The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Developing Visual Sensing Strategies through Next Best View Planning

Enrique Dunn, Jur van den Berg and Jan-Michael Frahm

Abstract— **We propose an approach for acquiring geometric 3D models using cameras mounted on autonomous vehicles and robots. Our method uses structure from motion techniques from computer vision to obtain the geometric structure of the scene. To achieve an efficient goal-driven resource deployment, we develop an incremental approach, which alternates between an accuracy-driven next best view determination and recursive path planning. The next best view is determined by a novel cost function that quantifies the expected contribution of future viewing configurations. A sensing path for robot motion towards the next best view is then achieved by a cost-driven recursive search of intermediate viewing configurations. We discuss some of the properties of our view cost function in the context of an iterative view planning process and present experimental results on a synthetic environment.**

I. Introduction

This work presents a novel approach for the concurrent solution of the problems of viewpoint selection and path planning for a mobile robotic platform used for visual 3D reconstruction. In general terms, sensor planning systems strive to determine the pose and settings of a vision sensor to undertake a vision task usually requiring multiple views [2]. The application of these systems span from robotic exploration and navigation to automated surveying and modeling of complex and/or unstructured 3D environments. In the context of mobile robotics, the next best view problem (NBV) consists in determining the most favorable future sensing action to be performed by the robot in an effort to achieve specific task goals. For the scenario of vision-based reconstruction considered in this work, sensing actions involve moving the robotic platform to a desired location and acquiring images. We refer to the accumulation of sensing actions by the term of *sensing strategy*.

Incremental 3D reconstruction by means of iterative NBV planning allows the development of systematic and principled sensing strategies for an autonomous observer. However, the use of a robotic platform to perform sensing actions brings up the issue of how to best utilize such infrastructure during the reconstruction process. Moreover, given knowledge of the NBV position and settings, *is it possible (and favorable) to execute additional sensing actions during the robot's motion toward the NBV?* An affirmative answer to this question leads to the fact that taking intermediate sensing actions modifies our knowledge of the environment and may render obsolete our existing estimation of the NBV. A straightforward solution to this problem is to recompute the NBV after each single intermediate sensing action, but this may either be computationally not feasible or lead to reactive non-convergent sensing behavior. Instead, in this paper we explore an alternative solution where a path towards the NBV is followed until either the NBV is reached or the expected qualitative benefits of the NBV are achieved.

The goal of our NBV planner is to guide the process of systematically increasing the precision and completeness of the estimated 3D model. Our proposed NBV planning approach uses adaptive planar primitives as the basic element of structure representation while using their covariance matrices as the representation for the 3D reconstruction uncertainty of the primitives. The approach is aimed at (quasi-)dense 3D reconstructions, which commonly output millions of surface points for even simple object-centered scenes. Accordingly, scalability and efficiency are major concerns when developing a viewpoint selection algorithm in this context. To this end we propose a data parallel hierarchical approach that can efficiently deploy commodity parallel architectures like GPUs or multi-core processors.

II. Related Work

Determining the NBV for 3D reconstructions based on a range scanner sensor is an active research field [9],[13], [6], [1]. In this work we address the task of 3D reconstruction based on intensity images. The challenge of automatic viewpoint selection has been widely studied in robotics, computer vision and photogrammetry. Surveys that span from early approaches in this field to recent advances were published by Newman *et al.* [10], Tarabanis *et al.* [17] and Scott *et al.* [15]. Recently Chen *et al.* [2] provide a broad coverage of multiple research areas within sensor planning.

The task of designing a viewing configuration for precise 3D reconstruction is known in photogrammetry as the photogrammetric network design (PND) problem. Fraser [3] early on identified the analytical difficulties of designing an optimal imaging geometry in the context of rigorous photogrammetric 3D measurements. His work identified the high non-linearity and multi-modality, which makes the PND problem ill-suited for canonical optimization methods. Mason [8] adopted an expert systems approach based on generic networks to achieve strong viewing configurations for model based PND. The developed system used CAD model as input and followed a series of predetermined rules for each CAD element in order to design an imaging geometry. Olague and Mohr [12] addressed the PND problem by developing a criterion based on forward covariance propagation of image measurement uncertainty. The criteria was the maximum element along the diagonal of the reconstruction's covariance matrix and the optimal multiview configuration was obtained by global evolutionary search. Note that the aforementioned PND systems where designed to generate sets of multiple viewpoints used for highly precise 3D reconstruction tasks,

carried out in well controlled and customized environments (i.e. fiducial markers, high accuracy calibration patterns, etc.). Accordingly, they mainly address the geometric aspects of 3D reconstruction omitting considerations on the role of texture saliency in the image measurement process.

Robot vision researchers have studied how a controlled camera can be best used to achieve accurate 3D reconstructions. Whaite and Ferrie [20] developed a 3D modeling system that used parametric modeling of scene elements and used the internal model uncertainty to determine sensing actions. Marchand and Chaumette [7] developed a system for structured scene reconstruction by developing optimal strategies for surveying a set of volumetric primitives. We note that, while these systems successfully achieved autonomous operation, they used simple parametric models to represent scene elements, whereas our approach does not place any restrictions on the observed geometry. Accordingly, our approach is better suited to perform over a large variety of scenes.

In the computer vision community, camera placement and configuration has recently received renewed interest [19],[14]. Wenhardt *et al.* [18] proposed a 3D reconstruction based on a probabilistic state estimation framework where the NBV is determined by a metric of the state estimation's uncertainty. The authors propose the use of three different metrics corresponding to the concepts of D-, E- and T-Optimality found in the optimal experimental design literature. Hornung *et al.* [5] propose an image selection scheme for multi-view stereo, that selects images in order to improve the coverage of a voxel based proxy. Their approach strives to achieve sufficient sampling of the entire object's surface while identifying regions with poor photo-consistency for additional redundant sampling. The authors make use of GPU assisted computation and present results on multiple object oriented scenes.

