Multiframe combination and blur deconvolution of video data

Timothy F. Gee[*], Thomas P. Karnowski, Kenneth W. Tobin

Image Science and Machine Vision Group[†], Oak Ridge National Laboratory, Oak Ridge, TN 37831

ABSTRACT

In this paper we present a technique that may be applied to surveillance video data to obtain a higher-quality image from a sequence of lower-quality images. The increase in quality is derived through a deconvolution of optical blur and/or an increase in spatial sampling. To process sequences of real forensic video data, three main steps are required: frame and region selection, displacement estimation, and original image estimation. A user-identified region-of-interest (ROI) is compared to other frames in the sequence. The areas that are suitable matches are identified and used for displacement estimation. The calculated displacement vector images describe the transformation of the desired high-quality image to the observed low quality images. The final stage is based on the Projection Onto Convex Sets (POCS) super-resolution approach of Patti, Sezan, and Tekalp.[1] This stage performs a deconvolution using the observed image sequence, displacement vectors, and an *a priori* known blur model. A description of the algorithmic steps are provided, and an example input sequence with corresponding output image is given.

Keywords: Blur Deconvolution, De-interlacing, Super-resolution, Motion Estimation

1. INTRODUCTION

Restoration of video data is important for a number of applications, including surveillance video processing, motion picture restoration, advancements to video capture electronics, upsampling for higher-resolution television monitors, and the removal of video compression artifacts. In this paper, we discuss software developed at Oak Ridge National Laboratory (ORNL) which can be used in the processing of surveillance video. This type of restoration software can aid law enforcement personnel when they attempt to identify a person in a criminal investigation.

In this application, the analyst performing the restoration may or may not be an expert in image processing. That person may restore one video a year or several in a day. The analyst is available to assist the processing, but automation is advantageous. Since the restoration occurs well after the incident, real-time processing is generally not necessary. Therefore, an iterative process is acceptable. In the approach taken here, multiple video frames are used to obtain one high-quality still image. This might be applied successively to restore the entire sequence, but generally that is not the goal in forensic restoration, and the computational cost would be large with current desktop computing hardware.

The main causes of degradation to surveillance video are the same as for general video and to some extent photography. They are motion and optical blur, noise, and low resolution. In surveillance video these problems can be a particularly great concern. Motion blurring is often present due to fast movements of a suspect. Optical blur is a problem since surveillance cameras are often expected to monitor a large area with a long depth-of-field. Noise is aggravated by low-cost cameras and excessive reuse of videocassettes. Lastly, resolution is limited by the finite

[*] Correspondence: Email: geetf@ornl.gov; WWW: http://www-ismv.ic.ornl.gov/~gee/; Telephone: 865 574 0338; Fax: 865 574 6663

[†] Prepared by OAK RIDGE NATIONAL LABORATORY, Oak Ridge, Tennessee, 37831-6285, managed by LOCKHEED MARTIN ENERGY RESEARCH CORP. for the U.S. DEPARTMENT OF ENERGY under contract DE-AC05-96OR22464

number of pixel cells, and in surveillance imagery, a small increase in resolution might make the difference in gaining key visible features in a crime scene.

Solving for the final high-quality image follows the POCS architecture outlined by Patti, Sezan, and Tekalp.[1] The POCS method enables a variety of constraints to be placed on the high-quality image estimate. Initially an image is created by interpolation to the higher resolution. Then the image estimate is iteratively projected toward the constraint sets. One such constraint is clipping the intensity amplitude so that it falls within an acceptable range, such as 0 to 255. However, the most important constraint is that the image estimate must be able to create the observed images given a model of the video system. This requires creating an accurate model of the video system. In the following section we will introduce the video system model. In sections 3-5, we will discuss parts of the model. Those parts are frame and region selection, displacement estimation, and blur modelling. Section 6 briefly discusses the POCS approach used to project errors back to the original image. An example is provided in section 7, and finally conclusions are offered.

