Selection of Reference Images for Image-Based Representation

Tomáš Werner
Tomáš Pajdla
Václav Hlaváč
Aleš Leonardis[†]
Martin Matoušek

Center for Machine Perception, Prague
[†]Computer Vision Laboratory, Ljubljana

May 13, 2001

AMS Subject Classifications: 68T04, 90C90
Key Words: Three-dimensional computer vision, image based rendering, reference images.

Abstract

In computer vision, an attention has been devoted to image-based scene representations. They allow to construct an arbitrary view of a rigid 3-D scene by transfer from two real 2-D reference images, rather than by rendering an explicit 3-D model. We focus on the problem, how to select the optimal set of reference images from a large set of images representing the scene. The selected set minimizes a weighted sum of the number of reference views and the total fit error. We propose two different algorithms solving this optimization problem. The experimental results on synthetic and real data indicate the feasibility of the approach for 1-parametric camera motion. We discuss the possibility to extend one of the algorithms for more general cases.

Contents

1 Introduction

Let us consider the task of capturing and representing a real three-dimensional (3-D) scene or object in such a way that any view on the scene can be computed from the stored representation. There exist two approaches to this problem. The traditional model-based approach reconstructs a 3-D model of the scene and render it using computer graphics algorithms (see e.g. [6]). The image-based approach [15, 5, 19, 21] avoids an explicit 3-D model and synthesizes novel images by combining already captured reference images.

The image based approach has the advantage over the 3-D model based one in avoiding a difficult reconstruction of an explicit CAD 3-D model. To reconstruct such a model, typically, laser range finders are first used to acquire depth maps from many viewpoints, and then the depth maps are registered and fused. In this process, much user intervention is still needed and it is far from being fully automatic. E.g., making manually constructed models in the computer-animated movie “Toy Story” took over ten person-years [14].

To explain the image-based approach, we need the notion of corresponding points, illustrated in Fig. 1. These are image points $\mathbf{u}, \mathbf{u}', \mathbf{u}''$ that are projections of a single 3-D scene point $\mathbf{X}$ in several images. Combining real reference images means computing

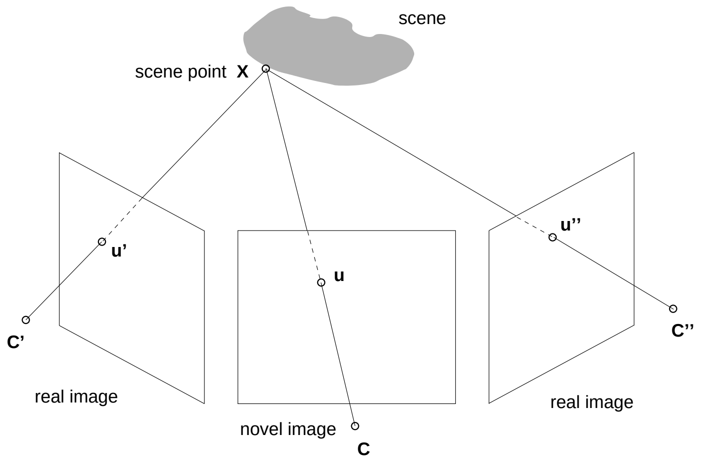

Figure 1: Corresponding image points and transfer from two real images to a novel image.

positions of image points $\mathbf{u}$ in a new retina as a function of positions of corresponding image points $\mathbf{u}', \mathbf{u}''$ in reference retinas. This is known as a *transfer*. It is known that image points can be transfered from two image planes to another one without knowledge of camera calibrations and explicit reconstruction of 3-D points. This is possible because of the existence of projective invariants of views, such as fundamental matrix or trifocal or quadrifocal tensors. For affine cameras, the transfer can be done by simple linear interpolation of coordinates, for projective camera the functions are generally multilinear [10].

In a world of isolated mathematical 3-D points, arbitrary two different reference views suffice for computing any other view of a scene. Yet, due to occlusions, image formation, unknown photometric properties of surfaces, etc., not all views can be rendered from only two reference images in a real situation, and the synthesized images are not exactly equal to the corresponding ones if they were actually captured. To achieve an acceptable similarity of the synthesized and original images, more than two reference images are needed.

The problem we address is, how to find the set of reference images such that their number is small and the similarity of the synthesized and corresponding real images is high. In more detail, we assume this task in the following scheme (see Fig. 2): (i) A large set of *primary images* of the scene is captured. (ii) A small subset of them is selected and stored as *reference images*, forming the scene representation. (iii) Any of the primary images can be re-synthesized from some subset of reference views by transfer, up to some accuracy.

The related work on reference views or images and their selection goes back to introducing aspect graphs [13] and characteristic views [3]. Gdalyahu, Weinshall *et al.* has investigated canonical views of 3-D objects [7] and representative silhouettes from a set of views in the object recognition context [8]. This paper is a continuation of work of Werner *et al.* [12, 22].

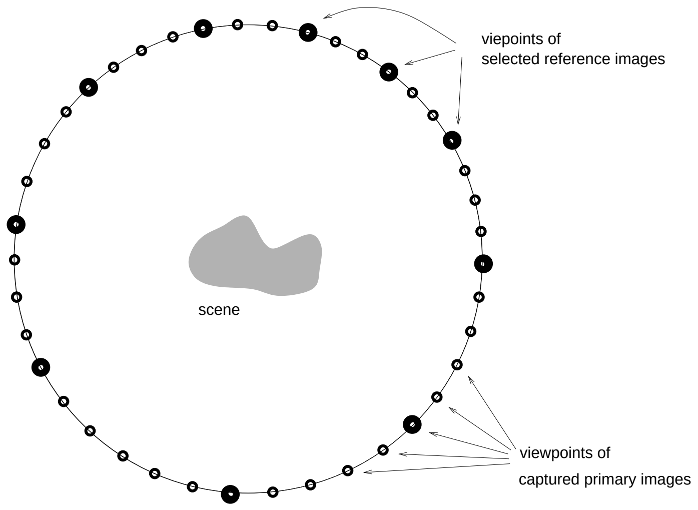

Figure 2: An example setup of the selection scheme: a small set of reference images is selected from a large set of captured primary images, such that all primary images can be re-synthesized from the reference images with some accuracy.

