Automated Placement of Multiple Stereo Cameras

Rahul Malik, Peter Bajcsy

Automated Placement of Multiple Stereo Cameras

Rahul Malik†and Peter Bajcsy‡

† Department of Computer Science
University of Illinois at Urbana-Champaign, USA
Email: *rmalik4@uiuc.edu*
‡ National Center for Supercomputing Applications,
University of Illinois at Urbana-Champaign, USA
Email: *pbajcsy@ncsa.uiuc.edu*

Abstract. This paper presents a simulation framework for multiple stereo camera placement. Multiple stereo camera systems are becoming increasingly popular these days. Applications of multiple stereo camera systems such as tele-immersive systems enable cloning of dynamic scenes in real-time and delivering 3D information from multiple geographic locations to everyone for viewing it in virtual (immersive) 3D spaces. In order to make such multi stereo camera systems ubiquitous, solving the problem of optimal deployment (configuration) of 3D imaging components is motivated by the need (a) to create high quality 3D content and (b) to accommodate application specific requirements into optimal deployment without ad-hoc experimentations. One of the configuration parameters is the placement of stereo cameras that affects the quality of 3D reconstructions as well as the resolution achieved by the reconstruction. The novelty of our work is in formulating an optimization framework for optimal camera placement using error based objective function and five constraints. We generate an initial solution to the optimization problem using Genetic Algorithms and then refine the solution using Gradient Descent. The algorithm is validated using actual camera placement as well as using simulation results. The results not only show promising features of our optimization approach but also eliminate ad-hoc experimentation of camera placement for each end application where multiple stereo camera systems can be deployed.

1 Introduction

Multiple stereo camera systems are becoming ubiquitous these days. Applications of such systems such as tele-immersive systems are being used for performing activities such as remote monitoring of physiotherapeutic patients, understanding and annotating dance movements, identification and tracking, etc. The main requirement for all these applications is to have a high resolution and accuracy of 3D video content. Therefore, it is important to place the 3D stereo cameras in space optimally. If the cameras are placed too near from an object then although the resolution will be high but the 3D reconstruction error is also high. Also, the entire object may not be captured. On the other hand, if they are placed too far from the object, then the resolution is low. So, a good resolution is required ensuring that the whole object is visible in order for the object features to be easily distinguished from one another. Therefore, optimal camera placement is important for achieving for the highest resolution and lowest 3D error.

Another requirement comes from the fact that it eliminates ad-hoc experimentation in camera placement, which is application specific. Different applications have different working space requirements. For example, the requirements for camera placement for application involving wheelchair basketball players differs from dancers and from application of tele-conference. Thus, the solution of this problem aids in easy and quick placement of many cameras in space in an application specific manner. It also helps in saving user's time in camera deployment.

One of our applications for which we are developing camera placement is tele-immersive systems (TEEVE)[1][2][3]. In our work, we use TEEVE experimental system, which is organized into several layers depending on the functionality required. These consist of the *capturing layer*, the *transmission layer* and finally the *displaying layer*[1]. The *capturing layer* is of our interest since it consists of a series of 3D cameras, each with four 2D cameras. Each 3D camera is a set of three greyscale cameras used for 3D reconstruction and one color camera for providing color information. These four 2D cameras are fixed and calibrated in the beginning.

In this paper, we consider the task of determining the placement of 3D stereo cameras in space so as to reduce reconstruction error in the rendered video as well as to improve the spatial resolution 3D content. The input parameters include the application defined working volume for the users in space, camera properties and the room dimensions. The output is a determination of the 3D camera positions and orientations in space. Identifying camera placement positions is hard due to all the large and unknown number of configurations having a very similar accuracy, but with a very different imaging geometry. Several constraints need to be taken into account such as placement limitations, camera visibility and so on. The goal of stereo camera placements is to view all sides of a working space except from the bottom surface since it would be defined by the room floor. We assume that the working space is represented by a convex hull in 3D.

