

# Smooth Transitions for Large Scale Changes in Multi-Resolution Images

M. Lancelle<sup>1</sup> and D. W. Fellner<sup>1,2</sup>

<sup>1</sup>Institute of ComputerGraphics and KnowledgeVisualization (CGV), TU Graz, Austria

<sup>2</sup>GRIS, TU Darmstadt & Fraunhofer IGD, Darmstadt, Germany

## Abstract

Today's super zoom cameras offer a large optical zoom range of over 30x. It is easy to take a wide angle photograph of the scene together with a few zoomed in high resolution crops. Only little work has been done to appropriately display the high resolution photo as an inset. Usually, to hide the resolution transition, alpha blending is used. Visible transition boundaries or ghosting artifacts may result. In this paper we introduce a different, novel approach to overcome these problems. Across the transition, we gradually attenuate the maximum image frequency. We achieve this with a Gaussian blur with an exponentially increasing standard deviation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

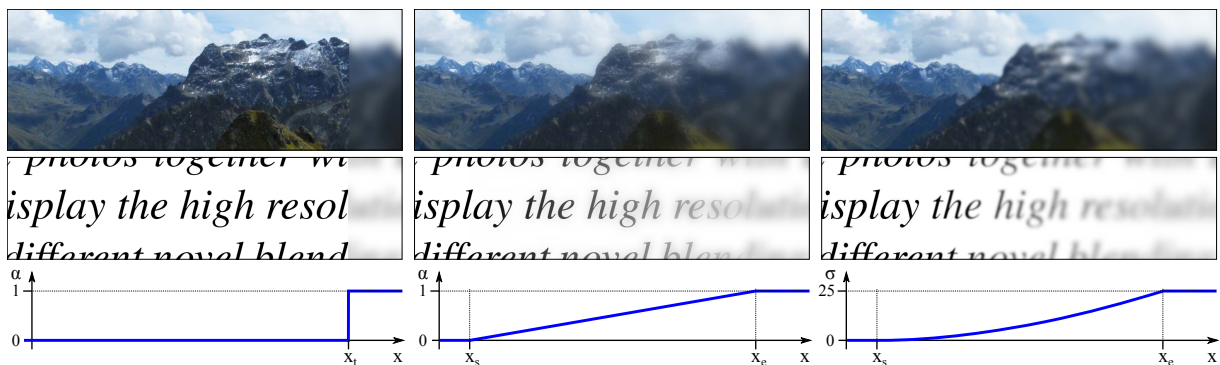


Figure 1: Two sample images with transitions from high to low resolution. Hard transition, standard linear blending and our method. The plots below show the alpha values for blending in the left and center and the standard deviation (in pixels) used for the Gaussian blur on the right. The images have a width of 1024 pixels.

## 1. Introduction

Nowadays it is easily possible to interactively display gigapixel images. Zooming by a large factor and panning the image is an important application, considering that computer displays can only show a small fraction of the data at the same time.

For the application of photographically acquired images, the creation requires the combination of many tiles and is still a long or expensive process. On the other hand, the increasing optical zoom ranges of consumer cameras allow to

easily take high resolution zoomed in inset photos together with a wide angle photo of the same scene. Often, there are some obvious points of interest like a distant village or a mountain top. It is not so important to acquire the whole image in a high resolution.

While automatic stitching is possible, the transition from high resolution content to low resolution content may be annoyingly apparent, especially when the resolution difference is high. When gradually alpha blending over a large area, the transition is less visible, but a lot of the high resolution information is lost and ghosting artifacts may result.



Figure 2: Transition generation workflow. Aligned and photometrically corrected low resolution and high resolution images are the input images on the left. The center shows an intermediate result where high resolution patches have been copied to areas with similar low frequency content. The transition map is derived from the blurred alpha mask of the high resolution input image and controls the amount of blur. The final result is shown on the bottom right.

In the following sections we will focus on photographic imagery, however our solution for the transition can be applied to all kind of data.

