Cylindrical Surface Localization in Monocular Vision

William PUECH [a] Jean-Marc CHASSERY [a] and Ioannis PITAS [b]

[a] *TIMC-IMAG Grenoble laboratory, UMR CNRS 5525, Domaine de la Merci, 38706 LA TRONCHE Cedex, FRANCE, William.Puech@imag.fr, Jean-Marc.Chassery@imag.fr*

[b] *Department of Informatics, Aristotle University of Thessaloniki, P.O. Box 451, 54006 THESSALONIKI, GREECE - pitas@zeus.csd.auth.gr*

Abstract

In this paper we present a method to localize a cylindral surface with only one perspective view. Based on a priori knowledge we find two axes in the image in order to obtain the three rotation angles between the cylindrical surface coordinate system and the camera coordinate system. Various applications of the proposed method are presented.

Key words: Monocular vision, straight uniform generalized cylinder, projected cross-sections, perspective projection.

1 Introduction

This paper deals with a new method to localize a cylindrical surface using a single perspective view. In monocular vision, a certain a priori knowledge is needed to perform localization and reconstruction in order to recover a 3D surface from image data. In this work, we limit our study to the case when the surface is a Straight Uniform Generalized Cylinder (SUGC) [1], with Closed cross-sections (circles or ellipses) or Opened cross-sections (parabolas, ellipses or circles). We show how to locate an image on the 3D surface and how to backproject the image in the 3D space. Cross-sections are detected on the cylindrical surface and projected in the image plane. With the projections of these cross-sections, we detect two orthogonal axes giving us the necessary information to determine the three rotation angles of the camera.

[1] The definition will be given in the Subsection 2.2

This work is applied to works of art in order to obtain images of mural painting without geometrical distortions caused by the perspective view of the curvature of the surface. A lot of work has been done in the domain of monocular vision surface localization. The zero-curvature points of contours from an image can be used to solve the localization problem [11]. Localization has also been performed by interpreting a triplet of image lines as the perspective transformation of a triplet of linear ridges of an object model [4]. It is also possible to determine the location and the orientation of a cylinder with a label [15] or by using the radius lighting direction [14]. Another method for curved surface localization using a genetic algorithm is described to be used for works of art [13].

The structure of this paper is as follows. In Section 2, we provide some definitions concerning the camera model and generalized cylinders. Our method consists of three parts. The first part described in Subsection 3.1 deals with finding the projection of the SUGC axis in the image based on a priori knowledge about the projected cross-sections [9]. In Subsection 3.2, we show how to detect the second axis in the image corresponding to the projection of one particular cross-section. After detecting these two axes, we are able to localize a cylindrical surface in the camera coordinate system in Subsection 3.3. In Section 4, synthetic images are used to validate our results and real images illustrate the efficiency of our method.

2 Definitions

2.1 Camera model

In this subsection, we present the notations and the coordinate systems used in this paper, shown in Figure 1:

- The camera coordinate system is defined by the view point O and the focal axis. The focal axis passes through O and is perpendicular to the image plane. The O_x and O_y axes are respectively parallel to the lines and columns of the image.
- The picture coordinate system is a two-dimensional coordinate system (u, v). The coordinates are given in pixels and the equation of the image plane is $z = f$, where f is the focal distance of the camera expressed in mm.
- The coordinate system of the surface is independent of the two others. The equation defining the surface is described in this coordinate system.

For the perspective projection and camera/image transformation, we need to use the internal parameters of the camera:

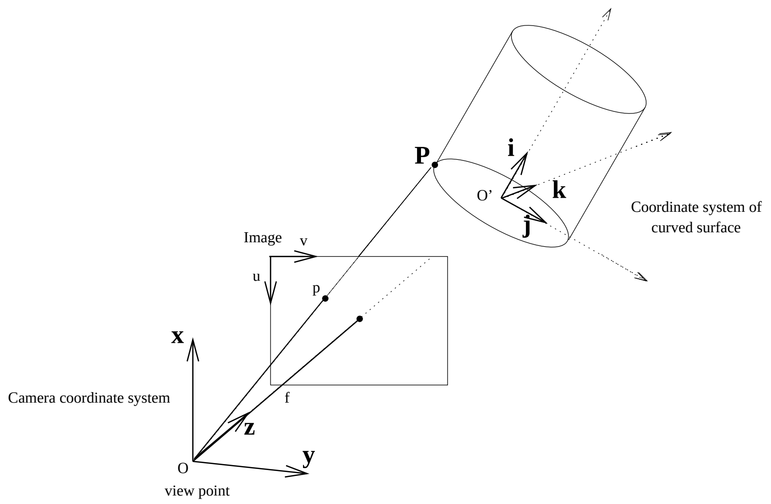


Fig. 1. *Different coordinate systems used in our method.*

- Perspective projection: a point P with the coordinates $(x, y, z)^t$ in the camera coordinate system is transformed in a point $p(x', y', z')^t$ in the image with:

$$\begin{cases} x' = \frac{fx}{z} \\ y' = \frac{fy}{z} \\ z' = f. \end{cases} \quad (1)$$