2. VIDEO MODEL

In Figure 1 we have a generic video model. It consists of *K* channels, where each channel *k* is a different image. The original images are represented by $f_k(x, y)$. The original image is convolved with blur $h_k(x, y)$, noise $w_k(x, y)$ is added, and then the image is sampled on a 2-dimensional grid to obtain the observed image frame $g_k(m, n)$. This model ignores correlation between successive images. To achieve a benefit from multiframe processing, it is necessary to relate the images. This is generally done with a motion transform. In Figure 2, we represent the sequence with one image and *K* motion transforms. These motion transforms must be very precise in order to make this type of processing possible. We have experimented with a dense optical flow method[2] as well as a parametric affine-displacement matching algorithm[3]. The optical flow method has the advantage of allowing spatially-variant motion data; however, the affine-displacement matching algorithm has better performance when the moving object fits the affine motion model.

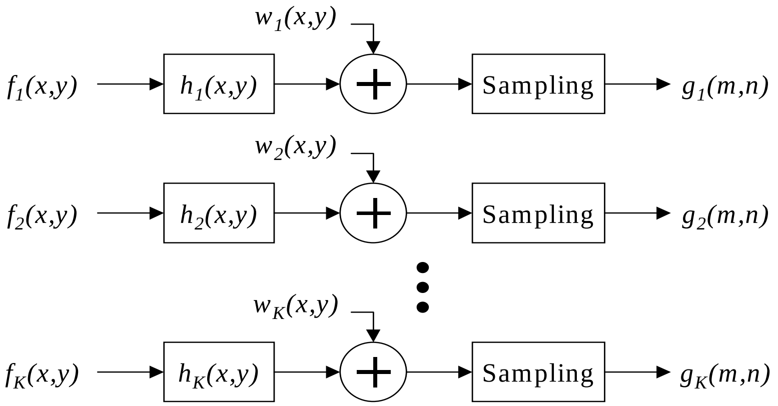

Figure 1: Generic video model

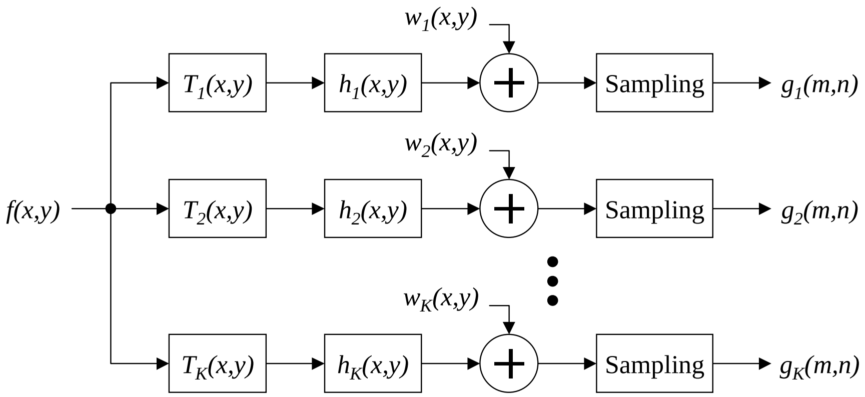

Figure 2: Video model using motion compensation

In equation form, the model of Figure 1 becomes

$$g_k(m, n) = [f_k(x, y) \otimes h_k(x, y) + w_k(x, y)]\Big|_{x = mT_x, y = nT_y} \tag{1}$$

where $\otimes$ is a two-dimensional convolution operator, m and n are the indices for the discrete two-dimensional sequence $g_k(m, n)$, and T_x and T_y are the x and y sensor spacings. The model of Figure 2 is represented in the following equation

$$g_k(m, n) = [f(x - u_k(x, y), y - v_k(x, y)) \otimes h_k(x, y)]\Big|_{x = mT_x,\, y = nT_y} \tag{2}$$

where $u_k(x, y)$ and $v_k(x, y)$ are the displacement vectors for each pixel. Note that the displacements are subtracted. This is a backward-mapping approach, which is a convenient approach for getting values at integer pixel locations.[2]

Although not represented in the above video models, the input video is often obtained from an interlaced source. Since the odd and even fields are captured at different time instances, it is necessary to separate the fields before additional processing. Some de-interlacing technique might be applied to fill the missing lines in each field and to restore the original aspect ratio. This also aids the motion estimation algorithms in properly aligning the frames. Once de-interlacing has been performed, the fields are used in the model as separate frames.[1] If motion blur is to be modeled, care should be taken to make sure that the time-ordering of the fields is correct.

