation of the fundamental matrix and also of the two rectification matrices.

This section was presented because rectification is an important part of the pipeline to provide images where we can use similarity functions easily without introducing computational issues.

2.4 Multiple Baseline Stereo Theory

In this set up, we are going to use multiple images acquired with different baseline with respect to the first picture which will serve as reference. So we will have the following disparity definition:

$$d_i = fB_i\frac{1}{z} \tag{23}$$

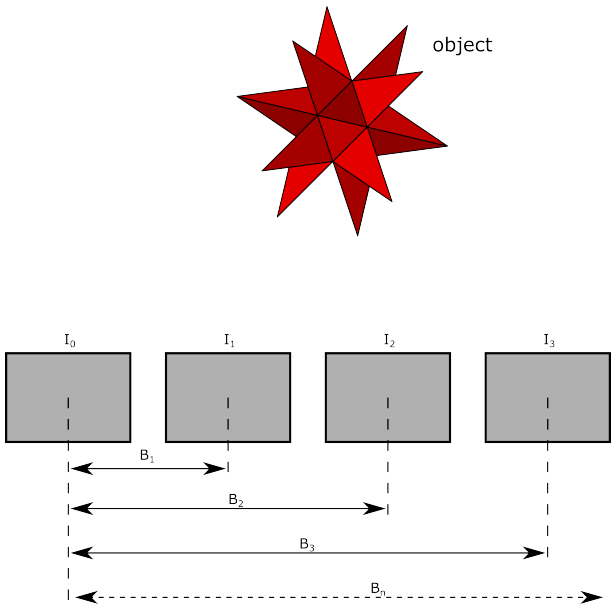

Figure 15: Setup of the multiple baseline system

As we have discussed before, to reconstruct depth, we will rely on the computation of disparity. As presented before, it exists a lot of different methods to compute it. In this project, in accordance with Okutomi and Kanade we are going to use the Sum of Square Distances (SSD). We need to introduce the inverse distance variable defined by:

$$\zeta = \frac{1}{z} \tag{24}$$

which will leads to the following modification of the disparity function :

$$d_i = fB_i\zeta \tag{25}$$

If we compute the similarity measurements based on the ζ variable we are going to show on various data set that this could lead to some improvement of our results. Indeed, in the previous results we showed in a previous project or in the previous section, there could be ambiguities introduced by the similarity metric. Indeed, the implementation consists of a sliding kernel on a line of two rectified images to find the optimal location providing the optimal value for the associated metric.

In fact, small baseline based disparity measurements have some issues due to triangulation. Moreover if we only use a pair of two rectified images we obtain a single similarity function. So finding the optimal value for the function is highly sensitive to ambiguities also depending on the implementation. Indeed, due to noise corruption of the images, we can identify an absolute minimum for the similarity function which is not point we are looking for due to various sources of noise.

So, the objective of Multiple Baseline Stereo is to create a set of similarity functions and find a common point between them to reduce the influence of noise. As mentioned in the equations above, the value of the baseline is taken in account to regularize the values and provide a baseline independent similarity function. In the following section we are going to present a synthetic data set where we will have a very strong ambiguity issue to solve using this framework.

3 Practical set up

3.1 Synthetic data set

In this section we are going to introduce the key objective of Multiple Baseline Stereo. Indeed, when we rely on similarity measurement we can often encounter a large number of ambiguities. Ambiguities occur when scanning a repeated pattern or texture of the image and computing a similarity function.

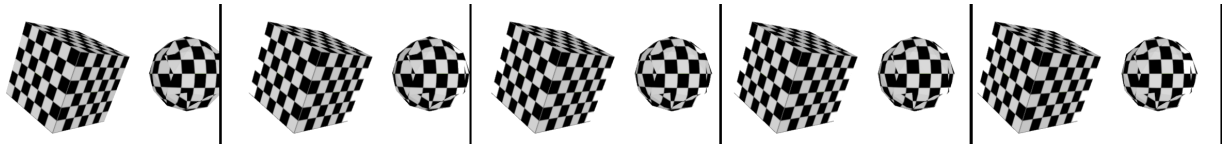

Figure 16: Images of the test data set

If we consider the previous images with repeated pattern on objects which will lead to the ambiguity issue.

