A Stochastic Quality Metric for Optimal Control of Active Camera Network Configurations for 3D Computer Vision Tasks

Adrian Ilie, Greg Welch and Marc Macenko

University of North Carolina at Chapel Hill

Abstract. We present a stochastic state-space quality metric for use in controlling active camera networks aimed at 3D vision tasks such as surveillance, motion tracking, and 3D shape/appearance reconstruction. Specifically, the metric provides an estimate of the aggregate steady-state uncertainty of the 3D resolution of the objects of interest, as a function of camera parameters such as pan, tilt, and zoom. The use of stochastic state-space models for the quality metric results in the ability to model and accommodate virtually all traditional quality factors, such as visibility, field of view, occlusion, resolution, surface normals, image contrast, focus, and depth of field. In addition, the stochastic state-space approach naturally addresses camera networks that are aided by other sensing modalities. We begin by surveying the traditional quality factors. We then present our new quality metric, aided by some background in the relevant stochastic state-space models, and an evaluation strategy that scales the computation of our metric to allow its use in a real-time active camera network system. Finally we present some simulation results that illustrate the incorporation of some traditional quality factors, and the use of our metric and evaluation strategy for some simulated scenes containing multiple objects of interest.

1 Introduction

Active camera networks have traditionally been used in surveillance, but are also emerging as a viable means to capture shape and appearance over wide areas using a limited number of cameras. 3D vision applications such as tracking, motion capture, and 3D reconstruction (shape and appearance) have specific requirements for how the PTZ cameras would be actively configured. For example, stereo reconstruction usually performs best with pairs or triples of cameras that have an appropriate baseline and are pointing in the same direction, while visual hull approaches usually favor placing the cameras uniformly around the object of interest. Space carving approaches typically surround the object but try to keep the views of nearby cameras as similar as possible, so that the measures of voxel consistency are more likely to be accurate.

The real-time management of a PTZ camera network for use in such applications requires both a *quality metric* for comparing configurations, and a *management component* that generates candidate configurations and uses the quality metric to compare them. We focus here on the former—a stochastic quality metric that reflects the system's ability to resolve object features directly in the 3D working volume. We compute

the steady-state uncertainty P^{∞} in the system's ability to resolve a set of 3D points $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p\}$ associated with the objects of interest, as a function of the camera parameters. The points could be vertices of models of the objects, vertices on a surrogate representative object such as a sphere for an object or a cylinder for a person, or simply a sparse set of points in the regions where the objects are expected to be. Specifically, at each point $\bar{x}_i$ we estimate:

$$P_i^{\infty} = \lim_{t \to \infty} \mathsf{E}\left\{ (\bar{x}_i(t) - \tilde{x}_i(t))(\bar{x}_i(t) - \tilde{x}_i(t))^{\mathsf{T}} \right\}, \qquad 1 \leq i \leq p \tag{1}$$

where $\bar{x}_i$ and $\tilde{x}_i$ represent the *true* and *estimated* states (respectively) at point i, and E denotes statistical expectation. We aggregate the P_i^{∞} estimates into a single scalar number to determine a cost for each candidate camera network configuration.

Rather than estimating $\bar{x}_i$ or $\tilde{x}_i$, we estimate P_i^{∞} directly from state-space models of the system and stochastic estimates of the various noise sources. The use of stochastic state-space models for the quality metric results in the ability to model and accommodate virtually all traditional quality factors, such as visibility, field of view, occlusion, resolution, surface normals, and depth of field. In addition, the stochastic state-space approach naturally addresses camera networks that are aided by other sensing modalities. Because we are estimating the uncertainty P in the steady state, we refer to the metric as the *PSS metric*.

2 Related Work

In this section we briefly mention a few of the many related research efforts, grouped by their application domain. We selected research that mentions quality factors affecting camera performance. We then summarize the quality factors and show how they can be implemented with the PSS metric.

Camera Placement. Given knowledge about the cameras, the environment and the required task, determine the camera configurations (placement and other parameters) that best satisfy the task requirements. Methods include: generate and test [1], synthesis [2, 3], simulation [4, 5], expert systems [6] and fuzzy logic [7]. A comprehensive review of these methods can be found in [8]. Example quality factors include extrinsic camera parameters, optical camera parameters (pixel size, aperture, focal length, exposure time, gain, hue, saturation), camera models (such as perspective projection), object models (polygonal description, motion), and task constraints (visibility, field of view, focus, pixel resolution, incidence angle). Denzler et al. [9] present an information theoretic framework and derive a metric based on the uncertainty in the state estimation process. Wu et al. [10] estimate the uncertainty in multi-camera stereo using the 2D quantization error. Chen [11] proposes a metric that takes into account resolution and occlusion for motion capture. Olague and Mohr [12] analyze the propagation of uncertainty in 3D reconstruction, and use the maximum value of the covariance matrix as an optimization criterion. The use of steady-state uncertainty as a performance metric was first introduced in [13] as a means for optimizing the design of multi-sensor systems.

