Camera Placement Considering Occlusion for Robust Motion Capture

Xing Chen, James Davis
Computer Graphics Laboratory, Stanford University

Abstract

In multi-camera tracking systems, camera placement can have a significant impact on the overall performance. In feature-based motion capture systems, degradation can come from two major sources, low image resolution and target occlusion. In order to achieve better tracking and automate the camera placement process, a quantitative metric to evaluate the quality of multi-camera configurations is needed.

We propose a quality metric that estimates the error caused by both image resolution and occlusion.. It includes a probabilistic occlusion model that reflects the dynamic self-occlusion of the target. Using this metric, we show the impact of occlusion on optimal camera pose by analyzing several camera configurations. Finally, we show camera placement examples that demonstrate how this metric can be applied toward the automatic design of more accurate and robust tracking systems.

1 Introduction

In designing a vision-based tracking system it is important to define a metric to measure the "quality" of a given camera configuration. Such a quality measure has several applications: first, by combining it with an optimization process we can automate the camera placement process and do better than a human designer, especially as the tracking environment gets more complex and the number of cameras increase; second, there are classes of applications where camera configurations change dynamically and some metric is needed to guide the automatic choice of best configuration. For example, a multi-target tracking system with multiple pan-tilt cameras might want to dynamically focus different subsets of cameras on each target. Other applications might require dynamic configuration due to bandwidth or processor power limitations, for instance in a system with hundreds of cameras, only a subset of the cameras can be active. In these situations it is crucial to have a quality metric so that the camera configuration that enables the best tracking performance can be found.

In a motion capture system, multiple cameras observe a target moving around in a working volume. Features on the target are identified in each image. Triangulation or disparity can be used to compute each feature's 3D position. In such a system, performance degradation can come from two major sources: (1) *low resolution* which results in poor feature identification; and (2) *occlusion* which results in failure to see the feature. Occlusion may be due to either the target itself or other objects in the scene. When not enough cameras see a feature, it is difficult or impossible to calculate its 3D position. In fact, the primary reason commercial motion capture systems consist of many cameras is to reduce occlusion, rather than to increase resolution or coverage. In order to achieve accurate and robust tracking, both occlusion and resolution must be considered. A quality metric for placing cameras should reflect the impact of both factors.

In this paper we propose a quality metric that accounts for both the resolution and occlusion characteristics of a camera configuration. Its target application is the automatic placement and control of cameras for motion capture systems. It can be used both to design static camera arrangements, and dynamically focus subsets of cameras on different targets in a multi-target tracking system.

The metric computes the uncertainty or error in the tracking system's ability to estimate the 3D positions of features. This uncertainty is caused by limited 2D image resolution, as well as occlusion due to the environment and/or the moving target itself. The error due to image resolution is computed by projecting the 2D image error of each camera into 3D space and measuring the size of the resulting 3D error volume. The uncertainty due to target occlusion is estimated by sampling the space of possible occluders and computing the probability that points are occluded.

The major contribution of this paper is a quality metric for multi-camera configurations that includes a probabilistic occlusion model. This metric allows cameras to be placed more robustly than previous resolution-only metrics. It models the target self-occlusion behavior that can be commonly found in feature-based motion capture systems. In addition, the use of sampling in the metric computation allows for easy adaptation to tracking scenarios with disparate occlusion characteristics. Lastly, we present simulation and analysis for some camera placement scenarios. These examples illustrate the usefulness of the metric, and provide insight on the location of "good places" to put cameras for accurate and robust tracking.

The rest of the paper is organized as follows. Section 2 describes previous work related to camera placement. Section 3 describes the metric we propose, focusing specifically on how dynamic occlusion is modeled. Section 4 shows simulation results that illustrate the impact of resolution and occlusion on optimal camera placement. Section 5 presents some camera arrangement tasks where this metric can be applied. Section 6 gives conclusion and future work.

2 Related work

The camera placement problem can be regarded as an extension to the well-known art-gallery problem [1]. Both problems have the goal of covering a space using a minimum number of cameras and the solutions are greatly affected by the visibility relationship between the sensor and target space. However, there are significant differences between these two problems. The art-gallery problem focuses on finding the theoretical lower-bounds on the number of guards for spaces of various (possibly very complex) shapes. Both the target space and the locations of guards are restricted to 2D. Additionally the visibility model is very simple—it assumes the guard has a 360-degree field of view (FOV) and there is no resolution degradation with viewing distance. The camera placement problem we consider is a practical problem and the camera has a more complex model which includes 3D projection, limited FOV and image resolution. Although the space to cover in practical camera placement is usually geometrically simple, there are usually constraints on allowable camera placement, such as ceilings and walls. In our work, the goal is not necessarily to find the absolute global optimum, but rather to enable the evaluation and comparison of the subset of potential solutions.

