NORSIGD INFO

Nummer 2 1999

NORSK SAMARBEID INNEN GRAFISK DATABEHANDLING

ISSN 0803-8317

The CaRo Project

A Robot Guided Camera as an Input Device in Computer Graphics

Alfred Schmitt, Michael Fautz, Peter Oel[2], Universität Karlsruhe

The CaRo project (CaRo = Camera Robot) at the University of Karlsruhe uses a new approach to solve the problem of the acquisition of geometry and surface data of three-dimensional objects. Within the CaRo project a robot arm moves a camera according to the acquisition strategy and points it from various directions towards the object which should be digitized, e.g. a workpiece of mechanical engineering. Using image analysis algorithms the coordinates of surface points of the workpiece can be determined.

Our method has several advantages. The camera can be controlled in an adaptive way by the analysis software. It can easily be switched between global views and detail enlargements. Texture and color as well as the geometry data can be supplied since the surface of the workpiece can be represented using high-resolution color images. The flexibility of the approach becomes clear by looking at the different requests that can be fulfilled. It is even possible to digitize books with cambered pages, which means not getting a three-dimensional representation but getting a textual representation. Furthermore three-dimensional data can be produced. This is needed e.g. in reverse engineering and in advertising and film industry (virtual worlds, computer animations and commercial trick film).

The digitization hardware known so far does not provide the required width and flexibility for object digitization. There is the justified hope that the CaRo approach, the movable camera eye, will become a standard input technique in computer graphics.

In this note we give an overview of the hardware used in the CaRo project and introduce some subprojects. In the next paragraphs the general problem will be described first and a solution will be shown. Then we will focus on two subprojects, object reconstruction and document image analysis.

1 Problem Definition

Whoever is processing graphical data today - especially in the field of 3D - wants to process 3D objects in a flexible way. Concerning the general structure of design systems not only design and manipulation of objects on the graphic display belong to that area, but also import and export functions [9]. By import is meant the recording of the data of objects existing in the real world in order to process them further within the graphics system [2, 7]. For example existing workpieces of mechanical engineering, skeleton bone sections, teeth, plants and leave shapes in biology, and additionally products already existing, such as plates, tools etc. are to be translated in a 3D representation to be used in computer animation and visualization.

The requirements and needs of the users to the imported 3D data are very different. The engineer wants to change the workpiece, thus he must be able to transfer the data into the format, which is needed by his CAD system. The paleontologist wants to measure the pieces of bone exactly, to be able to to draw conclusions from them. With the cooperation of designers and engineers, models are often created at first, whose geometry is then to be transferred to the CAD systems. The computer animator is to produce a commercial video sequence of real products, therefore he needs exact surface textures, to be able to raytrace highly realistic visualizations. While the first case only requires the accurate geometry, the second case needs also information on parameters of the surface as colour, texture, captions etc.

The export function is inverse to the import function, the conversion of 3D data into genuine models, part of which is e.g. the CNC technique and form cutting but other very complex processes as well. Both directions, the input as well as the output of data have always been problematic in graphics processing and are usually solved through specialized processes with a restricted area of application.

In the following we develop a more generalized import function.

[2]email: {aschmitt,fautz,oel}@ira.uka.de

2 Solution

As a technical basis of the CaRo project the following system structure is used: The 3D object to be digitized is placed in the work space of the robot (6 axis, 3 kg load, repeat-accuracy in x-y-z coordinates 0.02 mm, motor: brushless AC servo motor with absolute angle measurement, harmonic drive technique), see figure 1. The robot however is not equipped with a gripping device but guides in place of the gripper a CCD video camera, which supplies a high resolution video signal. The robot can move the camera around the 3D object to view it from all sides, even from above, but of course it cannot view the bottom. The fixed video images thus generated are digitized by a frame-grabber card in full color resolution when needed and are available for the analysis software.