## 2. Related Work

In photography, vignetting, radial lens distortion as well as different exposure and white balance settings may lead to visible seams, even with the same image frequencies present. These effects are well known and studied, especially in the field of panoramic stitching. They can be corrected automatically to a great extent, see [Sze05] for a comprehensive presentation of these algorithms.

The capturing and interactive display of gigapixel images is described in [KUDC07]. Also, Microsoft’s Deep Zoom technology allows such interactive exploration in a web browser [Dee] by only streaming the necessary resolution and image region, hence avoiding the download of the whole data set. It also allows sparse compositions with different resolutions, but currently there is a hard, very visible spatial transition.

To not loose too much original high resolution data when using a wide transition, the area of high resolution can be extended by filling in plausible high frequency content. This is

the area of super resolution algorithms. For our purposes, Example Based Super Resolution [FJP02] is particularly well suited, as the high resolution image can serve as a training set.

Sunkavalli et al. also address the problem of sharp image frequency changes in compositing by texture generation [SJMP10]. However, it is not our goal to generate high frequencies throughout the whole low resolution image.

Han et al. [HH10] optimize continuity for temporal transitions from one data set to the next when zooming in. They point out the problem of ghosting and blurring that occurs when linearly blending between different resolutions. Their solution is to replace the low frequencies of the high resolution content with the ones from the low resolution content. Thus, they optimize the temporal transition from one low resolution data set to another high resolution data set. However, the border of the high resolution data set has a hard transition to the low resolution data set.

Eisemann et al. present the photo zoom project [EESM10] which is most related to our work. They use SIFT features to automatically find a homography between images of different resolutions within a certain amount of zoom range, do a detail transfer to extend regions with high resolution and