There is some previous work in automatic sensor planning in the area of robotic vision [2-5], motion planning [7], and image-based modeling [8]. To a large degree, the previous work shares our view of camera placement as an optimization problem, and that an important step toward automatic solutions is the construction of a quality metric to evaluate various camera configurations. However, the problem domain and goals of the previous work are quite different from ours. Their target is usually a static object whose geometry is assumed to be known a priori, and the task is to find a viewpoint or a minimum number of viewpoints that exposes the features of interest on the target as much as possible. There is usually only one camera in the system and the quality metric includes the target geometry explicitly. The camera placement problem for motion capture differs because the target is moving and its geometry and motion is not known a priori. In addition, there are multiple cameras working together in the system and system performance is affected by their relative pose.

A few researchers have proposed uncertainty analysis for placing multiple cameras. Olague and Mohr [9] approximated the projective transformation of a camera using Taylor expansion, and used a scalar function of the covariance matrix as the uncertainty measure. Wu et. al. [10] proposed a computational technique to estimate the 3D uncertainty volume by fitting an ellipsoid to intersection of projected error pyramids Both papers consider limited image resolution as the only cause of 3D uncertainty. However, occlusion is frequently present in feature-based motion tracking systems and is sometimes the dominant source of error. The quality metric presented in this paper estimates the 3D uncertainty caused by both occlusion and image resolution. This metric is able to model the dynamic and probabilistic characteristics of a moving target, as well as the mutual resolution compensation among multiple cameras.

3. Construction of the quality metric

Given some camera configuration (i.e. the focal length, position, and orientation of multiple cameras), the quality metric is a mapping from this configuration to a real number, with respect to some target space. Since resolution and occlusion characteristics vary over the target space, we define a spatially dependant quality metric. By sampling the target space and aggregating the per-point metric, the overall quality can be computed. For example if the target space is a room, then we calculate the uncertainty at each point in the room, and aggregate these values.

The next step is to define the quality at a given point in the target space. As described in the previous section, poor image resolution and occlusion cause error and uncertainty in determining the 3D position of a point. It follows that we can use this 3D uncertainty as the per-point quality metric. For any given point, we

estimate uncertainty as a combination of two factors. We estimate the error due to poor image resolution by projecting the 2D image error region into 3D. We estimate the uncertainty due to occlusion by simulating an occluder at a sampled set of locations and calculating the visibility from all cameras. In the following subsections we describe in detail how these estimates are computed and combined.

3.1 Resolution

As is well known in the vision community, one major source of error in determining 3D position is limited 2D image resolution. Most vision-based tracking system perform some sort of triangulation on rays from two or more cameras. Some target feature is detected on the image and the ray defined by its 2D location and the camera center is back-projected into 3D space. The intersection point of rays from multiple cameras defined by the same feature is the 3D location of the feature. However, one can only determine the 2D location of a feature on an image to a certain precision, as limited by the feature detection process and the image resolution. Thus each 2D observation from a camera gives rise to a cone of rays with some probability density rather than a single ray. This cone can be approximated as pyramid of rays. As shown in Figure 1, the intersection of ray pyramids from multiple cameras form a 3D volume rather than a point. The actual target point has high probability of being located anywhere in the error volume. The bigger the error volume, the higher the uncertainty in the point's actual position.

It should be noted that the size of the uncertainty volume is not isotropic, it varies with direction. In order to estimate the size of the volume, we measure its dimension in a set of sampled directions and then aggregate the error in all directions (Figure 2).

To compute the 3D error for a given direction, we first compute the 3D error in that direction for each camera, as shown in Figure 2. When there are multiple cameras, the cameras having better resolution (smaller |PR|) will cut the error volume and make it smaller. Thus, the combined uncertainty of all the cameras is the minimum of all.

