

Interactive Sensor Planning *

To appear at the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998

Ioannis Stamos and Peter K. Allen

Department of Computer Science, Columbia University, New York, NY 10027
{istamos, allen}@cs.columbia.edu

Abstract

This paper describes an interactive sensor planning system that can be used to select viewpoints subject to camera visibility, field of view and task constraints. Application areas for this method include surveillance planning, safety monitoring, architectural site design planning, and automated site modeling. Given a description of the sensor's characteristics, the objects in the 3-D scene, and the targets to be viewed, our algorithms compute the set of admissible view points that satisfy the constraints. The system first builds topologically correct solid models of the scene from a variety of data sources. Viewing targets are then selected, and visibility volumes and field of view cones are computed and intersected to create viewing volumes where cameras can be placed. The user can interactively manipulate the scene and select multiple target features to be viewed by a camera. The user can also select candidate viewpoints within this volume to synthesize views and verify the correctness of the planning system. We present experimental results for the planning system on an actual complex city model.

1 Introduction

Automatic selection of camera viewpoints is an important problem in computer vision tasks. In cluttered and complex environments such as urban scenes, it can be very difficult to determine where a camera should be placed to view multiple objects and regions of interest. It is important to note that this camera placement problem has two intertwined components. The first is a purely geometric planner that can reason about occlusion and visibility in complex scenes. The second component is an understanding of the optical constraints imposed by the particular sensor (i.e. camera) that will affect the view from a particular chosen viewpoint. These include depth-of-field, resolution of

the image, and field-of-view, which are controlled by aperture settings, lens size and focal length. To properly plan a correct view, all of these components must be considered.

The focus of this paper is to extend our earlier sensor planning results [1, 11, 12] to urban scene planning. Urban environments are characterized by cluttered and complex object models which places a heavy emphasis on 3-D occlusion planning. In addition, these models themselves may be partial or incomplete, and lack topological relations that are central to performing the planning task. Application areas for these methods include surveillance planning, safety monitoring, architectural site design planning, and choosing viewpoints for automatic site modeling.

Related previous work on the geometric planning component includes computational geometry algorithms for determining visibility with much of the work focusing on 2-D visibility algorithms [8]. Other work includes Gigus and Canny [5] with aspect graphs, and work by Coorg [3] on efficient overestimation of visible polygons. Also, Bern [2] discusses visibility with a moving point of view.

Work most closely related to ours in integrating sensor and visibility constraints in 3-D includes the work of [10, 7, 4]. These systems have focused on highly constrained and well-understood environments for which accurate and complete object models exist.

The core of our system is a sensor planning module which performs the computation of the locus of *admissible* viewpoints in the 3-D space with respect to a 3-D model of objects and a set of target features to be viewed. This locus is called the *Visibility Volume*. At each point of the visibility volume a camera has an unoccluded view of all target features, albeit with a possibly infinite image plane. The finite image plane and focal length constraints will limit the field of view, and this imposes a second constraint which leads to the computation of *field of view cones* which limit the minimum distance between the sensor and the target for each camera orientation. The integration of those

*This work was supported in part by an ONR/DARPA MURI award ONR N00014-95-1-0601, DARPA AASERT awards DAAHO4-93-G-0245 and DAAH04-95-1-0492, and NSF grants CDA-96-25374 and IRI-93-11877.

two constraints leads to a volume of *candidate* viewpoints.

This core is part of a larger system that is being built to automatically create models of urban environments, and plan sensor placements to build site models, update existing models, and detect change in existing models. This paper describes an interactive graphical system where sensor planning experiments are performed. This system allows us to generate, load and manipulate different types of scenes and interactively select the target features that must be visible by the sensor. The results of the sensor planning experiments are displayed as 3-D volumes of viewpoints that encode the constraints. Virtual sensors placed in those volumes provide a means of synthesizing views in real-time and evaluating viewpoints.

