Computing Box Dimensions from Single Perspective Images in Real Time

Leandro A. F. Fernandes [1] Manuel M. Oliveira [1] Roberto da Silva [1] Gustavo J. Crespo [2]

[1] UFRGS – Instituto de Informática [2] Stony Brook University

{laffernandes, oliveira, rdasilva}@inf.ufrgs.br

Abstract

We present a new method for computing the dimensions of boxes from single perspective projection images in real time. Given a picture of a box, acquired with a camera whose intrinsic parameters are known, the dimensions of the box are computed from the extracted box silhouette and the projection of two parallel laser beams on one of its visible faces. We also present a statistical model for background removal that works with a moving camera, and an efficient voting scheme for identifying approximately collinear segments in the context of a Hough Transform. We demonstrate the proposed approach and algorithms by building a prototype of a scanner for computing box dimensions and using it to automatically compute the dimensions of real boxes. The paper also presents some statistics over measurements obtained with our scanner prototype.

1 Introduction

The ability to measure the dimensions of three-dimensional objects directly from images has many practical applications including quality control, surveillance, analysis of forensic records, storage management and cost estimation. Unfortunately, unless some information relating distances measured in image space to distances measured in 3D is available, the problem of making measurements directly on images is not well defined. This is a result of the inherent ambiguity of perspective projection caused by the loss of depth information.

This paper presents a method for computing boxes dimensions from single perspective projection images in a completely automatic way. This can be an invaluable tool for companies that manipulate boxes on their day-by-day operations, such as couriers, airlines and warehouses. The approach uses information extracted from the silhouette of the target boxes and can be applied when at least two of their faces are visible, even when the target box is partially occluded by other objects in the scene (Figure 1). We eliminate the inherent ambiguity associated with perspective images by projecting two parallel laser beams, apart from each other by a known distance and perpendicular to the camera's image plane, onto one of the visible faces of the box. We demonstrate this technique by building a scanner prototype for computing box dimensions and using it to compute the dimensions of boxes in real time (Figure 1).

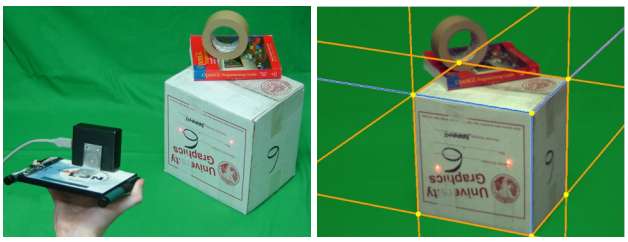

Figure 1. Scanner prototype: (left) Its operation. (right) Camera's view with recovered edges and vertices.

The main contributions of this paper include:

- An algorithm for computing the dimensions of boxes in a completely automatic way in real-time (Section 3);
- An algorithm for extracting box silhouettes in the presence of partial occlusion of the box edges (Section 3);
- A statistical model for detecting background pixels under different lighting conditions, for use with a moving camera (Section 4);
- An efficient voting scheme for identifying approximately collinear line segments using a Hough Transform (Section 5).

2 Related Work

Many optical devices have been created for making measurements in the real world. Those based on active techniques project some kind of energy onto the surfaces of the target objects and analyze the reflected energy. Examples of

active techniques include optical triangulation [1] and laser range finding [18] to capture the shapes of objects at proper scale [13], and ultrasound to measure distances [6]. In contrast, passive techniques rely only on the use of cameras for extracting the three-dimensional structure of a scene and are primarily based on the use of stereo [14]. In order to achieve metric reconstruction [9], both optical triangulation and stereo-based systems require careful calibration. For optical triangulation, several images of the target object with a superimposed moving pattern are usually required for more accurate reconstruction.

Labeling schemes for trihedral junctions [3, 11] have been used to estimate the spatial orientation of polyhedral objects from images. These techniques tend to be computationally expensive when too many junctions are identified. Additional information from the shading of the objects can be used to improve the labeling process [21]. Silhouettes have been used in both computer vision and computer graphics for object shape extraction [12, 17]. These techniques require precise camera calibration and use silhouettes obtained from multiple images to define a set of cones whose intersections approximate the shapes of the objects.

Criminisi *et al.* [4] presented a technique for making 3D affine measurements from a single perspective image. They show how to compute distances between planes parallel to a reference one. In case of some distance from a scene element to the reference plane is known, it is possible to compute the distances between scene points and the reference plane. The technique requires user interaction and cannot be used for computing dimensions automatically. Photogrammetrists have also made measurements based on single images. However, these techniques can only be applied to planar objects and require user intervention.

