Deblurring Merged Images
August, 2010

Mark DeNies
DeNies Video Software

Abstract

Much research has been devoted to the problem of recovering clear images from blurry ones. Blurring may be caused by imperfect lenses, the image being out of focus, and camera or object motion. In this paper, techniques will be presented which work well for the particular case of deblurring (also know as deconvolving) images created by merging multiple images of the same scene, each image being taken from a different point and at a different angle. In this problem, the blur is similar to an out-of-focus blur. Several deconvolution algorithms are investigated, one of which was found to give superior results if an accurate point spread function (PSF) was given.

For this problem, a PSF which is symmetric horizontally and vertically works well. Several methods of determining the PSF from the image are presented.

The limitations of these techniques are discussed, followed by a presentation of current and future research to extend the deblurring technique to address more general situations.

The Problem

A typical video sequence contains many sequential frames of similar but not identical information. By combining these images, the additional spatial and temporal information of the sequence may be used to reduce the noise and simultaneously increase the resolution. This is done by registering all of the overlapping frames to the chosen frame to be enhanced, merging the images, and then deblurring the resultant composite image. More information on these steps can be found in the “Image Matching and Warping” and “High Resolution from Low Resolution Video” whitepapers on this website. The blur in the merged image is a combination of the original blur in each of the images added to the blur created by resizing the original frames to the target resolution. The original blur is unknown – it may be caused by motion, the scene being out-of-focus, poor optics, etc.

The primary difficulty in any deblurring solution arises from the fact that this is a classic ill-posed problem – there are potentially many solutions. To overcome this, additional constraints are imposed, limiting the solutions. Complicating the problem is the existence of noise in any image, which may be amplified by many deblurring algorithms if insufficient additional constraints are imposed. If the constraints are too strict, the solution may be little better than the original blurred image.

Many deblurring algorithms are extremely sensitive to noise and susceptible to “ringing” artifacts at sharp edges. Finding a suitable PSF for an image is difficult for most blurred images, since both its size and shape are unknown. It is not uncommon for a PSF to be quite asymmetric and even bimodal (e.g., camera shake / double image).

Deconvolution

In solving the problem of creating unblurred images from a blurred images, several deblurring methods were evaluated which derive an
unblurred image given a blurred image and a PSF. All methods which were evaluated are iterative – the error in the proposed solution for an iteration is used to improve the solution at the next iteration.

The deconvolution techniques which were evaluated were Richardson-Lucy, an accelerated Richardson-Lucy, Matlab’s blind deconvolution, simulated annealing, and a method developed by Levin and his team at MIT. To create the PSF, methods evaluated included Matlab’s blind deconvolution, a modified Richardson-Lucy algorithm that alternated finding an improved true image with finding an improved PSF, simulated annealing, and an approach using edges to determine the PSF.

To get a baseline, the classic Richardson Lucy algorithm was evaluated. A perfect image was blurred using a known PSF. Given this blurred image and the PSF, Richardson Lucy approached the true image, but is highly sensitive to the number of iterations. When too few iterations are performed, the final image is overly blurry. Too many, and there are ringing artifacts around the edges. Convergence is slow. The number of iterations needed varied from image to image.

Not unexpectedly, Matlab’s blind deconvolution is similar in result, as it is describes as an accelerated Richardson-Lucy algorithm which alternates between finding a better image and finding a better PSF.

An accelerated Richardson Lucy algorithm was significantly faster (by about a factor of four). It operated by running the basic Richardson Lucy method for a few iterations, then projecting the trend of each pixel’s value ahead (if a trend was detected). As expected, the same sensitivity to proper choice of iterations was needed.

Simulated annealing was evaluated using an evaluation function which blurred the current trial image using the PSF and returning the difference between it and the actual blurred image. A perturbation function randomly changed the values of a random number of pixels according to the current temperature. In order to make this method operate more quickly, the perturbation and evaluation were restricted to small sections of the image at a time. The resultant images tended to be grainy. Adding a “smoothness” criterion to the evaluation function improved the results. Increasing the weight of the smoothness criteria, however, resulted in greater blurriness. So, an alternate “smoothness” criterion may have better success. Even if a better smoothness criteria could be found, simulated annealing was an order of magnitude slower than the accelerated Richardson-Lucy method.