2 Visibility Planning

The computation of the visibility volume involves the computation of the boundary of the *free space* (the part of the 3-D space which is not occupied by objects) and the boundary between the visibility volume and the *occluding volume*, which is the complement of the visibility with respect to the free space. In order to do that we decompose the boundary of the scene objects into convex polygons and compute the *partial* occluding volume between each convex boundary polygon and each of the targets which are assumed to be convex polygons. Multiple targets can be planned for, and the system can handle concave targets by decomposing them into convex regions. We discard those polygons which provide redundant information, thus increasing the efficiency of our method. The boundary of the intersection of all partial visibility volumes (see next paragraphs) is guaranteed to be the boundary between the visibility and the occluding volume. The boundary of the free space is simply the boundary of the scene objects.

The locus of occlusion-free viewpoints with respect to the 3-D solid model of the scene objects $U = \{u_1, u_2, \dots, u_m\}$ and the set of target polygonal features $T = \{t_1, t_2, \dots, t_n\}$ is the visibility volume $V(U, T)$. Each target feature t_i is a 2-D connected part of the scene's boundary $\mathcal{B}(U)$. All solid models are described using polyhedral boundary representation. The complement of the visibility volume with respect to the free space $F(U)$ (open set in space which is not occupied by any objects) is the occluding volume $O(U, T)$, that is $O(U, T) = F(U) - V(U, T)$. Both $O(U, T)$ and $V(U, T)$ are open sets.

We can create the polyhedral cones C_i whose apex is a point P_f of the free space and whose bases are the polygonal targets t_i ($i = 1, \dots, n$). If the point P_f

belongs to the visibility volume then all cones do not intersect any object of our scene. If on the other hand P_f belongs to the occluding volume then at least one cone intersects at least one object of our scene (the result of this intersection is a 3-D volume). When P_f belongs to the common boundary surface between the visibility and the occluding volume, then at least one cone is tangent to at least one object of the scene and no cone intersects any object of the scene. This boundary surface $\mathcal{S}(U, T)$ together with the boundary of the free space uniquely characterize the visibility volume, using perspective projection.

In order to compute the visibility volume for all targets T , we can compute the volume for each individual connected target and then intersect the final results:

$$V(U, T) = V(U, t_1) \cap V(U, t_2) \cap \dots \cap V(U, t_n) \quad (1)$$

In the computation of the $V(U, t_i)$ not all boundary faces of U have to be used. Only points which lie in the half-space, defined by the plane of t_i towards the direction of its outward-pointing normal, are candidates for the visibility volume. That means that only the part of the boundary of U which lies in this half space is relevant to the visibility computation. If this part of the boundary consists of the planar polygonal faces f_j ($j = 1, \dots, N$), then

$$V(U, t_i) = V(f_1, t_i) \cap V(f_2, t_i) \cap \dots \cap V(f_N, t_i) \quad (2)$$

The computation of a *partial* visibility volume $V(f_j, t_i)$ between an object face and a target feature is the core of the visibility computation. The polygonal boundary surface $\mathcal{S}(f_j, t_i)$ of this volume can be computed in *linear* time in the total number of vertices of the object face and the target feature [13]. This surface consists of two parts: $\mathcal{S}(f_j, t_i) = f_j \cup \mathcal{S}_o(f_j, t_i)$ (f_j is the boundary of the free space, while $\mathcal{S}_o(f_j, t_i)$ is the visibility-occluding volume boundary).

The *local separating planes* Π_i defined by an object's edge e_o and a target's vertex v_t (case I) or by a target's edge e_t and an object's vertex v_o (case II), are the planes lying on the boundary of the partial visibility volume. The distinguishing attribute of a local separating plane is that it separates the object face and the target feature into two different half-spaces. In case I the boundary face of the visibility volume is an unbounded trapezoid having e_o as its base and in case II it is an unbounded triangle having v_o as its apex. Those faces lie on the corresponding local separating plane. Details can be found in [13].

