



# Design of Low-Artifact Interpolation Kernels by Means of Computer Algebra

Peter Karpov

Received: 22 September 2021 / Revised: 26 April 2022 / Accepted: 27 April 2022 / Published online: 24 September 2022  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

**Abstract** We present a number of new piecewise-polynomial kernels for image interpolation. The kernels are constructed by optimizing a measure of interpolation quality based on the magnitude of anisotropic artifacts. The kernel design process is performed symbolically using the Mathematica computer algebra system. An experimental evaluation involving 14 image quality assessment methods demonstrates that our results compare favorably with the existing linear interpolators.

**Keywords** Interpolation · Polynomial systems · Image processing

**Mathematics Subject Classification** 65D05

## 1 Introduction

The problem of image interpolation consists of reconstructing a function  $u(x, y)$ ,  $x, y \in \mathbb{R}$  that agrees with the known samples on a uniform square grid  $s(m, n)$ ,  $x, y \in \mathbb{Z}$ . One of the basic operations in image processing and computer graphics, interpolation is required when the sampling points do not match the original pixel grid, image upscaling and texture sampling being two prime examples. Implementations of various interpolation algorithms can be found both in software (image processing applications and libraries) and hardware (graphics processing units). The computer graphics industry demonstrates significant interest in high-quality, high-performance interpolation methods [18].

Linear interpolation is an important class of methods that reconstruct  $u(x, y)$  by convolving the image  $s(m, n)$  with an interpolation kernel  $\psi(x, y)$ :

$$u(x, y) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} s(m, n)\psi(x - m, y - n).$$

The kernel can be constructed as a product of two one-dimensional kernels:  $\psi(x, y) = \psi(x)\psi(y)$ . Such separable kernels are often preferred by virtue of their computational convenience since in this case interpolation can be

---

P. Karpov (✉)

Artificial Intelligence, Neurotechnology and Business Analytics Lab, Plekhanov Russian University of Economics, Stremyanny Lane 36, Moscow 117997, Russia  
e-mail: PeterKarpov@inversed.ru

performed in two one-dimensional steps along each axis. Many established methods such as bicubic [15] and B-spline interpolation [4] belong to this class. Non-separable kernels have also been studied in [26].

Typical criteria employed for the design of interpolation kernels include smoothness, an exact representation of Taylor expansion terms, similarity to the ideal low-pass filter in Fourier space, or performance for a particular model of the Fourier spectrum [23]. However, these properties only indirectly correlate with the perceived image quality. We explore another avenue by trying to directly quantify the perceived artifacts.

Linear interpolation methods produce several types of undesirable effects, the most notable of which are the following.

- **Blurriness.** Overly smooth transitions in areas where the sharp transitions were present in the original image, usually around object edges. This artifact type is especially noticeable for linear interpolation ( $\psi(x) = 1 - |x|$ ,  $|x| \leq 1$ ).
- **Ringing.** Oscillating kernels produce noticeable halos around hard edges.
- **Staircasing or blocking.** The square pixel lattice coupled with the kernel separation process introduces anisotropic effects. For example, the isolevel contours of diagonal edges on the interpolated image form meandering, staircase-like curves instead of straight lines.

Blurriness and ringing artifacts are inevitable within the framework of linear interpolation. Reducing blurriness typically increases ringing as the kernel becomes more oscillating in order to increase edge acuity. We focus on the third type of artifact, staircasing or blocking. The importance of contours in visual perception is universally acknowledged in the field of human vision [2, 24]. Image quality assessment methods also make heavy use of edge information in the form of gradients [17, 32] or phase congruency [37]. Minimizing the distortions of edge contours is therefore very important for high-quality image interpolation.

A number of edge-directed interpolation techniques have been proposed [5, 10, 12, 16, 31, 36, 39] in order to suppress the artifacts arising near sharp edges. While effective to varying degrees, these techniques are more complex and computationally expensive than linear interpolation as they include special steps in order to take edge directions into account. Four edge-directed interpolation methods benchmarked in [33] were slower than bicubic interpolation by factors ranging from 119 to 932.

The main contribution of this work is demonstrating that the staircase effect can be greatly reduced while staying within the linear interpolation framework. This goal is achieved by optimizing the kernel with respect to an appropriately defined quality metric. Most calculations are performed symbolically using the Mathematica computer algebra system [29].

## 2 Proposed Approach

We start by postulating a set of conditions that a good kernel should satisfy.

- **Interpolation:** in order to agree with the existing samples,  $\psi(x)$  must be zero at any integer  $x$  except at  $x = 0$  where  $\psi(0) = 1$ .
- **Continuity.**
- **Partition of unity:**  $\sum_{k=-\infty}^{+\infty} \psi(x - k) = 1$ .
- **Exact representation of the linear signal term:**  $\sum_{k=-\infty}^{+\infty} k\psi(x - k) = x$ .

These conditions can be further strengthened by requiring the continuity of derivatives or the exact representation of higher-order terms of the Taylor expansion of the underlying continuous signal. While these additional properties are considered desirable from a theoretical perspective, it will be shown in Sect. 3 that they do not necessarily translate to improved image quality in practice. We include the continuity of the first derivative as an optional constraint.

We have chosen a separable piecewise-polynomial kernel form for simplicity and computational efficiency. The kernels come in even and odd variants corresponding to integer and half-integer interval endpoints respectively:

$$\psi(x) = \sum_{j=0}^p c_{i,j} (|x| - i)^j, \quad i = \lfloor |x| + \Delta \rfloor,$$

**Table 1** The number of free variables for different kernels. 0 corresponds to a unique solution and dash denotes an overconstrained system

	Non-smooth			Smooth		
	$p = 2$	$p = 3$	$p = 4$	$p = 2$	$p = 3$	$p = 4$
$r = 1$	0	0	0	–	–	–
$r = 3/2$	0	0	1	–	–	0
$r = 2$	1	2	3	–	0	1
$r = 5/2$	1	2	4	–	0	2
$r = 3$	2	4	6	–	1	3

where  $\Delta$  is zero for even and  $1/2$  for odd kernels. The kernels are defined on the interval  $(-r, r)$  and are zero elsewhere. We shall denote our kernels with given  $r$  and  $p$  by  $K_{(r,p)}$  and those satisfying the additional smoothness constraint ( $C^1$ -continuity) by  $K_{(r,p)_S}$ .

The first two constraints translate to the following equations.

- Interpolation:  $c_{i,0} = \llbracket i = 0 \rrbracket$ , where  $\llbracket P \rrbracket$  is the Iverson bracket taking the value 1 if the statement  $P$  is true and zero otherwise. Since  $c_{i,0}$  are identical for all kernels satisfying the interpolation constraint, we shall omit their values when presenting  $c_{i,j}$  matrices.
- Continuity:  $\sum_{j=0}^p c_{i,j} = 0$  for any  $i$ .

The equations for the last two constraints are obtained by evaluating

$$\sum_{k=1-r}^r \psi(x - k) \quad \text{and} \quad \sum_{k=1-r}^r k\psi(x - k)$$

for  $x \in [0, 1)$  or  $x \in [0, 1/2)$  depending on the kernel type using the `Simplify` command and collecting the coefficients for each polynomial term. A general solution of the linear system incorporating all constraints can then be found by `Solve`. The number of free variables for different general solutions is reported in Table 1. The case  $r = 1$  has a unique solution corresponding to linear interpolation ( $\psi(x) = 1 - |x|$ ) for any  $p$  (this follows from the linear term condition alone). The unique solution  $K_{(2,3)_S}$  corresponds to Keys’ cubic kernel [15], and  $K_{(3/2,2)}$  to Dodgson’s kernel [8].

The general solutions for various values of  $r$  and  $p$  without the smoothness constraint are:

$K_{(2,2)}$  :

$$\begin{bmatrix} c_{0,2} \\ c_{1,1} \\ c_{1,2} \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,1} \end{bmatrix},$$

$K_{(3/2,2)}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,2} \\ c_{1,1} \\ c_{1,2} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 \\ -4 \\ -1 \\ 2 \end{bmatrix},$$

$K_{(2,3)}$  :

$$\begin{bmatrix} c_{0,3} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} -3 & -3 & -3 \\ -4 & -4 & -1 \\ 3 & 3 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,1} \\ c_{0,2} \end{bmatrix},$$

$K_{(3/2,4)}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,3} \\ c_{0,4} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{1,4} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -16 & -8 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 8 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \end{bmatrix},$$

$K_{(2,4)}$  :

$$\begin{bmatrix} c_{0,4} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{1,4} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} -3 & -3 & -3 & -3 \\ -5 & -5 & -2 & -1 \\ 6 & 6 & 3 & 3 \\ -4 & -4 & -4 & -5 \\ 3 & 3 & 3 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \end{bmatrix},$$

 $K_{(5/2,2)}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{1,1} \\ c_{1,2} \\ c_{2,1} \\ c_{2,2} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 0 \\ -6 & -2 \\ 4 & 0 \\ 2 & 1 \\ -4 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,1} \end{bmatrix},$$

 $K_{(3,2)}$  :

$$\begin{bmatrix} c_{0,2} \\ c_{1,2} \\ c_{2,1} \\ c_{2,2} \end{bmatrix} = \begin{bmatrix} -1 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,1} \\ c_{1,1} \end{bmatrix},$$

 $K_{(5/2,3)}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,3} \\ c_{1,2} \\ c_{1,3} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 0 & 0 \\ -24 & -8 & -16 \\ -1 & 0 & -2 \\ -4 & -2 & 0 \\ 12 & 4 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \\ c_{1,1} \end{bmatrix},$$

