Vol. 8, No. 1: 33–46

Improving the Two-Pass Resampling Algorithm

Michael Kallay
Microsoft Corporation

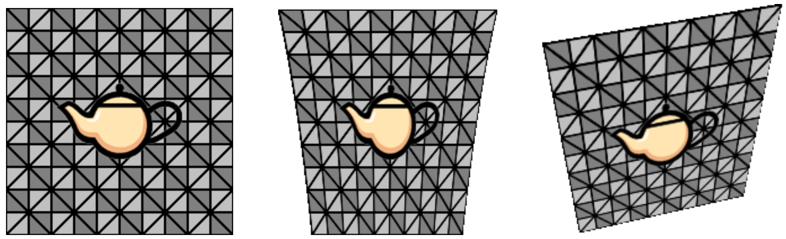
Jason Lawrence
Princeton University

Abstract. The Catmull-Smith two-pass resampling algorithm simplifies the nontrivial reconstruction of a transformed two-dimensional image by decomposing the transformation into two one-dimensional passes. We present a theoretically motivated modification to this algorithm that provides improved image quality. For the case of projective transformations, this improvement results in a final algorithm that is more robust and accurate than the original while even affecting correctness in some cases.

1. Introduction

From image mosaics to multibaseline stereo scene reconstruction, projective transformations play a fundamental role in many areas of computer graphics and computer vision. As with almost every application, a balance is sought between the quality of the result and the cost of the computation. Catmull and Smith's two-pass resampling algorithm [Catmull, Smith 80], [Wolberg et al. 00] provides a desirable balance between quality and cost by simplifying the reconstruction process to two inexpensive one-dimensional scans of an image.

1086-7651/03 $0.50 per page

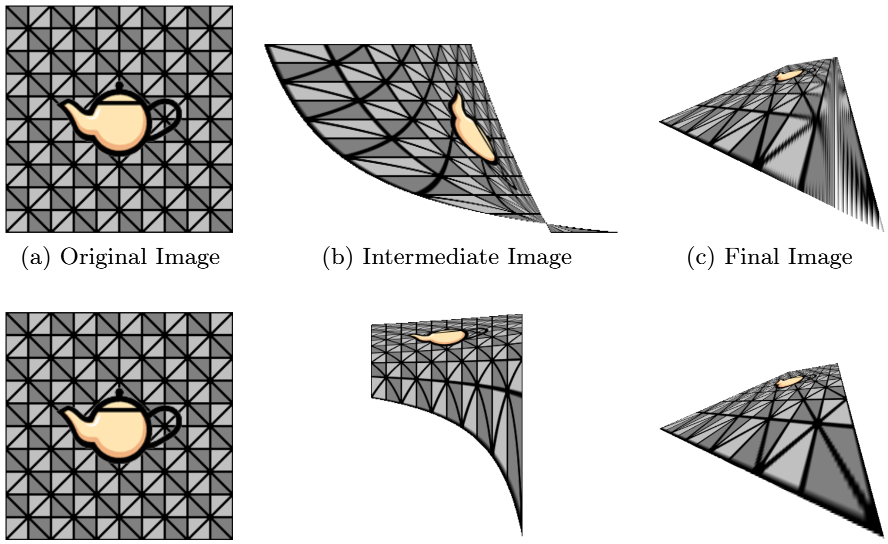


(a) Original Image (b)Intermediate Image (c)Final Image

Figure 1. This image sequence shows the result of decomposing a projective transformation into two one-dimensional passes. (a) Starting with a two-dimensional image, (b) the algorithm first transforms each row according to the perspective projection matrix. (c) The final image is obtained by transforming the columns of the intermediate image.

The first scan maps every row of the image to a row of an intermediate image, and the second scan maps every column of the intermediate image to a column of the final result (Figure 1). This decomposition of a three-dimensional transformation of an image into two one-dimensional passes not only provides clean streaming implementations amenable to computer graphics hardware, but it also simplifies the reconstruction process by avoiding the expensive consideration of a resampling kernel that has been projectively transformed itself. We will show that our improvements help achieve higher quality results within this framework, thus improving a very common algorithm used to efficiently compute photo-quality projective transformations of images.

The decomposition of a transformation into two one-dimensional scans is an approximation of the original mapping, and like any approximation, it involves some errors; these errors may sometimes seriously degrade the quality of the result. The quality can be significantly improved by either prerotating the image by 90°, or by changing the order to scan columns first. Combining these two choices provides four variations, and [Catmull, Smith 80] chooses the one with the largest intermediate image area. This is a reasonable approach, but it does not always lead to the best choice. The association of a large intermediate area with a good final result is heuristic, with no theoretical basis. The area of the intermediate image reflects the behavior of the algorithm on average, but good average behavior may conceal a bad spot, as illustrated in Figure 2(a–c). The transformation there is projective, and the intermediate image of greatest area contains a singularity; the choice in Figure 2(d–f) for the same transformation has a smaller intermediate area, but it produces a clean result. This document proposes a theoretically sound test for choosing the variation with the least worst offender. The test is general, and for projective (perspective) transformations, it turns out to be simple and inexpensive to compute.



