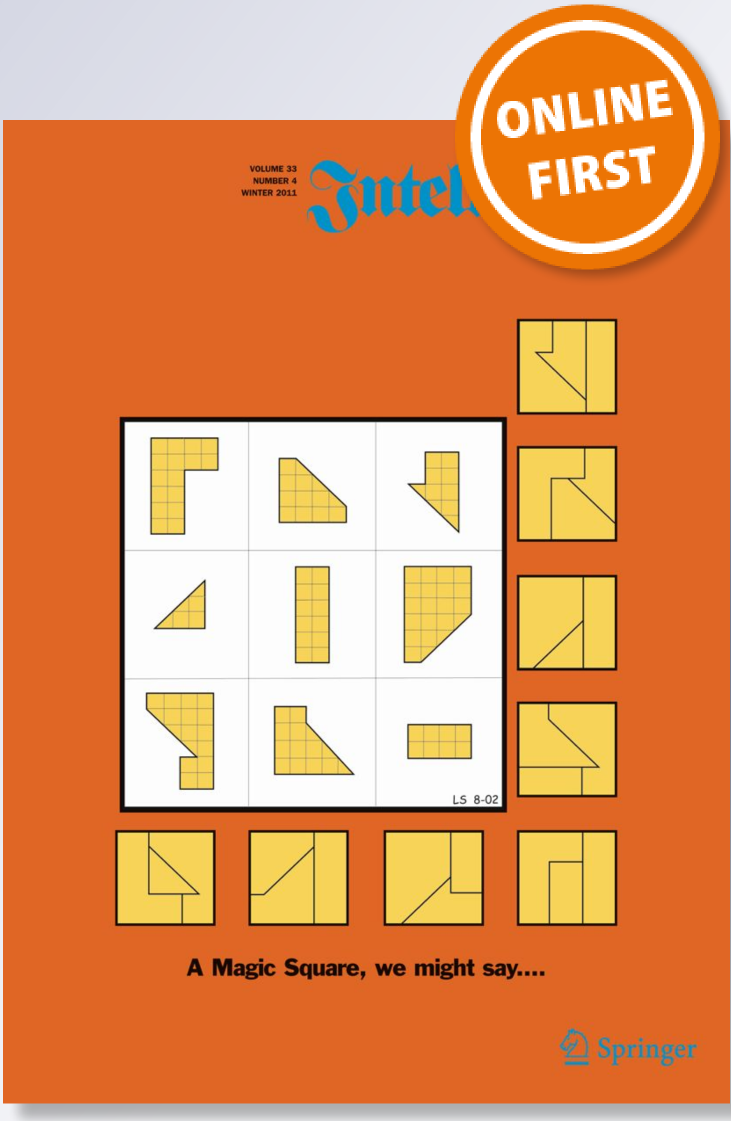
Through the Looking-Glass, and What the Quadratic Camera Found There

Bart de Smit, Mark McClure, Willem Jan Palenstijn, E. Isaac Sparling & Stan Wagon

The Mathematical Intelligencer

ISSN 0343-6993

Math Intelligencer
DOI 10.1007/s00283-012-9309-9



Through the Looking-Glass, and What the Quadratic Camera Found There

BART DE SMIT, MARK MCCLURE, WILLEM JAN PALENSTIJN, E. ISAAC SPARLING, AND STAN WAGON

When a computer with an attached camera points at an external display showing the current screen, one sees a familiar visual feedback loop related to the iteration of an affine function [5]. Of course, the same effect occurs when one looks into a mirror that faces another mirror; this is in essence the same as the familiar audio feedback that occurs when one speaks into a microphone that can also hear the output from nearby speakers. The term *Droste effect* [11] has been used to describe the visual loop because of the infinite recursion caused by self-reference on a packet of Droste cocoa (also on many other products). M. C. Escher used the phenomenon in an original way in *Print Gallery*, and it was the first author's analysis of a gap in Escher's work [9] that led to the discovery of the higher-order effect described here.

Frame and Neger [6] showed how one can modify the classic linear feedback method to obtain some fractal images. The new idea we present is to interpose a nonlinear mathematical filter into the image-capturing process. The filter can be complicated, such as one based on a complex function $f(z)$. Surprisingly, this idea, when used in a self-referential way, leads to the Julia set for $f(z) + c$ appearing on the screen, where the offset c arises from the camera pointing away from the exact center of the screen. If $f(z) = z^2$ then the classic Julia sets of $z^2 + c$ pop out. See [3] for an introduction to Julia sets and the Mandelbrot set.

