
**UNIVERSIDADE FEDERAL DE PERNAMUBUCO**
**CENTRO DE INFORMÁTICA**
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**RAFAEL FRANCISCO CAVALCANTI CAMPOS GOUVEIA**

# 3D RECONSTRUCTION AIDED BY MULTIPLE SEGMENTATIONS

**RECIFE**

**2017**

RAFAEL FRANCISCO CAVALCANTI CAMPOS GOUVEIA

# 3D RECONSTRUCTION AIDED BY MULTIPLE SEGMENTATIONS

Monograph submited to the Coordenation of the Computer Science Center for the bachelor degree in Computer Science at Federal University of Pernambuco - Campus Recife, as partial requirement to earn the degree.

Research Field: Computer Vision

Advisor: Dr. Silvio Barros Melo

Recife
2017

# Dedication and Acknowledgments

I dedicate this work to my family, that gave full support to conclude my education. I would like to thank to the teachers and colleagues either from UFPE and from Stevens Institute of Technology that cooperated with my education. Also, an special thanks to the professors Pedro Manhães, Philippos Mordohai, and Silvio Melo that had major contribution for my professional and academic development. At last, I would like to thank Voxar Labs, Francisco Paulo, Silvio Melo and Philippos Mordohai for helping me lending the resources and orientation I needed to conclude this work.

# Resumo

Reconstrução 3D no contexto de Visão Computacional sempre foi uma tarefa computacionalmente complexa. Aplicações baseadas nesse tipo de técnica vêm ganhando destaque na indústria nos anos recentes. Navegação autônoma para carros auto-dirigíveis, drones e robôs; óculos de realidade aumentada são exemplos dessas aplicações. Técnicas usando câmeras estéreo são bastante pesquisadas como uma solução barata e mais acessíveis quando comparado com sensores de luz. Este trabalho tem como objetivo apresentar uma nova técnica para uso de câmeras estéreo utilizando múltiplas segmentações das imagens, com objetivo de aumentar a precisão na construção de mapas de disparidade. Em conclusão, o trabalho deve validar experimentalmente o método proposto, inclusive com respeito à sua eficácia, apresentando suas vantagens e desvantagens com relação a técnicas existentes.

**Palavras-chave**: reconstrução 3D, câmeras estéreo, segmentação de imagens

# Abstract

3D reconstruction in Computer Vision has always been a computationally intensive task. Applications based on this technique have grown substantially in the industry in recent years. Autonomous navigation for cars, drones and robots; augmented reality glasses and many others are examples of such kind of applications. Techniques using stereo cameras are widely solicited for being cheaper and more accessible than light sensors, for instance. This work's goal is to present a novel technique based on stereo cameras to improve any base disparity maps on a set of image segmentations. In conclusion, it presents experiments that can show how effective the method performed, with its advantages and disadvantages.

**Keywords**: 3D reconstruction, stereo vision, image segmentation

# Index

CHAPTER 1

# INTRODUCTION

Computer vision and 3D reconstruction with use of stereo cameras has been actively researched by the academy for a long time. The use of cameras is a cheaper alternative when compared to structured light sensors and a time-of-flight(ToF). Several applications may make use of this technology, since autonomous navigation of cars, drones [2] and robots; to augmented reality applications [6]. In general, these applications need to understand the environment they are in and take decisions based on that, like to avoid obstacles or to interact with real-world objects.

Techniques based on cameras can provide more information like colors. Typical sensors have smaller resolutions when compared to cameras. The main disadvantages regarding cameras are that in the absence of light they lose the ability to collect images, and the processing power needed to create precise reconstructions is still high.

The concept of stereo cameras was created based on human vision, and consists in a set of two cameras separated by any distance. These cameras must be aligned on the same axis and pointed at the same direction. With this configuration we can guarantee that given any 3D point that is projected on the cameras as pixels with the same height, this way reducing the search space.

In order for a technique based on cameras to work perfectly it is necessary to map each pixel from one camera to another with no errors. Even with rectified stereo configuration, in which the search space is reduced to a single line for each pixel, image patterns like solid colors or repetitive textures make this process hard. In addition, more challenges may arise with application requirements, like real-time processing with limited resources. A good example is drone navigation, in which a drone cannot carry much weight, so it must carry limited hardware, and obviously it needs real-time processing in order to navigate.

