Optimal Polynomial Filters

Submitted to Journal of Graphics Tools

Zhouchen Lin Hai-Tao Chen Heung-Yeung Shum Jian Wang
Microsoft Research, Asia
{*zhoulin*|*i-htchen*|*hshum*|*jianw*}*@microsoft.com*

Abstract

In this paper, we present a family of circular or square optimal polynomial filters for pre-filtering 2D polygons and images. The criterion of designing polynomial filters is to maximize the energy concentration within a period of the spectra of the filters. The filters are non-negative and can have arbitrary radius and order. For a given radius, the filters converge very fast when the order increases, making low-order filters suffice for high-quality pre-filtering. With polynomial filters, it is convenient to evaluate the integral over the parts of polygons within the filter mask with closed-form solutions, or generate look-up tables quickly via analytic evaluation. The experiments demonstrate the excellent anti-aliasing performance of our polynomial filters.

1 Introduction

Anti-aliasing is a fundamental problem in computer graphics, in which choosing a good low-pass filter is critical to remove undesirable artifacts. Filter design has a long history in signal processing. For eliminating high frequencies, it is well known that the *sinc* function is the ideal filter. Unfortunately, it is of infinite support and is unusable in practice. Therefore, people have been searching for its alternatives. As simple filters can save much computation, box, conical, Gaussian, and cubic (including splines) filters are commonly used in computer graphics, either explicitly or implicitly. McCool proposed prism splines [7], but the evaluation is very complex and expensive. The Mitchell-Netravali filters [8] are also popular as people have found that they have good anti-aliasing performance [2, 4, 3]. The practitioners usually choose among the above-mentioned filters via visual examination on the rendered results. Therefore, the chosen optimal filter might be biased towards the experimental examples and be subjective.

This paper is motivated by our work on a high-quality pre-filtering algorithm for 2D polygons [6], which evaluates the integral inside the filter mask by breaking the integral region into basic component regions bounded by a polygon edge, a radius passing one of the edge ends and the filter boundary. Our goals are to have filters that objectively minimize aliasing, and that can provide closed-form solutions to or generate a look-up table analytically for the basic component integral, i.e., the integral over the basic component region. We have developed a family of circular or square low-order polynomial filters that maximize the energy concentration in a period of their spectra. These filters can be used for pre-filtering polygons as presented in [6], and for processing discrete images as will be illustrated in this paper.

2 Filter Design

We follow two criteria to design filters. First, the filter should minimize aliasing. Second, it must provide closed-form solutions to the basic component integral when a look-up table is not preferred.

For the first criterion, we have to measure the "amount" of aliasing. It is well known that after sampling, the spectrum of a continuous signal replicates in the spectral domain. When the sampling is on a 2D square grid, the period is $2\Omega = 2\pi/T$ in both x and y directions, where T is the sample spacing in the spatial domain, normalized to 1 in our problem. The amount of aliasing can be measured by the energy of the spectrum outside the square $\mathbf{\Omega} = [-\Omega,\ \Omega] \times [-\Omega,\ \Omega]$ in the spectral domain. Therefore, we should make the spectral energy most centered on the square. This leads to an

optimization problem:

$$h = \arg\max_h \frac{\int_{-\Omega}^{\Omega}\int_{-\Omega}^{\Omega} |[\mathcal{F}(h)](\omega_x,\omega_y)|^2 d\omega_x d\omega_y}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |[\mathcal{F}(h)](\omega_x,\omega_y)|^2 d\omega_x d\omega_y}, \tag{1}$$

where $[\mathcal{F}(h)]$ denotes the Fourier transform of the filter h. This is similar to the aliasing energy in [5]. The difference is that we use a ratio instead. Without any constraint, the solution to (1) is the well-known *sinc* function and the energy concentration is 1. However, in practice we want the filter be finitely supported. Under such a constraint, the solution to (1) becomes the prolate spheroidal wave function of order zero [9]. Unfortunately, the prolate spheroidal wave functions do not have closed-form solutions, even power-series solutions are unavailable.

For the second criterion, among all elementary functions, only polynomials are possible to have closed-form solutions to the basic component integral.

Following the above considerations, we design optimal polynomial filters, either square or circular, respectively. We only present these two types of filters because they are most commonly used. However, the same idea is applicable to other shapes of filters.

