

Multiscale Edge Detection in Range Images via Normal Changes ¹

Chwen-Jye Sze[†], Hong-Yuan Mark Liao^{††2},
Hai-Lung Hung[‡], Kuo-Chin Fan[†], Shing-Jong Lin[†]

† Institute of Computer Science and Information Engineering,
National Central University, Chung-Li, Taiwan

†† Institute of Information Science, Academia Sinica, Taiwan

‡ Department of Electrical Engineering and Computer Science,
Northwestern University, IL 60208

Corresponding author's address —

Hong-Yuan Mark Liao
Associate Research Fellow
Institute of Information Science, Academia Sinica
Nankang, Taipei, Taiwan
E-mail : liao@iis.sinica.edu.tw
TEL : 886-2-788-3799 ext. 1811
FAX : 886-2-782-4814

¹This work is supported by the National Science Council of Taiwan under grant no. NSC86-2745-E-001-004.

²To whom correspondence should be sent.

ABSTRACT

A new edge detection technique based on detecting normal changes is proposed. Most of the existing range-image-based edge detection algorithms base their detection criterion on depth or curvature changes. However, depth or curvature changes do not have keen sensitivity on detecting roof (or crease) edges. Using normal changes as a detecting criterion, on the contrary, can easily detect the existence of a roof edge even the change across a boundary is slight. Experimental results using both synthetic and real images demonstrate that the proposed method is indeed superb in detecting roof (or crease) edges.

Keywords: range edge detection, wavelet

Running head: Multiscale Edge Detection via Normal chnages

1. Introduction

Extracting edges from an image is an important early vision process. It has been of interest to research in the area of computer vision from the outset[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. The algorithms developed for edge detection can be classified into two kinds depending on the type of acquired images. For an intensity image, the devised algorithms usually aim to detect step edges. This is due to the natural limitation of this type of images. For a range image, since the depth information is available, it is possible to correctly detect both step edges and roof edges (or crease edges). In computer vision, since closed contours are more useful for higher level image analysis and using range images is easier to achieve this goal, we shall focus on range-image-based edge detection in this paper.

There are a number of good papers in range-image-based edge detection[4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Chung *et al.*[11] proposed to use the first and second derivatives of Gaussian kernels to detect jump edges and roof edges. Ghosal and Mehrotra[4, 13] proposed a moment-based approach for detecting generalized step edges and pulse/staircase edges. In [9, 10], the local polynomial approximation technique is adopted to estimate mean curvature and Gaussian curvature. Then, the existence and strength of edges can be decided by checking the sign changes of K-H map. Recently, Monga and Deriche[5] proposed a 3D Deriche filter to directly compute curvatures without performing local surface fitting. The edge points are the local extrema along the maximum curvatures direction. In [12], Wani and Batchelor used some rules to judge the type of an edge. Their approach can distinguish fold edge, semistep edge, and boundary edge.

In the above mentioned range-image-based edge detection algorithms [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], only those points that have significant depth or curvature changes with respect to their neighbors (e.g., step edges) are identified as edge points. However, for those points which have little depth or curvature variations with respect to their neighbors (e.g., roof

edges) are hardly detected. In this paper, we propose a new edge detection technique based on the detection of normal changes. The normal value is an important characteristic in differential geometry[10]. We find that by detecting significant normal changes, both step edges and roof edges can be easily identified. The whole detection procedure is divided into two stages. In the first stage, the normal of every point in a range image is decided. Since all data points in a range image are discrete, the partial derivatives which are required for deriving the normal value cannot be directly computed. For comparison purpose, we propose to use quadratic surface fitting[10, 15], orthogonal wavelet-based approach[1, 18], and non-orthogonal wavelet-based approach[2], respectively, to approximate the original object surfaces and then calculate the normal value of every discrete point on the surfaces. After the normal values of all surface points are determined, the non-orthogonal wavelet transforms (dyadic wavelet transform) proposed by Mallat *et al.*[2, 23] is applied to detect those points which have significant normal changes as edge points. From the experimental results, we find that the non-orthogonal wavelet-based approach can best approximate the original surfaces from a discrete data set. Further, we also find that edge detection based on normal change is a more promising alternative than other methods that based their detection criterion on depth or curvature changes. The proposed edge detector can detect both step edges and roof edges without introducing any edge models or heuristics.

The rest of this paper is organized as follows. In the first part of the next section, we overview some geometric properties of a 3D surface in differential geometry point of view. Then, we shall explain why normal change is good for detecting edges. In Section 3, three possible methods are proposed to calculate normal values from a set of discrete surface points. Section 4 describes how to detect edges based on normal changes. Experimental results using both synthetic range images and real range images are reported in Section 5. Section 6 includes some discussion issues and a concluding remark.

2. Range edge detection via normal change

In this section, we shall explain why normal change can be used as a cue for range edge detection. Some properties of a 3D surface in differential geometry viewpoint which are useful for edge detection will be addressed in Section 2.1. Then, a detailed explanation on why normal change is a better choice for edge detection will be described in Section 2.2.

2.1. Some properties in differential geometry useful for edge detection

In this section, some basic properties of a 3D surface will be addressed from the differential geometry viewpoint. These properties are useful for solving the edge detection problem on range images.

Let \mathcal{S} be a differentiable surface and $\vec{p}(u, v)$ be a point on \mathcal{S} with coordinate (u, v) . If \vec{p}_u and \vec{p}_v are the partial derivatives of $\vec{p}(u, v)$ with respect to u and v , respectively, then we can say \vec{p}_u, \vec{p}_v form the basis of a tangent plane, $T(p)$, of $\vec{p}(u, v)$ (Figure 1). The normal of $T(p)$ can be defined as

$$\vec{N}(u, v) = \frac{\vec{p}_u \times \vec{p}_v}{\|\vec{p}_u \times \vec{p}_v\|}. \quad (1)$$

Here, the norm of any $\vec{N}(u, v)$ is always equal to 1 and all $\vec{N}(u, v)$'s lie on a unit sphere in R^3 . The mapping, $\vec{N} : \mathcal{S} \rightarrow R^3$, is called the Gauss mapping, $G(R^3)$ [14].

Let $\vec{N}(u, v)$ be differentiable, the mapping, $d\vec{N}(u, v)$, is from $G(R^3)$ to a tangent plane, $G(R^3)$, at $\vec{N}(u, v)$. Since the tangent plane of $\vec{N}(u, v)$ is equal to that of $\vec{p}(u, v)$, $d\vec{N}(u, v)$ is also on the $T(p)$ plane, as shown in Figure 1. Thus, both $d\vec{p}$ and $d\vec{N}$ can be represented by the linear combination of du and dv as follows [10]:

$$d\vec{p} = \vec{p}_u du + \vec{p}_v dv, \quad (2)$$

and

$$d\vec{N} = \vec{N}_u du + \vec{N}_v dv. \quad (3)$$

Here, $p(u, v)$ is the gray level (intensity image) or depth (range image) at position (u, v) . Therefore, $d\vec{p}$ physically means the intensity or depth change with respect to $\vec{p}(u, v)$'s neighbors. As to \vec{N}_u and \vec{N}_v , they are mathematically defined as $\vec{N}_u = \frac{\partial \vec{N}}{\partial u}$ and $\vec{N}_v = \frac{\partial \vec{N}}{\partial v}$. Namely, they physically mean the normal change along u and v directions, respectively.

2.2. Why normal change is better for edge detection

Equations(2) and (3) represent, respectively, the depth (or intensity) change and the normal change with respect to the neighbors of $p(u, v)$. The drawback of using Equation(2) is that the depth (or intensity) change can only be detected when its value is significant. For example, Figure 2(a) shows an object sitting on the $x-y$ plane, its corresponding range image (in intensity change format) is shown in Figure 2(c). If the same object is viewed from another point as shown in Figure 2(b), then its base plane becomes $u-v$ plane. From this view, the object's corresponding range (or intensity) change is shown in Figure 2(d). The edge type shown in Figure 2(c) is the so-called step edge and the one shown in Figure 2(d) is known as a roof edge. Usually, people applies the first derivative in one direction to locate edge positions. From Figure 2, it is obvious that the derivatives applied to the x direction (Figure 2(a)) and the u direction (Figure 2(b)) both obtain zero values. The derivatives applied to the y direction (Figure 2(a)) and the v direction (Figure 2(b)) will have the responses shown in Figure 3(c) and Figure 3(d), respectively. From these outcome, it is apparent that the step edge can be easily detected. However, the roof edge which is actually existent is not detected. This phenomenon reflects the fact that simply using $d\vec{p}$, *i.e.*, the intensity or depth change of $p(u, v)$ with respect to its neighbors is not sufficient to detect all physically existent edges.

Since $d\vec{p}$ in Equation (2) cannot be applied to detect different types of edges, we propose to use $d\vec{N}$ (Equation (3)) instead. In what follows, we shall explain why $d\vec{N}$ is chosen to locate general edges. In Figure 4(a)-(d), four different types of crease(roof) edges, including convex and concave edges, are shown. Figure 4(e)-(h), respectively, show their corresponding normal changes across edges. These difference vectors are all normalized and fit into Gaussian spheres. From the normal changes we find that their values are much more significant than those of the depth (or intensity) changes. Considering a more general case shown in Figure 4(i), an edge occurs between a plane and a curve surface. It is clear that the normal change between p_2 and p_3 (two points located at different surfaces) is much larger than the normal change between p_4 and p_5 (two points belonging to the same surface) (Figure 4(j) and (k)). Therefore, we conclude that the use of the normal change - $d\vec{N}$, instead of the depth (or intensity) change - $d\vec{p}$, should be a good solution for finding a more powerful edge detector.