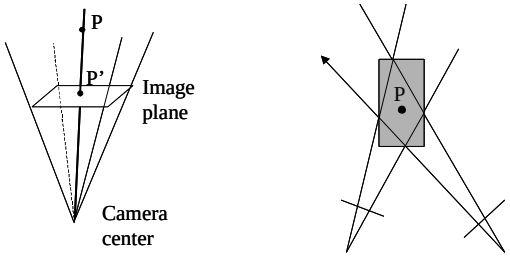

Figure 1 Left: Camera projection pyramid with a 3D square error region; Right: Intersection of the error regions of two cameras.

3.2 Static occlusion

Static occlusion is caused by static objects in the scene that are known a priori and time-invariant. Due to these occlusions, a point **P** is not visible from certain cameras. These occlusions can be included in the above formulation: Whenever we project **P** onto the image plane of a camera, we need to perform two visibility tests. One is to check whether **P** is within the camera's field of view; another is to see if any static object in the scene is between point P and the optical center of camera **C**. If **P** is not visible due to either cause, its 3D uncertainty is infinite.

3.3 Dynamic occlusion

Dynamic occlusion occurs when a target point is not visible from a camera due to occlusion by the target itself or other objects in the scene. The challenge to computing the error caused by this type of occlusion is that we do not know exactly where the target or occluder will be at any time. Without knowledge of target and occluder location it is impossible to arrange cameras such that the target is guaranteed to be visible.

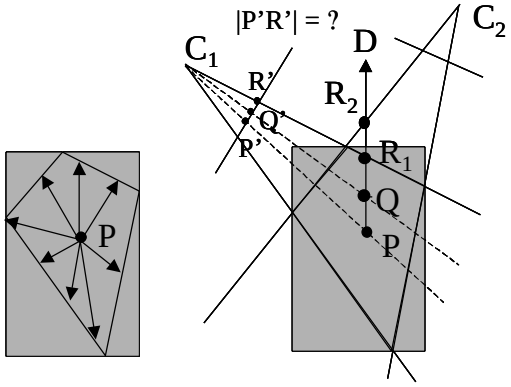

Figure 2 Left: We compute the per-point error volume by sampling various directions; Right: To compute the size of the 3D error volume at P in direction D: (1) Project the unit vector PQ in direction D to the image plane to get P'Q' (2) scale P'Q' by image plane error ? to R'; (3) Back-project P'R' in 3D to get PR. (4) For multiple cameras, pick the smallest PR.

We address the dynamic occlusion problem by first assuming perfect knowledge and then making several approximations that allow a probabilistic model to be developed. Assume first that we know exactly the geometric model of a target object and the path that the object takes during a tracking session. We can evaluate the error caused by dynamic occlusion precisely, because we can simulate the entire process and count exactly how many feature points are occluded at any time. Of course, this method is not very useful for designing a real tracking system. In reality, the path that a target takes could vary greatly and camera configurations optimized for one specific path may be poor for other paths. So the next question is, how can we find a camera configuration that avoids occlusion for all possible paths?

Instead of simulating a precise path when calculating occlusion, we can take a probabilistic approach. Though we may not know exactly where the "occluders" will be, we may have some idea as to how likely they are at certain positions and orientations. In other words, we have a probability distribution for the occluder. We can generate possible occluders by drawing samples from the distribution. For each possible occluder, we calculate how many cameras are obstructed from observing the target point **P**. The result for each occluder that is sampled from the distribution are aggregated into a single estimate of occlusion characteristics. The more often occlusion occurs and the more cameras that get occluded, the greater the 3D uncertainty. Sampling the space of all possible occluders gives us an occlusion metric for a particular camera configuration.

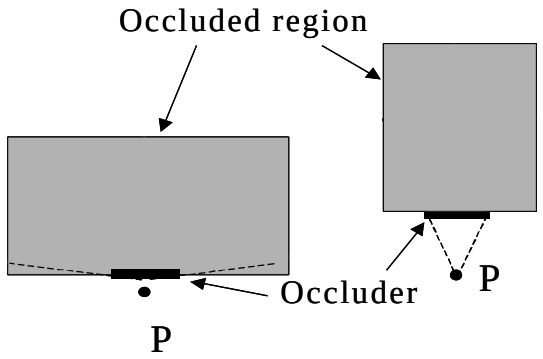

Figure 3 The size and orientation of an occluder determine the shape of the occluded region. A conservative estimate of occluded region can be made by assuming the occluder is very close to point P. In this case an entire hemisphere of camera locations will not be able to see point P.