Camera Selection. Given a set of cameras, select the optimal subset that provides the best performance. Selection is usually done in order to save power in wireless sensor

networks. The authors of [14] optimize for the minimum error of the best linear position estimate in the presence of occlusion. Isler and Bajcsy [15] present an approximation algorithm which selects a given number of cameras and guarantees that the quality of the localization is within a factor two of the optimal choice.

View Planning. Compute the sequence of sensor poses that best satisfy a quality criterion, while sometimes also minimizing the number of poses and the length of the path between them. Scott et al. [16] present a comprehensive survey of view planning methods. Chen and Li [17] present an approach based on a genetic algorithm that takes into account factors such as visibility, field of view, resolution, depth of field, overlap, occlusion, contrast, and reachability.

Surveillance. Given a camera network, control the cameras to track targets and acquire 2D images for other tasks. Bodor et al. [18] compute the optimal camera poses for maximum task observability given a distribution of possible target trajectories. Mittal and Davis [19] compute the probability of visibility in the presence of dynamic occluders, and optimize using cost functions such as the number of cameras, occlusion probabilities, or number of targets in a particular region of interest under constraints such as field of view, fixed occluders, resolution, and viewing angle. Yous et al. [20] propose a camera assignment scheme based on the analysis of a coarse 3D shape produced in a preprocessing step to control multiple Pan/Tilt cameras for 3D video of a moving object.

Table 1 lists the most common factors encountered in the various related domains, along with examples of previous work where they appear. In addition we indicate where/how they can be implemented using our PSS approach: MM indicates the *measurement model* (described in Section 3.1), and AG indicates the *aggregation component* (described in Section 3.3). These and other components of the PSS metric are described in detail in Section 3. In Section 4 we show some examples of implementing a few of the factors in experimental simulations.

Table 1. Quality factors and their mapping in our approach

Factor	Previous work	PSS metric component
Image resolution	[13, 18, 11, 17, 2, 4, 6, 19, 12, 7, 5, 3, 10, 20]	MM
Brightness, Contrast	[17, 1]	MM
Focus, Depth of field	[17, 2, 6, 19, 12, 7, 3, 10, 20]	MM
Field of view	[13, 18, 17, 2, 9, 6, 19, 12, 7, 5, 3, 20]	MM, AG
Occlusion	[13, 11, 17, 2, 4, 6, 19, 12, 7, 5, 3, 20]	MM, AG
Object distance	[13, 18, 6, 10]	MM
Incidence angle	[18, 6, 19, 12, 7, 20]	MM

3 The PSS Quality Metric

Being stochastic in nature, the PSS quality metric facilitates the incorporation of a wide variety of heterogeneous factors and sensing modalities, and the sampling can be dy-

namically adapted to meet timing goals. Before presenting the metric, we provide a brief review of state-space models and the stochastic aspects that are needed to understand it.

3.1 State-Space Models

State space representations use vectors and matrices to model a physical system as a set of input, output and state variables related by first-order differential equations [21]. Using this notation, we represent the *system state* of a point of interest by an n-dimensional state vector $\bar{x}$. One can model the motion of a point over time δt as a first-order dynamic process:

$$\bar{x}(t) = A\bar{x}(t-\delta t) + \bar{w}(t-\delta t) \tag{2}$$

where A is an $n \times n$ *state transition matrix*, and $\bar{w}$ is a zero-mean and spectrally white $n \times 1$ random signal known as the *process noise*. In general one can model an m-dimensional device measurement $\bar{z}$ at time t as:

$$\bar{z}(t) = H\bar{x}(t) + \bar{v}(t) \tag{3}$$

where H is an $m \times n$ matrix relating the n-dimensional state to the m-dimensional measurement, and $\bar{v}$ represents zero-mean, white *measurement noise*, presumed to be uncorrelated with $\bar{w}$. In practice the actual noise signals $\bar{w}$ and $\bar{v}$ are not known or even estimated. Instead we assume normal distributions $\bar{w} \sim \mathsf{N}(0,Q)$ and $\bar{v} \sim \mathsf{N}(0,R)$ with corresponding process and measurement noise covariance matrices Q and R.

