Computer Vision — CVWW'03, Ondřej Drbohlav (ed.)
Valtice, Czech Republic, February 3–6, 2003
Czech Pattern Recognition Society

Next View Planning for Shape from Silhouette

Robert Sablatnig, Srdan Tosovic, and Martin Kampel

Pattern Recognition and Image Processing Group,
Institute for Computer Aided Automation,
Vienna University of Technology,
Favoritenstraße 9/183/2, A-1040 Vienna, Austria

Abstract *In order to create a complete three-dimensional model of an object based on its two-dimensional images, the images have to be acquired from different views. An increasing number of views generally improves the accuracy of the final 3D model but it also increases the time needed to build the model. The number of the possible views can theoretically be infinite. Therefore, it makes sense to try to reduce the number of views to a minimum while preserving a certain accuracy of the model, especially in applications for which the performance is an important issue. This paper shows an approach to Next View Planning for Shape from Silhouette for 3d shape reconstruction with minimal different views. Results of the algorithm developed are presented for both synthetic and real input images.*

1 Introduction

One possibility for obtaining multiple views is to choose a fixed subset of possible views, usually with a constant step between two neighboring views, independent on the shape and the complexity of the object observed. This is illustrated in Figures 1a and 1b, which show a reconstruction of a corner of a square by drawing lines from the point O with a constant angle between two lines and connecting the points where the lines intersect the square. We can see that the corner reconstructed using 9 lines (Figure 1b) looks ”better” than the one reconstructed using 5 lines (Figure 1a), but also that neither of these two methods was able to reconstruct the corner perfectly. In addition to this, some of the views (20° in Figure 1a and 10°, 20°, 30°, 60° and 70° in Figure 1b) could have been omitted — without them the reconstruction of the corner in Figures 1a and 1b would have been exactly the same.

This simple example illustrates the need for selection of views based on the features of the object — this is called *Next View Planning* (in short, *NVP*). For the square from Figure 1, if we had a way of selecting the significant views only, we could reconstruct the corner of the square perfectly using 3 views only, as shown in Figure 1c.

A thorough survey of Next View Planning, also called *Sensor Planning*, is given in [19]. Tarabanis et al. [19], summarize the NVP problem as follows: ”Given the information about the environment (e.g., the object under observation, the available sensors) as well as the information about the task that the vision system is to accomplish (i.e., detection of certain object features, object recognition, scene reconstruction, object manipulation), develop strategies to automatically determine sensor parameter values that achieve this task with a certain degree of satisfaction”. Following this definition, in order to design an NVP algorithm for a given computer vision task, one has to identify the sensor parameters which can be manipulated (e.g., the position of the camera) and define the ”degree of satisfaction”, i.e., construct a metrics for evaluation of the parameter values proposed. The number of parameters that can be manipulated is also called the number of *degrees-of-freedom*. Increasing number of degrees of freedom increases the complexity of an NVP algorithm.

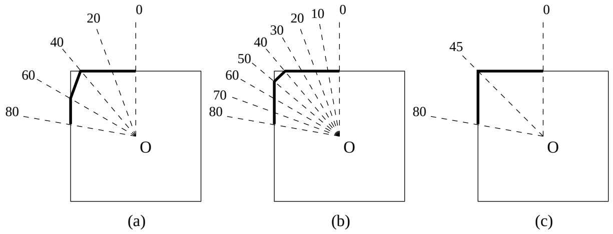


Figure 1: Reconstruction of a square corner

There are several computer vision tasks which can incorporate an NVP problem, differing in the necessary amount of an a priori knowledge about the object, the sensors and the environment:

- *Object feature detection*: here the goal of NVP is to determine the sensor parameters values for which the particular features of a known object in an image satisfy certain constraints, such as being visible and in-focus [5, 20]. A considerable amount of a priori knowledge about the approximate pose of the object and the environment is required.

- *Visual inspection*: this is a sub-area of object feature detection — a typical task of visual inspection is to determine how accurately a particular object has been manufactured [22, 21, 12]. A nearly perfect estimate of the geometry and the pose of the object have to be known.

