A Geometrically Inspired Approach to Active View Planning

Oliver Ruepp and Darius Burschka

Institut für Informatik, Technische Universität München,
Boltzmannstrasse 3, D-85748 Garching bei München, Germany
email: {ruepp|burschka}@in.tum.de

Abstract. Visual 3-D reconstruction and registration methods are gaining immense importance in many practical applications in everyday life. Unfortunately, the applicability of such techniques is often limited because the accuracy requirements of procedures are too high, or if seen the other way around, the reconstruction quality is too low. One possibility to improve this situation is active planning of camera movements such that the maximum reconstruction quality is assured. To adress this problem, we have developed a simple, easy to use approach that is directly inspired by geometric considerations.

1 Introduction

This paper presents a geometrical approach to define efficient exploration strategies depending on the perception properties of the underlying sensor in the system. This research is a basic component in our collaborative sensing approach where the exploration of a local area is performed by several small cognitive agents in a joint action effort. We plan to map it onto biological exploration strategies of insects collaborating in exploration tasks, like wasps. Our current research shows, that although it is desirable to learn from biological systems, it is easier to explain biological behaviour by mathematical models known from robotics and to use the biology to select the appropriate alternatives.

Vision-based technologies are becoming more and more important in everyday life. One of the most important tasks in that context is the 3-D reconstruction and modeling of objects and scenes. There are many problems that need to be dealt with in the context of that problem, and the accuracy of the reconstruction is one of them. Of course, it is always possible to increase the accuracy by increasing the resolution of the employed sensors. But that will also lead to an increase of sensor size, cost, and probably also computation time, because more data needs to be processed. This paper is concerned with a different approach of improving reconstruction quality, by finding a strategy for sensor placement such that the accuracy of 3-D reconstruction tasks will be improved.

One possible application of our ideas is in computer-aided surgery, where, e.g., Burschka et al. [1, 2] have developed techniques for 3-D reconstruction and active guidance of surgeons. It is clear that, especially in medical applications, the accuracy of the 3-D reconstruction is of high importance, because a too low accuracy can in extreme cases mean injuries to the patient. It would be interesting to combine the methods developed herein with their system.

Another important application of our approach is view planning for flying systems and manipulation purposes, where additional information needs to be acquired from, e.g., camera-in-hand images or interacting flying agents from as few additional positions as possible.

2 Related Work

In the context of SLAM methods, many different approaches to the accuracy maximization problem have been examined. Vidal-Celleja et al. [3] have developed a scheme for active control of a 6DOF camera, where possible movement actions are evaluated according to information-theoretic optimality criterions. It is assumed that the camera movement is chosen from a discrete set of actions at certain timesteps, and at each timestep the optimal action is determined.

Wenhardt et al. [4] use a different approach: They try to choose actions such that one of several characteristics of the covariance matrix are minimized. The minimization procedure is implemented as an exhaustive search, and the minimization criteria (examined as completely independent strategies) are the entropy, largest eigenvalue, and trace of the covariance matrix. As opposed to the approach of Vidal-Celleja et al. [3], it is assumed that the camera will directly "jump" to the computed optimal position.

That it is not advisable to blindly use information maximization schemes has been shown by Sim [5] in his work on bearings-only SLAM. It is shown that, when the sensor (e.g., a camera) is always driven to the "optimal" position (considering maximum information gain from the measurement), the update step for the Extended Kalman Filter becomes numerically unstable, and this in turn adversely affects the state estimation. The strategy developed in that work therefore aims at maximizing stability of the Kalman Filter update.

Another interesting method has been discussed by Whaite et al. [6]. Their approach is probably the one that is most similar to our work: A sensor that is constrained to move on a sphere surface is considered, and the optimal movement direction is computed with respect to the prediction variance. However, our idea uses a simplified approach to evaluate view points, and we also do not compute movement directions, but absolute positions instead.