 $K_{(3,3)}$  :

$$\begin{bmatrix} c_{0,3} \\ c_{1,3} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} = \frac{1}{5} \begin{bmatrix} -5 & -5 & -5 & 0 & 0 \\ 0 & 0 & 0 & -5 & -5 \\ -7 & -7 & -2 & -6 & -1 \\ 1 & 1 & 1 & 3 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,1} \\ c_{0,2} \\ c_{1,1} \\ c_{1,2} \end{bmatrix},$$

 $K_{(5/2,4)}$  :

$$\begin{bmatrix} c_{0,3} \\ c_{0,3} \\ c_{1,3} \\ c_{1,4} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \\ c_{2,4} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -24 & -8 & -2 & -16 & 0 \\ 16 & 0 & 0 & 0 & -16 \\ -1 & 0 & 0 & -2 & 0 \\ 0 & -2 & 0 & 0 & -4 \\ 12 & 4 & 1 & 8 & 0 \\ -16 & 0 & -2 & 0 & 16 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \\ c_{0,4} \\ c_{1,1} \\ c_{1,2} \end{bmatrix}.$$

The general solutions with the smoothness constraint are:

 $K_{(2,3)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 \\ -5 \\ 3 \\ -1 \\ 2 \\ -1 \end{bmatrix},$$

 $K_{(2,4)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,3} \\ c_{0,4} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{1,4} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 0 \\ -7 & -4 \\ 5 & 2 \\ -1 & 0 \\ -3 & -2 \\ 9 & 4 \\ -5 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \end{bmatrix},$$

$K_{(3,3)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,3} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 0 \\ -4 & -4 \\ -12 & -4 \\ 19 & 6 \\ -7 & -2 \\ 5 & 2 \\ -10 & -4 \\ 5 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \end{bmatrix},$$

 $K_{(3,4)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,4} \\ c_{1,1} \\ c_{1,3} \\ c_{1,4} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \\ c_{2,4} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -16 & -8 & -4 & 0 \\ 41 & 20 & 10 & -8 \\ -25 & -12 & -6 & 4 \\ 7 & 4 & 2 & 0 \\ 15 & 8 & 6 & -4 \\ -51 & -28 & -18 & 0 \\ 29 & 16 & 10 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \\ c_{0,3} \\ c_{1,2} \end{bmatrix},$$

 $K_{(3/2,4)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{0,4} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{1,4} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 \\ -6 \\ 0 \\ 8 \\ -1 \\ 3 \\ 0 \\ -4 \end{bmatrix},$$

 $K_{(5/2,3)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \end{bmatrix} = \frac{1}{32} \begin{bmatrix} 0 \\ -56 \\ 0 \\ -18 \\ 32 \\ -8 \\ 1 \\ -4 \\ 4 \end{bmatrix},$$

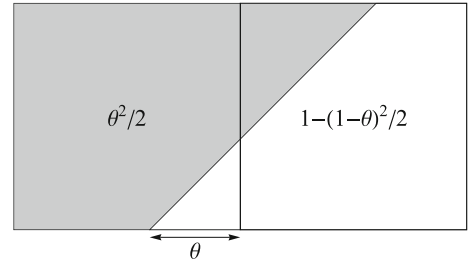
 $K_{(5/2,4)_S}$  :

$$\begin{bmatrix} c_{0,1} \\ c_{0,3} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ c_{1,4} \\ c_{2,1} \\ c_{2,2} \\ c_{2,3} \\ c_{2,4} \end{bmatrix} = \frac{1}{48} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -208 & -88 & -26 \\ 208 & 64 & 32 \\ 256 & 160 & 56 \\ -448 & -256 & -128 \\ 80 & 44 & 13 \\ -208 & -112 & -32 \\ -128 & -80 & -28 \\ 448 & 256 & 80 \end{bmatrix} \begin{bmatrix} 1 \\ c_{0,2} \\ c_{0,4} \end{bmatrix}.$$

The next step of our approach is optimizing the independent kernel coefficients with respect to an objective function measuring the severity of the staircasing effect. In order to define this function, we consider a sharp edge with a  $45^\circ$  orientation separating two half-spaces with values 0 and 1 (see Fig. 1). The pixel at location  $(i, j)$  in the corresponding rasterized image has one of the four different values that depend on the position of the grid:

$$d(i, j) = \begin{cases} 0 & i - j < -1, \\ \theta^2/2 & i - j = -1, \\ 1 - (1 - \theta)^2/2 & i - j = 0, \\ 1 & i - j > 0. \end{cases}$$

**Fig. 1** Rasterization of a diagonal edge



The resulting interpolant is

$$u(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} d(i, j)\psi(x - i, y - j).$$

A perfect staircasing-free interpolation should have straight isolines. A measure of staircasing penalizing isoline distortions was therefore introduced:

$$E_g^2(\theta) := \int \int (\nabla u(x, y) \cdot (1, 1))^2 dx dy$$

For a staircasing-free interpolation  $E_g = 0$  since  $\nabla u(x, y)$  is always orthogonal to edge direction. The integration is performed over unit squares with a nonzero gradient. The boundaries of the square regions correspond to piecewise intervals, i.e. integers for even kernels and half-integers for odd kernels:

$$[k - \Delta, k + 1 - \Delta] \times [0 - k - \Delta, 1 - k - \Delta] \cup [k + 1 - \Delta, k + 2 - \Delta] \times [0 - k - \Delta, 1 - k - \Delta],$$

where  $k$  is the integer region index and  $\Delta = 0$  for even kernels and  $\Delta = 1/2$  for odd ones. The compound region thus defined covers a single period of  $u(x, y)$ . The integration is performed separately for each square region after simplifying the piecewise interpolant into a polynomial. The degree of  $E_g^2$  polynomials is at most 4.

We have considered two options for the choice of  $\theta$ :

- $\theta = 1/2$ , corresponding to the worst case scenario. This choice produces the sharpest edge and the maximal value of  $E_g(\theta)$  for all kernels tested.
- Averaging across all values of  $\theta$ :  $\langle E_g \rangle^2 := \int_0^1 E_g^2(\theta) d\theta$ .

In all our experiments, the kernels obtained with  $E_g(1/2)$  and  $\langle E_g \rangle$  quality metrics are nearly identical, with the maximal absolute deviation between the two variants never exceeding 0.006. Therefore we chose the  $E_g(1/2)$  metric, as it leads to less complex algebraic manipulations.

It is worth noting that the normalized sinc kernel  $\text{sinc}(x) = \sin(\pi x)/(\pi x)$  has  $E_g(\theta) = 0$ . This can be seen by performing the summation along a single diagonal:

$$\sum_{k=-\infty}^{+\infty} \text{sinc}(x - k) \text{sinc}(y - k) = \text{sinc}(x - y).$$

Despite the absence of staircasing and theoretical optimality for band-limited signals, the sinc kernel is a poor choice for image interpolation. It is computationally inconvenient because of infinite support, and its prominent and slowly fading oscillations produce severe ringing artifacts.

We have also experimented with a simpler measure of staircasing, the squared deviation of the interpolant from  $1/2$  integrated along the edge:

$$E_d(\theta) = \int_0^1 (u(t, \theta + t) - 1/2)^2 dt.$$

A staircasing-free interpolation implies  $E_d = 0$  since the central isoline always has the value  $1/2$ . However, the converse is not true – the interpolant may have wavy isolines at values other than  $1/2$ . This measure thus performed poorly, often producing pathologically oscillating kernels.

The kernel coefficients are then optimized with respect to  $E_g(1/2)$ . We start by analytically differentiating  $E_g^2(1/2)$  with respect to free kernel coefficients in order to obtain the zero partial derivative conditions. The resulting systems of polynomial equations are solved with the `Solve` command. Internally, `Solve` employs cylindrical algebraic decomposition for dealing with such systems [30]. All the real critical points found by `Solve` are then checked using a second partial derivative test to find the local minima. The Hessian of the objective function is calculated and the `PositiveDefiniteMatrixQ` command is employed to verify its positive-definiteness. In all cases, a single real critical point that is also a local minimum has been found.