III. 3D Structure and Uncertainty Estimation

In our approach, scene geometry is represented by a set of primitives of varying scale. Hence, our model can represent general scene geometry (by using primitives as small as a planar surface of the size of a pixel at the scene distance), while efficiently representing larger planes through a single model plane. Each primitive is parameterized by

$$P_i = [\mathbf{X}_i, \mathbf{\Sigma}_i, \mathbf{S}_{ij}] : \{\mathbf{X}_i \in \mathbf{R}^3, \mathbf{\Sigma}_i \in \mathbf{R}^{3\times 3}, \mathbf{S}_{ij} \in \mathbf{R}^{p\times p}\},$$

where $\mathbf{X}_i$ is the primitive's 3D position, $\mathbf{\Sigma}_i$ is the 3D covariance matrix and $\mathbf{S}_{ij}$ is the square set of $p \times p$ (p is a user defined integer value) neighboring image pixels to the projection of P_i onto image j. Also, viewpoint configurations are parameterized in terms of sensor position and orientation angles,

$$\nu_j = [\mathbf{x}_j, \theta_j] : \{\mathbf{x}_j \in \mathbf{R}^3, \theta_j \in SO(3)\}.$$

In this work, the estimation of 3D structure and the associated geometric uncertainty of each primitive is performed by an individual extended Kalman filter (EKF). While this approach does not consider the uncertainty correlation among the estimates of different primitives, it does provide a computationally scalable framework for 3D structure and uncertainty estimation of large 3D environments. For a given viewpoint ν and a 3D primitive P, we use the well known collinearity equations as our non-linear observation function $\phi(\mathbf{X}, \nu)$. Furthermore, by considering a static 3D scene the state propagation (e.g. time update) equations of the EKF can be obviated. In this way, each EKF incorporates new observations $o_t = (u_t, v_t) \in \mathbf{R}^2$ into the state estimate by the following measurement update equations:

$$K_t = \Sigma_{t-1} H_t^T(\nu_t)(H_t(\nu_t)\Sigma_{t-1}H_t^T(\nu_t) + R)^{-1} \quad (1)$$

$$\hat{\mathbf{X}}_t = \hat{\mathbf{X}}_{t-1} + K_t(o_t - \phi(\hat{\mathbf{X}}_{t-1}, \nu_t)) \quad (2)$$

$$\Sigma_t = (I - K_t H_t(\nu_t))\Sigma_{t-1} \quad (3)$$

where K_t is the Kalman gain matrix, R is the image measurement covariance matrix and $H_t(\nu_t)$ is the Jacobian matrix of $\phi(\cdot,\cdot)$, derived at $\hat{\mathbf{X}}_{t-1}$ and using the sensor placement ν_t. We use this framework to compute the effect of incremental visual sensing for each primitive. Moreover, the conceptual motivation for using an EKF framework lies on the non-incremental nature of uncertainty estimates.

Lemma 1. A steady state EKF framework presents non increasing uncertainty estimates for successive measurements.

Proof. From (1) we can define

$$W = \Sigma_{t-1} H_t^T(\nu_t)$$

$$S = (H_t(\nu_t)\Sigma_{t-1}H_t^T(\nu_t) + R)^{-1}$$

and rewrite (3) as $\Sigma_t = \Sigma_{t-1} - WSW^T$. Accordingly, the term WSW^T is a PSD (positive semidefinite) matrix and subtracting one PSD matrix from another can't cause the eigenvalues to increase $\square$

The geometric structure of the 3D uncertainty of the input model is captured by the eigenvectors and eigenvalues of the primitive's covariance matrix $\mathbf{\Sigma}$. Namely, in Euclidian 3D space the eigenvectors $\mathbf{e}_k | k = 1 \ldots 3$ convey the orientation of the 3D uncertainty, while the eigenvalues λ_k specify the magnitude in each direction. We define $\mathbf{\Psi}$ to be the matrix of eigenvectors scaled by their corresponding eigenvalue,

$$\mathbf{\Psi} = [\lambda_1 \mathbf{e}_1 \ \lambda_2 \mathbf{e}_2 \ \lambda_3 \mathbf{e}_3].$$

The planning approach presented here uses the information contained in $\mathbf{\Psi}$ as the guide for viewpoint selection.

IV. A Novel Next Best View Criterion

It is well known that for 3D reconstruction approaches based on optical triangulation, a larger viewing angle among observing camera positions helps attain precise 3D estimations [4]. However, by increasing the baseline and incidence among cameras, image measurement and matching are made more difficult. The main reason for these difficulties is that the surface texture appearance may vary widely across distant viewpoints. Accordingly, a novel viewpoint must achieve a balance between the reduction of geometric uncertainty and the attainment of reliable image measurements. Our proposed criterion seeks the balance by considering the geometric uncertainty and the matching uncertainty simultaneously.

The proposed criterion relies on 3D uncertainty information contained in a primitive's covariance matrix $\mathbf{\Sigma}$. In this way, an uncertainty volume (i.e. ellipsoid) can be estimated from each covariance matrix and the problem of reducing the overall uncertainty can be posed as the problem of reducing a chosen metric defined over the values of all covariances matrices. Instead of optimizing an experimental design criterion as in [18], we define a criterion based on a set of geometric relationships, which contribute to systematically reducing a 3D primitive's uncertainty.

The first geometric relationship being sought is achieving an adequate incidence with respect to a primitive's 3D uncertainty. The second desired relationship is to obtain a favorable imaging resolution for a given 3D primitive. Finally, we condition the relevance of these factors on the primitive's texture saliency. In this way, our planning approach is applicable to feature based reconstruction algorithms as well as to their contour based counterparts.

