THE GENERALIZED RADIAL HILBERT TRANSFORM AND ITS APPLICATIONS TO 2-D EDGE DETECTION (ANY DIRECTION OR SPECIFIED DIRECTIONS)

Soo-Chang Pei,   Jian-Jiun Ding

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C
Email address: pei@cc.ee.ntu.edu.tw

ABSTRACT

It is well-known that the Hilbert transform (HLT) is useful for generating the analytic signal, and saving the bandwidth required in communication. However, it is known by less people that the HLT is also a useful tool for edge detection. In this paper, we introduce the generalized radiant Hilbert transform (GRHLT), and illustrate how to use it for edge detection. The GRHLT is the general form of the two-dimensional HLT. Together with some techniques (such as section dividing and shorter impulse response modification), we can use the GRHLT to detect the edges of images exactly. Using the GRHLT for edge detection has higher ability of noise immunity than other edge detection algorithms. Besides, we can also use the GRHLT for directional edge detection, i.e., detect the edges with certain directions.

1. INTRODUCTION

The Hilbert transform (HLT) is defined as [1]:

$$O_{Hlt}(g(x)) = IFT(H(\omega) \cdot FT(g(x))) \qquad (1)$$

$$\text{where} \quad H(\omega) = -j \, \mathrm{sgn}(\omega). \qquad (2)$$

There are some ways to generalize the HLT into the two-dimensional (2-D) form. The simplest way is combining two 1-D HLT's together:

$$O_{Hlt(2D)}(g(x, y)) = O_{Hlt(y)}\left[O_{Hlt(x)}(g(x, y)) \right] \qquad (3)$$

Recently, in [2], Davis, McNamara, and Cottrell introduced the radial Hilbert transform (RHLT):

$$g_H(x, y) = IFT_{2D}\left[\exp(iP\theta) FT_{2D}(g(x, y)) \right] \qquad (4)$$

where $\theta = \cos^{-1}(\omega/R) = \sin^{-1}(s/R)$, $R = (\omega^2 + s^2)^{1/2}$, ω, s are the independent variables in the frequency domain, and P is any integer.

It is well-known that we can use the HLT to generate the analytic signal, and save the bandwidth required for a real signal. In fact, the HLT can also be applied to edge detection [2][3]. Fewer people know this application, but it is very important. The most serious problem for most of the edge detection algorithms is that their performance is affected by noise. However, using the HLT for edge detection can much reduce the effects of noise.

In this paper, we introduce the generalized radial Hilbert transform (GRHLT), which is the generalization of the separable 2-D HLT (see (3)) and the RHLT (see (4)). We introduce it in Sec. 2. Then, we illustrate how to use the GRHLT for 2-D edge detection and its advantage in Sec. 3. The most important two advantages are noise immunity and that the ramp edges can be detected successfully. Using the GRHLT together with the techniques introduced in Sec. 4, we can detect the edges of an image exactly. Besides, in Sec. 5, we will show that we can also use the GRHLT for directional edge detection, i.e., detect the edges with certain directions.

2. GENERALIZED RADIAL HILBERT TRANS-FORMS

We define the **generalized radial Hilbert transform** (**GRHLT**) as follows:

$$g_H(x, y) = IFT_{2D}\left(H(\omega, s) FT_{2D}(g(x, y)) \right) \qquad (5)$$

where the transfer function $H(\omega, s)$ is rotational symmetric:

$$H(\omega, s) = \Phi(\theta) \ \text{ when } (\omega, s) \neq 0, \qquad H(0,0) = 0, \qquad (6)$$

where $\quad \theta = \cos^{-1}(\omega / r) = \sin^{-1}(s / r)$,

$$r = \sqrt{\omega^2 + s^2} \ , \quad \Phi(\theta) \text{ is any function.} \qquad (7)$$

The separable 2-D HLT defined as in (3) is a special case of the GRHLT where $\Phi(\theta) = 1$ when $0 < \theta < \pi/2$ or $\pi < \theta < 3\pi/2$, $\Phi(\theta) = -1$ when $\pi/2 < \theta < \pi$ or $3\pi/2 < \theta < 2\pi$, and $\Phi(\theta) = 0$ when $\theta = N\pi/2$. The RHLT introduced in [2] is a special case of the GRHLT where

$$\Phi(\theta) = \exp(jP\theta). \qquad (8)$$

Besides, many other operations (such as the analytic signals generating operations, the 1-D HLT, and identity operation) are also the special cases of the GRHLT.