- Camera/image transformation changes the camera coordinates $p(x', y', z')^t$ to the image coordinates $p(u, v)$. In the image coordinate system, the coordinates of this point are expressed in pixels. This transformation is defined by:

$$\begin{cases} u = -k_u x' + u_0 \\ v = k_v y' + v_0, \end{cases} \quad (2)$$

where (u_0, v_0) are the coordinates of the intersection of the focal axis with the image plane corresponding to the image center, and k_u, k_v are the vertical and the horizontal scale factors given in $pixels/mm$. By grouping the equations (1) and (2) we obtain:

$$\begin{cases} u = -k_u f \frac{x}{z} + u_0 \\ v = k_v f \frac{y}{z} + v_0. \end{cases} \quad (3)$$

The relationships between the coordinate systems of the surface and of the camera are based on external parameters. These relationships are composed of three translations and three rotations. If $P(X,Y,Z)^t$ is a point of the surface, it will be transformed in the coordinate system of the camera in $(x,y,z)^t$ according to the relation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t}, \tag{4}$$

where $\mathbf{t} = (T_x, T_y, T_z)^t$ represents the translation vector between O and O', and $\mathbf{R}$ is a 3×3 rotation matrix to localize the coordinate system of the surface in the camera coordinate system:

$$\mathbf{R} = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x = \begin{pmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{pmatrix} \tag{5}$$

In our approach, in Subsection 3.3 we show how to determine the three rotation angles: θ_x, θ_y and θ_z.

2.2 Cylindrical surfaces

The interpretation of the results will be done in Subsection 3.3 on a cylindrical surface. In this section we give some definitions about Generalised Cylinders (GC).

A generalized cylinder is composed of a planar cross section shifted along a rotation axis. Shape and size could change along this axis. Using the definitions of [12], a GC is Straight when its axis is rectilinear, Homogeneous when the transformation of the cross section is only scaled and Uniform if shape and size throughout the axis are constant. The cross-sections of a CG is Closed is the cross-sections are closed curves (circles, ellipses) or Opened is they are opened curves (arcs, parabolas).

We introduce a coordinate system $(O', \vec{i}, \vec{j}, \vec{k})$ as illustrated in Figure 1. The axis is denoted by $\vec{i}$ and the plane $(\vec{j}, \vec{k})$ contains the cross sections. If P denotes a point on the Straight Homogeneous Generalized Cylinder (SHGC), we have:

$$\vec{O'P}(h,\theta) = h\vec{i} + \rho(\theta) r(h) (\cos\theta \vec{j} + \sin\theta \vec{k}), \tag{6}$$

where $(h, \theta) \in [a, b] \times [0, 2\pi]$, $\rho(\theta)$ and $r(h)$ are functions which identify, respectively, the reference cross-section and the scaling sweeping rule of the SHGC. Curves with θ constant drawn on a SHGC are called meridians, and curves with h constant are called cross-sections [2]. The normal $\vec{N}$ at point P is the cross-product between the two partial derivates $\frac{\partial}{\partial \theta}\vec{O'P}$ and $\frac{\partial}{\partial h}\vec{O'P}$. Forming this cross product, we derive the normal $\vec{N}$:

$$
\begin{aligned}
\vec{N} &= \tfrac{\partial}{\partial \theta}\vec{O'P} \times \tfrac{\partial}{\partial h}\vec{O'P} \\
&= \left[(\rho' \cos\theta - \rho \sin\theta) r\vec{j} + (\rho' \sin\theta + \rho\cos\theta) r\vec{k}\right] \times \left[\vec{i} + \rho r'(\cos\theta \vec{j} + \sin\theta \vec{k})\right] \quad (7)\\
&= -\rho^2 r r' \vec{i} + r(\rho\cos\theta + \rho' \sin\theta)\vec{j} + r(\rho\sin\theta - \rho'\cos\theta)\vec{k},
\end{aligned}
$$

where $r = r(h)$, $r' = r'(h)$, $\rho = rho(\theta)$ and $\rho' = rho'(\theta)$.

3 Searching for the location

In this part, we show how to localize a SUGC in the camera coordinate system. In Section 3.1, we find the projection of the axis in the image plane. In Section 3.2, we explain how to find the cross-section which is projected in the image plane as a straight line, thus defining a second axis, in order to obtain the three rotation angles (5). Section 3.3 provides an interpretation of these two axes in the case of a SUGC with Circular cross-sections.