2 Used Concepts

$\mathbb{R}$ denotes the set of real numbers. $\{x, y, z\}$ denotes the set with elements x, y, z. A set defined by the properties of its elements is denoted by e.g. $\{x \in \mathbb{X} \mid f(x) = 0\}$. $[m, n]$ denotes an integer interval, $[m, n] = \{i \mid m \leq i \leq n\}$. $|A|$ denotes the number of elements of the set A. $\mathbb{X}$ denotes a universal set of all possible images. For images of height H and width W we can take $\mathbb{X} = \mathbb{R}^{W \times H}$.

Let $\mathcal{X} = \{X_k \in \mathbb{X} \mid k \in [1, K]\}$ be a primary image set of $K = |\mathcal{X}|$ different primary images of a rigid scene. The images are indexed, X_k denotes the k-th image. $Y(k, m, n) \in \mathbb{X}$ denotes the novel image meant to approximate X_k and constructed from X_m and X_n. We assume obvious properties $Y(k, m, n) = Y(k, n, m)$ (symmetry) and $Y(k, k, n) = Y(k, m, k) = X_k$ (the synthesis is a trivial substitution of X_k). In fact, evaluating the function $Y(k, m, n)$ involves two steps: (i) Compute corresponding pixel pairs between the images X_m and X_n, e.g. by a stereo. (ii) Synthesize the novel image $Y(k, m, n)$ by transfer using the computed correspondences between X_m and X_n and transfer parameters $t(k, m, n)$. The meaning of the transfer parameters $t(k, m, n)$ depends on the particular transfer procedure. If the transfer is done via trifocal tensor [11], the values of $t(k, m, n)$ are pre-computed trifocal tensors attached to the images X_k, X_m, X_n for appropriate triplets (k, m, n). If the camera motion is known, $t(k, m, n)$ can be computed from the motion parameters.

We want to select the reference image set $\mathcal{R} \subset \mathcal{X}$ such that each image $X_k \in \mathcal{X}$ can be reasonably approximated by some synthesized image $Y(k, m, n)$ where $m, n \in \mathcal{R}$. To quantify how well $Y(k, m, n)$ approximates X_k, we introduce image difference function $d : \mathbb{X}^2 \to \mathbb{R}^{0+}$. The greater the value of $d(X, Y)$, the less similar the images X, Y are. It is $d(X, X) = 0$ and $d(X, Y) = d(Y, X)$ for any $X, Y \in \mathbb{X}$. We further introduce the symbol $e(k, m, n) = d[X_k, Y(k, m, n)]$, denoting the error caused by replacing X_k by an image synthesized from X_m and X_n.

3 Task Formulation

At the moment, no knowledge about $\mathcal{X}$ is assumed except that its elements are images of a single rigid scene. On this general level, we are in the position to define the selection task as follows: find mappings $m, n : [1, K] \to [1, K]$ such that the reference image set $\mathcal{R} = \left\{ X_i \,\middle|\, i \in \bigcup_{k \in [1,K]} [m(k) \cup n(k)] \right\}$ is small and, at the same time, for all k the errors $e[k, m(k), n(k)]$ are small. The mappings m, n assign to each k a pair of images from which X_k is to be synthesized. Based on the Minimum Description Length principle [18], the requirement that "many quantities be simultaneously small" can be formalized as minimizing their weighted sum

$$C = a \left| \bigcup_{k \in [1,K]} [m(k) \cup n(k)] \right| + b \sum_{k \in [1,K]} e[k, m(k), n(k)] \tag{1}$$

The weights can be chosen with respect to a particular task at hand. E.g., if both reference images and residual images are stored to be able to retrieve the original images, a, b should be their respective data volumes. If only reference images are stored, they should be chosen experimentally. In fact, not only a, b but also the function e are free parameters, however, some solid ground for choosing them can be found in Minimum Description Length or Minimum Likelihood considerations.

To make this difficult optimization task feasible, we adopt the three following simplifications:

Low-parametric camera change. We assume a structure on $\mathcal{X}$ in the following sense. The only reason why primary images X_k are different is the change of camera projection matrix, i.e., camera extrinsic and/or intrinsic parameters. The measured primary images can be imagined as points P_k in a universal space of all possible camera projection matrices. The constraint is that these points lie on a low-dimensional surface (more exactly, curve for 1-dimensional, surface for 2-dimensional, etc.) in this space. If the surface's dimension is n, we speak about n-parametric camera motion. This rather abstract formulation of the constraint is obvious once the following examples are given. A typical setup for a 1-parametric camera change is a video camera moving on a curve, possibly changing its zoom, and observing a static scene, or a static camera observing an object on a rotating table. A setup for a 2-parametric camera change is a static camera observing an object on a table that can rotate and tilt.

1-parametric camera change. We restrict the camera motion that is considered in full detail in this paper to a 1-parametric case, although we will also discuss extension to a 2-parametric motion. In a 1-parametric motion, the primary images are ordered on a curve, assuming further that the topology of the curve is that of a line segment rather than that of a circle. Images can be renumbered so that X_{k+1} is a successor of X_k, we will assume that this has been done.