Recently, some researchers have started looking the the problem of camera placement. In [4, 5], authors have looked at placement of 2D cameras in order to avoid occlusions. They have not looked at 3D reconstruction or the resolution of features that our paper does. In the past in photogrammetry [6], authors have looked at the problem of optimal camera placement. However, their approach was limited to very few number of cameras and the kind of assumptions made in the work are very limiting for actual placement. In this paper, we have actually implemented the algorithm and tested it in both real life scenarios as well as on a simulator.

The major contributions of this paper are:

1. Formulation of objective function and constraints for automatic placement of multiple stereo cameras.
2. Solving the optimization problem using genetic algorithms and improving the result by using a gradient descent algorithm.
3. Validating the algorithm by comparing it against other heuristic placements by actual placement as well as by simulations.

The paper is organized as follows. Section 2 provides related work. In section 3, we formulate the objective function as well as the constraints for optimization of camera placement. In section 4, we provide experimental results in order to validate the algorithm. In section 5, we provide the conclusions and scope for future work.

2 Related Work

Researchers in computer vision and computer graphics have recognized the need to automate the process of camera placement. The camera placement can be broadly divided into two broad categories: single camera placement and multiple camera placement.

In single camera systems, Sakane et al. [7] developed a system that finds possible camera positions using a generate and test strategy. This is for the inspection of an object tessellated by a sphere of a given radius. They have incorporated an analysis of light source placement. This work was extended by Sakane and Niepold [8]. For camera placement, the main task constraint considered is edge visibility. The sensor is positioned to minimize the occlusion of selected feature edges. The evaluation of this criteria is based on an aspect graph representation of the object. Other works have adopted a synthetic approach. Work by Cown et al. [9] optimizes the camera locations from which a specified set of object features can be viewed. The machine vision planning (MVP) system developed by Tarabanis et al. [10] determines the optimal location and camera parameters such as focal length, focus setting and aperture for viewing a set of surfaces and avoiding occlusion.

Work has been done in past in the area of multiple camera systems also. Cowan et al. [11] have experimented methods to place multiple sensors overcoming the occlusion problems associated with 3D objects. Fritsch and Crosilla [12] have investigated the potential of optimizing multi-camera configurations using an analytical first order design (FOD) approach by iteratively shifting the cameras until the covariance matrix of the estimated object feature coordinates was better than a criterion matrix. In [13], a subset of horizontal camera sensors are selected to minimize the visual hull of all the objects in the scene. This problem is solved using heuristics. These works use numerical techniques or heuristics to compute the viewpoint scores. In [14, 15], a metric is defined for the next best view based on most faces seen (given a 3-D geometric model of the scene), most voxels seen or overall coverage. The solution requires searching through all camera positions to find the highest scoring viewpoints. Recently, Cerfontaine et al. [16] have proposed a method to determine the optimal camera alignment for a tracking system with multiple cameras by specifying the volume to be tracked and an initial camera setup.

In photogrammetry [17, 6, 18], the goal is to place the cameras so as to minimize the 3D measurement error. The error propagation is analyzed to derive an error metric that is used to rank camera placements. The best camera placement is then solved numerically. The computational complexity of this approach only allows solutions involving only a few cameras.

3 Problem Formulation

3.1 Problem Statement

We formulate the problem as an optimization problem. The *input* to the optimization problem are following:

1. Number and type of cameras
   - Horizontal and vertical field of view
   - Focal length
   - Minimum and maximum cutoff length
   - Minimum and maximum camera placement heights
2. Room dimensions
   - Minimum and maximum X, Y, Z for room
3. Working volume
   - Coordinates of the vertices of working volume

In our problem formulation, the cameras could all be of different kinds. We assume that the working volume is an arbitrary 3D shape and user just needs to input the vertices of the 3D shape. The working volume is quantized into grid points by the program as shown in figure 1 and for our purposes, we just use the grid points.

The *output* is to find the camera placement for all the cameras. This includes finding the position and rotational orientation for the cameras.

3.2 Formulation of Objective Function

Here we describe the formulation of objective function for the problem of optimal camera placement. We have two objectives to be fulfilled. The first one is to minimize stereo localization error which is obtained from stereo localization geometry. The second is to maximize the pixel resolution for each of the stereo cameras. As previously mentioned, although each 3D camera consists of four 2D cameras, only two are used for stereo reconstruction, one for verification of reconstruction and the last, which is a color camera, for adding color.