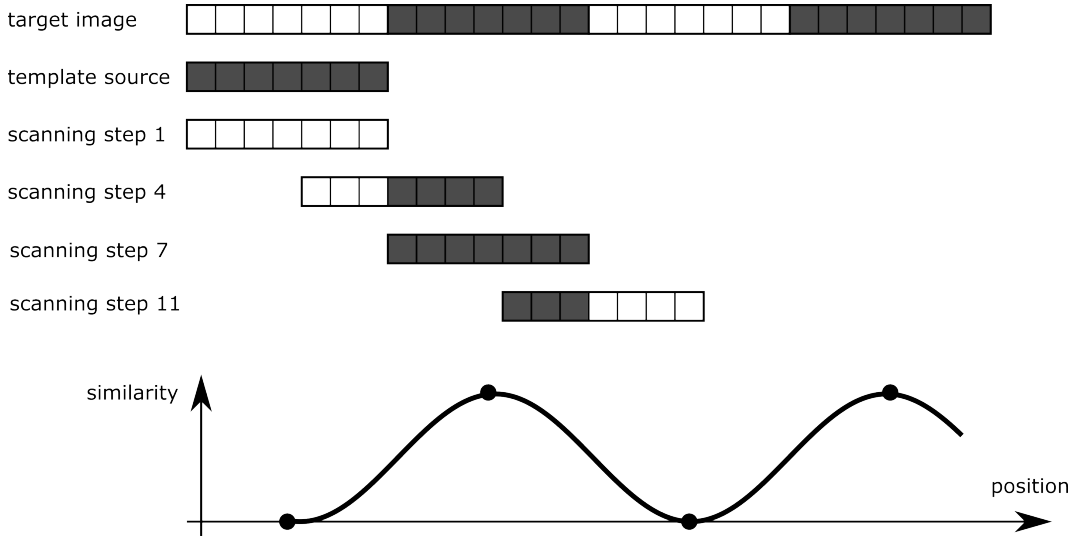

Figure 17: Ambiguity for a repeated pattern

So, in the previous illustration, our template coming from the source image which is moved on horizontal epipolar lines to find the best match. At each step we compute the similarity measurement between the template and the image. And this leads to a periodic similarity function, so there is a set of best match instead of a single one.

According to the kind of metric we use, this will constrain the search for best maximal values or best

minimal values. For instance, the Normalized Cross Correlation is looking for a best match with a maximal value of 1, while if we use a difference based metric we look for a minimal value close to 0.

Consequently, if we only have two images, we end with a sequence like the previous one, and we can't decide which position gives the best match. Here is an example of a kind of result we could have if we only use a standard stereo pair of images:

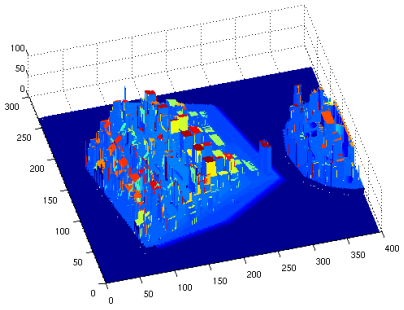

Figure 18: Reconstructed surface using single and small baseline

As, we can see on the previous result, the reconstructed surface based on a single stereo pair is really bad due to the unresolved ambiguity issue. Our implementation detect a good match but the issue is that it's not the only one. So, we are here unable to reconstruct the true depth because one minimal value is picked by the algorithm with no verification on consistency with neighbours because the computer does not have a prior knowledge of the shape. This allow us to discuss the influence of the baseline length. Indeed, the size of the baseline. Indeed, a short baseline could create a large uncertainty on distance estimation due to a narrow triangulation. This is why larger baseline produced a better result, but with as consequence a larger scanning range.

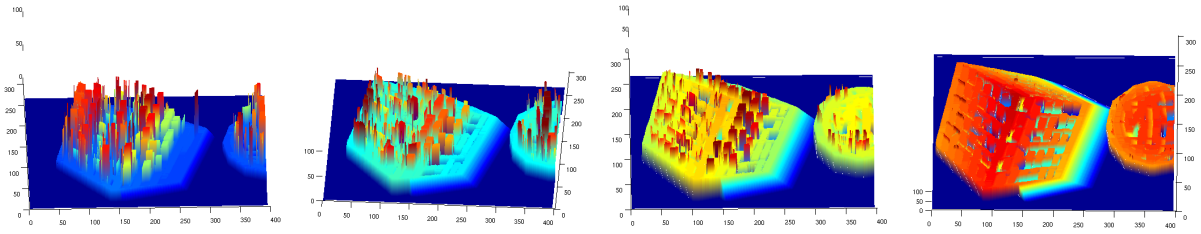

Figure 19: Reconstructed surface using increasing baseline length

So, as mentioned before we can see that increasing baseline enhance the result of the reconstructed surface so let's apply Okutomi and Kanade method to try to solve this issue. So if we look at the Sum of Square differences for a given pixel of the cube we obtain the following result.