3.1 Detection of the revolution axis

We want to find the projection of the SUGC axis in the image plane. A lot of work has already been done on this topic by using mathematical morphology [1], finding local symmetries [7] or by using a method based on expectation-maximization [5].

Our approach is based on the analysis of the shape of curves resulting from the projection of cross-sections existing on the cylindrical surface. Let us suppose that two curves C_1, C_2 in the image plane correspond to the projection of two cross-sections. Thus, in this approach we identify first the common normal P_1P_2 of the two curves C_1 and C_2, as shown in Figure 2. As described in [8] we use an iterative method. In the initialization phase, we select a point M_1 on the curve C_1. M_2 is defined as the intersection of the curve C_2 and the normal line to C_2 passing through M_1. Afterwards, we determine a new point

[2] these notations will be used frequently in subsequent Sections

M'_1 on the curve C_1 where the normal line to C_1 passes through M_2. Next, we iterate the method successively for the curves C_1 and C_2. This method stops when two successive points on the same curve are very close to each other. The sensitivity depends of the detection method and of the approximation algorithm of the two curves [8].

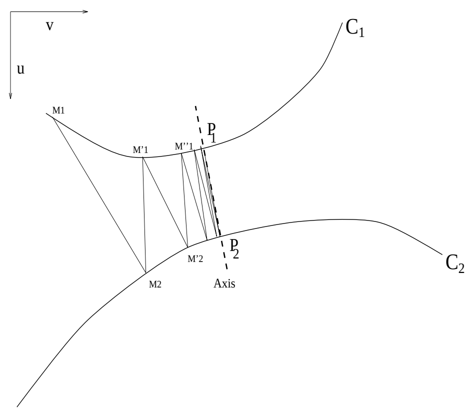


Fig. 2. *Common normal for a curved surface.*

The slope of the straight line P_1 P_2 gives us the direction of the SUGC axis. In the image coordinate system the equation of this axis is:

$$v = A_1.u + B_1, \tag{8}$$

where A_1 and B_1 are the coefficients of the straight line determining this axis.

3.2 Finding the equation of the second axis

Among all the cross-sections of the cylindrical surface, only one is projected on the image plane as a straight line. Only the plane of this cross-section contains the view point. The intersection of this cross-section plane with the image plane defines the second axis. To find this second axis we show how to obtain a point P_0 belonging to the SUGC axis[3] and a curve for which the curvature is equal to zero, Figure 3 (a). We define the curvature of P_0, P_1 and P_2 as:

$$K_i = \lim_{P \to P_i} \frac{\alpha(P) - \alpha(P_i)}{|\widehat{PP_i}|}, \tag{9}$$

where $i \in \{0, \ldots, 2\}$, $\alpha(P_i)$ is the angle of the tangent to P_i, and $|\widehat{PP_i}|$ is the length of the arc between P and P_i. For the curves between C_1 and C_2, when

[3] We will also note the SUCG axis, the revolution axis.

K_1 and K_2 have different signs, we can assume[4] that the curvature changes linearly throughout the axis P_1P_2. For each curve, the direction of the normal line for the intersection point with the revolution axis is the same with the common normal as defined in Subsection 3.1. We define P_0 as the intersection between the revolution axis and the curve for which the curvature is equal to zero. Then, this curve is the straight line defining the second axis, and these two axes are orthogonal. By including such information in the equation (8), we obtain the equation of the second axis [10]:

$$v = \frac{-u}{A_1} + (v_0 + \frac{u_0}{A_1}). \tag{10}$$

3.3 Evaluation of the rotation angles and interpretation of the results

In this section, we evaluate the rotation angles of the camera. In order to localize the object in the camera coordinate system we want to match the two vectors $\vec{x}$ and $\vec{i}$, shown in Figure 1. In order to obtain this matching we perform the rotation by the angle θ_z corresponding to the angle between u and the revolution axis, shown in Figure 3 (a):

$$\theta_z = \arctan(A_1), \tag{11}$$

where A_1 is the coefficient of the straight line defined in (8). Let us denote by (Δ_u, Δ_v) the vector distance between the image center (u_0, v_0) and P_0, the intersection of the two axes. By assuming $k_v = k_u = k$, we obtain:

