Pixel-Aligned Warping for Multiprojector Tiled Displays

Mark Hereld[1,2] and Rick L. Stevens[1,2,3]
[1]*Mathematics and Computer Science Division, Argonne National Laboratory*
[2]*Computation Institute, University of Chicago and Argonne National Laboratory*
[3]*Department of Computer Science, University of Chicago*
{hereld,stevens}@mcs.anl.gov

Abstract

Multiprojector tiled displays offer a scalable path to high-resolution display systems, often in a large format. Such displays have been employed in a range of applications from scientific visualization to collaborative virtual environments. Images projected onto these systems have always looked less sharp than expected owing to the image warping required to achieve geometrical alignment of the partial images to form the seamless whole. Similar degradation of image quality accompanies keystone correction techniques and image warping onto nonplanar surfaces used in single projector applications. In this paper we discuss the underlying cause of the degradation of apparent image resolution, and we introduce a new class of warping techniques, called pixel-aligned warping, that tend to preserve image sharpness at the expense of strict adherence to underlying geometrical constraints that define the desired warp. We demonstrate how such techniques can be used to significantly improve the apparent sharpness of the final image.

1. Introduction

Most warping techniques used for tiled display systems are based on rubber sheeting the image texture into the frame buffer at suitable granularity. The texture is usually mapped by bilinear interpolation on a warped mesh of triangles. The resulting projected image does the best job of placing data properly onto the display surface but results in a significant loss of crispness for many types of content.

Image warping enters into the technical arsenal of projected imagery in several ways. Projector misalignment is often corrected with keystone corrections implemented by warping. Misalignment and sometimes distortion are corrected in a related way when tiling multiple projectors together to form a larger display.

In a similar way, image resampling (a kind of simple subset of image warping) is used extensively in connection with digital publishing and other forms of image processing in order to produce properly scaled images. Also, resampling often comes into play (with often poor results) when converting digital imagery for projection in the form of PowerPoint and other presentation media as a result of the mismatch between resolution of original digital matter and the projector that finally renders it onto a wall. Specifically, most laptops in a business or research context have higher resolution than the projectors used to present technical presentations. Resampling, either at the computer end of the cable or in the projector itself, is often responsible for serious degradation of fidelity in text and line-based graphical material.

In a wide range of applications, it is neither necessary nor advantageous to shift important high-resolution features such as text by fractional pixels to ensure exact placement on the display surface. The alternative approach described in this paper can produce clear, crisp image content where it matters: wherever there are high-contrast features.

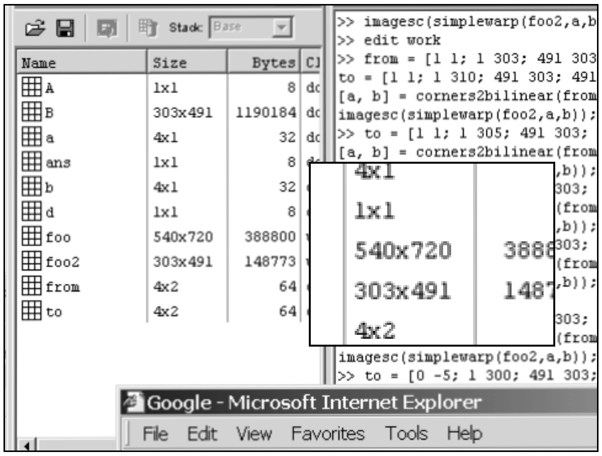
In the rest of this paper we discuss a different approach to rendering content into warped frame buffers that emphasizes image crispness. First we discuss the essential issue, the frequency spatial content of a pixel. Next we outline the ingredients to a class of warping methods, pixel-aligned warping, that can produce clearer images. We then present experimental results and discuss these results.