- *Model-based object recognition*: in this area NVP tries to find the sensor parameter values which make it possible to identify an object and/or estimate its pose in a most accurate and efficient way [7, 8]. Based on models of sensors and possible objects, a search on object's identity and pose is performed, usually using the hypothesize-and-verify method: in the first step, the hypotheses regarding the object's identity and pose are formed; then, these hypotheses are evaluated according to certain metrics; finally, the new sensor parameter values are proposed based on a given criterion until a stopping condition is met.

- *Scene or object reconstruction*: in this case, the task of NVP is to find the best values of the sensor parameters in order to build a model of an unknown scene or object [11, 4, 14, 16]. A model is built incrementally, guided by the information about the scene/object acquired to this point. Usually there is no a priori known scene information.

The approach presented in this work falls into the category of object reconstruction. For this task, many different NVP strategies have been developed: Maver and Bajcsy [14] proposed an NVP algorithm for an acquisition system consisting of a light stripe range scanner and a turntable. They represent the unseen portions of the viewing volume as $2\frac{1}{2}$D polygons. The polygon boundaries are used to determine the visibility of unseen portions from all candidate next views. The view which can see the largest area unseen up to that point is selected as the next best view.

Connolly [4] used an octree to represent the viewing volume. An octree node close to the scanned surface was labeled as *seen*, a node between the sensor and this surface as *empty* and the remaining nodes as *unseen*. Next best view was chosen from a sphere surrounding the object. Connolly proposed two NVP algorithms: one called *planetarium*, which used a form of ray tracing to determine the number of unseen nodes from each candidate view and selected the one seeing the most unseen nodes, and a *normal* algorithm, which selected the next best view from 8 candidate positions only and did not take occlusions into account, and therefore was significantly faster.

Whaite and Ferrie [23] use the range data sensed so far to build a parametric approximate model of the object. The view from which the data fits the current model the worst is chosen as the next best view. This approach does not check for occlusions and does not work well with complex objects because of limitations of a parametric model.

Pito [16] uses a range scanner which moves on a cylindrical path around the object. He partitions the viewing volume into its *seen* and *unseen* portions, and defines the surface separating the two volume portions as *void surface*. This surface is approximated by a series of small rectangular oriented *void patches*. In his *positional space (PS)* algorithm, the next best view is chosen as the position of the scanner which samples as many void patches as possible while resampling at least a certain amount of the current model.

Our approach is based on the work of Liska [10], who uses a system consisting of two lasers projecting a plane onto the viewing volume and a turntable. The next best view (the next position of the turntable) is computed based on information from the current and the preceding scan. In each of the two scans the surface point farthest from the turntable's rotational axis is detected as well as the corresponding point in the other scan. The pair of points with the greater change in the distance from the rotational axis is used to determine whether the current turntable step should be enlarged or made smaller.

This paper is organized as follows: Section 2 describes the basic Shape from Silhouette method used to perform the 3d model reconstruction and Section 3 presents the Next View Planning method developed. Experimental results with both synthetic and real data are given in Section 4. At the end of the paper conclusions are drawn and future work is outlined.

2 Shape from Silhouette

Shape from Silhouette is a method of automatic construction of a 3D model of an object based on a sequence of images of the object taken from multiple views, where the object's silhouette represents the only interesting feature of an image [18, 17]. The object's silhouette in each view (Figure 2a) corresponds to a conic volume in 3D space (Figure 2b). A 3D model of an object (Figure 2c) can be obtained by intersecting the conic volumes, which is also called *Space Carving* [9]. Multiple views of the object can be obtained either by moving the camera around the object or by moving the object inside the camera's field of view. In our approach the object rotates on a turntable in front of a stationary camera. Shape from Silhouette can be applied on objects of arbitrary shapes, including objects with certain concavities (like a handle of a cup), as long as the concavities are visible from at least one input view.