A. Reducing 3D uncertainty

Let $\mathbf{X}_i$ denote the estimated 3D position of a primitive P_i. The goal is to find the viewpoint ν_j with camera position $\mathbf{x}_j$ such that the unit length viewing direction

$$\mathbf{v} = \frac{\mathbf{X}_i - \mathbf{x}_j}{\|\mathbf{X}_i - \mathbf{x}_j\|_2}$$

best reduces the 3D uncertainty contained in Σ_i. Given an estimate of a 3D point with non isotropic 3D uncertainty, the most favorable viewing rays $\mathbf{v}$ for minimizing triangulation uncertainty are the ones orthogonal to the main uncertainty direction vector. Accordingly, for 3D estimates where the majority of the uncertainty is found along a single direction $\mathbf{e}_1$ (e.g. the eigenvector with the largest associated eigenvalue), a viewing ray orthogonal to this vector is desired. Such viewing ray corresponds to a solution of the product equation $\mathbf{v}^T\mathbf{e}_1^i = 0$. However, a more general criterion is desired for nearly isotropic uncertainty. We propose to find the viewing ray minimizing

$$f(P,\nu) = \|\mathbf{v}^T \left[\lambda_1\mathbf{e}_1^i \ \lambda_2\mathbf{e}_2^i \ \lambda_3\mathbf{e}_3^i\right] \|_2 \ = \ \|\mathbf{v}^T\mathbf{\Psi}_i\|_2. \quad (4)$$

The above arguments consider 3D reconstruction as a merely geometric task, not taking into account practical aspects such as robustness of image measurements and matching. We incorporate these aspects into our approach by also considering the effects of varying a viewpoint's incidence and proximity with respect to a given 3D primitive.

B. Combining incidence and proximity

3D reconstruction deals with estimating the position of points located on a 3D supporting surface. The visual appearance of this supporting surface allows the identification and measurement of the projection of a given set of 3D points onto the image plane. In general terms, better accuracy in image measurements can be obtained as the imaging resolution increases. Moreover, given knowledge of a camera's intrinsic parameters, the main factors in determining a surface's projection on the image plane are the viewing angle and the distance from a given 3D surface. We propose to combine both incidence and proximity by measuring a single quantity: the area of projection of a 3D surface onto the image plane. It is straightforward to compute this quantity analytically for simple geometric primitives. Alternatively, it can be computed with high efficiency in a GPU by using a 3D rendering engine such as OpenGL. The benefits of using a GPU computation in this context are that aspects such as resolution, field of view and occlusions can be handled by the graphics engine. Moreover, we define a function $g(\nu, P)$ for a primitive's projected area to be included into our criterion for NBV selection.

At this point we have defined geometric relationships favoring a suitable observation of a generic surface. The motivation behind such definitions is to attain reliable image measurements. The geometric relationship presented in the next subsection incorporates into our criterion the contingency cases where a given surface does not provide sufficient texture to make reliable image measurements.

C. Incorporating Texture

Visual saliency of a scene surface is a requirement for robust matching across images in feature based reconstruction. Accordingly, textureless scene regions present a major difficulty in the application of these algorithms. On the other hand, contour based approaches do not rely on texture saliency to estimate bounding volumes, but instead favor tangent views of the scene surface. In our approach, we strive for oblique views of textureless regions. Note that the motivation behind measuring the projected area of a 3D primitive was to consider jointly a viewpoint's incidence and proximity. Taking into account that the projected area of a perpendicularly observed planar surface is null, the relevance of the projected area of a primitive is conditioned on its texture. This relevance factor can be modeled by a continuous step function with transition at a given texture threshold. We propose to use a modification of the well known *Gauss error function* (encountered by integrating the normal distribution) of the form

$$erf(x) = \frac{1}{\gamma\pi}\int_0^x e^{\frac{(t-\tau)^2}{\gamma}}dt + \frac{1}{2} \quad (5)$$

where x is the texture measure estimated for a given primitive, τ is the texture threshold value and γ is a decay factor controlling the slope of the transition in the step function. Using Eq. (5) we can obtain a value in the range $[0, 1]$ to describe the relevance of the projected area of a given primitive. The measure used to describe texture saliency is described next.

Let $\mathbf{S}_i$ denote the image region corresponding to the surface of a 3D primitive. We propose measuring the entropy of the autocorrelation function $A(\mathbf{S}_i)$ of a given patch to describe texture saliency. This is motivated by the fact that homogeneous texture regions will display fairly "flat" profiles for $A(\mathbf{S}_i)$, leading to high entropy. On the other hand, surfaces with salient texture will provide a well localized "peak" on the autocorrelation function landscape, leading to

low entropy. For an image region of size $p \times p$, the $A(\mathbf{s})$ will output a matrix of dimensions $2p-1 \times 2p-1$ with values a_i in the range $[-1, 1]$. The values of this matrix are normalized and used to evaluate Shannon entropy.We empirically define a texture threshold value τ as the cutoff point for our step function (5), as well as the decay value γ. Hence, we have a function of the form

$$h(P_i) = erf\left(-\sum_{a_i \in A(\mathbf{S}_i)} p(a_i) \log p(a_i)\right). \quad (6)$$

The proposed function (6) accordingly measures the quality of a correlation match given the local appearance. It does not include currently any correction for the texture frequency reduction due to potential projective distortions. This can easily be integrated into the computation of the auto-correlation. We found that in practice it does not change the results significantly, but the simpler measurement (6) is more efficient to compute since it is a constant for a given model and does not depend on the location of the new camera.