Vision based techniques for object reconstruction are not new. But due to a number of technical problems they have not yet become widely accepted. To be mentioned are image analysis and above all the camera calibration problems, especially determining the exact camera position based only on the picture material. We have already researched the problem how to implement a 3D reconstruction from photos taken from multiple locations at this institute [6] between 1988 and 1991. There were distinct problems with accuracy, especially in deriving the camera locations sufficiently accurate from the image material. It succeeded only roughly, although this is theoretically possible.

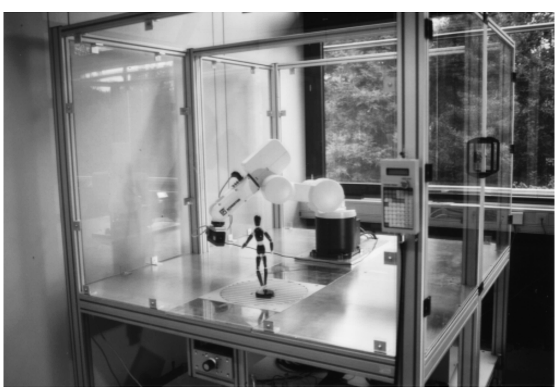

Figure 1: Work space with robot, camera fixed to the gripper of the robot. The model to be digitized, a stylized human puppet, is positioned on the rotation platform, which is sunk into the worktable.

In our approach this problem is solved as the external camera parameters are well-known, because the robot operates in its own fixed coordinate system, so that one knows the camera position as well as the line of sight exactly with each picture taken by the camera.

A further substantial advantage of the CaRo concept arises as a result of the possibility of being able to close up on details of an object with the camera and thus get strongly magnified detail views. With a suitably equipped lens one can almost advance into the microscopic area. Therefore also rather small objects can be scanned. Generally one can say that the resolution of the reconstruction procedure can be given in advance for wide areas and that surfaces can be digitized with exact colouring, which clearly expands the range of application.

With the reconstruction of larger objects the limited work space of the robot has consequences, as the camera cannot view the object from all sides any longer. In such cases the well-known technique of the rotating platform is used. The robot is set up in a suitable distance to the platform. And the object which is to be digitized is fixed on the platform. Now this can be rotated up to an angle of 360 degrees by the controlling software, so that also lateral and rear views are accessible. In particular the silhouette of the object can be obtained.

Worth mentioning as a further advantage is the possibility that the camera can be moved depending on the respective digitization strategy. So we do not have to proceed according to a fixed pattern, but we are able to make use of a variety of strategies depending on the data evaluation and the digitization software [4]. In this way a characteristic point of an object to be digitized can be focused several times and from different views, in order to obtain especially high coordinate accuracy with compensation formulas.

3 Parts of the project

3.1 Technical Environment

As a first part of the project the required hardware was connected and the technical environment (see figure 1) was prepared. In order to develop application software from different workstations, we set up a client-server architecture. Remote computers can get access to the CaRo machinery over the net. Using this architecture we can for example telemanipulate the robot via WWW and the Internet.

3.2 Object reconstruction

One part of the project deals with the reconstruction of threedimensional objects. In contrast to other object scanning techniques as laser range scanners or the structured lighting approach we don't want to get a dense point set of the object to be reconstructed.

We rather want a small data volume and nevertheless a good object approximation. Thereby the CaRo technique is used - the selective use of global views and detail enlargements. Beginning with a simple geometry the object representation gets refined up to the desired quality by adding more images from different views. A small data volume is obtained by this approach as the mesh density of the reconstructed object is correlated with the granularity of the object to be reconstructed. In this way the loss of accuracy by an additional reduction of the measurement data is avoided.

The actual status of the work

In our project up to now we have algorithms implemented for object segmentation from color images and a novel volumetric intersection algorithm which was developed by M. Löhlein in his diploma thesis. As a data structure for 3d object representation a boundary representation (b-rep) is used. In contrast to approaches using a voxel representation for volume intersection one can describe a surface in a geometric way without the loss of accuracy obtained by discretization. As a disadvantage the b-rep intersection algorithm is much more complicated. Actually there are two processing steps to get a coarse approximation of the object to be reconstructed:

In a first step the object to be reconstructed is identified in the color image and its contour is approximated with a polygonal boundary. Holes can also be detected. This silhouette together with the center of projection of the camera builds a generalized cone where the silhouette lines represent its profile and the center of projection the top of the cone.