Levin and his group proposed using natural image priors to deblur the image. Specifically, this method assumes that the gradient distribution is concentrated in as few pixels as possible. That is, the image is piecewise smooth with abrupt changes. It is CPU intensive, but produces good results. As a result, this is the best technique for our purposes. This method requires a PSF to be determined by some other technique.

**Creation of a PSF**

Several techniques are available that attempt to deblur an image with little or no guidance about the PSF that causes the blur. These are known as blind deconvolution methods. Matlab’s blind deconvolution, deconvblind(), is one of these. Many of the blind deconvolution techniques described in the literature and the techniques researched in developing this paper returned deblurred images which were substantially inferior to the non-blind techniques. A possible exception is that used by Levin et al (2007). However, it is claimed to be very CPU intensive, and was not studied in this research.
One of the more widely investigated techniques is to analyze an image for edges. Specifically, the image is searched for areas where the color is relatively uniform on both sides of the boundary and gradually changes from one color to the other. Assuming that the boundary is actually a sharp edge, the PSF may be directly calculated from the gradient. Applying this analysis to all directions, good results have been achieved for the class of images deblurred in this study. By applying a requirement that the PSF show symmetry when reflected through its center, results were improved further. It was interesting to find that the PSF for deblurring high resolution frames was neither Gaussian nor flat, but somewhere between the two.

The limitation of this method is that it cannot resolve camera jitter, which generally will be characterized by an asymmetric PSF.

**Limitations**

The Levin (2007) deblurring algorithm and using edges for PSF construction works well for deblurring images which are created by registering multiple frames of the same scene. The process of registering multiple frames and merging their mean values (described in the whitepaper “Creating High Resolution Video from Low Resolution Video”) tends to reduce the image noise and to average out the effects of camera jitter. However, motion blur may vary across the video sequence, and if it does, only a portion of the motion blur can be removed.

In many sequences, there may be other types of blurring, such as out-of-focus, jitter, and poor optics. These are present in all frames of the video and should be removed prior to creating the merged image. Since this blur is often the result of irregular PSF’s, an alternate method is needed to deblur these types of images.

If any of the blur is an out-of-focus blur, the amount of blur will vary with the distance of the scene from the camera. Since the same blur is used for the entire frame, only those portions of the frame which are at an average distance from the camera will be truly deblurred.

**Future work**

The techniques found in this study are sufficient for creating video at four times the resolution of the original images. However, it is unlikely that the true PSF is as symmetrical as the one which has been generated in this technique. The camera optics (in low quality cameras) and camera jitter both will tend to create asymmetric PSF kernels.

An important unresolved issue is to discover a more robust method of automatically tuning the parameters used by the Levin/MIT algorithm. There are only three of these, so achieving this goal should be possible. For now, these parameters have been fixed at values which have worked well for the examples which were studied. Preliminary indications suggest that the parameters should be changed as the solution converges: the error parameter should be smaller, and the number of iterations should increase.

Deblurring the original image prior to merging images will likely improve the registration of images. In single images (e.g. photos), the PSF may be of any shape and can be caused by motion (camera or objects), out-of-focus blur, imperfect lenses, and camera jitter.

In video sequences, the direction of the motion blur can be inferred from the images, reducing the determination of the motion PSF to determining the magnitude (i.e., the exposure length).

Another technique which has shown promise for single images is to search for areas which show an object displayed against a uniform background, and then search for the PSF which
concentrates the non-background colors into the minimum number of pixels. In trials, a simulated annealing algorithm was used to search for the PSF with some success. Other iterative searches may perform better.

Determining the PSF from edges has been more difficult when the blur is due to jitter and focus errors. Focus errors are especially problematic since the size of the PSF is dependent on the distance from the camera. Thus different parts of the image need to be deblurred using different PSF’s. One possible solution is to partition the image into sections and derive PSF’s for each section. Or, for video sequences, 3D information may be derived from the sequence using stereo pairs of images, and using this to assist in discovering the PSF.

References