D. The aggregate criterion

In developing our geometric criterion we seek to attain a trade-off between two (typically conflicting) objectives involved in depth estimation. These objective are: 1) maximizing the visibility of a given patch on the novel image, 2) aligning the camera viewing direction to the direction of smallest uncertainty for the considered 3D primitive P_i. We propose the following function to evaluate the contribution of a viewpoint ν for a single 3D primitive P:

$$C(\nu, P) = \frac{g(\nu, P)^{h(P)}}{f(\nu, P)} \quad (7)$$

where $g(\nu, P)$ denotes the computed projection area of the 3D primitive (as discussed in section IV-B), while $h(P)$ and $f(\nu, P)$ are defined in Eqs. (6) and (4) respectively. The function (7) is evaluated for each primitive and combined through a weighted sum to define our NBV criterion

$$F(\nu) = \sum_{i=0}^{N} w_i\, C(\nu, P_i). \quad (8)$$

We define a primitive's weight value to be

$$w_i = \det(\mathbf{\Sigma}) = \prod_{j=1}^{3} \lambda_j^i \quad (9)$$

where λ_j^i represent the eigenvalues associated with the primitive's covariance matrix $\mathbf{\Sigma}_i$. In this way, patches with larger uncertainty are given more attention in the viewpoint search process. It is important to note that the weight value could alternatively be defined in terms of more specific experimental design criteria (as presented in [18]) in order to favor the reduction of a particular uncertainty metric.

V. Path Planning

Our approach to path planning seeks a balance between robot motion efficiency and 3D reconstruction quality of the estimated 3D model $\mathcal{M}$. For our planning purposes we consider the viewpoint specification ν to completely define the robot configuration and assume the absence of non-holonomic motion constraints. Moreover, by considering a constant value for sensor elevation we can assume path planning for a point robot in a 2D plane. Given knowledge of the existing 3D model $\mathcal{M}$, the current viewpoint configuration ν_{init} and the desired sensing configuration ν^* (determined by global NBV planning), the goal of our path planning approach is to determine a sequence $\nu_i : \{i = 1 \dots n,\ \nu_1 = \nu_{init}, \dots, \nu_n = \nu^*\}$ of intermediate sensing configurations to be adopted in the transition from ν_{init} to ν^* by means of robot motion.

We consider a discretization of the search space for path planning by means of a 2D grid $G \in \mathbf{R}^{n \times n}$ centered around the object being observed. For each cell G_{ij} in the grid, we consider a viewing configuration ν_{ij} located at the center of the cell and oriented towards the center of the grid. The *cost* of each cell is determined as the inverse of the NBV evaluation function (8) and scaled linearly to reside in the $[0, 1]$ range. In this way, the cell containing the NBV will be the one with the globally lowest cost and the selected route will be the minimum cost path connecting ν_{init} and ν^*. The solution to such path planning problem can be sought by using a gradient descent search strategy. However, in order to avoid scenarios where the gradient based search may stagnate in local minima, we implement a recursive approach for cost guided path planning and describe it in the following subsection.

A. Recursive Path Planning

Our path planning approach is restricted to determining sensor positioning in terms of a fixed 2D grid, while sensor orientation is pre-computed for each cell G_{ij} as the orientation of maximal benefit. In this way, the viewpoint configuration assigned for each cell approximates the local optimum among the viewpoints located within the cell's area.

Our recursive path planning approach determines a set S of viewpoints approximately equidistant to both the start and goal 2D positions $\mathbf{x}_0$ and $\mathbf{x}_1$. We select the valid viewpoint position $\mathbf{x}^*$ corresponding to the greatest element in the set S with respect to $F(\nu_{ij})$. The validity of a viewpoint $\mathbf{x}^*$ depends on the existence of a path from $\mathbf{x}_0$ to $\mathbf{x}_1$, passing through $\mathbf{x}^*$ (i.e. $[\mathbf{x}_0 \rightarrow \mathbf{x}^* \rightarrow \mathbf{x}_1]$). To perform this test we utilize an auxiliary path planning function PathQry$(\mathbf{x}_0, \mathbf{x}^*, \mathbf{x}_1)$ which determines the shortest connecting path by *breadth-first* search. NBV planning recursion is applied by treating the intermediate viewpoint position $\mathbf{x}^*$ as the starting and goal 2D position of two different paths (i.e. $[\mathbf{x}_0 \rightarrow \mathbf{x}^*]$ and $[\mathbf{x}^* \rightarrow \mathbf{x}_1]$). Recursion halts once a proximity threshold among starting and goal positions is reached. The range of candidate viewpoint positions S is reduced at every recursion level to guarantee convergence toward the goal position. Moreover, the range of equidistant

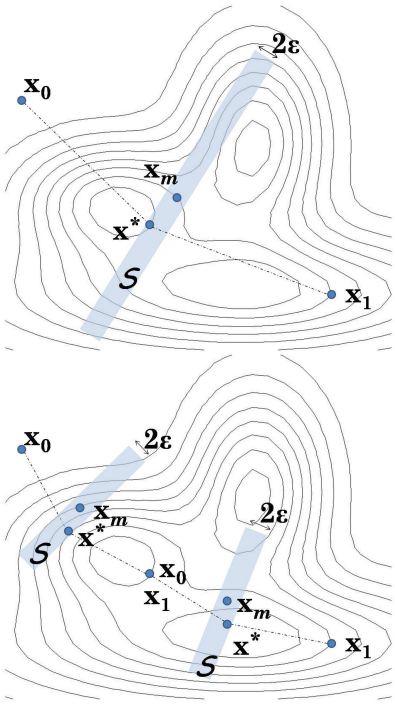

Fig. 1. Recursive Path Planning. Starting from two viewpoint positions ($\mathbf{x}_0$,$\mathbf{x}_1$), intermediate viewpoint positions $\mathbf{x}^*$ are selected from a set $\mathcal{S}$ of positions approximately equidistant to $\mathbf{x}_0$ and $\mathbf{x}_1$. The position $\mathbf{x}^*$ is the minimal element of $\mathcal{S}$. Note how the extent of $\mathcal{S}$ is proportional to the distance between $\mathbf{x}_0$ and $\mathbf{x}_1$. Two successive levels of recursion are illustrated.