For the scope of this project, limitations such as hardware and real-time processing

will not be taken into consideration. The focus will be in proposing a novel technique that can improve the precision of any given disparity map previously calculated. Typically methods with global optimizations surpass local methods in precision. However, local methods are usually faster and are necessary for applications that may need to be executed in real-time [7]. Trying to achieve the middle ground between these two, methods that use image segmentation started to be introduced [3] [5]. Segmentation allows optimizations to be done based on regions with similar context, reducing scope when compared with global methods, but absorbing more information than traditional local methods.

The technique presented here makes use of multiple segmentations in order to improve a given disparity map. In each segmentation planes are fitted, with the goal to correct noise from the initial map. Next, the information from the various planes will be combined in order to produce a more accurate result. Other post-processing methods may still be applied on the final result to improve it even further.

## 1.1 Goals and Motivation

The goal of this work is to propose a novel method for enhancing disparity maps based on multiple image segmentations and a disparity map previously calculated. The technique gives as an output a new disparity map with improved disparity when compared to the input. Also, it will be done experiments to evaluate the proposed technique.

CHAPTER 2

# FUNDAMENTS

This chapter presents the base knowledge to understand the work developed. First it will present the concept of disparity maps, following Semi Global Optimization (SGM) and image segmentation based on superpixels.

## 2.1 Stereo Cameras

As described before rectified stereo images are images taken from a specific camera configuration. Figure 1 illustrates this configuration with a set of two cameras. The basic idea behind it, is that object further away will have a smaller disparity when compared to objects closer to the camera, and using this information, together with the distance between cameras and the cameras focus length, it is possible to estimate the 3D position of the closest scene point projected at the images.

### 2.1.1 Disparity Maps

There are many ways to create disparity maps [9] [2] [5] [10], the simplest idea is for each pixel for row $r$ in the left image, look for the most similar pixel in the same row $r$ at the right image. A single pixel does not have enough information to make a good estimation, so it is used a window of size $w$ centered at the target pixels. The windows are compared using cost or similarity function defined by each implementation, for example, NCC, SSD, SAD, etc [4]. The pixel with smaller cost or higher similarity is chosen and the distance between the two pixels is calculated and used as disparity. Taking the left image as the base, the calculation may be described as:

$$\begin{aligned} \text{If } cost(P_{ij},P_{id}) &= \min_{k=0}^{k<j}(cost(Pl_{ij},Pr_{ik})) \\ \text{then } disparity &= abs(j-d) \end{aligned} \tag{2.1}$$