(a) Original Image (b) Intermediate Image (c) Final Image

(d) Original Image (e) Intermediate Image (f) Final Image

Figure 2. A bottleneck singularity can arise when the maximum area of the intermediate image is used in guiding the two-pass decomposition. (a–c) The top sequence shows the intermediate and final images of one such degenerate decomposition where prerotating the original image by 90° and then scanning the rows first produces the intermediate image with the maximum area, but exposes a singularity in the final image. (d–f) The bottleneck error, however, is minimized by scanning the columns first without any prerotation and avoids this degenerate situation.

2. The Bottleneck Problem

The main source of trouble in the two-pass resampling algorithm is the so-called bottleneck problem. The problem is best illustrated when the mapping is a 90° rotation. Sampling rows first takes a horizontal line to a vertical one, so the first pass will shrink the entire image to a single vertical line, from which the end result cannot be recovered. Similarly, a first vertical scan will shrink the entire image to a single horizontal line. The problem will obviously disappear if we prerotate the image by 90°.

The rotation example is an extreme case; the intermediate image may have just one collapsed row, as in Figure 2(b). The result shows up as a tear in the final image (Figure 2(c)). However, even when the row does not shrink entirely, a very short row in the intermediate image represents a substantial loss of information, reducing the quality of the final result.

To analyze the bottleneck problem, consider a mapping $(x(u,v), y(u,v))$ from the unit square $0 \leq u, v \leq 1$, and suppose we scan rows first. Let du be an infinitesimal horizontal line segment in the source image. The first pass maps it to a horizontal segment of length $|dx| = \left|\frac{\partial x}{\partial u} du\right|$ in the intermediate image, and the second pass takes it to a segment of length $\sqrt{(dx)^2 + (dy)^2}$. The loss of information in the first pass is the ratio

$$\frac{\sqrt{(dx)^2 + (dy^2)}}{|dx|}. \tag{1}$$

The worst spot in the image will be at the maximum of this ratio, so we should choose the variation that minimizes this maximum. (Note that this test does not involve the intermediate image!).

The maximum of this ratio coincides with the maximum of the slope

$$\left|\frac{dy}{dx}\right| = \left|\frac{\partial y/\partial u}{\partial x/\partial u}\right|. \tag{2}$$

Thus, the row-first bottleneck error is

$$\max\left\{\left|\frac{\partial y/\partial u}{\partial x/\partial u}\right| : 0 \le u, v \le 1\right\}. \tag{3}$$

Similarly, the columns-first error is

$$\max\left\{\left|\frac{\partial x/\partial v}{\partial y/\partial v}\right| : 0 \le u, v \le 1\right\}. \tag{4}$$

With prerotation the rows-first error is

$$\max\left\{\left|\frac{\partial y/\partial v}{\partial x/\partial v}\right| : 0 \le u, v \le 1\right\}, \tag{5}$$

and the columns-first error is

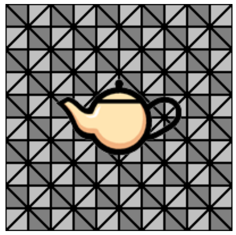
$$\max\left\{\left|\frac{\partial x/\partial u}{\partial y/\partial u}\right| : 0 \le u, v \le 1\right\}. \tag{6}$$

To minimize the bottleneck error, choose the variation with least error.

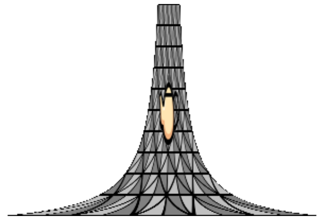
In Figures 2(a–c), scanning rows first with prerotation was the choice with maximal intermediate area, but it led to a bottleneck singularity, which manifests itself as a tear in the final image. The least bottleneck error indicates scanning columns first without prerotation, which is clearly superior as Figures 2(d–f) shows.