We have applied our optimization procedure to kernels with  $r$  ranging from  $3/2$  to  $3$  and  $p$  from  $2$  to  $4$ . The complexity of the stationary point equations increases with the number of free coefficients  $v$ . The maximal number of terms of a  $E_g$  polynomial is given by multinomial coefficient  $(v, 4)!$ , and the maximal number of terms in each of the corresponding equations is  $(v, 4)! - (v - 1, 4)!$ . These upper bounds are usually reached. The largest system we were able to solve symbolically corresponds to  $K_{(3,3)}$  and consists of 4 equations with 35 terms each. We were unable to obtain a symbolic solution for kernel  $K_{(3,4)}$  (6 equations, 84 terms each) in a reasonable time (8 h on a Xeon 6146-based workstation) and only report the numeric results for this instance. The obtained kernels are plotted in Fig. 2. Below we provide four exemplar polynomials along with the equations whose real roots correspond to the minima:nnnn

$$K_{(2,2)} : E_g^2(1/2) = (752 + 2611c_{0,1} + 3192c_{0,1}^2 + 1334c_{0,1}^3 + 196c_{0,1}^4)/1440,$$

$$2611 + 6384c_{0,1} + 4002c_{0,1}^2 + 784c_{0,1}^3 = 0;$$

$$K_{(2,4)_S} : E_g^2(1/2) = (9318135 + 7949688c_{0,2} + 3041872c_{0,2}^2 + 323456c_{0,2}^3 + 12544c_{0,2}^4)/33868800,$$

$$993711 + 760468c_{0,2} + 121296c_{0,2}^2 + 6272c_{0,2}^3 = 0;$$

$$K_{(3,2)} : E_g^2(1/2) = (4373 + 760c_{0,1}^4 + 5c_{0,1}^3(1051 + 276c_{1,1}) + 3c_{0,1}^2(4381 + 6c_{1,1}(405 + 62c_{1,1}))$$

$$+ 3c_{1,1}(2146 + c_{1,1}(1099 + 4c_{1,1}(53 + 13c_{1,1}))) + c_{0,1}(12826 + 3c_{1,1}(4176$$

$$+ c_{1,1}(1299 + 152c_{1,1}))) / 1440,$$

$$\left\{ \begin{array}{l} 12826 + 3040c_{0,1}^3 + 15c_{0,1}^2(1051 + 276c_{1,1}) + 6c_{0,1}(4381 + 6c_{1,1}(405 + 62c_{1,1})) + \\ 3c_{1,1}(4176 + c_{1,1}(1299 + 152c_{1,1})) = 0, \\ 1073 + 230c_{0,1}^3 + 3c_{0,1}^2(405 + 124c_{1,1}) + c_{1,1}(1099 + 2c_{1,1}(159 + 52c_{1,1})) + \\ 3c_{0,1}(696 + c_{1,1}(433 + 76c_{1,1})) = 0; \end{array} \right.$$

$$K_{(3,3)_S} : E_g^2(1/2) = (92669325 + 117493344c_{0,2} + 52220952c_{0,2}^2 + 9325760c_{0,2}^3 + 598096c_{0,2}^4)/25804800,$$

$$7343334 + 6527619c_{0,2} + 1748580c_{0,2}^2 + 149524c_{0,2}^3 = 0.$$

Since many other polynomials and their stationary points in the symbolic form are too cumbersome to reproduce here, we only list the selected numeric kernel coefficients. The coefficients of  $K_{(2,4)_S}$  admit a simple rational approximation resulting in a nearly identical kernel (maximal deviation  $1.2 \cdot 10^{-4}$ ).

$$K_{(2,2)} : \begin{bmatrix} -0.621913 & -0.378087 \\ -0.378087 & 0.378087 \end{bmatrix},$$

$$K_{(2,4)_S} : \begin{bmatrix} 0 & -1.751899 & 0.003798 & 0.748101 \\ -0.5 & 0.251899 & 0.996202 & -0.748101 \end{bmatrix} \approx \frac{1}{4} \begin{bmatrix} 0 & -7 & 0 & 3 \\ -2 & 1 & 4 & -3 \end{bmatrix},$$

$$K_{(5/2,3)} : \begin{bmatrix} 0 & -1.581352 & 0 \\ -0.825153 & 1 & 0.463315 \\ 0.162576 & -0.209324 & -0.231657 \end{bmatrix},$$

$$\begin{aligned}
K_{(3,3)} &: \begin{bmatrix} -0.435330 & -0.753337 & 0.188667 \\ -0.548062 & 0.379468 & 0.168595 \\ 0.092578 & 0.046312 & -0.138890 \end{bmatrix}, \\
K_{(3,3)_S} &: \begin{bmatrix} 0 & -2.067867 & 1.067867 \\ -0.932133 & 1.648200 & -0.716067 \\ 0.216067 & -0.432133 & 0.216067 \end{bmatrix}, \\
K_{(3,4)_S} &: \begin{bmatrix} 0 & -1.851913 & 0.542139 & 0.309774 \\ -0.838313 & 0.693843 & 0.958096 & -0.813626 \\ 0.169156 & 0.165539 & -0.838547 & 0.503852 \end{bmatrix}.
\end{aligned}$$

Some of the obtained kernels are nearly identical, namely

$$\begin{aligned}
|\psi_{(2,3)}(x) - \psi_{(2,2)}(x)| &< 2.7 \times 10^{-4}, \\
|\psi_{(2,4)}(x) - \psi_{(2,2)}(x)| &< 1.4 \times 10^{-3}, \\
|\psi_{(3,4)}(x) - \psi_{(3,3)}(x)| &< 6.7 \times 10^{-4}, \\
|\psi_{(5/2,4)}(x) - \psi_{(5/2,4)_S}(x)| &< 3.2 \times 10^{-3}.
\end{aligned}$$

### 3 Kernel Evaluation

In addition to our optimized kernels, we have also tested several interpolators proposed in the literature. Apart from the widely used cubic kernel  $K_{(2,3)_S}$ , Keys also derived a unique  $C^1$ -smooth piecewise-polynomial interpolator  $K_{S(3,3)}$  with higher interpolation order. Its  $c_{i,j}$  coefficients are

$$\frac{1}{12} \begin{bmatrix} 0 & -28 & 16 \\ -8 & 15 & -7 \\ 1 & -2 & 1 \end{bmatrix}.$$

The Lanczos kernel is the most popular member of the windowed sinc family of interpolators:

$$Ls_r(x) = \text{sinc}(x) \text{sinc}(x/r) \llbracket |x| < r \rrbracket$$

According to Turkowski [27], it provides the best compromise between sharpness and ringing among several tested windowed sinc filters. We have tested the commonly employed values  $r = 2$  and  $r = 3$ . The Lanczos kernel does not satisfy the partition of unity condition. This resulting ripple is noticeable for  $r = 2$  (maximal deviation from unity 0.019) but is tolerable for  $r = 3$  (maximal deviation 0.0057). In the case of 2D kernel  $Ls_r(x)Ls_r(y)$ , the deviation increases by a factor of nearly two.

Lagrange interpolation is a classical method that interpolates the given data with a polynomial of the lowest possible degree. When applied globally, it is susceptible to large oscillations. However, it can be applied locally to  $2r$  points around the current  $x$ . For uniformly-spaced data, this technique is equivalent to convolution with a piecewise-polynomial kernel. In the case of integer  $r$ , the kernel can be found by noting that a subpolynomial on the interval  $[i, i + 1)$  must evaluate to  $\delta(x)$  for integer  $x$ ,  $i - r + 1 \leq x \leq i + r$ . The coefficients for  $r = 2$  and  $r = 3$  are

$$Lg_{(2,3)}(x) : c_{i,j} = \frac{1}{6} \begin{bmatrix} -3 & -6 & 3 \\ -2 & 3 & -1 \end{bmatrix},$$

$$Lg_{(3,5)}(x) : c_{i,j} = \frac{1}{120} \begin{bmatrix} -40 & -150 & 50 & 30 & -10 \\ -60 & 80 & -5 & -20 & 5 \\ 6 & -5 & -5 & 5 & 1 \end{bmatrix}.$$



