



Pandemonium: A Panorama App to Maximize Jank

Dave Pagurek

March 2025

Abstract

The panorama feature on early smartphones produced frequent delightful visual glitches. Modern panorama apps are better at accurately reproducing their subjects, and subsequently forget their true goal: surprising the user with the unexpected and uncanny. We propose a novel camera system that maximizes occurrences of weird visuals rather than trying to sweep them under the rug.

1 Introduction

Who doesn't want to see their cat have eight legs? [2] Panoramas used to provide a steady source of delight as subjects inevitably move while the panorama is being captured, such as the cat in Figure 1. We refer to these glitches as *jank*. While jank still occurs in the capture phase of modern panorama apps, the final image disappointingly removes most jank.

App makers have lost sight of their true goal: producing images that inspire a sense of biblical awe. Like how paintings got weird and embraced the surreal and abstract once photographs became common, panoramas too need to get weird now that it's easy



Figure 1: A panorama of a cat featuring more limbs than typically expected. [2]

to capture reality.

We aim to fill this niche. First, we explore what used to produce panorama jank. We then design a new camera system optimized for jank. Finally, we evaluate this system by exploring its effectiveness at producing surprising results in the hands of users.

2 Background

Slit scanning is a photography technique where a full, two-dimensional image is created by scanning a series

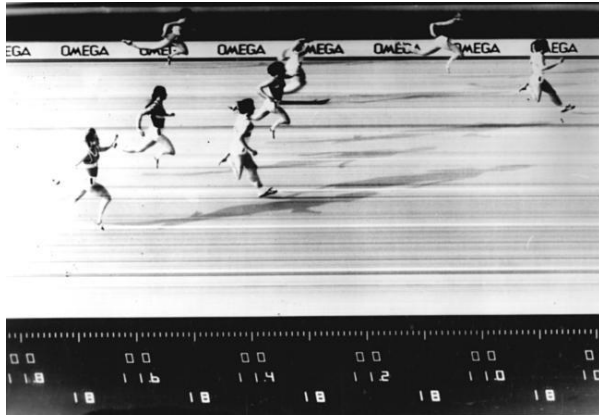


Figure 2: Finish photo of a world record 100m run by Marlies Oelsner, 1 July 1977. [1] The “OMEGA” text in the background appears undistorted because it was moving horizontally over time.

of one-dimensional images over time. This is used for photo finishes in sports, such as the one in Figure 2. The camera only sees the vertical finish line. As the film slides through the camera horizontally, that single vertical line is stretched across the full frame. Although the slit captures the same space, the time varies, allowing one to easily see who crosses the finish line first by seeing who appears first horizontally. It also has the side effect of distorting everything that isn’t moving purely horizontally, leading to curvy and stretched limbs.

Slit scanning is similar to how digital cameras take photos. Unlike slit scanning, each pixel is looking at a different position in space, but each row of pixels gets exposed with a slight time offset from the row preceding it. This leads to an effect called *rolling shutter*: fast-moving objects like airplane propellers appear distorted, and if the camera itself is moving, video looks wiggly and jello-like. [4]

Old panorama apps work by adding columns of pixels to the image as the user pans the camera. [7] Like rolling shutter, this means adjacent pixels are offset in time. Any point where adjacent pixels have a time offset introduces a temporal “seam” where changes in time of the subject turn into spatial variations in the resulting image.

Modern panorama apps in the “photo sphere” style, rather than making the user take one continuous scan, allow discrete snapshots to be taken at the user’s leisure. This makes it easier to plan and compose a panorama, but unfortunately greatly reduces the number of temporal seams where jank might appear.

3 Characteristics of Jank

These are the types of jank that can appear in a traditional panorama, and how they come to be. These are the visual properties we hope to replicate.

Duplication. If an object gets scanned, and then moves into the path of the scan again, it can get scanned a second time, showing up multiple times in the resulting image. This is what has happened to the cat’s legs in Figure 1.

Bend. Even if an object moves rigidly over time, if its velocity is nonlinear, it will appear spatially nonlinear in a panorama. This can result in objects looking rubbery and curvy. The legs of the runners in Figure 2 show this, especially the one in the top left.

Smear. Objects that move in the same direction of the camera get stretched out. Objects that move in the opposite direction get squished. Figure 3 demonstrates some extreme examples of this.

4 Method

Pandemonium allows users to build up a composite incrementally by stamping content piece by piece. Each additive stamp contains only the isolated foreground from the camera, with random sinusoidal distortion applied, and is smoothly blended with the previous content.

The distortion on stamps creates **bend jank**. Repeated stamping of the same object creates **duplication jank**. The smooth blending creates **smear jank** at the seams of stamps.