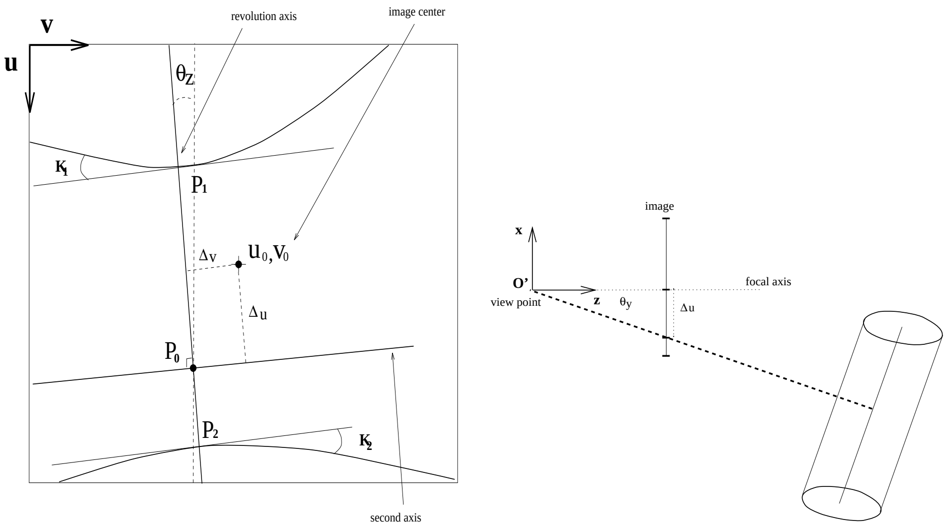
$$\begin{cases} \theta_x &= \arctan(\frac{\Delta_v}{f.k}) \\ \theta_y &= \arctan(\frac{\Delta_u}{f.k}), \end{cases} \tag{12}$$

as shown in Figure 3 (b). In order to match the intersection of the two axes with the image center we perform a rotation of the camera coordinates by the angles θ_x, θ_y and θ_z.

In the case of a SUGC with circular cross-sections, noted cylinder, after these three rotations the equation (6) becomes:

$$\vec{O'P}(h, \theta) = h\vec{i} + R(\cos\theta\vec{j} + \sin\theta\vec{k}), \tag{13}$$

[4] Because of the space limit we do not include the proof.



(a) (b)

Fig. 3. *(a) The revolution and the second axes detected from two projected cross-sections, (b) Rotation θ_y deducted from Δ_u.*

as shown in Figure 4, with $(h, \theta) \in [a, b] \times [0, 2\pi]$ and R fixed. We derive the normal $\vec{N}$ and the tangent $\vec{t}$ to the surface:

$$\begin{aligned} \vec{N} &= (R\cos\theta)\vec{j} + (R\sin\theta)\vec{k}, \\ \vec{t} &= (-R\sin\theta)\vec{j} + (Rcos\theta)\vec{k}. \end{aligned} \tag{14}$$

We shall prove now that, in the particular case of a cylinder, the projection of the revolution axis in the image plane after the rotations corresponds to the set of points belonging to the nearest meridian from the view point.

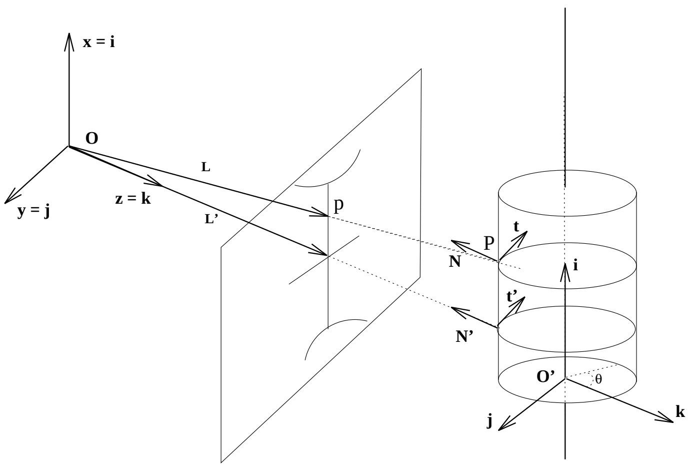


Fig. 4. *Localization of a cylinder in the coordinate system of the camera.*

We need to compute the length $\vec{L}$, i.e., the vector between the view point O and a point $p(x_p, 0, f)^t$ belonging to the revolution axis projection in the new image plane, as can be seen in Figure 4:

$$\vec{L} = x_p \vec{i} + f \vec{k}. \tag{15}$$

Such a vector $\vec{L}$ is projected on a vector $\vec{L'}$ on the plane $(\vec{j}, \vec{k})$, as shown on Figure 4. The minimal distance will be realized when $\vec{L'}$ and $\vec{t}$ are orthogonal. The scalar product is given by:

$$\vec{t}.\vec{L'} = \begin{vmatrix} 0 \\ -R\sin\theta \\ R\cos\theta \end{vmatrix} \begin{vmatrix} 0 \\ 0 \\ f \end{vmatrix} = fR\cos\theta. \tag{16}$$