An additional aspect of Active View Planning that is not discussed in this paper is dealing with self-occlusions of objects and assuring that all parts of an object are seen. Since the problem has been researched for several years, a lot of different methods have also been proposed [7–9]. A more recent contribution to solving that problem is the paper of Chen and Li [10].

3 3-D Reconstruction

Our method can generally be applied for any 3-D reconstruction technique that generates point position estimates as well as estimates of point position uncertainty in form of a covariance matrix. Typically, SLAM methods employing the Kalman Filter generate that kind of information, and one such method is what we have used for testing.

In our specific system, the state vector $\hat{x}$ contains the camera state estimates $\hat{x}_1, \hat{x}_2$ as well as the 3-D coordinates $\hat{y}_i$ of the n points under consideration. The state vector can be partitioned as follows:

$$\hat{x} = (\hat{x}_1, \hat{x}_2, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)^T \tag{1}$$

Because we are mainly concerned with uncertainties of our estimate in this work, we also make use of the convenient notation for relevant parts of the state covariance matrix, as introduced in Davison's [11] work:

$$P = \begin{pmatrix} P_{x_1x_1} & P_{x_1x_2} & P_{x_1y_1} & P_{x_1y_2} & \cdots \\ P_{x_2x_1} & P_{x_2x_2} & P_{x_2y_1} & P_{x_2y_2} & \cdots \\ P_{y_1x_1} & P_{y_1x_2} & P_{y_1y_1} & P_{y_1y_2} & \cdots \\ P_{y_2x_1} & P_{y_2x_2} & P_{y_2y_1} & P_{y_2y_2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{2}$$

We will primarily be interested in the submatrices $P_{y_iy_i}$, which provide a direct description of the uncertainty of the point y_i.

4 View Planning Method

As has been pointed out by, e.g., Sim [5], maximizing the information gain from measurements is equivalent to moving the camera to a position that is orthogonal to the principal direction of an observed landmark's covariance ellipsis. This is also what one would intuitively expect, since by looking at a point feature from a specific direction, we can determine the feature position pretty well in the directions parallel to the camera plane, while we are not able to determine the 3-D depth of the feature. Figure 1 visualizes the concept.

Inspired by this idea, our approach to finding the optimal camera position can be described informally as follows: First of all, we compute the covariance axes for all points, establishing the principal axes of the covariance ellipsoid. If we place the camera on a plane that contains the point and has a normal that is orthogonal to one of the axes, we will maximally reduce the covariance in the direction of that axis after a measurement.

Using this information, we can, for each point and one of its covariance axes, determine a plane on which the camera should be placed if we wish to minimize the covariance in that direction for that point. This method alone would lead to ill-conditioned filter updates as shown by Sim [5], but we are not finished yet.

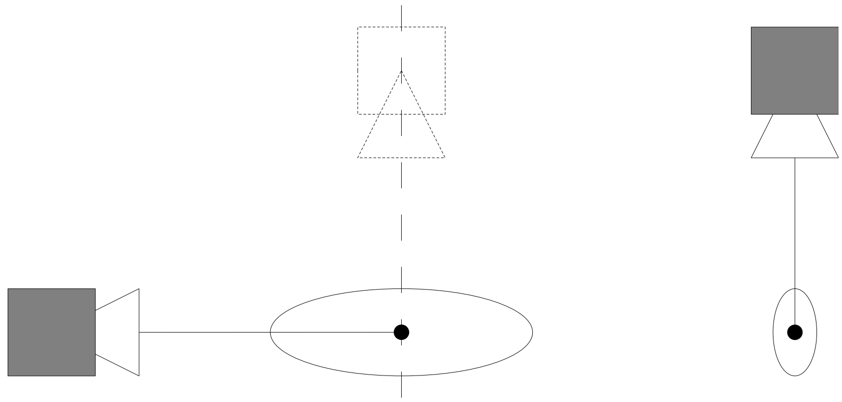

Fig. 1. The left picture shows the following situation: A feature has been measured from the shown camera position, and the measurement lead to the shown point estimate and covariance. The dashed line is orthogonal to the major covariance ellipsis axis, and also with dashed lines we indicate a camera position on that line, which seems like a good choice intuitively. The right image shows how the situation could look like after a measurement from that position: The worst covariance has been maximally reduced.