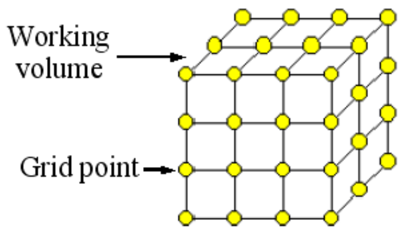

Fig. 1. Division of working volume into grid.

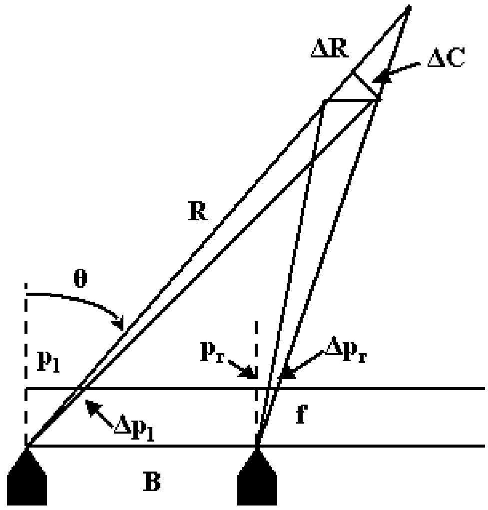

Fig. 2. Stereo localization geometry

Stereo Localization Error We study both the down-range error as well as the cross-range error and use both of them for minimization. Localization error in the direction of viewing from a camera is called down-range error and localization error normal to that is called cross-range error. The two errors are shown in figure 2 as ΔR and ΔC respectively.

Kim et al.[19] have derived the relationship of the disparity error to the stereo 3D localization error. For a general case when both Δp_l and Δp_r are non-zero, the expression for down-range error that they derived is,

$$\Delta R = -(R^2 \cos\theta / fB)(\Delta p_l - \Delta p_r), \tag{1}$$

where, B is stereo baseline, f is the focal length, θ is the angle made with normal to the camera axis and R is the distance of the feature from the camera. Since we are interested in the magnitude of down-range error only, we just take the magnitude and use it as,

$$|\Delta R| = |(R^2 \cos\theta / fB)(\Delta p_l - \Delta p_r)|. \tag{2}$$

This provides an expression for down-range error. For the cross-range error, which is perpendicular to the line of sight, they found it to be,

$$\Delta C = (R \cos^2\theta / f)(\Delta p_l + \Delta p_r)/2, \tag{3}$$

And as we are interested in its magnitude only, this gives us an expression for cross-range error as,

$$|\Delta C| = |(R \cos^2\theta / f)(\Delta p_l + \Delta p_r)/2|. \tag{4}$$

Pixel Resolution Considering the pixel resolution, we need to ensure that a pair of points in space at distance l apart from one another should be spaced apart in the camera plane also. Tarabanis et al.[20] analyzed the size of a linear feature in the image plane. They found out that the equation defining the size of a linear feature of size l in the image plane is:

$$PR_{ab} = \frac{dl|[(r_a - r_o) \times u] \times v|}{((r_a - r_o) \cdot v)((r_b - r_o) \cdot v)} \tag{5}$$

where l is the length of the linear feature to be viewed having r_a and r_b as its end points, u is the unit vector along the linear feature (from r_a to r_b), PR_{ab} is the size of that feature in the image plane, d is the distance from the back nodal point of the lens to the image plane, r_o is the position vector of the frontal nodal point of the lens, and v is the unit vector along the optical axis in the viewing direction.

3.3 Formulation of Constraints

Here, we discuss the constraints that are applicable to camera placement. The following constraints are enforced: room constraint, camera placement constraint, field of view constraint, cutoff length and 360° visibility constraint. We describe and formulate each of these constraints now.

Room Dimensions Constraint This constraint specifies the dimensions of space where the cameras can be placed. The user specifies minimum and maximum values of the x, y dimensions. Therefore, the constraint can be formulated as

$$x_{min} \leq x_i \leq x_{max} \tag{6}$$

$$y_{min} \leq y_i \leq y_{max} \tag{7}$$

$$\forall i \in Cameras$$

where, x_i, y_i are the coordinates of the camera placement of ith camera.