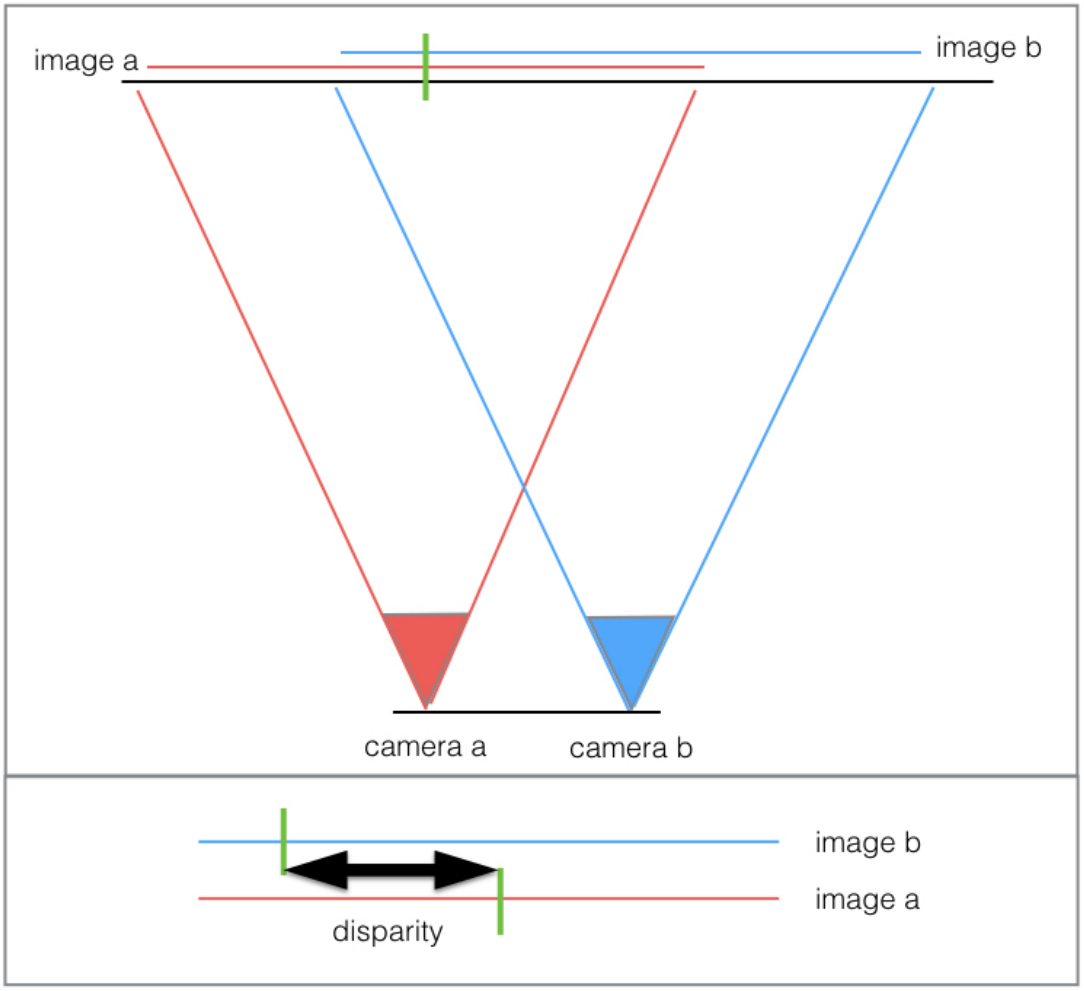



**Figure 1:** Stereo image representation. This disparity is calculated in pixels, and together with camera focus and distance of each other it is possible to triangulate a 3D position. Also it is possible to see that the right image(blue) has the matching pixel(green) to the left of the reference column of the left image(red).

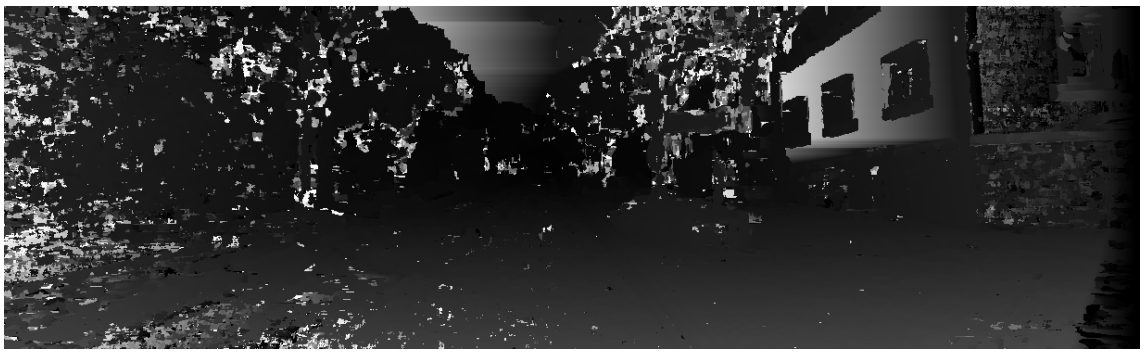
Where $Pl_{ij}$ is the pixel in the left image in row $i$ and column $j$ and $Pr_{i,d}$ is the pixel in the right image in row $i$ and column $j$. The matching pixel in the right image only need to be looked for at the left of the $j$ column as it is possible to see in figure 1. Also the you only need to look as far left as the disparity range the dataset or application requires. As shown in figure 2 the results of this simple strategy is not good enough, it is possible to see some noise in regions without texture.

Using disparities the depth for each pixel may be calculated by a simple equation:

$$Z(p) = b * f / D(p) \tag{2.2}$$

(a) Left Image

(b) Right Image



(c) Disparity Map

(d) SGM Disparity Map

**Figure 2:** (c) is the disparity map using the strategy described at 2.1.1, with NCC and a window of size 9. (d) is the disparity map for the same images using SGM.

Where $Z(p)$ is the depth for pixel $p$, $b$ is the baseline distance between the cameras, $f$ is the camera focus and $D(p)$ is the disparity for pixel $p$.