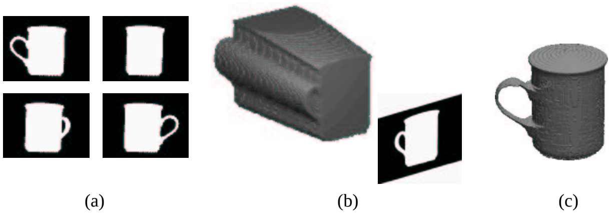


Figure 2: Image silhouettes (a), a conic volume (b) and the final model (c)

There have been many works on construction of 3D models of objects from multiple views [1, 13, 3, 17]. Szeliski [18] first creates a low resolution octree model quickly and then refines this model iteratively, by intersecting each new silhouette with the already existing model. Niem [15] uses pillar-like volume elements instead of an octree for the model representation. Wong and Cipolla [24] use uncalibrated silhouette images and recover the camera positions and orientations from circular motions. In recent years there have been also Shape from Silhouette approaches based on video sequences [6, 2]. The work of Szeliski [18] was used as a base for the Shape from Silhouette part of the method.

3 Next View Planning Approach

Our idea was to implement a simple and straight-forward NVP algorithm which will at least nearly preserve the accuracy of the models built using all possible views while reducing the number of views significantly. In most of the object reconstruction tasks which involve some kind of Next View Planning, the NVP algorithm is part of the model building process and is guided by some features of the partial model built based on preceding views. In our 3D modeling approach the acquisition of multiple views of an object and the actual object reconstruction are separated tasks — the modeling algorithm takes the images acquired as input and does not perform any view planning itself. Therefore, our goal was to design an NVP algorithm which does not need the partial model but uses only the features of the images acquired.

The acquisition system consists of a turntable, two cameras and a laser. The cameras and the laser are fixed while the turntable can rotate around its rotational axis. That means, our system has *one* degree of freedom. The minimal rotation angle of the turntable is $1°$. Therefore, the maximal number of views for our system is 360. With one degree of freedom and 360 possible views our acquisition system is fairly simple from the NVP point of view. Having the additional constraint of using the features of the images only, we propose a simple approach which takes only the current and the preceding image to decide what the next rotational step of the turntable will be. It defines a normalized metric for comparison of the current and the preceding image. If the change is less than or equal to the maximal allowed change then the step is doubled. If the change is higher than the maximal change, then the current image is discarded and the turntable moves back by half the current step. In special cases where doubling the step exceeds the maximum or halving the step falls below the minimum, the new step is set to the maximum or minimum, respectively.

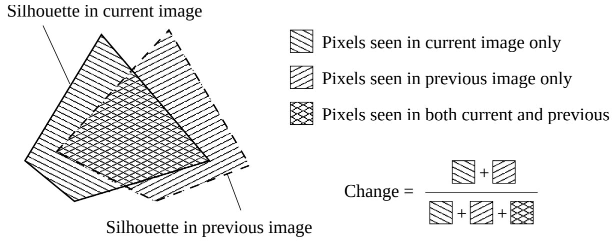

Silhouette in current image
Pixels seen in current image only
Pixels seen in previous image only
Pixels seen in both current and previous
Change = ▧ + ▨ / ▧ + ▨ + ▦
Silhouette in previous image

Figure 3: Change between two silhouette images

The only information provided by a pixel in a silhouette image is whether the pixel represents the object or the background. Following the notation common in NVP, we define a pixel representing the object as *seen* and a pixel representing the background as *empty*. Note that in a silhouette image there are no occlusions — the value of a pixel depends only on whether in the conic volume defined by the pixel there is a 3D point belonging to the object. Therefore, there can not be any *unseen* pixels, i.e., pixels for which we can not be sure whether they should be marked as seen or empty. In a binarized silhouette image all white pixels are seen and all black pixels empty. Therefore, our NVP algorithm binarizes an acquired image and compares two binary images in the following way (illustrated in Figure 3): it counts all pixels which are seen in one and empty in the other image; in order to normalize this value, it is divided by the number of pixels which are seen in at least one of the images.