SplitPath($\mathbf{x}_0$,$\mathbf{x}_1$):

//Compute mid-point
$\mathbf{x}_m \leftarrow (\mathbf{x}_0 + \mathbf{x}_1)/2$
//Compute distance between end points
$d \leftarrow \|\mathbf{x}_0 - \mathbf{x}_1\|_2$
`if` $d \leq 2\epsilon$ `return` TRUE
// Compute normal vector to separating axis
$\hat{\mathbf{x}} \leftarrow (\mathbf{x}_1 - \mathbf{x}_0)/\|\mathbf{x}_1 - \mathbf{x}_0\|_2$
// Group points in the ϵ-neighborhood of the
// separating axis and at distance not greater than $d/2$
// from the mid-point
$\mathcal{S} \leftarrow \{G_{ij}(\mathbf{y}) :| \ (\mathbf{y} - \mathbf{x}_m)^T \hat{\mathbf{x}} \ | \leq \epsilon, \ \|\mathbf{y} - \mathbf{x}_m\|_2 \leq d/2\}$
// Among elements of $\mathcal{S}$ reachable from both end points
// select the element with minimal cost
$\mathbf{x}^* \leftarrow \mathbf{x}\left(\underset{G_{ij} \in \mathcal{S}}{\arg\min} \ C\left(G_{ij}\right) : \exists \mathrm{PathQry}(\mathbf{x}_0, \mathbf{x}(G_{ij}), \mathbf{x}_1)\right)$
`if` $\mathbf{x}^* = \emptyset$ `return` FALSE
// Apply recursion on *left* half-plane
`if NOT` SplitPath($\mathbf{x}_0, \mathbf{x}^*$) `return` FALSE
// Insert mid-point after *left* half-plane recursion
$\mathcal{P} \leftarrow \mathcal{P} \cup G_{ij}(\mathbf{x}^*)$
// Apply recursion on *right* half-plane
`if NOT` SplitPath($\mathbf{x}^*, \mathbf{x}_1$) `return` FALSE
`return` TRUE

Fig. 2. Pseudo-code for SplitPath(). Taking an initial and goal positions as input parameters a path $\mathcal{P}$ is recursively generated by selecting intermediate viewpoint from a local subset $\mathcal{S}$.

positions is proportional to distance among starting and goal positions at the current recursion level.

The benefits of the proposed approach are two-fold. First, by considering a reduced local set $\mathcal{S}$ of candidate viewpoints at each recursion step, the total amount of NBV evaluations is reduced when compared against evaluating the entire grid. Moreover, controlling the cardinality and geometric extension of $\mathcal{S}$ provides a mechanism for balancing the divergence of our path from a straight line motion in favor of attaining more favorable viewing positions. Second, by using an auxiliary function PathQry$(\cdot)$ as an indicator function for intermediate viewpoint validity, the low level kinematic considerations of path planning are decoupled from the task of designing a suitable sensing strategy. In other words, vision related constraints can be addressed by the NBV evaluation function, while robotic motion considerations are to be handled by PathQry$(\cdot)$. In practice, we implement a global path planning approach based on the A^* algorithm [11]. However, alternative path planning techniques can be considered based on computational cost and completeness considerations.

B. Controlling Path Plan Execution

Once a sensing path $\mathcal{P} = \{\nu_j : j = 1 \dots n\}$ has been determined, robot motion towards the NBV $\nu^* = \nu_n$ initiates and sensing actions are executed as specified by the planned intermediate viewpoints. After each image acquisition our model $\mathcal{M} = \{\cup P_i : i = 1 \dots N\}$ is updated by a Kalman filter framework (see Section III). Under such an estimation framework it is possible to predict the state of the model after a single sensing action from ν^*. In turn, this estimation $\mathcal{M}_{\nu^*}$ can be compared with each of the sequentially updated models to determine whether the benefit of reaching the original NBV has been already met by the accumulation of intermediate sensing actions.

We propose the use of a utility function to describe the total reduction of 3D uncertainty for the model region $\mathcal{M}_{\nu^*} \subset \mathcal{M}$ observed by ν^*. Let $\gamma : \mathbf{R}^{3\times3} \rightarrow \mathbf{R}$ denote a criterion function describing the total 3D uncertainty of a given covariance matrix $\mathbf{\Sigma}$. We define $\gamma(\mathbf{\Sigma})$ =det($\mathbf{\Sigma}$), since this measure is equivalent to the product of the eigenvalues of $\mathbf{\Sigma}$, providing an approximate quantification of the uncertainty volume described by $\mathbf{\Sigma}$. Moreover, this criterion corresponds to the concept of D-optimality found in the experimental design literature [18]. Let $\mathbf{\Sigma}^P$ and $\mathbf{\Sigma}^P_{\nu^*}$ respectively denote the current and predicted 3D covariance matrices for primitive P. In this way, we define an utility function of the form

$$U(\mathcal{M}_{\nu^*}, \mathcal{M}) = \sum_{P \in \mathcal{M}_{\nu^*}} \gamma(\mathbf{\Sigma}^P) - \gamma(\mathbf{\Sigma}^P_{\nu^*}).$$

Note that the difference between $\gamma(\mathbf{\Sigma}^P)$ and $\gamma(\mathbf{\Sigma}^P_{\nu^*})$ for a given primitive P, will become negative when the current uncertainty volume represented by $\mathbf{\Sigma}^P$ is smaller than the volume for the predicted covariance $\mathbf{\Sigma}^P_{\nu^*}$. By defining our utility function as the sum of differences, we consider