3. FRAME AND REGION SELECTION

A forensic analyst must choose a video sequence for multiframe processing. Within that sequence, the analyst selects the reference frame that will be used for processing. In that reference frame, the analyst selects a rectangular region-of-interest (ROI) that contains some features that might be helpful to an investigation. Such areas might be a face or a clothing insignia (e.g. cap or T-shirt text). It is important to reduce the problem to the ROI because the algorithms require significant processing time. Including image content surrounding the ROI would only be beneficial if it contains edges or other features that might aid displacement estimation.

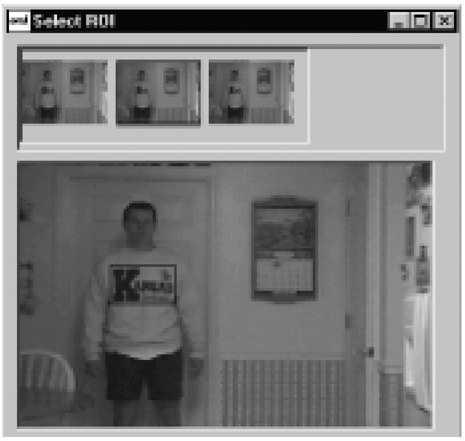

Figure 3: Defining of ROI

An automated region matching can be used to choose frames that will be useful for constructing a high-quality still image. This region matching involves performing an autocorrelation in the neighborhood of the ROI on the preceding and following frames. The neighborhood used for the search is a rectangular region slightly larger than the ROI. The ROI is scanned along the search neighborhood, performing an autocorrelation at each position. When the best match for the ROI is found in the adjacent frame, the procedure is continued to the next adjacent frame. As the search continues, the search rectangle is adjusted to correspond to the neighborhood of the most recently found ROI. The process continues, working outward from the reference frame, one search going backward in time, and the other search going forward in time. The result of this processing is a ROI for each frame in the sequence. Next, the autocorrelation values can be used to judge which frames are most suitable for multi-frame processing. Those with values lower than a predefined threshold are rejected on the assumption that when the autocorrelation is low, an occlusion occurred or there was a significant change in the object's orientation relative to the camera. This region selection can be extended to more elaborate matching techniques such as image warping using affine motion parameters or per-pixel motion data. However, as the frame selection method is made more complicated by motion estimation, one starts to defeat part of the purpose of frame selection, which is to reduce unnecessary processing. Figure 3 shows the interface used to select the frame and rectangular ROI.

4. DISPLACEMENT ESTIMATION

The performance of multiframe processing is very dependent on the accuracy of the displacement estimation. Ideally we would like to obtain an independent, or nearly independent, motion vector for each pixel. This type of information would account for subtle spatial variations in the movement of faces or clothing. However, allowing a motion vector for each pixel makes it difficult to obtain very accurate motion data.

In optical flow approaches such as Horn and Schunck[4], the motion estimation is dependent on pixels in a small area. Also, the calculation involves gradients which can have problems with noise. To overcome this problem, some of our work uses a type of affine motion block-matching similar to that of Fuh and Maragos.[3] Fuh and Maragos perform block matching to search for the optimum parameters: scale, rotation angle, *x* displacement, and *y* displacement.

We modified the algorithm by using a hierarchical multi-resolution approach. A pyramid of successively lowpass-filtered images are created from the input images. The block-matching to achieve parameter estimation begins at the highest pyramid level (lowest resolution) and successively works down the pyramid. The parameter search space for each lower-level image is limited to the neighborhood of the selected parameters from the next higher level. On the lowest level, two searches are conducted, one with integer displacements, and another with sub-pixel displacements obtained through interpolation. Hierarchical processing is common in block-matching type algorithms, and it is used to reduce the amount of processing required to find a match. However, since every displacement is not tested at the highest resolution, there is the possibility of not finding the best displacement parameters.