In a work closely related to ours, Lu [16] described a method for finding the dimensions of boxes from single gray-scale images. In order to simplify the task, Lu assumes that the images are acquired using parallel orthographic projection and that three faces of the box are visible simultaneously. The computed dimensions are approximately correct up to a scaling factor. Also, special care is required to distinguish the actual box edges from lines in the box texture, causing the method not to perform in real time.

Our approach computes the dimensions of boxes from single perspective projection images, producing metric reconstructions in real time and in a completely automatic way. The method can be applied to boxes with arbitrary textures, can be used when only two faces of the box are visible, even when the edges of the target box are partially occluded by other objects in the scene.

3 Computing Box Dimensions

We model boxes as parallelepipeds although real boxes can present many imperfections (*e.g.*, bent edges and corners, asymmetries, etc.). The dimensions of a parallelepiped can be computed from the 3D coordinates of four of its non-coplanar vertices. Conceptually, the 3D coordinates of the vertices of a box can be obtained by intersecting rays, defined by the camera's center and the projections of the box vertices on the camera's image plane, with the planes containing the actual faces of the box in 3D. Thus, before we can compute the dimensions of a given box (Section 3.3), we need to find the projections of the vertices on the image (Section 3.1), and then find the equations of the planes containing the box faces in 3D (Section 3.2).

In the following derivations, we assume that the origin of the image coordinate system is at the center of the image, with the X-axis growing to the right and the Y-axis growing down, and assume that the imaged boxes have three visible faces. The case involving only two visible faces is similar. Also, assume that the images used for computing the dimensions were obtained through linear projection (*i.e.*, using a pinhole camera). Although images obtained with real cameras contain some amount of radial distortion, such distortions can be compensated for with the use of simple warping procedures [9].

3.1 Finding the Projections of the Vertices

The projection of the vertices can be obtained as the corners of the box silhouette. Although edge detection techniques [2] could be used to find the box silhouette, these algorithms tend to be sensitive to the presence of other high-frequency contents in the image. In order to minimize the occurrence of spurious edges and support the use of boxes with arbitrary textures, we perform silhouette detection using a model for the background pixels. Since the images are acquired using a handheld camera, proper modeling of the background pixels is required and will be discussed in detail in Section 4.

However, as shown in Figure 2 (a, b and c), a naive approach that just models the background and applies simple image processing operations, like background removal and high-pass filtering, does not properly identify the silhouette pixels of the target box (selected by the user by pointing the laser beams onto one of its faces). This is because the scene may contain other objects, whose silhouettes possibly overlap with the one of the target box. Also, the occurrence of some misclassified pixels (see Figure 2, c) may lead to the detection of spurious edges. Thus, a suitable method was developed to deal with these problems. The steps of our algorithm are shown in Figures 2 (a, d, e, f and g).