ExecuteSensing($\mathcal{P}$):

// Predict 3D model after path execution
// to determine *path-specific* performance goals
$\nu^* \leftarrow$ LastElement($\mathcal{P}$)
$\mathcal{M}_{\nu^*} \leftarrow$ PredictModel(ν^*)
`while` $\mathcal{P} \neq \emptyset$
  // Move to next viewpoint and sense environment
  $\nu \leftarrow$ FirstElement($\mathcal{P}$)
  $\mathcal{P} \leftarrow \mathcal{P} \setminus \nu$
  MoveTo(ν)
  SenseEnvironment()
  // Update internal 3D model
  $\mathcal{M} \leftarrow$ UpdateModel()
  // Halt execution if *task-level* goals are met
  `if` PrecisionLevel($\mathcal{M}$) $< \rho$ `return` TRUE
  //Determine a novel NBV if either
    // the *path-specific* performance goals are met,
    // the path is not valid after model update,
    // the end of the original path has been reached
  `if` $U(\mathcal{M}_{\nu^*}, \mathcal{M}) < \eta$
    `OR` $\neg\exists$PathQry($\mathbf{x}(\nu), \mathbf{x}(\nu^*)$)
    `OR` $\mathcal{P} = \emptyset$
      $\nu^* \leftarrow$ ComputeNBV()
      $\mathcal{P} \leftarrow$SplitPath($\mathbf{x}(\nu), \mathbf{x}(\nu^*)$)
      $\mathcal{M}_{\nu^*} \leftarrow$ PredictModel(ν^*)

Fig. 3. Pseudo-code for ExecuteSensing(). A predetermined path $\mathcal{P}$ is executed by traversing trough a sequence of viewpoints while updating an internal 3D model. Path is recalculated when an utility threshold is reached or the original path is no longer feasible.

negative values of our utility function as favorable as they represent a reduction in uncertainty of the patches contained in $\mathcal{M}_{\nu^*}$. Accordingly, once we define a threshold value η for our utility function $U(\cdot)$, we can monitor the evolution of the function values and decide to halt the execution of the sensing plan. In this way, the determination of the NBV ν^* serves the double purpose of providing the required input for complete task specification as well as explicitly defining qualitative goals for the sensing plan.

The effects of model updates during path execution are two fold. First, augmentation of our internal model representation by sensing novel regions of the environment may render the original path unfeasible or cast regions of the object unobservable as obstructions may be detected. Second, if the internal 3D model updates are of sufficient scale as to significantly modify the 3D geometry of our internal model representation, the NBV may be rendered ineffective.

The feasibility of the current path towards the NBV can be readily evaluated after each augmentation to the internal model $\mathcal{M}$ and a novel NBV can be computed if the current path is rendered unreachable. Dealing with regions that are rendered unobservable by model augmentation is more complicated if those regions are the main focus of attention of the NBV. In that case, the 3D uncertainty of such regions will not be updated trough the course of path execution and an alternative NBV will need to be determined after path completion, when subsequent evaluations of our NBV criterion will consider the novel visual occlusion constraints.

C. Planning Completeness

The usefulness of an automated planner is dependent on its ability to avoid degenerate behavior. In this respect, the capability of our planner to escape local minima in the search space defined by our NBV criterion function (8) is essential for the achievement of planning convergence. Without loss of generality we can study the case of a single primitive and for the analysis of equal consecutive viewpoints since the Kalman filter guarantees that the cost function is not increased for any other primitive of the model.

Proposition 1. For a scene consisting of a single convex object, the use of the proposed Kalman filter framework for image measurement fusion, along with our NBV planning criterion, eliminates stagnation in the iterative process of viewpoint selection.

Proof. Our proof relies on showing that consecutive sensing actions from the same optimal NBV will reduce the value of criterion evaluation at that viewpoint. Moreover, iterated measurements from the current viewpoint will cause the local maximum in our criterion function (8) to be *consumed* and allow for a new global optimum to be considered as the NBV. This is achieved when

$$F(P^t, \nu^*) > F(P^{t+1}, \nu^*). \tag{10}$$

Here, we will present a brief sketch of our proof. Note that for consecutive sensing actions from the same viewpoint, the effects of texture thresholding $h(P)$ and projected area measurement $g(P, \nu)$ can be obviated, as the first is a constant function for each primitive while the second is a viewpoint dependent function. Accordingly, they both remain constant for repeated measurements from the same position. Hence, for the considered scenario and from the definitions of w (9) and $f(P, \nu)$ (4) we have the function profile of (8) given by

$$F(P, \nu) = \frac{cw}{f(P, \nu)} = \frac{c\lambda_1\lambda_2\lambda_3}{\|\mathbf{v}^T\,[\lambda_1\mathbf{e}_1\ \lambda_2\mathbf{e}_2\ \lambda_3\mathbf{e}_3]\,\|_2}. \tag{11}$$

Since $\mathbf{v}$ and $\mathbf{e}_j$ are unit vectors, an analysis of the partial derivatives of $F(P, \nu)$ indicates that decreasing the magnitude of the eigenvalues also reduces the value of (11). Finally, given that the utilized Kalman filtering framework provides non-increasing uncertainty estimation (see Lemma 1), and assuming discardable changes to the orientation of the 3D uncertainty, the eigenvalues are indeed decreased by iterated measurements, which in conjunction with (11) proves (10)$_\square$.

Proposition 1 allows for the normal operation of the planner not to get stuck on local minima of the NBV search space. In practice, this property holds even though the actual criterion is computed from an aggregation multiple primitives. This is due to the fact that primitives outside the visual field of the current viewpoint are not affected by our EKF framework, yielding a fixed 3D uncertainty estimate.