Based on the information that has been computed so far, we can introduce a penalty function for camera locations, depending on the location's distance to the "optimal" planes. A first, simple idea might be using, e.g., the squared distance to those planes. But that will not do: We would weigh all points evenly, which would apparently be suboptimal if some of them are localized already with high accuracy, and others are extremely inaccurate. Intuitively, the directions with high inaccuracy should have higher "weight" than the other points, so another thing to add to our penalty function are weights that should depend on the "imprecision" of the corresponding point's position estimate. This yields, all in all, a nice quadratic function that we need to optimize with certain side conditions.

These side conditions might be, e.g., visibility of all points, movement constraints due to limited physical mobility of our camera, etc. In our case, we focus on the point visibility constraint. It can easily be converted into some inequalities, but unfortunately, we find ourselves confronted with a quadratic programming problem with non-convex side conditions, which is probably not totally trivial to solve. It would be a lot nicer if it were possible to formulate the visibility side condition in form of a equality constraint, because this would allow us to use the concept of Lagrange multipliers.

Indeed, it does not seem too far-fetched to actually use such a constraint. If we think, e.g., of the area of medical imaging, we would mainly expect scenarios in which the surgeon needs to constantly observe a certain area of interest, without losing view of any part of that area. This means that our camera should always

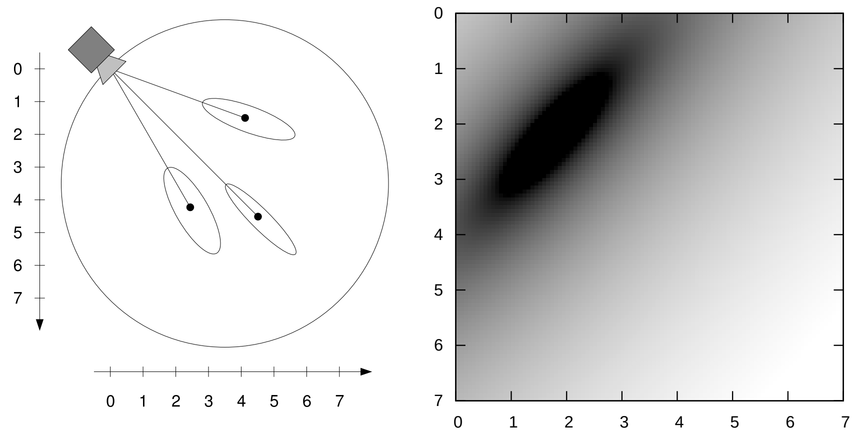

Fig. 2. An illustration of the basic idea of our approach, demonstrated for a 2-D reconstruction problem. In the left image, we show an example situation, where the camera in the upper left corner has taken an image of the scene, and the reconstruction process leads to the point positions and covariances shown. The right image shows the sum of squared distances to the principal covariance planes, with a logarithmic scale, where light gray levels correspond to high distance, and dark levels to low distance.

remain at a certain minimum distance from the observed object, and it should face towards the center of the object. Such a constraint can, e.g., be achieved easily by restricting the camera motion to a sphere around the reconstructed point cloud. Of course, using other surfaces might also be feasible, we could, e.g., consider planes, or generalize the constraint to quadrics. Also, it might be desireable to further restrict movement to parts of a surface. In this work however, we will use spheres as basis for the camera motion planning. Figure 2 summarizes the overall concept.