## 2.2 Semi Global Optimization (SGM)

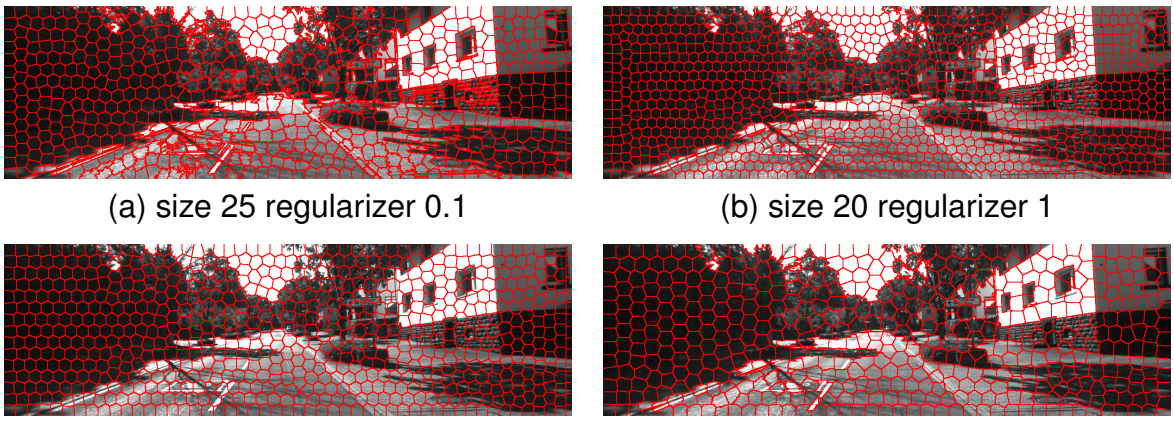
One simple optimization for the pixel matching strategy is SGM. This strategy is based on the idea that disparity discontinuities must be avoided. Using this idea the cost function is modified to add a penalization to disparity discontinuities. Discontinuities may be found and penalized in any direction, left-right, right-left, top-down, diagonals and so on. If the disparity difference between pixel $p_i$ and $p_{i+1}$, is equal to zero then no penalization is added, if it is equal to one then add $P1$ to the cost, and if the difference if bigger than one add then $P2$, where $P1 < P2$.

An initial cost cube is calculated for the whole image. This cost cube will have all costs for each disparity in an certain range, for example 0 to 255. The optimization is done by updating the cost cube with all penalizations. An exhaustive process is not necessary, using dynamic programming it is possible to update all values with a smaller cost.

Taking into consideration only one direction it is possible to reduce the problem to look for the shortest path. In this case each possible disparity for a pixel is a node, the cost to go from a pixel to another is the cost calculated for the initial cost cube plus a penalization in case of disparity discontinuities.

Further optimizations for this technique take into consideration the image gradient in the penalization. Here if there is a big gradient between the pixels being compared it is likely to be and edge of an object in the image, in this case discontinuities are acceptable. So in case the gradient is bigger than a threshold the penalization added is divided by a constant. The whole optimized process is well described by Zbontar and LeCun at [10]

As seen in Figure 3 the results of SGM are a lot better when compared to a more simple approach. However, it is possible to see that even being better there are a lot of black spots with no disparity.



(a) size 25 regularizer 0.1

(b) size 20 regularizer 1

(c) size 25 regularizer 10

(d) size 30 regularizer 1

**Figure 3:** The images in this grid show multiple SLIC superpixel segmentations varying the input parameters

## 2.3 SLIC Superpixels

Superpixel is an strategy for image segmentation that groups pixels based on pixel coordinates and color similarity. There are some variations of this technique, SLIC is one of them. SLIC may take two parameters: one to control the size of the superpixels groups and the other to control how regular its shape is. This algorithm is the superpixel extraction (segmentation) method based on a local version of k-means. This strategy is fast and have interesting results, with a good cost-benefit, as shown by Radhakrishna Achanta et al. at [1]. Through this project we used the implementation from VLfeat library.

As shown in previous work [3] and by Zbontar at [10] it is possible to fit planes or quadratic surfaces with superpixels in order to correct noise on the disparity map. However, it is difficult to optimize the parameters for each image. For different regions of the image the optimal size will vary. In this work I propose a way to combine multiple segmentations in order to get the best fitting for each region.

CHAPTER 3

# PROPOSED METHOD

The proposed method requires multiple segmentations that aggregate pixels based on their neighborhood and color. The assumption is that pixels in the same cluster should have congruent disparities, for example, all disparities of pixels in a wall should be contained in the same plane. More complex shapes as sphere can be reduced to multiple planes, or quadratic curves with acceptable accuracy. Here we will use SLIC superpixels as the segmentation method, because it allows to easily generate multiple segmentations with interesting results.

However, it is difficult to estimate how big the superpixels should be for each region of the image. They need to be big enough so the plane fitting can ignore outliers. At the same time, if they are too big, regions not connected in depth might be put together in the same segmentation. So we use multiple segmentations and fit multiple planes in order to choose the best fit for each region later.