Image interpolation. Any two reference images can be used to synthesize only images *between* them, i.e., for each $k \in [1, k]$ it is $m(k) \leq k \leq n(k)$. In other words, images are interpolated rather than extrapolated [23].

In the next two sections we describe two approaches to minimizing (1) under the above simplifications. The first one is based on the dynamic programming, the second one on selection of viewing intervals.

4 Optimization by Dynamic Programming

Under all three simplifications, the primary images can be renumbered so that X_{k+1} is a successor of X_k. We assume this has been done. Then the task is to select a reference image set $\mathcal{R} = \{X_{k(1)}, X_{k(2)}, \ldots, X_{k(I)}\}$ minimizing (1). $\mathcal{R}$ is specified by the number of reference images $I = |\mathcal{R}|$ and the mapping $k : [1, I] \to [1, K]$. The mapping k satisfies the following requirements: $k(1) = 1, k(I) = K$, and $k(i + 1) > k(i)$. Fig. 3 shows an example of the mapping k for $K = 5, I = 4$ and $\mathcal{R} = \{X_1, X_2, X_4, X_5\}$. Any valid mapping k in the figure corresponds to a monotonic function starting in the node $(1, 1)$ and ending at the top horizontal line.

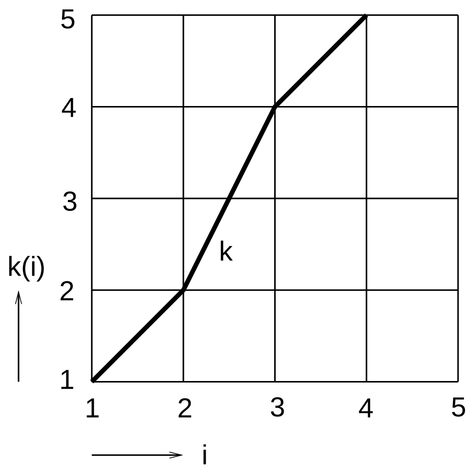

Figure 3: Example of mapping specifying the selected reference images.

Using k, (1) can be rewritten to

$$C = aI + b \sum_{i=2}^{I} \sum_{l=k(i-1)}^{k(i)} e[l, k(i-1), k(i)] \tag{2}$$

which can be further written as

$$C = a + \sum_{i=2}^{I} c_{k(i-1)k(i)}, \quad c_{mn} = a + b \sum_{k=m}^{n} e(k, m, n)\,. \tag{3}$$

Since the error function is additive and the images are ordered, the global minimum C^* of (3) can be computed by dynamic programming [2]. The algorithm proceeds in K steps, each step computing value C_n from values $C_1, C_2, \ldots, C_{n-1}$ as

$$C_n = \min_{i=1}^{n-1}(C_i + c_{in}) \,. \tag{4}$$

starting with $C_1 = a$ and finishing with $C_K = C^*$. The value C_n can be interpreted as the optimum for the optimization problem equal to (3) up to the modification that the number of primary images is $n < K$ rather than K.

The optimal mapping k^* can be found by backtracking the indices of the minima. Let us store these indices in J_n, i.e., J_n is the index i for the minimum in the n-th step in (4). By backtracking, k^* is obtained in a reversed form, denoted by $\hat{k}^*$, from the recurrent formula $\hat{k}^*(j+1) = J_{\hat{k}^*(j)}$, $\hat{k}^*(1) = K$.

The algorithm steps can be conveniently imagined in Fig. 3. The optimal mapping for the n-th step is a function with minimal cost C_n connecting the node $(1,1)$ and the n-th horizontal line. The cost C_n of the path is the sum of costs c_{mn} of primitive paths connecting horizontal lines m, n.

5 Optimization by Growing and Selecting Viewing Intervals

Dynamic programming cannot be extended to a 2-parametric camera motion, because it requires ordering of primary images, which is possible only for a 1-parametric motion. We will describe a technique to minimize (1) without ordering primary images. Although we present results only for a 1-parametric case, the technique has a potential to be extended also for 2-parametric case. Unlike for dynamic programming, circular rather than linear topology can be taken into account in 1-parametric case.

Minimizing (1) for a low-parametric camera motion can be viewed as recovering parametric models. The *model* is Y our transfer function, the *model domain* the set of images synthesized from a single pair of reference images, the *model parameters* is a particular pair of reference images, and the *fit error* the transfer error, $e[k, m(k), n(k)]$.

We use the framework of Leonardis *et al.* [20, 16], applied successfully for several applications. We consider image domains (intervals in 1-parametric case) rather than images as primitive features for the optimization. We introduce *plausible viewing domains*, which is a domain (an interval in a 1-parametric case) of primary images such that it can be synthesized from a single pair of images with the interpolation error $e(k, m, n)$ smaller than a threshold ε. The algorithm has two parts:

Growing. For each primary image independently, grow the largest plausible viewing interval starting in this image.

Selection. From these intervals, select the optimal subset which minimizes the criterion (1).

The growing is done by a technique similar to region growing in segmentation: the images are added until the error exceeds a threshold ε. For simplicity, we consider also the 3-rd simplification from Section 3, i.e., the reference images for each viewing interval are its end images.

In the selection step, the following objective function is minimized:

$$F(\mathbf{i}) = \mathbf{i}^T \mathbf{C} \mathbf{i} \tag{5}$$

The vector $\mathbf{i}^T = [i_1, i_2, \ldots]$ denotes a set of selected viewing intervals, where i_k is a *presence variable* having the value 1 for the presence and 0 for the absence of the k-th interval. The diagonal terms of the square symmetric matrix $\mathbf{C} = [c_{ij}]$ express the cost of presence of an interval:

$$c_{ii} = 2a - a|D_i| + be_i \ , \tag{6}$$

where D_i is the set of images forming the i-th interval and $e_i = \sum_{k \in D_i} e(k, m_i, n_i)$ where m_i, n_i are end images of the i-th interval. The term $2a$ is the cost of including two end images of the interval. The term $-a|D_i|$ is the benefit of covering $|D_i|$ primary images by the model. This term is present because the algorithm, unlike the dynamic programming, does not ensure that all primary images are covered by the union of all the intervals. Those that are not covered have to be represented by storing the whole images, i.e., the cost a must be added for each of them.