4.1 Mathemetical Formulation

Now that we have given an informal description of our approach, it is time to move on to the mathematical treatment of the problem. As mentioned above, the first step is the determination of the covariance ellipsoid axes for each point y_i. This is equivalent to determining the eigenvectors and eigenvalues of the associated 3×3 covariance matrix $P_{y_i y_i}$, which can conventiently be done by computing the singular value decomposition of that matrix. For each point y_i, we denote the 3 normalized, orthogonal axis vectors by $n_{i,1}, n_{i,2}, n_{i,3}$, assuming that these vectors have been sorted according to their associated eigenvalues in descending order. Those eigenvalues will be denoted by $\lambda_{i,1}, \lambda_{i,2}, \lambda_{i,3}$. With this information, we can further determine values $d_{i,j}$ such that

$$n_{i,j} \cdot x - d_{i,j} = 0 \tag{3}$$

finally describes a plane with normal vector $n_{i,j}$ passing through y_i in Hessian Normal Form in 3-D Euclidean space.

The Hessian Normal Form of a plane has the convenient property that substituting any point $\bar{x}$ into the expression $n_{i,j} \cdot x - d_{i,j}$ will yield the distance of $\bar{x}$ to that plane. The value will be positive or negative indicating which side of the plane $\bar{x}$ is located on. Using this knowledge, we can already formulate our penalty function as follows:

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} (x^T n_{i,j} - d_{i,j})^2 = x^T A x - 2b^T x + c \tag{4}$$

Where A, b, c are defined as follows:

$$A := \left(\sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} n_{i,j} n_{i,j}^T \right), \quad b := \left(\sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} d_{i,j} n_{i,j}^T \right), \tag{5}$$

$$c := \sum_{i=1}^{n} \sum_{j=1}^{3} w_{i,j} (d_{i,j})^2 \tag{6}$$

It is clear that c can be left out for minimization purposes, so it will be dropped in future references to f. Here, $w_{i,j}$ is the weight assigned to the axis $n_{i,j}$, the choice of which will be discussed later.

Specifying the side condition is straightforward: If m is the center point of the sphere that the camera can move on, and r is the radius of that sphere (which has been determined by whatever means), then the simple equation

$$g(x) = 0 \quad \text{with} \quad g(x) = (x - m)^T \cdot (x - m) - r^2 \tag{7}$$

defines the sphere surface. Applying the principle of Lagrange multipliers, we arrive at the following equations for finding candidates for extremal points of $f(x)$ on $g(x) = 0$:

$$\nabla f(x) = \mu \nabla g(x) \quad \Leftrightarrow \quad Ax - b = \mu(x - m) \tag{8}$$

We can simplify this somewhat by looking at functions $f'(x) = f(x+m), g'(x) = g(x+m)$ that are just shifted versions of f and g. This will change above equations as follows:

$$A(x + m) - b = \mu x \Leftrightarrow Ax + Am - b = \mu x \tag{9}$$

Defining $b' = -(Am - b)$, the equation finally becomes

$$Ax - b' = \mu x \tag{10}$$

Solving this problem is not altogether trivial. Fortunately, the problem of optimization of quadratic functions on a sphere surface has already been researched thoroughly. It is briefly discussed, e.g., as a special case in Hager's paper [12] on quadratic optimization within a sphere. Using the explanation there, we can devise a simple algorithm that allows us to compute the desired optimum. We explain the basic idea here.

If we rearrange terms in the last equation, we see that x can be computed depending on μ:

$$x = (A - \mu I)^{-1} b' \tag{11}$$

The problem is then one of finding the right value of μ, such that the constraint $|x| = r$ is satisfied. Let ϕ_i be the eigenvectors of A, and λ_i the corresponding eigenvalues, sorted ascendingly. It is known that $(A - \mu I)^{-1}$ has the same eigenvectors ϕ_i as A, with eigenvalues $1/(\lambda_i - \mu)$. This observation allows us to express the right side of above equation as

$$\sum_{i=1}^{3} \frac{\beta_i}{(\lambda_i - \mu)} \phi_i, \tag{12}$$

where $\beta_i = \phi_i^T b$. If this representation is combined with the length constraint, we arrive at the following equation:

$$\sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_i - \mu)^2} = r^2 \tag{13}$$

This is essentially a polynomial of degree 6, which is in general not possible to solve. However, it can be shown that $\mu < \lambda_1$ must hold for a solution. We can also see that the left side of above equation is strongly convex. With this knowledge, it is possible to compute upper and lower bounds of μ, between which we can search for the solution using a bisection algorithm.