The GRHLT is very general and flexible. Because of its flexibility, it can solve many problems that can't be solved well by its special cases. In Secs. 3~5, we show that the GRHLT is a very powerful tool for edge detection.

When doing the edge detection, we usually use the discrete counterpart of the GRHLT, i.e., the **discrete generalized radial Hilbert transform**:

$$g_H[m, n] = IDFT_{2D}\left(H[p, q] DFT_{2D}(g[m, n]) \right) \qquad (9)$$

where p, q, are the discrete independent variables in the frequency domain, and $H[p, q]$ is rotational symmetric:

$$H[p, q] = \Phi(\theta) \qquad \text{when } [p, q] \neq [0, 0],$$

$$H[0,0] = 0, \qquad H[M/2, 0] = 0 \ \text{ if } M \text{ is even,}$$

$$H[0, N/2] = 0 \quad \text{if } N \text{ is even,} \quad (M \times N: \text{ the size of } g[m, n]),$$

$$\theta = \cos^{-1}(p / r) = \sin^{-1}(q / r), \quad r = \sqrt{p^2 + q^2}$$

$$\Phi(\theta) \text{ is any function.} \qquad (10)$$

Due to the DFT and IDFT, the discrete generalized radial Hilbert transform has fast algorithm. Its complexity is $MN \cdot \log_2 MN$.

3. USING THE GRHLT FOR EDGE DETECTION

The simplest way for 2-D edge detection is doing the difference operation. That is, if

(horizontal) $\left| g[m_0, n_0] - g[m_0, n_0 + 1] \right| >$ threshold

(vertical) $\left| g[m_0, n_0] - g[m_0 + 1, n_0] \right| >$ threshold $\qquad (11)$

we can conclude that the pixel (m_0, n_0) is on the edge. Besides, there are also some edge detection methods based on the convolution with 3×3 matrix, e. g., the compass gradient mask, Laplacian mask, and statistical mask methods [4]. Their ideas are similar to the difference operation.

However, most of the above edge detection methods are highly influenced by noise. We do an experiment in Fig. 1. Fig. 1(a) is the input image, and Fig. 1(b) is the image interfered by the noise. In Fig. 1(c), 1(d), we use the Laplacian mask method to detect the edges of Fig. 1(a), 1(b), respectively. It is appearance that the effect of the Laplacian mask method is much influenced by noise. This is the most important problem for Laplacian mask method and most of the existed edge detection algorithms.
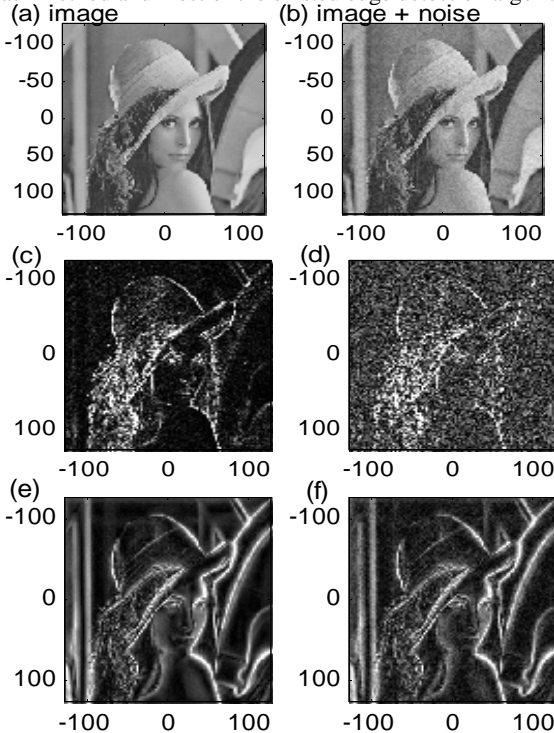

Fig. 1 The experiment for noise immunity. (a): Image. (b): Image + noise. (c)(d): Using the Laplacian mask to detect the edges of (a) and (b). (e)(f): Using the GRHLT to detect the edges of (a) and (b).

However, if we use the GRHLT for edge detection, we can much reduce the effect of noise.

In (10), if the transfer function we choose satisfies

$$\Phi(\theta) = -\Phi(\theta + \pi) \qquad \text{for all } \theta, \qquad (12)$$

we can use the GRHLT for 2-D edge detection. This is so because in this case the impulse response (denoted by $h[m, n] = \text{IDFT}(H[\omega, s])$ of the GRHLT has the following two properties:
(1) odd symmetric: $h[m, n] = -h[-m, -n]$,
(2) $|h[m, n]|$ has the trend of becoming smaller when $|m|$, $|n|$ grow larger.