With this metric definition, if two silhouette images are identical, the change is 0, and if the silhouettes do not intersect at all, it is 1. Note that calculating the change uses features of the images only and none of the information about the geometry of the acquisition system. This means that the system does not need to be calibrated prior to applying the NVP algorithm. Our NVP approach performs these steps:

1. Parameters are initialized. The user sets the initial step α_{init} and the maximal step α_{max} ($\alpha_{init} \leq \alpha_{max}$), as well as the maximal allowed change C_{max} between two subsequent images. This change is assumed to be normalized, i.e., $0 \leq C_{max} \leq 1$. The minimal step α_{min} is implied by the resolution of the turntable ($1°$ for our turntable).

2. The first image I_1 is taken. The current step α_{curr} is set to the initial value: $\alpha_{curr} = \alpha_{init}$. Number of acquired views n is set to one: $n = 1$.

3. If the turntable already has made a complete revolution of $360°$, we are done. Otherwise, the turntable is rotated by the angle α_{curr}, the image I_{n+1} is taken and we continue with Step 4.

4. The change C_{curr} between the images I_{n+1} and I_n is evaluated. If $C_{curr} \leq C_{max}$ or $\alpha_{curr} = \alpha_{min}$ the image I_{n+1} is accepted, jump to Step 6. Otherwise the image I_{n+1} is discarded, continue with Step 5.

5. The step α_{curr} is halved: $\alpha_{curr} = \frac{1}{2} \cdot \alpha_{curr}$. If α_{curr} became smaller than α_{min} it is set to α_{min}. The turntable is rotated by $-\alpha_{curr}$ (i.e., back by the half of the previous step). Go back to Step 4.

6. Increment the image counter n by one and double the step α_{curr}: $n = n + 1$, $\alpha_{curr} = 2 \cdot \alpha_{curr}$. Jump back to Step 3.

4 Results

Experiments were performed with both synthetic and real objects. For synthetic objects we built a model of a virtual camera and laser and created input images such that the images fit perfectly into the camera model. As synthetic object we created a virtual cuboid with dimensions $100 \times 70 \times 60$ mm. For tests with real objects we used 6 objects: a metal cuboid, a wooden cone, a globe, a coffee cup, and two archaeological vessels. The real volume of the first 3 objects can be computed analytically, for the other objects we can only compare the bounding cuboid of the model and the object.

The user definable parameters for NVP are the maximal and the initial step between two neighboring views, as well as the maximal allowed difference between them. The parameter with the greatest impact on the number of the views selected is the difference between two images. For all objects presented the range is from 2–15%. It was low for highly symmetrical objects (the cuboids and the cone) and high when the object was not placed in the center of the turntable. For all objects the maximal step was set to $16°$ and the initial to $4°$.

In order to evaluate the NVP-based models, we compare them with models built with a fixed number (60) of equiangular views and with models built using all 360 possible views. We expect to see that the volume of NVP-based models is closer to the volume of models built using all views than the models built with equiangular views. Figure 4 shows the models built and Table 1 summarizes the results.

The results in Table 1 indicate that for none of the objects there is a significant difference between the volume computed using NVP-based and equiangular views. This can be expected for objects with asymmetric, highly detailed surfaces, such as the vessels or completely rotationally symmetric objects, such as the cone or the globe. For simply shaped, but asymmetrical objects, such as the cuboids and the cup, a certain increase in the accuracy of the models built using NVP could be expected. In order to additionally examine our NVP algorithm, in Figure 5 we illustrate the views selected for the synthetic and real cuboid, the cone and the cup. All figures show the objects from the top view, facing the x-y plane of the world coordinate system.

In Figure 5 each dashed line indicates the direction the camera was viewing from, i.e., it represents the camera's optical axis. High density scanning areas should be those for which the silhouette border moves fast, e.g., when the width of the silhouette changes rapidly. This happens when an object's part which is far from the rotational axis starts or ends being visible from the camera. Figure 5a illustrates the difference between two views and dashed lines represent the optical axis of the camera in Figure 5. For the cuboids (Figures 5c and 5d) such parts are its corners, for the cone (Figure 5b) there are no such parts and for the cup (Figure 5e) it is its handle.