2.1 The optimal circular polynomial filters

For circular polynomial filters, we may assume that $h(r,\theta) = \sum_{k=0}^{M} h_k r^k$, where $r \in [0, R]$ and $\theta \in [0, 2\pi)$. Then the numerator of (1) is given by

$$\Gamma_{\mathbf{\Omega}} \equiv \int_{-\Omega}^{\Omega}\int_{-\Omega}^{\Omega} |[\mathcal{F}(h)](\omega_x,\omega_y)|^2 d\omega_x d\omega_y = \sum_{k,l=0}^{M} h_k h_l \Psi_{kl},$$

where

$$\Psi_{kl} = \int_{-\Omega}^{\Omega}\int_{-\Omega}^{\Omega} \psi_k \psi_l d\omega_x d\omega_y,$$

and ψ_m is the Fourier transform of r^m ($0 \le r \le R$):

$$\psi_m(\omega_x,\omega_y) = \int_0^R r^{m+1} dr \int_0^{2\pi} e^{-ir(\omega_x\cos\theta+\omega_y\sin\theta)} d\theta = 2\pi \int_0^R r^{m+1} J_0(r|\omega|) dr,$$

in which $J_0(\cdot)$ is the 0-th order Bessel function and $|\omega| = \sqrt{\omega_x^2+\omega_y^2}$.

By Parseval's theorem on Fourier transform, the denominator of (1) is given by

$$\Gamma \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |[\mathcal{F}(h)](\omega_x,\omega_y)|^2 d\omega_x d\omega_y = (2\pi)^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |h(x,y)|^2 dx dy = (2\pi)^2 \sum_{k,l=0}^{M} h_k h_l \Phi_{kl},$$

where

$$\Phi_{kl} = \int_0^{2\pi}\int_0^R r^{k+l+1} dr d\theta = \frac{2\pi R^{k+l+2}}{k+l+2}.$$

The maximization of $\gamma = \Gamma_{\mathbf{\Omega}}/\Gamma$ can be achieved by utilizing the Lagrangian multiplier:

$$\frac{\partial(\Gamma_{\mathbf{\Omega}} - \lambda(\Gamma-1))}{\partial h_k} = 0, \text{ and } \frac{\partial(\Gamma_{\mathbf{\Omega}} - \lambda(\Gamma-1))}{\partial \lambda} = 0. \tag{2}$$

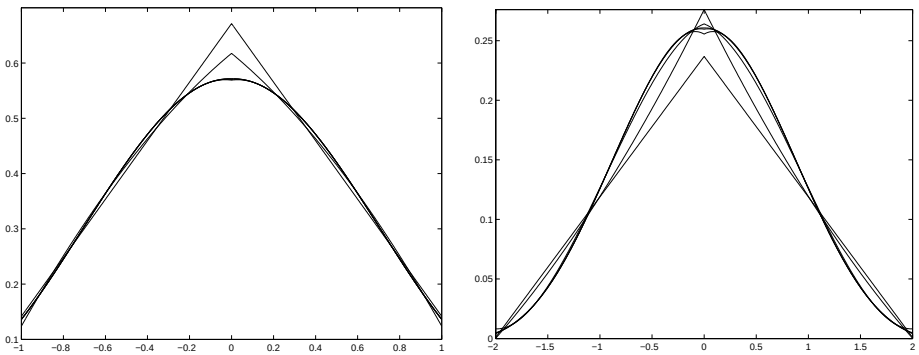
This leads to an eigenvector problem:

$$\mathbf{\Phi}^{-1}\mathbf{\Psi}\mathbf{h} = \lambda \mathbf{h}, \tag{3}$$

$$\text{and} \quad \mathbf{h}^T\mathbf{\Phi}\mathbf{h} = 1, \tag{4}$$

Table 1: Some data on low-order polynomial filters.

Filter Type	Circular		Square	
Radius	1	2	1	2
h_0	0.56904256713865	0.26412404360302	0.71247167744650	0.45090000658189
h_1	0.05056692464147	-0.03455158299490	-0.79090785614255	-0.22055267318930
h_2	-0.91026906187835	-0.15934681709384	0.25582137300511	0.02911569510537
h_3	0.42672641722501	0.05631744983035		
Energy Concentration	0.93801371604876	0.99950079154631	0.96260897769620	0.99888861744658
Maximum Energy Concentration	0.93801385499638	0.99965857174068	0.96261252017346	0.99988655625306



(a) (b)

Figure 1: The central cross sections of the circular polynomial filters. (a) Filters with radius 1, where the top, middle and bottom curves correspond to $M = 1$, $M = 2$ and $M \geq 3$, respectively. (b) Filters with radius 2, where the bottom, top and middle curves correspond to $M = 1$, $M = 2$, and $M \geq 3$, respectively. Note that the curves for $M > 3$ are nearly indistinguishable from that for $M = 3$.

where $\mathbf{\Phi} = (2\pi)^2 (\Phi_{kl})$, $\mathbf{\Psi} = (\Psi_{kl})$, and $\mathbf{h} = (h_0 \; h_1 \; \cdots \; h_M)^T$. For maximization, $\mathbf{h}$ must be chosen as the eigenvector that corresponds to the maximal eigenvalue of $\mathbf{\Phi}^{-1}\mathbf{\Psi}$ and be normalized according to (4).