Camera Placement Constraint This constraint corresponds to the height of the camera placement. We assume that the cameras are placed on the tripods and the camera placement height is constrained by a minimum and maximum value. Therefore, minimum and maximum camera placement heights are provided as input and the constraint is formulated as following

$$z_{min\_placement} \leq z_i \leq z_{max\_placement} \tag{8}$$

$$\forall i \in Cameras$$

where, z_i is the height coordinate of ith camera.

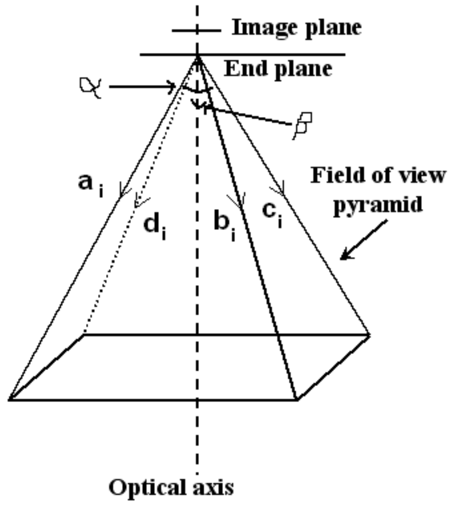

Fig. 3. The field of view pyramid

Field of View Constraint The field of view constraint is concerned with determining viewpoints from which the features of interest can be seen. If the field of view constraint is violated, then certain features will not be seen at all. Consequently, the field of view constitutes a *hard* constraint in camera placement.

The field of view of a general optical system is described by the region in object space that is bounded by the *field of view pyramid* of the system. This pyramid is described in figure 3. In practice, the field of view is specified in terms of the angle that the extreme rays make while entering the optical system. There are two field of view angles, viz. *horizontal field of view angle* denoted as α and the *vertical field of view angle* denoted as β. These two angles are the angles subtended at the entrance pupil by the entrance window of the system.

Intuitively, this constraint can be decomposed into two constraints, i.e. the desired point must lie between two planes for both the sets of planes. Mathematically, this constraint can be formulated as follows

$$((P_i - [x_i\ y_i\ z_i]^T) \cdot (a_i \times b_i)).((P_i - [x_i\ y_i\ z_i]^T) \cdot (c_i \times d_i)) < 0 \tag{9}$$

$$((P_i - [x_i\ y_i\ z_i]^T) \cdot (a_i \times d_i)).((P_i - [x_i\ y_i\ z_i]^T) \cdot (b_i \times c_i)) < 0 \tag{10}$$

where, P_i is a point that we want to determine whether it lies inside pyramid, $[x_i\ y_i\ z_i]^T$ are the coordinates of camera placement, a_i, b_i, c_i, d_i are the four unit vectors along the edges of pyramid as shown in figure 3. These four unit vectors can be written as

$$\{a_i, b_i, c_i, d_i\} = \frac{\pm w_i.l_\alpha/2 \pm u_i.l_\beta/2}{|| \pm w_i.l_\alpha/2 \pm u_i.l_\beta/2||} \tag{11}$$

where, w_i, u_i are the vectors parallel to the camera plane and l_α, l_β are the horizontal and vertical widths between the planes at a unit distance from the end plane. They can be written as

$$l_\alpha = 2\tan(\alpha/2) \tag{12}$$

$$l_\beta = 2\tan(\beta/2) \tag{13}$$

Cutoff Length Constraint This constraint corresponds to the minimum and maximum cutoff length in stereo cameras. In the case of stereo cameras, if the object to be viewed is too near or too far from the camera, then the reconstruction quality is bad. As a result of this, a minimum and maximum cutoff length is defined for the stereo cameras and only the features lying within the cutoff lengths are taken into account. Mathematically, this constraint can be formulated as follows

$$CLi_{min} \leq ||([x_i\ y_i\ z_i]^T - P_j) \cdot v_i|| \leq CLi_{max} \tag{14}$$

$$\forall i \in Cameras, \forall j \in Grid\ points$$

where, $||\cdot||$ represents norm of the vector, CLi_{min}, CLi_{max} are the minimum and maximum cutoff lengths of the ith camera, $[x_i\ y_i\ z_i]^T$ are the coordinates of its placement, v_i is the unit vector along the optical axis in the viewing direction of the camera and P_j are the coordinates of the jth grid point.