The target box silhouette is obtained starting from one

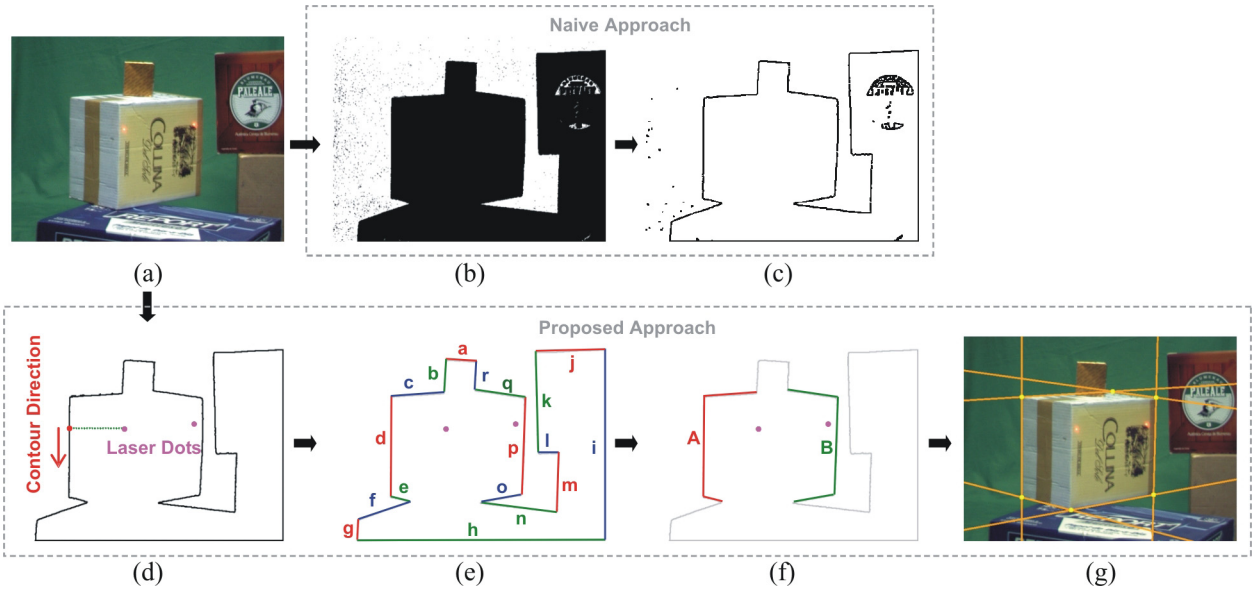

Figure 2. Identifying the target box silhouette. Naive approach: (b) Background segmentation, followed by (c) High-pass filter (note the spurious "edge" pixels). Proposed approach: (d) Contouring of the foreground region, (e) Contour segmentation, (f) Grouping candidate segments for the target box silhouette, and (g) recovery of supporting lines for silhouette edges and vertices.

of the laser dots, finding a silhouette pixel and using a contour-following procedure [8]. The seed silhouette pixel for the contour-following procedure is found stepping from the laser dot within the target foreground region and checking whether the current pixel is matches the background model. In order to be a valid silhouette, both laser dots need to fall inside of the contouring region. Notice this procedure produces a much cleaner set of border pixels (Figure 2, d) compared to results shown in Figure 2 (c). But the resulting silhouette may include overlapping objects, and we need to identify which border pixels belong to the target box. To facilitate the handling of the border pixels, the contour is subdivided into its most perceptually significant straight line segments [15] (Figure 2, e). Then, the segments resulting from the clipping of a foreground object against the limits of the frame (*e.g.*, segments h and i in Figure 2, e) are discarded. Since a box silhouette defines a convex polygon, the remaining segments whose two endpoints are not *visible* by both laser dots can also be discarded. This test is performed using a 2D BSP-tree [7]. In the example of Figure 2, only segments c, d, e, o, p and q pass this test.

Still, there is no guarantee that all the remaining segments belong to the target box silhouette. In order to restrict the amount of possible combinations, the remaining chains of segments defining convex fragments are grouped (*e.g.*, groups A and B in Figure 2, f). We then try to find the largest combination of groups into valid portions of the silhouette. In order to be considered a valid combination, the groups must satisfy the following validation rules: (i) they must characterize a convex polygon; (ii) the silhouette must have six edges (the silhouette of a parallelepiped with at least two visible faces); (iii) the laser dots must be at the same box face; and (iv) the computed lengths for pairs of parallel edges in 3D must be approximately the same. In the case of more than one combination of groups pass the validation tests, the system discards this ambiguous data and starts processing a new frame (our system is capable of processing frames at the rate of about 34 fps).

Once the box silhouette is known, the projections of the six vertices are obtained intersecting pairs of adjacent supporting lines for the silhouette edges (Figure 2, g). Section 5 discusses how to obtain those supporting lines.

3.2 Computing the Plane Equations

The set of all parallel lines in 3D sharing the same direction intersect at a point at infinite whose image under perspective projection is called a vanishing point ω. The line defined by all vanishing points from all sets of parallel lines on a plane Π is called the vanishing line λ of Π. The normal vector to Π in a given camera's coordinate system can be obtained multiplying the transpose of the camera's intrinsic-parameter matrix by the coefficients of λ [9]. Since the resulting vector is not necessarily a unit vector, it needs to be normalized. Equations (1) and (2) show the re-

lationship among the vanishing points ω_i, vanishing lines λ_i and the supporting lines e_j for the edges that coincide with the imaged silhouette of a parallelepiped with three visible faces. The supporting lines are ordered clockwise.

$$\omega_i = e_i \times e_{i+3} \quad (1)$$

$$\lambda_i = \omega_i \times \omega_{(i+1) mod 3} \quad (2)$$

where $0 \leq i \leq 2$, $0 \leq j \leq 5$, $\lambda_i = (a_{\lambda_i}, b_{\lambda_i}, c_{\lambda_i})^T$ and $\times$ is the cross product operator. The normal N_{Π_i} to plane Π_i is then given by

$$N_{\Pi_i} = \frac{RK^T\lambda_i}{\|RK^T\lambda_i\|} \quad (3)$$

where $N_{\Pi_i} = (A_{\Pi_i}, B_{\Pi_i}, C_{\Pi_i})$, $0 \leq i \leq 2$. K is the matrix that models the intrinsic camera parameters [9] and R is a reflection matrix (Equation 4) used to make the Y-axis of the image coordinate system grows in the up direction.

$$K = \begin{pmatrix} \frac{f}{s_x} & \gamma & o_x \\ 0 & \frac{f}{s_y} & o_y \\ 0 & 0 & 1 \end{pmatrix} \quad R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

In Equation (4), f is the focal length, and s_x and s_y are the dimensions of the pixel in centimeters. γ, o_x and o_y represent the skew and the coordinates of the principal point, respectively.