The second step is to intersect such a cone with the object reconstructed so far. Initially there is used a cube which represents the working area of CaRo. Repeating the two processing steps the cube is intersected with silhouette cones and becomes an approximation of the visual hull [5] of the object. In the following a brief description of the object identifying algorithm and the volumetric intersection method is given.

Object segmentation with color images

The aim is to get polygonal boundaries which represent the contour of the object in the image plane. To get the object and hole regions we assume that every region is enclosed by edges of the image. Therefore an edge detection in the RGB image and in an image resulting from color differences is done. Afterwards the background of the image is identified and colors of the inner of the regions are analyzed and compared to the colors classified from the background. Doing this, object and hole boundaries can be identified. In a finishing step the identified regions are approximated by polygonal boundaries - the silhouette of the object.

The results are amazing. As shown in figure 2 we are able to identify even a black-white checked cup in front of a black-white background.

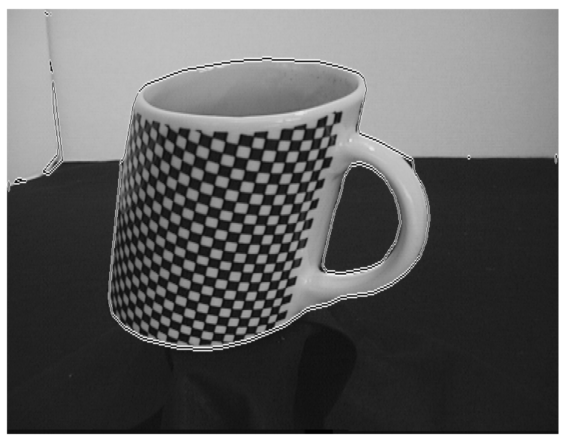

Figure 2: The segmentation result of a black-white coffee cup in front of a black-white background.

Volumetric intersection using a boundary representation

Universal intersection algorithms with b-reps are very difficult to implement. Fortunately we don't have to solve the general b-rep intersection problem. In our case we only intersect a generalized cone with an object given in b-rep which allows us to reduce the 3d intersection problem to a 2d polygon intersection. Two processing steps are needed to solve this problem:

1. Intersection of the 2d silhouette polygons with the projection of the object reconstructed so far to the 2d image plain,

2. closing the gaps that are produced by the intersection step.

Reduction to a 2d intersection problem

To get a 2d problem we first have to project the generalized cone and the current object approximation (the b-rep data structure) to the 2d image plane (see Figure 3). Actually only the object approximation needs to be projected as the cone's contour is already known by the first processing step, the object segmentation.

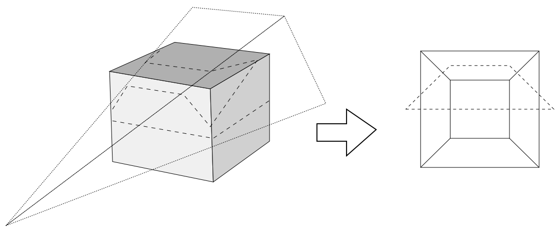

Figure 3: Projection to the 2d plane.

For each polygon which corresponds to a facet of the approximated object an intersection with the cone's contour polygons is done.
Now let O be a polygon of the reconstructed object and C be a cone's polygon; both projected to the image plane of the cone's image. The polygons are oriented so that the inner is located at its right side. This means that O is always oriented CW and C can be oriented CW or CCW (if it's a hole). The intersection of two polygons O, C is done with a well known algorithm [8] as follows:

- Find an edge of O that intersects an edge of C.
- If the O's edge enters the polygon C then beginning with O alternately report the edges of O and the edges of C until the next cut is reached. Otherwise if O's edge leaves the polygon C then start reporting with the edges of C.
- Repeat until all intersection points are processed.