360° Visibility Constraint Finally, the last constraint that we dealt with is 360° visibility constraint. This constraint deals with the fact that each of the points in the working volume must be visible from 360° by at least one camera. This constraint ensures 360° visibility of the entire working volume. We define a point to be visible from two cameras if the angle between those cameras is less than 90°. Mathematically, this constraint can be formulated as

$$min(\theta_{i,j}) < 90^\circ \qquad \forall j \in Cameras, j \neq i \tag{15}$$

$$\forall i \in Cameras$$

where, $\theta_{i,j}$ is the angle subtended between camera i and camera j for a grid point. The subtended angle can be calculated as

$$\theta_{i,j} = \arccos \frac{([x_i\ y_i\ z_i]^T - P_k) \cdot ([x_j\ y_j\ z_j]^T - P_k)}{||[x_i\ y_i\ z_i]^T - P_k||\ ||[x_j\ y_j\ z_j]^T - P_k||} \tag{16}$$

where, $[x_i\ y_i\ z_i]^T$ and $[x_j\ y_j\ z_j]^T$ are the coordinates of ith and jth camera placement, P_k is the point where the subtended angle is calculated and $||\cdot||$ denotes the norm of the vector.

3.4 Solution Methodology

Upon combining the objective functions and the constraints, the optimization problem can be written as follows

$$
\begin{aligned}
&\textbf{minimize:}\\
&\frac{\gamma}{\gamma+\delta}\sum_i\sum_j((\Delta R_{ij})^2+(\Delta C_{ij})^2)-\frac{\delta}{\gamma+\delta}\sum_i\sum_j(PR_{ij})\\
&\qquad\qquad \forall i \in Cameras, \forall j \in Grid\ points\\
&\textbf{Subjected to:}\\
&\quad \textbf{(1) Room Dimensions Constraint}\\
&\quad x_{min} \le x_i \le x_{max}\\
&\quad y_{min} \le y_i \le y_{max}\\
&\quad \textbf{(2) Camera Placement Constraint}\\
&\quad z_{min\_placement} \le z_i \le z_{max\_placement}\\
&\quad \textbf{(3) Field of View Constraint}\\
&\quad ((P_i-[x_i\ y_i\ z_i]^T)\cdot(a_i\times b_i)).((P_i-[x_i\ y_i\ z_i]^T)\cdot(c_i\times d_i))<0\\
&\quad ((P_i-[x_i\ y_i\ z_i]^T)\cdot(a_i\times d_i)).((P_i-[x_i\ y_i\ z_i]^T)\cdot(b_i\times c_i))<0\\
&\quad \textbf{(4) Cutoff Length Constraint}\\
&\quad CLi_{min} \le ||([x_i\ y_i\ z_i]^T-P_j)\cdot v_i|| \le CLi_{max}\\
&\quad \textbf{(5) } 360^\circ \textbf{ Visibility Constraint}\\
&\quad min(\theta_{i,j}) < 90^\circ
\end{aligned}
\tag{17}
$$

where, ΔR_{ij} denotes down-range error and ΔC_{ij} denotes cross-range error of ith camera at jth grid point and PR_{ij} represents pixel resolution. The negative sign in front of expression for pixel resolution denotes that it needs to be maximized instead of being minimized.

In order to solve the optimization problem, we used *Genetic Algorithms* [21] to develop an initial solution to the optimization problem. Genetic algorithms are a category of global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. The main advantage of using genetic algorithms to get an initial solution is that they are global search techniques and thus do not get stuck at a local optimum easily.