Once we have N_{Π_i}, finding D_{Π_i}, the fourth coefficient of the plane equation, is equivalent to solving the projective ambiguity and will require the introduction of one more constraint. Thus, consider the situation depicted in 2D in Figure 3 (right), where two laser beams, parallel to each other and to the camera's XZ plane, are projected onto one of the faces of the box. Let the 3D coordinates of the laser dots defined with respect to the camera coordinate system be $P_0 = (X_{P_0}, Y_{P_0}, Z_{P_0})^T$ and $P_1 = (X_{P_1}, Y_{P_1}, Z_{P_1})^T$, respectively (Figure 3, left). Since P_0 and P_1 are on the same plane Π, one can write

$$A_\Pi X_{P_0} + B_\Pi Y_{P_0} + C_\Pi Z_{P_0} = A_\Pi X_{P_1} + B_\Pi Y_{P_1} + C_\Pi Z_{P_1} \quad (5)$$

Using the linear projection model and given $p_i = (x_{p_i}, y_{p_i}, 1)^T$, the homogeneous coordinates of the pixel associated with the projection of point P_i, one can reproject p_i on the plane $Z = 1$ (in 3D) using

$$p'_i = RK^{-1}p_i \quad (6)$$

and express the 3D coordinates of the laser dots on the face of the box as

$$X_{P_i} = x_{p'_i} Z_{P_i} \;\; , \;\; Y_{P_i} = y_{p'_i} Z_{P_i} \;\; and \;\; Z_{P_i} \quad (7)$$

Substituting the expression for X_{P_0}, Y_{P_0}, X_{P_1} and Y_{P_1} (Equation 7) in Equation (5) and solving for Z_{P_0}, we obtain

$$Z_{P_0} = kZ_{P_1} \quad (8)$$

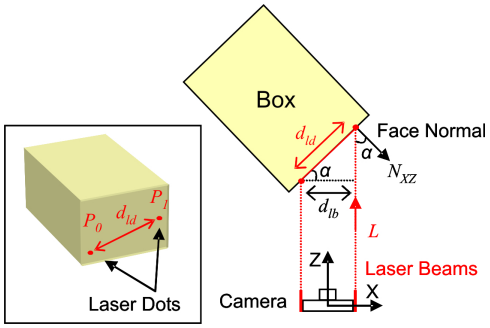

Figure 3. Top view of a scene. Two laser beams apart in 3D by d_{lb} project onto one box face at points P_0 and P_1, whose distance in 3D is d_{ld}. α is the angle between $-L$ and N_{XZ}.

where

$$k = \frac{A_\Pi x_{p'_1} + B_\Pi y_{p'_1} + C_\Pi}{A_\Pi x_{p'_0} + B_\Pi y_{p'_0} + C_\Pi} \quad (9)$$

Now, let d_{lb} and d_{ld} be the distances, in 3D, between the two parallel laser beams and between the two laser dots projected onto one of the faces of the box, respectively (Figure 3). Section 6 discusses how to find the laser dots on the image. d_{ld} can be directly computed from N_Π, the normal vector of the face onto which the dots project, and the known distance d_{lb}:

$$d_{ld} = \frac{d_{lb}}{\cos(\alpha)} = \frac{d_{lb}}{-(N_{XZ} \cdot L)} \quad (10)$$

where α is the angle between N_{XZ}, the normalized projection of N_Π onto the plane defined by the two laser beams. By construction, such a plane is parallel to the camera's XZ plane. Therefore, N_{XZ} is obtained by dropping the Y coordinate of N_Π and normalizing the resulting vector. $L = (0, 0, 1)^T$ is the vector representing the laser beam direction. d_{ld} can also be expressed as the Euclidean distance between the two laser dots in 3D:

$$d_{ld}^2 = (X_{P_1} - X_{P_0})^2 + (Y_{P_1} - Y_{P_0})^2 + (Z_{P_1} - Z_{P_0})^2 \quad (11)$$