Equations (2) and (3) serve as the basis for most stochastic estimation methods and are often referred to as the *process model* and the *measurement model*, respectively. The interested reader can find more details in [13, 21]. A classification of various linear process models and corresponding state transition matrices encountered in practice can be found in [13]. If the process model is non-linear, it can be linearized about the point by using the Jacobian of the function (evaluated at that point) in place of matrix A in Equation (2).

For cameras, the measurements are 2D image coordinates, i.e. $m = 2$. We model the relationship between $\bar{x}$ and $\bar{z}$ using a measurement function that implements a pinhole camera model, initialized for each camera using parameters obtained through geometric calibration. The measurement function also deals with view frustum culling and occlusions. A more complicated model could also include factors such as radial distortion. As this function is non-linear, we replace H with the Jacobian of the function. The magnitude and shape of the elements of the measurement noise covariance matrix R reflect the expected camera measurement noise. The effects of many of the quality factors found in Section 2 can be incorporated into R, as shown in a few practical examples in Section 4.

When additional sensing modalities are employed, their contributions can be accounted for with sensor-specific measurement functions. Allen [13] presents an example hybrid system, where a multi-camera acquisition system is combined with an acoustic tracker.

3.2 Metric Computation

As described earlier, our PSS metric aims to evaluate the uncertainty with which we can resolve a set of 3D points $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p\}$ associated with the objects of interest, given a particular *configuration* (set of values for pan, tilt and zoom) of an active camera network. The basic approach for computing the PSS metric is to compute the asymptotic error covariances at the 3D points associated with the objects of interest, then aggregate the estimates into a single number. Specifically, at each point of interest $\bar{x}_i$ we estimate P_i^∞ as the solution to the Discrete Algebraic Riccati Equation (DARE) [21]:

$$P^\infty = AP^\infty A^\mathsf{T} + Q - AP^\infty H^\mathsf{T} \left(R + HP^\infty H^\mathsf{T}\right)^{-1} HP^\infty A^\mathsf{T}. \tag{4}$$

We use the MacFarlane–Potter–Fath Eigenstructure Method [21] to calculate the DARE solution P^∞ as follows. For each point of interest $\bar{x}_i$, we determine the set of c cameras from which the point is visible. If $c \geq 2$, we compute the corresponding measurement Jacobians $\{H_1, H_2, \ldots, H_c\}$ and the measurement noise covariances $\{R_1, R_2, \ldots, R_c\}$ to form a $2c \times n$ batch version $\widehat{H}$ and a $2c \times 2c$ block-diagonal batch version $\widehat{R}$:

$$\widehat{H} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_c \end{bmatrix}, \qquad \widehat{R} = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & R_c \end{bmatrix}. \tag{5}$$

Given the process model parameters A and Q, and the measurement model parameters $\widehat{H}$ and $\widehat{R}$, we next calculate the $2n \times 2n$ discrete-time Hamiltonian matrix Ψ as

$$\Psi = \begin{bmatrix} A + QA^{-\mathsf{T}}\widehat{H}^\mathsf{T}\widehat{R}^{-1}\widehat{H} & QA^{-\mathsf{T}} \\ A^{-\mathsf{T}}\widehat{H}^\mathsf{T}\widehat{R}^{-1}\widehat{H} & A^{-\mathsf{T}} \end{bmatrix}. \tag{6}$$

We then form

$$\begin{bmatrix} B \\ C \end{bmatrix} = [\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_n] \tag{7}$$

from the n characteristic eigenvectors $[\bar{e}_1, \bar{e}_2, \ldots, \bar{e}_n]$ of Ψ, and finally using B and C we compute the steady-state covariance as

$$P_i^\infty = BC^{-1}. \tag{8}$$

The computed steady-state covariance P_i^∞ is an $n \times n$ matrix. Allen [13] proposes several metrics that can be derived from it. We use the maximum eigenvalue (infinity norm), as suggested by Olague et al. in [12]. We compute the metric at each point $\bar{x}_i$ as the standard deviation σ_i^∞ corresponding to the maximum eigenvalue $\lambda(P_i^\infty)$ of the covariance matrix P_i^∞:

$$\sigma_i^\infty = \sqrt{\max_{j=1}^{n} \lambda_j(P_i^\infty)} \tag{9}$$

Note that the batch measurement Jacobian $\widehat{H}$ and measurement noise covariance $\widehat{R}$ in Equation (5) are only computed for cameras for which the point in question is visible, and only if the point is observable ($c \geq 2$). A more rigorous definition of *observability* and ways to test whether it is satisfied can be found in [21]. The metric value at

unobservable points can be arbitrarily set to an upper bound, such as the size of the environment. More conservative upper bounds can be estimated from the metric value at a previous time instant or by applying heuristics to constrain the bounding volume.