In fact, except for the GRHLT, all the 2-D LTI operations whose impulse responses satisfy the above two constraints can also be used for edge detection.

The GRHLT can much reduce the effect of noise because it has longer impulse response. The impulse response of the difference method (see (11)) are $[-1, 1]$ or $[-1, 1]^T$. Their lengths are 1×2 and 2×1. The impulse responses of the methods based on 3×3 matrix convolution (such as the Laplacian mask method) are 3×3. However, the GRHLT has much longer impulse response, so it can reduce the effect of noise. In Fig. 1(e) and 1(f), we do the GRHLT for Fig. 1(a) and 1(b), respectively. Fig. 1(e) shows that we can use the GRHLT to detect an image successfully. In Fig. 1(f), we show that even if the original image is interfered by noise, the performance of edge detection doesn't become worse. So using the GRHLT for edge detection is noise immunity.

Besides, if $\Phi(\theta)$ satisfies

$$\Phi(\theta) = \overline{\Phi(\theta + \pi)}, \qquad (13)$$

then the GRHLT has the property of real-input-real-output. That is, if the input $g[m, n]$ is real, then the output $g_H[m, n]$ of the GRHLT is also a real function. If we use the RHLT (see (4)), i.e., $\Phi(\theta) = \exp(j\theta)$, although it has good performance in edge detection, for real input, the output may not be real since $\Phi(\theta) = \exp(j\theta)$ doesn't satisfy (13). We can try to choose $\Phi(\theta)$ to satisfy (12) and (13). For example, we can choose $\Phi(\theta)$ as:

$$\Phi(\theta) = j\,\text{sgn}\big(\cos(\theta - \pi/6)\big)\cos^{0.2}(\theta - \pi/6). \qquad (14)$$

In this case, $\Phi(\theta)$ satisfy (12) and (13), and the corresponding GRHLT has the property of real-input-real-output.

The advantages of using the GRHLT for edge detection are:
(1) Noise immunity.
(2) Ramp edges, which are hard to detect by other edge detection algorithms, can be detected by the GRHLT.
(3) In the sense of sight, the output has better quality if we use the GRHLT for edge detection.
The 2nd and 3rd advantages can be seen from the comparison of Fig. 1(e) with Fig. 1(c).

4. SOME TECHNIQGES TO IMPROVE THE PERFORMANCE OF EDGE DETECTION

4.1. Dividing the input into several sections

We like to divide the input image into several sections, and use each of them as the input of the GRHLT for edge detection. It has two advantages: (1) The complexity can be reduced. (2) The performance can be improved.

Since the GRHLT is implemented by the 2-D DFT / IDFT, so its complexity is $MN \cdot \log_2 MN$ where $M \times N$ is the size of image. If we divide the image into S^2 sections, then the size of each section is near to $(M/S) \times (N/S)$. So the complexity becomes

$$S^2 \cdot \frac{MN}{S^2} \log_2 \frac{MN}{S^2} = MN \log_2 \frac{MN}{S^2} \qquad (15)$$

This is smaller than $MN \cdot \log_2 MN$, so the complexity is reduced. We can also write (15) as a function of $M_0 \times N_0 = (M/S) \times (N/S)$ where $M_0 \times N_0$ is the size of each section:

$$MN \log_2 M_0 N_0. \qquad (16)$$

That is, if the size of sections $M_0 \times N_0$ is fixed, the **complexity grows linearly with $M \times N$**. Thus, although the edge detection algorithm using GRHLT seems more complicated than other edge detection algorithms, if we use the section-division method, the complexity is in fact $O(MN)$. This is the same as the complexities of other edge detection algorithms.