Substituting Equations (7), (8) and (10) into (11) and solving for Z_{P_1}, one gets

$$Z_{P_1} = \sqrt{\frac{d_{ld}^2}{ak^2 - 2bk + c}} \quad (12)$$

where $a = (x_{p'_0})^2 + (y_{p'_0})^2 + 1$, $b = x_{p'_0} x_{p'_1} + y_{p'_0} y_{p'_1} + 1$ and $c = (x_{p'_1})^2 + (y_{p'_1})^2 + 1$. Given Z_{P_1}, the 3D coordinates of P_1 can be computed as

$$P_1 = (X_{P_1}, Y_{P_1}, Z_{P_1}) = (x_{p'_1} Z_{P_1}, y_{p'_1} Z_{P_1}, Z_{P_1}) \quad (13)$$

The projective ambiguity can be finally removed by computing the D_Π coefficient for the plane equation of the face containing the two dots:

$$D_\Pi = -(A_\Pi X_{P_1} + B_\Pi Y_{P_1} + C_\Pi Z_{P_1}) \qquad (14)$$

3.3 Computing the Box Dimensions

Having computed the plane equation, one can recover the 3D coordinates of vertices of that face. For each such vertex v on the image, we compute v' using Equation (6). We then compute its corresponding Z_V coordinate by substituting Equation (7) into the plane equation for the face. Given Z_V, both X_V and Y_V coordinates are computed using Equation (7). Since all visible faces of the box share some vertices with each other, the D coefficients for the other faces of the box can also be obtained, allowing the recovery of the 3D coordinates of all vertices on the box silhouette, from which the dimensions are computed.

Although not required for computing the dimensions of the box, the 3D coordinates of the inner vertex (see Figure 1, right) can also be computed. Its 2D coordinates can be obtained as the intersection between three lines (Figure 1, right). Each such a line is defined by a vanishing point and the silhouette vertex falling in between the two box edges used to compute that vanishing point. Since it is unlikely that these three lines will intersect exactly at one point, we approximate it using least-squares. Given the inner vertex 2D coordinates, its corresponding 3D coordinates can be computed using the same algorithm used to compute the 3D coordinates of the other vertices.

4 A Model for Background Pixels

One of the most popular techniques for object segmentation is chroma keying [20]. Unfortunately, standard chroma keying techniques do not produce satisfactory results for our application. Shading variations in the background and shadows cast by the boxes usually lend to misclassification of background pixels. Horprasert *et al.* [10] describe a statistical method that computes a per-pixel model of the background from a set of static background images. While this technique is fast and produces very good segmentation results for scenes acquired from a static camera, it is not appropriate for use with moving cameras.

In order to support a moving camera, we developed an approach that works under different lighting conditions using a background with a known color. The approach computes a statistical model of the background considering multiple possible shades of the background color and proved to be robust, lending to very satisfactory results.

The algorithm takes as input a set of n images I_i of the background acquired under different lighting conditions. In the first step, we compute E, the average color of all pixels in images I_i, and the eigenvalues and eigenvectors associated with the colors of those pixels. E and the eigenvector associated with the highest eigenvalue define an axis in the RGB color space, called the *chromaticity axis*. The chromaticity distortion d of a given color C can be computed as the distance from C to the chromaticity axis.

After discarding the pixels whose projections on the chromaticity axis have at least one saturated channel (they lend to misclassification of bright foreground pixels), we divide the chromaticity axis into m slices (Figure 4). For each slice, we compute $\bar{d}_j$ and $\sigma_{\bar{d}_j}$, the mean and the standard deviation, respectively, for the chromaticity distortion of the pixels in the slice. Then, we compute a threshold d_{Tj} for the maximum acceptable slice chromaticity distortion considering a confidence level of 99% as $d_{Tj} = \bar{d}_j + 2.33\sigma_{\bar{d}_j}$.

Finally, the coefficients of the polynomial background model are computed by fitting a curve through the d_T values at the centers of the slices. Once the coefficients have been computed, the d_T values are discarded and the tests are performed against the polynomial. Figure 4 illustrates the case of a color C_k being tested against the background color model. C'_k is the projection of C_k on the chromaticity axis. In this example, as the distance between C_k and the chromaticity axis is bigger than the threshold defined by the polynomial, C_k will be classified as foreground.

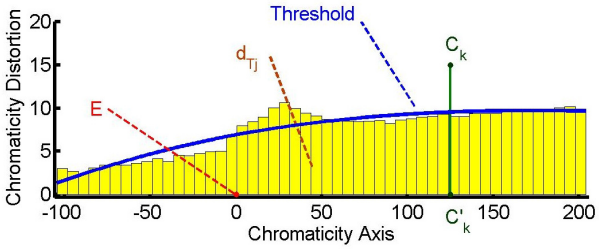

Figure 4. Chromaticity axis. The curve is the polynomial fit to the chromaticity distortions.