It is important to note that there are two types of displacement estimation that must be addressed. Those are the reference frame displacement and the sequential frame displacement. These two are related, and this relationship may be used to reduce processing. The reference frame displacements are the motion transforms shown in Figure 2. These displacement vectors describe how to warp the reference frame into the position of the observed frames. The sequential frame displacements describe the warpings between frames that are adjacent in time. These are needed to describe motion between frames, allowing us to model the motion blur.

5. BLUR MODELING

There are two components to the blur modeling. Those are the motion blur and the optical blur. Each blur is modeled with a point-spread function, and the total blur is the convolution of the two. The motion blur is modeled as a line response by assuming that the motion between frames for each pixel is at a constant velocity and in a straight line. The optical blur is assumed to be a uniform-intensity disk.

As mentioned above, the motion blur relies on the sequential frame displacements for its description. At each pixel the length of the motion blur is obtained by

$$l_k(x, y) = \frac{\alpha \cdot \sqrt{r_k^2(x, y) + s_k^2(x, y)}}{T} \tag{3}$$

where α is the aperture's open time, $r_k(x, y)$ and $s_k(x, y)$ are the sequential displacement vectors for the *x* and *y* dimensions, and *T* is the time between frames or fields. The end result is a blur length that is in units of pixel spacings and is proportional to the displacement vector at the corresponding pixel. The direction of the blur is identical to the displacement vector.

Once the length of the blur is obtained, we use a model introduced by Tull and Katsaggelos[5] and shown in Figure 4. The blur length is rounded to the nearest integer, and the blur is uniformly distributed among points of unit spacing. These are shown as dark dots in Figure 4. Each of those points are distributed among its four surrounding neighboring pixels. The weighting assigned to each of the four pixels is inversely proportional to the distance of its center from the point of the blur response.

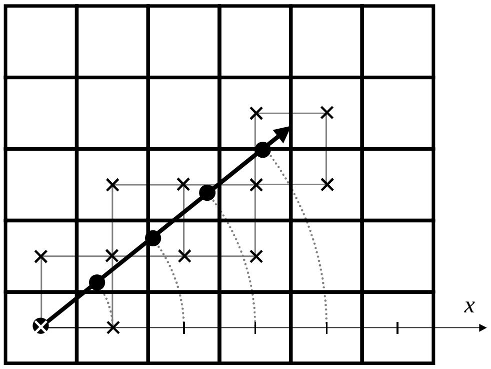

Figure 4: Motion blur model used by Tull and Katsaggelos

Optical blur models are discussed by Lee.[6] He explains how geometric optics models optical blur as a circular disk of uniform intensity. According to geometric optics, light rays travel in straight lines unless refracted by a change in medium. In a camera, a lens or combination of lenses is used to converge the light from an observed object to a point inside the camera. If the object is properly focused, the convergent point is on the camera's sensor plane. Otherwise, a spreading occurs on the sensors because the light rays have not yet converged, or have already converged. The shape of this spreading on the sensor array is the point-spread function (PSF) for the observed object. Generally a camera has an aperture that is formed by several metal blades arranged in a circle, so the converging rays of light are shaped into a circle of uniform intensity. This is shown in Figure 5. In this model, we are assuming that the light source is on the optical axis of the lens, and we are neglecting diffraction and spherical aberration.[7] The uniform-intensity disk is a crude approximation, and Lee shows how real optics can deviate from that model. However, the disk is a commonly used approximation because it is easy to compute.

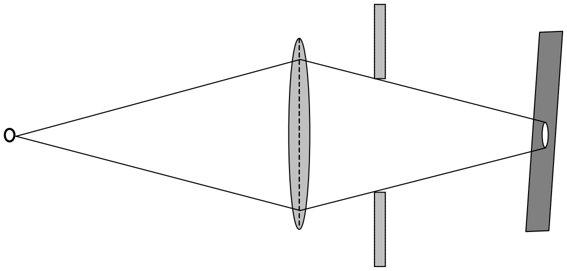

Figure 5: Converging light rays as predicted by geometric optics and several approximations