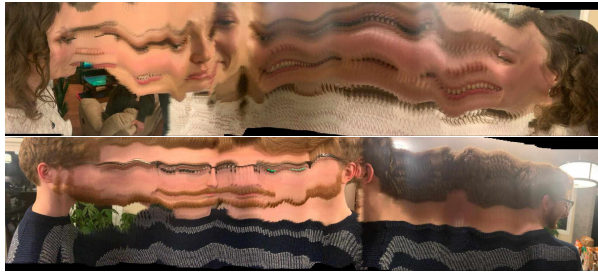


Figure 3: Smear jank in a slit scan panorama caused by a subject moving with the camera as it pans while also rotating in place.

4.1 Flow

The app keeps track of three image buffers: the background layer, the foreground layer, and the “scratch” layer previewing what the result will be when a snapshot is taken. The background layer captures the camera in the first snapshot. On the scratch layer, we preview the result of stamping the snapshot. When a snapshot is taken, we simply replace the foreground layer with the scratch buffer. Figure 5 shows this process in a control flow diagram.

When creating a composite, we run the camera feed through `SEGMENT()`, which isolates the foreground from the background. We then run the result through `WARP()`, which applies a sinusoidal offset to the pixels. Finally, we combine the result with the current foreground layer using `AMALJ()`, the jank amalgamate or “amaljamate” operator.

4.2 Segmentation

The `SEGMENT()` function applies the MediaPipe body segmentation model [5] to isolate the subject. The result is an image with just the foreground on a transparent background. The segmentation mask is shown in Figure 4b,e.

4.3 Warp

The `WARP()` function applies a horizontal sinusoidal offset to each pixel in the image. The amplitude and frequency of the offset is controlled by a single param-

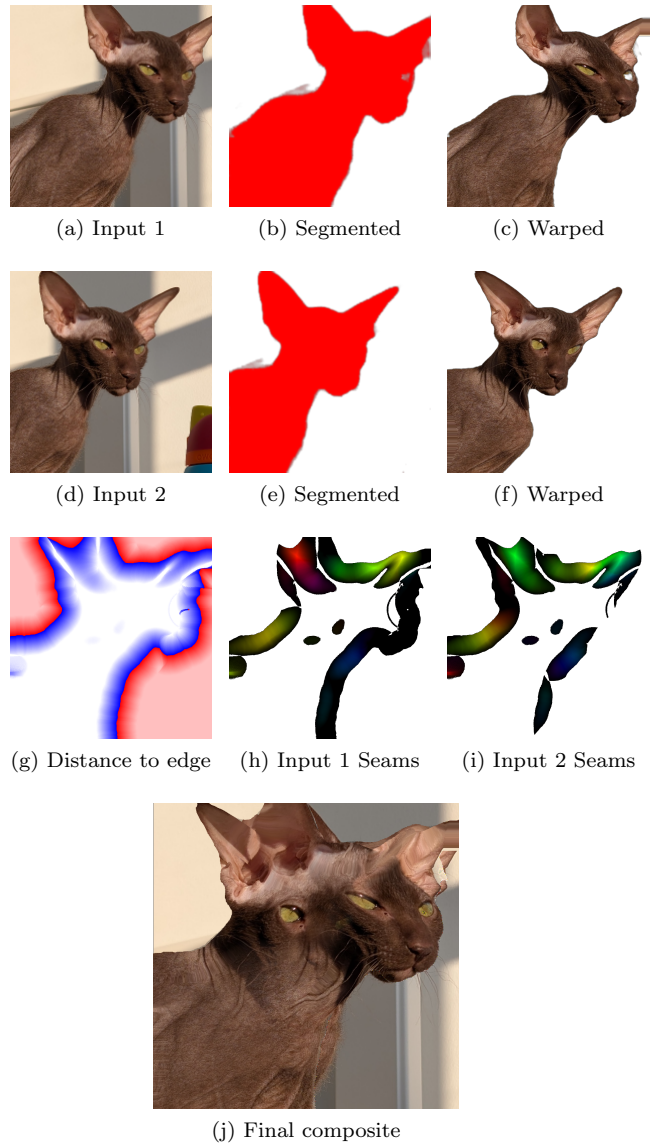


Figure 4: Given inputs (a) and (d): (b) and (e) show masks separating foreground from background; (e) and (f) show applied warp; (g) shows signed distance to the joint shape (inside is blue, outside is red); (h) and (i) show the regions from each input on the seams between the two inputs that need to smear (hue angle indicates direction, brightness indicates magnitude of smear.) The final image is shown in (j).