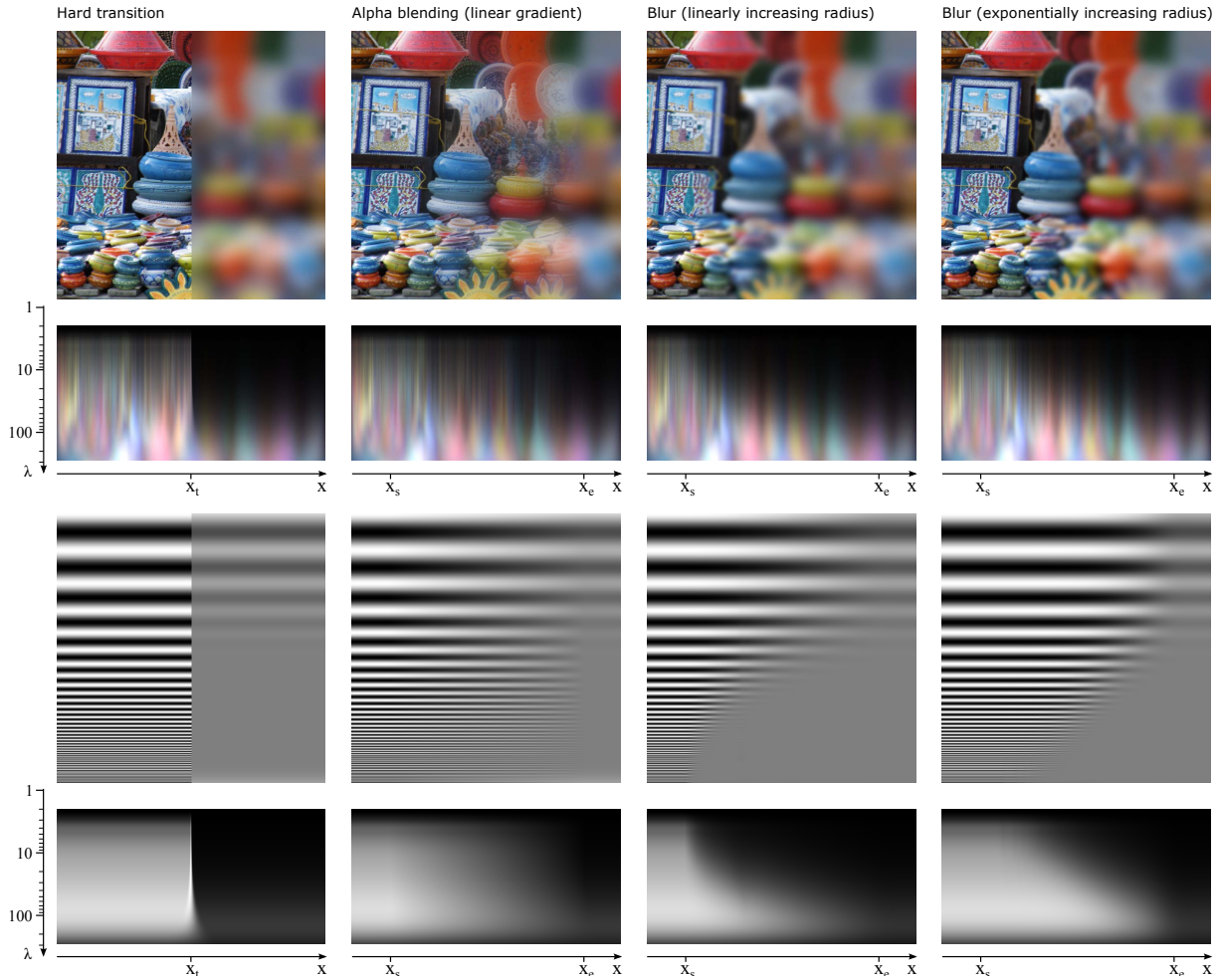


Figure 3: Frequency analysis for the market image and for an image with a sine with exponentially increasing frequency. The rightmost column shows our suggested method. For details, refer to the description in section 4.

finally generate a content dependent blend mask to get less ghosting artifacts. This irregular blend mask is generated by a cost function growing from the image border, growing stronger at edges in the image.

Ray et al. address the problem of texture continuity on 3D meshes, so that also interpolation in case of magnification leads to continuous seams in case of a magnified view [RNLL10]. They can also handle resolution differences, but the transition still occurs within a pixel of the fine resolution.

### 3. Alignment and Detail Generation

In this section we present our application in photography and show how we generate the input images for the smooth transition, explained in the next section.

We focus at panoramic images, especially allowing different focal lengths. We assume, that photos are approximately

taken from the same place and the image content does not change too much over time. The focal length in the EXIF data can be used to find the scale difference. For a first match of the position, the high resolution image can be downsampled to match the low resolution image and a simple cross correlation based search will likely find the correct position. Other parameters like rotation or slight perspective misalignment can be optimized to find an optimal homography. We have not implemented this yet. In the meantime, we use a tool for manual alignment by freeform deformation.

We avoid possible problems with different exposure or white balance by replacing the low frequency content of the high resolution image by the low resolution image. We do this by blurring both images with a large radius and then adding the difference to the high resolution image.

With the steps described above, we can handle hand held photos with different exposure and white balance settings. At the moment, we only implemented our algorithms for

images with 8 bits per channel, but they can also be used for high dynamic range images.

To enlarge the area with high resolution information, we implemented a simple detail transfer (see figure 2). We compute a good cross correlation based match of random position pairs and copy high resolution patches. However, we recommend to use a more sophisticated approach like the one described in [FJP02] to get more plausible results.

Having geometrically and photometrically matching content, at a magnified view the transition between the images is still apparent (see figure 1, left), because the image frequencies are different. Using a gradual linear blending between both resolutions (figure 1, center) leads to ghosting and blurring artifacts, especially visible in the text example.

#### 4. Transition

To help understand the present image frequencies across a transition, we perform a space-frequency analysis [HHH09] by looking at sine kernel responses with varying frequencies (see figure 3). As kernel we use the sine function multiplied with a Tukey window function, resulting in a Gabor filter as convolution kernel. Similarly, we generate a rotated version by 90 degrees, resulting in the kernels below.