In order to further refine the solution, we used the *Gradient Descent Algorithm* [22]. Gradient descent is a local optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or the approximate gradient) of the function at the current point. We used the initial solution obtained from genetic algorithm as a starting point for gradient descent. This helps to further refine the solution and get better results.

4 Experiments and Results

We compared our algorithm for camera placement with three other heuristic placement techniques. These heuristic placements are:

1. **Random Placement:** Place the cameras randomly in space. Only take care that they follow all the constraints.
2. **Nearest Placement:** Place the cameras as close to the working volume as possible, still following the constraints.
3. **Farthest Placement:** Same as nearest, instead place the cameras as far as possible from the working volume.

For the purpose of our experiments, we selected a 125" × 105" room. The working volume was a cube with dimensions 20" × 20" × 20" and it was located in the center of the room. We implemented the algorithm using Matlab[1] and C++. We used the Genetic Algorithms and Gradient descent implementations from Matlab and Open CASCADE[2] was used for visualizing the camera placements. In order to test the algorithm, we compared it against the heuristics by actual camera placement as well as by simulations. We describe both of these in detail now.

4.1 Actual Camera Placement

In order to do this, we actually placed the cameras according to the three placement techniques described above and the results obtained from the developed algorithm. Figure 4 shows the camera placements achieved by different placement techniques. In the figure, the blue circles represent the cameras. As described earlier, one 3D camera consisted of four 2D cameras. Therefore, each 3D camera is shown as a cluster of four cameras. The blue line in the figure shows the central camera axis and the red circles show the points in the volume of space that was used for our experiments.

For the purpose of our experiments, we had three identical 3D cameras and we placed them according to different placement techniques. In order to compare different placements, we moved a spherical ball in the working volume. We measured the average reprojection error in terms of number of pixels and the average size of the object in terms of number of pixels. The results are shown in table 1. The reprojection error is a geometric error corresponding to the image distance between a projected point and a measured one. This was measured during calibration using the algorithm developed by Svoboda et al. [23]. This corresponds to our first objective function and better the camera placement, the lower the reprojection error should be. The average object size is a measure of resolution and higher the resolution, higher the average object size. So, better the camera placement, higher the object size. Therefore, for a good placement, we want the reprojection error to be minimum and object size to be maximum.

