IEEE Proc. Southwest Symposium on Image Analysis and Interpretation (SSIAI'02), Santa Fe, USA, pp. 144-147, April 2002

Geometry-based Automatic Object Localization and 3-D Pose Detection

Marcus A. Magnor
Stanford University
Computer Graphics Lab
Stanford, CA 94305 USA
magnor@graphics.stanford.edu

Abstract

Given the image of a real-world scene and a polygonal 3-D model of a depicted object, its apparent size, image coordinates, and 3-D orientation are autonomously detected. Based on matching silhouette outline to edges in the image, an extensive search in parameter space converges to the best-matching set of parameter values. Apparent object size may a-priori be unknown, and no initial search parameter values need to be provided. Due to its high degree of parallelism, the algorithm is well suited for implementation on graphics hardware to achieve fast object recognition and 3-D pose estimation.

1. Introduction

The vast number of algorithms developed to extract information on the existence, position, and pose of a specific object in an image reveals the importance of the problem, but also the magnitude of the challenge [12, 1]. In this work, a rather 'brute force' object-recognition technique is presented that yields reliable detection results while offering the potential to be implemented on fast, parallel-processing graphics hardware [8, 11, 13]. Given the 3-D geometry of an object, the proposed recognition scheme autonomously determines the object's apparent size, image coordinates, and its pose in an uncalibrated image. When compared to previous research on geometry-based recognition [3, 10, 9, 4], the presented scheme offers the advantages that no initial search parameter values must be provided, apparent object size may a-priori be unknown, and that objects of arbitrary shape can be detected.

In the succeeding section, the generation of multiple different object silhouette outlines is described. To identify object edges in the recorded image, the Canny Edge detector is applied [5]. The edge pixels are convolved with all stored object outlines to find the best-matching object orientation and position. Scaled versions of the image are compared to determine the size of the object in the image plane. A refining search around the detected 3-D pose, size, and position concludes the automatic object detection. Its highly parallel nature allows implementing the presented algorithm in programmable graphics hardware, yielding a fast and robust object recognition system.

2. Object Outline Generation

To compare the three-dimensional model of the object to its two-dimensional projection in the image plane, a list of object outlines is generated. Object pose is parameterized by three angles α, β, γ, corresponding to rotation around the object's Y-axis, X-axis, and the rotation of the image plane, respectively. To avoid degeneracies, the range of values for α, β, γ depends on the symmetry characteristics of the object; in the most general case, $-180° \leq \alpha < 180°$, $-90° \leq \beta \leq 90°$, and $-180° \leq \gamma < 180°$. The object model is rendered for N^{pose} different parameter values $(\alpha_j, \beta_j, \gamma_j)$. The number of samples N^{pose} depends on the angular step sizes $\Delta\alpha, \Delta\beta, \Delta\gamma$ which are chosen such that no silhouette point rotates by more than a preset number of pixels per step. To render the geometry model, parallel projection is used which is valid for small fields of view.

Since the algorithm relies on edges in the image to identify the object, only the silhouette outline of each projec-