3. The Aliasing Error

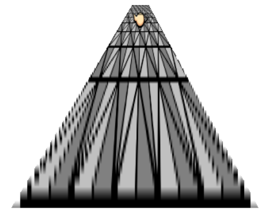
Minimizing the bottleneck error alone does not address the problem illustrated in Figures 3(a–c). The transformation there stretches the lower portion of the



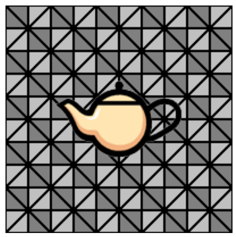
(a) Original Image



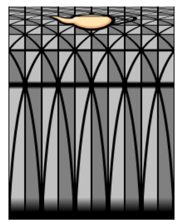
(b) Intermediate Image



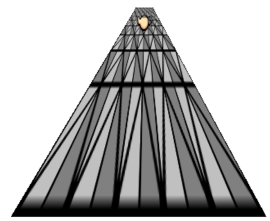
(c) Final Image



(d) Original Image



(e) Intermediate Image



(f) Final Image

Figure 3. (a–c) The bottleneck error alone does not account for the unwarranted aliasing of vertical edges as shown in this image sequence. The bottleneck error is minimized by transforming the rows first, but neglects to account for the aliasing error this decomposition produces. (Notice the severe "bending" of the diagonal lines in the intermediate image). (d–f) The aliasing error is minimized by transforming the columns first and causes vertical edges to stay vertical in the intermediate image.

image vertically (while shrinking the upper portion). As a result, the fine pixels of any diagonal feature-line at the bottom will be magnified by the transformation to visible steps, aliasing the line (Figure 3(c)). The flaw is inherent in the transformation, but vertical lines are immune to it. Unfortunately, the first pass (rows-first) in Figure 3(b) takes vertical lines in the source to highly slanted diagonal lines in the intermediate image, and the vertical stretching in the second pass aliases them (Figure 3(c)). This defect is highly visible on the image's vertical edges. It cannot be blamed on the transformation; it is introduced by its two-pass decomposition. The choice indicated by the bottleneck test is bad from the aliasing point of view; scanning column-first would avoid the problem because its first pass leaves vertical lines vertical (Figure 3(d–f)).

The maximum-area test seems to address this problem in this example, but theoretically it is only good on the average. We should be able to do better by eliminating the worst offender, and this requires an effective metric for the offense. For rows-first without prerotation, we want vertical lines to stay as vertical as possible in the intermediate image. Differentially, consider an infinitesimally small vertical line segment dv in the source image; the first pass will map it to an infinitesimal segment whose deviation from vertical is measured by $\left|\frac{\partial x}{\partial v}\right|$. (This is actually the inverse of the segment's slope in the intermediate image.) A reasonable metric would therefore be the greatest product of stretching and slanting:

$$\max\left\{\left|\frac{\partial y}{\partial v}\right|\left|\frac{\partial x}{\partial v}\right|\right\}. \tag{7}$$

Similarly, for columns-first without prerotation, it would be

$$\max\left\{\left|\frac{\partial x}{\partial u}\right|\left|\frac{\partial y}{\partial u}\right|\right\}. \tag{8}$$

With prerotation, this error is

$$\max\left\{\left|\frac{\partial y}{\partial u}\right|\left|\frac{\partial x}{\partial u}\right|\right\} \tag{9}$$

for rows first and

$$\max\left\{\left|\frac{\partial x}{\partial v}\right|\left|\frac{\partial y}{\partial v}\right|\right\} \tag{10}$$

for columns first. To minimize aliasing, choose the variation with least aliasing error.

Although the aliasing error does not distinguish between the four cases (i.e., it is unaffected by prerotating the source image), the bottleneck error does. Consequently, the combination of the two leads to an unambiguous choice in most cases.

4. Combining the Two Tests

A good algorithm for choosing the best variation needs to consider both the bottleneck and the aliasing errors, but it is not clear how to decide when the two tests indicate conflicting choices. A simple strategy may be to minimize a linear combination of the two errors. It is somewhat akin to adding apples and oranges, but understanding the relative impacts of the two errors may help us choose the coefficients. For example, a simple sum (i.e., linear combinations with coefficients 1) assigns to the bottleneck effect of a 45° rotation the same error as it does to a vertical stretch by a factor of 2 with slanting a vertical line to 45° by the first pass (probably overrating the aliasing error). Ultimately, the coefficient should be determined empirically, based on the visual importance of the two effects. Fortunately, the aliasing error is always bounded while the bottleneck error becomes infinite when the first pass has a singularity; the bottleneck error will then outweigh the aliasing error (as it should) with any choice of positive coefficients. We have found that assigning an equal weight to the bottleneck error and aliasing error produces nice results.

5. Computing the Errors for a Projective Transformation

A projective (perspective) transformation can be written as:

$$x(u,v) = \frac{au+bv+d}{mu+nv+p} \qquad y(u,v) = \frac{eu+fv+h}{mu+nv+p} \tag{11}$$

The mapping is valid for the unit square only if $mu+nv+p \neq 0$ for all $0 \leq u, v \leq 1$.