To be precise, a quadratic filter works as follows: given a pixel located at $z = (x, y) = x + iy$, one forms $f(z) = z^2 = x^2 - y^2 + 2xyi$, finds the color that the camera sees at the point $f(z)$, and uses that color on the display at the point z. Figure 1 shows the result of aiming such a quadratic camera at a rectangular grid; the origin is the black dot just southwest of center. To see why there are two copies of each letter, consider the **Q** at $(1.5, -0.5)$ in the grid. In the transformed image, points near $(1, 0)$ get mapped to points near $(1, 0)$, but so do points near $(-1, 0)$. This square-root behavior (two solutions to $z^2 = 1$) is why two copies of **Q** appear. Plots of the four sides of the rectangle using the complex square-root function explain the eight hyperbolic borders of the transformed image (e.g., the line $y = y_0$ becomes the hyperbola $2xy = y_0$). Loosely speaking, this operation computes the complex square root of the original image. In the general case with nonzero offset, the transforming function is $z \mapsto z^2 + c$, and the quadratic camera's image is derived from the inverse functions $z \mapsto \sqrt{z - c}$.

Figure 2 shows how the Statue of Liberty would look to a tourist using a quadratic lens. Note that the black borders in Figures 1 and 2 arise because those points square to points outside the domain of the original image; by default, the programming uses black for such points.

We have programs available that implement the quadratic camera for either a Macintosh [10] or a Linux or Windows platform [8; this web site contains more images related to our study]; code that implements the camera in *Mathematica*, as well as some other supplementary material, is available at the location described in reference [7]. The program includes an

DOI 10.1007/s00283-012-9309-9

Published online: 02 August 2012

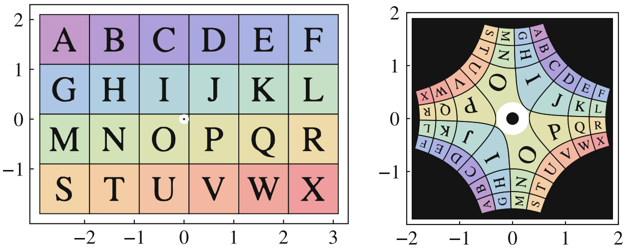


Figure 1. The image at right shows how the alphabet grid appears when viewed by the quadratic camera pointing at the black dot at the origin.

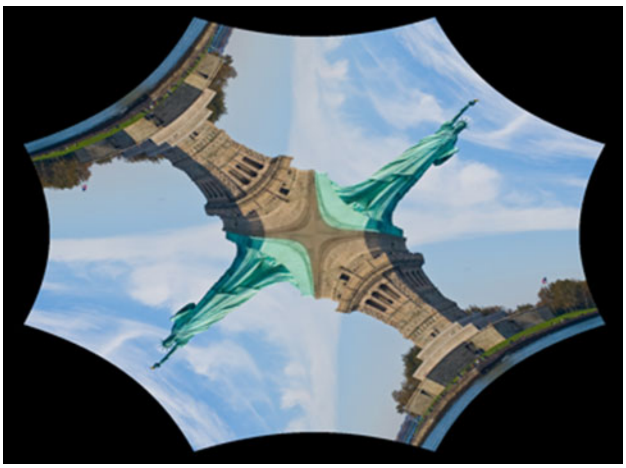


Figure 2. A quadratic camera view of the Statue of Liberty.

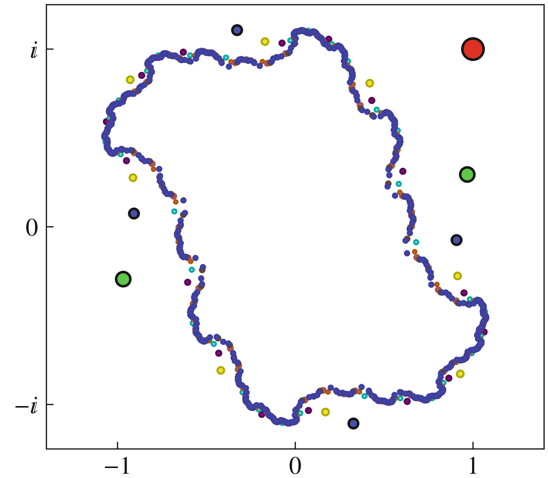


Figure 3. Starting with $1 + i$ and repeatedly applying the inverse of $f(z) = z^2 + 0.15 + 0.43i$ leads to points that approach the Julia set of f.

inset that shows the raw image together with the focal point. The location of that point indicates the value of the offset c, the parameter that gives us the large variety of feedback images from the single camera. As we shall see, the relation of c to the Mandelbrot set controls the sort of feedback image that results.