As the input, it is needed the left and the right images and also an initial disparity map that may be calculated by any method. SLIC is used to compute N superpixel segmentations and a RANSAC is used to fit planes for each segmentation using the disparity map with the Z axis as the disparity. Now there are N disparity maps. Confidence measures for each disparity pixel in all disparity maps need to be estimated in order to choose which disparities should be on the combined output.

For the purpose of this work all initial disparity maps were calculated using MATLAB SGM implementation. Also multiple confidence methods defined at [3] were used. These methods were combined with the disparity value and superpixel size and used as a feature vector to train an regression model that unify all these measures. The size of the superpixel used are the same as the ones given as input to VLfeat. As stated in the API documentation [8], the input size is used to determine how many regions the algorithm is initialized with.

## 3.1 RANSAC

Random sample consensus (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers should not have any influence on model estimate. A basic assumption is that the data consists of "inliers", data whose distribution can be explained by some set of model parameters, though may be subject to noise, and "outliers" which are data that do not fit the model. It is a non-deterministic algorithm because it produces a reasonable result only with a certain probability, with this probability increasing as more iterations are allowed. At each iteration a sample of the data is randomly selected to create a model that tries to explain the whole set, and a threshold is used to determine which entries are inliers and outliers. With more iterations it is more likely to find a model that better segment the data in these two groups.

Given the initial disparity $D$, 3D planes were fitted on disparity space using RANSAC. The standard RANSAC procedure was followed, and hypotheses were generated by a minimal sample of three points, computed the equation and counted the inliers to the hypothesized plane. The procedure was exactly same as the one described at [3]. A fixed threshold was used because it was simpler and had no significant difference on an adapting threshold. The final plane for a superpixel is estimated using least squares on all inliers of the best hypothesis. A new disparity map $D_{SP}^{k}$ for segmentation $k$ is obtained by replacing all disparities with those generated by intersecting the ray of each pixel with the estimated plane for its superpixel.

## 3.2 Confidence Measure

It is possible to rank the disparities that are more likely to be correct or not. This can be used to replace the disparities with low confidence. There are many confidence measures that can be taken, and they can be combined using any regression technique. In a previous work [3], it was proposed four methods to evaluate confidence for disparity maps modified with plane fitting; here we will use 3 of them: Inliers Ratio, Neighborhood Consistency, and Slant. These methods show an interesting ability to judge correct and incorrect planes, but it applies the same value for the whole plane. In order to gain more granularity on the confidence estimation a regression model is applied using a 5D feature vector with these three measures, the disparity itself and

the superpixel size. Adding the disparity in the feature vector will the enabled the regression to learn that some disparity values are more likely to be wrong than others. In addition, the superpixel size will help the model to differentiate between the different segmentations. A more complete feature vector like the one used in [3] might improve the results but it will also imply in a higher cost.

### 3.2.1 Random Forest

Random Forests were used due to its performance on inhomogeneous features as it does not requires a metric in feature space. With a 5D feature vector, the biggest limitation is that it can only look at information of one pixel at a time. Other approaches like convolution networks may look to a patch centered at the target pixel to estimate the confidence, and with this extra information it may learn other patterns. The scikit-learn random forest implementation was used through out this work.

## 3.3 Disparity Selection

Now that we have multiple disparity maps with a confidence estimate for each pixel of each image, it is needed to merge all of them in a single disparity map. There are multiple strategies to merge them, the simplest one is to choose the disparity with highest confidence for each pixel. This method does not take into consideration any of the pixels surroundings but it perform better than any of the disparity maps alone.

During the computation of the initial disparity map, it is possible to see the same problem, in which the disparity choosing might be done by looking only for the similarity measure. The SGM tries to optimize the disparity by choosing performing changes to the cost map to penalize discontinuities. Here the same process can be applied. An SGM-like strategy in which instead of the score for each possible disparity, we have the confidence for each of the N disparity maps.

### 3.3.1 Implementation Details

Given all disparities transformed with plane fitting, its source images and the confidence for each pixel of the N disparity maps, the first step to achieve the unified disparity map is to convert confidences in cost following the formula:

$$Cost^k(p) = \frac{1}{(1+Conf^k(p))} \tag{3.1}$$

Where $Conf^k(p)$ is the confidence of the disparity for the pixel at position $p$ in the segmentation k. Other approaches like negate the confidence were tried but ended in significantly worse results.