Besides, the performance can also be improved if we divide the input into several sections. This is so because, to conclude whether a pixel is on the edge, we just require the information surrounding the pixel. It is unnecessary to use the whole image as the input of the GRHLT for edge detection. In Fig. 2, we do some experiments. In Fig. 2(b), we don't divide the image, and in Fig. 2(c)(d), we divide the image into 3^2 and 9^2 sections before using the GRHLT for edge detection. We can see that, in Fig 2(c)(d), the brightness differences between edge regions and non-edge regions are more obvious than that in Fig. 2(b).
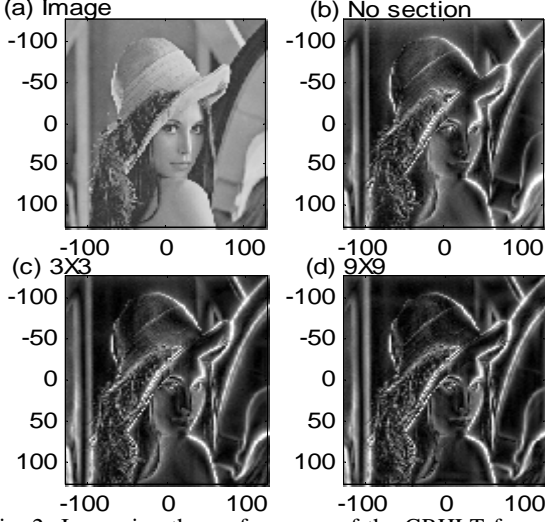

Fig. 2 Improving the performance of the GRHLT for edge detection by section-division method. (a) Image. (b) The results if we do not separate the input. (c)(d) The results if we separate the input into 3^2 sections and 9^2 sections.

4.2. Appending tapered borders

When doing edge detection, the borders of an image are usually misunderstood as the edges. This problem can be overcome by appending tapered borders to the original image before doing the edges detection. That is, if the original image is g[m, n] where $1-N_1 \le m, n \le N-N_1$, $N_1 = [N/2]+1$, then we can append the upper and lower tapered borders to $g[m, n]$ by:

$g[m,n] = (m+B-1+N_1)g[1-N_1, n]/B$ when $2-N_1-B \le m <1-N_1$,

$g[m,n] = (N+N_1+B-m)g[N-N_1,n]/B$ when $N-N_1 \le m< N+N_1+B-1$. (17)

We can use the similar way to append the left and right tapered borders to g[m, n]. After appending tapered borders, the borders of an image are no longer detected as edges.

4.3. Choosing the adaptive threshold

In Figs. 1, 2, we directly show the output of the GRHLT. In fact, we can also choose a threshold. If

$$\left| g_H \left[m_0, n_0 \right] \right| \quad > \quad \text{threshold} \qquad (18)$$

where $g_H[m, n]$ is the output of the GRHLT, then we can conclude (m_0, n_0) is on an edge. Then, one may ask how to choose the threshold. The simplest way is choosing the threshold as some constant. However, it has worse performance for a complicated image. Here, we propose an localized and adaptive method to choose the threshold function $T[m, n]$:

$$T[m,n] = \frac{Z_c[m,n] + aT_0}{(a+1)b} \qquad (19)$$

where $Z_c[m,n] = \left| g_H[m,n] \otimes A_c \right|$,

Ac is a $c \times c$ matrix where $A_c(m, n) = 1$ for all m's, n's, T_0 is the average value of $|g_H[m, n]|$. (20)

In the above, there're 3 parameters a, b, c. We can adjust them to achieve better performance. In principle, for a simple image (such as the fruit image in Fig. 3(a)), we choose larger values of a and c and smaller value of b. For a complicated image (such as the Lena image in Fig. 3(b)), we choose smaller value of a and c and larger value of b. In Fig. 3(c), we choose (a, b, c) as (0.65, 0.8, 15), and in Fig. 3(d), we choose (a, b, c) as (0.5, 0.85, 9).
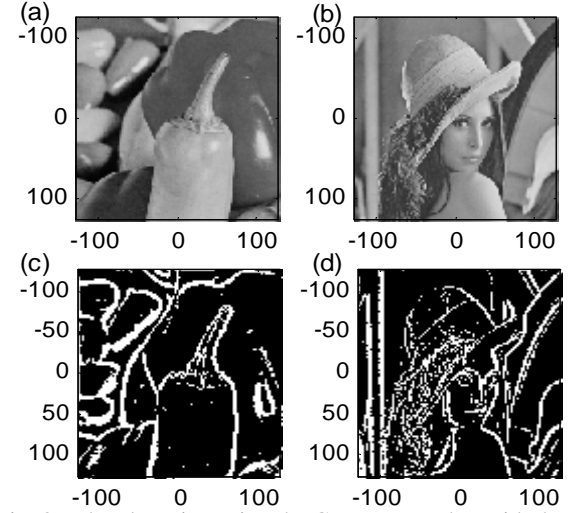

Fig. 3 Edge detection using the GRHLT together with the adaptive threshold function defined as in (19), (20).