If i-th and j-th intervals overlap, their combined cost is not just the addition of their costs but rather $c_{ii} + 2c_{ij} + c_{jj}$, where the off-diagonal term is expressed by

$$2c_{ij} = 2c_{ji} = -a\theta_{ij} + a|D_i \cap D_j|) - be_{ij} \ , \tag{7}$$

where θ_{ij} is 1 if the i-th and j-th intervals are adjacent (i.e., one reference image is spared) and 0 if they are not. The term e_{ij} is

$$e_{ij} = \max \left\{ \sum_{k \in D_i \cap D_j} e(k, m_i, n_i) \, , \sum_{k \in D_i \cap D_j} e(k, m_j, n_j) \right\}$$

It can be verified that $F = C - aK$, provided that the viewing intervals overlap only pairwise. As aK is constant, minimizing (5) is equal to minimizing (1) then. In this sense F can be viewed as a "second order approximation" of C.

The minimization of the objective function $F(\mathbf{i})$ belongs to the class of quadratic Boolean problem that are NP-hard [9]. The exhaustive search is not tractable in our case. To find a "good" sub-optimum, we are currently using two different methods. One is a simple and computationally efficient greedy search and the other one is a tabu search. Tabu search is computationally a little more demanding but it provides consistently better results than the greedy algorithm.

More details on this strategy can be found in [20, 16].

6 Experiments and Results

The goal of the experiments is to verify the feasibility of the dynamic programming and grow-and-select algorithms for selecting reference images. All three constraints described in Section 3 are considered, i.e., the primary image set originates from 1-parametric camera motion, and only novel images between reference images are synthesized. More experimental details can be found in [17].

6.1 Experimental Setup

A 3-D object was placed on a calibrated rotating table, so the rotation angle was known accurately. For each object, 300 primary images $X_1, \ldots, X_{300}$ were captured, each step corresponding to the rotation 1.2° (degrees). A CCD camera capturing images was placed at a long distance from the object such that the projection was near to orthographic. Moreover, the camera position was adjusted so that projection of any 3-D point on the object moved in a single row in the image during rotation, i.e., vertical coordinates of all corresponding points were equal.

We used two types of data, synthetic and real. For synthetic data, the images were rendered by a ray tracer [1], modeling the same experimental setup as for real data, except that only a single-line camera was used, capturing scene points in a single horizontal plane, perpendicular to the rotation axis. For the real data, the conventional TV CCD camera was used, see Fig. 4 for the camera and the table.

Figure 4: The experimental setup used for capturing primary images; the CCD camera and the rotating table.

6.2 Synthesis of Novel Images

Synthesis of the novel image $Y(k, m, n)$ proceeds in the following steps:

Finding correspondences. Pairs of corresponding pixels were computed in images X_m, X_n. For synthetic data, the correspondences were computed by the ray tracer. For real data, stereo algorithm based on dynamic programming [4] was used as follows. For small angle difference between the input images, the correspondence was computed by the stereo algorithm in a single step. For large angle difference, several correspondences computed for small enough angles were linked. The reason for not computing correspondences in a single step is the fact that the stereo algorithm does not yield reliable results for large image displacements.

Transferring position. For each pair of corresponding pixels from X_m, X_n, the position of the corresponding image points in the novel image $Y(k, m, n)$ is computed by transfer. In our case of orthographic projection, a linear function is sufficient for transfer. It is $(u_k, v_k) = (\alpha u_m + \beta u_n, v_m)$, where $(u_k, v_k), (u_m, v_m), (u_n, v_n)$ are image coordinates of a triplet of corresponding image points in $Y(k, m, n), X_m, X_n$, respectively. The coordinate u is horizontal, v is vertical. The parameters α, β are simple functions of k, m, n. In fact, α, β are components of the transfer parameter vector t from Section 2, $t(k, m, n) = (\alpha(k, m, n), \beta(k, m, n))$.

Transferring intensity. The intensity of the image point (u_k, v_k) is computed as a linear interpolation of intensities of the pixels $(u_m, v_m), (u_n, v_n)$ along the rotation angle.

Solving visibility. Since occlusion can occur in the novel image, visibility must be solved explicitly by rendering first points farthest from the novel camera.

6.3 Image Difference Function

The function d (introduced in Section 2) quantifies the dissimilarity of the original and the corresponding synthesized image. The concept of similarity is of course vague and depends on the exact storage-and-retrieval scheme and the application at hand. The function can reflect image difference perceived by an observer, the data volume of compressed residuals, etc.

We computed $d(X, Y)$ as sum of squared differences of intensities of pixels with the same coordinates in the experiments. Besides, this d were modified as follows: if d is greater than a threshold ε, it is set to infinity. This forces the dynamic programming optimization to keep all image dissimilarities under ε, which the grow-and-select optimization does naturally.

6.4 Presentation of Results

The selection results are presented for synthetic data, namely a textured cube viewed by a single-line camera, and for 4 real objects. The selection results are in Figures 6 to 9. For each object, experiments were done for both dynamic programming and grow-and-select algorithms.

The result for each object is presented in six figures, labeled by a to f. The figures a show one of the primary images of the object. The exception is the synthetic object, where Fig. 6a shows all single-line primary images, stacked one above another; the horizontal coordinates determines image point position and the vertical coordinate the index k of the primary image.