Many occluder distribution models are possible. We derive a relatively simple distribution function by considering a motion tracking system where reflective point features are attached to the human body. A feature can be occluded either by the part of body it's attached to, or by another part of body from a distance (figure 3). Obviously in the most general case, the visibility of a point depends on the position, orientation and size of the occluder. However, the worst case occlusion occurs when the occluder is very near the point. In this location a whole hemisphere is occluded. To obtain a conservative estimate of occlusion, we need only simulate this worst case occluder pose. When the occluder is in this position, very near the target point, the size of the occluder does not matter. Therefore, to generate a conservative estimate of occlusion at point **P**, we can generate planar occluders through **P**. The orientation of these occluders is determined by some probability distribution.

To summarize, the 3D uncertainty due to dynamic occlusion of point **P** is determined by aggregating the number of visible cameras for many occluders, sampled from the orientation probability density function.

3.4 Combined quality metric

We can now combine resolution and occlusion to define the overall metric. Given a camera configuration $\Omega = \{C_1, C_2, \ldots C_N\}$, the overall quality metric *E.* is defined as the *k*-norm of the per-point quality, where the per-point quality is the weighted sum of 3D error due to the resolution, and dynamic occlusion:

$$E = \sum_{\text{all target point } \mathbf{P}} \left(w_{\mathbf{P}} \left(w_{res} E_{res}(\mathbf{P}) + w_{oc} E_{oc}(\mathbf{P}) \right) \right)^k$$

where $w_{\mathbf{P}}$ is the importance weighting of point **P**, and E_{res} and E_{oc} are the per-point 3D error caused by resolution and dynamic occlusion, respectively. The wres and woc terms are their relative weightings respectively and $(\cdot)^k$ denotes *k*-norm. The per-point error caused by image resolution is defined as:

$$E_{res}(\mathbf{P}) = \sum_{\text{directions } \mathbf{D}} \left(\min_{\text{all cameras } \mathbf{C}} \left(L(\mathbf{C}, \mathbf{D}, \mathbf{P}) \right) \right)^k$$

where L(C,D,P) is the 3D error of camera C in direction D at target P; The per-point uncertainty due to dynamic occlusion is:

$$E_{oc}(\mathbf{P}) = \sum_{\text{all occluders } oc} \left(V(\mathbf{P}, oc, \Omega) \right)^k$$

where $V(\mathbf{P}, oc, \Omega)$ measures the "occludability" of point P given the occluder *oc* and camera configuration

. To be consistent with the fact that larger values indicates worse quality, V is set to 1 if fewer than two cameras see the point (i.e. the "bad" cases), and it is set to 0 if two or more cameras sees the point.

It should be noted that random samples of target points, directions and occluders are used in the computation of the quality metric and they are generated according to some probability distribution. The use of sampling and the distribution allows us to encode further application-specific knowledge into the quality metric. For example, one might know a priori that certain parts of the target spaces needs more resolution, or that the occluders tend to be in certain directions more often than others. Even with a simple uniform distribution, the use of a probabilistic approach allows us to model the dynamic and unpredictable nature of motion tracking tasks, providing us with a general estimate of the uncertainty in the reconstruction of 3D positions.

3.5 Practical Issues

The quality metric defined above is the theoretical quantity we would like to use to measure camera configurations. However in practice there are situations in which we need to use a slightly modified version of the original definition. For example, the choice of the k-norm with which to combine the samples usually depends on the application: If one wants to evaluate the worst-case, one should use the infinity-norm, which is the max() function. If one wants to evaluate the average quality, 1-norm or 2-norm can be used. If one wants to combine this metric with some generic optimizer to obtain an optimal solution, it should be noted that the infinity-norm results in flat regions on the function landscape. Therefore, optimizers which rely on gradient in the landscape will not work optimally. We have found that a 2-norm works well in practice.

4. Impact of occlusion and resolution

The occlusion and resolution terms of our quality metric are often opposed. This has a significant impact on the optimal pose and field of view of camera configurations. In order to understand the impact of adding an occlusion model to our coverage quality metric, we consider several simple examples. These examples can provide some intuition into the tradeoff between resolution and robustness. In each example we consider a metric determined entirely by occlusion *or* resolution, and analyze the optimal configuration under these conditions. We find that there is a tradeoff between these two terms, which must be balanced. Unless specifically stated, the probability distribution we used for dynamic occluder directions is uniform.