Before we get into feedback by pointing the quadratic camera at the display of the camera, let's review Julia sets. A simple characterization is that the Julia set of f is the boundary of the set of points whose f-orbits remain bounded. To see why iterating the inverse of a function f leads to the Julia set of f, consider the direct iteration of the function $f(z) = z^2$ in the complex plane. Start with an initial value z_0 and compute z_n recursively by $z_n = f(z_{n-1}) = z_{n-1}^2$. If $|z_0| < 1$, then $|z_n|$ forms a decreasing sequence, but if $|z_0| > 1$, then $|z_n|$ is increasing. From both directions, the unit circle is a dynamical repeller. It is exactly this dynamical behavior that makes the unit circle the Julia set of z^2.

On the other hand, it now makes sense that the unit circle is attractive under inverse iteration of z^2. If we start with an initial point z_0, we can find its two square roots $\pm\sqrt{z_0}$ to get two points closer to the unit circle. We can then find the square roots of these points to get four points even closer to the unit circle. After n steps, we have 2^n points that are quite close to the unit circle.

This process can be performed with any quadratic function (or any polynomial) to yield a large variety of Julia sets. Figure 3 shows the inverse iteration for $f(z) = z^2 + 0.15 +$

AUTHORS

BART DE SMIT earned a Ph.D. from the University of California, Berkeley, in 1993. Currently he is an Associate Professor in the number theory group at the Universiteit Leiden. His main interests are algebraic number theory, algorithms, and mathematical visualization projects. In 2011 he was the chairman of the Problem Selection Committee of the International Mathematical Olympiad.

Universiteit Leiden
Leiden
The Netherlands
e-mail: desmit@math.leidenuniv.nl

E. ISAAC SPARLING graduated from Macalester College with a Bachelor of Arts in Computer Science in 2010. He now works at VMware, building social software for the enterprise. He is passionate about building software to understand and visualize complex social systems that allow individuals to find where they fit in a wider communal fabric. When not contemplating data-driven software problems, Isaac spends his free time on the road, racing his bicycle.

Socialcast, Inc.
San Francisco, CA
USA
e-mail: isaac.sparling@gmail.com

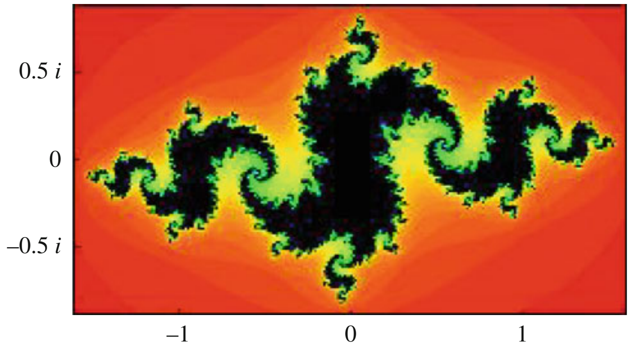


Figure 4. The filled Julia set for $z^2 - 0.83 - 0.18i$.

$0.43i$. The initial point is red, the two inverse iterates are green, the next four are blue, and so on.

As many readers know, varying c can lead to visually attractive and mathematically intricate Julia sets, especially when c is near the boundary of the Mandelbrot set; see Figure 4, where $c = -0.83 - 0.18i$. Note the articulation points where symmetric spirals converge to a single point.

Now to the main point: we set things up so that the quadratic camera points at itself. This can be done either by attaching a camera to one's computer and pointing it at the display, or by using a computer with a built-in camera and pointing it at an external monitor or even just at a mirror that reflects the display. Then the feedback loop becomes a pseudo-analog version of the inverse iteration method for generating Julia sets.

Figure 5 presents an example with offset c near -0.83 $-0.18i$, the value used in Figure 4. The feedback image is a nice match to the algorithmically generated Julia set.