3. Calculating normals from discrete surface points

In the previous section, we have discussed that the change of normals at every point on a surface (Equation (3)) can be used to detect the position of edges. The continuous domain normal value derivation process is summarized as follows. Let $\vec{p}(u, v) = (u, v, f(u, v))$ be a point located on a surface. The partial derivatives of $\vec{p}(u, v)$ in u and v directions are :

$$\vec{p}_u = \frac{\partial \vec{p}(u, v)}{\partial u} = (1 \ 0 \ f_u(u, v)), \quad (4)$$

and

$$\vec{p}_v = \frac{\partial \vec{p}(u, v)}{\partial v} = (0 \ 1 \ f_v(u, v)). \quad (5)$$

Plugging these two values into Equation (1), the value of $\vec{N}(u, v)$ can be rewritten as

$$\vec{N}(u, v) = \left(\frac{-f_u}{\sqrt{1+f_u^2+f_v^2}} \frac{-f_v}{\sqrt{1+f_u^2+f_v^2}} \frac{1}{\sqrt{1+f_u^2+f_v^2}} \right) \quad (6)$$

$$= (n_1(u, v) \ n_2(u, v) \ n_3(u, v)), \quad (7)$$

where $f_u = \frac{\partial f(u, v)}{\partial u}$ and $f_v = \frac{\partial f(u, v)}{\partial v}$, $n_1(u, v) = \frac{-f_u}{\sqrt{1+f_u^2+f_v^2}}$, $n_2(u, v) = \frac{-f_v}{\sqrt{1+f_u^2+f_v^2}}$ and $n_3(u, v) = \frac{1}{\sqrt{1+f_u^2+f_v^2}}$. The ranges of n_1 , n_2 and n_3 are all bounded by (-1,1). In real implementation, since all data points acquired in a range image are discrete by nature, the above calculation does not apply. Therefore, we have to seek some appropriate methods to deal with this problem. In the implementation stage, we split the edge detection procedure into two steps. In the first step, the normal at every point on a surface should be decided. This step involves the calculations of partial derivatives on a set of discrete data points. Then, in the second step, a detector is required to accurately detect the points where significant variation of normals are encountered. In order to calculate the normals on a set of discrete surface points, some existing methods [2, 10, 15, 16, 17, 18, 19] can be applied. For comparison purpose, we choose quadratic surface fitting[10, 15], orthogonal wavelet-based approach[20], and non-orthogonal wavelet-based approach[2] to calculate the normal value of every point on a surface. In what follows, we shall briefly describe how to use these methods to compute normals based on a set of discrete surface points.

A. Quadratic surface fitting

This method was proposed by Besl and Jain[10, 15]. In their method, the first and second partial derivatives at a given location (u, v) are estimated by overlapping $N \times N$ windows centered at pixel (u, v) and fitting local quadratic surfaces. The points in the $N \times N$ window are associated with a position $(u, v) \in U \times U$, where N is odd and $U = -(N-1)/2, \dots, -1, 0, 1, \dots, (N-1)/2$. The quadratic surface fit is obtained by

using a set of discrete orthogonal polynomials

$$\phi_0(u) = 1; \quad \phi_1(u) = u; \quad \phi_2(u) = (u^2 - M(M+1)/3), \quad (8)$$

where $M = (N - 1)/2$. The following $b_i(u)$ functions are normalized forms corresponding to the ϕ_i polynomials and are defined as

$$b_0(u) = \frac{1}{N}; \quad b_1(u) = \frac{3}{M(M+1)(2M+1)}u; \quad \text{and} \quad b_2(u) = \frac{1}{P(M)} \left(u^2 - \frac{M(M+1)}{3} \right), \quad (9)$$

where $P(M) = 8/45M^5 + 4/9M^4 + 2/9M^3 - 1/9M^2 - 1/15M$. The normalized $b_i(u)$ and $\phi_i(u)$ must satisfy the orthogonality relationship

$$\sum_{u \in U} \phi_i(u) b_j(u) = \delta_{ij}, \quad (10)$$

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. A surface function estimate $g(u, v)$ is obtained by

$$g(u, v) = \sum_{i+j \leq 2} a_{ij} \phi_i(u) \phi_j(v), \quad (11)$$

that minimizes the total square error term

$$\epsilon^2 = \sum_{(u,v) \in U^2} (f(u, v) - g(u, v))^2, \quad (12)$$

where $f(u, v)$ is the original surface function. The solution for the unknown coefficients can be obtained by

$$a_{ij} = \sum_{(u,v) \in U^2} f(u, v) b_i(u) b_j(v). \quad (13)$$

Having the surface function estimate $g(u, v)$, the first partial derivatives which are required to derive the normal at position (u, v) can thus be determined by

$$\frac{\partial g}{\partial u} = a_{10} \quad \text{and} \quad \frac{\partial g}{\partial v} = a_{01}. \quad (14)$$

B. Orthogonal wavelet-based approach[20]

This subsection will describe how to use an orthogonal wavelet basis to approximate a surface and then based on the approximated surface to calculate the normal of every surface point. The critical issues of using this approach include: (a) how to use an orthogonal wavelet basis to describe a surface, (b) how to deal with the differentiation problem which is a must in deriving normal values. Part of the subsequent derivations can be found in our previous paper[20].

The process of wavelet transform represents a continuous function, $f(x)$, with a limited number of successive approximations, each of which is a smoothed version of $f(x)$ [16]. In this paper, we will employ the Daubechies scaling function[17, 18] to represent continuous functions. Denote the Daubechies scaling function by $\phi(x)$ and its dilation and translation functions $2^{j/2}\phi(2^jx - n)$ by $\phi_{j,n}(x)$ for $j, n \in Z$. Let V_j be the function space spanned by $\{\phi_{j,n}(x)\}_{n \in Z}$. In fact, $\{\phi_{j,n}(x)\}_{n \in Z}$ is an orthonormal basis of V_j . The function spaces V_j , $j \in Z$ have the following properties:

- (1) $V_j \subseteq V_{j+1}$ for all $j \in Z$ and
- (2) $\bigcup_{j=-\infty}^{\infty} V_j = L^2(R)$.

Let $S_j f(x)$ be the projection of a continuous function, $f(x)$ on V_j , i.e.

$$S_j f = \sum_n c_{j,n} \phi_{j,n}(x), n \in Z,$$

where $c_{j,n} = \int f(x) \phi_{j,n}(x) dx$. By property (2), $\{S_j f(x)\}_{j \in Z}$ is an approximation scheme of $f(x)$ in which $S_{j+1} f(x)$ is a better approximation than $S_j f(x)$ for all $j \in Z$.

The difference between two successive approximations, $S_j f(x)$ and $S_{j+1} f(x)$, can be expanded by another set of orthonormal basis $\psi_{j,n}$ which is generated by dilation and translation from another prototype function $\psi(x)$, called the Daubechies wavelet function. Let $W_j f(x) = S_{j+1} f(x) - S_j f(x)$. We have

$$W_j f(x) = \sum_n d_{j,n} \psi_{j,n}(x), n \in Z,$$

where $\psi_{j,n}(x) = 2^{j/2}\psi(2^jx - n)$ and $d_{j,n} = \int f(x)\psi_{j,n}dx$. Therefore, any continuous function $f(x) \in L^2(R)$ can be approximated by

$$f(x) = S_j f(x) + \sum_{k=j}^{\infty} \sum_{n \in Z} d_{k,n} \psi_{k,n}(x). \quad (15)$$

Since the derivation of normals requires the computation of the first partial derivatives on surface points, the relationship between the original wavelet basis and its differentiated form should be built. In what follows, we shall describe how to make this link. Let the original signal (or surface) $f(x)$ be represented as[19]

$$f(x) = \sum_n c_n \phi_n(x), \quad (16)$$

where $c_n = \int f(x)\phi(x - n)dx$. Since $\int \phi(x - n)dx = 1$ and $\phi_n(x)$ is compactly supported, we can think of $\phi(x - n)$ as a function which is similar to a delta function. Therefore, we have

$$f(n) \approx \int f(x)\phi(x - n)dx. \quad (17)$$

Substituting Equation (17) into Equation (16) and differentiating it, we have

$$f'(x) = \sum_n f(n)\phi'_n(x). \quad (18)$$

If we expand $\phi'_n(x)$ based on the scaling function and wavelet function, then the following equation is obtained,

$$\phi'_n(x) = \sum_k \Gamma_{n,k} \phi_k(x) + \sum_{j \geq 0, k} \Gamma_{n,k}^j \psi_{j,k}(x), \quad (19)$$

where

$$\Gamma_{n,k} = \int \phi'(x - n)\phi(x - k)dx, \quad (20)$$

and

$$\Gamma_{n,k}^j = \int \phi'(x - n)\psi_j(x - k)dx. \quad (21)$$

The above formulation is the so-called wavelet-Galerkin method[19]. $\Gamma_{n,k}$ and $\Gamma_{n,k}^j$ are called connection coefficients[16]. These coefficients can be explicitly calculated for specific families of wavelets. If we substitute Equation (19) into Equation (18), we have

$$f'(x) = \sum_{n,k} f(n)\Gamma_{n,k}\phi_k(x) + \sum_{j \geq 0, n,k} f(n)\Gamma_{n,k}^j\psi_{j,k}(x). \quad (22)$$