4.4. Shorter impulse response modification

We have stated that the advantage of the GRHLT is that it can reduce the effects of noise. It is mainly due to the GRHLT has longer impulse response. However, it has some side effect. That is, the result is not sharp enough, and sometimes the brightness of the edges region may not be obviously higher than the brightness of the non-edge regions. This problem can be overcome if we modify the transfer function of the GRHLT a little. We can modify the transfer function of the GRHLT in (10) as:

$$H_d[p,q] = \Phi(\theta) \otimes A_d , \qquad (21)$$

where $\otimes$ is circular convolution, A_d is a $d \times d$ matrix, $A_d(m, n) = 1$ for all m's, n's, and $\Phi(\theta)$ is defined the same as previous sections. If c is larger, the impulse response $h_d[m, n] = \text{IDFT}(H_d[p, q])$ will become shorter. Shorter impulse response makes the brightness difference between the edge regions and the non-edge regions become more obvious.

In Fig. 4, we do some experiments. In Fig. 4(a)(c), we show the slicing (along x-axis) of the impulse response of the original GRHLT and the modified GRHLT ($d = 9$). It can be seen that the impulse response of the modified GRHLT becomes shorter. Then, in Fig. 4(b)(d), we show the transform results of the original GRHLT and modified GRHLT, respectively. The brightness difference between the edge and the non-edge regions of the result of the modified GRHLT is more obvious.

Thus, using the shorter impulse response GRHLT for edge detection can obtain better performance. However, the ability of noise immunity is reduced. There is a tradeoff between the two

goals: (1) better performance of edge detection, (2) higher ability for noise immunity.
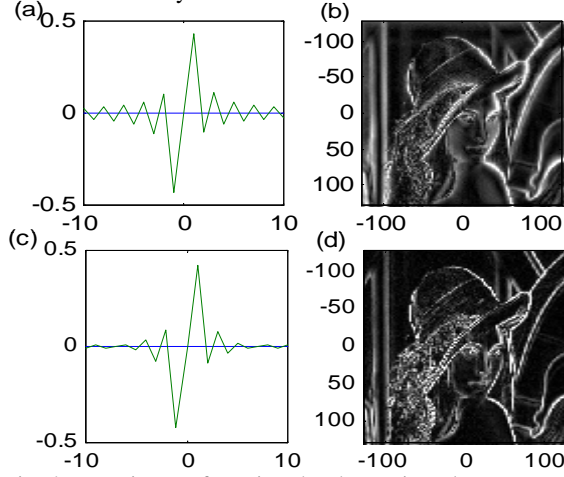

Fig. 4 Experiments for using the shorter impulse response modified GRHLT for edge detection. (a)(b) The impulse response (*x*-axis slicing) and the transform results of the original GRHLT. (c)(d) The impulse response and he transform results of the shorter impulse response modified GRHLT.

5. DIRECTIONAL EDGE DETECTION

In Secs. 3, 4, we detect the edges whatever the direction it has. In fact, if the transfer function of the GRHLT is chosen properly, we can detect the edges with certain directions. For example, If we want to detect the edges with the direction in the region of $[\phi_1, \phi_2]$ where $\phi_2 - \phi_1 < \pi$, then, in (10), we can choose

$$\Phi(\theta) = -j \text{ when } \phi_1 \le \theta \le \phi_2, \qquad \Phi(\theta) = j \text{ when } \phi_1 - \pi \le \theta \le \phi_2 - \pi,$$
$$\Phi(\theta) = 0 \quad \text{otherwise.} \tag{22}$$

Then we do the GRHLT. If the output $|f_H(m,n)| >$ threshold, then we can conclude that (m, n) is on an edge, and the direction of the edge is in the range of $[\phi_1, \phi_2]$. In Fig. 5, we give some example. We use a circle (Fig. 5(a)) and the Lena image (Fig. 5(b)) as the input. In Fig. 5(c), we plot the imaginary part of the transfer function $H[p, q] = \Phi(\theta)$, where $\Phi(\theta)$ is defined as (22). Here, we choose $\phi_1 = \pi/3$ and $\phi_2 = 2\pi/3$. Then we do the GRHLT for the two inputs, and use the method described in subsection 4.3 to choose the adaptive threshold. We plot the transform results in Fig. 5(e) and 5(g). The edges found in Fig. 5(e) are just two arcs, and the angle ranges of the two arcs are $[\pi/3, 2\pi/3]$ and $[-\pi/3, \pi/3]$, respectively. In Fig. 5(g), we also find the edges with direction in the range of $[\pi/3, 2\pi/3]$ successfully.