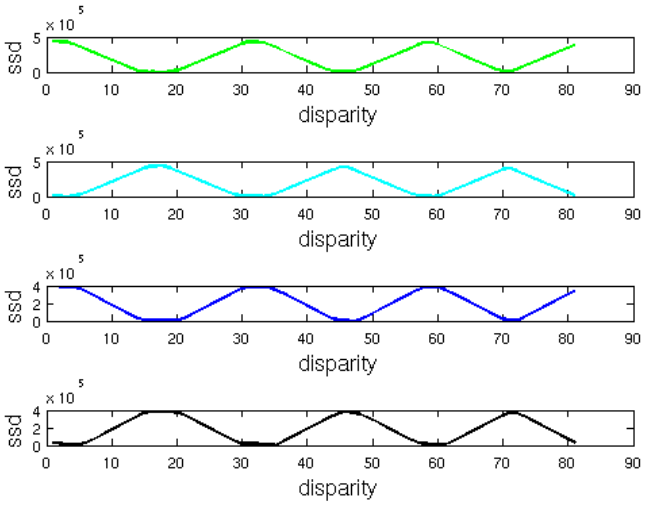

Figure 20: SSD functions of disparity for each pair of stereo images

We can here clearly see that we have an ambiguity phenomenon on each of the plotted SSD function of the disparity. So we can see that in this case finding a minimum value is ambiguous and impossible to implement without prior knowledge of the scene.

To solve this issue we are plotting each of the previous SSD functions according to the inverse depth : ζ and we also add on this plot a new function defined by Okutomi and Kanade : the Sum of n Sum of Square Differences functions defined by:

$$SSSD(\zeta) = \sum_{i=1}^{n} \sum_{j \in W} (I1(i,j) - I_2(i, y+j))^2 \tag{26}$$

Which allow us to obtain the following plot in inverse depth:

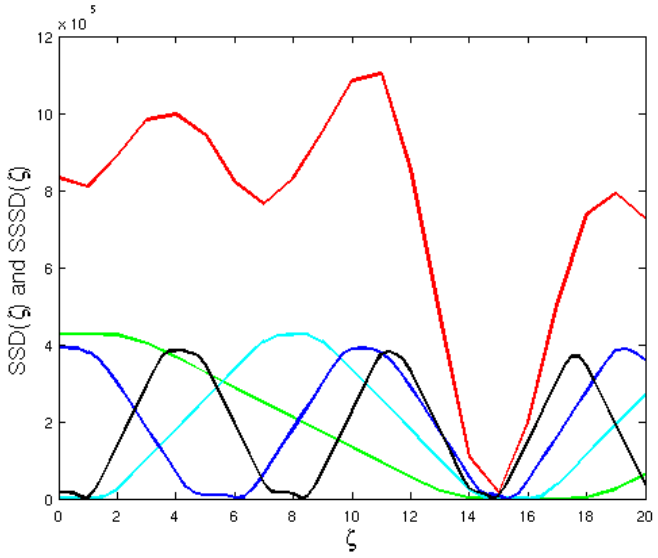

Figure 21: $SSD(\zeta)$ and $SSSD(\zeta)$ associated to the previous SSD functions of disparity

We can now clearly see on the previous result the existence of an "invariant" point in terms of inverse depth where all the disparity curves seems to agree. But, as we can see curves are not exactly matching each other at the same point. There is an uncertainty distribution of the minimal position due to the structure of images and the possible noise in the image and computation operations. So, to obtain an estimate of this position, we rely on the Sum of Sum of Square Differences functions which will add for each inverse depth point the sum of the similarity measurement. This create a new similarity function of the inverse depth variable which has a minimal value located where all the previous SSD function seems to match. Indeed, this new function also have its own associated uncertainty but we can clearly see on the previous figure that the result is fairly well located.

So, the whole point of Okutomi and Kanade's method is relying on this gathering of multiple baseline information into a derived similarity measurement which will allow us to locate a matching point. Once we have located this point we have the inverse depth map and we can reconstruct the surface associated.

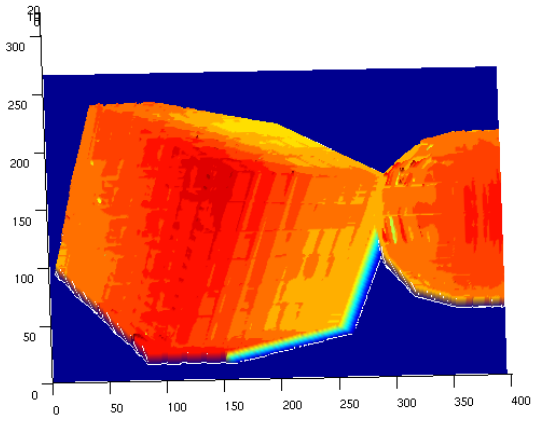

Figure 22: Reconstructed surface based on min of $SSSD(\zeta)$ function

The previous result is quite neat, it reconstructs pretty well the correct shape and orientation of objects. We recognise the cube and its orientation, we understand that the second object is has a spherical shape and finally we can even feel the texture pattern.

Compared to the reconstructed surfaces we presented before with a single stereo pair this result really appear better in terms of real shape. We mentioned earlier that using a larger baseline was improving the quality of the reconstructed shape by reducing the triangulation uncertainty. And we could clearly see this aspect earlier, so let's compare the biggest single baseline scene reconstruction and the multiple baseline reconstructed surface.