By multiplying both sides of Equation (22) with $\phi_k(x)$ and integrating, we find that

$$f'(x) \approx \int f'(x)\phi(x-k)dx = \sum_n f(n)\Gamma_{n,k} = \sum_n \Gamma_{n,0}f(k+n).$$

If the scaling function $\phi(x)$ has N vanishing moments, the above equation can be further simplified as[19]

$$f'(x) \approx \sum_{n=-2N+2}^{2N-2} \Gamma_{n,0}f(k+n). \quad (23)$$

The above discussions are all in one variable case. In what follows, we shall extend the study to a two variable case. Let a function $f(x, y) \in L^2(R)$ be represented as

$$f(x, y) = \sum_m \sum_n c_{m,n}\phi(x-m, y-n), \quad (24)$$

where $c_{m,n}$ are the weighting coefficients, and $\phi(x-m, y-n)$'s are the wavelet basis of a certain subspace at the fine resolution 0. According to the tensor product method[2, 17, 18, 21], we can set $\phi(x, y) = \phi(x)\phi(y)$. Therefore, the 2-D connection coefficients of Equation(24) can be converted into 1-D form. More details about the derivation can be found in [20]. Finally, the partial derivatives in the x and y directions at point (i, j) can be respectively simplified as follows

$$\frac{\partial f(x, y)}{\partial x} \Big|_{(x,y)=(i,j)} = f_x(i, j) \approx \sum_{m=-2N+2}^{2N-2} \Gamma_{m,0}f(i+m, j), \quad (25)$$

and

$$\frac{\partial f(x, y)}{\partial y} \Big|_{(x,y)=(i,j)} = f_y(i, j) \approx \sum_{n=-2N+2}^{2N-2} \Gamma_{n,0}f(i, j+n), \quad (26)$$

where $\Gamma_{k,0}$ are the connection coefficients with $N=k$ vanishing moments. With Equations (25) and (26), we can calculate the normal value of every point on a surface which is approximated by linear combination of orthogonal wavelet bases.

C. Non-orthogonal wavelet-based approach

In this approach, the most important issue is to choose a set of continuous and differentiable “smoothing” functions. A smoothing function $\theta(x)$ [2] is a function whose integral is equal to 1 and converges to 0 at infinity. Let $f(x)$ be a discrete function which is composed of a set of discrete data as follows :

$$f(x) = \begin{cases} \text{sample value} & \text{if } x = \dots, -1, 0, 1, \dots, \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

Based on the above mentioned $f(x)$ and $\theta(x)$, a new continuous and differentiable function $h(x)$ can be defined by convolving $f(x)$ and $\theta(x)$, that is,

$$h(x) = (f * \theta)(x), \quad x \in R. \quad (28)$$

Since $h(x)$ is an approximation of $f(x)$ and is continuous and differentiable, the first partial derivatives can be obtained by direct differentiation on $h(x)$, *i.e.*,

$$\frac{dh(x)}{dx} = \frac{d}{dx}(f * \theta)(x) = \left(\frac{df}{dx} * \theta\right)(x) = \left(f * \frac{d\theta}{dx}\right)(x). \quad (29)$$

If the smoothing function $\theta(x)$ is “compact” enough, then $h(x)$ can approximate closely to $f(x)$ at $x \in \{\dots, -1, 0, 1, \cdot\}$. By the same token, the derivative of $h(x)$ can approximate that of $f(x)$ at every $x \in Z$, *i.e.*,

$$\frac{df(x)}{dx} \approx \frac{dh(x)}{dx} = \left(f * \frac{d\theta}{dx}\right)(x) \Big|_{x \in Z}. \quad (30)$$

4. Detecting edges based on abrupt normal changes

In [2, 22, 23], Mallat and his students have developed some pioneering works for multiscale edge detection based on gray level changes. Here, we shall review part of their

works which will be useful in our work. Define two wavelet functions, $\psi^1(x, y)$ and $\psi^2(x, y)$ [2, 22] where

$$\psi^1(x, y) = \frac{\partial\theta(x, y)}{\partial x}, \quad (31)$$

and

$$\psi^2(x, y) = \frac{\partial\theta(x, y)}{\partial y}. \quad (32)$$

$\theta(x, y)$ is a smoothing function whose integration over the full domain is equal to 1 and converges to 0 at infinity. These two functions have to satisfy the following conditions:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi^1(x, y) dx dy = 0 \quad (33)$$

and

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \psi^2(x, y) dx dy = 0. \quad (34)$$

Let $f(x, y) \in \mathbf{L}^2(\mathbf{R})$. The so-called dyadic wavelet transform [2, 22] of $f(x, y)$ at scale 2^j along x and y directions can be represented, respectively, by

$$W_{2^j}^1 f(x, y) = f * \psi_{2^j}^1(x, y), \quad (35)$$

and

$$W_{2^j}^2 f(x, y) = f * \psi_{2^j}^2(x, y), \quad (36)$$

where $\psi_{2^j}^1(x, y) = \frac{1}{2^{2j}}\psi^1(\frac{x}{2^j}, \frac{y}{2^j})$ and $\psi_{2^j}^2(x, y) = \frac{1}{2^{2j}}\psi^2(\frac{x}{2^j}, \frac{y}{2^j})$. In what follows, we shall use the above mentioned dyadic wavelet transform to detect abrupt normal changes as edge points.

From Equation(3), it is obvious that the vector of the normal change, $d\vec{N}(u, v)$, can be represented by the linear combination of the two bases on the $du-dv$ plane, *i.e.*, $N_u du + N_v dv$. And, their associated weights are the gradients of \vec{N} along du and dv directions, respectively. Since the dyadic wavelet transform proposed by Mallat [2, 23] is able to calculate the magnitudes of these gradients, we can apply their method directly to calculate $d\vec{N}(u, v)$.

According to the formulation reported in [2, 23], the dyadic wavelet transform of $\vec{N}(u, v)$ at scale 2^j is

$$\mathcal{W}_j \vec{N}(u, v) = W_{2^j}^1 \vec{N}(u, v) du + W_{2^j}^2 \vec{N}(u, v) dv, \quad (37)$$

where

$$W_{2^j}^1 \vec{N}(u, v) = (W_{2^j}^1 n_1(u, v), W_{2^j}^1 n_2(u, v), W_{2^j}^1 n_3(u, v)), \quad (38)$$

and

$$W_{2^j}^2 \vec{N}(u, v) = (W_{2^j}^2 n_1(u, v), W_{2^j}^2 n_2(u, v), W_{2^j}^2 n_3(u, v)). \quad (39)$$

Since the $\mathcal{W}_j \vec{N}(u, v)$ vector also lies on the $du-dv$ plane, the magnitude and argument of $\mathcal{W}_j \vec{N}(u, v)$ can be directly computed. Referring to [2], the norms of $W_{2^j}^1 \vec{N}(u, v)$ and $W_{2^j}^2 \vec{N}(u, v)$ should be defined, respectively, as follows:

$$\|W_{2^j}^1 \vec{N}(u, v)\| = \sqrt{[W_{2^j}^1 n_1(u, v)]^2 + [W_{2^j}^1 n_2(u, v)]^2 + [W_{2^j}^1 n_3(u, v)]^2}, \quad (40)$$

and

$$\|W_{2^j}^2 \vec{N}(u, v)\| = \sqrt{[W_{2^j}^2 n_1(u, v)]^2 + [W_{2^j}^2 n_2(u, v)]^2 + [W_{2^j}^2 n_3(u, v)]^2}. \quad (41)$$

The magnitude of $\mathcal{W}_j \vec{N}(u, v)$ at scale 2^j can thus be computed as follows:

$$M_{2^j} \vec{N}(u, v) = \sqrt{[\|W_{2^j}^1 \vec{N}(u, v)\|]^2 + [\|W_{2^j}^2 \vec{N}(u, v)\|]^2}. \quad (42)$$

Besides, the angle of $\mathcal{W}_j \vec{N}(u, v)$ with respect to du direction is

$$A_{2^j} \vec{N}(u, v) = \text{argument}(\|W_{2^j}^1 \vec{N}(u, v)\| + i\|W_{2^j}^2 \vec{N}(u, v)\|). \quad (43)$$

From the above calculation, every point in a range image will obtain two values. One is the magnitude of its normal change with respect to its certain neighbors, and the other is the direction tendency of this point. Like other multiscale edge detection[2, 22, 24], the edge points can be determined by locating those local extrema whose normal change exceed a preset threshold.