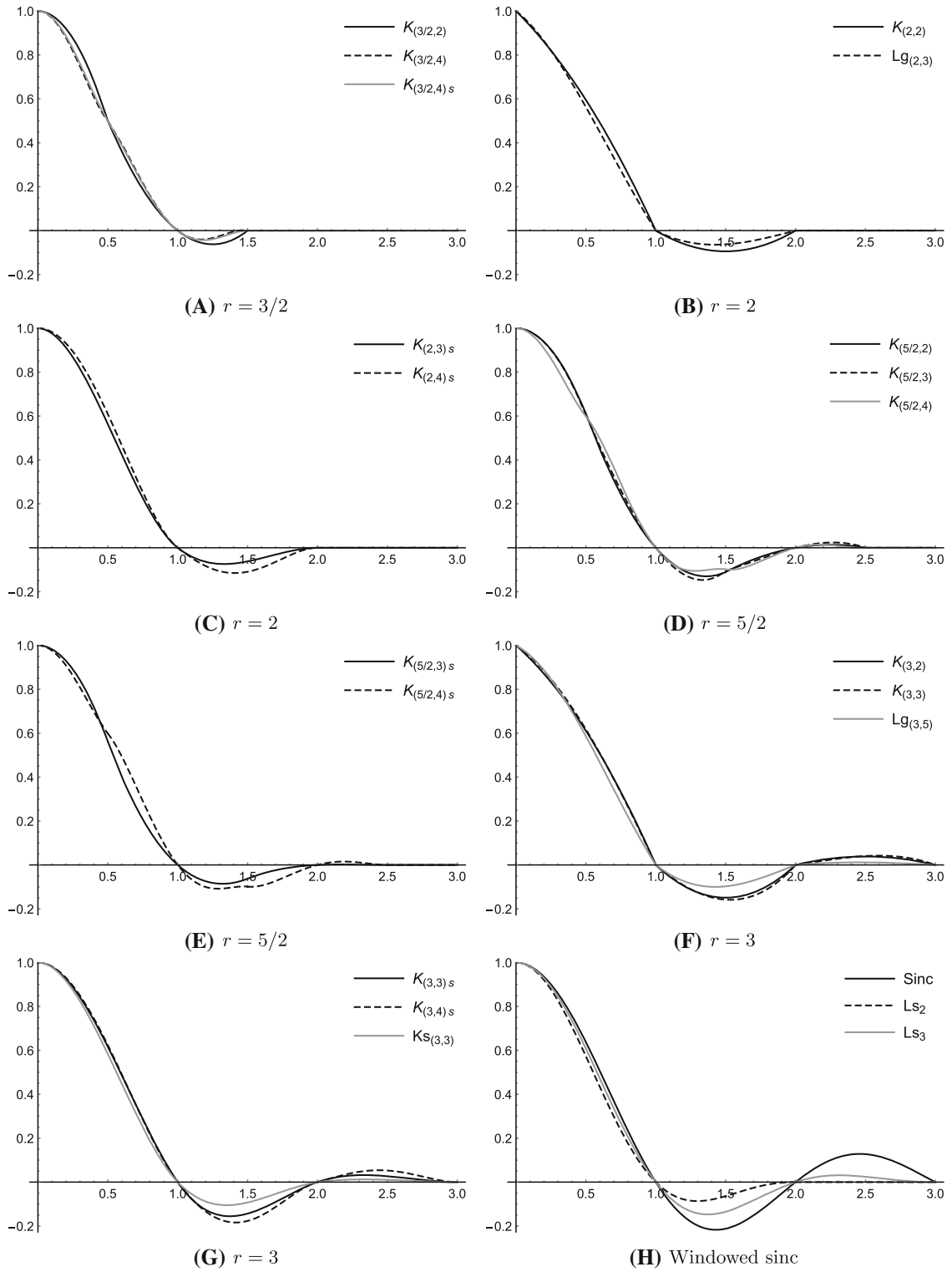


Fig. 2 Plots of interpolation kernels

Lagrange kernels are  $C^0$ -smooth in case of an integer  $r$  and discontinuous in case of a half-integer  $r$  (the latter thus being excluded from the tests). They converge to  $\text{sinc}(x)$  as  $r \rightarrow +\infty$  (see [19] for the proof), optimal in the low frequency region of the spectrum [23] and have the minimal support for a given interpolation order among the functions satisfying the interpolation constraints [3].

Schaum studied the performance of interpolators for different models of power spectrum  $|f(v)|^2$  [23]. For  $|f(v)|^2 \sim 1/v^4$ , the optimal interpolator supported on  $[-2, 2]$  is a piecewise-polynomial kernel

$$S_{C(2,3)}(x) = \frac{1}{15} \begin{cases} 3(1 - |x|)(5 + 4|x| - 5|x|^2) & 0 \leq |x| < 1, \\ (2 - |x|)(1 - |x|)(12 - 5|x|) & 1 \leq |x| < 2, \\ 0 & |x| \geq 2. \end{cases}$$

$S_{C(2,3)}(x)$  satisfies the partition of unity and linear term representation constraints and is  $C^0$ -smooth.

B-splines are piecewise-polynomial functions defined recursively as

$$\beta_0(x) = \begin{cases} 1 & |x| < 1/2 \\ 1/2 & |x| = 1/2, \\ 0 & |x| > 1/2 \end{cases}, \quad \beta_{p+1} = \beta_p * \beta_0,$$

where  $*$  denotes the convolution operator. For  $p > 1$ ,  $\beta_p$  are non-interpolating, so an additional prefiltering step is required to satisfy the interpolation condition (see [4,9] for details). The prefilter can be combined with  $\beta_p$  to get the actual interpolation kernel  $\beta_p^*$ . In particular, it can be shown (see [6]) that

$$\beta_2^*(x) = \sum_{k=-\infty}^{+\infty} \sqrt{2}(2\sqrt{2} - 3)^{|k|} \beta_2(x - k),$$

$$\beta_3^*(x) = \sum_{k=-\infty}^{+\infty} \sqrt{3}(\sqrt{3} - 2)^{|k|} \beta_3(x - k).$$

As the spline order increases,  $\beta_p^*(x)$  converges to  $\text{sinc}(x)$ . B-spline interpolation is generally considered to be one of the highest quality linear methods, but the fact that the kernel is not compactly supported complicates the implementation.

The Mitchell–Netravali kernel introduced in [20] is given by

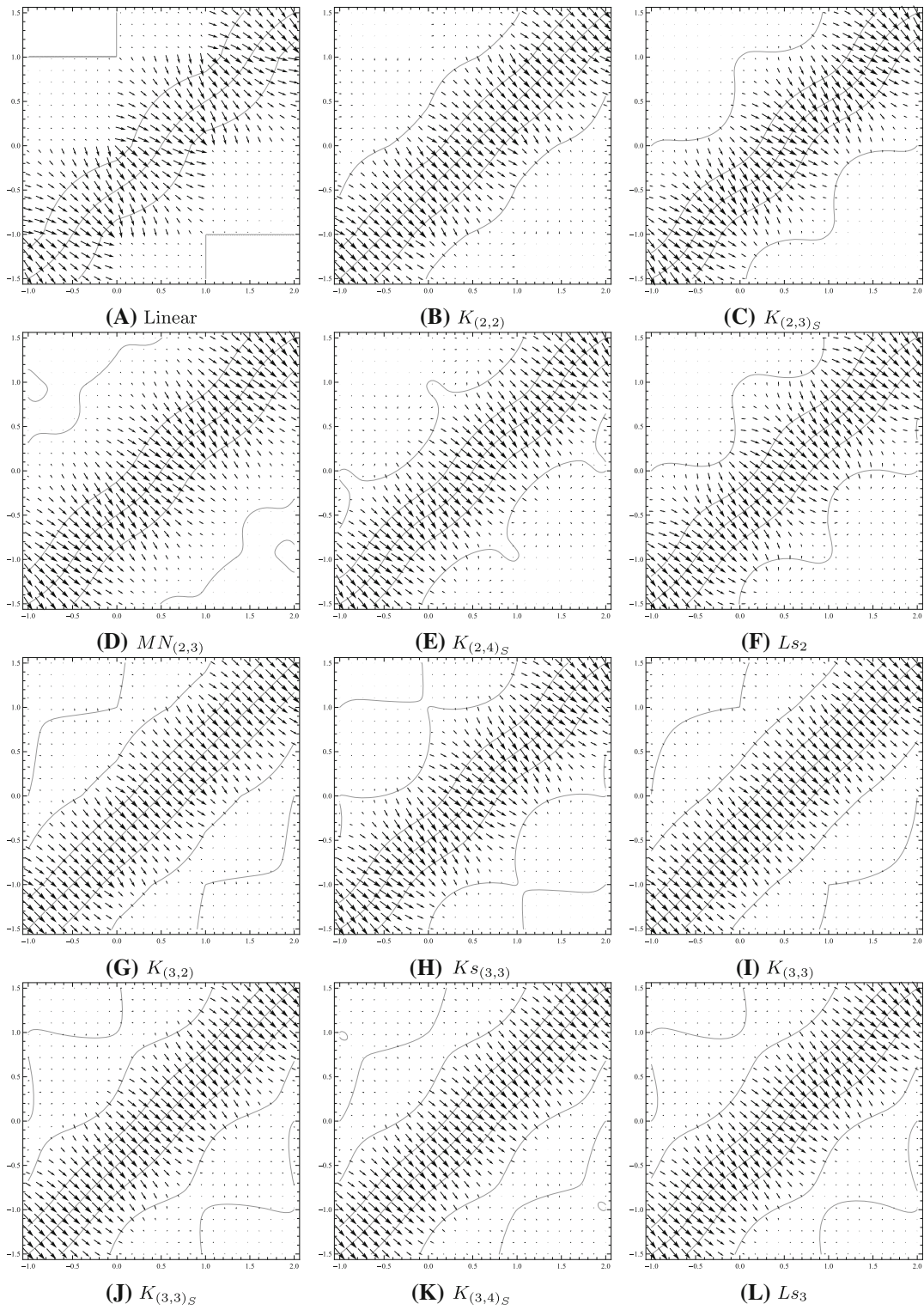
$$MN_{(2,3)}(x) = \frac{1}{18} \begin{cases} 16 - 36|x|^2 + 21|x|^3 & 0 \leq |x| < 1, \\ 32 - 60|x| + 36|x|^2 - 7|x|^3 & 1 \leq |x| < 2, \\ 0 & |x| \geq 2. \end{cases}$$

It is a linear combination of  $\beta_3(x)$  and  $K_{(2,3)_S}$  with weights  $1/3$  and  $2/3$  respectively. Unlike all other kernels in our tests,  $MN_{(2,3)}(x)$  is non-interpolating. This property makes it a poor choice if the target sample rate is close to that of the original image. Nonetheless, we have included it in the comparison, as it was identified in [20] as the optimal  $C^1$ -smooth cubic kernel supported on  $[-2, 2]$  interval in terms of perceived image quality.