3 Field of View

A viewpoint which lies in the visibility volume has an unoccluded view of all target features in the sense

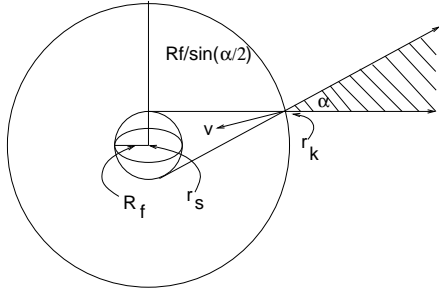


Figure 1: Field of view cone (shaded region) for viewing direction \mathbf{v} and field of view angle a . The targets are enclosed in the sphere of radius R_f .

that all lines of sight do not intersect any object in the environment. This is a *geometric* constraint that has to be satisfied. Visual sensors however impose *optical* constraints having to do with the physics of the lens (Gaussian lens law for thin lens), the finite aperture, the finite extent of the image plane and the finite spatial resolution of the resulting image formed on the image plane, as well as lens distortions and aberrations.

We now consider the *field of view* constraint which is related to the finite size of the active sensor area on the image plane. The targets t_i are imaged if their projection lies entirely on the active sensor area on the image plane. This active sensor area is a 2-D planar region of finite extent. Thus the projection of the target features in their entirety on the image plane depends not only on the viewpoint P_f , but also on the orientation of the camera, the effective focal length and the size and shape of the active sensor area. Those parameters control the position and orientation of the active sensor area in space.

For a specific field of view angle a and a specific viewing direction \mathbf{v} we compute the locus of viewpoints which satisfy the field of view constraint for the set of targets T . If we approximate the set of targets with a sphere S_f of radius R_f and of center \mathbf{r}_s containing them, then this locus is a circular cone $C_{fov}(\mathbf{v}, a, \mathbf{r}_s, R_f)$, called the field of view cone (figure 1). The cone axis is parallel to \mathbf{v} and its angle is a . Viewpoints can translate inside this volume (the orientation is fixed) while the targets are imaged on the active sensor area. The locus of the apices of these cones for all viewing directions is a sphere S_{lim} whose center is \mathbf{r}_s and its radius is $R_f/\sin(a/2)$ (figure 1). For every viewpoint lying outside of this sphere there exists at least one camera orientation which satisfy the

field of view constraint, since this region is defined as:

$$\bigcup_{\forall \mathbf{v}} C_{fov}(\mathbf{v}, a, \mathbf{r}_s, R_f)$$

For viewpoints inside the sphere S_{lim} there does not exist any orientation which could satisfy the field of view constraint (the camera is too close to the targets). The approximation of the targets by a sphere simplifies the field of view computation. It provides, however, a conservative solution to the field of view problem since we require the whole sphere to be imaged on the active sensor area.

The field of view angle for a circular sensor having a radius of I_{min} , is $a = 2 \tan^{-1}(I_{min}/2f)$, where f is the effective focal length of the camera. For rectangular sensors the sensor area is approximated by the enclosing circle. The field of view angle a does not depend on the viewpoint or the orientation of the camera.

4 Intersecting the Two Constraints

The locus of viewpoints which satisfy more than one constraint is calculated by intersecting the locus of viewpoints which independently satisfy each individual constraint. Integrating the loci calculated in the previous sections we have:

$$I(U, T, \mathbf{v}, a) = V(U, T) \cap C_{fov}(\mathbf{v}, a, T)$$

where $I(U, T, \mathbf{v}, a)$ is the integrated locus (candidate volume), when the viewing direction is \mathbf{v} and the field of view angle is a . Both the visibility volume and field of view cone are represented as solid CAD models. The integrated locus is the result of a boolean operation (intersection) between solids. Intuitively, this solid is the result of the intersection of the visibility volume with a field of view cone. Examples of these regions are given in section 6.