Though the energy concentration improves when M increases, the convergence to the maximum energy concentration is very fast, especially when $R \leq 2$. Because polynomials can approximate arbitrary continuous functions uniformly, the fast convergence indicates that our low-order optimal polynomial filters are actually very close to the prolate spheroidal wave function. Our numerical computation shows that $M = 3$ is enough for the corresponding filter to achieve over 99.98% of the maximum energy concentration at a given radius, achieved by the prolate spheroidal wave function. Figure 1(a) is the central cross section of the optimal circular filters with $R = 1$, where the top, middle and bottom curves correspond to $M = 1$, $M = 2$ and $M \geq 3$, respectively. We can see that the curves for $M > 3$ are nearly indistinguishable from that for $M = 3$. Similar results are observable in Figure 1(b), where $R = 2$. The coefficients of the optimal circular polynomial filters at different radius are listed in the left part of Table 1.

Now we give the closed-form solution to the basic component integral. For circular filters, the basic component region is shown in Figure 2(a). By defining a new coordinate, where the t-axis is along the marching direction of the polygon edge so that the interior is on the right hand side of the edge and the d-direction is $\pi/2$ behind the t-direction, the basic component region can be parameterized by the coordinate of the polygon vertex V in the new coordinate, i.e., the distance d from the pixel center to the polygon edge and the distance t between the vertex V and the projection of the center onto the polygon edge. The basic component integral is $I(d,t) = \sum_{k=0}^{M} h_k I_k(d,t)$, where $I_k(d,t)$ is the integral of

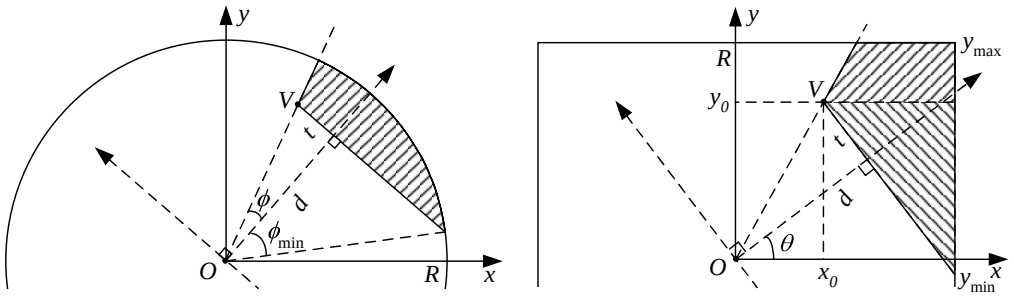


Figure 2: The basic component region (shaded area) is bounded by one polygon edge, one radius passing the polygon vertex and the filter boundary. (a) For circular filters, the parameterization (d, t) is 2D. (b) For square filters, the parameterization (θ, d, t) is 3D.

r^k over the same region. Converting to polar coordinates (Figure 2(a)), we have [1]:

$$\begin{aligned} I_k(d,t) = \int_{\phi_{\min}}^{\phi} d\xi \int_{\frac{d}{\cos\xi}}^{R} r^{k+1} dr &= \frac{1}{k+2} \int_{\phi_{\min}}^{\phi} \left[R^{k+2} - d^{k+2} (\cos\xi)^{-(k+2)} \right] d\xi \\ &= \frac{1}{k+2} \left\{ R^{k+2}(\phi - \phi_{\min}) - d^{k+2}[P_{k+2}(\phi) - P_{k+2}(\phi_{\min})] \right\}, \end{aligned}$$

where

$$\phi = \arctan(t/d),\ \phi_{\min} = -\arccos(d/R),\ \text{and } P_k(\xi) = \int (\cos\xi)^{-k} d\xi.$$

By partial integral, $P_k(\xi)$ can be computed via the following recursion:

$$P_k(\xi) = \frac{1}{k-1}(\tan\xi)(\cos\xi)^{-(k-2)} + \frac{k-2}{k-1} P_{k-2}(\xi).$$