5. Experimental results

In the experiments, a number of synthetic and real range images were adopted as test images to corroborate the proposed method. The four synthetic images were: agpart, column, half sphere and taperoll. All of these images were of size 240×240 . Besides, we also used ten real images with different sizes. In the first stage of the experiment, the normal of every point in a range image has to be decided. In order to make comparison, we used three different methods, *i.e.*, quadratic surface fitting, orthogonal and non-orthogonal wavelet-based approaches, to calculate the normal values. This part we used one of the synthetic images - “agpart”, as the test image. Experimental results of this part were reported in Figure 6. Figure 6(a) showed the three normal components (n_1, n_2, n_3) detected by applying the quadratic surface fitting method. Figure 6(b) and (c) showed, respectively, the normal components obtained by applying the orthogonal (Daubechies’ wavelets with $N = 3$) and the non-orthogonal wavelet-based approaches. Based on the derived normals, we then used the dyadic wavelet transforms propsoed by Mallat *et al.*[2] to calculate the normal change of every point in an image and selected those local extrema as edge points. Figure 7(a)-(c) showed, respectively, the multiscale edges (2^1 , 2^2 , and 2^3) detected from three differently approximated surfaces. Among them, Figure 7(a) showed the multiscale edges detected from the normals generated by the quadratic surface fitting method. Figure 7(b) and (c) were the multiscale edges detected when the approximation method were the orthogonal and non-orthogonal wavelet-based approach, respectively. It is apparent that when the quadratic surface fitting or the orthogonal wavelet-based approach was adopted to estimate the normal values, the detected edges contained some spurious results or the original edges delocalized from their original position. On the other hand, the edges detected from the normals estimated by the non-orthogonal wavelet-based approach were the best results. Figure 8(a)-(c) reported the results of the other three synthetic images based on the non-orthogonal wavelet-based approximation. One thing to be noticed is that the first stage of the proposed approach, *i.e.*, the normal determination step, is crucial because a “poor” estimation on normal values may result in irrecoverable effect on the edge detection stage. A poor estimation method may smooth out the original image and thus delocalize edges from

their correct locations. By comparing the empirical results, the non-orthogonal wavelet-based approximation was chosen because it produced the best results out of the three mentioned methods.

Figures 9 - 18 is a sequence of experiments based on real range images. All of these results were obtained by applying the non-orthogonal wavelet-based approach to estimate the normal values. From the results, we can find that most of crease (roof) edges were detected correctly.

6. Conclusion and discussions

In this paper, we have proposed a new edge detection technique based on detecting normal changes. We have found that by detecting normal changes, both step edges and roof edges can be easily identified. Therefore, the new technique has been proven a more promising method than other methods that based their detection criterion on depth or curvature changes. The whole detection procedure is divided into two stages. In the first stage, the normal of every point is decided. Since these data points are discrete, we have proposed to use three different approximation techniques to approximate the original surfaces. Based on the approximated continuous surfaces, the normals of all points on the surfaces can be computed. Then, we have applied the dyadic wavelet transforms to calculate the normal change value of every point on the surfaces. Finally, the edge points can be determined by locating those local extrema whose normal change values exceed a preset threshold. Experimental results using both synthetic and real images have demonstrated that the proposed method is indeed superb in detecting both step and roof (or crease) edges in a range image.

Acknowledgement

The authors wish to thank the Pattern Recognition and Image Processing Laboratory at Michigan State University, East Lansing, MI for range images from their range database on their WWW site.

References

- [1] J. W. Hsieh, M. T. Ko, H. Y. Mark Liao, and K. C. Fan, “A new wavelet-based edge detector via constrained optimization”, *Image and Vision Computing*, 1996, under revision.
- [2] S. Mallat and S. Zhong, “Characterization of signal from multi-scale edges”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 710–732, July 1992.
- [3] D. Marr and E. Hildreth, “Theory of edge detection”, *Proc. Royal Soc. London B*, pp. 187–217, 1980.
- [4] S. Ghosal and R. Mehrotra, “Segmentation of range images: an orthogonal moment-based integrated approach”, *IEEE Transactions on Robotics and Automation*, vol. 61, no. 2, pp. 171–189, Aug. 1993.
- [5] O. Monga and S. Benayoun, “Using partial derivatives of 3d images to extract typical surface features”, *Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 171–189, Mar. 1995.
- [6] O. Monga and R. Deriche, “3d edge detection using recursive filtering: application to scanner images”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989, pp. 28–35.
- [7] O. Monga, R. Deriche, G. Malandain, and J. P. Cocquerez, “Recursive filtering and edge tracking: two primary tools for 3d edge detection”, *Image and Vision Computing*, vol. 9, no. 4, pp. 203–214, Aug. 1991.
- [8] M. Djebali, M. Melkemi, and D. Vandorpe, “3d range image segmentation based on deriche’s optimun filters”, in *Proceedings of IEEE International Conference on Image Processing*, 1994, pp. 503–507.
- [9] T. J. Fan, G. Medioni, and R. Nevatia, “Segmented descriptions of 3-d surfaces”, *IEEE Transactions on Robotics and Automation*, vol. 3, no. 6, pp. 527–538, Dec. 1987.

- [10] P. J. Besl, *Surface in range image understanding*, Springer-Verlag, New York, 1988.
- [11] H. M. Cung, P. Cohen, and P. Boulanger, “Multiscale edge detection and classification in range images”, in *IEEE international Conference on Robotics and Automation*, 1990, pp. 2038–2044.
- [12] M. A. Wani and B. G. Batchelor, “Edge-region-based segmentation of range images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 314–319, March 1994.
- [13] S. Ghosal and R. Mehrotra, “Detection of composite edges”, *IEEE Transactions on Robotics and Automation*, vol. 3, no. 1, pp. 14–25, Jan. 1994.
- [14] J. McCleary, *Geometry from a Differentiable Viewpoint*, Cambridge university, Cambridge, 1994.
- [15] P. J. Besl and R. C. Jain, “Invariant surface characteristics for three-dimensional object recognition in range images”, *CVGIP*, vol. 33, pp. 33–80, 1986.
- [16] G. Beylkin, “On the representation of operators in bases of compactly supported wavelets”, *SIAM J. Numer. Anal.*, vol. 29, pp. 1716–1740, Dec. 1992.
- [17] I. Daubechies, “Orthonormal bases of compactly supported wavelets”, *Communications Pure Applied Mathematics*, vol. 41, pp. 909–996, 1988.
- [18] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis”, *IEEE Trans. Information Theory*, vol. 36, no. 5, pp. 961–1005, Sept. 1990.
- [19] X. Zhou R. O. Wells, “Wavelet interpolation and approximate solutions of elliptic partial differential equations”, Tech. Rep., Computational Mathematics Laboratory, Rice University, 1993.
- [20] J. W. Hsieh, H. Y. Mark Liao, M. T. Ko, and K. C. Fan, “Wavelet-based shape from shading”, *Graphical Models and Image Processing*, vol. 57, no. 4, pp. 343–362, July 1995.
- [21] Y. Meyer, *Wavelets and operators*, Cambridge University, Cambridge, 1992.

- [22] S. Mallat and W. L. Hwang, “Singularity detection and processing with wavelets”, *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 617–643, March 1992.
- [23] S. Mallat, “Multiresolution approximation and wavelet orthonormal bases of $l(r^2)$ ”, *Transactions on Amer. Math.*, vol. 15, no. 3, pp. 69–87, Sept. 1989.
- [24] J. W. Hsieh, H. Y. Mark Liao, M. T. Ko, K. C. Fan, and Y. P. Hung, “Image registration using an edge-based approach”, *Computer Vision and Image Understanding*, 1996, to appear.

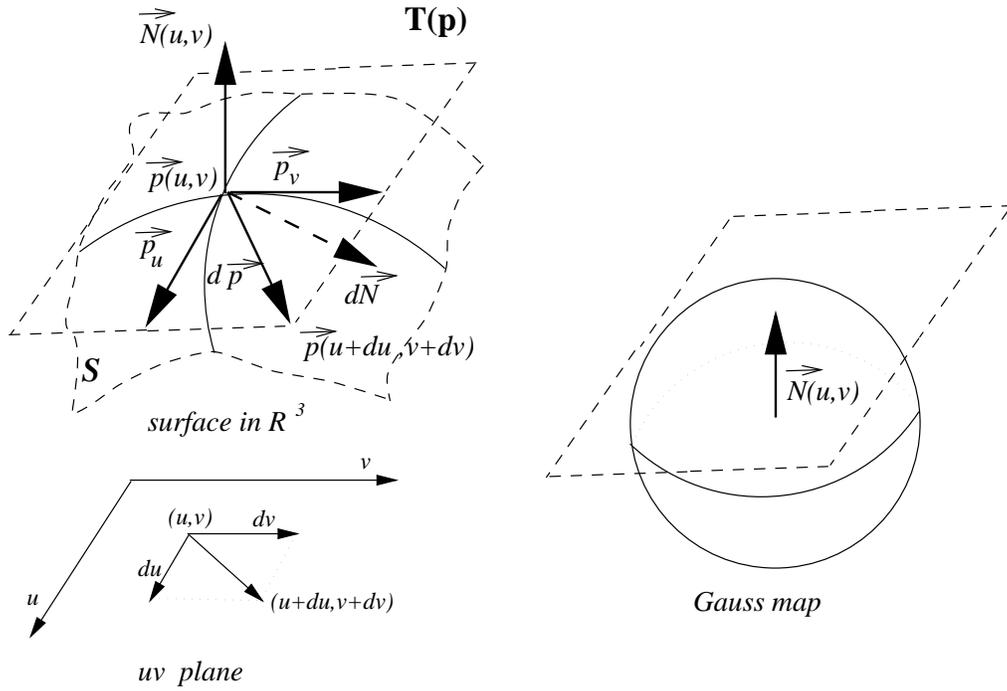


Fig. 1: Relations between a surface point, its corresponding parameter coordinate and Gauss map.