The figures labeled by b are visualizations of $e(k, m, n)$. As $e(k, m, n)$ is a trivariate function, we visualize only the bivariate function $\sum_{k=m}^{n} e(k, m, n)$ for simplicity, being in fact the summed error in the viewing interval $[m, n]$. The bivariate function is visualized as a family of curves in polar coordinates in which the two independent variables are the angle and the curve number. The angle is $1.2°[(m+n)/2-1]$, i.e. the center of the viewing interval in degrees. It ranges in $1.2°[0,259]$. The curve number is $m-n$, i.e. the interval length. It ranges in $[1,20]$ in Fig. 6b and in $[1,30]$ in the figures for real objects.

The figures labeled by c and d are selections results for optimization by dynamic programming. c shows selected reference images. Since c shows often almost uniform distribution of reference images around the object, d visualizes the lengths of intervals between neighboring reference image pairs (in degrees).

The figures labeled by e and f are selection results for the grow-and-select algorithm. e are selected viewing intervals, the ends of which are in fact the reference images. Similarly as d, f shows the interval lengths.

To enable a visual interpretation, the silhouette of the top view of the object is superimposed in c, d, e, f. The interpreting needs some care because of the fact that the camera is orthographic is counter-intuitive in the figures. E.g., in Fig. 5 both sides A and B are visible from the view R.

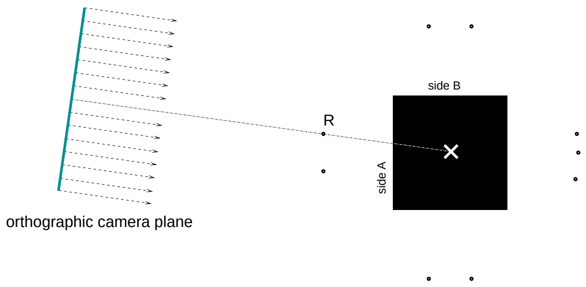

Figure 5: Interpretation of viewpoints in a plot. The reference view R sees both the side A and B of the cube.

6.5 Discussion

6.5.1 Synthetic cube

Fig. 6 shows the results for the textured synthetic cube, captured by a single-line camera. The synthesis error (b) shows peaks for viewing intervals with centers in $0°, 90°, 180°, 270°$. These peaks are caused by the fact that e.g. a virtual image in $85°$ cannot be accurately rendered from reference images in $80°$ and $100°$ because the virtual camera sees a cube face that is invisible in the second reference image. This fact implies that the selected reference images in c and e form pairs of images close to each other.

There is an unexpected triple of images on the right-hand side of c: it is because the dynamic programming algorithm considers the image sequence with linear rather than circular topology, i.e., the 1-st and 299-th images are not neighbors.

The result e of the grow-and-select algorithm is very similar to c. Small holes between the selected intervals and their overlap can be seen. They occur because the algorithm does not ensure that the selected intervals are disjoint and cover all primary images.

6.5.2 Complex real objects

Figs. 7, 8 and 9 show results for real objects of complex shapes. For complex shapes, the selected images resp. intervals tend to be distributed uniformly around the object. The reason is the fact that (i) influences of rows tend to cancel each other and (ii) the objects are smooth, i.e., they are not polyhedra.

6.5.3 Real cylinder

Fig. 10 verifies that the selected images are distributed uniformly as expected. The slight non-uniformity is caused by the irregularity of the texture.

6.5.4 Note on computational complexity

The dynamic programming and grow-and-select algorithms yield comparable results, however, the latter is superior in computational complexity. By far, the most demaninng operation is the evaluation of $Y(k, m, n)$, which includes stereo and image synthesis. The second one is the evaluation of e. In the experiments, the number of evaluating Y was more than one order of magnitude higher for dynamic programming.

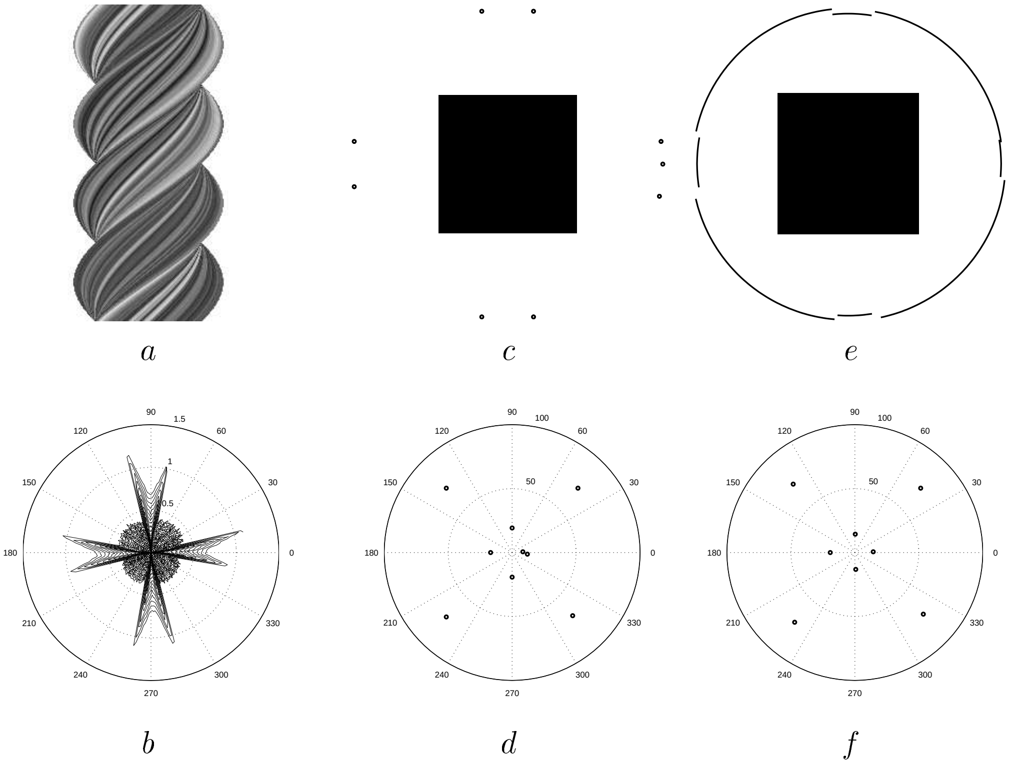

Figure 6: The results for synthetic textured cube, captured by a single-line camera.