To obtain $\vec{t}.\vec{L'} = 0$, we have two possible solutions: $\theta = \frac{\pi}{2}$ or $\theta = \frac{3\pi}{2}$. The only possible solution is $\theta = \frac{3\pi}{2}$, and all the points belong to the nearest meridian of the cylinder from the view point. In fact, with $\vec{OO'} = T_x\vec{i} + T_z\vec{k}$, the vector between a point P of the cylinder and O will be $(h+T_x)\vec{i} + R\cos\theta\vec{j} + (R\sin\theta + T_z)\vec{k}$. So, let us denote by D_{OP} this distance:

$$D_{OP} = \sqrt{(h+T_x)^2 + R^2 + T_z^2 + 2RT_z\sin\theta}. \tag{17}$$

D_{OP} is minimal for $\theta = \frac{3\pi}{2}$ and equals to:

$$D_{OP} = \sqrt{(h+T_x)^2 + (T_z - R)^2}. \tag{18}$$

Concerning the position of the second axis, it corresponds to the points belonging to a same cross-section, as described in Section 3.2. The second axis belongs to the cross-section plane passing through the view point. After three rotations, this projected cross-section belongs to the plane $(\vec{j}, \vec{k})$. The intersection of the two axes in the picture gives us the projection of a single point of the surface. Only this point verifies the following relation:

$$\frac{\vec{L} \times \vec{t}}{|\vec{L} \times \vec{t}|} = \vec{i}. \tag{19}$$

In conclusion, by forming the scalar product of $\vec{L}$ and the tangent $\vec{t}$ to the surface, we obtain $\vec{i}$, the revolution axis vector.

Now let us localize this cylinder in the original camera coordinate system $(O, \vec{x}, \vec{y}, \vec{z})$, shown in Figure 1. So, from the equations (5), we obtain the inverse of the rotation matrix $\mathbf{R}^{-1}$:

$$\mathbf{R}^{-1} = \begin{pmatrix} C\theta_y.C\theta_z & S\theta_z.C\theta_y & -S\theta_y \\ C\theta_z.S\theta_x.S\theta_y - S\theta_z.C\theta_x & S\theta_x.S\theta_y.S\theta_z + C\theta_x.C\theta_z & C\theta_y.S\theta_x \\ C\theta_z.S\theta_y.C\theta_x + S\theta_z.S\theta_x & S\theta_z.S\theta_y.C\theta_x - C\theta_z.S\theta_x & C\theta_y.C\theta_x \end{pmatrix}, \quad (20)$$

with $C\theta_i = \cos\theta_i$ and $S\theta_i = \sin\theta_i$. We obtain then, for each vector of the object coordinate system:

$$\begin{cases} \vec{i} = (C\theta_y.C\theta_z)\vec{x} + (C\theta_z.S\theta_x.S\theta_y - S\theta_z.C\theta_x)\vec{y} + (C\theta_z.S\theta_y.C\theta_x + S\theta_z.S\theta_x)\vec{z} \\ \vec{j} = (S\theta_z.C\theta_y)\vec{x} + (S\theta_x.S\theta_y.S\theta_z + C\theta_x.C\theta_z)\vec{y} + (S\theta_z.S\theta_y.C\theta_x - C\theta_z.S\theta_x)\vec{z} \\ \vec{k} = (-S\theta_y)\vec{x} + (C\theta_y.S\theta_x)\vec{y} + (C\theta_y.C\theta_x)\vec{z} \end{cases} . \quad (21)$$

By substituting the equations (21) in the equation of the cylinder (13) we obtain:

$$\begin{aligned} \vec{O'P}(h,\theta) = {} & [h.(C\theta_y.C\theta_z) + R\cos\theta.(S\theta_z.C\theta_y) - R\sin\theta.(S\theta_y)] \quad \vec{x} \\ + {} & [h.(C\theta_z.S\theta_x.S\theta_y - S\theta_z.C\theta_x) + R\cos\theta.(S\theta_x.S\theta_y.S\theta_z + C\theta_x.C\theta_z) + R\sin\theta.(C\theta_y.S\theta_x)] \quad \vec{y} \quad (22) \\ + {} & [h.(C\theta_z.S\theta_y.C\theta_x + S\theta_z.S\theta_x) + R\cos\theta.(S\theta_z.S\theta_y.C\theta_x - C\theta_z.S\theta_x) + R\sin\theta.(C\theta_y.C\theta_x)] \quad \vec{z} \end{aligned}$$