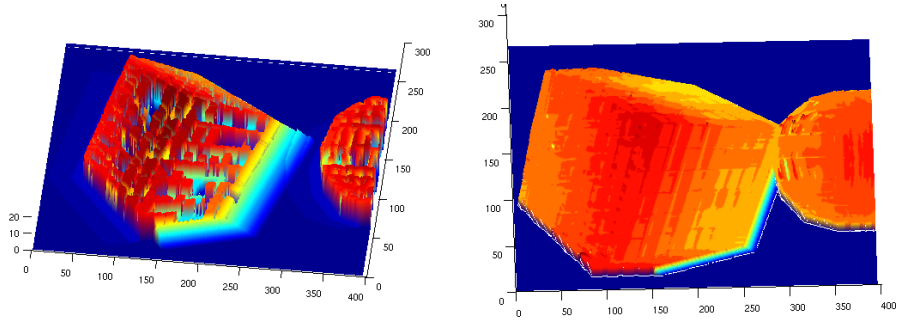

Figure 23: Comparison between largest single baseline and mutliple baseline reconstruction

The first thing that we can notice is that, on the first picture coming from the largest single baseline stereo, the reconstructed scene is neater than in the second one. Indeed, the last comment is only made on shape of objects, where the combination of multiple information from multiple sources, tend to make the object appear bigger than in reality. In the first image, we can clearly see that the drawback of it, is that we can clearly see the unresolved ambiguity issue which leads

to a large number of holes in the shape. So with our second reconstruction, we manage to resolve almost all ambiguities (some small glitches still remain) and the reconstructed shape make a lot of sense because, in my opinion, the feeling of depth seems more accurate than on the single baseline reconstruction.

Thanks to this data set, we presented and introduced the ambiguity issue that Okutomi and Kanade are solving with their multiple baseline technique. We showed and explained the origin of ambiguities and how gathering information from multiple images could lead to an invariant point in a specific framework which will allow us to resolve most of these ambiguities. Here is another example of SSD and SSSD curves of inverse depth that we can obtain for an other pixel.

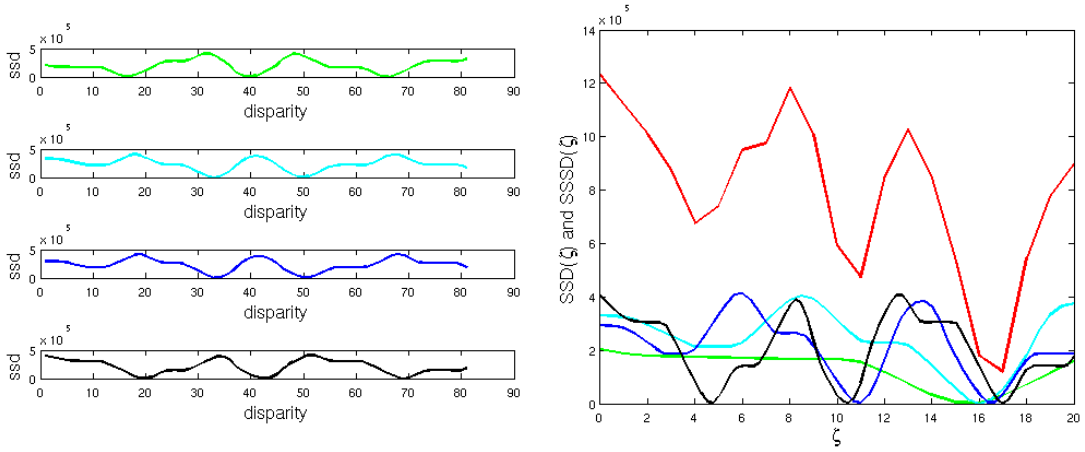

Figure 24: SSD of disparity and $SSD(\zeta)$ and $SSSD(\zeta)$

Finally we plotted the reconstructed surface, the result looks better compared to what we had with the single baseline method to estimate depth. According to our depth map we showed before, we could clearly see that most ambiguities have been solved but with this reconstruction we can see the shapes.

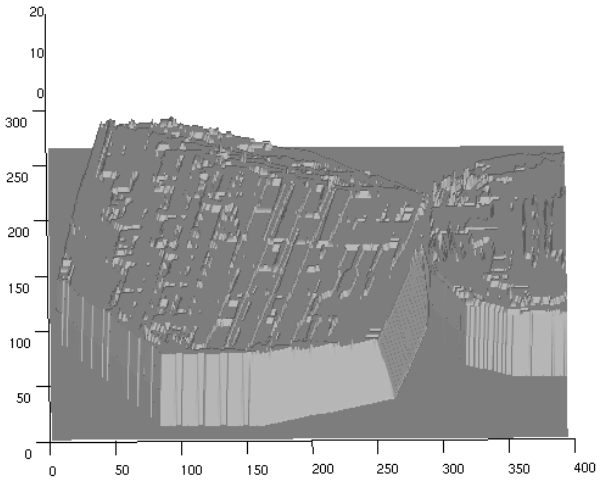

Figure 25: Simple surface reconstruction using Matlab

As we can see on the previous result, we can see clear sharp edges where there is no information available on any camera for reconstruction, but we can also see some glitches and holes which still remain on the shape. But as mentioned before this method is significantly improving depth and shape reconstruction.