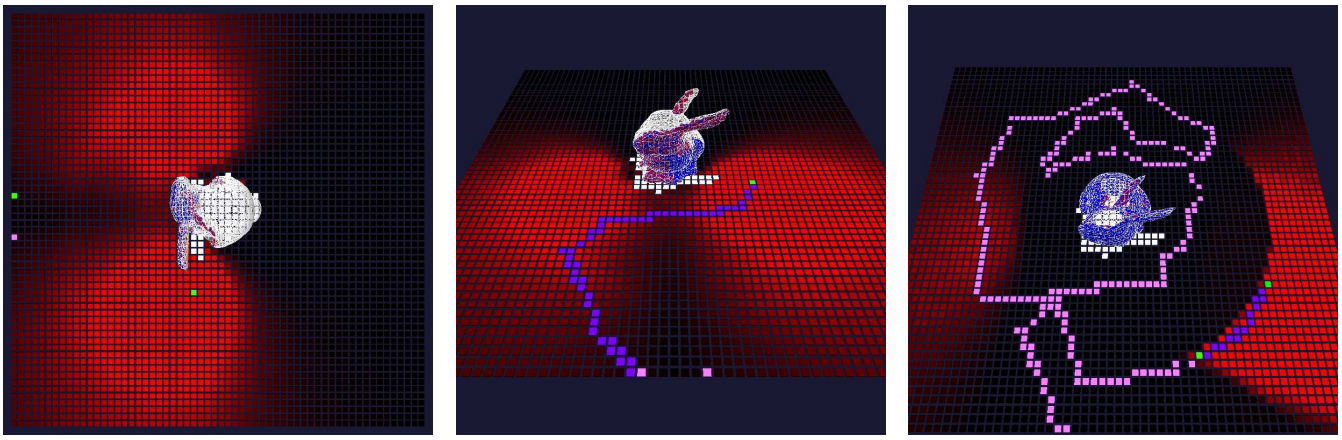

Fig. 4. Planning from an initial configuration. At left, the navigational grid is depicted with the two initial viewing configurations highlighted as well as the NBV computed for the current 3D model. At center, the recursively interpolated sensing path between the initial viewing configuration and the NBV is illustrated along with the predicted model. At right, the complete sensing path obtained after several iteration of the NBV planner. Higher uncertainty levels of the reconstructed model surface are rendered as red patches while more precise patches are rendered blue.

VI. Experiments

We present experimental results on a synthetic environment. The object under observation is a 3D model located at the center of a $M \times M$ grid. Image acquisition and measurements were simulated through OpenGL rendering with synthetic lighting as the only source of surface texture. The rendered model consisted of 70K triangles and two initial images with a small baseline were simulated to obtain an input 3D model for our NBV planner. The simulated scenario consists of an open environment where the only inaccessible regions of the grid are those occupied by our object model.

The goal of our reconstruction is to achieve a reconstruction precision of 100 times greater than 2% of the a viewing distance of 40 units. This viewing distance value corresponds to configurations where the entire rendered model occupies half the image viewing area, while 2% is in practice a typical error range for binocular stereo depth estimation. The two termination criteria for our algorithm are: 1) Achieving 99% object coverage and 2) attainting an average 3D uncertainty value $\sigma < 0.008$ units across all primitives. Such precision values can be readily obtained from the covariance structure of each primitive.

Viewpoint evaluation works at a rate of 130Hz on a laptop powered by a 2.4 GHZ Centrino processor with 2GB of RAM and an Nvidia Quadro 570M graphics card. In the presented experiments, the grid resolution is specified as $M = 60$, allowing for the determination of the NBV by means of complete grid evaluation with a processing time of under 30 seconds. Alternatively, we have developed an evolutionary computation based global optimizer for our NBV criterion (not presented here). However, for the our simulation scenario full grid evaluation is adopted, as it assures the attainment of a global NBV.

A. Experimental Results

Figure 4 depicts the initial viewpoint configuration used in the experiment. The NBV ν^* is highlighted (i.e. the cell with the minimal cost value) and the set of intermediate viewpoints determined by our recursive path planning approach is depicted. This sensing plan will be followed until path completion or the achievement of an utility threshold due to intermediate sensing and model updates. In this scenario the sampling rate is set at one sample for each single cell displacement. However, the sampling rate is a tunable parameter not inherently restricted by the resolution of the path planning grid, but instead determined by the operational characteristics of the mobile platform as well the computational throughput of the image processing infrastructure. The availability of near real-time 3D reconstruction systems allows flexibility in specifying a reasonable sampling rate.

On the right of Figure 4 the results of our sensor planner are illustrated after achieving a 99% coverage of the object's visible area as well as precision level of 0.002% with respect to a viewing distance of 40 units. These *task level* qualitative goals were reached after 270 imaging samples. The geometry of the obtained sensing path highlights the properties of our planning approach. Namely, the path initially favors viewpoints in close proximity of the object, since high resolution measurements of the object surface provide improved reconstruction accuracy. Due to the 3D object's shape, these initial images are unable to favorably sense the regions near the top of the object. Accordingly, a small nearly horizontal region on top of the object can not be observed from any cell in the grid and, as such, it is not considered part of the visible object surface. Moreover, the initially higher 3D uncertainty found in the upper regions of the object caused the planner to guide the selection of subsequent NBV's away from the object to bring those regions in to view. In fact, the final landscape for our NBV criterion function reflects this property. The displayed final path is achieved by 12 iterations of our sense-replan approach.