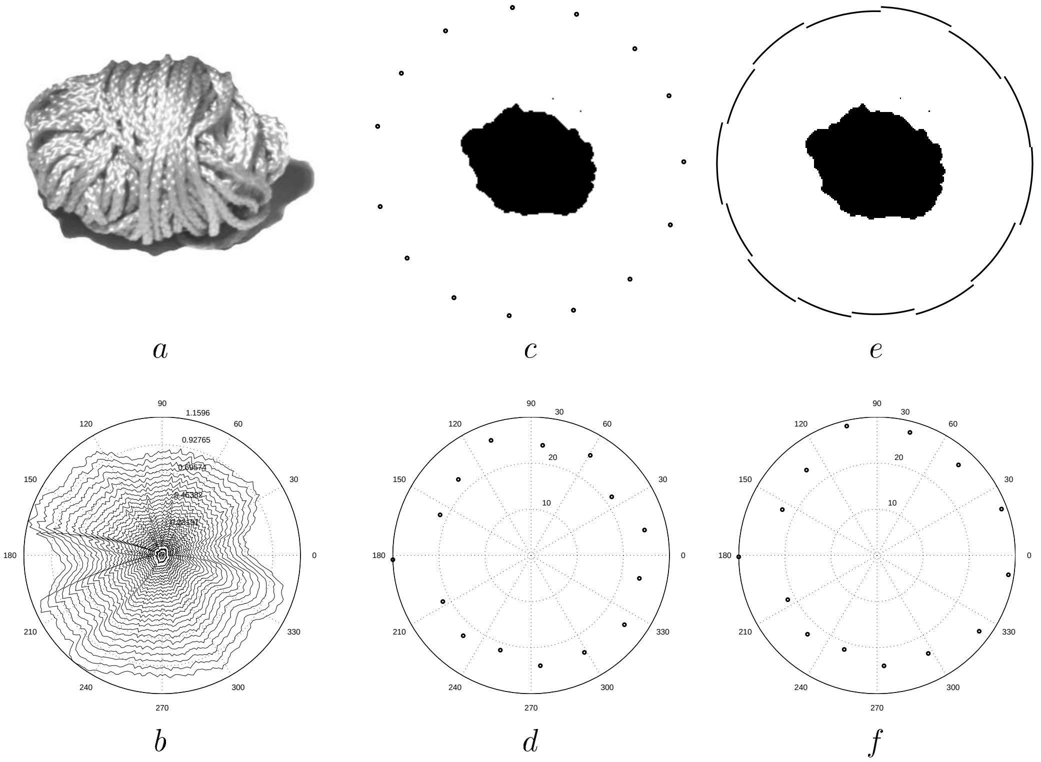

Figure 7: The results for the Rope object.

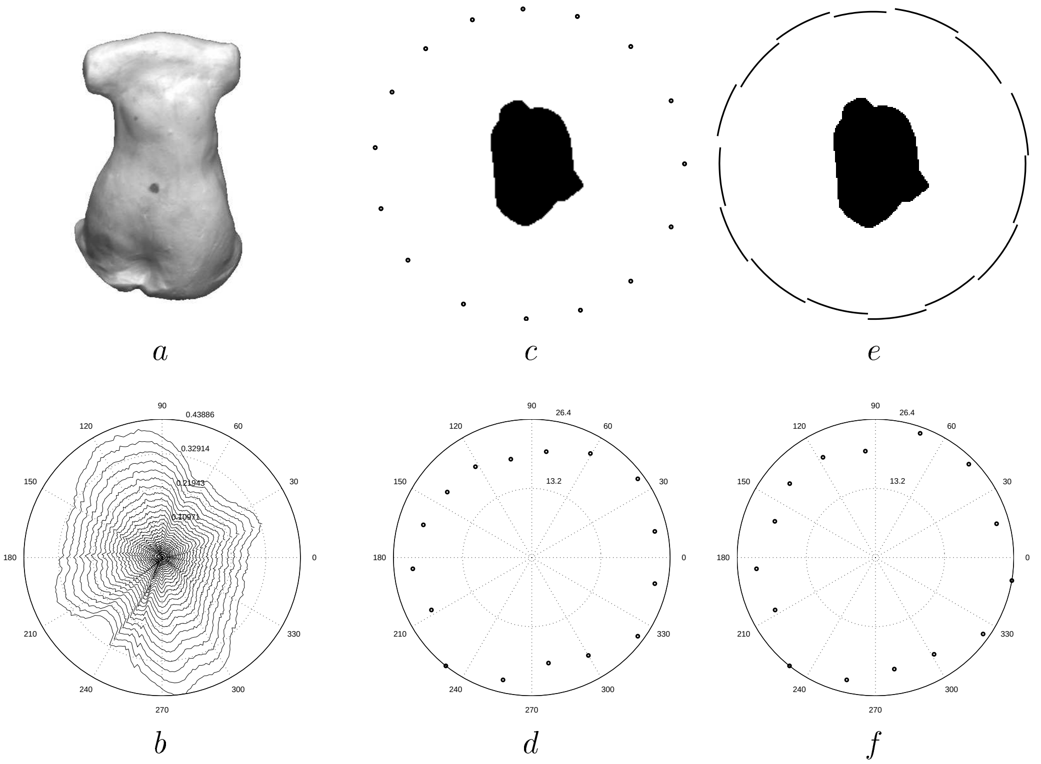

Figure 8: The results for the Torso object.