3.3 Aggregation

The aggregation of the metric values σ_i^∞ derived from each P_i^∞ corresponding to each observable point $\bar{x}_i$ into a single number is application-dependent, and can be done separately for each object of interest or globally. The infinity-norm can be used to compute a worst-case value, but it usually leads to flat regions in the function that are ill-suited for gradient-based optimizers. The 1-norm or the 2-norm can be used to compute an average value that works in most situations. More complex aggregation functions can be designed for special cases where, for example, the system designer wants to differentiate between having very low uncertainty in some regions and very high uncertainty in others, versus having average uncertainty everywhere.

For the experiments in Section 4, we compute the PSS metric globally, and use the inverse of the standard deviation σ_i^∞ to get a better dynamic range in the plots. We do not attempt to estimate the value of the metric at unobservable points. Instead, we compute the PSS metric as the average inverse standard deviation $(\sigma_i^\infty)^{-1}$ for all the observable points $i = 1,2,...,v$, multiplied by the ratio between the number observable points v to the total number of points p:

$$PSS = \frac{\sum_{i=1}^{v}(\sigma_i^\infty)^{-1}}{v}\frac{v}{p} = \frac{\sum_{i=1}^{v}(\sigma_i^\infty)^{-1}}{p} \tag{10}$$

3.4 Evaluation Strategy Using Surrogate Models

An important component of our approach is deciding at what points $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p\}$ to evaluate P^∞. The fundamental difference when compared to approaches such as [13, 11] is that when computing our PSS metric we are only interested in the system performance in the regions where the objects of interest are, or will be moving. We take advantage of the fact that information about the trajectories of the objects of interest (and hence their points) can be made available from an external tracking system that runs in parallel with our metric computation, possibly using the same sensors. In our case, target trajectories are directly incorporated into our metric using the process models, instead of just taken into account for precomputing probabilistic occlusion as in [11].

In order to better organize the information available about the points associated with the objects of interest, we introduce the concept of *surrogate models*. The choice of surrogate models depends on the specific application and the information available from the external tracking system. Possible surrogate models include a single point, the eight points of a bounding box, points on a bounding sphere or cylinder, a dense point cloud, points on a convex hull computed separately [20], or the points of a reconstructed model from the previous time step. Using surrogate models allows us to evaluate the metric only where needed. We can also choose to evaluate the metric at a subset of the points, reducing the computation time as necessary for real-time evaluation.

4 Experimental Simulation Results

In this section we show examples of how some of the quality factors mentioned in Section 2 can be incorporated into $\widehat{R}$, the noise component of the measurement model (MM) of our performance metric. For each camera $k = 1,2,\ldots,c$ we compute the contributions of each factor f as 2×2 covariance matrices R_{kf}, and then sum them into a single R_k. We also illustrate the effect of varying some of the factor values while keeping all other factors constant. We show camera icons to illustrate the position, orientation and field of view of the cameras. We use a different color and numbered labels to indicate cameras that are active and dotted lines to indicate their viewing direction. We use tesselated surfaces as surrogate models for the objects, evaluate the metric at points spread evenly on the surfaces, and show the values on a color scale, interpolating between the sample points. Unobservable points (not visible from at least two cameras) are colored grey.

4.1 Static Scene

We evaluate various configurations of an active camera network with $c = 7$ simulated PTZ cameras placed along the ceiling of a $6 \times 6 \times 3$ meter room, covering two static objects of interest: a rectangle and a sphere which occludes a portion of the rectangle in the view of the cameras.

Object Distance and Incidence Angle. The number of photons that reach the camera sensor falls off with the square of the distance D from the object to the camera. If the surrogate models are accurate representations of the objects in the scene, we can also compute the incidence angle α between the ray from the camera to a point on the object, and the surface normal at that point. The accuracy of triangulation is generally proportional to the cosine of the incidence angle. Both factors appear on the diagonal of R_{k1} as follows:

$$R_{k1}(i,i) = D^2 cos(\alpha), \qquad i = 1,2. \tag{11}$$

Image Resolution. Camera intrinsic parameters include the image center, which depends on the sensor size. We use it when computing the measurement Jacobians H_k. We also use sensor size in the computation of the noise covariance matrices R_{k2}, as suggested in [13]:

$$R_{k2}(1,1) = (CCDnoise_k * w_k * cos(\theta))^2 \tag{12}$$

$$R_{k2}(2,2) = (CCDnoise_k * h_k * cos(\theta))^2, \tag{13}$$

where w_k and h_k are the width and height of the sensor in pixels, θ is the off-axis angle and *CCDnoise* is the measured sensor black-level noise.