2. Pixel exploit

It is well known, often exploited, but seldom appreciated that pixelation of images introduces high spatial frequencies at pixel boundaries. These spectral components are often used to advantage when rendering points, horizontal or vertical lines, corners, text, and other specially aligned content. Exploiting this characteristic of common display systems produces crisp features with resolution that far exceeds the Nyquist limit imposed on general data sampled at the pixel pitch. Unfortunately, these high frequencies can be used to create sharp edges only at pixel boundaries. An excellent summary of data sampling, image reconstruction, and antialiasing is given in an article by Wolberg [1].

However, when images that exploit pixel edges to convey real information must be warped, scaled, or otherwise mangled to satisfy unrelated conditions, the artificial crispness is typically lost in the transformation and reconstruction (Figure 1). When the geometrical relationships in the original image must be accurately preserved, little can be done besides applying one of the standard antialiasing techniques [2]. The result is an appropriately smooth and artifact-free reconstruction of the image that preserves distances between features, subject to the Nyquist limit.

On the other hand, often a crisp rendition of original image features is more important than accurate placement on the display surface of these features. Textual image content is a prime example of this situation.



Name	Size	Bytes	Cl
A	1x1	8	dc
B	303x491	1190184	dc
a	4x1	32	dc
ans	1x1	8	
b	4x1	32	
d	1x1	8	
foo	540x720	388800	
foo2	303x491	148773	
from	4x2	64	
to	4x2	64	

```
>> imagesc(simplewarp(foo2,a,b
>> edit work
>> from = [1 1; 1 303; 491 303
to = [1 1; 1 310; 491 303; 491
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));
>> to = [1 1; 1 305; 491 303;
[a, b] = corners2bilinear(from
,b));
303;
(from
,b));
303;
(from
,b));

303;
(from
imagesc(simplewarp(foo2,a,b));
>> to = [0 -5; 1 300; 491 303;
```

Figure 1. Example of textual image content "damaged" by interpolation. The image has been warped with pixel values determined by bilinear interpolation. Inset shows 2x magnification of a piece of the image for detail.

The example in Figure 1 shows a portion of a display screen that has been distorted. This 360x480 pixel portion has only approximately two pixels of coordinate skew in each direction. It shows clearly the softening and blurring of single-pixel features from vertical lines to individual characters, as well as the characteristic modulation into and out of clarity as the coordinate alignment between source and destination pixel "beat" against one another.

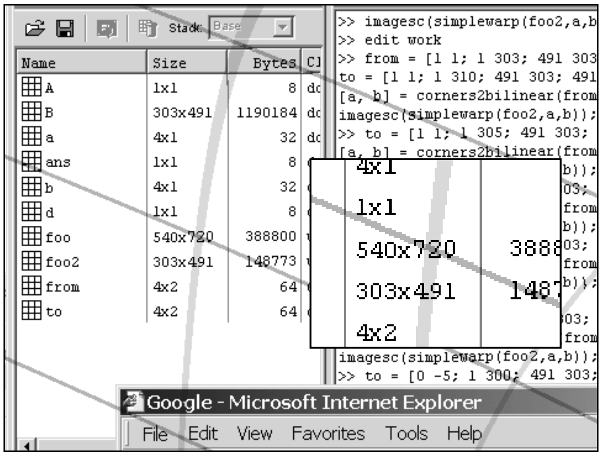
3. Islands in the ocean

Consider the example of the workstation display populated with an array of icons, text windows, visualizations, window borders and decorations, and widget panels. Such imagery is typically poorly represented when warped or scaled into a new display context. Yet, such imagery is probably the most commonly encountered by most people in most applications.

A mapping from the frame buffer coordinates (x,y) to the image source coordinates (u,v) can be written in the following way:

$$(u,v) = (f_u(x,y) , f_v(x,y)). \qquad (1)$$

This equation expresses the transformation of coordinate systems needed to compensate for any or all misalignments, optical distortions, and a nonplanar display surface. It is used in Figure 1 to identify the pixels near (u,v) in the source image to include in the interpolation resulting in the value to place at pixel position (x,y) in the frame buffer.