with $(h, \theta) \in [a, b] \times [0, 2\pi]$ and R fixed. Using the equations (11) and (12), finally we show the localization of a cylindrical surface in the camera coordinate system :

$$\begin{aligned} \vec{O'P}(h,\theta) = {} & \left(\frac{fk(h+R\cos\theta)}{\sqrt{1+A_1^2}} - R\sin\theta\Delta_u\right) \frac{\vec{x}}{\sqrt{f^2k^2+\Delta_u^2}} \\ + {} & \frac{1}{\sqrt{(1+A_1^2)}}\left(\frac{\Delta_u\Delta_v(h+R\cos\theta A_1)}{\sqrt{f^2k^2+\Delta_u^2}} - hA_1fk + R\cos\theta fk\right) + \frac{R\sin\theta fk\Delta_v}{\sqrt{f^2k^2+\Delta_u^2}} \quad \frac{\vec{y}}{\sqrt{f^2k^2+\Delta_v^2}} \quad (23) \\ + {} & \frac{1}{\sqrt{1+A_1^2}}\left(\frac{\Delta_u fk(h+R\cos\theta A_1)}{\sqrt{f^2k^2+\Delta_u^2}} + hA_1\Delta_v + R\cos\theta\Delta_v\right) + \frac{R\sin\theta f^2k^2}{\sqrt{f^2k^2+\Delta_u^2}} \quad \frac{\vec{z}}{\sqrt{f^2k^2+\Delta_v^2}}, \end{aligned}$$

with $(h, \theta) \in [a, b] \times [0, 2\pi]$ and R fixed. By assuming that the parameters of the surface are known, we are able to localize a cylinder in the camera coordinate system, by using A_1, Δ_u, Δ_v and (23).

4 Results

In this section, we illustrate the efficacy of our method on two different examples. The first example consists of a synthetic image in order to validate our method. For this purpose, we map the original image Figure 5 (a) on a cylindrical surface, as shown in Figure 5 (b). After using an edge detection [3], whose output is shown in Figure 5 (c) we select the projections of two cross-sections shown in Figure 5 (d). Figure 5 (e) illustrates the extraction of the two axes in the image. In Figure 5 (f) we show the superposition of Figure 5 (e) with Figure 5 (b).