The only difficulty in using the disk model is in determining how to approximate it using rectangular pixels. The disk covers a grid of rectangular sensor cells as shown in Figure 6. The edge pixels are partially covered by the response, so they will receive a coefficient proportional to the area of the circle occurring in that cell. Here we are assuming that each sensor cell is a perfect rectangle, and there is no space between adjacent sensor cells. Also, we assume that the value of a pixel is the result of a uniformly weighted integration of the light impinging on that particular sensor cell. To avoid the complication of integrating the area bounded by the curved line in the border pixels, those pixels are divided by a sub-grid of rectangles. The resolution depends on the desired accuracy of the circular shape. The rectangles are turned "on" or "off" depending on whether their center falls within the boundary of the circle. Those pixels that are fully covered by the blur circle have all of their sub-rectangles turned "on". The response for each pixel in the PSF is defined as

$$\text{pixel response} = \frac{\text{number of sub-rectangles "on" in pixel}}{\text{number of sub-rectangles "on" in total blur response}} \tag{4}$$

so that the total blur response is equal to one.

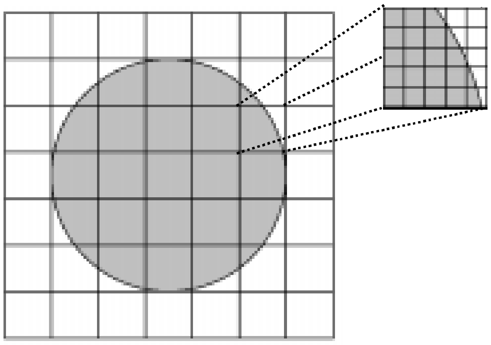

Figure 6: Circular blur approximation

6. POCS ITERATIONS

The method of frame combination is the POCS method used by Patti, Sezan, and Tekalp.[1] Their main group of convex sets is shown here using slightly different notation for a brief review. The convex sets $C_k(m, n)$ are defined as

$$C_k(m, n) = \{y(m_r, n_r):(\left|r_k^{(y)}(m, n)\right| \le \delta_k(m, n))\} \tag{5}$$

where m_r and n_r are the indices of the estimate for $f(x, y)$ which is at the reference time, $r_k^{(y)}(m, n)$ is the residual for any image $y(m_r, n_r)$ being used as the estimate, and $\delta_k(m, n)$ is the allowed error, which is due to modeling errors and noise. The residual is defined as

$$r_k^{(y)}(m, n) \equiv g_k(m, n) - \sum_{(m_r, n_r)} \{y(m_r, n_r) \otimes h_k(m_r, n_r; m, n)\} \tag{6}$$

where $h_k(m_r, n_r; m, n)$ is the response observed at (m, n) in frame *k* for the source pixel at (m_r, n_r) in the estimate image. A projection is defined which is applied to the image estimate once for each output pixel. Each projection causes the residual to decrease for the corresponding output pixel, although it may cause an increase of the residual at another pixel. The projections are applied iteratively until the average error decreases or a maximum number of iterations have been performed.

7. EXAMPLES

The algorithms were applied to three images chosen from a video sequence. The images were digitized at quarter-frame size, and then down-sampled by 2 in each dimension to reduce the resolution. Part of the reference frame is shown in Figure 7. Here it is very difficult or impossible to read the word “Kansas” on the sweatshirt.

