Image Stacks

Michael F. Cohen Alex Colburn Steven Drucker
Microsoft Research Technical Report
MSR-TR-2003-40

Abstract

We present a simple but powerful Image Stack process for creating an enhanced image from a *stack* of registered images. This paradigm combines pixels using multi-image operations on a set of images of the same subject matter. We demonstrate how Image Stacks can help create group photographs, enhance high dynamic range images, combine images captured under different lighting conditions, remove unwanted objects from images, and combine images captured at different times and with different focal lengths.

1 Introduction

Taking group photographs can be frustrating because capturing a single image in which everyone is smiling and has their eyes open is nearly impossible (e.g., the top 4 pictures in Figure 1). Most photographers take a series of photographs hoping to capture at least one satisfactory image of the group. However, this approach may never yield such an image. On the other hand, within the series of images, it is likely that at least one good image of each individual within the group will be captured. A group photograph could be created by combining the best portions of each individual image into a single composite image such as the result at the bottom of Figure 1. This is only the simplest idea for ways to combine images.

This paper presents an Image Stack process for easily combining individual images into an enhanced composite image. An *image stack* is a set of identically sized registered images that may originate from any stationary still or video camera. If a stationary camera was not used to capture a set of images, the images may be registered by manually aligning them or using an automated registration procedure [1]. Image Stacks also provides *filters* that may be applied to the 3D image stack to create new 2D intermediate images. A user then selects at least one *source image,* either one of the original images or an intermediate image, from which pixel values may be composited (or painted) to a new resultant image. The user may successively select as many source images as desired to create the final composite image.

In addition to improving group photographs, Image Stacks may be used for a variety of applications such as enhancing high dynamic range images, combining images captured under different lighting conditions, removing objects from images, and combining images captured at multiple points in time or with different focal lengths.

Some of the individual techniques and ideas used within Image Stacks have been seen before in different settings. However, to date no one has combined all of the methods under a simple user interface for combining multiple images. For example, the *rubber stamp* tool in Photoshop or *clone brush* in PaintShop Pro foreshadow the uses of brushes to move "paint" from one image to another. Massey and Bender's work in *Salient Stills* [6] introduced the idea of using multiple images in creative ways. However, they do not discuss the breadth of techniques we describe or provide a simple UI to bring them together. Lastly, some of the techniques implemented within Image Stacks were borrowed from works concerning high dynamic range images [3], matting techniques [2], and basic image processing methods.

Figure 1: A camera mounted on a tripod was used to capture a series of still images of a family. Four of these still images are shown above. The bottom image was created by selecting sections of the original images and painting those sections into a new composite image. This composite image was created in about four minutes while the family looked over the author's shoulder.

2 Image Stacks

Figure 2 gives a flow diagram of the Image Stack process. A user applies filters to the image stack to create new images referred to as *intermediate images*. A user can then select one of the original images or an intermediate image to serve as a source image from which pixels may be composited or painted into a new resultant image. A painting paradigm is used to select which pixels from the source image are added and how these pixels are painted into the resultant image.

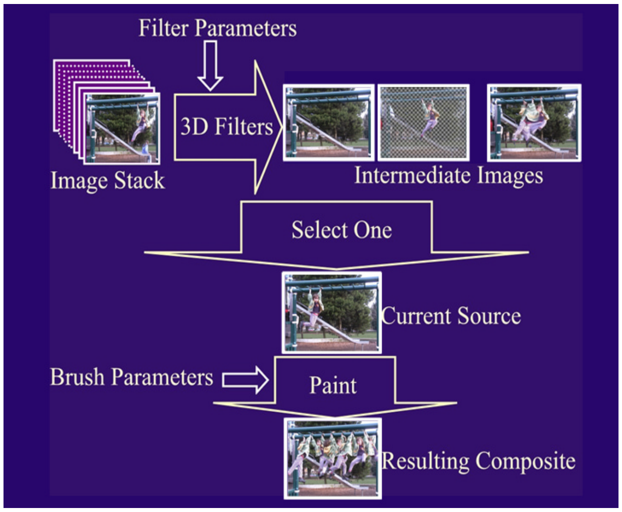

Figure 2: The Image Stacks Process

2.1 3D Filters

The Image Stack may be thought of as a 3D cube-like set of pixels *I(x,y,z)*. Filters are functions capable of mapping from a 3D volume to a 2D image. At each *(x,y)*, we refer to a *span* of pixels over all values of *z*. We have created a variety of 3D filters including:

- *Median(x,y)* returns *I(x,y,z)* where *z* is the depth with the median luminance along the span at *(x,y)*. A more expensive *MaxHist* filter returns the pixel with the minimum sum of squared distances in RGB space to all other pixels in the span. These filters are useful for removing objects found in only one or a few images. *MinHist* does just the opposite of the MaxHist filter by returning the pixel furthest from all others.
- *MaxY(x,y)* simply returns the pixel in the span at *(x,y)* with the maximum luminance, *Y*.
- *MaxContrast(x,y)* returns the pixel in the span at (x,y) that has the highest contrast in a small (5x5 pixel) neighborhood around it. This filter was inspired by Koenderink and Pont's "Texture at the Terminator" [5]. This filter has proven useful when combining images under different light conditions or taken at multiple focal lengths.
- *TemporalSmooth/Sharpen(x,y,z,dz)* returns, as its name implies a low/high pass filtered version of the image stack. Parameters for this filter include *z* and *dz*, the center and width of the filter along the span.
- *HighDynamicRange(x,y,map(R→Y))* assumes the input stack has images with different exposures and exposure information contained in the XIF portion of jpeg files. This filter computes a radiance value for each pixel [3] and then tone maps the radiance back to the gamut of the monitor via a user defined map (automatic tone mapping can also be used as in [4][8]). The map can be adjusted in real time as the user observes the resulting intermediate image.
- *Slice(x,y,z(x,y))* returns the pixel *I(x,y,z(x,y))* where *z(x,y)* defines a surface through the image stack volume. Two surface design tools have been implemented. One creates a planar slice by adjusting rotations about the *x* and *y* axes and translating in *z*. The other produces a swept surface by defining a curve *z(x)* for all values of *y*. Each of these can be defined interactively and the resulting intermediate image produced in real time.
- $Mat(C_1(x,y), C_2(x,y), \alpha(|C_1(x,y)-C_2(x,y)|))$ takes two images and a matting function as input and produces a *mat* of the first image, C_1, with its alpha channel modified.. The *Mat* filter is particularly useful for adding specific temporal events to the final composite image without erasing earlier edits. The median image typically provides the comparison image, C_2. Generally, this *matting* operation is complex (see [1] for a detailed discussion and results). We provide a simple approach to define a mapping from color differences to alpha in which the user adjusts two parameters, an absolute difference and a smoothing factor defining a smoothed step function.

2.2 Brushes

The composite image is created by *painting* pixels from one or more source images into a new resultant image. A user may perform the painting operation by using a 2D *brush* to select which pixels in the source image are painted into the resultant composite image. While a huge variety of possible 2D brush shapes and alpha cross-sections are available (see Photoshop and similar competing products), our focus is the 3D filtering operations; therefore, we will not dwell on brush features. Two types of brushes were used to create the composite images presented in the next section. The first transfers *all* of the pixels from the source image to the composite image. The second brush is circular with a user-defined radius. When using either brush, the user can specify the brush's alpha for compositing with an *over* operation.

3 Results

The results shown in Figures 1 and 3-8 were created with Image Stacks from sets of still and video images. Some represent "one button" results from a single filter, others combine filters and brushing. The composite images presented required between one and ten minutes of user interaction each.

4 Discussion and Future Work

We are working to extend the set of available filters and brushes. We also intend to integrate vision based registration tools. As digital imaging advances, many of the filter operations could also take place in the camera (e.g., multiple simultaneous exposures as in [7]). Image Stacks has already changed the way the authors perceive and practice the art of photography. You don't have to settle for only what you see *now*.

5 References

[1] J. R. Bergen, P. Anandan, K. J. Hanna, R. Hingorani, *Hierarchical Model-Based Motion Estimation*, Second European Conference on Computer Vision, 1992

[2] Yung-Yu Chang, Aseem Argarwala, Brian Curless, David Salesin, Richard Szeliski, *Video Matting of Complex Scenes,* SIGGRAPH 2002

[3] Paul Debevec, Jitendra Malik, *Recovering High Dynamic Range Radiance Maps from Photographs,* SIGGRAPH 1997

[4] Frédo Durand and Julie Dorsey, *Fast Bilateral Filtering forteh Display of High-Dynamic-Range Images,* SIGGRAPH 2002

[5] Jan Koenderink and Sylvia Pont, *Texture at the Terminator,*

[6] M. Massey and W. Bender, *Salient Stills: Process and practice,* IBM Systems Journal, Vol. 35, Nos. 3&4

[7] S. K. Nayar and T. Mitsunaga, *High Dynamic Range Imaging: Spatially Varying Pixel Exposures,* Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, June 2000.