To evaluate the performance of the optimized kernels on image features of various sizes and orientations we employed a zone plate function given by

$$I(x, y) = (1 + \cos(2\pi F(x^2 + y^2)))/2, \quad F = 6.$$

$I(x, y)$  was sampled in the region  $[0, 1] \times [0, 1]$  with sampling interval  $\Delta x = \Delta y = 1/30$  and then resampled with  $\Delta x = \Delta y = 1/360$  using various kernels. The resulting images are reproduced in Appendix 5. Table 2 lists the interpolation errors and the staircasing metrics of the new and existing kernels. Our results compare favorably with those of the existing kernels for  $r \geq 2$ . The optimized interpolators outperform the popular Keys' kernels with the same  $r$  in terms of both the staircasing magnitude and RMSE even at lower polynomial orders. The reduction of anisotropy is also apparent in the plots of the gradients of interpolant  $u(x, y)$  (Figs. 3 and 4). Odd kernels are

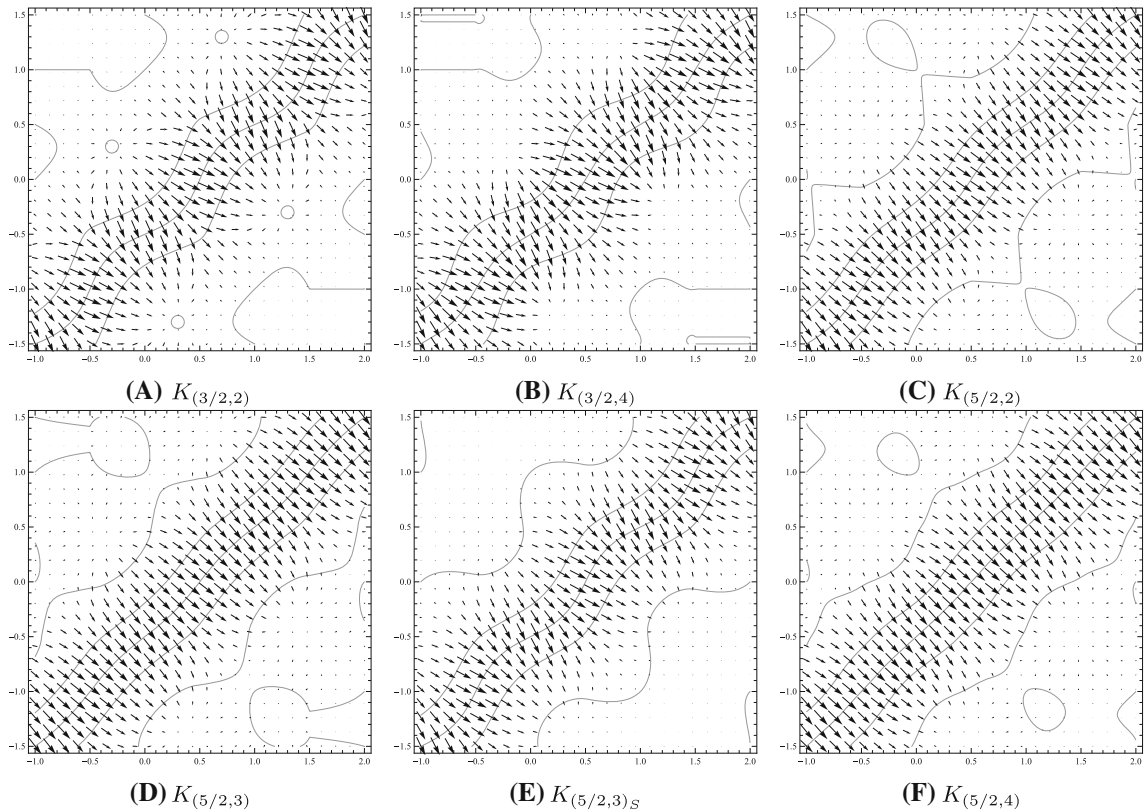


**Fig. 3** Gradients of  $u(x, y)$  for even kernels. Isolines of  $u(x, y)$  at levels 0, 1/4, 1/2, 3/4, 1 are shown in gray

**Table 2** Staircasing metrics  $E_g(1/2)$  and zone plate root mean square interpolation errors of various kernels

Kernel	$E_g(1/2)$	RMSE	Kernel	$E_g(1/2)$	RMSE
Linear	0.368	$1.26 \times 10^{-1}$	$K_{(5/2,3)}$	0.300	<b><math>4.48 \times 10^{-2}</math></b>
$K_{(3/2,2)}$	0.480	<b><math>1.04 \times 10^{-1}</math></b>	$K_{(5/2,3)_S}$	0.378	$7.68 \times 10^{-2}$
$K_{(3/2,4)}$	0.428	$1.14 \times 10^{-1}$	$K_{(5/2,4)}$	0.262	$5.16 \times 10^{-2}$
$K_{(3/2,4)_S}$	0.429	$1.12 \times 10^{-1}$	$K_{(5/2,4)_S}$	0.263	$5.12 \times 10^{-2}$
$K_{(2,2)}$	<b>0.222</b>	<b><math>5.98 \times 10^{-2}</math></b>	$K_{(3,2)}$	<b>0.185</b>	<b><math>3.33 \times 10^{-2}</math></b>
$K_{(2,3)}$	0.222	$5.98 \times 10^{-2}$	$K_{(3,3)}$	<b>0.172</b>	<b><math>2.82 \times 10^{-2}</math></b>
$Lg_{(2,3)}$	0.265	$7.84 \times 10^{-2}$	$K_{(3,3)_S}$	0.240	$3.18 \times 10^{-2}$
$Sc_{(2,3)}$	0.278	$6.86 \times 10^{-2}$	$Ks_{(3,3)}$	0.285	$5.76 \times 10^{-2}$
$K_{(2,3)_S}$	0.339	$7.72 \times 10^{-2}$	$K_{(3,4)}$	0.172	$2.83 \times 10^{-2}$
$MN_{(2,3)}$	<b>0.209</b>	$1.09 \times 10^{-1}$	$K_{(3,4)_S}$	0.223	<b><math>2.35 \times 10^{-2}</math></b>
$K_{(2,4)}$	0.222	$6.00 \times 10^{-2}$	$Lg_{(3,5)}$	0.233	$5.62 \times 10^{-2}$
$K_{(2,4)_S}$	0.303	<b><math>5.33 \times 10^{-2}</math></b>	$Ls_3$	0.254	$3.58 \times 10^{-2}$
$Ls_2$	0.368	$7.29 \times 10^{-2}$	$\beta_2^*$	0.313	$5.43 \times 10^{-2}$
$K_{(5/2,2)}$	0.316	<b><math>5.04 \times 10^{-2}</math></b>	$\beta_3^*$	0.236	$3.70 \times 10^{-2}$

Values improving upon the results of kernels with smaller  $r$  or identical  $r$  and smaller  $p$  are shown in bold



**Fig. 4** Gradients of  $u(x, y)$  for odd kernels. Isolines of  $u(x, y)$  at levels 0, 1/4, 1/2, 3/4, 1 are shown in gray

less effective at reducing staircasing even when compared to even kernels with smaller support. The  $E_g$  data offers some justification for the Mitchell–Netravali kernel, as it has the lowest staircasing among the kernels of its size. However,  $K_{(2,2)}$  has only 6% higher  $E_g$  while being interpolating.

We have also performed resampling tests on twelve benchmark images from the collection [1] (“apples”, “billiard balls a”, “cards a”, “coins”, “ducks”, “flowers”, “keyboard a”, “lion”, “garden table”, “tomatoes b”, “tools b”, “wood game”). The reduced  $300 \times 300$  images provided as a part of the image set were resampled to their original size ( $2400 \times 2400$ ) and compared with the ground truth. Since the root mean square error (RMSE) is not a reliable indicator of perceived image quality, we employed 12 additional full-reference image quality assessment (IQA) methods implemented in the PIQ library [14]: SSIM [40], MS-SSIM [28], VIFp [25], FSIM [37], GMSD [32], VSI [35], HaarPSI [22], MDSI [41], MS-GMSD [34], LPIPS [38], PieAPP [21], DISTIS [7]. In order to specifically assess the reconstruction of gradients, we have added gradient cosine similarity (GCS) to the set of IQA methods. The gradients were calculated using the maximally isotropic Scharr operator [11]:

$$g_x = \frac{1}{2(2 + \sqrt{12})} \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{12} & 0 & \sqrt{12} \\ -1 & 0 & 1 \end{bmatrix}, g_y = g_x^\top$$

The gradient cosine similarity was then calculated as

$$GCS = \frac{\sum_i \sum_j G_{i,j} \cdot G'_{i,j}}{\sqrt{\sum_i \sum_j G_{i,j}^2} \sqrt{\sum_i \sum_j G'^2_{i,j}}},$$

where  $G_{i,j}$  and  $G'_{i,j}$  are the per-pixel gradients of the original and the iterpolated images.

Since different IQA methods have different scales, the results of each method were rescaled into the [0, 100] range with 0 corresponding to the worst interpolation kernel and 100 to the ground truth image. The resulting standardized quality scores averaged across all images are listed in Table 3.

The kernels ranked best by various IQA methods are  $K_{(3,4)_S}$  (MS-SSIM, GMSD, MS-GMSD, HaarPSI, GCS),  $K_{(3,3)_S}$  (RMSE, FSIM, MDSI),  $\beta_3$  (SSIM, LPIPS),  $\beta_2$  (VIF),  $K_{(2,4)_S}$  (VSI),  $L_{S3}$  (DISTIS),  $L_{S2}$  (PieAPP). The only case where our optimized kernels demonstrate no improvement is  $r = 3/2$ . For larger values of  $r$ , the new kernels  $K_{(2,4)_S}$ ,  $K_{(3,3)_S}$  and  $K_{(3,4)_S}$  outperform the existing interpolators with identical support according to the vast majority of IQA methods. The kernel  $K_{(5/2,3)}$  compares favorably even with the larger Keys, Lagrange, and Lanczos interpolators. Remarkably, the kernels  $K_{(5/2,3)}$ ,  $K_{(3,3)_S}$  and  $K_{(3,4)_S}$  outperform the significantly more costly cubic B-spline interpolation according to the majority of quality metrics (9, 10 and 10 metrics respectively). Increasing the support size and polynomial degree gives diminishing returns, so the potential improvements arising from kernels with  $r > 3$  or  $p > 4$  are likely marginal.