Let us analyze each of the objects from Figure 5. For the silhouette views of the cuboids (Figures 5c and 5d) the views with the highest density are $0°$–$60°$ and $180°$–$240°$. That makes sense, because the width of the cuboid silhouettes as defined in Figure 5 is smallest for views from $30°$ and $210°$ and largest from approximately $75°$, $165°$, $255°$ and $345°$. For views close to $30°$ and $210°$ the silhouette-width is determined by the two corners close to the camera. Because of being close to the camera these corners move almost orthogonally as the turntable moves, so the silhouette-width changes rapidly here and the scans are most dense in these areas. For the views of the cone (Figures 5b) all views look nearly the same, so the step between two views was constantly equal to the maximal allowed step. The step was smaller only for views close to $0°$, solely because of the starting angle being smaller than the maximal angle. For the silhouette-views of the cup (Figure 5e) high density views were taken from angles close to $165°$ and $255°$. This is expected, because for those views the cup handle starts/ends being visible (i.e., not occluded by the body of the cup).

5 Conclusion and Outlook

Obviously, our NVP algorithm did not fail in choosing the ”right” views (except for the laser views of corners of the cuboids), and did not bring any significant differences in the results (measured in terms of the volume and the size of the objects) compared to the models built using an equivalent number of equiangular views. Therefore, the number of significant views was dramatically decreased while preserving the geometry of the object. Measuring the volume only is also not the best similarity measure since this does not necessarily describe correct geometry. For example, the NVP-based model of the cup in Figure 4 contains the complete handle, whereas the model built using equiangular views misses some parts close to the top of the handle. In conclusion we proofed that the NVP algorithm for shape from silhouette is able to decrease the number of views to be computed (and thus save acquisition and computing time) for not highly structured objects.

In the future we want to test our NVP algorithm with complex, asymmetric synthetic objects and want to extend the acquisition procedure by combining the NVP guided shape from silhouette technique with an active laser triangulation method in order to be able to detect concavities.

Acknowledgements

This work was partly supported by the Austrian Science Foundation (FWF) under grant P13385-INF, the European Union under grant IST-1999-20273 and the Austrian Federal Ministry of Education, Science and Culture.

References

[1] H. Baker. Three-dimensional modelling. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 649–655, 1977.

[2] A. Bottino and A. Laurentini. Non-intrusive silhouette based motion capture. In *Proceedings of 4th World Multiconference on Systemics, Cybernetics and Informatics*, pages 23–26, July 2000.

[3] C. H. Chien and J. K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 36:100–113, 1986.

[4] C. Connolly. The determination of next best views. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 432–435, 1985.

[5] C. K. Cowan and P. D. Kovesi. Automatic sensor placement from vision task requirements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:407–416, May 1988.

[6] Q. Delamarre and O. Faugeras. 3D atriculated models and multi-view tracking with silhouettes. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, pages 716–721, 1999.

[7] K. Kemmotsu and T. Kanade. Sensor placement design for object pose determination with three light-stripe range finders. In *Proceedings of IEEE*