Now the SGM process is applied replacing the disparity range for all disparity maps $D^k_{SP}$ .The penalization follows the same strategy described by ZBontar and LeCun at [10]. The cost is transformed by the following equation:

$$E^k(p) = Cost^k(p) + \sum_{dx}^{SD}(\min_{l}^{N}(\begin{cases} P1 & \text{if } \{|D^k(p) - D^l(p-dx)| = 1\} \\ P2 & \text{if } \{|D^k(p) - D^l(p-dx)| > 1\}) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Where $k$,$l \in N$, $N$ is the set of disparities calculated for each segmentation, $E^k(p)$ is the new cost for pixel $p$ in the segmentation $k$, $dx \in SD$ and $SD$ is the set of explored directions to apply penalizations, $D^k(p)$ is the disparity for pixel $p$ in the segmentation $k$ and $P1$ and $P2$ are the penalizations.

For optimization purposes $P1$ and $P2$ may vary based on the original images gradient. Let $Gr_{dx} = |I_r(p - D(p)) - I_r(p - D(p) - dx)|$ and $Gl_{dx} = |I_l(p) - I_l(p - dx)|$, where $I_r$ is the right image, and $I_l$ is the left image, $P1$ and $P2$ may be defined as:

| P1 | P2 | Condition |
|---|---|---|
| $sgm\_p1$ | $sgm\_p2$ | if $Gl_{dx} < sgm\_d, Gr_{dx} < sgm\_d$ |
| $sgm\_p1 \div sgm\_q2$ | $sgm\_p2 \div sgm\_q2$ | if $Gl_{dx} \geq sgm\_d, Gr_{dx} \geq sgm\_d$ |
| $sgm\_p1 \div sgm\_q2$ | $sgm\_p2 \div sgm\_q2$ | otherwise |

Here $sgm\_p1$ and $sgm\_p2$ are set as the base penalty for discontinuities. In case only one gradient, for left or right images, is strong, the penalty is reduced by the factor $sgm\_q1$. If the gradients agree and both are strong, the penalty is reduced by $sgm_q2$ where $sgm\_q1 < sgm\_q2$. Also in case the analyzed direction is vertical the penalty is further reduced by a factor $sgm\_v$, because small changes in disparities are more likely to happen when compared to horizontal directions. In the experiments the values used were $sgm\_p1 = 3, sgm\_p2 = 6, sgm\_q1 = 3, sgm\_q2 = 6, sgm\_d = 80, sgm\_v = 2$.

For this implementation only two directions were taken in consideration: left-right, and top-bottom. More directions were tested but it resulted in a higher run-time without significant improvement.

CHAPTER 4

# EXPERIMENTAL RESULTS

In this chapter details the experimentation to evaluate the proposed method performance. Also it describes the database and the parameters values used in experimentations.

## 4.1 Database

All experiments were made with the 2012 KITTI dataset. This dataset contains stereo images of real world street scenarios taken from cameras on the top of a car. A Velodyne laser scanner is used to get a sparse ground truth for almost a third of the pixels. According to the dataset organization a disparity is considered correct using and error margin of three. This means that, if the proposed disparity map calculated a disparity of 10 for a pixel $p$ and the ground truth says its correct disparity is any value between 13 or 7, it will be considered correct. In addition, the dataset has an average resolution of 1241 $x$ 76 and a valid disparities range from 0 to 256.

## 4.2 Experiments

Because of time and resource limitation, for all experiments developed here only the training set of KITTI was used, with 194 images. To begin, an initial disparity map is calculated using MATLAB SGM implementation. Also, regarding the regression model, the training set was divided in two subsets of 97 images, one being used for the training and the other to test. The terms training and test are used bellow to refer to these two subsets.

**Table 1:** Sample results for the oracle with base disparity from SGM

| Image | SGM Disparity Acc | Oracle for $N = 10$ Acc | Oracle for $N = 15$ Acc |
|---|---|---|---|
| 0 | 92,20% | 98,97% | 99,07% |
| 1 | 78,02% | 93,43% | 95,04% |
| 2 | 71,04% | 90,15% | 92,76% |
| 3 | 90,63% | 97,80% | 99,16% |
| 4 | 82,60% | 97,50% | 98,71% |

**Table 2:** Sample results for the oracle with base disparity from NCC

| Image | NCC Disparity Acc | Oracle for $N = 10$ Acc |
|---|---|---|
| 0 | 79,10% | 94,72% |
| 1 | 47,47% | 68,08% |
| 2 | 46,85% | 63,44% |
| 3 | 71,28% | 90,22% |
| 4 | 73,77% | 92,65% |

### 4.2.1 Oracle

To evaluate the method potential, first an implementation that disregard all pixel selection strategy was tested. Here an oracle was used to determine which value of the $N$ disparities calculated by the plane fitting should be chosen. An oracle means that if a correct disparity exists among all the $N$ possibilities it will be selected to the final disparity map. This tests were used to tune how many segmentations should be used.