Field of View and Occlusion. We compute the field of view by dividing its maximum value of 90° by a zoom multiplier: e.g., $\times 2 \rightarrow 45°$, and we use it when computing the measurement Jacobians H_k. We enforce the field of view constraint by clipping the points that project outside the area covered by the image sensor. At each point we also test for occlusion, either self-occlusion or occlusion by other objects. Figure 1 shows the effect of adjusting the fields of view of cameras 2, 4 and 6.

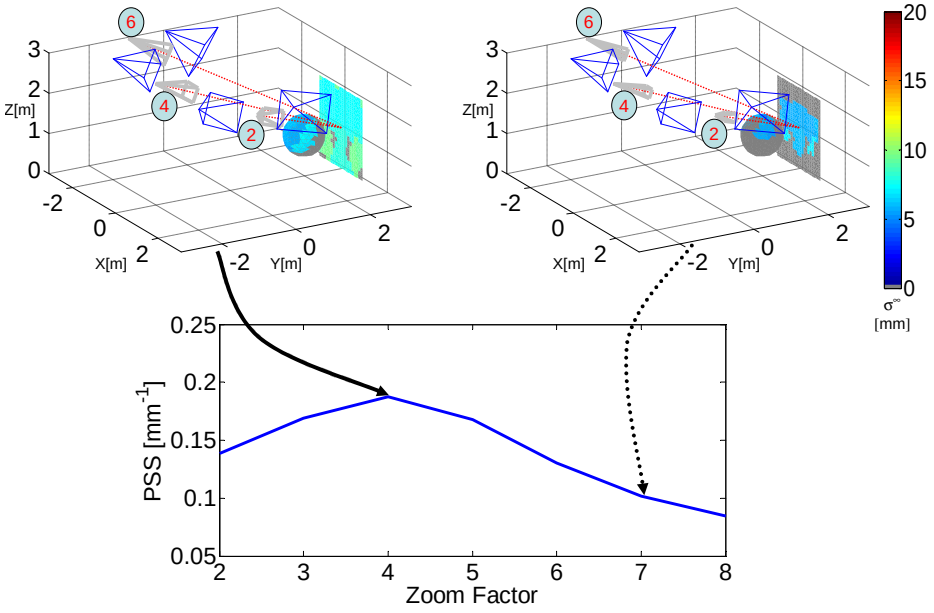

Fig. 1. Changing the field of view of 3 cameras. Bottom: plot of PSS vs. zoom factor (higher is better). PSS first increases due to increasing resolution, then decreases due to points outside the field of view being clipped. Top left: illustration of a configuration with wide field of view ($\times 4$ zoom). Notice also the effect of occlusion: the unobservable points on the back of the sphere and some regions of the rectangle are shown in grey. Top right: a narrow field of view ($\times 7$ zoom) leads to many unobservable points, shown in grey.

Focus Distance and Depth of Field. Points on the surrogate models can be out of focus due to factors such as lens aberrations and light diffraction. We incorporate this effect as a source of noise in the measurement model of each camera. We equate the amount of noise injected to the diameter of the circle of confusion cc_k, computed using the thin lens equation as:

$$cc_k = \frac{a_k}{ps_k} \frac{f_k}{d_k - f_k} \frac{|D - d_k|}{D}, \tag{14}$$

where a_k is the camera aperture diameter, f_k is the focal length of the lens, d_k is the focus distance and D is the distance from the camera to the point of the surrogate model where the uncertainty is evaluated. The term ps_k is the pixel size in real-world units, and is used as a scaling factor because cc_k is measured in pixels. We incorporate the result on the diagonal of the noise covariance matrix R_{k3} as follows:

$$R_{k3}(i,i) = cc_k^2, \qquad i = 1,2. \tag{15}$$

Figure 2 shows the effect of varying the focus distance d_k of 3 of the 7 cameras (2, 4 and 6). The focal planes are shown as semitransparent quadrilaterals.