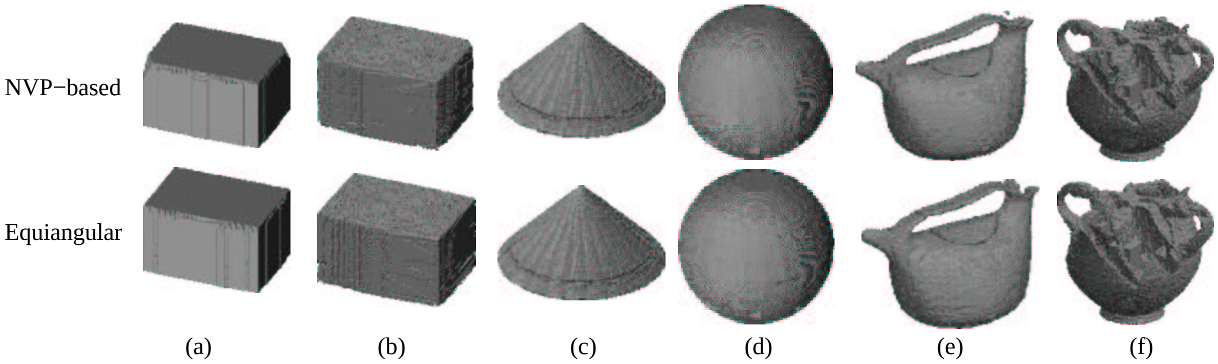


Figure 4: Comparison of models built using NVP-based and equiangular views

object	*view selection*	*#views*	*dimensions* (mm)	*volume* (mm^3)	*error*
synthetic	all	360	$100.0 \times 70.0 \times 60.0$	420 000	—
cuboid	NVP-based	54	$103.5 \times 74.0 \times 60.0$	436 666	+3.97%
(Fig. 4a)	equiangular	60	$104.0 \times 73.0 \times 60.0$	434 248	+3.39%
real	all	360	$101.0 \times 71.0 \times 60.0$	384 678	—
cuboid	NVP-based	54	$101.6 \times 72.3 \times 60.0$	397 937	+3.45%
(Fig. 4b)	equiangular	60	$101.6 \times 71.9 \times 59.5$	397 684	+3.38%
	all	360	$150.1 \times 149.4 \times 77.5$	435 180	—
cone	NVP-based	24	$151.6 \times 151.6 \times 76.5$	462 155	+6.20%
(Fig. 4c)	equiangular	60	$151.6 \times 152.2 \times 76.5$	462 207	+6.21%
	all	360	$149.1 \times 148.2 \times 144.6$	1 717 624	—
globe	NVP-based	24	$150.0 \times 149.1 \times 144.6$	1 733 613	+0.93%
(Fig. 4d)	equiangular	60	$150.0 \times 150.0 \times 144.6$	1 732 919	+0.89%
vessel	all	360	$139.2 \times 83.2 \times 92.8$	341 733	—
#1	NVP-based	52	$139.2 \times 84.0 \times 92.8$	348 699	+2.04%
(Fig. 4e)	equiangular	60	$139.2 \times 83.2 \times 92.8$	346 611	+1.43%
vessel	all	360	$112.9 \times 111.8 \times 86.4$	340 739	—
#2	NVP-based	55	$113.4 \times 112.8 \times 86.3$	349 918	+2.69%
(Fig. 4f)	equiangular	60	$113.4 \times 112.3 \times 86.3$	348 978	+2.42%
	all	360	$111.6 \times 79.0 \times 104.3$	408 344	—
cup	NVP-based	36	$112.2 \times 80.4 \times 104.3$	417 360	+2.21%
(Fig. 2c)	equiangular	60	$112.2 \times 79.7 \times 104.3$	416 726	+2.05%

Table 1: Comparison of silhouette models built using all views, NVP-based views and equiangular views

International Conference on Robotics and Automation, pages 1357–1364, 1994.

[8] H. S. Kim, R. C. Jain, and R. A. Volz. Object recognition using multiple views. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 28–33, 1985.

[9] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):197–216, July 2000.

[10] C. Liska. Das Adaptive Lichtschnittverfahren zur Oberflächenkonstruktion mittels Laserlicht. Master's thesis, Vienna University of Technology, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, Vienna, Austria, April 1999.

[11] E. Marchand and F. Chaumette. Controlled camera motions for scene reconstruction and exploration. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 169–176, June 1996.

[12] A. Marshal and D. Roberts. Automatically planning the inspection of three-dimensional objects using stereo computer vision. In *Proceedings of SPIE International Symposium on Intelligent Systems and Advanced Manufacturing*, 1995.