Changing the background color only requires obtaining samples of the new background and computing the new values for the chromaticity axis and the coefficients of the polynomial. According to our experiments, 100 slices and a polynomial of degree 3 produce very satisfactory results.

5 Identifying Almost Collinear Segments

To compute the image coordinates of the box vertices, first we need to obtain the supporting lines for the silhouette edges. We do this using a Hough Transform procedure [5]. However, the conventional voting process and the detection of the most significant lines had shown to be a bottleneck to

our system. To reduce the amount of computation, an alternative to the conventional voting process was developed.

Although the silhouette pixels are organized into its most perceptually significant straight line segments, we don't known whether two or more of these segments are pieces of the same box edge. The new voting scheme consists in casting votes directly for the segments, instead of for individual pixels. Thus, for each perceptually significant segment, the parameters of the line are computed using the average position of the set of pixels that are represented by the segment and by the eigenvector of the highest eigenvalue of the pixel distribution. The use of the eigenvector allows handling lines with arbitrary orientations in a consistent way.

We distribute the votes in the parameter space by means of a Gaussian kernel, with votes weighted by the segment length. The use of a Guassian kernel distributes the votes around a neighborhood, allowing the identification of approximately collinear segments. This is a very important feature, allowing the system to better handle discretization errors and boxes with slightly bent edges. The size of the used Gaussian kernel was experimentally defined as a 11×11 pixels. Special care must be taken when the θ parameter is close to $0°$ or $180°$. In this situation, the voting process continues in the diagonally opposing quadrant, at the $-\rho$ position (see peaks labeled as d and p in Figure 5).

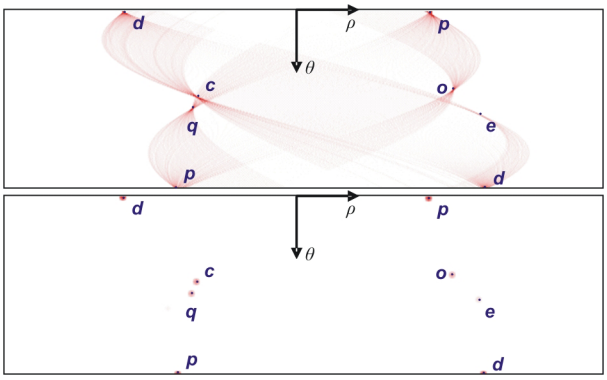

Figure 5. Hough Transform parameter space: conventional (top) and obtained with the new voting scheme (bottom). Labeled dots represent the peaks.

Using the new approach, the voting process and the peak detection are improved because the amount of cells that receive votes is greatly reduced. Figure 5 shows the parameter space after the traditional (top) and the new (bottom) voting processes have been applied to the segments shown in Figure 2 (f). The parameter space was discretized using 360 angular values in the range $\theta = [0, 180]$ as well as 1600 ρ values in the range $[-400, 400]$.

6 Finding the Laser Dots

The ability to find the proper positions of the laser dots in the image can be affected by several factors such as the camera's shutter speed, the box materials and textures, and ambient illumination. Although we are using a red laser (650nm class II), we cannot rely simply on the red channel of the image to identify the positions of the dots. Such a procedure would not distinguish between the laser dot and red texture elements on the box. Since the pixels corresponding to the laser dots present very high luminance, we identify them by thresholding the luminance image. However, just simple thresholding may not work for white boxes or boxes containing white regions, which tend to have large areas with saturated pixels. We solved this problem by setting the camera's shutter speed so that the laser dots are the only elements in the image with high luminance.

Since the image of a laser spot is composed by several pixels, we approximate the actual position of the dot by the centroid of its pixels. According to our experiments, a variation of one pixel in the coordinates of the estimated center of the laser spot produces a variation of a few millimeters in the computed dimensions. These numbers were obtained assuming a camera standing about two meters from the box.

Before the position of the laser dots can be used for computing dimensions, one needs to identify the face onto which the dots fall. This is done after computing the position of the inner vertex (Section 3.3) and checking whether both dots fall inside one of the three quadrilaterals defined by the edges of the box (Figure 1, right).

The system may fail to properly detect the laser dots if they project onto some black region or if the surface contains specularities that can lead to peaks in the luminance image. This, however, can be avoided by aiming the beams on other portions of the box. Due to the construction of the scanner prototype and some epipolar constraints [9], we only need to search for the laser dots inside a small window in the image.