[8] Erik Reinhard, Michael Stark, Peter Shirley, James Ferwerda, *Photographic Tone Reproduction for Digital Images,* SIGGRAPH 2002

Figure 3: Two stills from a video are shown above. The bottom left most image is the median of all images. Note that although a child appears in all of the original images, the child is absent in the median. The strobe-like composite on the right was created by using a brush to add mats (like the one shown at the bottom right) to the composite image. The mats were created by comparing the individual frames to the median.

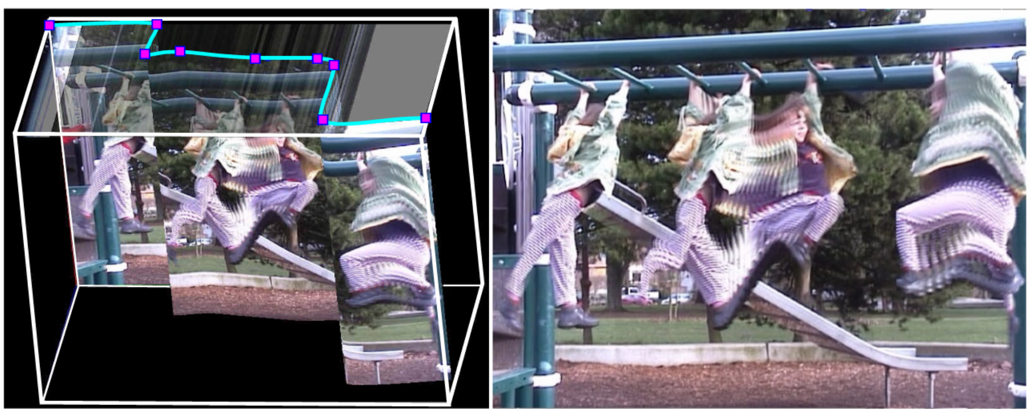

Figure 4: The same video used in Figure 3 were used to create a video cube shown left. In this figure, the user defined a function z(x) by interactively constructing a Hermite cubic curve. The UI allowed the video cube to be tilted so the user could draw a curve on the top surface of the volume. The result at the right is the surface as seen head on.

Figure 5: High Dynamic Range imagery is created from an image stack of stills taken at different exposures. Two of four input images are shown at left. A radiance image is first computed following Debevec et al [Debevec96]. The user created an initial tone mapping by defining a function from radiance to gamut luminance. A result of this tone mapping is shown in the third panel from the left. The brush tool was then used to paint lighter flags from the image in the second panel and darker stained glass from an intermediate exposure (not shown) onto the resultant image (fourth panel). The combination of automatic HDR methods and interactive tone mapping provides a power tool for combining multiple exposures.

Figure 6: Lighting a stationary object to bring out all of its detail is a difficult task requiring great skill. The four images (of 15 images captured) shown to the left were taken while moving a desk lamp around a bronze sculpture. Each image enhances specific regions but none provides adequate overall detail. The center panel is a one button result created by applying the MaxY filter to the image stack. In this image all of the highlights are included and the shadows have disappeared from the background. The right image depicts a one button result created by applying the MaxContrast filter to the image stack. The increased local contrast adds detail to the surface of the sculpture. Note that neither of these results could have been created using real world lighting configurations. The texture in the background of the second image can be easily "brushed out" by combining these two images.

Figure 7: The four images at the top were captured by moving the camera to the left and right, and up down a few feet. The images were then registered manually to align the background mountains. The right hand resulting composite was created by first invoking the MaxY filter which effectively removed all the wires but also included multiple ghosted versions of the foreground. The upper right of the four small images was then selected to brush in the foreground.

Figure 8: Two images from a 120 frame video sequence are provided in the left most panels. These two images were captured at different focal lengths. The upper right image is a one button result created by applying the MaxContrast filter to the image stack. The MaxContrast filter successfully selected pixels in focus for most regions of the image. The fourth image was created by brushing texture from another image onto the MaxContrast result thereby removing artifacts in areas with low texture content such as the walls in back.