The first example shows the impact of occlusion on camera placement pattern. Cameras are constrained to move on an outer sphere, and both have a fixed field of view (FOV) that is enough to cover the target sphere. Because of symmetry, the parameter that really independently affects the quality is the relative angle *theta* between the two cameras (Figure 6). We plot the quality for thetas from 0 to 180 degree. with and without considering occlusion. If we consider resolution only, the best configuration occurs when the two cameras are oriented perpendicular to each other, since the "good" direction of the 3D error of one camera cancels the "bad" direction of the other. However, when dynamic occlusion is considered to be dominant, the best configuration is with the two cameras facing opposite to each other, because this increases the probability that there is at least one camera seeing the target space for occluders with arbitrary orientations.

The second example illustrates the impact of

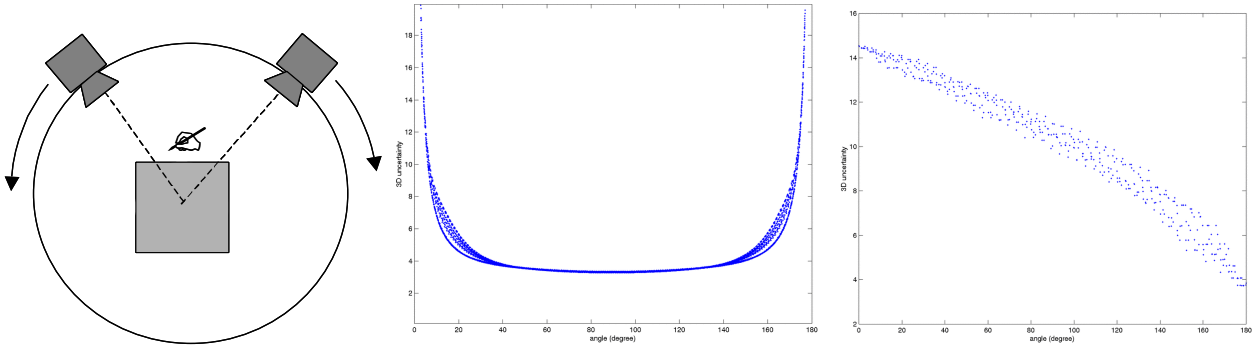

Figure 4 Left: Two cameras constrained to move on an outer sphere try to cover a inner spherical target space; the angle with which they intersect the sphere center is ?. The occlusion and resolution metrics are minimized at different angles. Center: 3D uncertainty vs. ? considering resolution only . Right: 3D uncertainty vs. ? considering occlusion only. (? is from 0 to 180 degrees.)

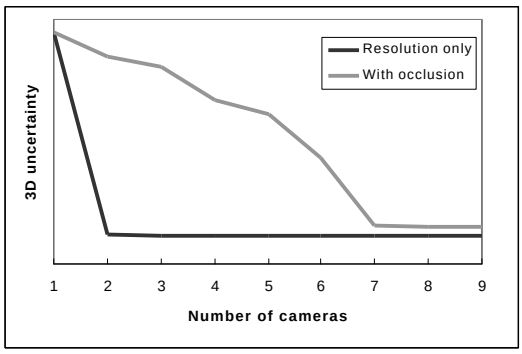

Figure 5 3D uncertainty vs. number of cameras. Error due to imager resolution is nearly minimized by only two cameras. Many more cameras are needed to robust insure against occlusion.

occlusion on the number of cameras needed to cover a certain space. As in the first example, the task is to cover a spherical space and cameras are constrained to stay on an outer sphere with fixed FOV. We used an evolutionary algorithm based optimizer [10] to find the best quality achievable for a given number of cameras. Figure 5 plots the best quality vs. the number of cameras. It can been seen that, if resolution is the only consideration, the best quality doesn't improve much after the space is covered by two cameras. However, if dynamic occlusion is considered, adding more cameras continues to improve the quality.