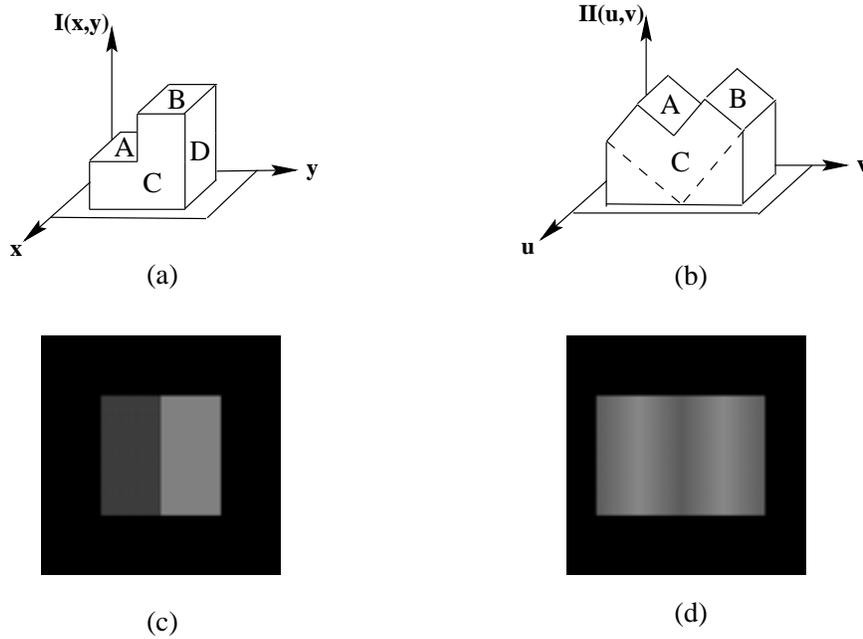
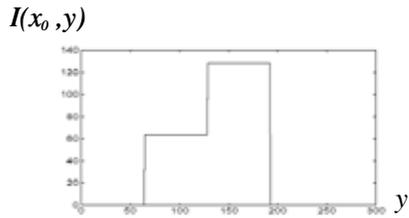


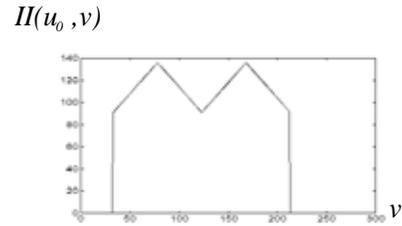
Fig. 2: An object viewed from different orientations may result in different edge types.

Step edge response

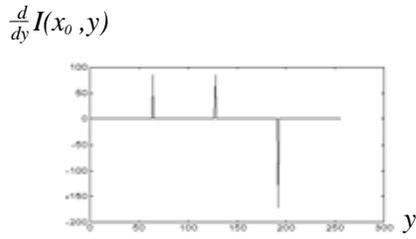


(a) Signals observed as step edges from viewpoint I

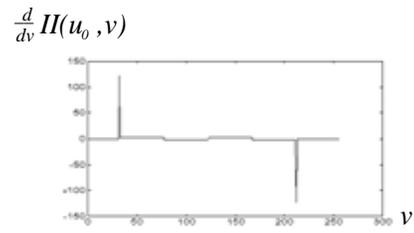
Roof edge response



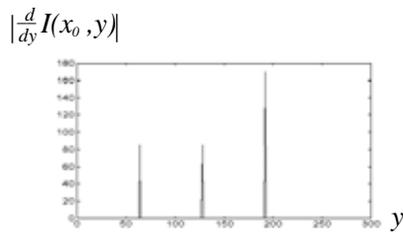
(b) Signals observed as roof edges from viewpoint II



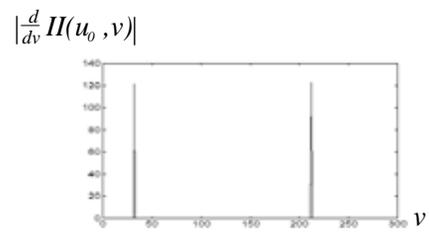
(c) First derivative of $I(x_0, y)$



(d) First derivative of $II(u_0, v)$



(e) Magnitude of $|\frac{d}{dy} I(x_0, y)|$



(f) Magnitude of $|\frac{d}{dv} II(u_0, v)|$

Fig. 3: Edge responses of step edges and roof edges.

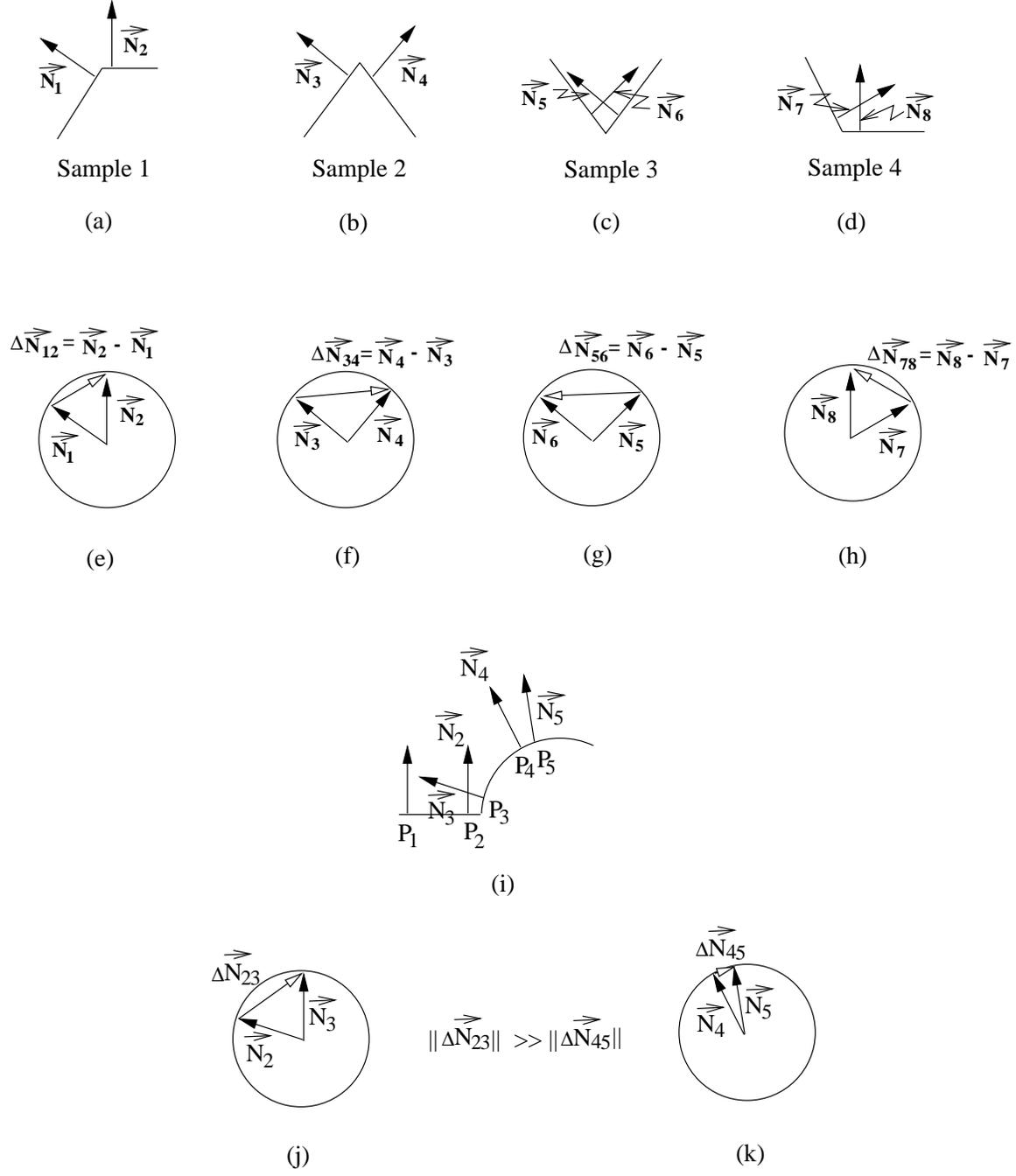
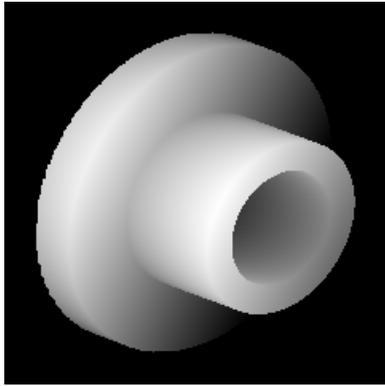
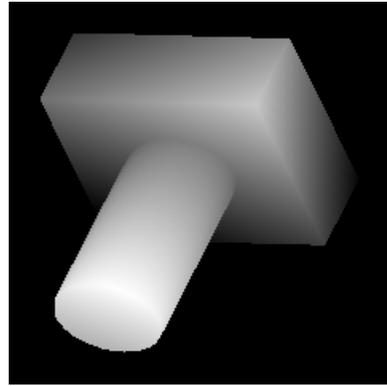


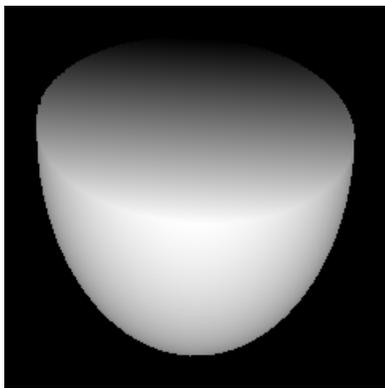
Fig. 4: Five different type of edges and their corresponding Gauss mappings.