2.2 The optimal square polynomial filters

For square polynomial filters, it is easy to see that $h(x, y)$ must be symmetric with respect to lines $x = 0,\ y = 0$, and $y = x$. Therefore, we may assume that $h(x, y) = \sum_{p=0}^{M} \sum_{q=0}^{M} h_{pq} x^{2p} y^{2q}$, with $h_{pq} = h_{qp}$, where $|x|, |y| \le R$. Then

$$\Gamma_{\mathbf{\Omega}} \equiv \int_{-\Omega}^{\Omega} \int_{-\Omega}^{\Omega} [\mathcal{F}(h)](\omega_x, \omega_y)|^2 d\omega_x d\omega_y = \sum_{p,q,k,l=0}^{M} h_{pq} h_{kl} \Psi_{2p,2k} \Psi_{2q,2l},$$

where

$$\Psi_{2m,2n} = \int_{-\Omega}^{\Omega} \psi_{2m} \psi_{2n} d\omega,$$

and ψ_{2m} is the Fourier transform of x^{2m} ($|x| \le R$):

$$\psi_{2m}(\omega) = \int_{-R}^{R} x^{2m} e^{-ix\omega} dx.$$

By partial integral, $\psi_{2m}(\omega)$ can be computed via the following recursion:

$$\psi_{2m}(\omega) = \frac{2(R\omega)^{2m-1}}{(2m)!} \left[R\sin(R\omega) + \frac{2m}{\omega} \cos(R\omega) \right] - \psi_{2m-2}(\omega).$$

[1] For simplicity we do not discuss all possible (d, t), such as the case where $d > R$ or $t < -\sqrt{R^2 - d^2}$. Section 2.2 also follows this convention.

The computation of Γ is similar, where $\Psi_{2m,2n}$ changes to

$$\Phi_{2m,2n} = \int_{-\infty}^{\infty} [\mathcal{F}(x^{2m})](\omega)\overline{[\mathcal{F}(x^{2n})](\omega)}d\omega = 2\pi \int_{-R}^{R} x^{2m}x^{2n}dx = \frac{4\pi R^{2(m+n)+1}}{2(m+n)+1},$$

in which Parseval's theorem is applied again.

Utilizing the Lagrangian multiplier as in (2), we have:

$$\mathbf{\Psi H \Psi} = \lambda \mathbf{\Phi H \Phi}, \tag{5}$$

$$\text{and} \quad trace(\mathbf{H\Phi H\Phi}) = 1, \tag{6}$$

where $\mathbf{\Phi} = (\Phi_{2i,2j})$, $\mathbf{\Psi} = (\Psi_{2i,2j})$, $\mathbf{H} = (h_{ij})$, and $trace(\mathbf{X})$ is the trace of matrix $\mathbf{X}$. Equation (5) can be rewritten as:

$$(\mathbf{A} \otimes \mathbf{A})vec(\mathbf{h}) = \lambda \cdot vec(\mathbf{h}),$$

where $\mathbf{A} = \mathbf{\Phi}^{-1}\mathbf{\Psi}$, $\otimes$ is the Kronecker product [1] of matrices, and $vec(\mathbf{X})$ is the vectorization [1] of matrix $\mathbf{X}$. Let $\tilde{\lambda}$ be the largest eigenvalue of $\mathbf{A}$ and $\tilde{\mathbf{h}} = (\tilde{h}_0\ \tilde{h}_1\ \cdots\ \tilde{h}_M)^T$ be the corresponding eigenvector, then $\tilde{\lambda}^2$ is the maximum eigenvalue of $\mathbf{A} \otimes \mathbf{A}$ and $vec(\tilde{\mathbf{h}}\tilde{\mathbf{h}}^T)$ is the corresponding eigenvector ([1], more general theorem exists). Therefore, we may choose separable $\mathbf{H} = \tilde{\mathbf{h}}\tilde{\mathbf{h}}^T$, where $\tilde{\mathbf{h}}$ is normalized via $\tilde{\mathbf{h}}^T\mathbf{\Phi}\tilde{\mathbf{h}} = 1$. This leads to a separable 2D filter $h(x,y) = h(x)h(y)$, where $h(x) = \sum_{k=0}^{M} \tilde{h}_k x^{2k}$.

Again, the convergence of $h(x)$ is fast when M increases. Our numerical experiments show that $M = 2$ is sufficient for the corresponding filter $h(x,y)$ to achieve over 99.9% of the maximum energy concentration when $R \leq 2$. The coefficients of $h(x)$ are listed in the right part of Table 1.