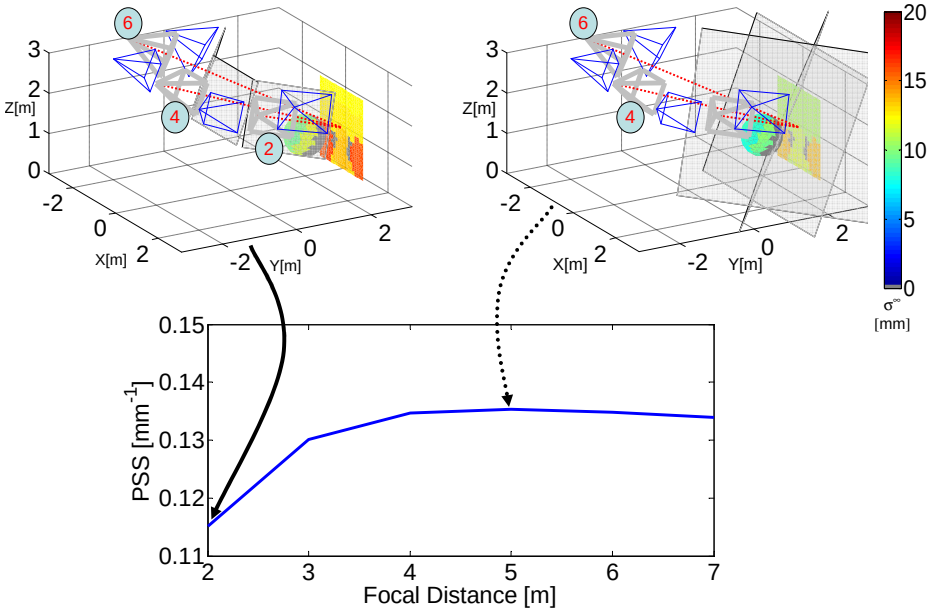

Fig. 2. Adjusting the focus distance. Bottom: plot of PSS vs. focus distance (higher is better). PSS first increases as more points come into focus, then starts to decrease as some points begin to get out of focus again. Top left: illustration of configuration with focus at 2 meters in front of the cameras. Both the sphere and the plane are out of focus. Top right: focus at 5 meters in front of the cameras.

Camera Selection. Selecting which cameras best cover a particular target is a fundamental component of *camera scheduling*, a complicated resource allocation process that assigns tasks to cameras in order to best satisfy the application requirements. Figure 3 shows the effect of trying to cover the objects of interest in our virtual room using only 2 of the 7 cameras, one of which is fixed. This experiment is similar to one encountered in [11], but we model occlusion differently.

4.2 Dynamic Scenes

When covering dynamic scenes, the full flexibility of PTZ cameras comes into play, as their pan and tilt settings are used to follow the objects of interest as they move, and other settings such as zoom and focus are used to ensure the best quality images. We evaluate the performance of an active camera network with $c = 4$ simulated PTZ cameras placed on the ceiling at the corners of a $10 \times 6 \times 3$ meter hallway, covering two human avatars that are passing each other while moving in opposite directions. When scheduling cameras in such a scenario, the choices are either to assign cameras to each avatar and have them follow it until it exits the scene, or to swap the camera assignments when the avatars cross paths. The scheduler also needs to take into account the physical