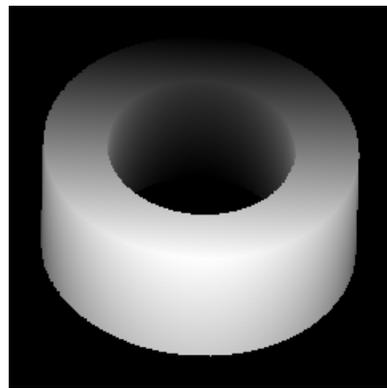
(a)



(b)



(c)



(d)

Fig. 5: Four synthetic range images.

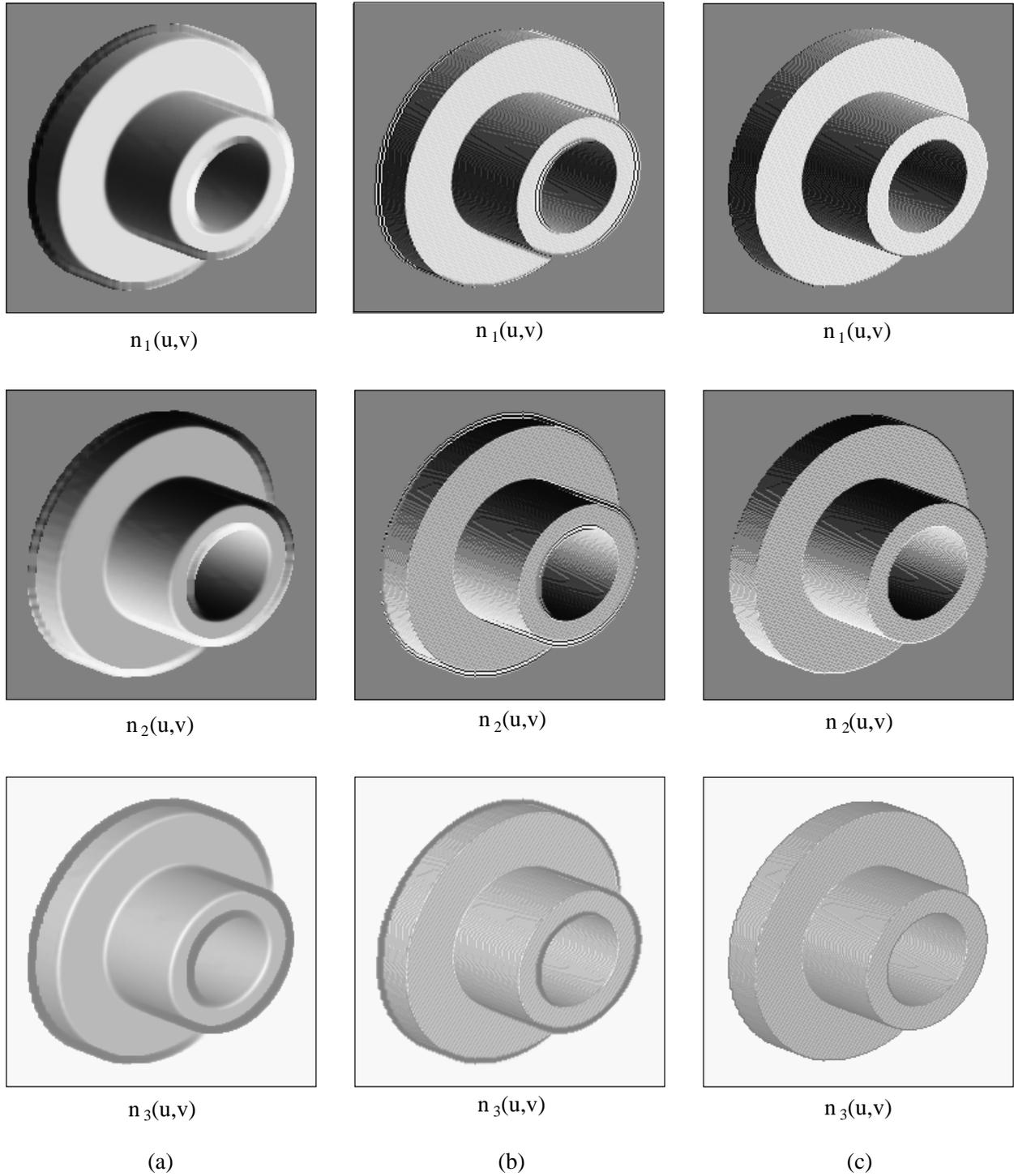
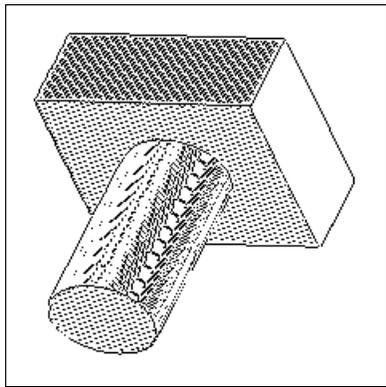
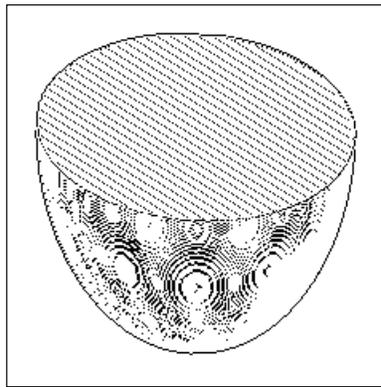


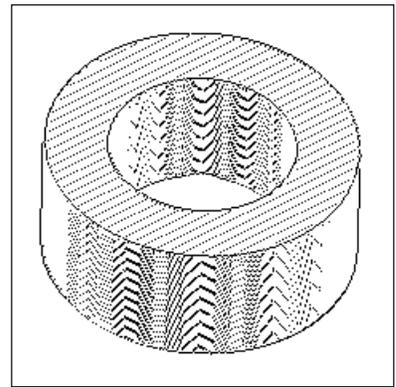
Fig. 6: Three component of normals (n_1, n_2, n_3) estimated by using (a) quadratic surface fitting (b) orthogonal wavelet-based approach with vanishing moment $N=3$, and (c) non-orthogonal wavelet-based approach.