[1] http://www.mathworks.com/products/matlab/
[2] http://www.opencascade.org/

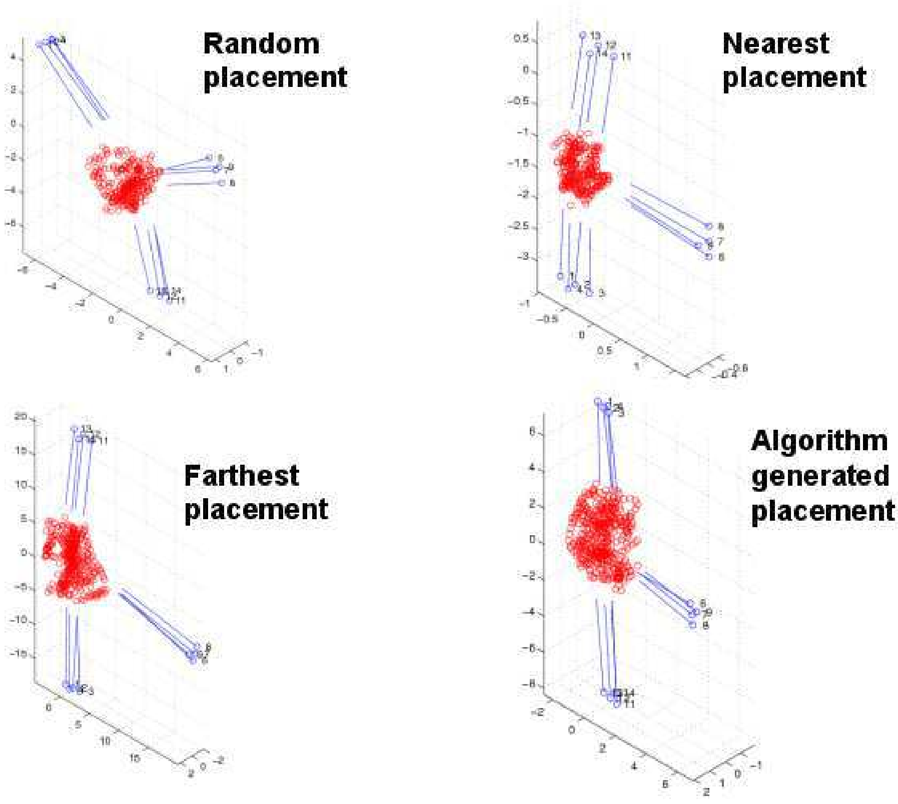

Fig. 4. Actual camera placements in space for different techniques.

As can be seen from the results of table 1, in terms of reprojection error, the camera placement generated from our algorithm gives the lowest error. The error is much lower than any other technique. For average object size, the nearest placement technique gives slightly better results than our placement technique. This is logical since closer we place the cameras, the higher the resolution will be achieved. Overall, we can see that the camera placement generated from our algorithm gives the best results.

4.2 Camera Placement Simulations

In this, instead of actually placing the cameras, we looked at the variation of the objective function per camera cluster as a function of number of camera clusters for different placement techniques. The objective function per camera cluster is the objective function divided by the number of camera clusters and gives an average value of objective function per cluster. The results are shown in figure

Table 1. Reprojection error and average object size for different actual camera placements.

Placement techniques	Reprojection error (pixels)	Avg. obj. size (pixels)
Nearest	0.27	112.5
Random	0.21	94.3
Farthest	0.35	57.4
Our algorithm	0.15	97.6

5 . As the objective function is a minimization function, so lower the value of objective function for a placement technique, the better is the placement. The aim of these experiments is to look at the results for larger number of camera clusters and to look at a combined objective function so that it is easy for comparing different techniques.

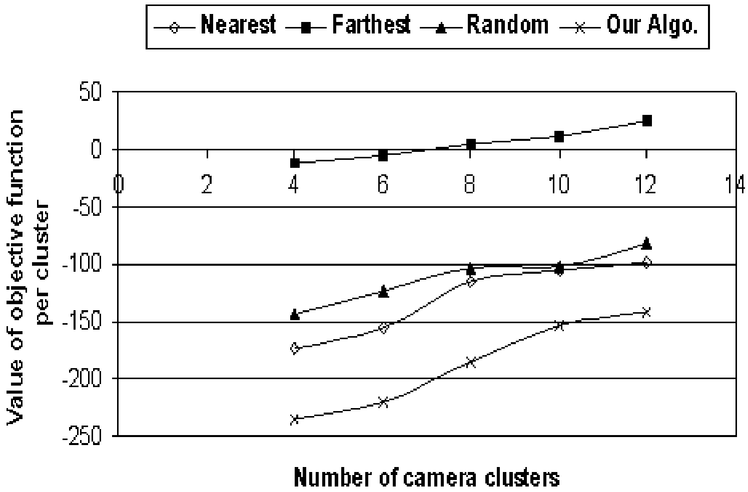

Fig. 5. Variation of objective function per cluster as a function of number of camera clusters for different placement techniques.

For the figure 5, we can observe that the placement of cameras achieved from our algorithm gives the best overall value of objective function as compared to other techniques. The results of farthest camera placement are the worst. This is because as the cameras are placed farther away, the reconstruction error increases and the resolution also decreases. As a result, it gives the worst results. Nearest and random placements give almost same kind of results. Another thing to notice from the graph is that as the number of camera clusters is increased, the value of objective function per cluster increases. This is because the optimization function that we use is a global minimization function and the average value for each cluster gets less optimized as the number of clusters increases.

5 Conclusions and Future Work

In this paper, we have presented an algorithm for the placement of multiple stereo cameras in space in order to minimize reconstruction error and to maximize the resolution of features at all the cameras. The problem was posed as a constrained optimization problem and solved using genetic algorithm and gradient descent. The problem of camera placement is centrally important for object identification, tracking and searching in 3D tele-immersion systems. This work also eliminates ad-hoc experimentations of camera placements for each end application and therefore there is a significant time saving and better system performance. We have tested the algorithm and compared it against other heuristic placements. The experimental results show the goodness of our technique.