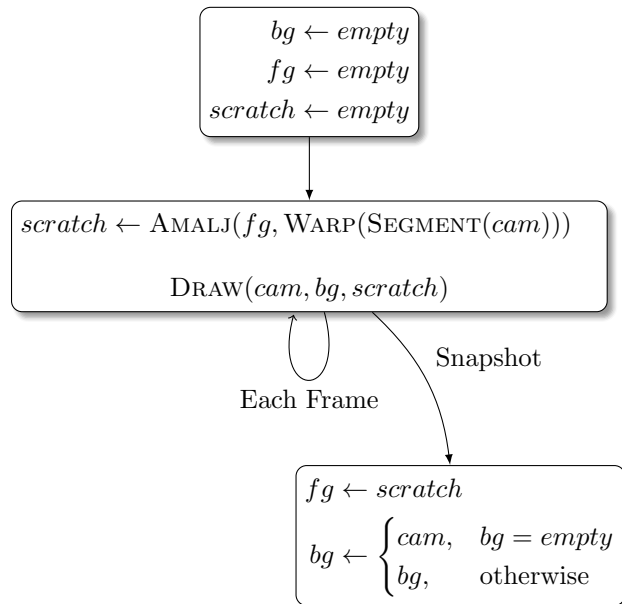


Figure 5: Pandemonium control flow.

eter $w \in [0, 1]$ for simplicity, and a constant $c \in [0, 2\pi]$ that is randomly picked for each snapshot:

$$x' = x + 0.1w \sin(20wy + c) \quad (1)$$

The warped foreground elements are shown in Figure 4c,f.

4.4 Amaljamate Operator

Our amaljamate operator is designed to run in a fragment shader in WebGL, and therefore is defined as a function run for each pixel coordinate, outputting a final merged pixel color.

Its high level goal is to merge two subjects. If the two subjects are far apart, they should appear unchanged. The goal, shown in Figure 6, is that as they grow closer together, they start to smear into each other. As they touch and combine, their smeared colors blend smoothly into each other. When two subjects fully overlap, shown in Figure 7, they have a seam all around the subjects' exteriors, leaving only the middle is undistorted.



Figure 6: As two subjects get closer together, distortion appears in the seam between them.



Figure 7: When two subjects exactly overlap, their seams follows their contour. The whole contour is distorted, and their centers are undistorted.

We first create a signed distance d_i ($i \in \{1, 2\}$) to each subject by sampling pixel locations around the origin pixel p_o to find the closest point p_i on the edge of each subject. Since it is a signed distance, a negative distance implies that the current pixel is inside the subject, while a positive distance implies it is outside. We also create a signed distance to a smooth, merged shape, d_m , by using a smooth union operator [9] that many scholars refer to as the “bean operator.” [8] We define a radius of influence r to control the maximum distortion, leading to Equation 2 for the merged signed distance:

$$d_m = \min(d_1, d_2) - r \left(\frac{\max(4r - |d_1 - d_2|, 0)}{4r} \right)^2 \quad (2)$$

The result of the smooth union distance is shown in Figure 4g, with the boundary between red and blue showing the outline of the final shape.

If we were to grab color data from the input textures at the origin pixel coordinate, $\text{texture}(t_i, p_o)$, the original images would be reproduced undistorted. We want to start to smear them as they get close to each other, so we apply an offset to p_o towards the closest edge. When a pixel on one subject is farther than r away from the other subject, it should appear undistorted, so the magnitude is 0. When it is a distance of r inside the other subject, it should also have a magnitude of 0. Where the subjects touch, there should be maximum distortion. The amount of distortion should smoothly interpolate between these extremes. We therefore base the magnitude based on the inverse of the absolute distance from any edge in Equation 3 and use it to offset the pixel coordinate used in a texture lookup in Equation 4. The angle and magnitude of offset are shown in Figure 4h,i.

$$m = r \prod_{i \in \{1, 2\}} \left(1 - \left| \frac{d_i}{r} \right| \right) \quad (3)$$

$$c_i = \text{texture} \left(t_i, p_o + m \frac{p_i - p_o}{\|p_i - p_o\|} \right) \quad (4)$$

We then produce the final output color by assigning a target weight to each input. Fully inside one