For all superpixel segmentations the *regularizer* parameter stays fixed as 0.1 and the size parameter varied. This was done because the size was more critical to get different segmentations. For $N$ equal to 10 the first segmentation starts with region size equal to 13 growing up to 40 with intervals of 3, giving approximately 2000 and 300 superpixels respectively. For $N$ equal to 15 the first segmentation starts at 10 growing up to 80 with intervals of 5, ranging from 5000 to 150 superpixels approximately.

Considering the whole set, all 194 images, the SGM initial disparity set had an average accuracy of 80.73%. Also, no single segmentation created an disparity with an average superior to 83,72%. At the table 3 it is possible to see how the accuracy can vary depending on the size of the superpixel, images with higher level of detail may benefit from smaller planes, and images in which big plane surfaces are predominant, bigger planes will generate better results. This can also vary depending on the image region.

**Table 3:** Disparity Accuracy by superpixel size based on the SGM disparity map

| | Image 0 | Image 1 | Image 2 | Image 3 | Image 4 |
|---|---|---|---|---|---|
| Superpixel size 10 | 93,62% | 81,23% | 74,70% | 92,31% | 86,26% |
| Superpixel size 15 | 93,41% | 82,86% | 76,21% | **92,34%** | 86,12% |
| Superpixel size 20 | 93,70% | 83,54% | 75,34% | 91,58% | 86,47% |
| Superpixel size 25 | 93,67% | 82,13% | 76,89% | 91,63% | 86,81% |
| Superpixel size 30 | **94,12%** | 81,85% | **78,05%** | 90,73% | 86,34% |
| Superpixel size 35 | 93,05% | 85,21% | 77,00% | 89,78% | **88,04%** |
| Superpixel size 40 | 92,95% | 84,04% | 75,05% | 89,81% | 84,40% |
| Superpixel size 45 | 92,29% | **84,49%** | 74,96% | 90,94% | 84,56% |
| Superpixel size 50 | 92,09% | 83,92% | 77,25% | 88,39% | 85,45% |
| Superpixel size 55 | 93,50% | 81,48% | 73,35% | 87,88% | 83,85% |
| Superpixel size 60 | 90,44% | 83,69% | 76,25% | 87,40% | 85,68% |
| Superpixel size 65 | 91,84% | 76,57% | 71,57% | 88,10% | 84,77% |
| Superpixel size 70 | 90,36% | 79,96% | 73,14% | 87,44% | 87,46% |
| Superpixel size 75 | 89,25% | 80,15% | 72,09% | 84,84% | 84,99% |
| Superpixel size 80 | 89,51% | 77,38% | 74,49% | 85,59% | 83,78% |

The oracle will be able to identify the regions of the image that will benefit from bigger or smaller planes, and choose it correctly. The oracle achieve 93,55% and 95,70% for N equals 10 and 15 respectively. This demonstrates how effective the method can be, given that the pixel choosing is done perfectly. Also, this demonstrates that more segmentations results in better accuracy.

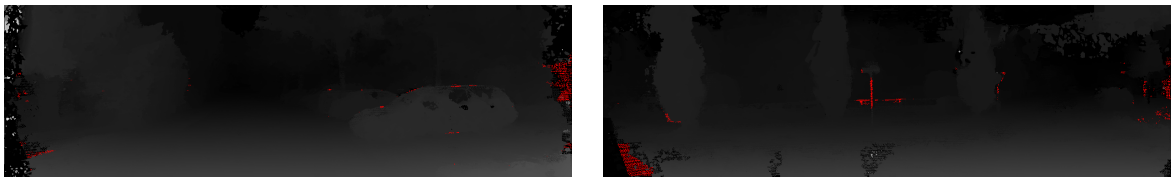
Tests were also made with NCC with a $9x9$ window size as the base disparity. In this case, average accuracy for the set was 58.14% and with and N equal to 10 with the size varying the same as described above, the accuracy was 76,82%. This shows a higher improvement for initial disparities with more noise. Aiming to achieve the smallest error rate in the final disparity map, only the SGM will be considered for the rest of the experiments.