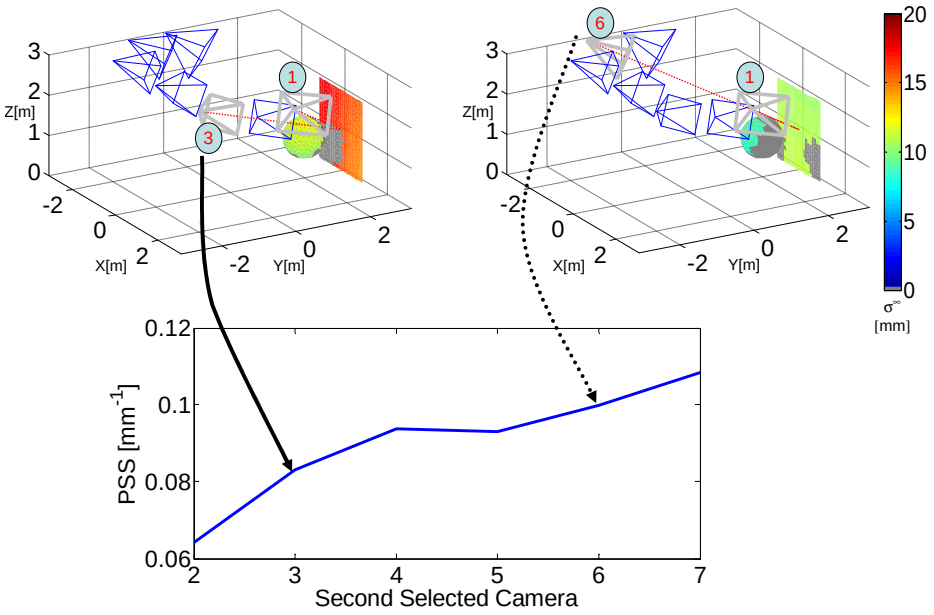

Fig. 3. Camera selection—choosing a second camera to pair with camera 1. Bottom: plot of PSS versus the alternative second camera (higher is better). PSS increases with the angle between the viewing directions of the two cameras, but also varies due to the number of points visible from both cameras. Top left: illustration of configuration using camera 3 (with camera 1). Top right: using camera 6 (with camera 1).

limitations of the camera settings. We exemplify these limitations with a maximum value for the zoom setting.

Figure 4 illustrates two alternative control strategies for the scenario where human avatars H1 and H2 cross paths moving in opposite directions: (1) cameras 1 and 2 continuously follow avatar H1, and cameras 3 and 4 continuously follow avatar H2; (2) the camera pairs swap assignments when the avatars cross paths. When swapping, the scheduler can further choose to assign each camera to follow either (2a) the furthest or (2b) the closest avatar from it. For example, when following (2b) the closest avatar, cameras 1 and 2 first follow H1 then swap to H2, and cameras 3 and 4 first follow H2 then swap to H1.

5 Conclusions

We presented a stochastic state-space quality metric for active camera networks aimed at 3D vision tasks such as surveillance, motion tracking, and 3D shape/appearance reconstruction. The PSS metric can incorporate virtually all traditional quality factors, and the metric exists in the 3D space of the objects of interest, which is the space where the quality ultimately matters for 3D vision tasks. The metric allows its sampling to be

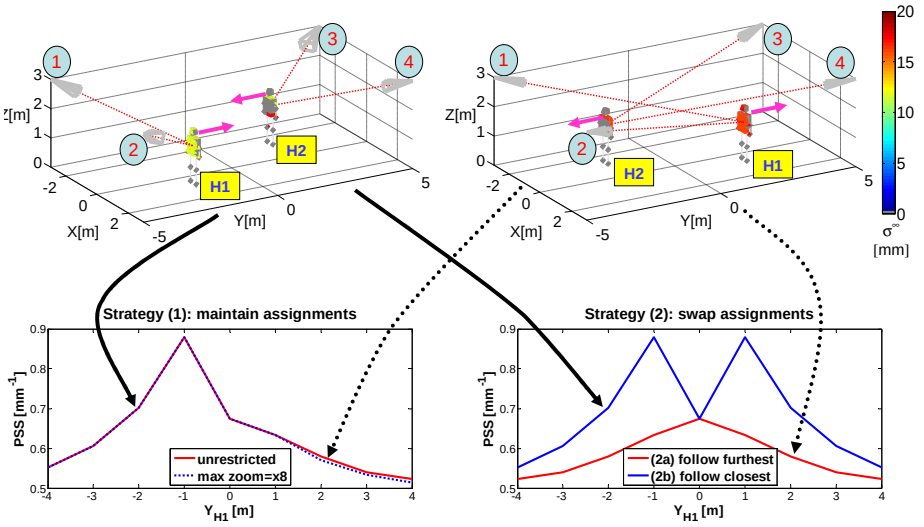

Fig. 4. Scheduling cameras in a dynamic scene. Top row: two moments in the sequence of strategy (1), for $Y_{H1} = -2$ and $Y_{H1} = 2$. The small arrows indicate the movement direction. Bottom left: plot of PSS vs. the position of H1 with cameras following their initial target–strategy (1). PSS first increases with the number of visible points, then starts to decrease with increasing distance. The spike at $Y_{H1} = -1$ is due to the fact that the fields of view of the cameras are wide enough to capture and contribute to the PSS values for points on the avatar they are not assigned to. Also, note the effect of limiting the maximum zoom to $\times 8$, shown as a dotted blue line: worse resolution leads to worse PSS values with increasing distance. Bottom right: plot of PSS vs. the position of H1 with cameras swapping coverage when the spheres pass each other-strategies (2a) and (2b). As expected, strategy (2b) would result in a higher PSS than (2a), mainly due to the size of the overlap between the fields of view of the camera pairs covering each avatar.

adjusted on line in real time to accommodate changes to the shapes of the objects of interest, or reductions in the metric computation time. If the platform performing the online optimization of the camera network configuration supports parallel computation, the PSS metric can be easily parallelized, as P^{∞} can be evaluated independently for each point. Finally, the PSS metric naturally allows incorporating inputs from other sensor modalities, making it applicable in multi-modal sensor fusion approaches.