3.2 Test data set

The previous data set was made of synthetic images generated with a computer in order to introduce and present Okutomi and Kanade's method. Here, we are going to use a test data set made of real images from the University of Tsukuba, Japan. This is the same dataset we used to present and discuss the different similarity measurement results in the first section. This data set is composed of 5 images acquired with a constant horizontal translation from one to the other. It implies that for this data set that no rectification needs to be performed because epipolar lines are a set of parallel lines. Here are the five images :

Figure 26: Images of the test data set

We decided to consider as reference image the first image of the data set which is the first on the left and then we computed disparity measurements with this image as reference, leading to 4

pairs of images with linear baseline relationship.

Indeed we have $B_1 < B_2 = 2 \times B_1 < B_3 = 3 \times B_1 < B4 = 4 \times B_1$.

Knowing the previous evolution of the baseline between images, we are then able to compute accordingly the disparity maps by defining correctly the scanning length on epipolar lines.

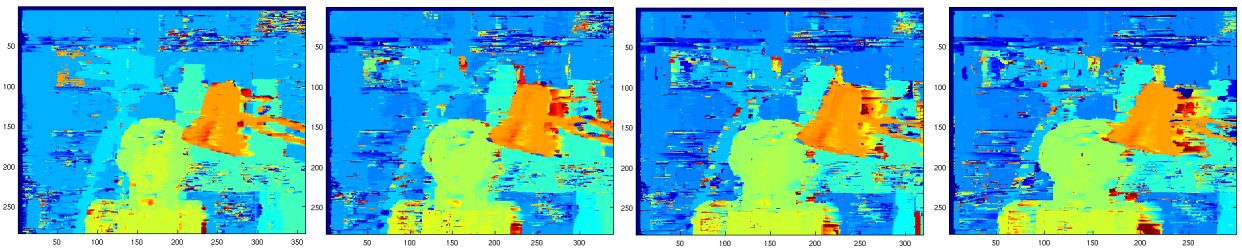

Figure 27: Computed disparity images with different scanning length

Indeed, we used the two previous images in a previous project and also in the first section so we know pretty well the range of disparity within this image. So between the reference image and the first one, we used a 20 pixels length scan for matching pixels, and then we used the previous baseline relationship to generate the previous disparity images. Indeed, the more we translate the camera to acquire images, the longer has to be the scanning lines to find correspondences. So the set of scanning length is: $scan(i) = (20, 40, 60, 80)$.
The use of the same color map is enhancing this linear relationship between baseline that creates a linear relationship in disparity computation. But, by doing so we introduce a prior knowledge on the evolution of disparity in images.

For this data set, we applied the same method as the one presented before so here is a presentation and a discussion of the results. The four disparity maps presented before where also obtained using the Sum of Square Differences so here are the associated curves for some interesting pixels.

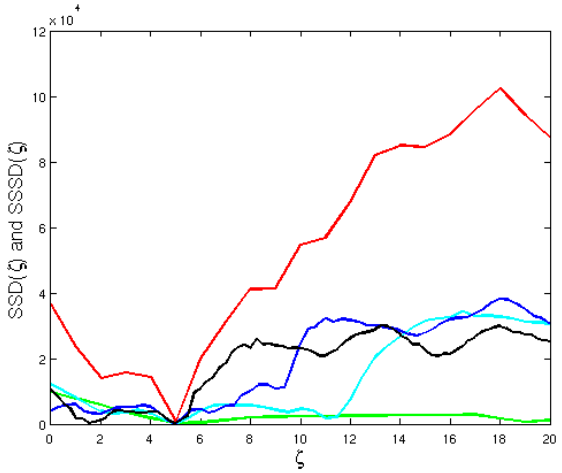

Figure 28: $SSD(\zeta)$ and $SSSD(\zeta)$ for a good match pixel

In the previous result we can firstly notice that due to the nature of the input pictures, our similarity measurement functions are no longer smooth and nice periodical functions but they are more likely to be sensible to noise or intensity variations. Then, we can notice that the 4 SSD functions match approximately at the same location which is a clearly good match with the position of a valley in the SSSD function. So in the previous case we can clearly state that our method is working fine to determine the accurate inverse depth. We can also show another quite good result for a different pixel location where SSD functions have a more unpredictable behaviour:

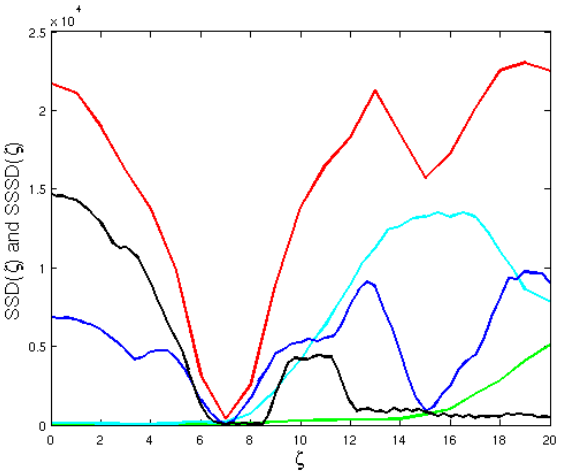

Figure 29: $SSD(\zeta)$ and $SSSD(\zeta)$ for another nice match pixel

Unfortunately, this good match scenario is not always true, and we also have very bad matches and unresolved ambiguities. Here is an example:

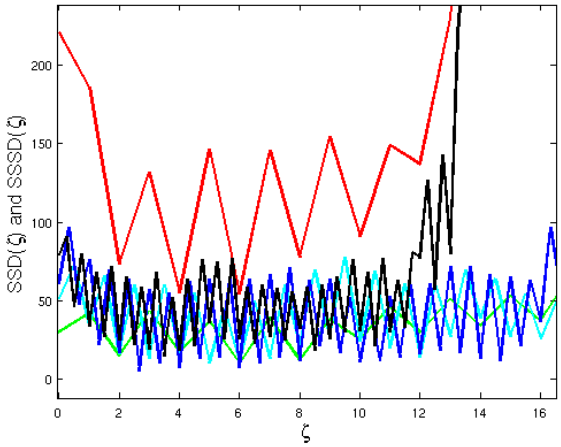

Figure 30: Issue with oscillations which could lead to wrong estimation

As we can see on the previous result, all similarity functions oscillates a lot, they have a high frequency which traduces an incapacity to match exactly the pattern on the original image. This behaviour is then carried in the derived SSSD function where some issue could occur. Indeed the SSSD function also oscillate, so there is no clear and absolute minimal values, with potentially several optimal points that does not resolve the ambiguity issue. So the previous result is a good illustration of unresolved ambiguity, with the behaviour of all functions involved.
Then, here is the final reconstruction that we can obtain with this method.

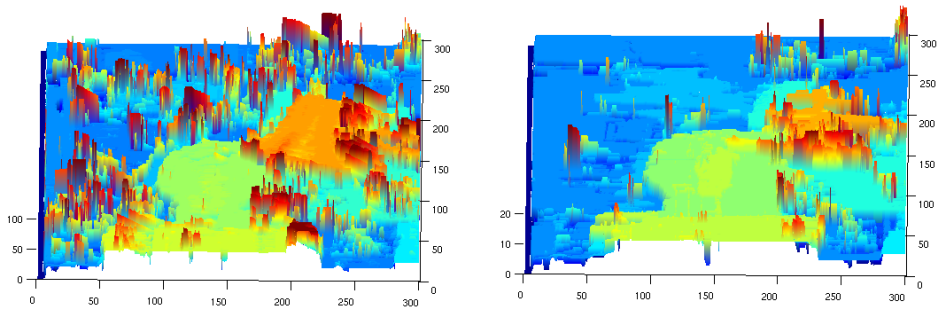

Figure 31: Comparison between longest baseline and multiple baseline reconstruction

There are still a lot of glitches but compared to other reconstruction using single baseline and versus ground truth it seems fairly better.

4 Results and discussion

4.1 Results analysis

If we analyse the results we presented so far, they seems to be quite good. In fact, we succeeded to resolve almost all ambiguities of the first data set. The reconstructed shapes we obtained are satisfying because shapes are correct and we could even say that we can even guess the texture pattern we applied to the objects. A small point can be considered bad is that in our reconstruction, we can see that the two shapes almost touch each other while it's not the case in the reality. This might come from the metric functions which still carry informations on the boundaries of objects and lead to a small increase of shapes size around boundaries. It's the kind of same effect obtained while smoothing an edge.

On the second data set, composed of real images, we also had to face the ambiguity problem. In the result we obtained, some ambiguities remain unsolved. This could be due to noise issues that could interfere because the similarity function we used is very noise sensitive. So an interesting test to do could be to change the SSD similarity function we used with a Zero Mean Normalized Cross Correlation (ZMNCC) which is independent to linear variation of intensities.

4.2 Framework and implementation discussion

A great subject of discussion is dealing with the application and the special link between hardware, maths and constraints. Indeed, as we presented in the theoretical question, mathematically we can define a wide range of metric functions with different complexity, which required more or less complex data structures and implementation or computational resources.