The bounds can be computed as follows: We have

$$\sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_i - \mu)^2} \leq \sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_1 - \mu)^2} \tag{14}$$

on the one hand, and

$$\sum_{i=1}^{3} \frac{\beta_i^2}{(\lambda_i - \mu)^2} \geq \sum_{i \in \mathcal{E}_1} \frac{\beta_i^2}{(\lambda_1 - \mu)^2} \tag{15}$$

on the other hand, where $\mathcal{E}_1$ is the set $\{i \mid \lambda_i = \lambda_1\}$. Both inequalities can be used to compute the bounds. One last difficulty are the so-called degenerate cases, where the boundary computation will fail. They correspond to cases where b' is orthogonal (or close to orthogonal) to the eigenvectors corresponding to the eigenvalues from $\mathcal{E}$. Fortunately, it is possible to use a simple alternate computation to compute the optimum.

4.2 Choice of Weights

One question that remains to be answered is the choice of the weights $w_{i,j}$ that are assigned to the planes in our scenario. So far, we have only explained that those weights should somehow be connected to the uncertainty of the point — now we will actually explain how we choose the weights and why we do so.

The first idea one might come up with is weighing only those planes corresponding to the highest uncertainty direction of a point. Thus, our method would be equal to trying to maximally reduce the worst uncertainties in the estimation for each point. The rule for choosing weights can be formulated as follows:

$$w_{i,j} = \begin{cases} 1 & j = 1 \\ 0 & j \neq 1 \end{cases} \tag{16}$$

One flaw of this idea is obvious: It would weigh all planes evenly. This is, of course, not optimal, because it might be that some of the points we are looking at are already localized very well, while others are localized very bad.

This observation leads to the following, slightly modified rule:

$$w_{i,j} = \begin{cases} \lambda_{i,j} & j = 1 \\ 0 & j \neq 1 \end{cases} \tag{17}$$

Instead of choosing the weight 1 for each of the planes corresponding to the highest uncertainty, we choose as weight the eigenvalue corresponding to that plane. This is justified by the fact that the eigenvalue can be interpreted directly as measure of uncertainty in the direction of the associated eigenvector.

The last rule for choosing weights is already a clear improvement, but we can still see one problem: What if the eigenvalues associated with a point are very close together, or even equal? We might, e.g., think of a case where the uncertainty ellipsoid looks similar to a disc, which would happen when two eigenvalues are of equal size and very big compared to the last eigenvalue. Another interesting case is when the ellipsoid is sphere-shaped, meaning that all eigenvalues are equal. The solution to these problems is simple: We use the rule

$$w_{i,j} = \lambda_{i,j}. \tag{18}$$

To see why this rule helps with the problems outlined above, let us think about an example: Let $\lambda_{i,1} = \lambda_{i,2} = 1, \lambda_{i,3} = 0$. In that case, optimal camera positions are characterized by being contained in the planes corresponding to $n_{i,1}$ and $n_{i,2}$. The best camera positions are on the intersection of those planes, and thus on a line, which makes sense intuitively: By taking a measurement of the point from some place on the line, we can achieve maximum reduction of the two worst uncertainties. In the case where $\lambda_{i,1} = \lambda_{i,2} = \lambda_{i,3} = 1$, the best camera position would be the intersection of the three planes, thus we would consider the point itself as optimal position. In our evaluation function, this would mean that we would simply try to position the camera as close to the point as possible, which is an acceptable strategy.