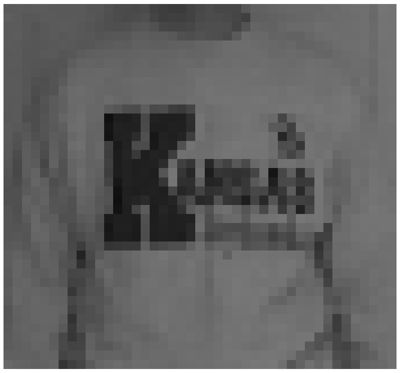

Figure 7: Part of reference frame from image sequence

Figure 8 contains the three extracted ROIs. The middle frame was chosen to be the reference frame, and its ROI was user-defined. The ROIs in the other frames were software-selected. Per-pixel optical flow information was calculated for the low-resolution images,[2] and the motion images were up-sampled and used to describe the warping from the reference frame to the observation frames. Multiple iterations of the POCS algorithm were applied to produce an image up-sampled by a factor of two in the *x* and *y* dimensions. The reconstructed image is the right image in Figure 9, while the left image has the same upsampling factor achieved by bilinear interpolating the reference image. A dramatic increase in quality is obtained, and the word “Kansas” is much more readable. It should be noted that this increase of resolution requires that aliasing is present in the image sequence. The digital down-sampling increased the amount of aliasing present in the original images. For the resolution increase to be beneficial with frames extracted directly from a camera, there must be aliasing caused by the spacing of the sensor cells. This can occur and is evidenced by the aliasing effect that may be observed when small patterns on textiles are imaged.

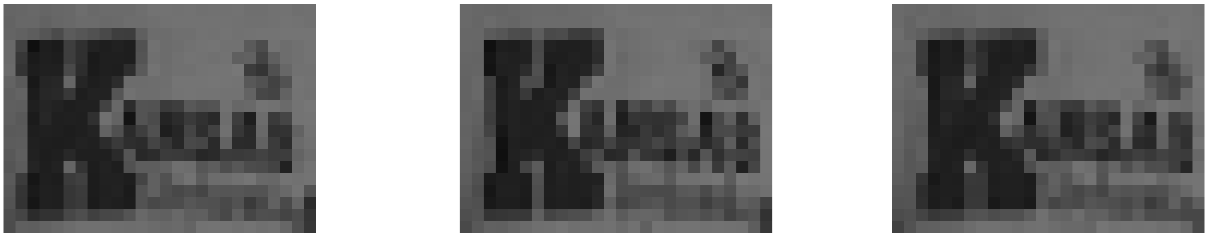

Figure 8: Extracted regions before processing

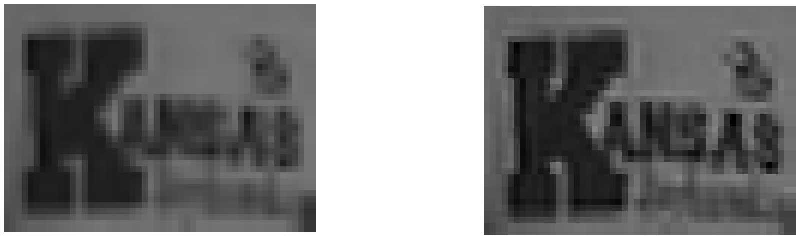

Figure 9: Bilinear interpolation of reference frame (left) and multiframe reconstructed image (right)

8. CONCLUSIONS

We discussed how POCS could be used with modeling and automation software to aid surveillance video restoration. We introduced a method for automated region and frame selection, a hierarchical region matching approach, and a method for approximating the circular response of optical blur. An example was given to demonstrate the effectiveness of using dense optical flow information with POCS reconstruction to increase resolution when aliasing is present. We are currently testing our software for the use of video degraded by motion and optical blur. The inclusion of blur models should prove helpful in these cases and increase the variety of image sequences that can be restored by this software.

REFERENCES

1. A. J. Patti, M. I. Sezan, and A. M. Tekalp, “Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time,” *IEEE Trans. Image Processing*, **6**, pp. 1064-1076, 1997.
2. M.J. Black, *Robust Incremental Optical Flow*, Ph.D. thesis, Yale Univ., New Haven, Connecticut, 1992.
3. C. Fuh and P. Maragos, “Motion displacement estimation using an affine model for image matching,” *Optical Engineering*,” **30**, pp. 881-887, 1991.
4. B.K.P. Horn and B.G. Schunck, “Determining optical flow,” *Artificial Intelligence*, **17**, pp. 185-203, 1981.
5. D.L. Tull and A.K. Katsaggelos, “Iterative restoration of fast-moving objects in dynamic image sequences,” *Optical Engineering*, **35(12)**, pp. 3460-3469, 1996.
6. H. Lee, “Review of image-blur models in a photographic system using the principles of optics,” *Optical Engineering*, **29(5)**, pp. 405-421, 1990.
7. W.J. Smith, *Modern Optical Engineering*, McGraw-Hill, New York, 1990.