An explanation of why these sets arise starts by considering the inverse of $z^2 + c$, because those two functions determine the quadratically transformed image. The inverse map transforms the image seen by the camera to an octagon. That octagonal image is then, temporarily, placed onto the screen, with points outside the octagon shown as black because the application of $z \mapsto z^2 + c$ to such points yields values outside

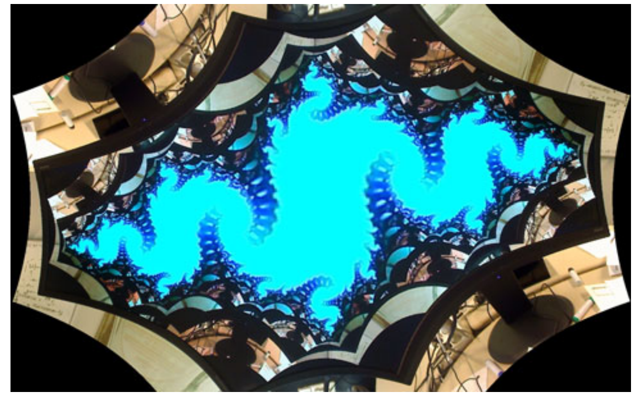


Figure 5. A quadratic feedback image with complex offset near $-0.83 - 0.18i$, a point near the boundary of the Mandelbrot set.

the original image (see Fig. 6). The camera then repeats this process with this new image, places that on the screen, and so on. Thus the complete process is one of continual overlaying, with parts of the image not changing after a certain point. For example, the background never changes.

A classic result [1] is that the Julia set of a rational function is the closure of the set of repelling periodic points of that function. Because the Julia set repels points under iteration of f, it is an attractor for the inverse of f, as we saw above. The overlaying process just described corresponds to iteration of the inverse function, and so the sequence of iterates converges to the Julia set of $z^2 + c$.

Figure 6 simulates this process when $c = 0$. We start at the upper left with a computer screen and a checkerboard pattern forming the background; the yellow circle indicates the unit circle. We assume here that the camera has a square view and sees exactly this image before the quadratic filter kicks in. Then the upper right image shows the result of placing the quadratically transformed camera image on the screen. The black border arises, because the complex square of those points ends up outside the camera's image. The final image

MARK MCCLURE received his Ph.D. from Ohio State University in 1994 studying Fractal Geometry under the direction of Gerald Edgar. Since then, he has published broadly on fractals, dynamics, and related topics, making frequent use of *Mathematica* in his research and teaching. He is now an associate professor of mathematics at the University of North Carolina, Asheville. He loves living in the mountains of western North Carolina, where he can frequently be found hiking or biking.

University of North Carolina at Asheville
Asheville, NC
USA
e-mail: mcmcclur@unca.edu

STAN WAGON teaches at Macalester College, where his main interest is in using the power of modern software to visualize abstract mathematical concepts. Among his mathematical books are *The Banach-Tarski Paradox*, *Mathematica in Action*, and *VisualDSolve*. Other interests include geometric snow sculpture, ski mountaineering, climbing, and mushroom hunting. He is one of the founding editors of *Ultrarunning* magazine, but now finds that covering long distances is much easier on skis than in running shoes.