5.1. Computing the Bottleneck Error

Elementary calculus yields

$$\frac{\partial y/\partial u}{\partial x/\partial u} = \frac{(en-fm)v+ep-hm}{(an-bm)v+ap-dm}. \tag{12}$$

If the sign of the denominator at $v=0$ differs from the sign at $v=1$, (or if either value is zero,) then the ratio is unbounded; otherwise its maximum for $0 \leq v \leq 1$ is attained either at $v=0$ or at $v=1$ (proof at the web site listed at the end of this article). The rows-first bottleneck error is therefore infinite if $(ap-dm)(an-bm+ap-dm) \leq 0$. Otherwise, the error is

$$\max\left\{\left|\frac{en-fm}{ap-dm}\right|, \left|\frac{en-fm+ep-hm}{an-bm+ap-dm}\right|\right\}. \tag{13}$$

Similarly, the column-first bottleneck error is

$$\frac{\partial x/\partial v}{\partial y/\partial v} = \frac{(bm-an)u+bp-dn}{(fm-en)u+fp-hn}. \tag{14}$$

It is unbounded if $(fp-hn)(fm-en+fp-hn) \leq 0$. Otherwise, the error is

$$\max\left\{\left|\frac{bp-dn}{fp-hn}\right|, \left|\frac{bm-an+bp-dn}{fm-en+fp-hn}\right|\right\}. \tag{15}$$

The errors in the prerotation cases are obtained by switching and inverting these ratios. Thus, for rows-first with prerotation, the error is unbounded if $(bp-dn)(bm-an+bp-dn) \leq 0$. Otherwise, the error is

$$\max\left\{\left|\frac{fp-hn}{bp-dn}\right|, \left|\frac{fm-en+fp-hn}{bm-an+bp-dn}\right|\right\}. \tag{16}$$

For columns-first with prerotation, the error is unbounded if $(ep-hm)(en-fm+ep-hm)\leq 0$. Otherwise, the error is

$$\max\left\{\left|\frac{ap-dm}{en-fm}\right|,\left|\frac{an-bm+ap-dm}{en-fm+ep-hm}\right|\right\}. \tag{17}$$

5.2. Computing the Aliasing Error

Unlike the bottleneck error, the maximum of our theoretical aliasing error for projective transformations is not proven to be attained of at the corners of the image for projective transformations. But a reasonable approximation for the maximum of a product is the product of the maxima of its factors, which is a conservative estimate. Since the product of maximal stretching and maximal slanting does attain its maximum at the corners for projective transformations, it is our choice as an approximate aliasing error for projective transformations.

$$\left(\max\left|\frac{\partial y}{\partial v}\right|\right)\left(\max\left|\frac{\partial x}{\partial v}\right|\right), \tag{18}$$

It is based on maxima of partial derivatives, and these attain their maxima at the corners (proof on the web site listed at the end of this article). This approximate aliasing error is obtained from four corner-values of simple expressions. For example,

$$\frac{\partial x}{\partial u}=\frac{(an-bm)v+ap-dm}{(mu+nv+p)^2} \tag{19}$$

hence, $\max\left|\frac{\partial x}{\partial u}\right|=$

$$\max\left\{\begin{array}{cc}\left|\frac{ap-dm}{p^2}\right| & \left|\frac{ap-dm}{(m+p)^2}\right| \\ \left|\frac{an-bm+ap-dm}{(n+p)^2}\right| & \left|\frac{an-bm+ap-dm}{(m+n+p)^2}\right|\end{array}\right\}. \tag{20}$$

The other maxima are found in a similar fashion from

$$\frac{\partial y}{\partial u}=\frac{(en-fm)v+ep-hm}{(mu+nv+p)^2}, \tag{21}$$

$$\frac{\partial x}{\partial v}=\frac{(bm-an)u+bp-dn}{(mu+nv+p)^2}, \text{ and} \tag{22}$$

$$\frac{\partial y}{\partial v}=\frac{(fm-en)u+fp-hn}{(mu+nv+p)^2}. \tag{23}$$

References

[Catmull, Smith 80] Ed Catmull and Alvy Ray Smith. "3-D Transformations of Images in Scanline Order." In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 279–285. City: Publisher, 1980.

[Wolberg et al. 00] George Wolberg, H. M. Sueyllam, M. A. Ismail, and K. M. Ahmed. "One-Dimensional Resampling with Inverse and Forward Mapping Functions. *journal of graphics tools* 5:3 (2000), 11–33.

Web Information:

The proofs that the error maxima are attained at the image corners along with pseudocode of the improved two-pass algorithm are availble online at http://www.acm.org/jgt/papers/KallayLawrence03

Michael Kally, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 (micha@microsoft.com)

Jason Lawrence, Princeton University, Dept. of Computer Science, 35 Olden Street, Princeton, NJ 08544 (jlawrenc@cs.princeton.edu)

Received September 13, 2002; accepted May 10, 2003.