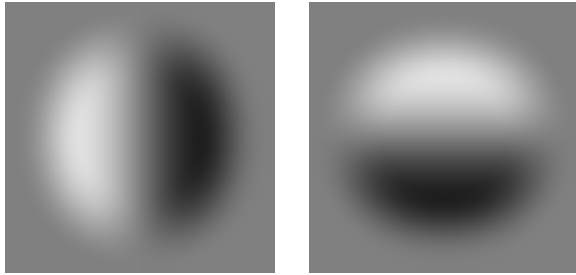


Figure 4: Gabor convolution kernels used for frequency analysis.

For each pixel in each row in the image, we add the absolute values of the convolutions with each kernel. The result is shown below the input images, with kernel dimensions of  $\lceil \lambda \rceil \times \lceil \lambda \rceil$ . For our example images, we use a high resolution image and generate a low resolution version by applying a large Gaussian blur. Note that this is an approximation, as common upscaling algorithms will produce slightly different interpolation results.

#### 4.1. Transition with Alpha Blending

As expected, the linear alpha blending frequency response in figure 3 shows, that high frequencies are present almost across the entire transition and then quickly drop to zero on the right side, leading to a relatively hard transition in the frequency domain.

Especially in the middle of a gradual transition, ghosting appears, i.e. edges look foggy.

#### 4.2. Transition with Frequency Blending

Our approach is to gradually suppress high frequency details across the transition. We use a Gaussian blur with an exponentially growing standard deviation (with a kernel radius of 2.5 times the standard deviation). We first tried a linearly growing radius and found, that the blur increases too sudden and high frequencies present drop to quickly at the start (left side) of the transition. We found that an exponentially growing standard deviation works better (see both right columns in figure 3).

As our images are stored in sRGB space we approximately linearize them with a gamma value of 2.2 and transform the result back appropriately. Note that if this linearizing was omitted, averaging of the non-linear intensities is incorrect for blurring and results in too dark images. To compute the required standard deviation of the Gaussian from the magnification factor, we found that a factor of 0.5 produces a blur that is similar to the magnified low resolution content, i.e.  $\sigma = 0.5 \cdot \text{magnification}$ .

Realtime visualization software often uses hardware magnification with linear interpolation. In general, the transition between the Gaussian blurred image and the linearly interpolated image will be slightly visible. Also, the high resolution inset is in general not aligned to the low resolution coarse pixel grid. To avoid a visible seam, we alpha blend the final result with a smooth gradient over a small distance. To examine the artifact, zoom in to figure 7, where we did not use blending in order to achieve a better image compression and a lower file size for this PDF document.

##### 4.2.1. Implementation Details

We ran our first tests with the Focus Blur plugin [GIM] for The GIMP. Unfortunately, there seems to be a bug in our version, leading to a spacial shift with a higher radius. Also, applying the exponential function to the 8 bit channel leads to rounding errors visible in the output. The Adobe Photoshop CS5 Lens Blur filter is similar but does not offer a Gaussian Kernel. Thus, we implemented the algorithm in C++.

As the algorithm is a preprocessing step, computation time is less critical. In our experiments on a standard PC, the computation usually took from a few seconds up to some 20 seconds. For high resolution images and very high magnification factors, the necessary kernel becomes large and computation time goes up quickly (i.e. many minutes). Note that due to the changing radius, the convolution is not separable.

To speed up computation, it is possible to build a blur stack, blurring the entire image with a constant standard deviation (which is separable). During computation, the closest less blurry image from the blur stack can be used to achieve the final amount of blur, needing a much smaller kernel. This is possible because multiple, successive Gaussian blurs result in a Gaussian blur with greater standard deviation. We have not implemented this idea yet.