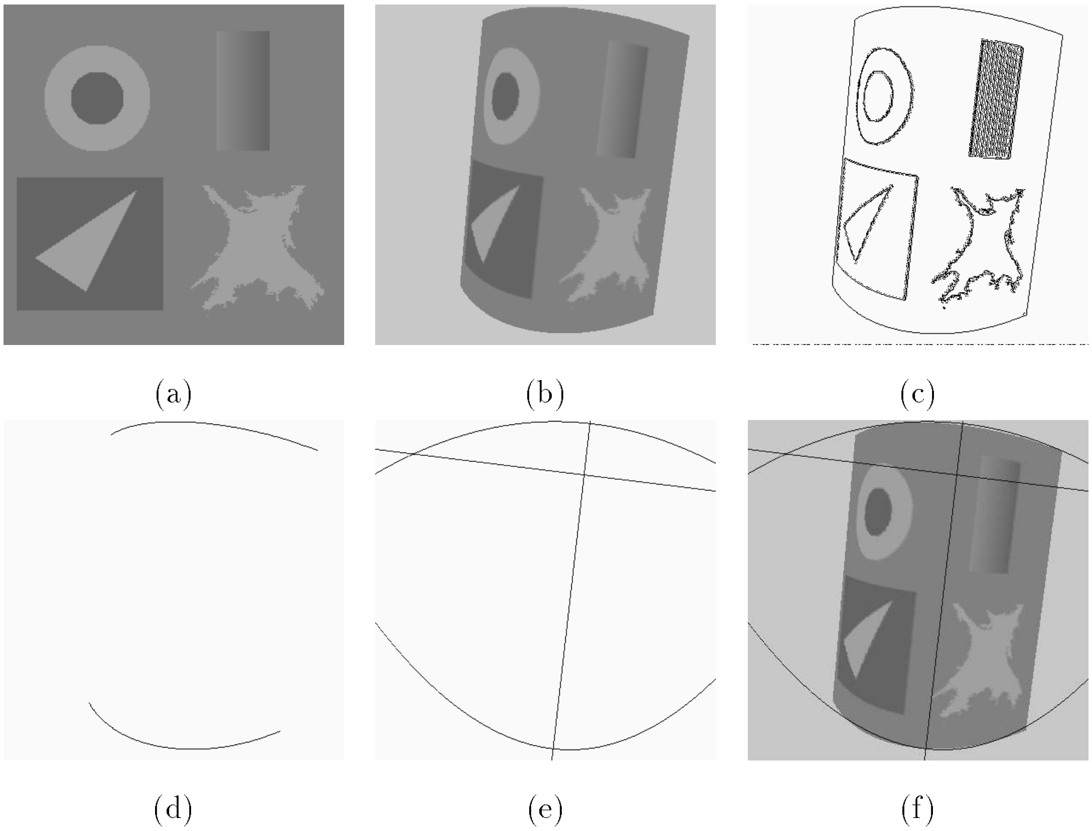


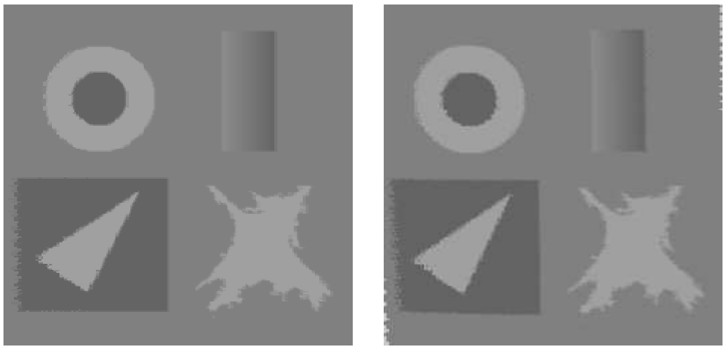
Fig. 5. *(a) A synthetic image, (b) The image mapped on a cylinder, (c) Edge detection, (d) Selection of two projected cross-sections, (e) The two axes, (f) Superposition of (b) and (e).*

In the Table 1, we list the internal parameters of the simulated camera:

Table 1
Internal parameters of the simulated camera.

focal distance = 60 mm	resolution = 10 pixels/mm	$u_0 = 200\ pixels$
	camera-surface distance = 400 mm	$v_0 = 200\ pixels$

Figure 6 (a) corresponds to the flattened image derived from the image Figure 5 (b) by using the real rotation angles. The Figure 6 (b) corresponds to the flattened image after applying our method. A first remark concerns the degradation of the details which disappear during the projection phase. The two images are similar, and the image representation is well restored (circle, triangle).



(a) (b)

Fig. 6. *Flattened pictures (a) With real rotation angles, (b) With the rotation angles found with our method.*

The Table 2 presents a comparison between the real rotation angles and the rotation angles found by our method. The size of the surface is 300 mm, and the error of localization at the surface center (projection of the image center) is described in the Table 2:

Table 2
Comparison of the angle values.

Real rotation angles	Computed rotation angle	Error at the surface center
$\theta_x = -12.00^o$	$\theta_x = -11.77^o$	1.61 mm
$\theta_y = 6.45^o$	$\theta_y = 5.59^o$	6.01 mm
$\theta_z = 5.84^o$	$\theta_z = 6.46^o$	4.33 mm

The equation of the cylinder is: $\vec{O'P}(h,\theta) = h\vec{i} + 90(\cos\theta\vec{j} + \sin\theta\vec{k})$. In the camera coordinate system we localize this cylinder:

$$\begin{aligned} \vec{O'P}(h,\theta) = \; & [0.988925h + 10.077705\cos\theta + 8.766828\sin\theta] \;\; \vec{x} \\ & + [-0.129888h + 87.347063\cos\theta - 18.271209\sin\theta] \; \vec{y} \\ & + \; [0.071806h + 19.207563\cos\theta + 87.688686\sin\theta] \;\; \vec{z}\,, \end{aligned}$$

with $(h,\theta) \in [a,b] \times [0,2\pi]$.

Figure 7 (a) illustrates the application of our method to a mural painting of a Byzantine church. From this image we select the projections of two cross-sections as shown in Figure 8 and we display the localization of the two perpendicular axes. On the Figure 7 (b), we display the image after localization, projection and flattening in order to obtain an image without any distortion due to the curved surface and the projective geometry.

(a)

(b)

Fig. 7. *(a) View of an arch of Acheiropeitos church. (b) Flattened picture after localization.*

Table 3
Internal parameters and computed parameters.

focal distance = 325 mm	camera-surface distance = 6500 mm
resolution = 5 $pixels/mm$	$\theta_x = -1.7^o$
Δ_u = 945 $pixels$	$\theta_y = 30.2^o$
Δ_v = -49 $pixels$	$\theta_z = 2.9^o$

By assuming the following parameters: focal distance, camera-surface distance and resolution, we list the computed parameters in the Table 4.