7 Results

We have built a prototype of a scanner for computing box dimensions and implemented the techniques described in the paper using C++. The system was tested on several real boxes. For a typical scene, such as the one shown in Figure 2, it can process the video and compute box dimensions at about 34 fps. As we replace the line-based voting scheme with the traditional pixel-based Hough Transform voting scheme (Figure 5, top), the rate drops to only 9 fps. This illustrates the effectiveness of the proposed voting solution. These measurements were made on a 1.91 GHz PC with 768 MB of memory. A

video illustrating the use of our scanner can be found at *http://www.inf.ufrgs.br/~laffernandes/boxdimensions*.

Figure 1 (left) shows the scanner prototype whose hardware is comprised of a firewire color camera (Point Grey Research DragonFly with 640x480 pixels, with $s_x = s_y = 7.4\mu m$ [19]), a 16mm lens (Computar M1614, with manual focus, no iris and 30.9 degrees horizontal field of view) and two laser pointers. The camera is mounted on the plastic box and the laser pointers were aligned and glued to the sides of this box. In such an assembly, the laser beams are 15.8 cm apart. For our experiments, we set the camera's shutter to 0.01375 seconds and acquired pictures of boxes from distances varying from 1.7 to 3.0 meters to the camera. The background was created using a piece of green cloth and its statistical model was computed from a set of 23 images. Figure 6 shows some examples of boxes used to test our system. The boxes in the bottom row are particularly challenging: the one on the left is very bright and has a reflective plastic finishing; the one on the right is mostly covered with red texture. The dimensions of these boxes vary from 13.9 to 48.3 cm. In our experiments, we assumed that the acquired images have no skew (*i.e.*, $\gamma = 0$) and the principal point is at the center of the image (*i.e.*, $o_x = o_y = 0$). Due to the small field of view, we also assumed the images contain no significant radial distortion.

Figure 6. Examples of boxes used for testing.

The geometry of the box is somewhat different from that of a parallelepiped because of imperfections introduced during the construction process and handling. For instance, bent edges, different sizes for two parallel edges of the same face, lack of parallelism between faces expected to be parallel, and warped corners are not unlikely to be found in practice. Such inconsistencies lend to errors in the orientation of the silhouette edges, which are cascaded into the computation of the box dimensions.

In order to estimate the inherent inaccuracies of the proposed algorithm, we implemented a simulator that performs the exact same computations, but on images generated with a pinhole camera model using computer graphics techniques. In this case, the boxes are exact parallelepipeds. The positions of the laser dots on a box face are determined by intersecting two parallel rays with one box face. As in the case of the physical device, the camera can move freely in the 3D scene. Using images generated by the simulator, our system can recover the dimensions of the box with an average relative error of 1.07%. Next, we analyze some of the results obtained on real boxes.

7.1 Statistical Analysis on Real Boxes

In order to evaluate the proposed approach, we carried out a few statistical experiments. Due to space limitations, we will describe only one of these experiments in detail and will present some data about other results. We selected a well-constructed box (shown in Figure 6, top right) and manually measured the dimensions of all its edges with a ruler. Each edge was measured twice, once per shared face of the edge. The eight measurements of each box dimensions were averaged to produce a single value per dimension. All measurements were made in centimeters. The average values for this box are 29.45 cm, 22.56 cm and 15.61 cm, respectively. We then used our system to collect a total of 30 measurements of each dimension of the same box. For each collected sample, we projected the laser beams on different parts of the box. We used this data to compute the mean, standard deviation and confidence intervals for each of the computed dimensions. The confidence intervals were computed as $CI = \left[\bar{x} - t_\gamma \frac{\sigma}{\sqrt{n}}, \bar{x} + t_\gamma \frac{\sigma}{\sqrt{n}}\right]$, where $\bar{x}$ is the mean, σ is the standard deviation, n is the size of sample and t_γ is a $t-$Student variable with $n-1$ degrees of freedom, such that the probability of a measure x belongs to CI is γ. The tightest the CI, the more precise are the computed values.

Table 1 shows the computed confidence intervals for values of $\gamma = 80\%, 90\%, 95\%$ and 99%. Note that the values of the actual dimensions fall inside these confidence intervals, indicating accurate measurements.