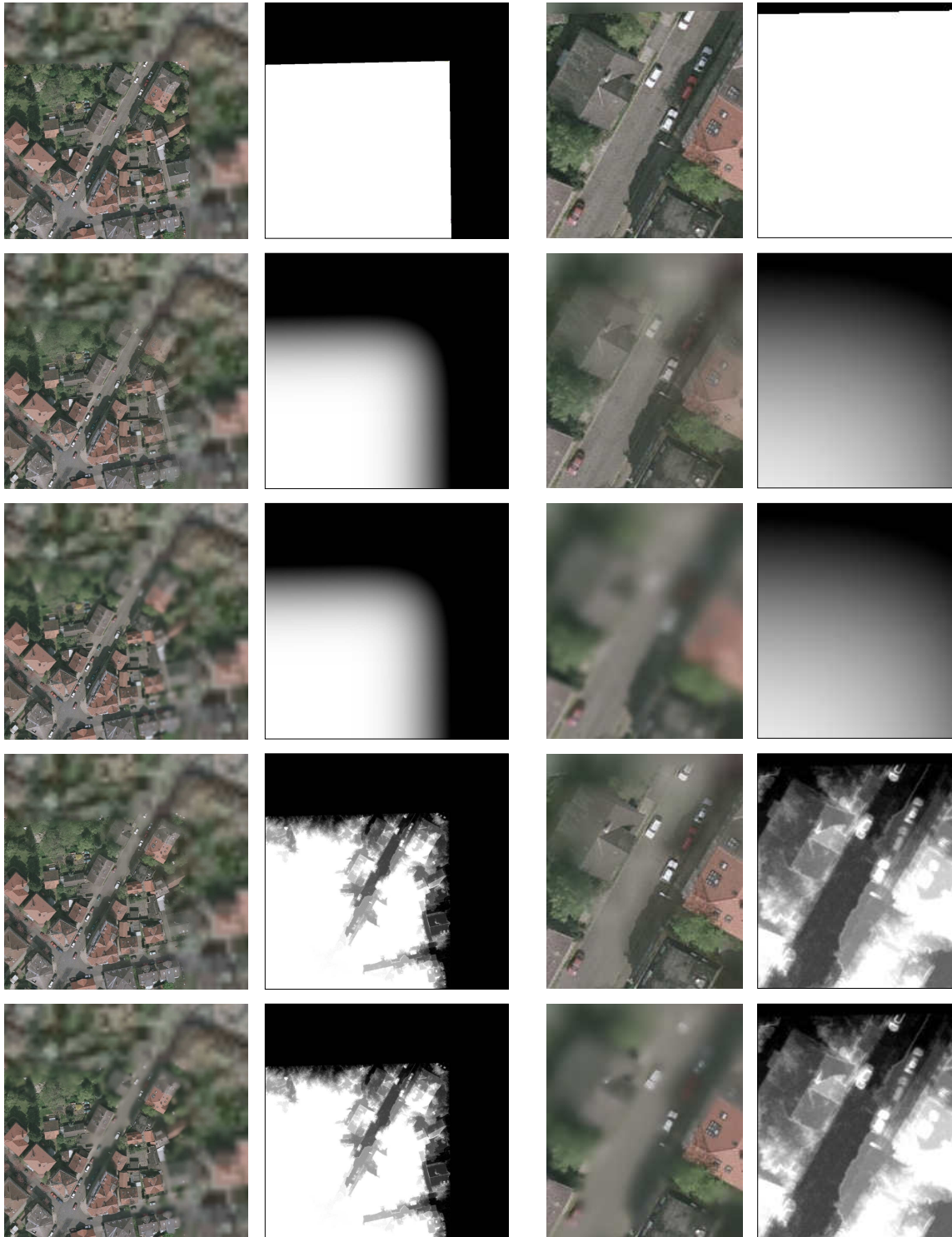


Figure 5: Aerial image of a city. The left columns shows the complete image with the respective maps, the right columns show a zoomed region close to the center. From top to bottom: hard transition, linear blending with smooth transition, our method with exponentially increasing blur, linear blending with irregular mask, blur with irregular map. The image is provided by the city of Braunschweig.