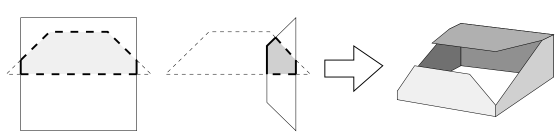

Figure 4: Intersection of the front plane and the right plane of the cube. All resulting polygons are projected back to the object space.

It is possible that no cut appears at all. Then one must check the polygons on inclusion. There are four possible cases:

- there is no inclusion,
- O is included in C,
- C is included in O,
- O and C include one another.

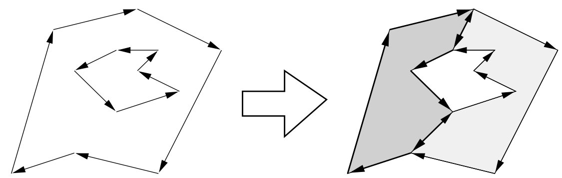

Figure 5: Polygons containing a hole have to been split into two parts.

In the first case the intersection of O and C is empty, in the second and third case the included polygon is the intersection result. The interesting fourth case occurs if C describes a hole and lies inside of O. It is obvious that the result of the intersection must be O without C. To keep the b-rep polygons simple we have to split O and C into two parts and connect each half of O with a half of C so we get two simple polygons as shown in Figure 5.

Now we have to project the resulting polygons of the intersection O, C from the image plane back to the plane defined by O. What we get is a b-rep object with some gaps in it similar to the one shown in Figure 4.

Closing the gaps

Taking a closer look at the intersection result received so far, one can see that the gaps result from the cut by the facets of the generalized cone. Therefore we have to find the parts of each side of the cone which covers the gap. This can be done as follows:

- For each side of the generalized cone collect the edges lying within this side. This can easily be done during the 2d intersection. When we walk along an edge of the cone's polygon then store the corresponding 3d line in a list connected to this side of the cone.

In some cases there are still a few edges missing. These edges are parts of the 3d-cone's edges. In the 2d-projection these edges correspond to the corners of the cone's polygon. In Figure 4 the intersection with the cube's front and top plane shows that case.

- Whenever during the 2d-intersection a corner of the cone's polygon is reached, this corner marks a start or an endpoint of a missing edge. Store all these points sorted by the distance from the center of projection in a list connected to this edge of the cone. The edges can be obtained by taking two successive points.
- Now search for cycles in the edge list connected to each side of the cone. Each cycle forms a polygon which closes one gap in the surface.

Note that some of the cycles found may describe holes. These polygons must be cut from the surrounding polygon as already described in the previous section (see Figure 5).

After this step the intersection of the b-rep and the generalized cone is complete. Figure 6 shows the reconstruction result received by the intersection of the silhouette cones extracted from eight synthesized images.

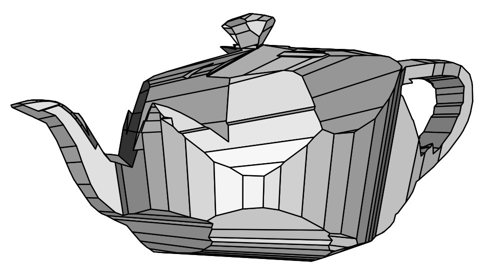

Figure 6: A teapot reconstructed from eight images.

3.3 Document image analysis

Another subproject focuses on using adaptive techniques in document image analysis. The classical approach is divided into four steps:

1. optical scanning and digitization,
2. block segmentation and labeling,
3. processing text and images, if needed: Getting meta information like font styles and layout structure,
4. building a document representation nearly equal to the source document.

Usually step 1, the optical scanning, is done by using a flat scanner. Problems arise when so called "non-flat" pages should be processed, that are documents that can not (e.g. label of a bottle) or must not (e.g. ancient books) lie flat. The usual flat scanner was not designed to scan such documents.