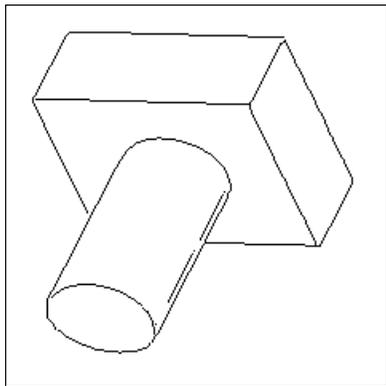
Edge points at scale 2^1



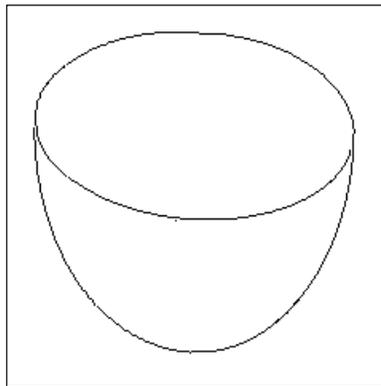
Edge points at scale 2^1



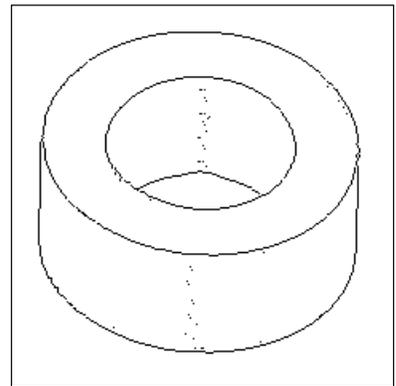
Edge points at scale 2^1



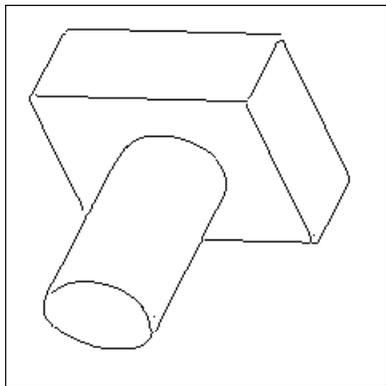
Edge points at scale 2^2



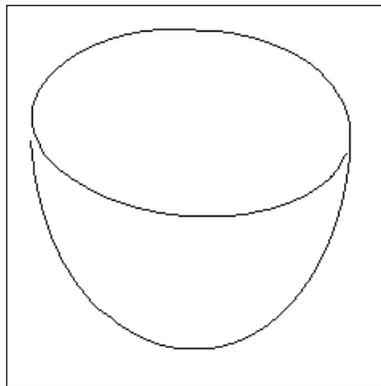
Edge points at scale 2^2



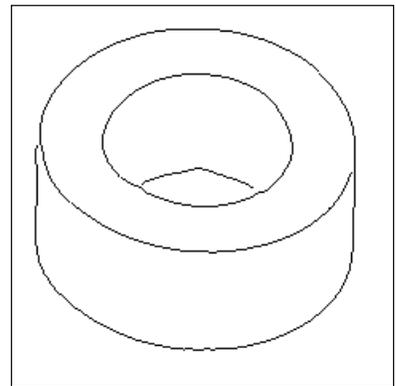
Edge points at scale 2^2



Edge points at scale 2^3



Edge points at scale 2^3



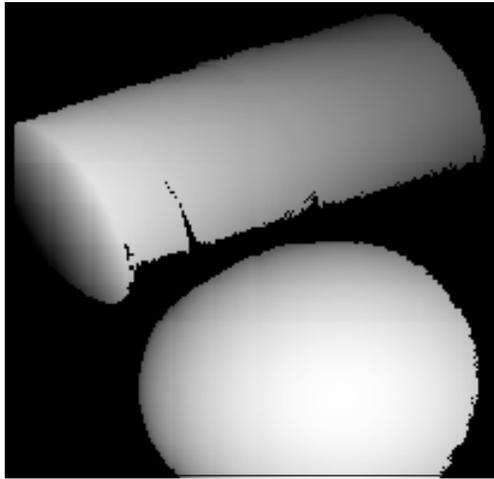
Edge points at scale 2^3

(a)

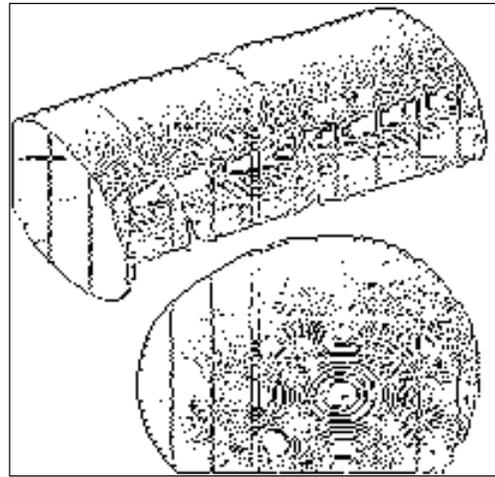
(b)

(c)

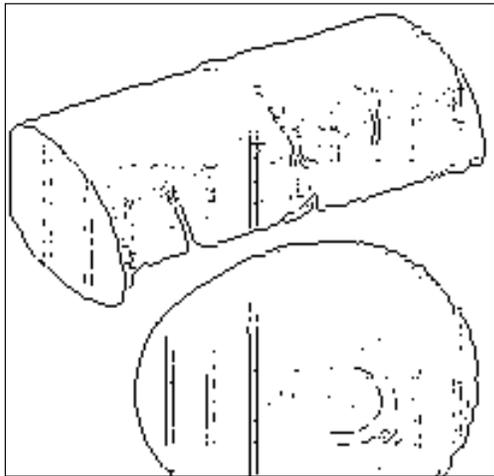
Fig. 8: Detected edge points at different scales from the other three synthetic images - (a) column (b) halfsph (c) taperoll.



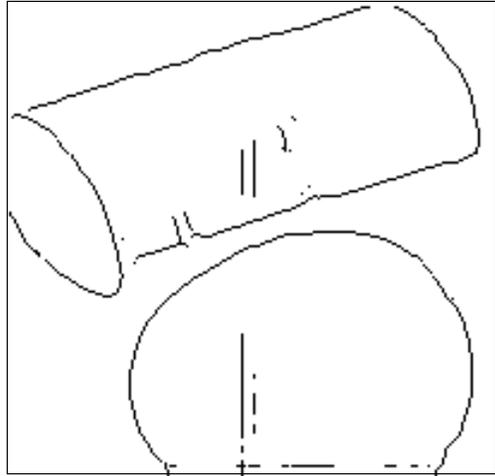
(a) Original range image



(b) Edge points at scale 2^1

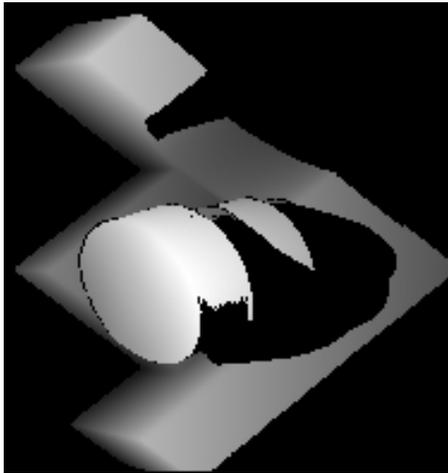


(c) Edge points at scale 2^2

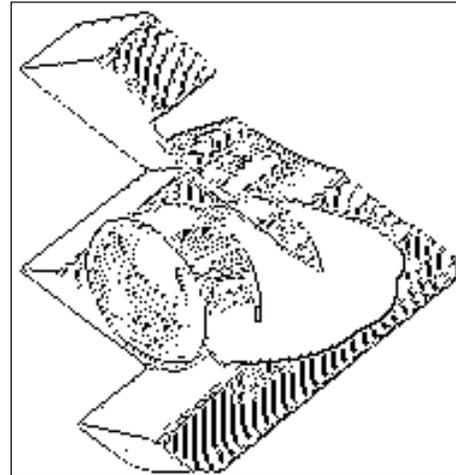


(d) Edge points at scale 2^3

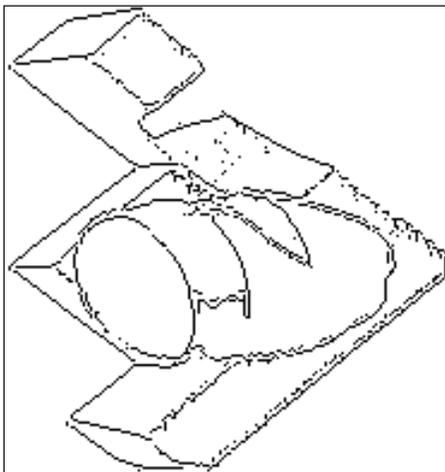
Fig. 9: Detected edge points at different scales from real images - "ball2+propane"



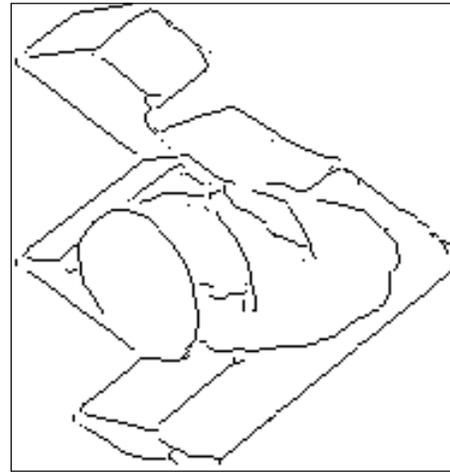
(a) Original range image



(b) Edge points at scale 2^1

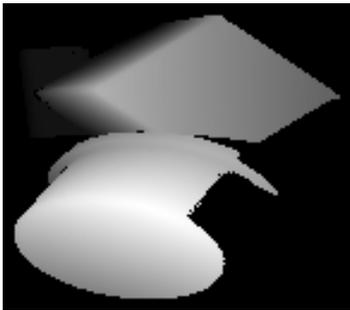


(c) Edge points at scale 2^2

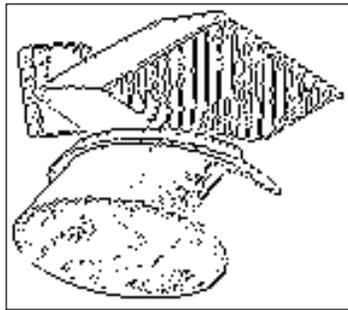


(d) Edge points at scale 2^3

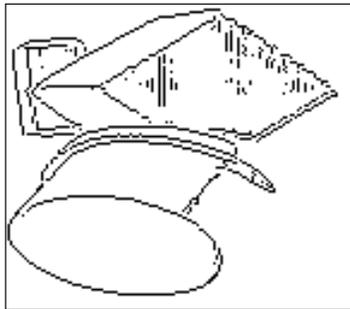
Fig. 10: Detected edge points at different scales from real images - “adapter+curvblock”



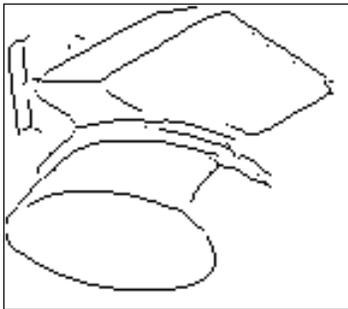
(a) Original range image



(b) Edge points at scale 2^1



(c) Edge points at scale 2^2



(d) Edge points at scale 2^3

Fig. 11: Detected edge points at different scales from real images - “cap+box2inch”