5 Interactive System Components

5.1 Model Translator: Graphics \rightarrow CAD

In most cases, existing urban/city models are graphics models that have no need to be topologically consistent and geometrically correct, since 2-D viewing is the main application. Those models are not guaranteed to be complete or to correspond to a proper polyhedron [9], since they lack topological information. Our planner uses solid models having a Polyhedral Boundary Representation (i.e. Quad Edge Data Structure [6]). So we need to be able to transform these existing graphics-based models to a solid, watertight boundary representation (i.e. no dangling faces). A common problem is that models are not watertight. Often, parts of the boundary are missing

and the object is not bounded or closed. Also, the topological data which describes the adjacency information between faces is inconsistent with the physical 3-D world. In addition, the direction of surface normals can be incorrect.

If each edge of the graphics-model is shared by exactly two planar faces and if adjacent faces have opposite orientation (the orientation of each face is the counter-clockwise ordering of its vertices with respect to its normal vector) then the graphics-model is a proper bounded polyhedron. Our method recovers the adjacency information between the faces of the graphics model and checks if the previous conditions are satisfied. In this case the result is a correct solid model. Using this method, we have successfully taken incomplete models of sites built graphically, and used them in our sensor planning experiments (see section 6).

5.2 Model Translator: CAD \rightarrow Graphics

This part of the system transforms the solid CAD models to graphics rendering models that can be used in the interactive system. The translation from a solid CAD model to a Graphics model involves the extraction of the polygonal faces of the solid CAD model and their syntactic transformation to a rendering format. No topological information is needed by the graphics model. For each scene object both the graphics and the CAD model is maintained by our system.

5.3 Sensor Planner and Camera Selection

The user interacts with the scene through the graphics models, which can be interactively manipulated. All actions are propagated to the CAD modeler where the boolean operations between models are performed and where the sensor planning algorithms are implemented.

The user selects the target features on the boundary of the scene model and the part of the scene which is going to be the occluding object. First the visibility volume (see section 2) is computed and displayed overlaid on the scene. After that the user selects a camera orientation \mathbf{v} and a field of view angle a and the corresponding field of view cone is computed (see section 3) and displayed. Finally, the intersection of the previous volumes is computed and this is the final result (candidate viewpoints).

The camera selection module allows a virtual camera to be placed and oriented inside the set of *candidate* viewpoints. The camera's field of view angle is interactively set by the user. The resulting image can be used to verify the correctness of the method. Sensor Planning experiments can be performed in real-time using this system.

6 Experimental Results

We have tested the system using a site model of Rosslyn, Virginia. Our initial input was a textured-mapped VRML model of the city provided by GDE Systems Inc (<http://gdesystems.com> - see Figure 2a). Using our model translator we transformed it to a solid CAD model (figure 2b) which consisted of 488 solid blocks. We applied the sensor planning algorithms to a part of this model whose boundary consisted of 566 planar faces.

In the first experiment (figure 3a) one target (black face) is placed inside the urban area of interest. The visibility volume is computed and displayed (transparent polyhedral volume). For a viewing direction of $\mathbf{v}_1 = (0^\circ, 22^\circ, 0^\circ)$ (Euler angles with respect to the global Cartesian coordinate system) and field of view angle of $a_1 = 44^\circ$, the field of view locus is the transparent cone on the left. The set of candidate viewpoints $I_1(\mathbf{v}_1, a_1)$ (intersection of visibility with field of view volume) is the partial cone on the left. For a different viewing direction $\mathbf{v}_2 = (0^\circ, 91^\circ, 0^\circ)$ the set of candidate viewpoints $I_1(\mathbf{v}_2, a_1)$ is the partial cone on the right.

In the second experiment (figure 3b) a second target is added so that two targets (black planar faces) need to be visible. The visibility volume, the field of view cone for the direction \mathbf{v}_1 and the candidate volumes $I_2(\mathbf{v}_1, a_1)$ (left) and $I_2(\mathbf{v}_3, a_1)$ (right) are displayed. The viewing orientation \mathbf{v}_3 is equal to $(0^\circ, 71^\circ, 0^\circ)$. The visibility volume and the candidate volume $I_2(\mathbf{v}_1, a_1)$ are subsets of the corresponding ones in the first experiment.