In the future, we would like to extend this work to identify subspaces of camera placement. This is because slight movement of the cameras in the space does not make a great difference to the results. So, instead of identifying the exact position of camera placement, we would like to identify the region of space where a camera can be placed. We would also like to perform more detailed analysis of the camera placement by other techniques such as perturbation theory. This would help in validating the algorithm theoretically as well.

References

1. Yang, Z., Cui, Y., Yu, B., Liang, J., Nahrstedt, K., Jung, S.H., Bajscy, R.: Teeve: The next generation architecture for tele-immersive environments. 7th IEEE International Symposium on Multimedia (2005)
2. Yang, Z.: A multi-stream adaptation framework for tele-immersion. ACM Multimedia (2006)
3. Yang, Z., Yu, B., Diankov, R., Wu, W., Bajcsy, R.: Collaborative dancing in tele-immersive environment. Proc. of ACM Multimedia (2006)
4. Erdem, U.M., Sclaroff, S.: Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. Computer Vision and Image Understanding **103**(3) (2006) 156–169
5. Ercan, A.O., Yang, D.B., Gamal, A.E., Guibas, L.J.: Optimal placement and selection of camera network nodes for target localization. DCOSS (2006) 389–404
6. Olague, G., Mohr, R.: Optimal camera placement for accurate reconstruction. Pattern Recognition **35** (2002) 927–944
7. Sakane, S., Sato, T.: Automatic planning of light source and camera placement for an active photometric stereo system. IEEE International Conference on Robotics and Automation (April 1991) 1080–1087
8. Sakane, S., Niepold, R., Sato, T., Shirai, T.: Illumination setup planning for hand-eye system based on an environmental model. Adv. Robot (1992) 461–482
9. Cowan, C., Kovesi, P.: Automatic sensor placement from vision task requirements. IEEE Trans. Pattern Analysis Machine Intelligence (1988) 407–416

10. Tarabanis, K., Tsai, R.: Computing view points that satisfy optical constraints. Proceedings of the Conference on Computer Vision and Pattern Recognition (1991) 152–158
11. Cowan, C., Bergman, A., Nitzan, D.: Automatic placement of vision sensors. 1990 NSF Manufacturing System Research Conference (1990) 389–395
12. Fritsch, D., Crosilla, F.: First order design strategies for industrial photogrammetry. Close-Range Photogrammetry Meets Machine Vision **1395** (1990) 432–438
13. Yang, D., Shin, J., Ercan, A., Guibas, L.: Sensor tasking in occupancy sensing in a network of cameras. Proceedings of BASENETS (2004)
14. Vazquez, P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. Proceedings of the Vision Modeling and Visualization (2001)
15. Wong, L., Dumont, C., Abidi, M.: Next best view system in a 3d object modeling task. Proceedings of Computational Intelligence in Robotics and Automation (1999)
16. Cerfontaine, P.A., Schirski, M., Bundgens, D., Kuhlen, T.: Automatic multi-camera setup optimization for optical tracking. Proceedings of the IEEE conference on Virtual Reality (March 2006) 295–296
17. Chen, X., Davis, J.: Camera placement considering occlusion for robust motion capture. Stanford University Computer Science Technical Report CS-TR-2000-07 (2000)
18. Wu, J., Sharma, R., Huang, T.: Analysis of uncertainty bounds due to quantization for three-dimensional position estimation using multiple cameras. Optical Engineering (1998)
19. Kim, W.S., Ansar, A.I., Steele, R.D., Steinke, R.C.: Performance analysis and validation of a stereo vision system. IEEE international conference on systems, man and cybernetics (2005)
20. Tarabanis, K., Tsai, R.Y., Allen, P.K.: Analytical characterization of the feature detectability constraints resolution, focus, and field-of-view for vision sensor planning. CVGIP: Image understanding **59**(3) (1994) 340–358
21. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers (1989)
22. Snyman, J.A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing (2005)
23. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. PRESENCE: Teleoperators and Virtual Environments **14**(4) (August 2005) 407–422