For square filters, the basic component region is shown in Figure 2(b), where an extra parameter θ, which is related to the polar sweep angle of the d-axis, is required for the parameterization. The basic component integral is $I(\theta,d,t) = \sum_{m=0}^{M}\sum_{n=0}^{M} h_{mn} I_{2m,2n}(\theta,d,t)$, where $I_{2m,2n}(\theta,d,t)$ is the integral of $x^{2m}y^{2n}$ over the basic component region. Due to symmetry, we may assume that $d \geq 0$ and $0 \leq \theta \leq \pi/4$. Breaking the basic component region into two shaded regions shown in Figure 2(b), we have

$$\begin{aligned} I_{2m,2n}(\theta,d,t) &= \int_{y_{\min}}^{y_0} y^{2n}dy \int_{\frac{d-y\sin\theta}{\cos\theta}}^{R} x^{2m}dx + \int_{y_0}^{y_{\max}} y^{2n}dy \int_{\alpha y}^{R} x^{2m}dx \\ &= \frac{R^{2m+1}(y_{\max}^{2n+1} - y_{\min}^{2n+1})}{(2m+1)(2n+1)} - \frac{\alpha^{2m+1}(y_{\max}^{2(m+n+1)} - y_0^{2(m+n+1)})}{2(2m+1)(m+n+1)} \\ &\quad - \frac{1}{(2m+1)(\cos\theta)^{2m+1}} \sum_{k=0}^{2m+1} (-1)^k C_{2m+1}^k (\sin\theta)^k d^{2m+1-k} \frac{y_0^{2n+k+1} - y_{\min}^{2n+k+1}}{2n+k+1}, \end{aligned}$$

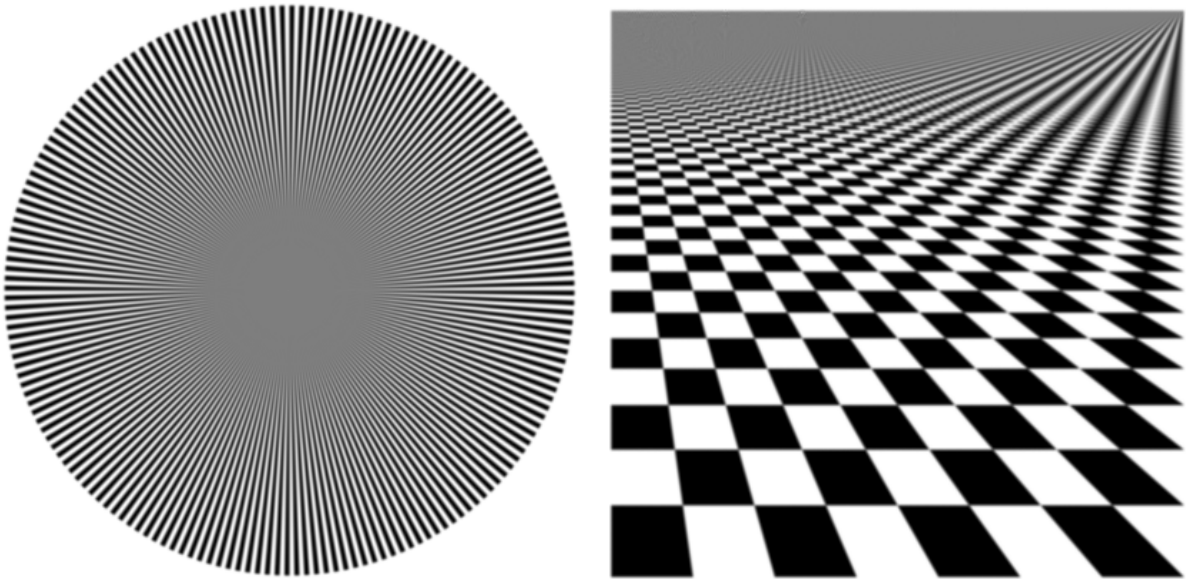
where

$$\begin{aligned} y_0 &= d\sin\theta + t\cos\theta, & x_0 &= d\cos\theta - t\sin\theta, \\ y_{\min} &= \max\left\{\frac{d - R\cos\theta}{\sin\theta}, -R\right\}, & y_{\max} &= \begin{cases} \max\{\min\{R/\alpha, R\}, -R\}, & \text{if } x_0 > 0, \\ R, & \text{if } x_0 \leq 0, \end{cases} \\ \alpha &= \frac{x_0}{y_0}, & C_m^n &= \frac{m!}{n!(m-n)!}. \end{aligned}$$

3 Experimental Results

Figure 3 shows a wheel and a checkerboard rendered by our rendering system [2], SplitRender [6], using the square polynomial filter with radius 2. There are 180 triangles in Figure 3(a), while Figure 3(b) has 92,100 quadrilaterals,

[2]The full-resolution images for Figures 3∼5 are available online at http://www.acm.org/jgt/papers/LinEtAl04.