If we place a virtual camera inside the volume $I_1(\mathbf{v}_2, a_1)$ (point (300.90, 56.18, 325.56)), set the field view angle to a_1 and the orientation to \mathbf{v}_2 , then the synthesized view is displayed on figure 4a. The target is clearly visible. Placing a virtual camera outside of the visibility volume (point (509.92, 41.70, 366.80)) results in the synthesized view of figure 4b. Clearly the target is occluded by one object of the scene. The orientation of the camera is $(0^\circ, 84^\circ, 0^\circ)$ (for every viewpoint outside the visibility volume there does not exist any camera orientation that would result in an unoccluded view of the target). If we place a virtual camera on the boundary of the the candidate volume $I_1(\mathbf{v}_2, a_1)$ (point (375.59, 52.36, 348.47)), then in the resulting synthesized view (figure 4c) we see that the image of the target is tangent to the image of one object of the scene. Again the camera orientation is \mathbf{v}_2 and the field of view angle a_1 .

In figure 4d we see a synthesized view when the camera is placed on the conical boundary of the can-

candidate volume $I_2(\mathbf{v}_3, a_1)$. The camera's position is (159.42, 30.24, 347.35). The transparent sphere is the sphere S_f (see section 3) used to enclose the targets. We see that S_f is tangent to the bottom edge of the image, because the viewpoint lies on the boundary of the field of view cone. Finally the figure 4e has generated by a camera placed on the polyhedral boundary of the candidate volume $I_2(\mathbf{v}_3, a_1)$ (position (254.78, 49.28, 350.45)).

7 Conclusions

We have implemented an interactive system where sensor planning experiments can be performed in real-time for complex urban scenes. The system can compute visibility and field of view volumes as well as their intersection. The resulting locus consists of viewpoints which are guaranteed to be occlusion-free and places targets within the field of view. Object models and targets can be interactively manipulated and camera positions and parameters selected to generate synthesized images of the targets that encode the viewing constraints. Given site models of scenes, the system can be used to plan view positions for a variety of tasks including surveillance, safety monitoring, and site design. We are currently extending this system to include resolution constraints and as a planner for mobile site navigation to acquire models of scenes.

References

- [1] S. Abrams, P. K. Allen, and K. Tarabanis. Computing camera viewpoints in a robot work-cell. In *Proc. IEEE Intl. Conference on Robotics and Automation*, pages 1972–1979, Apr. 22–25 1996.
- [2] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point of view. *Algorithmica*, 11:360–378, 1994.
- [3] S. Coorg and S. Teller. Temporally coherent conservative visibility. In *Twelfth Annual ACM Symposium in Computational Geometry*, Philadelphia, May 1996.
- [4] C. K. Cowan and P. D. Kovesi. Automatic sensor placement from vision task requirements. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(3):407–416, May 1988.
- [5] Z. Gigus, J. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):542–551, June 1991.
- [6] L. Guibas and S. J. Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. *ACM Transactions on Graphics*, 4:74–123, 1985.
- [7] R. Niepold, S. Sakane, and Y. Shirai. Vision sensor set-up planning for a hand-eye system using environmental models. In *Proceeding Soc. Instrum. Control Eng. Japan*, Hiroshima, Japan, July 1987.
- [8] J. O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series on Monographs on Computer Science. Oxford University Press, New York, 1987.
- [9] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
- [10] S. Sakane, M. Ishii, and M. Kakiura. Occlusion avoidance of visual sensors based on a hand eye action simulator system: HEAVEN. *Advanced Robotics*, 2(2):149–165, 1987.
- [11] K. Tarabanis, P. K. Allen, and R. Tsai. The MVP sensor planning system for robotic vision tasks. *IEEE Transactions on Robotics and Automation*, 11(1):72–85, February 1995.
- [12] K. Tarabanis, R. Tsai, and P. K. Allen. Analytical characterization of the feature detectability constraints of resolution, focus and field-of-view for vision sensor planning. *Computer Vision, Graphics, and Image Processing*, 59(3):340–358, May 1994.
- [13] K. Tarabanis, R. Y. Tsai, and A. Kaul. Computing occlusion-free viewpoints. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 18(3):279–292, March 1996.