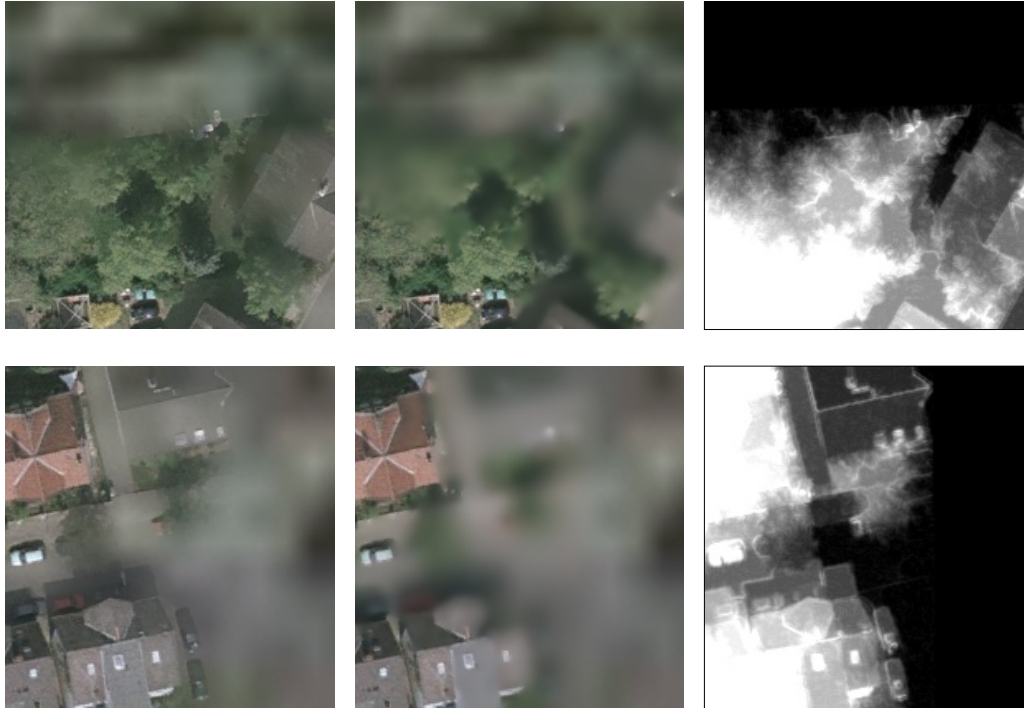


Figure 6: Two more closeup views from figure 5, comparing irregular masks/maps (right) for linear blending (left) and blur (center). Artifacts are visible in all cases. Top left: hard edge of a roof. Top center: some trees are only partially very blurry. Bottom left: cars at the bottom have ghosting. Bottom center: only some parts of the house are very blurry.

### 4.3. Irregular Masks

In [EESM10], Eisemann et al. suggest to use irregular masks. The idea is to get a sharp transition at object boundaries and hence avoid the ghosting artifacts. As this is a different, competing solution for the same problem we want to solve, we implemented an algorithm producing similar irregular masks to further analyze that idea and compare the results.

We also use a cost function which is a constant plus a weighted edge value. The edge value is computed from the maximum absolute sobel edge value of each color channel, setting low edge values below a threshold to zero. We manually tuned the parameters to get a good mask. Figures 5 and 6 show the generated mask and enlarged crops respectively. While the idea works well in the cropped region in figure 5, the cropped regions in figure 6 show ghosting at the cars and a visible edge caused by a roof on the high resolution region boundary.

We also tried to use the same mask as an input to our algorithm. To avoid blurring with a large kernel over masked out details, we could multiply the Gaussian weight with the mask. Instead, we used the GIMP Focus Blur plugin which behaves similarly. While the results do not look very convincing, it may help at a coarser level to hide the regular,

usually rectangular border of the high resolution content.