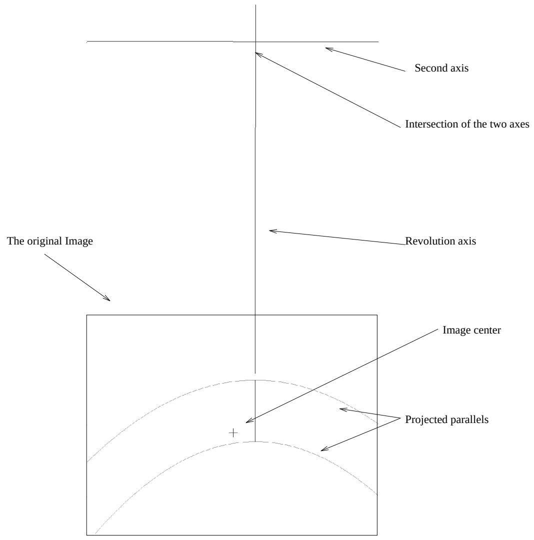


Fig. 8. it The localization of the two axes.

5 Conclusion

By using a priori knowledge about the geometry of surface, we have developed a 3D surface localization method based on a single perspective view. Two assumptions have been made: the scene is mapped on a SUGC, and cross-sections are detected in the cylindrical surface. We have shown how to find the two orthogonal axes in only one view of the curved surface and how to localize it in the camera coordinate system. The method has been illustrated by two different examples. In the first example we compare the image projection provided by our method with the ideal projection. The second one shows a direct application of this study. This work will be applied to improve the analysis of mural paintings on columns or vaults taken under different views [2], or to perform mosaicing in order to obtain more details [6].

References

[1] M. Brady, *Criteria for Representations of Shape* (Human and Machine Vision, A. Rosenfeld and J. Beck, New York, 1983).

[2] J.-M. Chassery, Image Processing And Analysis. A Challenge For Art Paintings, *In Proc. Art and Technology Workshop* (Athens, Greece, 1993) 366–377.

[3] J.P. Cocquerez and S. Philipp, *Analyses d'images: filtrage et segmentation* (Masson, Paris, 1995).

[4] M. Dhome, M. Richetin, J.-T. Lapresté and G. Rives, Determination of the Attitude of 3D Objects from a Single Perspective View, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 11(12), (1989) 1265–1278.

[5] R. Glachet, M. Dhome and J.-T. Lapresté, Finding the Perspective Projection of an Axis of Revolution, *Pattern Recognition Letters* 12(11), (1989) 693–700.

[6] P. Jaillon and A. Montanvert, Image Mosaicking Applied On 3D Surfaces, *In Proc. 12th International Conference on Pattern Recognition* (Jerusalem, Israel, 1994) I, 253–257.

[7] J. Ponce, D. Chelberg and W. Mann, Invariant Properties of Straight Homogeneous Generalized Cylinders and their Contours, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 11(9), (1989) 951–966.

[8] W. Puech, P. Schott and J.-M. Chassery, Discrete Method of Calculus Using Common Normals for Curved Surface Reconstruction, *In Proc. 5th Discrete Geometry for Computer Imagery* (Clermont-Ferrand, France, 1995) 83–92.

[9] W. Puech and J.-M. Chassery, Curved Surface Reconstruction Using Monocular Vision, *In Proc. 8th European Signal Processing Conference* (Trieste, Italy, 1996) 1, 9–12.

[10] W. Puech and J.-M. Chassery, A New Curved Surface Localization Method Using a Single Perspective View, *In Proc. 10th Scandinavian Conference on Image Analysis*, (Lappeenranta, Finland, 1997) II, 969–976,.

[11] M. Richetin, M. Dhome, J.-T. Lapresté and G. Rives, Inverse Perspective Transform Using Zero-Curvature Contour Points: Application to the Localization of Some Generalized Cylinders from a Single view, *IEEE Trans. on Pattern Analysis and Machine Intelligence* , 13(2), (1991) 185–192.

[12] S.A. Shafer, *Shadows and Silhouettes in Computer Vision* (Kluver, Boston, 1985).

[13] H. Tanahashi, K. Sakaue and K. Yamamoto, Recovering Decorative Patterns of Ceramics Objects from Monocular Image Using a Genetic Algorithm, *In Proc. 3rd International Conference on Document Analysis and Recognition*, (Montréal, Canada, 1995) I, 339–342.

[14] O. Wink, A.W.M. Smeulders and D.C. Koelma, D.C., Location Estimation of Cylinders From a 2-D Image, *In Proc. 12th International Conference on Pattern Recognition* (Jerusalem, Israel, 1994) I, 682–684.

[15] Y.C. You, J.D. Lee, J.Y. Lee and C.H. Chen, Determining Location and Orientation of a Labelled Cylinder Using Point-Pair Estimation Algorithm, *In Proc. 11th International Conference on Pattern Recognition* (The Hague, The Netherlands, 1992) I, 354–357.