Name	Size	Bytes	Cl
A	1x1	8	dc
B	303x491	1190184	dc
a	4x1	32	dc
ans	1x1	8	
b	4x1	32	
d	1x1	8	
foo	540x720	388800	
foo2	303x491	148773	
from	4x2	64	
to	4x2	64	

```
>> imagesc(simplewarp(foo2,a,b
>> edit work
>> from = [1 1; 1 303; 491 303
to = [1 1; 1 310; 491 303; 491
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));
>> to = [1 1; 1 305; 491 303;
[a, b] = corners2bilinear(from
b));
03;
from
b));
03;
from
b));

03;
from
imagesc(simplewarp(foo2,a,b));
>> to = [0 -5; 1 300; 491 303;
```

Figure 2. Effects of coordinate mapping between source image and frame buffer.

A simple nearest-neighbor reconstruction of pixel values instead places the pixel found at

$$(u',v') = \mathrm{ROUND}(\, f_u(x,y)\, ,\, f_v(x,y)\,) \qquad (2)$$

into the frame buffer. The result shown in Figure 2 can be compared directly with that in Figure 1 to see the degradation introduced by interpolation. Overlaid onto the image in Figure 2 are curved lines marking the coordinate *rifts* along half-integral values of u and v. They surround zones, or *plates*, that have been effectively clipped out of the source image and grafted into the frame buffer. Adjacent plates are either separated by or overlap by one pixel in the source image. As can be seen in the figure inset, imperfections lie along these rifts. In this example the image is almost everywhere a flawless reproduction of the source image.

The common perception is that some sort of interpolation or even optimal filtration, in the sampling theory sense, is good for preserving high resolution and that nearest-neighbor approaches are not [3][4]. However, in applications where the mapping function doesn't severely distort or reorient the image content, the artifacts from nearest neighbor pixel picking are minimal and confined to narrow corridors (the rift lines) in the image. If one is willing to loosen the tolerance in placement accuracy of critical features of the image, then the impact of these artifacts can be further minimized by making them fall on low contrast portions of the image.

We have considered several approaches to achieve this goal, collectively called *pixel-aligned warping* techniques. Here are a few of them:

- Soften the transition across the rifts by spreading the one-pixel discontinuity smoothly over a swath several pixels wide.
- Perturb the rift lines directly to effect better placement of the artifacts.
- Perturb the warp according to local measure of steepness.
- Segment the image into unwarpable islands of content.
- Minimize an energy function that includes terms reflecting the affinity of features for pixel boundaries and gives weight to keeping proximal features together.
- Crystallize pixel values around nucleation centers related to locations of high-contrast features in the image.

In this paper we expand on a technique that we refer to as *islands in the ocean*, chosen because it is relatively easy to implement for experimental purposes. The idea here is that the image is segmented into unwarpable *islands* surrounded by a warpable *ocean*. Each island is pinned to the pixel grid en masse, preserving maximum clarity, and the deformation is distributed through the ocean.

The process can be summarized in these steps: (1) compute the contrast image, (2) binarize the contrast image with a threshold test, (3) fill and open the image to merge areas of high contrast into islands, (4) pin islands to pixel grid using warp function, (5) warp ocean into place as texture using bilinear interpolation, and (6) transfer islands directly from source image onto the ocean.

We have experimented with this method using Matlab. We are not for the moment concerned about how such steps would be efficiently implemented to use graphics accelerator hardware.