### 4.2.2 Random Forest

At the same time that the planes were estimated for all segmentations and the new disparity maps were created, all confidence measures described in the previous section were taken. As parameters were used: a hundred trees , the criterion parameter was mean squared error, maximum tree depth of seven, minimum sample split of ten thousand, minimum impurity of $10^{-4}$, and the maximum number of feature in automatic mode. More trees may improve the results, also the maximum depth and the min split are limited to avoid over fitting.

**Table 4:** Accuracy table for final results

| | SGM Disparity | Max Confidence | SGM-like Choosing |
|---|---|---|---|
| Average Acc. Training | 80.45% | 89.59% | 90.25% |
| Average Acc. Test | 81.06% | 89.85% | 90.46% |
| Average Acc. Total | 80.76% | 89.75% | 90.38% |



(a) Error map of image 3 (b) Error map of image 4

**Figure 4:** The images show in red the pixels were the estimated disparity are wrong. It is important to remind that the ground truth is sparse, and that is why the regions are not dense.

## 4.3 Results

When the random forest confidence output was used, and the disparity selecting method is based only on the maximum confidence the method achieves an average precision of 89.85% for the training set. This clearly shows space for improvement when compared to oracle. Using the SGM-like strategy for the disparity choosing, the result showed an precision of 90.46%, these are better results but still have some room for improvement.

To better understand the results, error maps were created as shown in figure 4. The red regions illustrate where the method estimated wrong disparities. It is possible to see that some specific regions concentrated the errors. For example, some pixels close to the edges, probably it happened because they were misplaced by the segmentations. Also there are some regions where initial error rate did not allow a good plane estimation, like occlusions or regions without texture.

Further improvement is still possible with post processing techniques, for example using a simple median filter. The median filter, with window size $3x7$ used 20 times iteratively, was able to fix the regions close to the edges mentioned above, but some other regions might suffer, like thin vertical regions as street signs and tree trunks. At the end the results improved the average accuracy further 0.7%. The complete method, including the post processing, improved the initial disparity in 9.38% for the test set. More complex post processing may improve the accuracy even more.

(a) Right Image

(b) Final Disparity Map

(c) SGM Disparity Map

**Figure 5:** (a) Is the base left image. (b) Is the output of the method. (c) Output after the median filter iterations.

CHAPTER 5

# CONCLUSION AND FUTURE WORK

In this work, a technique that uses multiple segmentation to aid on the calculation of a disparity map was proposed. Multiple segmentation are taken from the base image, and later used to fit planes at the disparity space on a initial disparity map, creating a new disparity map for each segmentation. Then, all new disparity maps generated needs to be merged to achieve a single disparity map with a higher accuracy. When the oracle was used to merge the disparities, it demonstrated that the method has the potential to reach 95.7% accuracy. To replace the oracle, an RF regression model was used in order to compute a confidence measure for each pixel and SGM like strategy was used to choose the disparity for each pixel based on the RF confidence estimation reaching 90.38% accuracy.

The proposed method showed an ability to give a considerable improvement to the initial disparity map. Even with the initial disparity map having a high error rate as in the NCC, the technique showed the potential for large improvements. Also the method reasonably easy to implement.

In addition there is a large space for improvement on the pixel selection and on the regression model applied. Also, different segmentation strategy may apply. These parts can be further researched in order to improve the method even further. The results are interesting but need further improvement to be compared with the state of art. In the moment this monograph is being written, the rank available at KITTI website shows that state of the art techniques recently achieved an accuracy of 98.23%.

# References


[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Andrew J Barry and Russ Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3046–3052. IEEE, 2015.

[3] Rafael Gouveia, Aristotle Spyropoulos, and Philippos Mordohai. Confidence estimation for superpixel-based stereo matching. In *3D Vision (3DV), 2015 International Conference on*, pages 180–188. IEEE, 2015.

[4] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[5] Ang Li, Dapeng Chen, Yuanliu Liu, and Zejian Yuan. Coordinating multiple disparity proposals for stereo computation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4022–4030, 2016.

[6] Marko Markovic, Strahinja Dosen, Christian Cipriani, Dejan Popovic, and Dario Farina. Stereovision and augmented reality for closed-loop control of grasping in hand prostheses. *Journal of neural engineering*, 11(4):046001, 2014.

[7] Daniel Scharstein, Richard Szeliski, and Ramin Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision, 2001.(SMBV 2001). Proceedings. IEEE Workshop on*, pages 131–140. IEEE, 2001.

[8] Andrea Vedaldi. Simple linear iterative clustering (slic). `http://www.vlfeat.org/api/slic.html`, 2017. [Online; accessed 3-June-2017].

[9] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, pages 756–771. Springer, 2014.

[10] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.