The next example demonstrates the impact of occlusion on the setting of camera focal length (or equivalently, the FOV). Two typical extreme configurations are considered (Figure 6). One configuration has two clusters of cameras. Each cluster

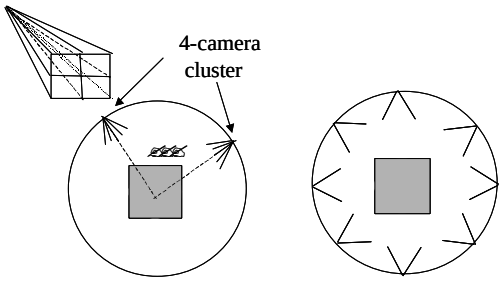

Figure 6 Left: Two camera clusters form 90 degrees, each cluster consists of 4 narrow angle cameras, equivalent to a single high resolution camera; Right: All cameras are wide-angle and evenly spread around the sphere in 3D. The diagram is a visualization only, the cameras are not coplanar. The resolution metric prefers the former case, while the occlusion metric prefers the latter.

consists of 4 narrow-angle cameras, each covering only one-quarter of the target space. The two clusters are placed so that they are 90 degrees apart on the outer sphere. This configuration is analogous to using two perpendicular high-resolution cameras. The other configuration consists of 8 cameras spread evenly around the sphere. In this configuration each camera's FOV is wide enough to cover the whole sphere. If no occlusion is considered, the narrow-angle solution gives the best overall resolution. If occlusion is considered, the wide-angle configuration is better since it is more robust. In the wide angle configuration each point is observed by eight cameras rather than two. This example shows that resolution improvement and occlusion reduction have conflicting requirements on the FOV of cameras. Thus, given limited camera resources, trade-offs have to be made between the importance of resolution and robustness depending on the specific application.

5 Practical camera placement examples

In this section we show an example that demonstrate how one can apply the proposed metric to automate the camera placement process.

We combine the metric with an evolutionary-algorithm-based optimizer [10]. in order to automatically find a placement solution. This optimizer is chosen because it is robust when function landscapes are discontinuous and also because of its ability to allow constraints on the variables we want to optimize.

First we try to place three cameras of fixed field of view to cover a square floor, considering resolution only. The cameras are constrained to be on the ceiling. As shown in Fig. 7, one of the configurations that the optimizer generated that gives good overall resolution put two cameras far enough to just cover the whole floor, with the relative orientation between them approximately 90 degrees. What's interesting is the pose of the 3rd camera. In the solution shown, it is placed on the ceiling looking straight down and covering the "back" region relative to the other two cameras. This makes sense because even though the two perpendicular cameras already cancel out the "bad" direction of each other and provide more-or-less even resolution in all directions, the region that is farther from the camera has worse resolution. To compensate, the 3rd camera is placed closest to that region to provide higher resolution.

In contrast, if our goal is to optimize for the least probability of occlusion, by applying the metric with a high occlusion weighting, the placement that the optimizer found is shown in Figure 8. As can be seen, this arrangement has all three cameras backed off so that each of them cover the whole square. In other

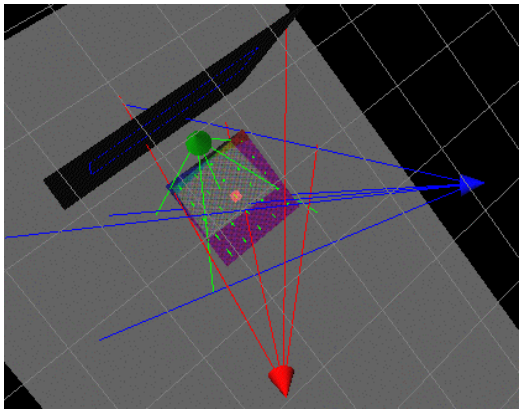

Figure 8 Three cameras constrained on the ceiling to cover a square floor, considering resolution only. Colored "shadows" are drawn and superimposed on the floor to show the cameras' coverage. The purple part on the floor is the result of both red and blue camera.

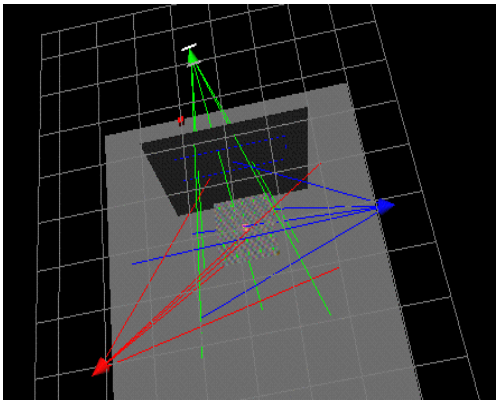

Figure 7 Three cameras constrained on the ceiling to cover a square floor, considering occlusion. The cameras are backed off and evenly distributed around the patch. Every point on the patch are covered by all three cameras.

words, every point on the floor is covered with all three cameras. This matches our intuition that the more camera seeing the target the less often occlusion might happen.