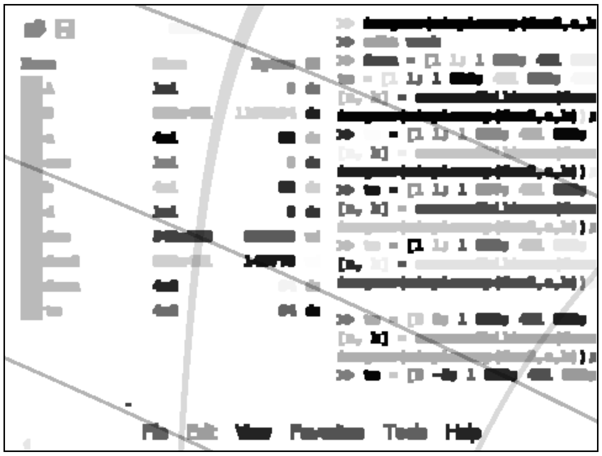


Figure 3. Intermediate result from processing showing identification of the islands by randomly selected shades of gray with rift lines overlaid to show which of the islands require special handling.

First the contrast image is computed, resulting in a value at each pixel of the source image based on the maximum and minimum values in the 3x3 pixel box. We use the usual definition for contrast:

$$(\, I_{max} - I_{min}\,) \,/\, (\, I_{max} + I_{min}\,) \qquad (3)$$

and subsequently threshold the image at values of the contrast greater than 0.9 to create a binary image.

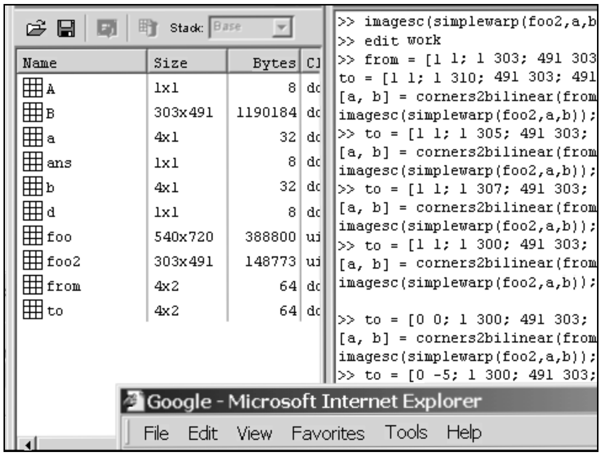
We consolidate and clean up this image using the morphological *close* operation (a *dilation* followed by an *erosion*). The resulting image (Figure 3) can be processed to identify the isolated objects, islands.

After transforming the source image into the buffer with a bilinear interpolated warp operation, we transfer each of the isolated and naturally sharp islands pixel-by-pixel into their corresponding position in the frame buffer. That position is determined by using the coordinates of the island centroid in the frame buffer coordinate system. This cookie-cutter operation is relatively quick.

We have expressly disregarded several possible issues in this implementation: (1) the possibility that particularly large islands might bump into or intersect with other islands by virtue of their accumulating several pixels worth of skew in crossing several *rift* lines, (2) effects of not aggressively filling in holes within islands, particularly with regard to the underlying warped ocean, and (3) fractional pixel skew at the interface between island and ocean (the *beach*). We expect that these and other omissions would introduce much complexity in the computation without improving the final result proportionately.

4. Discussion

The final results for this example image are shown in Figure 4. Possible artifacts around the edges of the island grafts detract only minimally from the appearance of the image. Also clearly visible are the slightly blurred features of areas of the image that didn't quite meet the contrast cut-off criterion. It is nonetheless considerably sharper than its interpolated cousin (in Figure 1).



Name	Size	Bytes	Cl
A	1x1	8	dc
B	303x491	1190184	dc
a	4x1	32	dc
ans	1x1	8	dc
b	4x1	32	dc
d	1x1	8	dc
foo	540x720	388800	ui
foo2	303x491	148773	ui
from	4x2	64	dc
to	4x2	64	dc

```
>> imagesc(simplewarp(foo2,a,b
>> edit work
>> from = [1 1; 1 303; 491 303
to = [1 1; 1 310; 491 303; 491
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));
>> to = [1 1; 1 305; 491 303;
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));
>> to = [1 1; 1 307; 491 303;
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));
>> to = [1 1; 1 300; 491 303;
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));

>> to = [0 0; 1 300; 491 303;
[a, b] = corners2bilinear(from
imagesc(simplewarp(foo2,a,b));
>> to = [0 -5; 1 300; 491 303;
```

Figure 4. Result of applying the islands in the ocean warping algorithm.