As Fig. 5 demonstrates, the new kernels with small  $E_g$  provide a noticeable reduction of staircasing around diagonal edges when compared to the popular Keys’ interpolators. However, for a given support size the kernels with the smallest  $E_g$  are usually not the ones preferred by IQA methods (the mean correlation between  $E_g$  and IQA scores is  $-0.54$ ). There are two reasons for this inconsistency. First, optimization of  $E_g$  alone does not take into account image sharpness. Even though  $K_{(3/2,2)}$  has the worst  $E_g$  among the tested kernels, it ranks above the linear interpolation according to all but one IQA method by virtue of producing sharper images. Second, the IQA methods themselves are imperfect and usually are not specifically designed for the distortion types introduced by interpolation.

## 4 Conclusion

We have constructed several new high-quality separable piecewise-polynomial interpolation kernels for image resampling. The kernel coefficients were obtained by minimizing a specifically defined measure of the magnitude of staircasing artifacts around diagonal edges. By using the Mathematica computer algebra system we were able to evaluate the resulting polynomials in symbolic form. This is an important prerequisite for the optimization process,

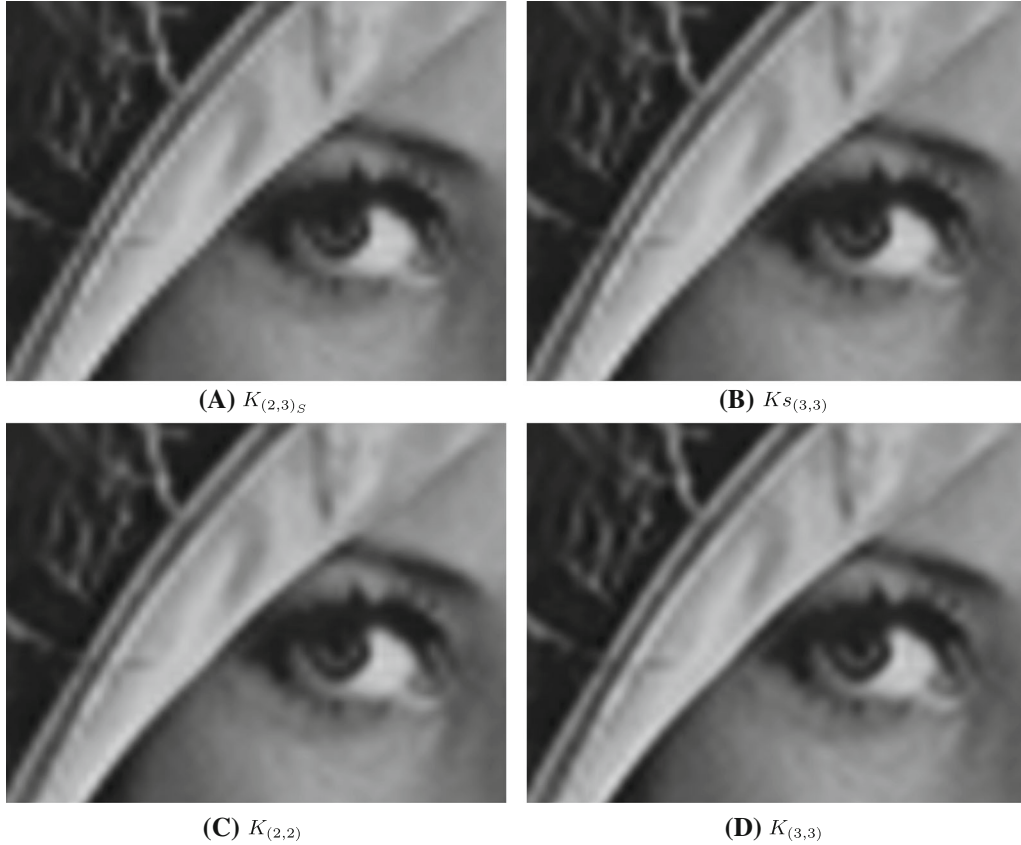
**Table 3** Averaged standardized quality scores for various kernels and IQA methods

	NN	Lin.	$K_{(3/2,2)}$	$K_{(3/2,4)}$	$K_{(3/2,4)_S}$	$K_{(2,2)}$	$Lg_{(2,3)}$	$Sc_{(2,3)}$	$K_{(2,3)_S}$
RMSE	0.00	22.58	<b>29.60</b>	26.30	26.94	35.04	32.53	34.67	33.84
SSIM	0.00	45.62	<b>49.27</b>	47.27	47.72	50.79	50.77	51.81	51.78
MS-SSIM	0.00	27.10	<b>39.13</b>	33.56	34.67	42.26	40.44	43.53	43.31
FSIM	34.79	0.00	<b>36.72</b>	20.40	23.53	65.54	52.87	61.22	56.55
GMSD	0.00	45.11	<b>48.84</b>	47.18	47.51	50.71	49.95	51.18	51.01
MS-GMSD	0.00	43.35	<b>48.80</b>	46.35	46.84	50.93	49.88	51.54	51.35
VIF	0.00	35.17	<b>35.06</b>	34.66	34.81	35.37	36.68	36.59	36.66
VSI	29.11	0.00	<b>29.51</b>	15.54	18.37	57.59	46.77	53.45	48.50
HaarPSI	0.00	56.87	<b>59.21</b>	57.88	58.17	62.41	61.70	62.49	61.97
MDSI	9.11	0.00	<b>10.15</b>	5.27	6.15	21.09	15.74	19.26	17.31
LPIPS	0.00	38.95	<b>40.70</b>	40.06	40.26	41.79	41.93	42.50	42.52
PieAPP	4.02	32.07	<b>45.13</b>	40.60	41.17	37.60	35.67	39.46	42.50
DISTS	0.00	56.49	<b>59.93</b>	58.84	58.96	62.89	61.03	62.19	61.74
GCS	0.00	55.04	<b>57.32</b>	55.61	56.09	59.83	59.79	60.64	60.34
	$MN_{(2,3)}$	$K_{(2,4)_S}$	$Ls_2$	$K_{(5/2,2)}$	$K_{(5/2,3)}$	$K_{(5/2,3)_S}$	$K_{(5/2,4)_S}$	$K_{(3,2)}$	$K_{(3,3)}$
RMSE	25.91	<b>37.58</b>	25.02	37.73	<b>38.02</b>	34.36	36.73	35.94	36.87
SSIM	47.79	<b>52.38</b>	35.25	<b>52.84</b>	52.84	52.05	52.00	51.32	51.57
MS-SSIM	31.28	<b>46.97</b>	31.41	47.69	<b>47.75</b>	44.55	45.26	44.01	45.05
FSIM	19.37	<b>71.58</b>	54.85	71.36	<b>72.37</b>	58.25	69.36	67.70	71.62
GMSD	46.41	<b>52.67</b>	50.76	53.06	<b>53.31</b>	51.33	52.34	52.06	52.51
MS-GMSD	45.18	<b>53.56</b>	51.53	54.09	<b>54.37</b>	51.88	52.98	52.76	53.36
VIF	<b>36.75</b>	35.53	18.87	36.12	36.05	36.51	35.82	35.85	35.44
VSI	16.36	<b>62.02</b>	24.18	58.89	58.99	48.70	57.74	48.35	51.57
HaarPSI	58.83	<b>63.26</b>	59.07	63.67	<b>63.95</b>	62.04	63.42	63.59	63.78
MDSI	4.85	<b>24.27</b>	16.09	24.27	<b>24.85</b>	18.05	23.15	22.52	24.53
LPIPS	39.77	<b>43.08</b>	24.94	<b>43.32</b>	43.27	42.64	42.81	42.40	42.34
PieAPP	27.37	46.80	<b>49.75</b>	46.93	46.51	45.99	42.96	41.57	42.84
DISTS	57.47	<b>64.84</b>	57.93	64.68	<b>65.00</b>	62.22	64.33	64.20	64.63
GCS	58.09	<b>61.15</b>	51.72	61.63	<b>61.78</b>	60.34	60.87	60.57	60.86
	$K_{(3,3)_S}$	$Ks_{(3,3)}$	$K_{(3,4)_S}$	$Lg_{(3,5)}$	$Ls_3$	$\beta_2^*$	$\beta_3^*$		
RMSE	<b>38.20</b>	35.86	38.12	35.22	34.05	36.45	37.21		
SSIM	<b>52.91</b>	52.71	52.70	52.03	51.25	53.14	<b>53.20</b>		
MS-SSIM	47.67	45.66	<b>47.81</b>	44.06	46.28	46.96	47.15		
FSIM	<b>73.49</b>	64.31	71.15	63.70	71.94	65.22	67.88		
GMSD	53.55	52.27	<b>54.09</b>	51.58	53.55	52.74	53.23		
MS-GMSD	54.64	52.96	<b>55.34</b>	52.06	54.62	53.64	54.19		
VIF	36.22	<b>37.15</b>	36.02	36.93	33.22	<b>37.26</b>	37.21		
VSI	56.14	50.74	50.45	50.38	39.71	51.90	53.40		
HaarPSI	64.39	63.38	<b>64.72</b>	63.11	64.00	63.63	64.29		

**Table 3** continued

	$K_{(3,3)_S}$	$K_{S(3,3)}$	$K_{(3,4)_S}$	$Lg_{(3,5)}$	$Ls_3$	$\beta_2^*$	$\beta_3^*$
MDSI	<b>25.59</b>	20.77	24.52	20.42	24.93	21.26	22.62
LPIPS	<b>43.32</b>	43.16	43.30	42.68	39.27	43.48	<b>43.58</b>
PieAPP	46.09	41.91	48.15	38.35	46.08	45.18	42.99
DISTS	64.97	63.03	65.88	62.69	<b>66.93</b>	63.60	64.14
GCS	61.91	61.46	<b>62.07</b>	60.97	61.00	61.69	62.04

Near-duplicate kernels have been excluded. The best results among the kernels with support size  $r$  or less are shown in bold for each  $r > 1$

**Fig. 5** A test image interpolated with Keys' (a, b) and minimal  $E_g$  (c, d) kernels

since the gradient of the objective function can be easily calculated. In most cases, we were able to find the stationary points and obtain the optimal kernel coefficients also in symbolic form. Although the optimization could have been performed numerically using Newton's method, a symbolic approach had the advantage of proving the solution's uniqueness.