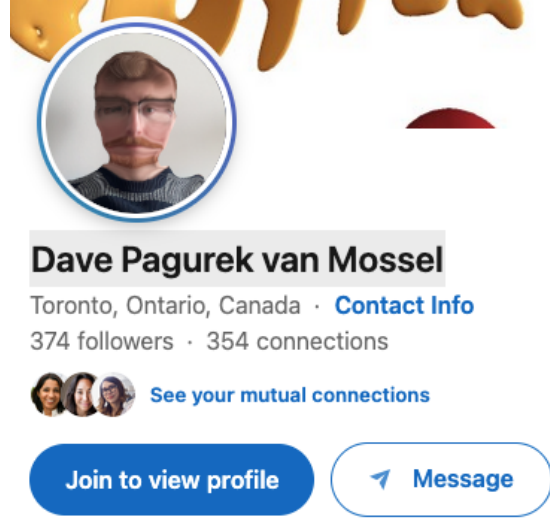


Figure 8: Figure 7 as a LinkedIn photo.

subject, the target weight is 1, and right on the outside, the target weight is 0. To get the final merged color c_m , Equation 6 uses a linear combination of the source colors using normalized weights.

$$w_i = \text{clamp} \left(\frac{d_i}{-r}, 0, 1 \right) \quad (5)$$

$$c_m = \frac{w_1}{w_1 + w_2} c_0 + \left(1 - \frac{w_1}{w_1 + w_2} \right) c_1 \quad (6)$$

5 Results and Analysis

To evaluate Pandemonium, we asked users to take photos with it to see what creations they would make using the jank characteristics described in Section 3.

Figure 10 shows different approaches to the self portrait. Compared with a traditional headshot, these images shock the viewer and provide a significantly more memorable viewing experience. For this reason, Pandemonium self portraits make for great LinkedIn photos.

Figure 11 demonstrates the possibilities of group photos: when faces are unrecognizably smeared into each other, the result is much more memorable than a straightforward portrait would have been.



Figure 9: A composite of two Timothée Chalamet pics from *A Complete Unknown* (2024).



Figure 10: Self portraits using different forms of jank

Figure 12 shows use of duplication that creates a sense of progression using layering that would be impossible in a slit scan panorama. The left image creates a sense of anticipation before an action. The right image shows an Animorphs-style [6] transformation from a regular human to a human eating noodles.

Finally, Figure 13 exhibits the breadth of emotions possible by combining figures. The left image reminds the viewer of a yin/yang dichotomy, and the right image evokes papal or saintly feelings.

6 Conclusion

Pandemonium breaks panorama apps out of a cycle of stagnation and creates truly memorable images. As Timothée Chalamet says in *A Complete Unknown* (2024), anyone who wants to attract attention has “gotta be a freak.” [3] With Pandemonium, it’s easier than ever to be a freak.



Figure 11: Group photos compositing multiple exposures of the same group



Figure 12: Uses of duplication to create unique effects



Figure 13: How do these images make you feel?

References

- [1] File:Bundesarchiv BILD 183-S0702-0019, 100m-Weltrekordlauf Marlies Oelsner, zielfoto.jpg - Wikimedia Commons. URL: https://commons.wikimedia.org/wiki/File:Bundesarchiv_Bild_183-S0702-0019,_100m-Weltrekordlauf_Marlies_Oelsner,_Zielfoto.jpg.
- [2] Panoramic photos of cats gone terribly wrong. URL: <https://cheezburger.com/9234693/panoramic-photos-of-cats-gone-terribly-wrong>.
- [3] A Complete Unknown (2024) — Biography, drama, music, 12 2024. URL: <https://www.imdb.com/title/tt11563598/>.
- [4] Wikipedia contributors. Rolling shutter, 12 2024. URL: https://en.wikipedia.org/wiki/Rolling_shutter.
- [5] Ivan Grishchenko, Valentin Bazarevsky, Ahmed Sabie, and Jason Mayes. Body Segmentation with MediaPipe and TensorFlow.js. URL: <https://blog.tensorflow.org/2022/01/body-segmentation.html>.
- [6] Sean Hollister. Here's how a '90s app made those transforming Animorphs book covers. 2 2019. URL: <https://www.theverge.com/2019/2/17/18228803/animorphs-book-cover-morphing-elastic-reality-lgr>.
- [7] Paul Kuehnel. Shooting great panorama photos with the Samsung Galaxy Nexus. 1 2012. URL: <https://www.ydr.com/story/tech/blogs/greenmesh/2012/01/07/samsung-galaxy-nexus-panorama-photo/31599643/>.
- [8] Pagurek, Dave and Banic, Milica. Bean there, done that: Computer-Assisted Design of Bean Sculptures. In *Proceedings of SIG-BOVIK*, 2024. URL: <https://www.davepagurek.com/programming/bean-there-done-that/>.
- [9] Inigo Quilez. Smooth Minimum, 2013. URL: <https://iquilezles.org/articles/smin/>.