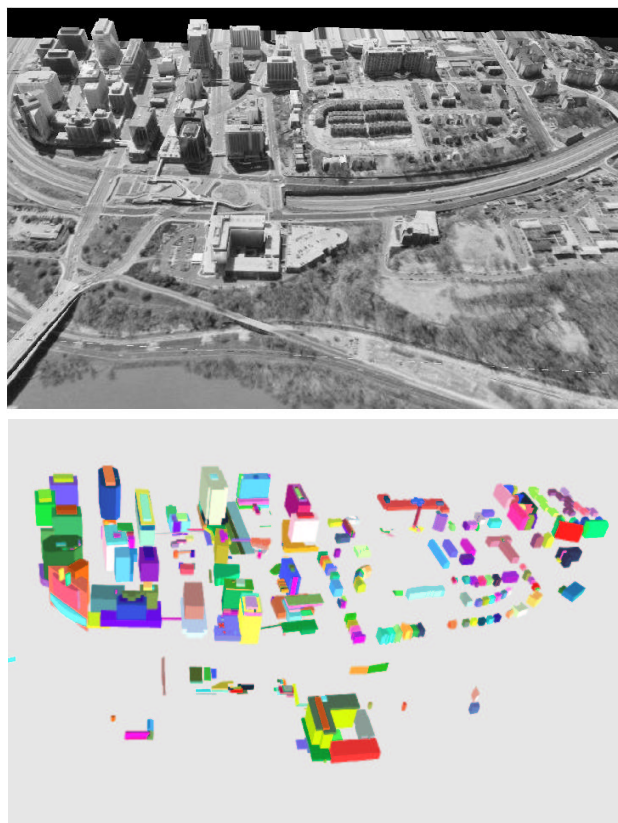


Figure 2: a) VRML Graphics model of Rosslyn, VA. b) Solid CAD model computed from the graphics model.