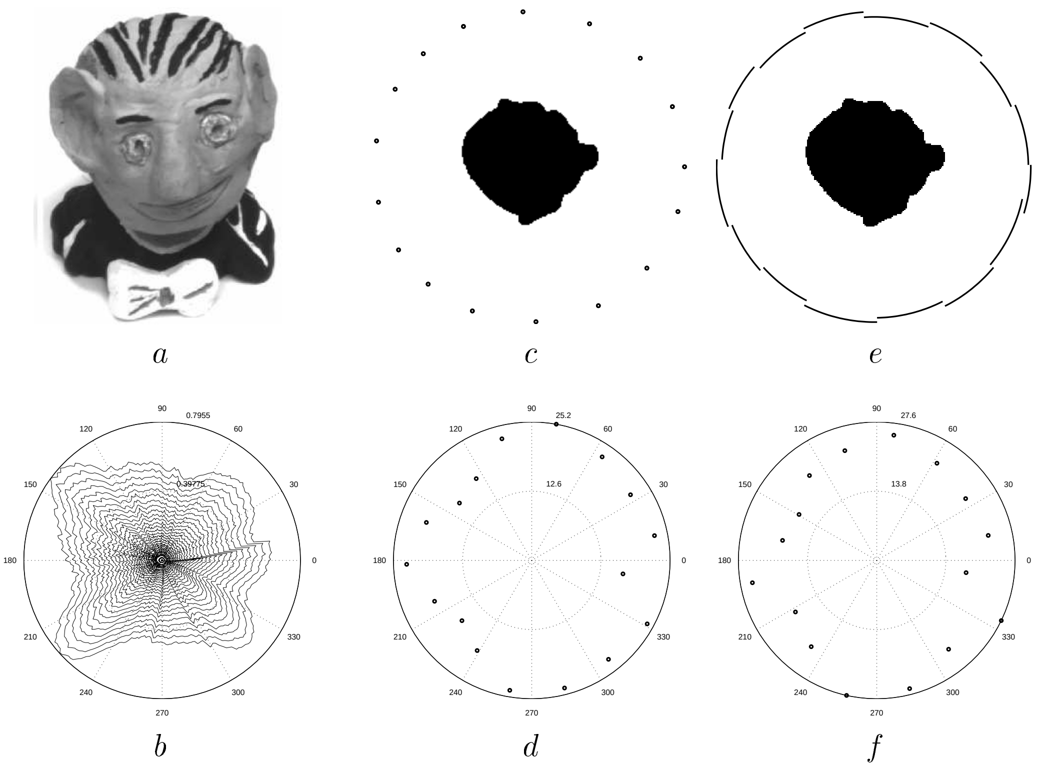

Figure 9: The results for the Head object.

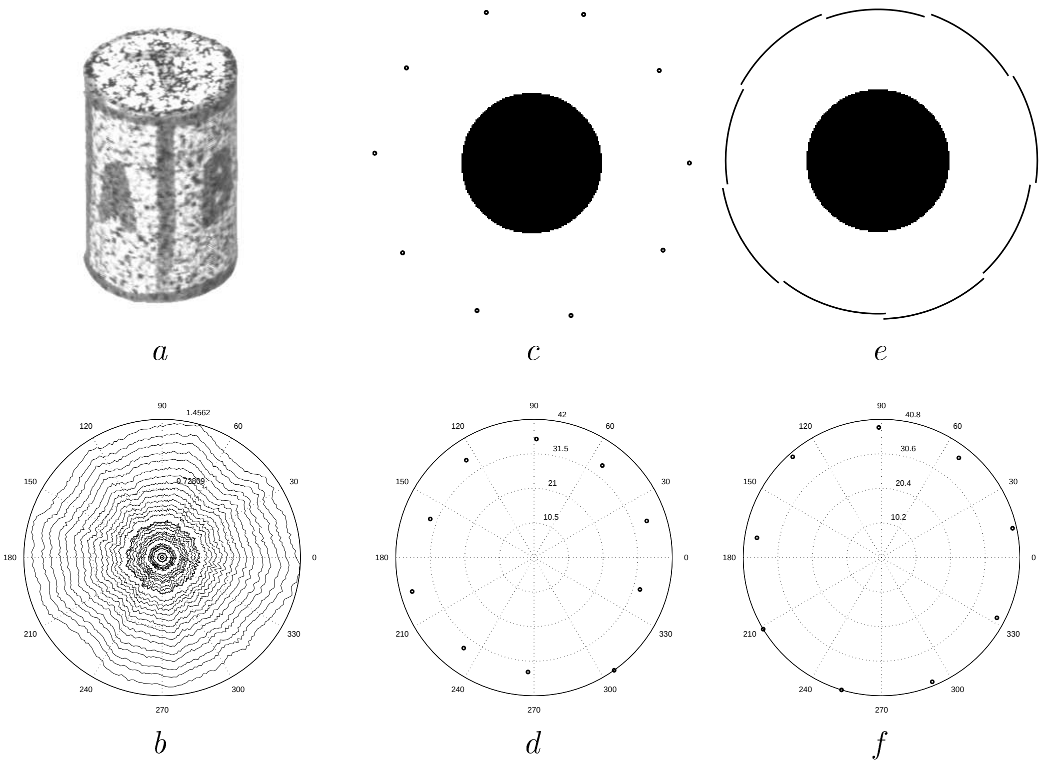

Figure 10: The results for the Cylinder object.