Table 1. Confidence intervals for the measurements for the box in Figure 6 (top right)

$CI(\gamma)$	dim 1	dim 2	dim 3
$CI(80\%)$	$[27.99, 29.60]$	$[21.83, 22.60]$	$[15.12, 15.82]$
$CI(90\%)$	$[27.76, 29.83]$	$[21.72, 22.72]$	$[15.02, 15.92]$
$CI(95\%)$	$[27.55, 30.03]$	$[21.62, 22.81]$	$[14.93, 16.01]$
$CI(99\%)$	$[27.39, 30.20]$	$[21.54, 22.89]$	$[14.86, 16.08]$

Another estimate of the error can be expressed as the relative error $\epsilon = \sigma/\bar{x}$. Table 2 shows data about the dimensions of five other boxes and the relative errors in the

measurements obtained with our scanner prototype. The errors were computed with respect to the average of the computed dimensions over a set of 20 measurements. In these experiments, the operator tried to keep the device still while the samples were collected. The relative errors varied from 0.20% up to 11.20%, which is in accordance with the accuracy predicted by the experiment summarized in Table 1.

Table 2. Relative errors for five real boxes.

Real Size (cm)				Relative Error (%)		
dim 1	dim 2	dim 3	n	dim 1	dim 2	dim 3
35.5	32.0	26.9	68	1.44	2.68	1.40
28.6	24.1	16.8	61	6.81	0.80	11.20
36.0	25.2	13.8	50	5.19	10.15	9.43
29.9	29.1	22.9	69	0.20	5.47	10.22
48.2	45.5	28.3	20	3.98	4.55	0.91

8 Conclusions and Future Work

We have presented a completely automatic approach for computing the dimensions of boxes from single perspective projection images in real time. The approach uses information extracted from the silhouette of the target box and removes the projective ambiguity with the use of two parallel laser beams. We demonstrated the effectiveness of the proposed techniques by building a prototype of a scanner and using it to compute the dimensions of several real boxes even when the edges of the target box are partially occluded by other objects and under different lighting conditions. We also presented a statistical discussion of the measurements made with our scanner prototype.

We have also introduced an algorithm for extracting box silhouettes in the presence of partially occluded edges, an efficient voting scheme for grouping approximately collinear segments using a Hough Transform, and a statistical model for background that works with a moving camera. Our algorithm for computing box dimensions can still be used with applications requiring heterogeneous backgrounds. In these situations, background detection can be performed using a technique like the one described in [10]. In this case, the camera should remain static while the boxes are moved on some conveyor belt.

We believe that these ideas may lend to optimizations on several procedures that are currently based on manual measurements of box dimensions. We are currently exploring ways of using arbitrary backgrounds, the use of a single laser beam, and analyzing the error propagation through the various stages of the algorithm.

References

[1] P. Besl. *Active Optical Range Imaging Sensors. Advances in Machine Vision, in Jorge Sanz (editor)*, pages 1–63. Springer-Verlag, 1988.

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

[3] M. B. Clowes. On seeing things. *Artificial Intelligence*, 2:79–116, 1971.

[4] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, volume I, pages 434–441, Kerkyra, Greece, September 20-27 1999. IEEE.

[5] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 1972.

[6] F. Figueroa and A. Mahajan. A robust method to determine the coordinates of a wave source for 3-D position sensing. *ASME Journal of Dynamic Systems, Measurements and Control*, 116:505–511, September 1994.

[7] H. Fuchs et al. On visible surface generation by a priori tree structures. In *Proc. of SIGGRAPH 1980*, pages 124–133.

[8] J. Gauch. KUIM, image processing system. http://www.ittc.ku.edu/ jgauch/research, 2003.

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

[10] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. of the 7th IEEE ICCV-99, FRAME-RATE Workshop*, 1999.

[11] D. A. Huffman. Impossible objects as nonsense sentences. In *Machine Intelligence 6*, pages 295–324. Edinburg University Press, 1971.

[12] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. on PAMI*, 16(2):150–162, February 1994.

[13] M. Levoy et al. The digital Michelangelo project: 3D scanning of large statues. In *Proc. of SIGGRAPH 2000*, pages 131–144, 2000.

[14] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[15] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, March 1987.

[16] K. Lu. Box dimension finding from a single gray-scale image. Master's thesis, SUNY Stony Brook, New York, 2000.

[17] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proc. of SIGGRAPH 2000*, pages 369–374, 2000.

[18] L. Nyland et al. The impact of dense range data on computer graphics. In *Proc. of Multi-View Modeling and Analysis Workshop (MVIEW99) (Part of IEEE CVPR99)*, 1999.

[19] Point Grey Research Inc. *Dragonfly IEEE-1394 Digital Camera System*, 2.0.1.10 edition, 2002.

[20] P. Vlahos. Composite color photography. U.S. Patent 3.158.477, 1964.

[21] D. L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Technical Report MAC-TR-271, Cambridge, MA, USA, 1972.