(a) (b)

Figure 3: The rendering results of SplitRender using the square polynomial filter with radius 2, closed-form solution used.

most of which are extremely small and are at the top of the image. The rendering results are almost aliasing-free. For comparisons of our polynomial filters against various filters when pre-filtering polygons, please see [6].

Our polynomial filters can also be applied to filtering images. We now compare the anti-aliasing performance of various filters over discrete images. The test image is Figure 3(a) and will be scaled down by 1.8 times using various filters. The chosen filters include: square polynomial filters (M=2), circular polynomial filters (M=3), square Gaussian filters, circular Gaussian filters, conical filters, box filters and the Mitchell-Netravali filters [8]. The radii of the former 6 kinds of filters can be either 1 or 2, while those of the Mitchell-Netravali filters can only be 2. For Gaussian filters, we change σ from 0.1 to 1.2 to find the best filtering results, which should have good balance between eliminating aliasing and keeping the image sharp. For Mitchell-Netravali filters, the chosen (B, C) are: (1/3, 1/3), (0, 1), and (0, 0.5), respectively, as they have been mentioned in the literature [2, 4, 3]. The resultant images are computed via the following formula:

$$I_r(p,q) = \frac{\sum\limits_{||(\tilde{m}-p,\tilde{n}-q)||\leq R} h(\tilde{m}-p,\tilde{n}-q) I_t(m,n)}{\sum\limits_{||(\tilde{m}-p,\tilde{n}-q)||\leq R} h(\tilde{m}-p,\tilde{n}-q)},$$

where I_t and I_r are the test image and resultant image, respectively, $(\tilde{m}, \tilde{n})$ is the coordinate of pixel (m, n) of image I_t in image I_r, and

$$||(x,y)|| = \begin{cases} \sqrt{x^2+y^2}, & \text{if the filter is circular,} \\ \max\{|x|,|y|\}, & \text{if the filter is square.} \end{cases}$$

Figures 4 are closeups of the down-scaling results of filters with radius 1, where $\sigma = 0.3$ for Gaussian filters. We see that square and circular polynomial filters (Figures 4(a) and (b)) result in least aliasing. The performance of square Gaussian (Figure 4(c)), circular Gaussian (Figure 4(d)) and conical filters (Figure 4(e)) is very close, and the box filter (Figure 4(f)) is the worst.

Figures 5 are closeups of the down-scaling results of filters with radius 2, where $\sigma = 0.9$ for Gaussian filters. We see that square and circular polynomial filters and square Gaussian filters (Figures 5(a)$\sim$(c)) have the least aliasing. Circular Gaussian (Figure 5(d)) and conical filters (Figure 5(e)) come next, and again the box filter (Figures 5(f)) is the worst. The results of Mitchell-Netravali filters (Figures 5(g)$\sim$(i)) are very close to each other. They look sharper than the other images but their Moiré patterns are also more severe. It is interesting to compare them with our polynomial filters with