The reduction of staircasing comes at a cost of increased kernel oscillations. Nonetheless, when compared to other popular interpolating kernels our results provide a noticeable improvement of subjective image quality in areas around sharp transitions. Depending on the desired computational cost and subjective preferences between sharpness and blocking, we recommend selecting a kernel from the following set:  $K_{(2,2)}$ ,  $K_{(2,4)_S}$ ,  $K_{(5/2,3)}$ ,  $K_{(3,3)}$ ,  $K_{(3,3)_S}$ , and  $K_{(3,4)_S}$ .

We note the discrepancy between theory and practice. By the standards of interpolation theory, our kernels are inferior to many other piecewise-polynomial interpolators proposed in the literature as they have low interpolation order and are not necessarily continuously differentiable. Nonetheless, they demonstrate superior performance both subjectively and according to various image quality assessment methods. We conclude that the theoretical considerations pertaining to one-dimensional interpolation are insufficient for the design of high-quality image resampling kernels. The often neglected anisotropic artifacts arising from the kernel separation process are a major factor determining the subjective image quality.

Optimization with respect to a single type of artifact is a limitation of the present work. A more complex objective function incorporating other artifact types could further improve the subjective image quality. The use of separable kernels is another limitation. With nonseparable kernels, the staircasing error metrics could be reduced further, but we decided not to pursue this direction for two reasons. First, it would greatly increase the number of coefficients and would lead to reduced speed and higher complexity. Second, the approach would introduce further complications since the vertical and horizontal edges would no longer be staircasing-free.

Since our artifact reduction technique stays within the linear interpolation framework, it retains the simplicity and computational efficiency of the linear methods. High-quality texture filtering on modern graphical processors is therefore a possible application. The use of the proposed kernels as a basis of more complex nonlinear methods is a promising direction for future work.

All Mathematica code used for kernel construction can be found on the author's GitHub page [13].

**Acknowledgements** We would like to thank Timur Sadykov for helpful suggestions made during the preparation of this manuscript.

## 5 Interpolated Zone Plate Images

Figs. 6 and 7 demonstrate the function  $I(x, y)$  interpolated with various kernels.



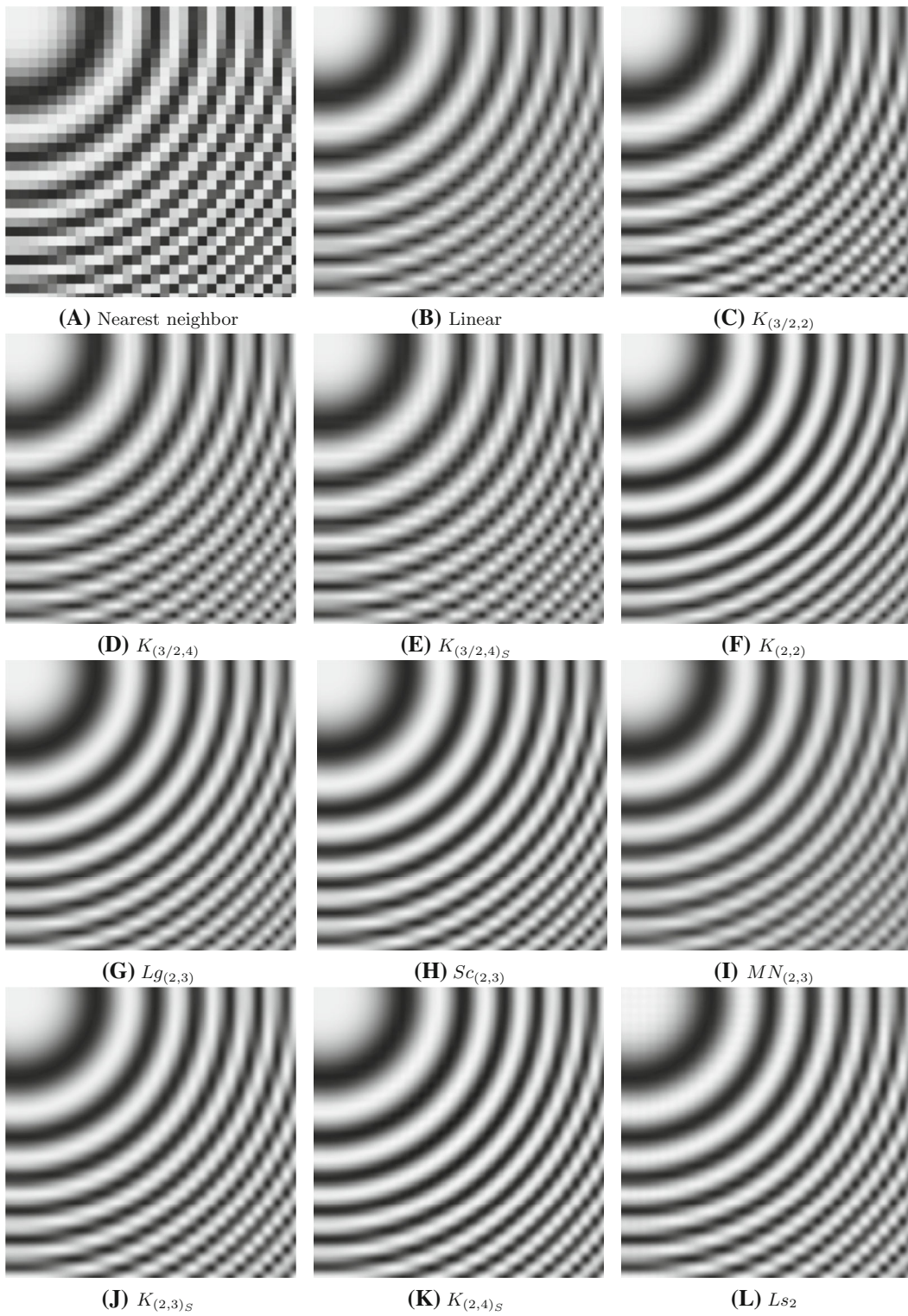
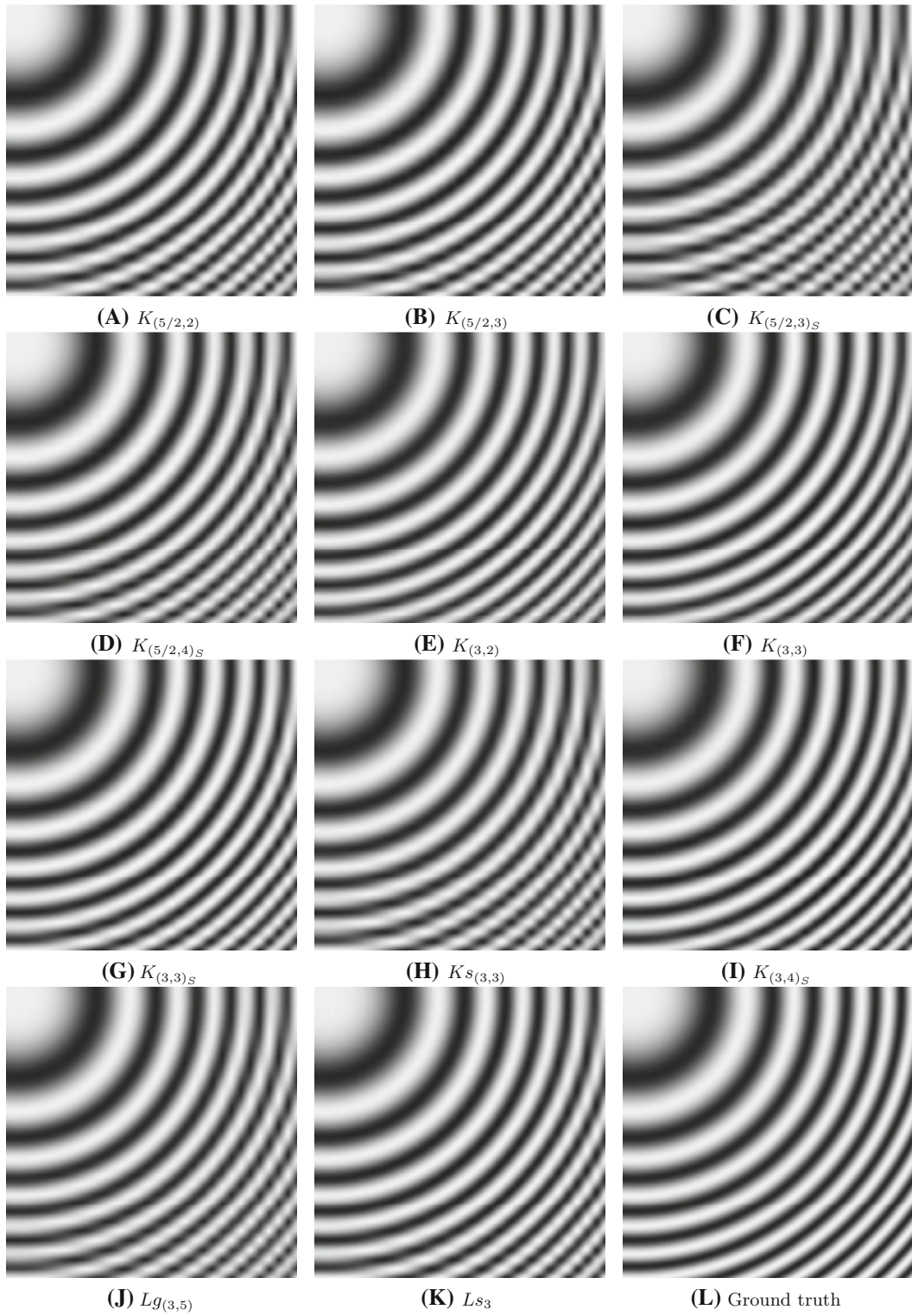