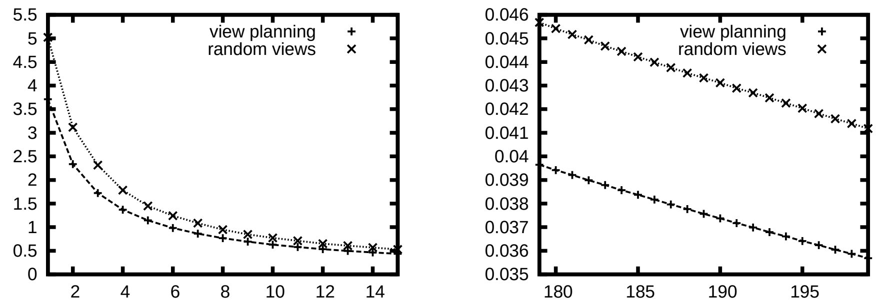

Fig. 3. Traces of the covariance matrices produced with our view planning method, and with a randomized view point choice. The left diagram only show the first 15 steps, where the gain from using our method is most significant. The right diagram shows the covariance trace for steps 180 to 200, where the difference is smaller, but our strategy still performs better. The data shown for the random viewpoint selection has been obtained by averaging over 10 runs.

Note that the last rule for choosing weights means that the computation of the matrix A is specifically simple, we have

$$A := \left(\sum_{i=1}^{n} P_{y_i y_i} \right) . \tag{19}$$

5 Results

We have tested our approach in a simulated environment, where in each step, an optimal camera position is computed, and the camera position is set accordingly for the reconstruction step. We have compared the approach to a randomized viewpoint selection, and found that the results are significantly better.

Especially in the first steps of the reconstruction process, the advantage of using our view planning approach is striking. Figure 3 shows a diagram of our results. As a measure of overall estimation uncertainty, we have used the trace of the point covariances. This corresponds to the sum of eigenvalues, and thus seems to be a good measure. To make sure that the results from randomized view planning are not biased, we have built the average of the results of 10 reconstruction runs.

Also, a comparison of the different weight choosing schemes has been performed. It turns out that the final method really is the one that works best, as can be seen in the diagrams of Figure 4.

The evaluation has also been tried using other measures of uncertainty. In one case, this lead to an interesting observation: When using the spectral norm, the randomized view point selection still performed worse than our view planning method in the beginning. But surprisingly, after some time, the norm of the

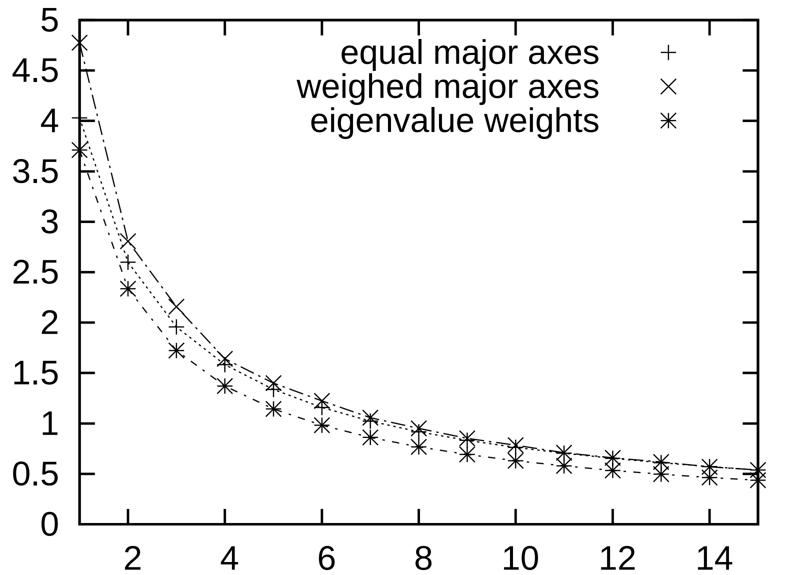

Fig. 4. The three weight choice strategies in comparison: The weighing using the eigenvalues clearly performs best.

covariance matrix achieved with the randomized planner becomes smaller than that achieved with our view planning scheme.

The explanation for that phenomenon is simple: The spectral norm of a symmetric matrix is equal to its biggest eigenvalue. Consequently, this means that the worst uncertainty after using randomized views is better than after using view planning. Since the trace of a matrix can be interpreted as the sum of eigenvalues of that matrix, we can deduce the following conclusion: While the highest uncertainty produced with our algorithm might be worse, the average uncertainty is better, and also by an significant amount.