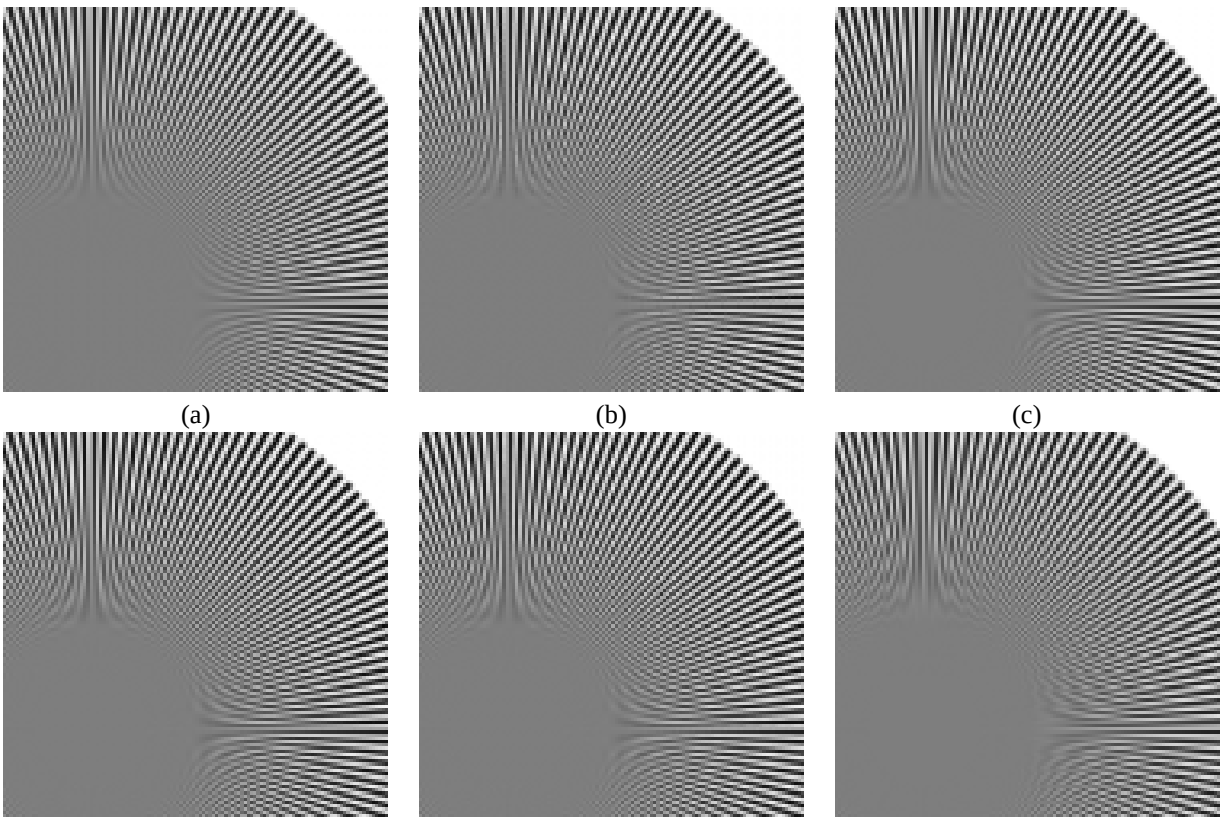


Figure 4: Closeups of downscaling Figure 3(a) by 1.8 times using various filters with radius 1. (a) Square polynomial filter. (b) Circular polynomial filter. (c) Square Gaussian filter with $\sigma = 0.3$. (d) Circular Gaussian filter with $\sigma = 0.3$. (e) Conical filter. (f) Box filter.

radius 1 (Figures 4(a) and (b)). Figures 4(a) and (b) are as sharp as Figures 5(g)$\sim$(i), but their aliasing effect is slightly less severe.

From the above comparisons, and from the polygon-filtering comparisons in [6], we may draw the following conclusions:

1. Our low-order optimal polynomial filters do have excellent anti-aliasing performance.

2. The anti-aliasing performance of our polynomial filters with radius 1 is comparable with that of the Mitchell-Netravali filters, which has a radius 2 and negative lobes. This makes our polynomial filters with radius 1 very useful because, without sacrificing performance, smaller radius often saves computation and at the same time its non-negativity can avoid the problems of clipping and ringing artifact that may result from the negative lobes of other filters.

3. Filters using closed-form evaluation are more suitable for high-quality anti-aliasing. Using look-up tables or super-sampling always introduces random noise if the sizes of graphical objects are beyond their precision. [3]

References

[1] Yun-Peng Cheng, Kai-Yuan Zhang, and Zhong Xu. *Matrix Theory (in Chinese)*. Northwest Industrial Undersity Press, Xi'an, Shaanxi, China, 2000.

[3] This is a conclusion in [6].

[2] Tom Duff. Polygon scan conversion by exact convolution. *Raster Imaging and Digital Typography '89*, pages 154–168, 1989.

[3] A. E. Fabris and A. R. Forrest. Antialiasing of curves by discrete pre-filtering. In *SIGGRAPH 1997 Conference Proceedings*, Annual Conference Series, pages 317–323, August 1997.

[4] Brain Guenter and Jack Tumblin. Quadrature prefiltering for high quality antialiasing. *ACM Trans. on Graphics*, 15(4):332–353, 1996.

[5] J. Kajiya and M. Ullner. Filtering high quality text for display on raster scan devices. In *SIGGRAPH 1981 Conference Proceedings*, Annual Conference Series, pages 7–15, August 1981.

[6] Zhouchen Lin, Hai-Tao Chen, Heung-Yeung Shum, and Jian Wang. Pre-filtering 2d polygons without clipping. *Journal of Graphics Tools*, ??(??):??, 2004.

[7] M. D. McCool. Analytic antialiasing with prism splines. In *SIGGRAPH 1995 Conference Proceedings*, Annual Conference Series, pages 429–436, August 1995.

[8] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer graphics. In *SIGGRAPH 1988 Conference Proceedings*, Annual Conference Series, pages 221–227, August 1988.

[9] D. Slepian and H. O. Pollak. Prolate spheroidal wave functions, fourier analysis and uncertainty —1. *The Bell System Technical Journal*, 40:43–64, 1961.