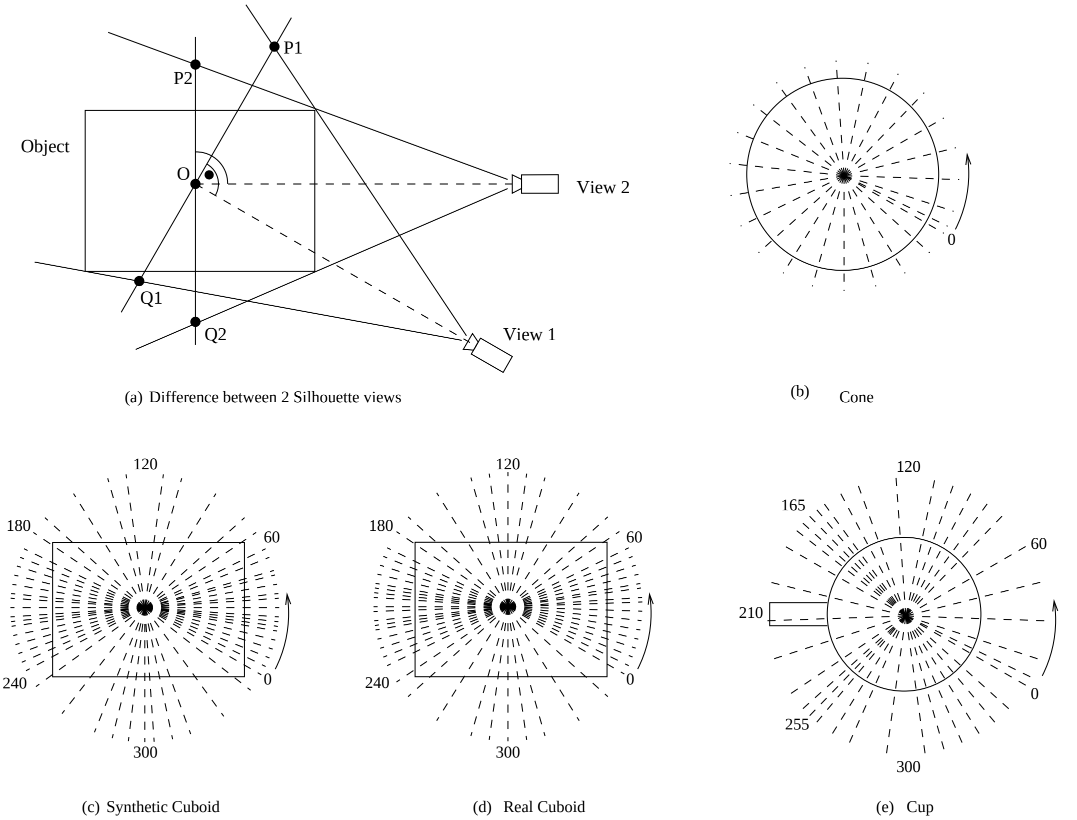
[13] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.

[14] J. Maver and R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):417–432, May 1993.

[15] W. Niem. Robust and fast modelling of 3D natural objects from multiple views. In *Image and Video Processing II, Proceedings of SPIE*, pages 388–397, 1994.

[16] 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, October 1999.

[17] M. Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40:1–29, 1987.



(a) Difference between 2 Silhouette views

(b) Cone

(c) Synthetic Cuboid

(d) Real Cuboid

(e) Cup

Figure 5: Analysis of selected views for cuboids, cone and cup

[18] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Understanding*, 58(1):23–32, July 1993.

[19] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai. A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation*, 11(1):86–104, February 1995.

[20] K. A. Tarabanis, R. Y. Tsai, and A. Kaul. Computing viewpoints that satisfy optical constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–158, 1991.

[21] G. Tarbox and S. Gottschlich. IVS: An integrated volumetric inspection system. *Computer Vision and Image Understanding*, 61:430–444, May 1995.

[22] G. Tarbox and S. Gottschlich. Planning for complete sensor coverage in inspection. *Computer Vision and Image Understanding*, 61:84–111, January 1995.

[23] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 339–346, 1994.

[24] K. Y. K. Wong and R. Cipolla. Structure and motion from silhouettes. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, pages 217–222, 2001.