### 5. Application and Results

We applied our technique to several different types of images. We successfully avoid ghosting artifacts completely. Figure 7 shows an interactive example with two blended high resolution insets. The first has a zoom factor of 5.6, the second has an additional zoom factor of 3.3 (18.4 in total, i.e.  $18 \times 18$  high resolution pixels correspond to one low resolution pixel). As the images are scaled down but still have about 1.5 megapixels, this is only visible at a high magnification, please use the electronic version of this document and zoom in. Common PDF viewers usually only seem to interpolate using the nearest neighbor, leading to pixel artifacts at high magnification levels. Even when explicitly searching for the transition, it is not easy to find. For reference and to ease inspection, the right image visualizes the position of the three images by inverting the color of the middle image.

We also display our zoomable images on our multi touch setups, as rotating, scaling and zooming images is a good demonstration application. To prevent images accidentally being scaled too large, we use a maximum scale factor. We extend this by adapting the maximum scale factor to the dimensions of the smallest zoom region, thus allowing to zoom in far enough.



Figure 7: In the electronic version, this is an interactive zoomable image. The location of the high resolution images is shown in the crop on the right. Note that for better compression and unknown magnification algorithm of your PDF viewer, in this example we do not blend the outer border.

## 6. Conclusion and Future Work

For transitions from spatially high resolution image data to a lower resolution area, a combination of high resolution monitors, low resolution images and large zoom differences lead to visible transitions at a high magnification level. We successfully improve hiding such transitions, preventing ghosting artifacts by gradually attenuating high image frequencies. This is achieved by a Gaussian blur with an exponentially growing standard deviation.

Our algorithm is part of a preprocessing pipeline, where image registration and photometric correction are the most important components. These necessary steps can already be applied automatically, but we have not yet implemented all of them, so some manual work is required. Particularly, we plan to write an automatic export of the remapped images with axis aligned rectangular borders for the SVG format and for Microsoft's Deep Zoom Composer.

Compared to a hard transition or alpha blending transition, there is usually no overhead for an interactive viewer. Only when increasing the high resolution area by superresolution, the amount of data and thus memory requirement and rendering time rise slightly. This is usually not a problem, especially when using graphics hardware for visualization.

We have not yet considered multiple partially overlapping images with different resolutions and plan to look into the generation of consistent blur maps.

## References

- [Dee] Microsoft deep zoom, <http://zoom.it/>. 2
- [EESM10] EISEMANN M., EISEMANN E., SEIDEL H.-P., MAGNOR M.: Photo zoom: High resolution from unordered image collections. In *Proc. Graphics Interface (GI) 2010* (Ottawa, Canada, 2010), pp. 71–78. 2, 6
- [FJP02] FREEMAN W. T., JONES T. R., PASZTOR E. C.: Example-based super-resolution. *IEEE Comput. Graph. Appl.* 22 (March 2002), 56–65. 2, 4
- [GIM] Focus blur gimp plugin, <http://registry.gimp.org/node/1444>. 4
- [HH10] HAN C., HOPPE H.: Optimizing continuity in multiscale imagery. In *ACM SIGGRAPH Asia 2010 papers* (New York, NY, USA, 2010), SIGGRAPH ASIA '10, ACM, pp. 171:1–171:10. 2
- [HHH09] HYVÄRINEN A., HURRI J., HOYER P. O.: *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. Springer, 2009. 4
- [KUDC07] KOPF J., UYTTENDAELE M., DEUSSEN O., COHEN M. F.: Capturing and viewing gigapixel images. In *ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 2
- [RNLL10] RAY N., NIVOLIERI V., LEFEBVRE S., LÉVY B.: In-visible seams. In *EUROGRAPHICS Symposium on Rendering Conference Proceedings* (2010), Lawrence J., Stamminger M., (Eds.), Eurographics, Eurographics Association. 3
- [SJMP10] SUNKAVALLI K., JOHNSON M. K., MATUSIK W., PFISTER H.: Multi-scale image harmonization. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 125:1–125:10. 2
- [Sze05] SZELISKI R.: *Image alignment and stitching: A tutorial*. Tech. rep., MSR-TR-2004-92, Microsoft Research, 2005. 2