Fig. 6 Interpolated zone plate image,  $r = 1..2$



**Fig. 7** Interpolated zone plate image,  $r = 5/2..3$

## References

1. Asuni, N., Giachetti, A.: Testimages: a large data archive for display and algorithm testing. *J. Gr. Tools* **17**(4), 113–125 (2013). <https://doi.org/10.1080/2165347X.2015.1024298>
2. Bansal, A., Kowdle, A., Parikh, D., Gallagher, A., Zitnick, L.: Which edges matter? In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops (2013)
3. Blu, T., Thevenaz, P., Unser, M.: Moms: maximal-order interpolation of minimal support. *IEEE Trans. Image Process.* **10**(7), 1069–1080 (2001). <https://doi.org/10.1109/83.931101>
4. Briand, T., Monasse, P.: Theory and practice of image b-spline interpolation. *Image Process. Line* **8**, 99–141 (2018). <https://doi.org/10.5201/ipol.2018.221>
5. Cha, Y., Kim, S.: The error-amended sharp edge (ease) scheme for image zooming. *IEEE Trans. Image Process.* **16**(6), 1496–1505 (2007). <https://doi.org/10.1109/TIP.2007.896645>
6. Champagnat, F., Sant, Y.L.: Efficient cubic b-spline image interpolation on a GPU. *J. Graph. Tools* **16**, 218–232 (2012)
7. Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: unifying structure and texture similarity. *IEEE Trans. Patt. Anal. Mach. Intell.* **44**, 2567–2581 (2020). <https://doi.org/10.1109/TPAMI.2020.3045810>
8. Dodgson, N.A.: Quadratic interpolation for image resampling. *IEEE Trans. Image Process.* **6**(9), 1322–1326 (1997). <https://doi.org/10.1109/83.623195>
9. Getreuer, P.: Linear methods for image interpolation. *Image Proces. Line* **1**, 238–259 (2011). [https://doi.org/10.5201/ipol.2011.g\\_lmii](https://doi.org/10.5201/ipol.2011.g_lmii)
10. Giachetti, A., Asuni, N.: Real time artifact-free image upscaling. *Image Process. IEEE Trans.* **20**(10), 2760–2768 (2011). <https://doi.org/10.1109/TIP.2011.2136352>
11. Jähne, B., Scharr, H., Körkel, S., Jähne, B., Haußecker, H., Geißler, P.: Principles of Filter Design, pp. 125–151. Academic Press (1999)
12. Jensen, K., Anastassiou, D.: Subpixel edge localization and the interpolation of still images. *IEEE Trans. Image Process.* **4**(3), 285–295 (1995). <https://doi.org/10.1109/83.366477>
13. Karpov, P.: Interpolation Kernels Repository. <https://github.com/inversed-ru/Interpolation-Kernels> (2021). Accessed 15 May 2021
14. Kastryulin, S., Zakirov, D., Prokopenko, D.: PyTorch Image Quality: Metrics and measure for image quality assessment (2019). <https://github.com/photosynthesis-team/piq>. Open-source software available at <https://github.com/photosynthesis-team/piq>
15. Keys, R.: Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Sign. Process.* **29**(6), 1153–1160 (1981)
16. Li, M., Nguyen, T.: Markov random field model-based edge-directed image interpolation. In: 2007 IEEE International Conference on Image Processing, vol. 2, pp. II – 93–II – 96 (2007). <https://doi.org/10.1109/ICIP.2007.4379100>
17. Liu, A., Lin, W., Narwaria, M.: Image quality assessment based on gradient similarity. *IEEE Trans. Image Process.* **21**(4), 1500–1512 (2012). <https://doi.org/10.1109/TIP.2011.2175935>
18. Lottes, T., Garcia, K.: Improved spatial upscaling through fidelityfx super resolution for real-time game engines. <http://advances.realtimerendering.com/s2021/Unity%20AMD%20FSR%20-%20SIGGRAPH%202021.pdf> (2021). SIGGRAPH talk
19. Meijering, E.H.W., Niessen, W.J., Viergever, M.A.: The sinc-approximating kernels of classical polynomial interpolation. In: Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), vol. 3, pp. 652–656 (1999). <https://doi.org/10.1109/ICIP.1999.817196>
20. Mitchell, D.P., Netravali, A.N.: Reconstruction filters in computer graphics. *SIGGRAPH Comput. Graph.* **22**(4), 221–228 (1988). <https://doi.org/10.1145/378456.378514>
21. Prashnani, E., Cai, H., Mostofi, Y., Sen, P.: Pieapp: Perceptual image-error assessment through pairwise preference. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
22. Reisenhofer, R., Bosse, S., Kutyniok, G., Wiegand, T.: A Haar wavelet-based perceptual similarity index for image quality assessment. *Sign. Process. Image Commun.* **61**, 33–43 (2018). <https://doi.org/10.1016/j.image.2017.11.001>
23. Schaum, A.: Theory and design of local interpolators. *CVGIP: Graph. Mod. Image Process.* **55**(6), 464–481 (1993). <https://doi.org/10.1006/cgip.1993.1035>
24. Shapley, R.M., Tolhurst, D.J.: Edge detectors in human vision. *J. Physiol.* **229**(1), 165–183 (1973). <https://doi.org/10.1113/jphysiol.1973.sp010133>
25. Sheikh, H.R., Bovik, A.C.: Image information and visual quality. *IEEE Trans. Image Process.* **15**(2), 430–444 (2006). <https://doi.org/10.1109/TIP.2005.859378>
26. Shi, J., Reichenbach, S.: Image interpolation by two-dimensional parametric cubic convolution. *IEEE Trans. Image Process.* **15**, 1857–1870 (2006)
27. Turkowski, K.: Filter for common resampling tasks. In: A.S. Glassner (ed.) *Graphics Gems*, pp. 147–165. Academic Press (1990)
28. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, vol. 2, pp. 1398–1402 (2003). <https://doi.org/10.1109/ACSSC.2003.1292216>
29. Wolfram Research, Inc.: Mathematica, Version 13.0.0. <https://www.wolfram.com/mathematica>. Champaign, IL, 2021

30. Wolfram Research, Inc.: Fast symbolic solving of real polynomial systems: New in mathematica 10. <https://www.wolfram.com/mathematica/new-in-10/enhanced-algebraic-computation/fast-symbolic-solving-of-real-polynomial-systems.html> (2014). Accessed 17 April 2022
31. Xin Li, Orchard, M.T.: New edge-directed interpolation. *IEEE Trans. Image Process.* **10**(10), 1521–1527 (2001). <https://doi.org/10.1109/83.951537>
32. Xue, W., Zhang, L., Mou, X., Bovik, A.C.: Gradient magnitude similarity deviation: a highly efficient perceptual image quality index. *IEEE Trans. Image Process.* **23**(2), 684–695 (2014). <https://doi.org/10.1109/TIP.2013.2293423>
33. Yu, S., Li, R., Zhang, R., An, M., Wu, S., Xie, Y.: Performance evaluation of edge-directed interpolation methods for noise-free images. In: Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service, ICIMCS '13, pp. 268–272. Association for Computing Machinery, New York, NY, USA (2013). <https://doi.org/10.1145/2499788.2499859>
34. Zhang, B., Sander, P.V., Bermak, A.: Gradient magnitude similarity deviation on multiple scales for color image quality assessment. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1253–1257 (2017). <https://doi.org/10.1109/ICASSP.2017.7952357>
35. Zhang, L., Shen, Y., Li, H.: Vsi: a visual saliency-induced index for perceptual image quality assessment. *IEEE Trans. Image Process.* **23**(10), 4270–4281 (2014). <https://doi.org/10.1109/TIP.2014.2346028>
36. Zhang, L., Wu, X.: An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Trans. Image Process.* **15**, 2226–2238 (2006)
37. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: a feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **20**(8), 2378–2386 (2011). <https://doi.org/10.1109/TIP.2011.2109730>
38. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018). <https://doi.org/10.1109/CVPR.2018.00068>
39. Zhou, D., Shen, X., Dong, W.: Image zooming using directional cubic convolution interpolation. *IET Image Process.* **6**(6), 627–634 (2012)
40. Zhou Wang, Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861>
41. Ziaei Nafchi, H., Shahkolaei, A., Hedjam, R., Cheriet, M.: Mean deviation similarity index: efficient and reliable full-reference image quality evaluator. *IEEE Access* **4**, 5579–5590 (2016). <https://doi.org/10.1109/ACCESS.2016.2604042>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.