In this implementation, we were only motivated by discovering and explaining the principles and methods, we were not dependant of limited amount of time and could afford waiting several minutes to get a result. But on most applications, we deal with robotics and real time or close to real time applications so we can definitely not afford to spend several minutes to get a nice and precise result.

The use of SSD requires more computation time, so in real time applications, often based on video footage, the Sum of Absolute Differences (SAD) is often use to optimize computation time because it requires less mathematical operations than Sum of Square Differences and Normalized Cross Correlation.

Some other solutions such as down sampling or pyramidal strategies can be combined to optimize computation according to available time and resources.

4.3 Further improvements

By working on the Optical Flow project, we came up with the idea that using the displacement vector provided by the optical flow could be in this project a great source of information. Indeed, our disparity measurement is computed based on a pixel displacement between rectified images, and this is basically what optical flow is also determining. So the idea was to bring an optical flow

measurement to estimate the approximate location of the matching point and by doing so scaling the searching range to only focus on a specific area and have improved similarity measurements.

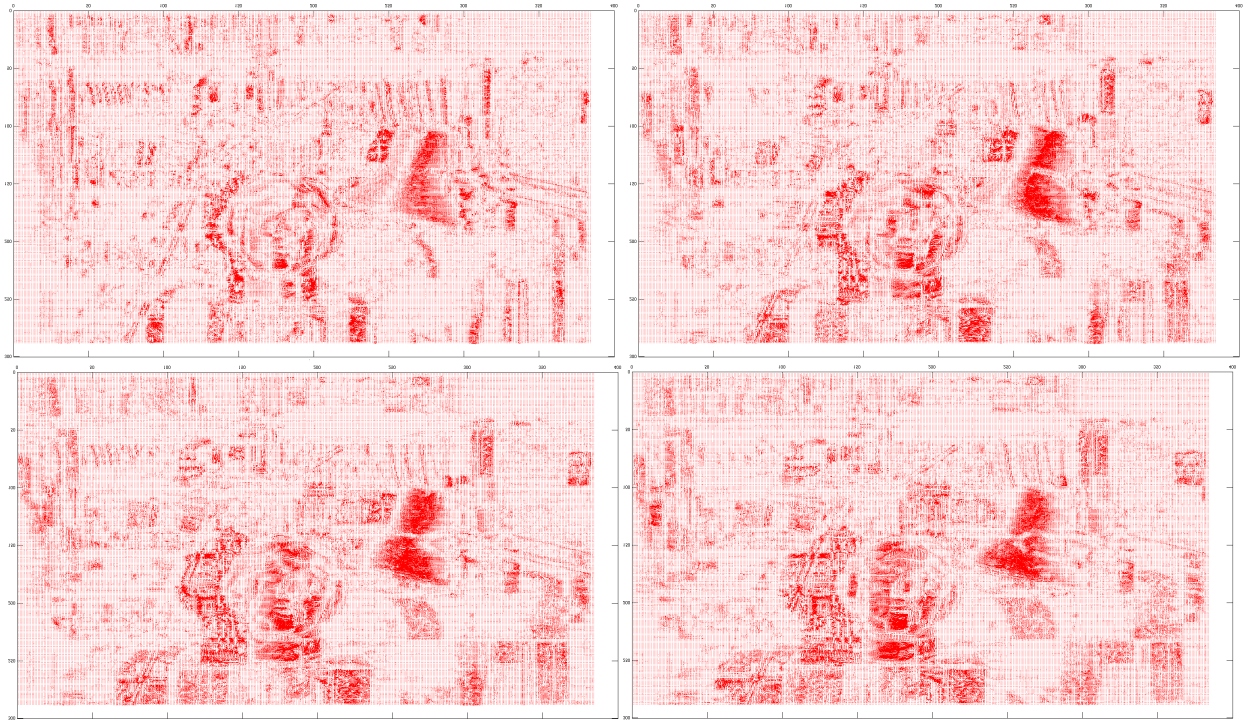

Figure 32: Evolution of optical flow with increasing baseline (top B1,B2, bottom B3,B4)

In the previous set of results we can clearly see that optical flow is increasing with baseline length and that the closer to the camera the more important is the displacement field. These two observations clearly make sense and could be used as a prior knowledge for searching for pixel match.