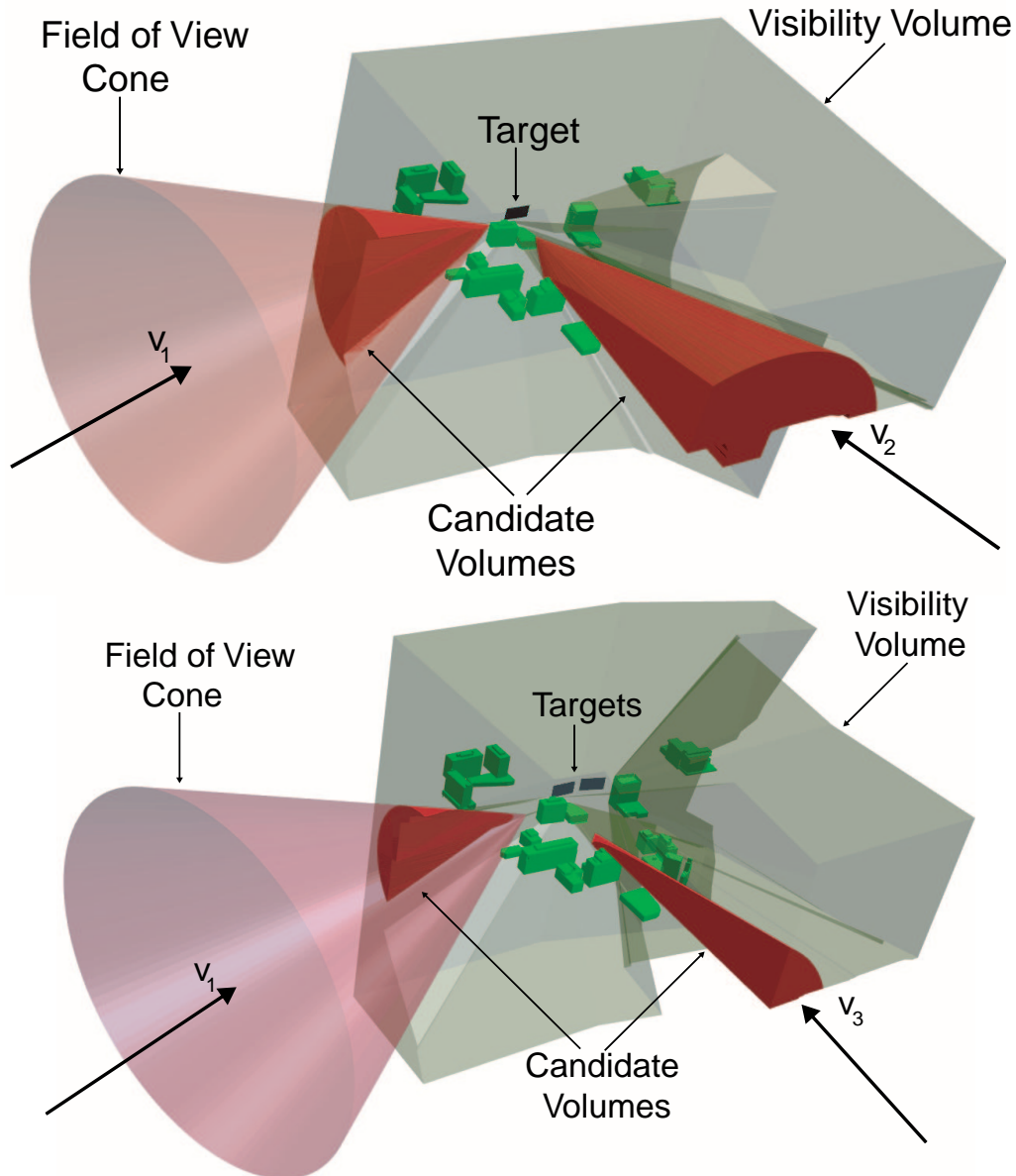


Figure 3: Two experiments. a) (top figure) One target and b) (bottom figure) two targets are placed in the urban area. The targets are planar faces. The Visibility Volumes (transparent polyhedral volumes), the Field of View Cones for the direction \mathbf{v}_1 (transparent cones) and the Candidate Volumes (intersection of the visibility volumes with the field of view cones) for the viewing direction \mathbf{v}_1 (left partial cones) and for the directions \mathbf{v}_2 (right partial cone, top figure) and \mathbf{v}_3 (right partial cone, bottom figure) are displayed. The Field of View Cones for the directions \mathbf{v}_2 (top) and \mathbf{v}_3 (bottom) are not shown

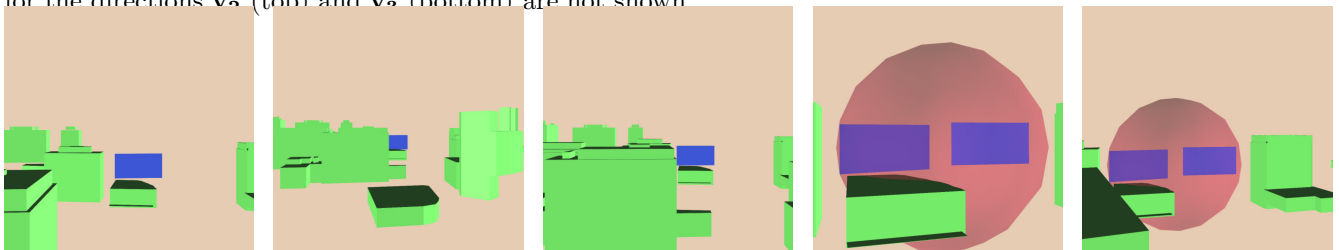


Figure 4: Synthesized views. Single target (black face): the camera is placed a) (left image) inside the candidate volume, b) out of the visibility volume and c) on the boundary of the candidate volume. Two targets: the camera is placed on d) the conical boundary and e) the polyhedral boundary of the candidate volume.