In the near term we plan on concentrating on the aggregation component of the metric, for example adding some dynamically-adjustable weighting factors that can be used to encourage different coverage objectives. For example, one might want to prioritize the objects of interest, or to cover all of them, even at the expense of a worse average cost. Beyond the quality metric we are also developing the camera network management component which will generate candidate configurations and use the quality metric to compare them. We will be applying the management and quality metric components to wide-area marker-less motion capture and 3D reconstruction of people moving within the working volume of an active camera network testbed. Such a testbed should further help support the empirical evaluation of the PSS metric compared to other metrics.

References

1. Yi, S., Haralick, R.M., Shapiro, L.G.: Optimal sensor and light source positioning for machine vision. Computer Vision and Image Understanding **61** (1995) 122–137
2. Cowan, C.K.: Model-based synthesis of sensor location. In: IEEE International Conference on Robotics and Automation. Volume 2. (1988) 900–905
3. Tarabanis, K.A., Tsai, R.Y., Kaul, A.: Computing occlusion-free viewpoints. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996) 279–292
4. Fleishman, S., Cohen-Or, D., Lischinski, D.: Automatic camera placement for image-based modeling. Computer Graphics Forum **19** (2000) 101–110
5. State, A., Welch, G., Ilie, A.: An interactive camera placement and visibility simulator for image-based vr applications. In: Engineering Reality of Virtual Reality 2006 (3D Imaging, Interaction, and Measurement; IS&T/SPIE 18th Annual Symposium on Electronic Imaging Science and Technology), San Jose, CA (2006)
6. Mason, S.: Expert system-based design of close-range photogrammetric networks. ISPRS Journal of Photogrammetry and Remote Sensing **50** (1995) 13–24
7. Saadat-Seresht, M., Samdzadegan, F., Azizi, A., Hahn, M.: Camera placement for network design in vision metrology based on fuzzy inference system. In: XXth ISPRS Congress. (2004)
8. Tarabanis, K.A., Tsai, R.Y., Allen, P.K.: A survev of sensor planning in computer vision. IEEE Transactions on Robotics and Automation **11** (1995) 86–104
9. Denzler, J., Brown, C.: Information theoretic sensor data selection for active object recognition and state estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 145–157
10. Wu, J.J., Sharma, R., Huang, T.S.: Analysis of uncertainty bounds due to quantization for three-dimensional position estimation using multiple cameras. Optical Engineering **37** (1998) 280–292
11. Chen, X.: Designing Multi-Camera Tracking Systems for Scalability and Efficient Resource Allocation. PhD thesis, Stanford University (2002)
12. Olague, G., Mohr, R.: Optimal camera placement for accurate reconstruction. Pattern Recognition **35** (2002) 927–944
13. Allen, B.D.: Hardware Design Optimization for Human Motion Tracking Systems. PhD thesis, University of North Carolina at Chapel Hill (2007)
14. Ercan, A.O., Gamal, A.E., Guibas, L.J.: Camera network node selection for target localization in the presence of occlusions. In: Distributed Smart Cameras. (2006)
15. Isler, V., Bajcsy, R.: The sensor selection problem for bounded uncertainty sensing models. In: IPSN. (2005) 151–158
16. Scott, W.R., Roth, G., Rivest, J.F.: View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys **35** (2003) 64–96
17. Chen, S.Y., Li, Y.F.: Automatic sensor placement for model-based robot vision. IEEE Transactions on Systems, Man and Cybernetics **34** (2004) 393–408
18. Bodor, R., Schrater, P., Papanikolopoulos, N.: Multi-camera positioning to optimize task observability. In: IEEE Conference on Advanced Video and Signal Based Surveillance. (2005) 552–557 static cameras.
19. Mittal, A., Davis, L.: Visibility analysis and sensor planning in dynamic environments. In: European Conference on Computer Vision. (2004)
20. Yous, S., Ukita, N., Kidode, M.: An assignment scheme to control multiple pan/tilt cameras for 3d video. Journal of Multimedia **2** (2007) 10–19
21. Grewal, M.S., Andrews, A.P.: Kalman Filtering Theory and Practice. Information and System Sciences Series. Prentice Hall, Upper Saddle River, NJ USA (1993)