5 Implementation

5.1 Similarity measurements

The general principle of this section is the same, then it's just the mathematical equation that we use to compare vectors which change. This can be extended to any similarity measurement we want to use.

```
weight = ones(5);
max_line_scan = 20;
for i = 1+ceil(size(weight,1)/2):1: size(I4,1)-ceil(size(weight,1)/2)
        for j = 1+ceil(size(weight,1)/2):1: size(I4,2)-ceil(size(weight,1)/2)-max_line_scan
            previous_shd_val = 64768483;
             estimated_disparity = 0;
             for(position=0:1:max_line_scan)
                  t=1;
                  shd(i,j,position+1)=0;
                 %compute convolution for the neighbourhood associated to
                 %the kernel
                  for(a=-ceil(size(weight,1)/2):1:ceil(size(weight,1)/2))
                     for(b=-ceil(size(weight,1)/2):1:ceil(size(weight,1)/2))
                           w(t) = I4(i+a,j+b);
                           w2(t) = I2(i+a,j+b+position);
                           t=t+1;
                     end
                  end

                  w = w ;
                  w2 = w2 ;

                  % Metric function to be used : Here it's Hamming distance
                  shd(i,j,position+1) = sum(bitxor(w,w2));

                  if (previous_shd_val > shd(i,j,position+1))
                         previous_shd_val = shd(i,j,position+1);
                         estimated_disparity = position;
                  end
             end
              OutputIm(i,j)=estimated_disparity;
     end
end
```

5.2 Rectification

I did NOT implement this piece of code myself, I used it as part of a pipeline, combining functions previously developed. Indeed to make this code work we need to estimate the Fundamental matrix associated to the unrectified pair of image. The full algorithm and tutorial we studied and try to apply is available here: [4]

5.3 Multiple Baseline

Here is the code to compute the Sum of Sum of Square Differences, find its minimum and compute the resulting inverse depth map

```
%% Compute SSSD defined by Okutomi and Kanade
sssdt2 = ssd1(:,:,1:1:21)+ssd2(:,:,1:2:41)+ssd3(:,:,1:3:61)+ssd4(:,:,1:4:81);
for i = 1:1: size(sssdt2,1)
        for j = 1:1: size(sssdt2,2)
             ttt2=sssdt2(i,j,1:21);
             ttt2=reshape(ttt2,1,21);
             [val,pos]=min(ttt2);
             min_sssd_img(i,j) = pos-1;
        end
end
min_sssd_img_2 = flipdim(min_sssd_img,1);
figure();surf(min_sssd_img_2,'edgecolor','none');grid off;shading interp
```

6 Conclusion

In this final project we had the opportunity to review a large number of computer vision techniques dealing with depth reconstruction based on images and we extended this to an exploration of the state of the art to be able to understand the connexions between the mathematical framework and the constraints coming from real life application based on specific hardware.

We studied and implemented Okutomi and Kanade paper which present a framework to solve ambiguities that occur in a simple baseline stereo. This framework allows us by gathering informations from multiples disparity maps and by introducing an intermediate variable: ζ, to create a common domain for all similarity measurements and obtain a point where all of them agree.
The results we presented with a synthetic data set to illustrate this method seems to be fairly nice and we can resolve almost all ambiguities although the exact shape of the objects of the scene is not strictly conserved. When applied to a real data set of acquired pictures, we can see that this method is a great source of improvement but sometimes fail to get the exact match. This is mainly due to the fact that we cannot keep working with a single point where all similarity measurements agree. Indeed, noise and occlusion will in real situation occur so we have to find the point where most camera agree rather than all of them. This last point should contribute to even better results.

This project also allows us to deeply look at the literature and see all the available methods and techniques involved in Stereoscopy problems and in Computer Vision. Some of them are really linked with each other and sometimes depend a lot on each other. The rectification step is a crucial step in many computer vision pipelines to ensure consistency in exploitation, analysis of images and also in measurements.

Improvements could be made to this framework we developed especially by bringing in Optical Flow which could help us scale the scanning length between rectified pairs according to the displacement and only focus on specific areas for a good match which could hopefully lead to reduced computation time and reduce probabilities of false match.

Finally, this project was maybe not impressive in terms of use of new technologies or graphical representations or interactions but it was really interesting to dig deeply into mathematical frameworks, implementation challenges with respect to hardware capabilities and finally by solving an issue that we noticed in the second project, without being really able to explain it at that time. So, we can conclude this report and project by saying that we explained and contributed through this project to answer our initial question and solve the ambiguity issue.

References

[1] M. Okutomi, T. Kanade, A Multiple Baseline Stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No 4, April 1993.

[2] S. Chambon, A. Crouzil, Similarity measures for image matching despite occlusions in stereo vision, Pattern Recognition 44, pages 2063-2075, 2011.

[3] R. Hartley, Theory and practice of projective resctification, The International Journal of Computer Vision, 1997.

[4] M. Dailey, Stereo Rectification with Matlab/Octave, 2007. Available at: http://se.cs.ait.ac.th/cvwiki/matlab:tutorial:rectification

[5] F. Devernay, Vision stereoscopique et proprietes differentielles des surfaces, These de Doctorat Ecole Polytechnique, FRANCE, 1997.

[6] Wikipedia contributors. Wikipedia, The Free Encyclopaedia, 2013. Available at: http://en.wikipedia.org, Accessed March-April, 2013.

[7] D. A. Forsyth, J. Ponce, Computer Vision: A modern approach, Second Edition, Pearson, 2011.

List of Figures