Besides, if we want to detect the edges in the region of $[\phi_1, \phi_2]$ and $[\phi_3, \phi_4]$ at the same time, we can choose $\Phi(\theta)$ as:

$$\Phi(\theta) = -j \qquad \text{when } \phi_1 \le \theta \le \phi_2, \ \phi_3 \le \theta \le \phi_4,$$
$$\Phi(\theta) = j \qquad \text{when } \phi_1 - \pi \le \theta \le \phi_2 - \pi, \ \phi_3 - \pi \le \theta \le \phi_4 - \pi,$$
$$\Phi(\theta) = 0 \qquad \text{otherwise.} \tag{23}$$

For example, in Fig. 5(d), we choose $[\phi_1, \phi_2] = [\pi/6, \pi/3]$, and choose $[\phi_3, \phi_4] = [2\pi/3, 5\pi/6]$. In Fig. 5(f) and 5(h), we plot the transfer results of the GRHLT (with threshold) with this transfer function for the circle image and Lena image. We detect the edges with the direction in the ranges of $[\pi/6, \pi/3]$ and $[2\pi/3, 5\pi/6]$ successfully.

Thus, using the GRHLT for directional edge detection is very flexible and convenient.
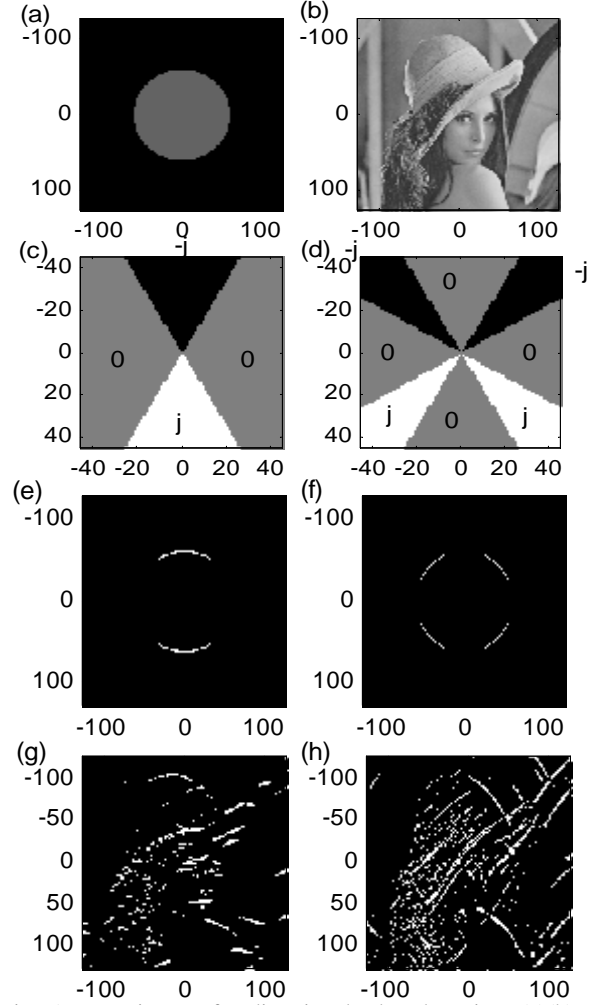

Fig. 5 Experiments for directional edge detection. (a)(b): The inputs. (c)(d): Transfer functions $\Phi(\theta)$. (e)(g): The transform results if we use (c) as the transfer function. (f)(h) The transform results if we use (d) as the transfer function.

6. CONCLUSION

We have introduced the generalized radiant Hilbert transform (GRHLT), and illustrated how to use it for 2-D edge detection. Using the GRHLT for edge detection is very flexible, and can much reduce the effect of noise. We can even use the GRHLT for directional edge detection. Thus, except for communication, the GRHLT is also an important tool for image processing.

7. REFERENCES

[1] S. C. Hahn, *Hilbert Transforms in Signal Processing*, Boston: Artect House, 1996.

[2] J. A. Davis, D. E. McNamara, and Don M. Cottrell, "Image processing with the radial Hilbert transform: theory and experiments", *Opt. Lett.*, vol. 25, p. 99-101, Jan. 2000.

[3] A. W. Lohmann, D. Mendlovic, and Z. Zalevsky, "Fractional Hilbert transform", *Opt. Lett.*, vol. 21, p 281-283, Feb. 1996.

[4] Pratt, William K, *Digital image processing*, 3rd Ed., New York: Wiley, 2001.