It is interesting to note that, we have asked people in our lab to propose camera configurations. No one proposed a design simlar to the optimzer's optimal resolution placement, however once they saw the computer generated design, they agree that it makes sense. This illustrates the necessity of a quantitative approach to camera placement.

6 Conclusion and future work

We have proposed a metric to evaluate the quality of a multi-camera configuration, in terms of both resolution and occlusion. It includes a probabilistic occlusion model that reflects the target self-occlusion behavior which is commonly found in feature-based motion tracking systems. The computation of the model is sample-based, making it easily adaptable to applications with various occlusion characteristics. In addition, a metric that takes into account both resolution and occlusion allows the previously ad hoc process of placing multiple cameras to become an automated quantitative process. The inclusion of a probabilistic based occlusion model makes it especially useful for designing motion capture systems and other occlusion-dominant tracking systems. In simple situations, it should do as well as a human designer's intuitive solution. In more complex situations, it enables the automatic generation of solutions that may take a human designer a long time to find. Moreover, it enables the finding of optimal camera configurations when the placement solution *has* to be automatically generated. We have demonstrated how this metric can help us in understanding the design trade-offs of camera placement. Additionally, camera configurations can be designed automatically or semi-automatically for tracking systems with various requirement and constraints.

The metric defined in the paper steps toward solving the general problem of optimal sensor placement for multi-camera vision-based systems. It provides us with a quantitative tool for evaluating various camera configurations. We plan to analyze various camera configurations using this metric and to synthesize a set of widely applicable guidelines for systematically placing cameras. This will contribute to the theoretical understanding of the impact of camera placement, and enable the building of "smart tools" to aid the design of practical motion tracking systems.

References

[1] J. O'rourke, *Art gallery theorems and algorithms*: Oxford University Press, 1987.

[2] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 86-104, 1995.

[3] K. A. Tarabanis, R. Y. Tsai, and P. K. Allen, "The MVP sensor planning system for robotic vision tasks," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 72-85, 1995.

[4] B. Triggs and C. Laugier, “Automatic camera placement for robot vision tasks,” in *Proceedings of 1995 IEEE International Conference on Robotics and Automation (Cat. No.95CH3461-1)*. New York, NY, USA: Ieee, 1995, pp. (xxxxviii+xxxvi+3166).

[5] S. Yi, R. M. Haralick, and L. G. Shapiro, “Automatic sensor and light source positioning for machine vision,” in *Proceedings. 10th International Conference on Pattern Recognition (Cat. No.90CH2898-5)*. Los Alamitos, CA, USA: IEEE Comput. Soc. Press, 1990, pp. (xxxi+xxv+1676).

[6] J.-C. Latombe, *Robot Motion Planning*: Kluwer Academic Publishers, 1991.

[7] S. Fleishman, D. Cohen-Or, and D. Lischinski, “Automatic camera placement for image-based modeling,” in *Proceedings. Seventh Pacific Conference on Computer Graphics and Applications (Cat. No.PR00293)*. Los Alamitos, CA, USA: IEEE Comput. Soc, 1999, pp. xii+331.

[8] G. Olague, R. Mohr, S. Venkatesh, and B. C. Lovell, “Optimal camera placement to obtain accurate 3D point positions,” in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, A. K. Jain, Ed. Los Alamitos, CA, USA: IEEE Comput. Soc, 1998, pp. xlvii+1867.

[9] J. J. Wu, R. Sharma, and T. S. Huang, “Analysis of uncertainty bounds due to quantization for three-dimensional position estimation using multiple cameras,” *Optical Engineering*, vol. 37, pp. 280-92, 1998.

[10] J. Wakunda and A. Zell, “EvA: a tool for optimization with evolutionary algorithms,” in *Proceedings. 23rd Euromicro Conference: New Frontiers of Information Technology (Cat. No.97TB100167)*. Los Alamitos, CA, USA: IEEE Comput. Soc, 1997, pp. xviii+708.