Macalester College
St. Paul, MN 55105
USA
e-mail: wagon@macalester.edu

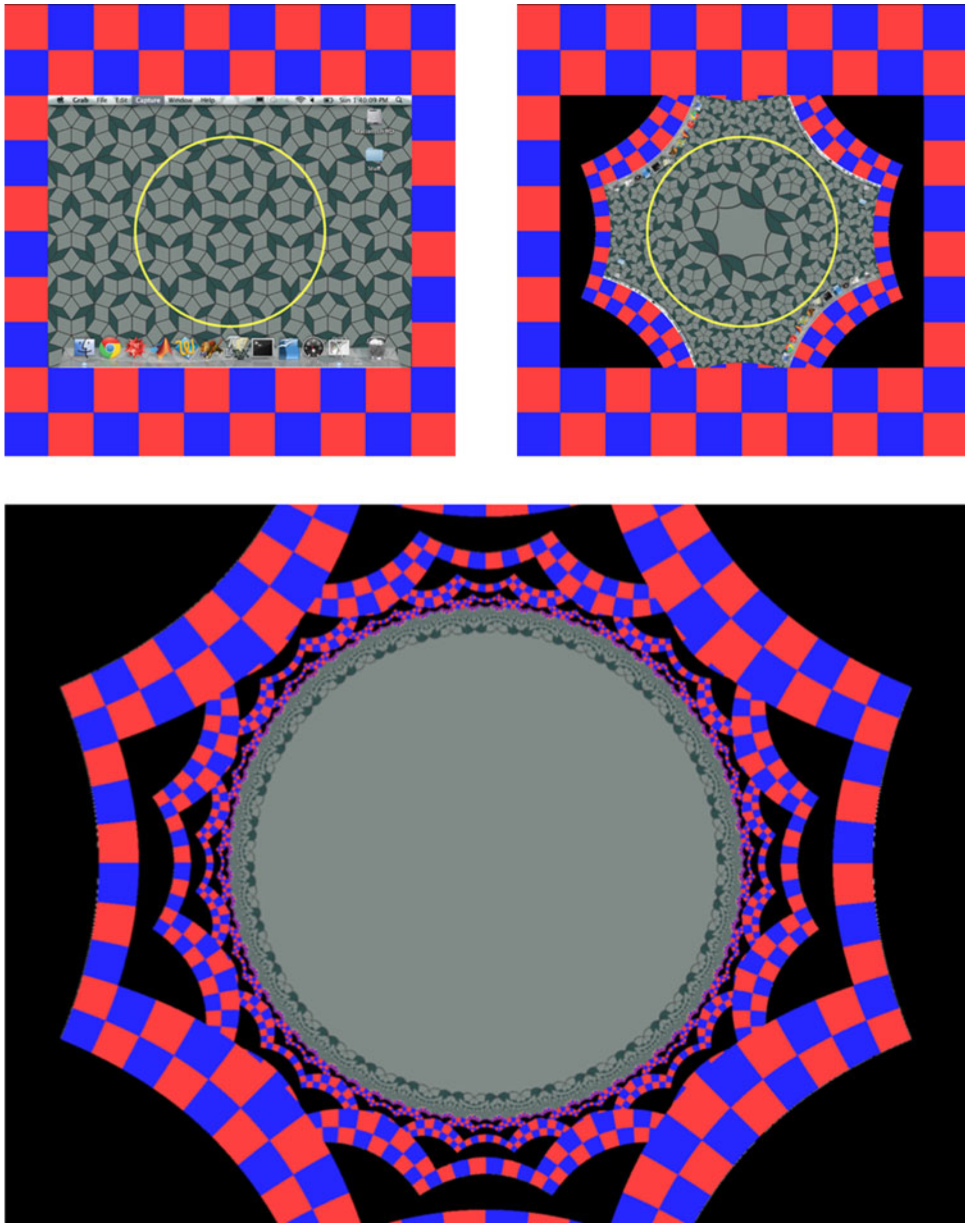


Figure 6. A simulation of quadratic camera feedback using the function $f(z) = z^2$. The upper left shows what the camera sees, the upper right is the result of placing the transformed image on the screen, and the lower image shows five iterations of this process.

WILLEM JAN PALENSTIJN studied number theory at the Universiteit Leiden and currently programs graphics cards at the University of Antwerp for use in medical imaging and materials science. Much of his free time is spent on open source programming projects.

University of Antwerp
Antwerp
Belgium
e-mail: willemjan.palenstijn@ua.ac.be

then shows the result of five iterations of this operation. The filled Julia set — the set of points whose orbits under $z \mapsto z^2$ are bounded — shows up as a disk; it is gray, because the original screen is gray in the region near the origin, and that region expands to fill the disk.

The simulation of Figure 6 differs from the situation in Figure 5 in one subtle way. For the actual feedback image, the window showing what the camera sees occupies less than the full screen. The part of the screen around this window should therefore be considered part of the background.

As first pointed out by Crutchfield [2], who experimented with feedback loops in 1984, there is a damping effect as the light progresses through many iterations of the camera. Thus we see increasing levels of cyan (or another color, depending

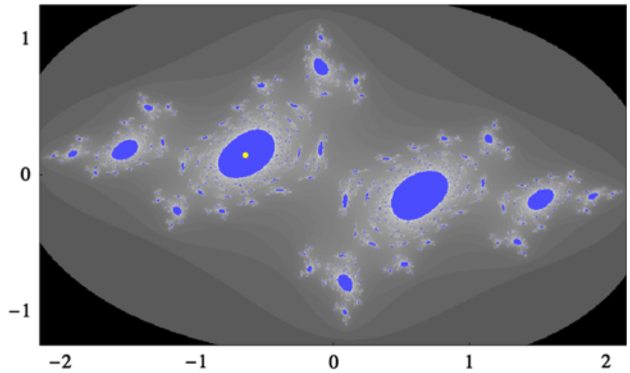


Figure 7. An image showing an approximate Julia-like set arising from an affine variation of the complex squaring function. There are several components with nonempty interior.

on the equipment and the ambient light) as the feedback progresses. For points in the interior of the Julia set, the damping factor completely dominates, and we generally see just a single color.