6 Discussion

The advantage of our approach versus other methods is that it is computationally very simple, and the view planning recommendation can be computed very efficiently. Most other approaches that rely on more complicated optimality measures also have to use expensive exhaustive sampling techniques, which leads to much higher complexity.

However, we have made some strongly simplifying assumptions. We totally ignore the uncertainty of the sensor. We ignore visibility problems, and we are only focusing on computing camera positions, while not considering the camera angle at all. To solve problems of visibility and self-occlusion, we hope to combine our approach with some other methods of Next Best View planning that have already been developed and can be used to assure that the object under consideration is explored completely.

We have constrained the sensor to a very simple surface, which is convenient mathematically, but also a too strong limitation for some applications. It would be interesting to constrain the camera position to mathematical bodies instead of surfaces. We might, e.g., allow the camera positions to be placed between an inner and an outer sphere. We could consider “cutting” parts of spheres by specifying a plane, and requiring that the camera is placed on a specific side of a plane. There are many possibilities to allow more general side conditions.

The method developed herein is for now constrained to situations where the camera position can be controlled directly, and the camera is able to “jump” to a recommended position between frames. It remains to examine how well it performs in settings where only direction indications are given.

Finally, we have not yet developed methods to present the results of the view point optimization to a human. Thinking again about the setting of endoscopic surgery, it would be necessary to find a representation that can easily be understood and executed by humans. So, we still need to work towards finding an appropriate way for presenting data to users of our system.

Acknowledgments

This work was supported by the DFG Cluster ”Cognition in Technical Systems” (CoTeSys).

References

1. Burschka, D., Corso, J.J., Dewan, M., Lau, W., Li, M., Lin, H., Marayong, P., Ramey, N., Hager, G.D., Hoffman, B., Larkin, D., Hasser, C.: Navigating inner space: 3-d assistance for minimally invasive surgery. Robotics and Autonomous System (2005)
2. Burschka, D., Li, M., Taylor, R., Hager, G.D., Ishii, M.: Scale-invariant registration of monocular endoscopic images to ct-scans for sinus surgery. Medical Image Analysis **9**(5) (October 2005) 413–439 Best Paper Award at MICCAI 2005.
3. Vidal-Calleja, T., Davison, A., Andrade-Cetto, J., Murray, D.: Active control for single camera slam. In: IEEE Int Conf on Robotics and Automation, Orlando, May 2006. (2006)
4. Wenhardt, S., Deutsch, B., Angelopoulou, E., Niemann, H.: Active visual object reconstruction using d-, e-, and t-optimal next best views. In: CVPR, IEEE Computer Society (2007)
5. Sim, R.: Stable exploration for bearings-only slam. In: ICRA, IEEE (2005) 2411–2416
6. Whaite, P., Ferrie, F.P.: Autonomous exploration: Driven by uncertainty. IEEE Trans. Pattern Anal. Mach. Intell. **19**(3) (1997) 193–205
7. Maver, J., Bajcsy, R.: Occlusions as a guide for planning the next view. IEEE Trans. Pattern Anal. Mach. Intell. **15**(5) (1993) 417–433

8. Kutulakos, K.N., Dyer, C.R.: Global surface reconstruction by purposive control of observer motion. Artif. Intell. **78**(1-2) (1995) 147–177
9. Cowan, C.K., Kovesi, P.D.: Automatic sensor placement from vision task requirements. IEEE Trans. Pattern Anal. Mach. Intell. **10**(3) (1988) 407–416
10. Chen, S.Y., Li, Y.F.: Active viewpoint planning for model construction. In: ICRA, IEEE (2004) 4411–4416
11. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(6) (2007) 1052–1067
12. Hager, W.W.: Minimizing a quadratic over a sphere. SIAM J. on Optimization **12**(1) (2001) 188–208