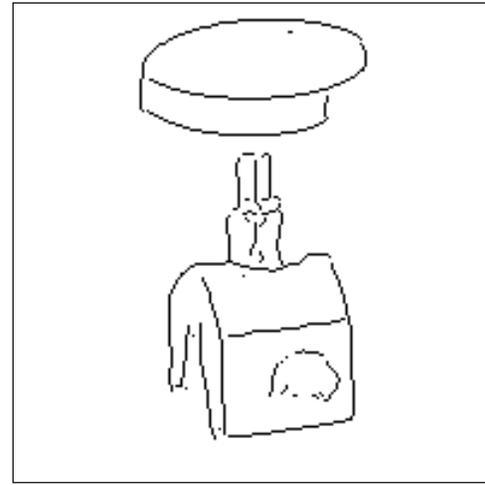
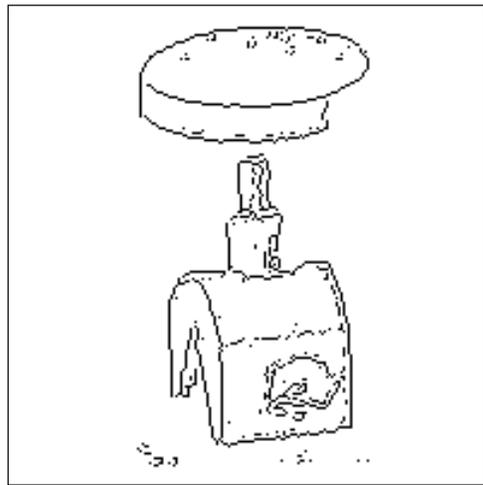
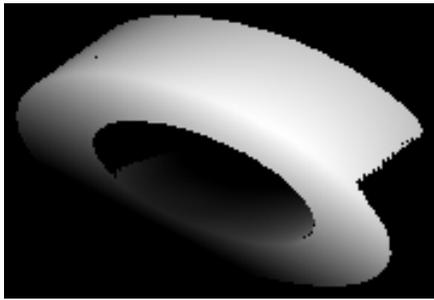
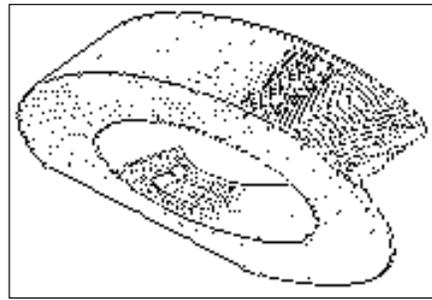


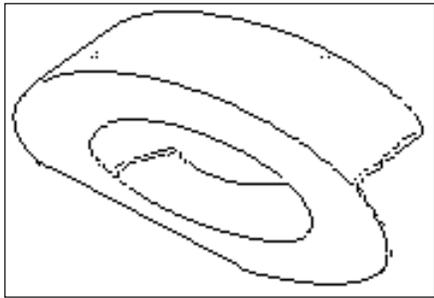
Fig. 12: Detected edge points at different scales from real images - "opticalStand"



(a) Original range image



(b) Edge points at scale 2^1

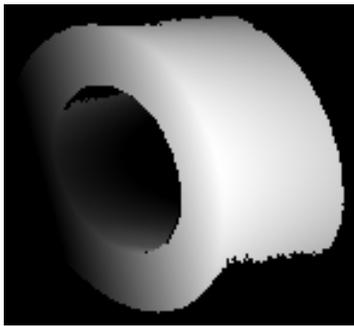


(c) Edge points at scale 2^2

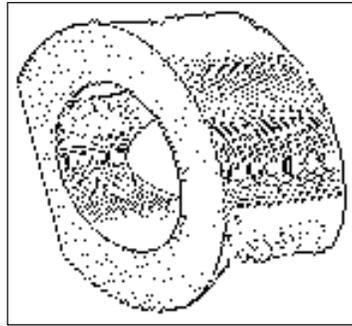


(d) Edge points at scale 2^3

Fig. 13: Detected edge points at different scales from real images - "taperoll3"



(a) Original range image



(b) Edge points at scale 2^1

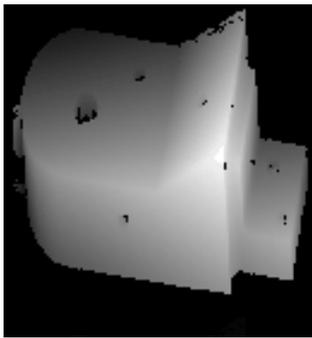


(c) Edge points at scale 2^2

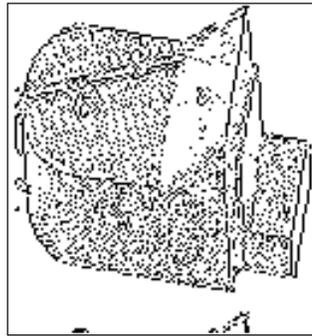


(d) Edge points at scale 2^3

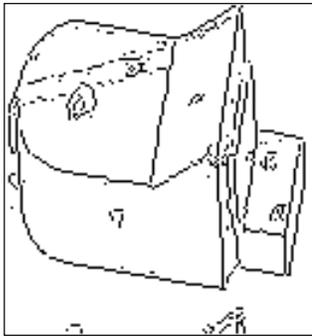
Fig. 14: Detected edge points at different scales from real images - “taperoll2”



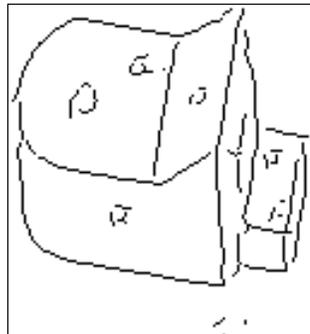
(a) Original range image



(b) Edge points at scale 2^1

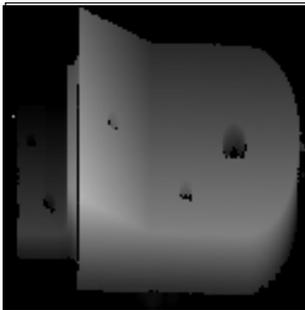


(c) Edge points at scale 2^2

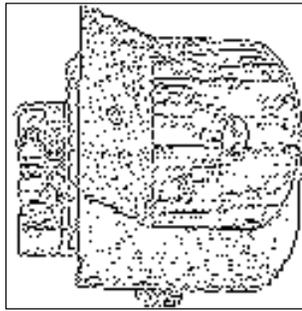


(d) Edge points at scale 2^3

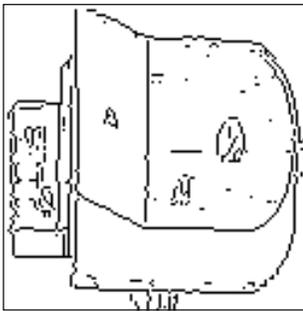
Fig. 15: Detected edge points at different scales from real images - “widget-1”



(a) Original range image



(b) Edge points at scale 2^1

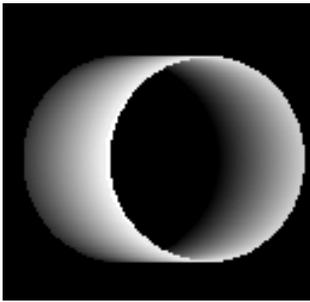


(c) Edge points at scale 2^2

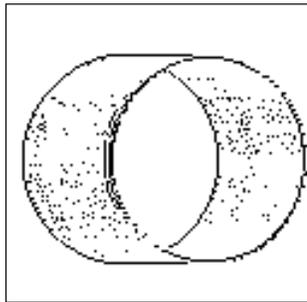


(d) Edge points at scale 2^3

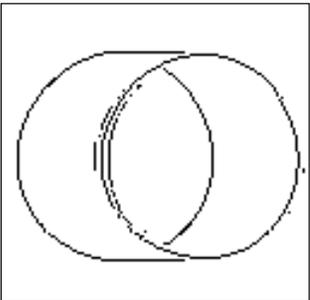
Fig. 16: Detected edge points at different scales from real images - “widget-2”



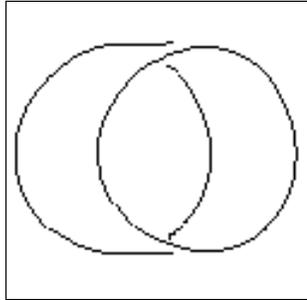
(a) Original range image



(b) Edge points at scale 2^1

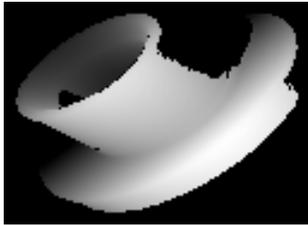


(c) Edge points at scale 2^2

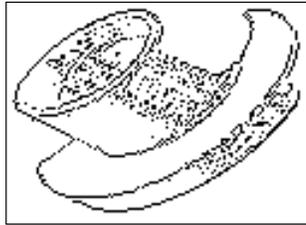


(d) Edge points at scale 2^3

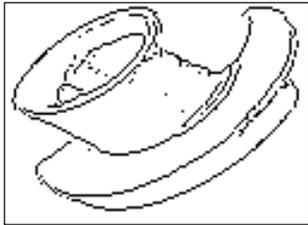
Fig. 17: Detected edge points at different scales from real images - “synthcyl5”



(a) Original range image



(b) Edge points at scale 2^1



(c) Edge points at scale 2^2



(d) Edge points at scale 2^3

Fig. 18: Detected edge points at different scales from real images - “agpart2”