In the feedback image of Figure 5, the articulation points are not well resolved, as opposed to the delicacy so evident in the mathematical rendering of Figure 4. We believe this is because of the combination of a damping effect with the fact that points near the articulation points require a long time to escape. Thus the damping effect overwhelms the idealized iteration procedure.

Our approach does not explain all details of the feedback, but it is a start at understanding the intriguing effects that arise when the quadratic camera looks at itself. It is somewhat similar to audio feedback, where it is clear that the sound should intensify, but analysis of every detail of the final screech is difficult.

The screen shot in Figure 5 just scratches the surface of the wide variety of effects one sees with the quadratic camera. If the camera is tilted or rotated, then effects related to those transformations will arise. The exact colors generated depend on the monitor used and the type and amount of ambient light. And of course one can modify the idea here to generate feedback Julia sets of cubics or other functions.

Here is a mathematical question that arose from our investigations. When we point the quadratic camera at itself, certain affine effects can come into play because of rotation, tilt, and issues related to screen size. In particular, there might be a stretch in the x-direction only. Such a function is not conformal in $\mathbb{C}$, so we move to $\mathbb{R}^2$. Define a function f by $f(x, y) = ((x/s)^2 - y^2, 2(x/s)y) + (a, b)$, where $(a, b) \in \mathbb{R}^2$ corresponds to the complex offset c used earlier, and s is a (real) horizontal stretching factor. Let R be a closed rectangle in the plane such that $f^{-1}(R) \subseteq R$, and let $X = \cap\{f^{-n}(R) : n < \infty\}$, the part of the feedback image we see as white (or cyan). What can be said about this closed set X?

We can experiment using standard iteration algorithms to investigate X; Figure 7 shows an example. In the first case, $s = 1.22$, the offset is $(-0.9, 0.3)$, and the filled set has several components with nonempty interior; we can prove this, because the yellow point is an attracting fixed point. Such components cannot occur for the traditional filled Julia sets of $z^2 + c$ (see [4]). Another interesting example arises with $s = 1.3$ and $(a, b) = (-1, 0.05)$.

We conclude with a question of physics. The approach here is only partially analog since the light signal is repeatedly transformed in a digital way. Can one construct a physical lens that realizes the complex function z^2 as described here? We hope the answer is YES, for such a lens used with a mirror would yield truly analog Julia sets at the speed of light.

REFERENCES

[1] A. F. Beardon, *Iteration of Rational Functions*, Graduate Texts in Mathematics 132, Springer, New York, 1991.

[2] J. P. Crutchfield, Space-time dynamics in video feedback, *Physica D: Nonlinear Phenomena*, 10:1–2 (1984) 229–245.

[3] R. Devaney, *A First Course in Chaotic Dynamical Systems*, Westview Press, Boulder, Colo., 1992.

[4] R. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed., Westview Press, Boulder, Colo., 2003.

[5] R. Dickau, Nested image rotations, from The Wolfram Demonstrations Project, http://demonstrations.wolfram.com/NestedImageRotations.

[6] M. Frame and N. Neger, Fractal videofeedback as analog iterated function systems, *Fractals* 16 (2008) 275–285.

[7] M. McClure, The quadratic camera, http://stanwagon.com/public/ARCHIVE/QuadraticCamera.

[8] W. J. Palenstijn and B. de Smit, Quadratic camera program for Linux and Windows, http://www.math.leidenuniv.nl/~desmit/qcam.

[9] B. de Smit and H. W. Lenstra Jr., Artful mathematics: The heritage of M. C. Escher, *Notices Amer. Math. Soc.* 50 (2003) 446–451; http://www.ams.org/notices/200304/fea-escher.pdf.

[10] E. I. Sparling, Quadratic camera program for Macintosh's PhotoBooth application, available at http://stanwagon.com/public/julia.app.zip.

[11] Wikipedia: Droste effect, http://en.wikipedia.org/wiki/Droste_effect.