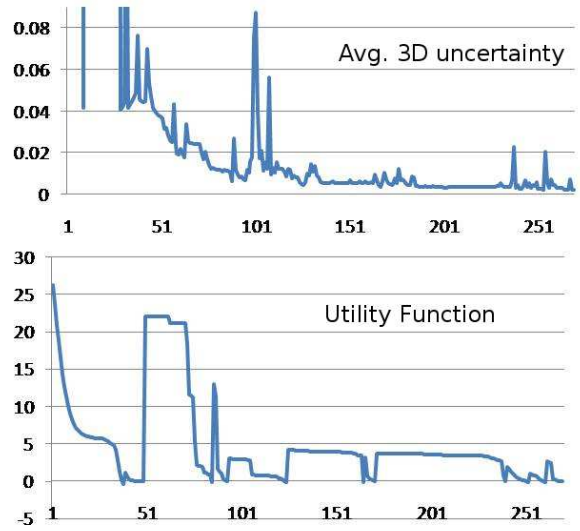

Fig. 5. Performance of the sensing strategy. Above, average standard deviation of 3D structure estimates across entire model. Below, utility function used to monitor achievement of path-level goals. In both cases the horizontal axis represents the sample count along the sensing path.

It is noteworthy, that while our planner does achieve a complete circuit around the object, it does present oscillation at a large scale of displacements. Large scale oscillation can be attributed to the fact that successfully improving the reconstruction accuracy on the currently sensed region may render previously sensed regions in need of additional sensing in order to compensate. Such behavior is inherent to the incremental and *greedy* nature of a NBV based approach. The lower portion of Figure 5 depicts the evolution of the utility function used to verify the achievement of path specific goals. Once a lower bound threshold is reached, a novel NBV is determined along with an updated sensing path. The sharp increases in the graph coincide with the determination of novel NBV's and the corresponding set of new path specific goals. The upper portion of Figure 5 depicts the downward trend of the uncertainty in our estimated 3D model. The non monotonic behavior of the graph is caused by the inclusion of novel object regions into our model. Newly sensed regions represent higher uncertainty as they are generally sensed from a small baseline stereo configuration (or perhaps from a degenerate stereo pair obtained from forward motion). A reactive NBV approach (i.e. recomputing the NBV after each sensing action) would essentially be guided by these abrupt changes in model uncertainty and compromise the systematic achievement of task level goals. Our approach avoids these difficulties by the inclusion of the aforementioned path specific sensing goals.

VII. Discussion and Future work

We have presented an approach for developing sensing strategies for autonomous 3D reconstruction. The combination of the proposed cost function and a recursive path planner have yielded satisfactory results on simulated environments. It is noteworthy that our planner exhibits an exploratory behavior although our NBV criterion is strictly driven by uncertainty reduction. This property may be deemed a consequence of observing an object centered scene. However, the criterion implicitly penalizes revisiting viewing configurations as they offer limited uncertainty reduction potential. The generic nature of our 3D model representation and the computational efficiency our approach makes it well suited for future real world experimentation. Among additional research goals we include further formal analysis of the computational bounds of our approach, as well as exploring the use of dynamic path planning framework similar to the one presented in [16].

Acknowledgements

We thank Dr. Simon Julier for the insights used on the proof of Lemma 1. Enrique Dunn was supported by CONACyT foreign posdoctoral visit award No. 82053.

References

[1] P. S. Blaer and P. K. Allen. Two stage view planning for large-scale site modeling. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 814–821, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[2] S. Chen, Y. Li, J. Zhang, and W. Wang. *Active Sensor Planning for Multiview Vision Tasks*. Springer-Verlag, Berlin Heidelberg, 2008.

[3] C. S. Fraser. Network design considerations for non-topographic photogrammetry. *PERS*, 50(8):1115–1126, 1984.

[4] C. S. Fraser. Network design. In *Close Range Photogrammetry and Machine Vision. (Ed. K. B. Atkinson). Whittles, Caithness. 371 pages*, pages 256–281, 1996.

[5] A. Hornung, B. Zeng, and L. Kobbelt. Image selection for improved multi-view stereo. In *Proccedings of CVPR*, pages 1–8, 2008.

[6] K.-L. Low and A. Lastra. Efficient constraint evaluation algorithms for hierarchical next-best-view planning. In *International Symposium on 3D Data Processing Visualization and Transmission*, pages 830–837, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[7] E. Marchand and F. Chaumette. Active vision for complete scene reconstruction and exploration. *PAMI*, 21(1):65–72, 1999.

[8] S. Mason. Heuristic reasoning strategy for automated sensor placement. *PERS*, 63(9):1093–1102, 1997.

[9] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(5):417–433, 1993.

[10] T. Newman and A. Jain. Survey of automated visual inspection. *Computer Vision and Image Understanding*, 61(2):231–262, 1995.

[11] N. J. Nilsson. *Principles of Artificial Intelligenc*. Tioga Publishing Company, 1980.

[12] G. Olague and R. Mohr. Optimal camera placement for accurate reconstructions. *Pattern Recognition*, 35(4):927–944, 2002.

[13] R. Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 1999.

[14] R. Sablatnig, S. Tosovic, and M. Kampel. Next view planning for a combination of passive and active acquisition techniques. In *Proceedings of 3DIM*, page 6269, 2003.

[15] W. R. Scott, G. Roth, and J.-F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1):64–96, 2003.

[16] A. Stentz. Optimal and efficient path planning for partially-known environments. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3310–3317, 1994.

[17] K. Tarabanis, P. Allen, and R. Stag. A survey of sensor planning in computer vision. *IEEE Trans. on Rob. and Automat.*, 11(1):86–104, 1995.

[18] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann. Active visual object reconstruction using d-, e-, and t-optimal next best views.

[19] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, and J. Denzler. An information theoretic approach for next best view planning in 3-d reconstruction. In *Proccedings of ICPR*, 2006.

[20] P. Whaite and F. Ferrie. Autonomous exploration: driven by uncertainty. *PAMI*, 19(3):193–205, 1997.