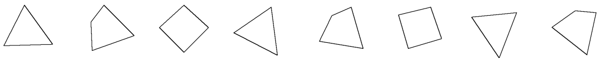
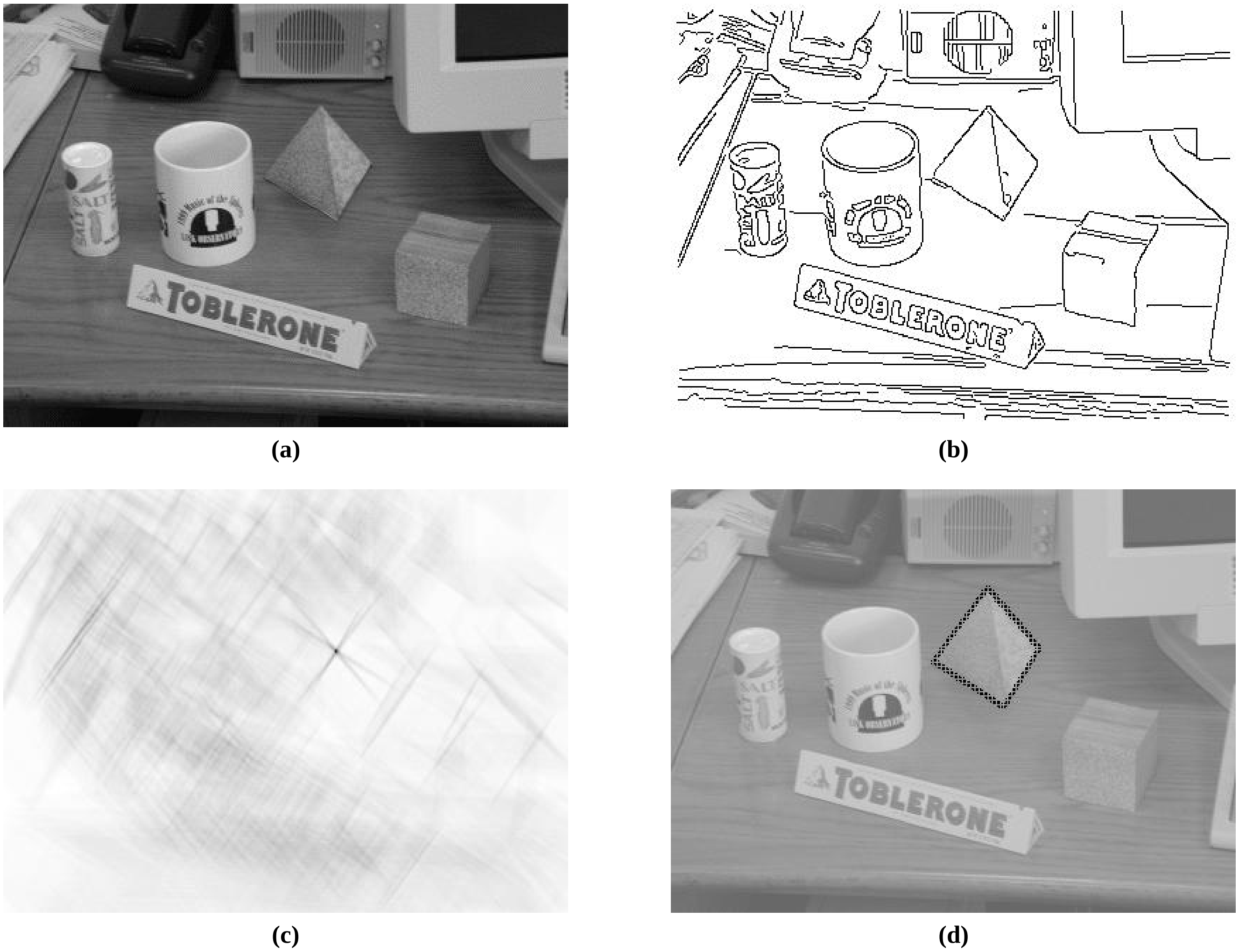


Figure 1. The 3-D geometry model of the object is rendered from different perspectives. The projected silhouette outlines are stored.


(a)

(b)

(c)

(d)

Figure 2. The image of a real-world scene (a) is processed to detect object edges (b). The edge pixels are convolved with one silhouette outline at a time (c): the location of the strongest signal indicates the best-matching object position in the image (d).

tion must be stored (Fig. 1). For each rendered object pose $(\alpha_j, \beta_j, \gamma_j)$, the N_j^{sil} outlining pixel coordinates $(x_{j,i}, y_{j,i})$ are stored in a list

$$L_j^{\text{pose}} \quad = \quad \{x_{j,i}, y_{j,i}\} \ , \ 0 \leq i < N_j^{\text{sil}}.$$

The number of outline pixels N_j^{sil} is kept approximately equal to a preset silhouette circumference length N^{sil} by scaling the rendered object projection for each set of parameter values $(\alpha_j, \beta_j, \gamma_j)$ appropriately.

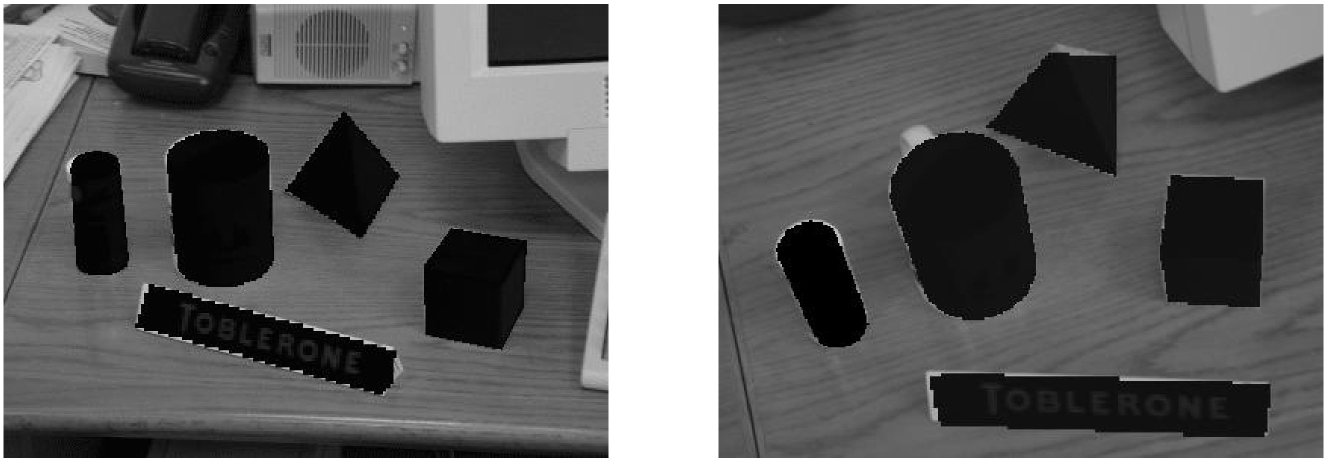
3. Object Localization and Pose Detection