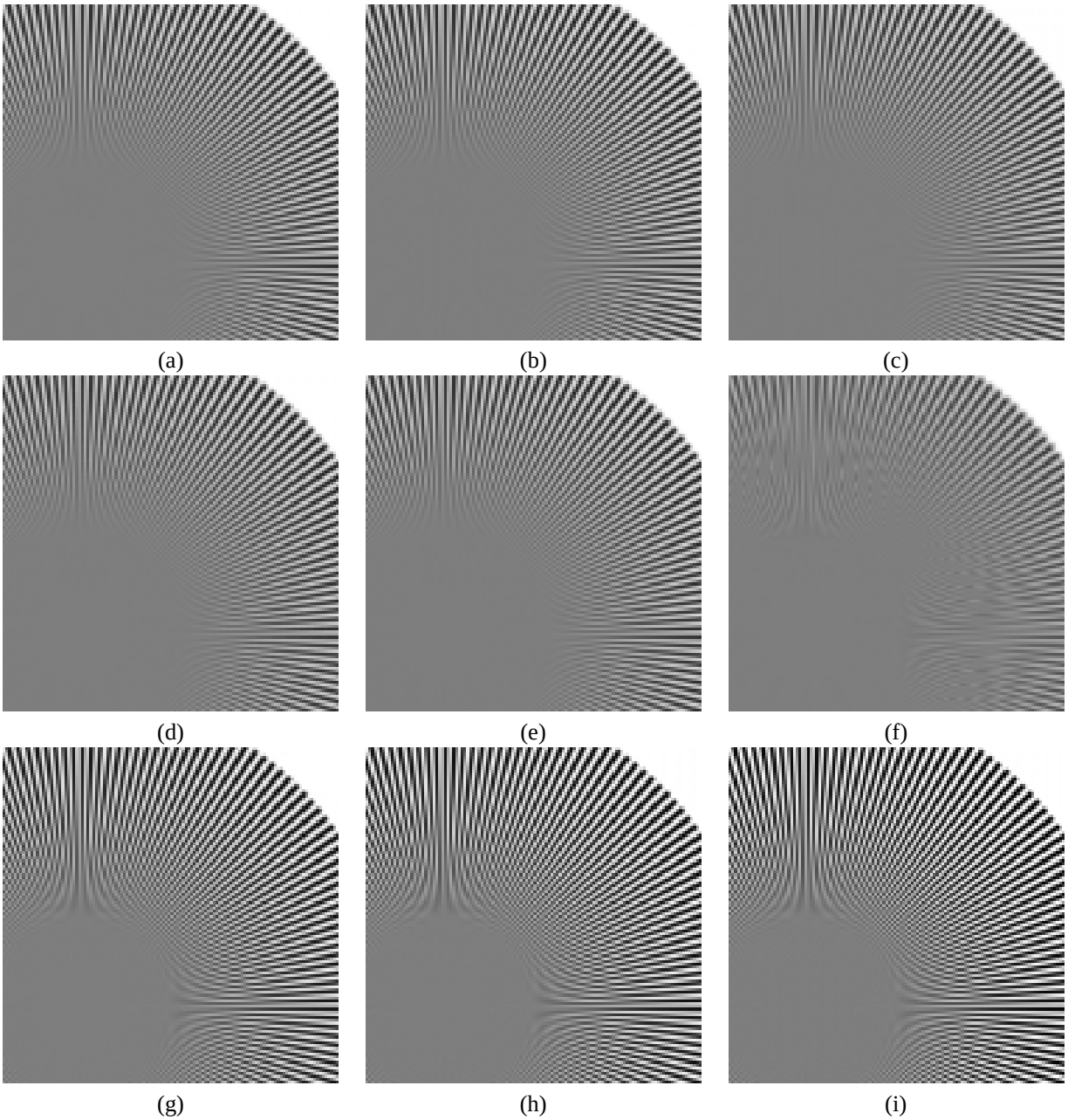


Figure 5: Closeups of downscaling Figure 3(a) by 1.8 times using various filters with radius 2. (a) Square polynomial filter. (b) Circular polynomial filter. (c) Square Gaussian filter with $\sigma = 0.9$. (d) Circular Gaussian filter with $\sigma = 0.9$. (e) Conical filter. (f) Box filter. (g) Mitchell-Netravali filter with $B = 1/3,\ C = 1/3$. (h) Mitchell-Netravali filter with $B = 0,\ C = 0.5$. (i) Mitchell-Netravali filter with $B = 0,\ C = 1$.