Thinking about a sensible solution the idea to use a robot-guided camera (CaRo) as a scanner was born. Such a camera which has six degrees of freedom has various characteristics: The page of an opened book, which lies usually curved, can be focused from different angles to ensure an orthogonal view of the paper. That of course implies that the final pixel image must be assembled from many overlapping single shootings. The camera may also vary the magnification and thus adapt to the necessary degree of detail depending on the variety of detail of the image content. Summed up, the CaRo hardware is used to do the optical scanning.

We want to use the flexibility provided by CaRo not only in step 1, but also in step 2 to step 4. For example: At a lower resolution level less storage space is needed and algorithms run faster. Currently realized are solutions for step 2 and a first version for step 3.

Block segmentation and classification is done at a very low resolution of about 60 dpi using a combination of Fletcher and Kasturi's connected component analysis [3] with the technique introduced by Wahl, Wong, and Casey [10]. See Figure 7 for some results. The actual work tries to take more advantage of multiresolution methods like the ones discussed by Cinque, Lombardi, and Manzini in [1].

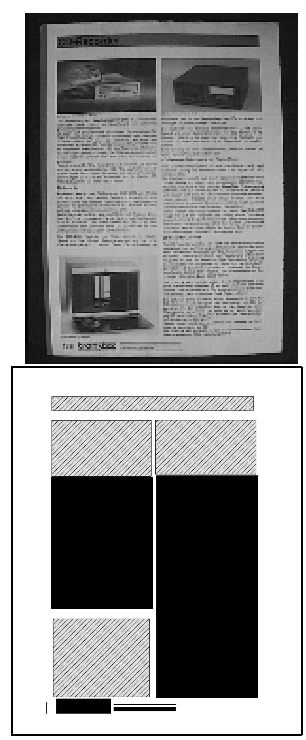

Figure 7: Original document image and the block

segmentation and classification result. The darker blocks represent text regions, the lighter ones images.

Actual work focuses on finding an adaptive solution for the optical character recognition (OCR) problem of step 3. We use an adaptive technique, where the first recognition is done at a lower resolution. Problematic characters are looked at more closely and at a higher resolution. Thus one arrives at a far-reaching analogy to the reading eye, humans fix hardly recognizable items by closing up, too.

Conclusion

Summed up, the approach introduced here has a number of important advantages, which let us hope that we found a flexible solution for the import problem of graphical data processing. The research project however is of a remarkable complexity and demands extensive experimental works as well as the development of new software architectures to obtain reliable statements about suitability in practical work.

Referanser

[1] L. Cinque, L. Lombardi, and G. Manzini. A multiresolution approach for page segmentation. *Pattern Recognition Letters*, 19:217–225, 1998.

[2] O. Faugeras. *Three-Dimensional Computer Vision, a Geometric Viewpoint.* MIT Press, Cambridge, Massachusetts, 1993.

[3] Lloyed Alan Fletcher and Rangachar Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[4] R. A. Jarvis. A perspective on range finding techniques for computer vision. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 122–139, 1983.

[5] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[6] W. Leister. *Geometrisches Modellieren durch interaktive Rekonstruktionsmethoden.* PhD thesis, Fakultät für Informatik, Universität Karlsruhe, 1991.

[7] B. Schunck R. Jain, R. Kasturi. *Machine vision.* McGraw-Hill, New York, 1995.

[8] A. Schmitt, O. Deussen, and M Kreeb. *Einführung in graphisch-geometrische Algorithmen.* Teubner, Stuttgart, 1995.

[9] A. A. Schmitt. *Dialogsysteme - Kommunikative Schnittstellen, Software-Ergonomie und Systemgestaltung.* BI Wissenschaftsverlag, Mannheim, 1983.

[10] Friedrich M. Wahl, Kwan Y. Wong, and Richard G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20:375–390, 1982.