7 Conclusion

Optimal set of reference images was defined as a small set of images selected from a large set of captured images such that the criterion (1) is minimized. This criterion is a weighted sum of the image number and the residual error of the original and corresponding synthesized images. The definition avoids explicit using the geometry of the underlying 3-D scene, because it is referred implicitly as the residual error between the original and corresponding synthesized image. This way, not only occlusions but also loss of information due to image formation process (such as different sampling frequencies of different images) included.

Two different algorithms for the selection of the optimal set of reference have been presented. The first algorithm, using dynamic programming, finds the global minimum, but requires larger computational effort, and can be used only for 1-parametric camera motion. The second algorithm finds sub-optimum of the objective function by first growing a large set of image domains each of them coverable by a single pair of reference images, followed by selecting a sub-optimal set from these domains. It needs no ordering of primary views and thus is open for extension to 2-parametric motion.

The feasibility of the approach is verified in experiments with synthetic and real data. The experiments were done for objects of simple or mildly complex shapes. Possible applications are the compression of an image sequence and finding an optimal image-based representations of 3-D objects for e.g. 3-D object databases.

7.1 Possible Generalization

Acknowledgment

Tomáš Werner was supported by the Czech Ministry of Education under Project LN00B096. Tomáš Pajdla, Vaclav Hlavac and Martin Matousek were supported by the Czech Ministry of Education under Project 4/11/AIP CR OCAMS, by the Grant Agency of the Czech Republic under Projects GACR 102/00/1679 and GACR 102/01/0971, and by the EU Fifth Framework Programme project 1999-29017 OMNIVIEWS.

References

[1] Persistence Of Vision ray tracer. `http://www.povray.org/`.

[2] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[3] I. Chakravarty and H. Freeman. Characteristic views as a basis for three-dimensional object recognition. *Proceedings of The Society for Photo-Optical Instrumentation Engineers Conference on Robot Vision*, 336:37–45, 1982.

[4] I. J. Cox, S. Hingorani, B. M. Maggs, and S. B. Rao. Stereo without disparity gradient smoothing: A Bayesian sensor fusion solution. In *British Machine Vision Conference*, pages 337–346, Berlin, 1992. Springer-Verlag.

[5] Charles R. Dyer. Image-based scene rendering and manipulation research at the university of wisconsin. In *Proc. Image Understanding Workshop*, pages 63–67, 1997.

[6] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics- Principles and Practice.* Addison-Wesley, 2-nd edition, 1990.

[7] Y. Gdalyahu, D. Weinshall, and M. Werman. Canonical views, or the stability and likelihood of images of 3d objects. In *ARPA94*, pages II:967–971, 1994.

[8] Yoram Gdalyahu and Daphna Weinshal. Measures for silhouettes resemblance and representative silhouettes of curved objects. In Bernard Buxton and Roberto Cipolla, editors, *European Conference on Computer Vision*, Lecture Notes in Computer Science, No. 1064, pages 364–375, Heidelberg, Germany, April 1996. Springer Verlag.

[9] F. Glover and M. Laguna. Tabu search. In C. R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 70–150. Blackwell Scientific Publications, 1993.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, UK, 2000.

[11] R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, March 1997.

[12] Václav Hlaváč, Aleš Leonardis, and Tomáš Werner. Automatic selection of reference views for image-based scene representations. In *Proceedings of the European Conference in Computer Vision*, volume 1 of *Lecture Notes in Computer Science, No. 1064*, pages 526–535, Heidelberg, Germany, April 1996. Springer.

[13] J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[14] J. Lasseter and S. Daly. *Toy Story: The Art and Making of the Animated Film.* Hyperion, November 1995. See also `http://www.disney.com/ToyStory/`.

[15] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images. In *Proc. of 12th International Conf. on Pattern Recognition, Jerusalem, Israel*, pages 689–691, October 9–13 1994.

[16] A. Leonardis. *Image Analysis Using Parametric Models.* PhD thesis, University of Ljubljana, Slovenia, March 1993.

[17] Martin Matoušek. Selection of optimal reference views. Master's thesis, Czech Technical University, Karlovo náměstí 13, Praha, Czech Republic, february 2000.

[18] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Co., Series in Computer Science, IBM Almaden Research Center, San Jose, California U.S.A, 1989.

[19] D. Scharstein. View synthesis using stereo vision. Technical Report TR96-1604, Cornell University, Department of Computer Science, Ithaca, New York, USA, January 1997.

[20] M. Stricker and A. Leonardis. ExSel++: A general framework to extract parametric models. In V. Hlaváč and R. ára, editors, *6th International Conference on Computer Analysis of Images and Patterns, CAIP'95*, number 970 in Lecture Notes in Computer Science, pages 90–97, Prague, Czech Republic, September 1995. Springer-Verlag, Heidelberg.

[21] Tomáš Werner. *Image-Based Visualization of Real 3D Scenes.* PhD thesis, Faculty of Eletrical Engineering, Czech Technical University, FEL ČVUT, Karlovo náměstí 13, Praha, Czech Republic, 1998.

[22] Tomáš Werner, Václav Hlaváč, Aleš Leonardis, and Tomáš Pajdla. Selection of reference views for image-based representation. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 1, pages 73–77, Vienna, Austria, August 1996. IEEE Computer Society Press, Los Alamitos, California.

[23] Tomáš Werner, Tomáš Pajdla, and Václav Hlaváč. Efficient 3-D scene visualization by image extrapolation. In *European Conference on Computer Vision, Freiburg, Germany*, volume Lecture Notes in Computer Science No. 1407, pages 382–396, 1998.