This method and other similar methods are able to trade away absolute precision in placement of image features on the display surface in favor of crisp appearance of features, improving legibility of text, for example. The example discussed is well suited to the method.

On the other hand, the technique will run into difficulties under several conditions. Image rotation, for example, is not typically well handled by these methods since the advantage conveyed by exploiting pixel boundaries is manifestly bound to horizontal and vertical crispness. (Of course the situation changes in display devices with pixels that are instead hexagonal, triangular, or random.)

Typical misalignments found in casually placed presentation projections usually have relatively severe keystone distortion. These gross errors are well handled by currently available correction techniques [5].

In the case of tiled display applications, however, it is often the case that the projectors have been carefully placed in a regular array and aligned. In such arrays, it is usually possible to arrange for distortions and misalignments to be kept under ten or so pixels. Pixel-aligned warping might be applied profitably in this case to produce crisp images across the display surface. This is particularly true in the case of non-overlapping or minimally overlapping tiles.

When there is significant overlap between tiles, other issues come into play. Specifically, the pixel edges of neighboring tiles won't often line up with one another. The benefits of pixel-aligned warping in one tile will be undercut by the offset of the other. A method for optimizing apparent resolution in this case has demonstrated the possibility of achieving some level of super-resolution [6].

5. Conclusions and future directions

We have demonstrated a new approach to image warping that could provide significant benefits in the form of image clarity when distortions are modest and alignment is decent. In addition to describing a number of possible approaches to pixel-aligned warping that might be better suited either in implementation or performance characteristics to different situations, we have burrowed into some of the details of a straightforward but effective technique — *islands in the ocean*.

We have demonstrated that this technique can produce very clear images without noticeable compromise when compared with standard interpolation-based warps.

For pixel-aligned warping methods to become generally useful, several issues deserve future consideration:

- Robust rules to handle pathological cases gracefully.
- Implementation that is sufficiently fast, possibly compatible with existing hardware accelerators.
- Ability to handle blending areas in tiled displays.
- Awareness of horizontal and vertical features and their relation to horizontal and vertical rifts in the warp: allows lines to stretch or shrink but not kink.
- Possible application to image scaling (resampling) in projectors — typically SXGA driving an XGA projector.

Acknowledgments

We have benefited from discussions with members of the Futures Laboratory. This work was supported by the U.S. Department of Energy, Contract W-31-109-ENG-38.

References

[1] G. Wolberg, “Sampling, Reconstruction, and Antialiasing,” in *CRC Handbook of Computer Science and Engineering*, Ed. by A. B. Tucker, CRC Press, 1997, 2004.

[2] F. C. Crow, “A Comparison of Antialiasing Techniques,” IEEE Computer Graphics and Appl., Vol. 1, No. 1, Jan. 1981.

[3] R. Surati, “A Scalable Self-Calibrating Technology for Seamless Large-scale Displays,” Ph.D. Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 1999.

[4] R. M. Clodfelter, D. Sadler, and J. Blondelle, “Large High Resolution Display Systems via Tiling of Projectors,” white paper, Barco Simulation Products, 2003.

[5] R. Sukthankar, R. G. Stockton, and M. D. Mullin, “Automatic Keystone Correction for Camera-assisted Presentation Interfaces,” in *Advances in Multimodal Interfaces – Proceedings of International Conference on Multimedia Interfaces, 2000*, pp. 607 - 614.

[6] C. Jaynes and D. Ramakrishnan, “Super-Resolution Composition in Multi-Projector Displays,” in Proceedings of *PROCAMS 2003: IEEE International Workshop on Projector-Camera Systems, at ICCV 2003*.