To identify object edges in the image of a natural scene (Fig.2a), the Canny Edge detector is applied [5]. The resulting edge image (Fig.2b) is traversed once, and all N^{edge} edge pixel coordinates (x_k, y_k) are stored in a list of edge pixels:

$$L^{\text{edge}} \quad = \quad \{x_k, y_k\} \ , \ 0 \leq k < N^{\text{edge}}.$$

To find an object's position and pose in the image, the edge pixels are convolved with the previously generated object outlines, Sect. 2. Using the list of edge pixel coordinates L^{edge} in conjunction with the list of silhouette outline pixel coordinates L_j^{pose}, the convolution becomes

$$\begin{aligned} H_j &= L^{\text{edge}} \otimes L_j^{\text{pose}} \\ &= \text{inc} \ \left(p_j \left(x_k + x_{j,i} \, , \, y_k + y_{j,i} \right) \right) \qquad (1) \\ & \quad \forall \quad 0 \leq k < N^{\text{edge}}, 0 \leq i < N_j^{\text{sil}}. \end{aligned}$$



(a) (b)

Figure 3. Autonomously detected pose, size, and position of different objects in two different views of a real-world scene.

The inc-operator increments the value of pixel $p_j(x, y)$. Since all edge and outline pixels have equal weight, (1) represents a binary convolution consisting only of incrementing memory addresses which can be implemented very efficiently. The convolution result H_j resembles the edge image's generalized Hough transform [2, 7, 6] for outline L_j^{pose} (Fig.2c). To find the best-matching position for outline L_j^{pose} in the image, the coordinates in the Hough transform with the highest pixel value $p_j^{\text{max}}(\hat{x}_j, \hat{y}_j)$ are sought. Since image edges as well as silhouette outlines are represented by one-pixel wide lines, however, rasterization aliasing (staircase effect, "jaggies") may prevent correct detection. The generalized Hough transform is therefore low-pass filtered using a small 3×3 kernel before the most probable object location for pose j is identified. The ratio of the highest Hough pixel value p_j^{max} to the number of total silhouette outline pixels N_j^{sil} gives a confidence measure c_j for the detected object position and pose,

$$c_j = \frac{p_j^{\text{max}}(\hat{x}_j, \hat{y}_j)}{N_j^{\text{sil}}}. \quad (2)$$

Eq.(2) is evaluated for all N^{pose} silhouette outlines to determine which outlines have a high probability to be present in the image. This way, the best-matching object orientation parameters $(\alpha_j, \beta_j, \gamma_j)$ and corresponding image coordinates $(\hat{x}_j, \hat{y}_j)$ are detected.

Since the object silhouettes are scaled to assert an approximately equal number of outline pixels N^{sil} for all poses, different silhouettes correspond, in general, to different apparent object sizes in the image plane. In order to determine the apparent size of the object in the image plane, the image itself is also scaled to correspond to different magnifications.

4. Object Size Determination

To find the unknown apparent size of the object in the image plane, the above-described convolution detection scheme is repeated using scaled versions of the image, corresponding to different apparent object sizes. This way, the object silhouettes need not be re-rendered. The original image is appropriately filtered and fractionally re-sampled before the Canny Edge detector is applied to the scaled image. The best-matching object size is determined by comparing detection probabilities c_j (2) between different image sizes. By keeping track of the scaling factors of the image and the geometry silhouettes, the actual apparent object size is determined. In the end, a short list of best-matching object parameters (position, size, and orientation) is obtained.

5 Parameter Refinement

So far, it is implicitly assumed that object silhouette is independent of position in the image plane. Unfortunately, this is true only for parallel projection. To account for finite focal length, the object model is re-rendered for the few most probable positions in the image plane. Within a small range around the detected parameter values, the search is refined by using smaller angular step sizes $\Delta\alpha, \Delta\beta, \Delta\gamma$ to enhance parameter accuracy. By additionally varying intrinsic camera parameters, it is also possible to estimate, e.g., the camera's focal length.

6. Results

Fig.2d depicts the autonomously detected position, size, and pose of the pyramid shown in Fig.2a using an approximate model of the object. The camera is not calibrated, it is only assumed that the image pixels are square in dimension and that the field of view is small enough such that parallel projection suffices. Fig.3a illustrates the algorithm's capability to automatically detect different objects. All objects have been modeled by hand using only a ruler. Note that the cube is identified correctly, even though numerous silhouette edge pixels are missing in Fig.2b, demonstrating the robustness of the detection scheme. The coarse candy bar outline originates from rendering the model at a smaller silhouette circumference length N^{sil} than actually present in the image. Fig.3b shows the same scene from a different perspective, and again all objects are autonomously detected. While the handle of the mug is not modeled, it does not interfere with the correct identification of the mug's modeled cylindrical body. The coarse silhouette match of the candy bar is again due to a too small circumference length during silhouette outline generation.

Since the generalized Hough transform is rather insusceptible to uncorrelated noise, the algorithm is found to return robust detection results even if confronted with excessively many edge pixels. Small step sizes $\Delta\alpha, \Delta\beta, \Delta\gamma$ during object outline generation (Sect. 2) prevent misidentifications in crowded image scenes. Detected parameter accuracy depends on the number of rendered silhouette outline pixels and can be continuously traded for computation time.

7. Conclusions

In this paper, an object recognition scheme is described that is based on matching projected 2-D silhouette outlines of the object's 3-D geometry model to edge pixels in the image. An extensive search is followed by a refinement step to autonomously detect the object's best-matching pose, size, and position in the image. No initial search parameter values need to be provided. In its presented form, the algorithm is intended to aid in extrinsic camera calibration by automatically detecting known objects in the camera's field of view.

The algorithm's high degree of parallelism lends itself to implementation on programmable graphics hardware in order to attain fast computational performance. With ever increasing programming flexibility, today's graphics boards rapidly evolve towards general-purpose image processing hardware, making available extremely high parallel-processing power at very low cost. Many computer vision tasks will be able to benefit from this development.

8 Acknowledgments

The author gratefully acknowledges a Feodor Lynen postdoctoral research fellowship from the Alexander von Humboldt Foundation, Germany, which allowed him to prepare the presented work while working as a Research Associate at Stanford University, USA.

References

[1] J. Aggarwal, J. Ghosh, D. Nair, and I. Taha. A comparative study of three paradigms for object recognition - bayesian statistics, neural networks and expert systems. In *Image Understanding: A Festschrift for Azriel Rosenfeld*, pages 241–262. IEEE Computer Society Press, 1996.

[2] D. Ballard. Generalized hough transform to detect arbitrary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):111–122, 1981.

[3] J. Beveridge and E. Riseman. Optimal geometric model-matching under full 3D perspective. *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding*, 61(3):351–364, May 1995.

[4] T. M. Breuel. A practical, globally optimal algorithm for geometric matching under uncertainty. *Proc. International Workshop on Combinatorial Image Analysis (IWCIA 2001),* Philadelphia, USA, Oct. 2001.

[5] F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[6] M. Costabile and G. Pieroni. Detecting shape correspondences by using the generalized hough transform. *Proc. International Conference on Pattern Recognition (ICPR'86),* Paris, France, pages 589–591, Oct. 1986.

[7] E. R. Davies. Reduced parameter spaces for polygon detection using the generalized hough transform. *Proc. International Conference on Pattern Recognition (ICPR'86),* Paris, France, pages 495–497, Oct. 1986.

[8] L. H. Jamieson, E. J. Delp, C.-C. Wang, J. Li, and F. J. Weil. A software environment for parallel computer vision. *IEEE Computer*, 25(2):73–77, 1992.

[9] F. Jurie. Solution of the simultaneous pose and correspondence problem using gaussian error model. *Computer Vision and Image Understanding*, 73(3):357–373, 1999.

[10] H. Kollnig and H. Nagel. Pose estimation by directly matching polyhedral models to gray value gradients. *International Journal of Computer Vision*, 23(3):283–302, 1997.

[11] H. Lensch, W. Heidrich, and H.-P. Seidel. Automated texture registration and stitching for real world models. *IEEE Proc. 8th Pacific Conf. Computer Graphics and Applications (PG-00),* Hong Kong, China, pages 317–326, October 2000.

[12] A. Pope. Model-based object recognition - A survey of recent research. Technical Report TR-94-04, Dept. Computer Science, Univ. British Columbia, Jan. 1994.

[13] R. Strzodka and M. Rumpf. Level set segmentation in graphics hardware. *Proc. IEEE International Conference on Image Processing (ICIP-2001),* Thessaloniki, Greece, 3:1103–1106, Oct. 2001.