Moving Vehicle Registration and Super-Resolution[†]

Frederick W. Wheeler
Visualization and Computer Vision Laboratory
GE Global Research
Niskayuna, NY, USA
wheeler@research.ge.com

Anthony J. Hoogs
Kitware, Inc.
Clifton Park, NY, USA
anthony.hoogs@kitware.com

Abstract

We describe a method for registering and super-resolving moving vehicles from aerial surveillance video. The challenge of vehicle super-resolution lies in the fact that vehicles may be very small and thus frame-to-frame registration does not offer enough constraints to yield registration with sub-pixel accuracy. To overcome this, we first register the large-scale image backgrounds and then, relative to the background registration, register the small-scale moving vehicle over all frames simultaneously using a vehicle motion model. To solve for the vehicle motion parameters we optimize a cost function that incorporates both vehicle appearance and background appearance consistency. Once this process accurately registers a moving vehicle, it is super-resolved. We apply both a frequency domain and a spatial domain approach. The frequency domain approach can be used when the final registered vehicle motion is well approximated by shifts in the image plane. The robust regularized spatial domain approach handles all cases of vehicle motion.

1. Introduction

The basic goal of image Super-Resolution (SR) is to estimate a high-resolution image from several low-resolution images of the same scene. Super-resolution gains come from a combination of noise reduction, de-aliasing and deblurring, or high-spatial frequency restoration. Super-resolution has a long history, primarily of applications to whole images of static scenes [3, 8, 9, 2].

In typical super-resolution applications, images are registered with a whole-image transform such as a homography, affine transform, or shifts. Such whole-image transforms can accurately register images of static scenes. However, with simple whole-image transforms, moving vehicles are not registered, and will in fact be distorted by super-resolution processing.

In this paper we describe a method specifically for registering and super-resolving moving vehicles from aerial surveillance video and show sample results. In many image exploitation applications, moving vehicles are of specific interest, either for manual viewing or for automatic classification or tracking. Restoring more detailed vehicle images can help in these and other tasks.

Image super-resolution in almost all cases produces restored images that appear more crisp and that are more pleasing to view. However, the real value of super-resolution in operation is when some new information about an imaged object can be determined. In the results section we are able to demonstrate this, especially in Figure 5 where a super-resolved image of a van makes it clear that the van has two rear door windows, not just one—a fact that is not discernible in the original video sequence.

2. Registration

A prerequisite to image super-resolution is image registration with sub-pixel accuracy. Accurate registration of whole images is generally achievable using the images themselves, in part because a very large number of pixels are used to solve for a small number of registration function parameters. For example, the 8 parameters of a homography, 6 affine parameters or 2 shift parameters, as appropriate for the scene and imaging conditions.

[†] The authors gratefully acknowledge the advice and support of Eamon Barrett of the Lockheed Martin Space Systems Corp.

For notational convenience we will represent registration functions as homography matrices, though the actual registration function may have fewer parameters, or be more general. Image points are thus represented by homogeneous coordinates [6].

The moving vehicles we will super-resolve are small, on the order of 10 by 20 pixels. Restricting ourselves to small image regions where the vehicle is the dominant object and registering the vehicle the way whole images are registered would result in poor registration. There is not enough information in such small image regions to accurately register a pair of them.

To accurately register vehicles we use a two-stage approach. First, standard whole image approaches are used to register the static dominant background portions of the images, working with the images in pairs. Second, vehicle motion is estimated over all N frames jointly using a constrained motion model and a frame-to-frame consistency criterion. Thus, much more image data is applied toward the estimation of a few registration parameters for the vehicles. A constant velocity motion model is appropriate for our target data.

Estimating the vehicle motion parameters, and thus the vehicle registration is performed with unconstrained nonlinear optimization. A cost function is defined with a foreground (vehicle) consistency component and a background consistency component.

2.1. Whole Image Initial Registration

The sequence of N video frames is initially registered with a homography for each consecutive pair of frames. These whole-image homographies are estimated using the Kanade-Lucas-Tomasi feature tracker [10]. Homography $H^B_{i,j}$ maps coordinates of a background feature point in frame i to the same feature point in frame j.

An arbitrary, but fixed, world frame of reference is selected (typically, but not necessarily, equivalent to on of the video frames) and we convert these homographies to $H^B_{w,i}$, which maps coordinates from the world frame ("w") to coordinates in frame i. Then, for example, mapping coordinates from frame i to frame j is done using the product $H^B_{w,j} {H^B_{w,i}}^{-1}$. Figure 1 shows the world frame of reference, the videos frames, and other frames of reference introduced below, with the homographies we will define and use.

2.2. Vehicle Selection

Initial vehicle waypoint locations, x_A and x_B, are coarsely found by hand in two different user-selected frames, n_A and n_B, typically the 1st and 4th in our experiments. These points are approximately on the center of the vehicle as it appears in the image. The background registration homographies for the frames in which these points are specified are set to as $H^B_{w,A} = H^B_{w,n_A}$ and $H^B_{w,B} = H^B_{w,n_B}$ for convenience.

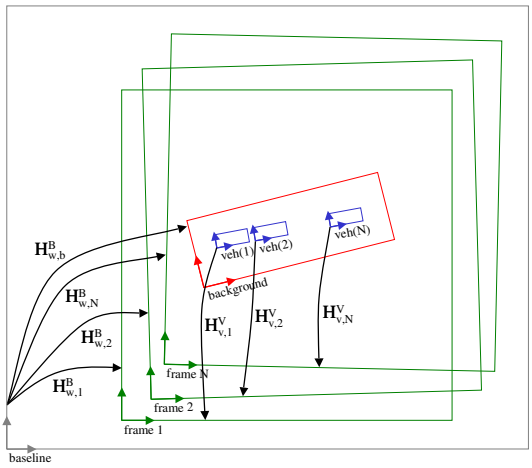

Figure 1. The frames of reference used are shown here, along with the homographies that map points from one frame of reference to another.

2.3. Whole Image Registration Refinement

From the initial vehicle waypoints we know the approximate location of the vehicle in each frame. For each video frame I_i we crop a 512 by 512 region around this location, and adjust the homographies $H^B_{w,i}$ appropriately. This region size is fairly arbitrary–other sizes may work just as well.

The background registrations $H^B_{w,i}$ for these regions are further refined by solving for additional shift-only registration components for frames 2 to N that minimize normalized correlation with frame 1. For our test data set we have found this to be sufficient to align the backgrounds. This process eliminates the error accumulation (dead reckoning) problem that occurs when a whole video sequence is registered by registering consecutive pairs of frames.

2.4. Parameterization

Our model for the vehicle is that it travels with a constant ground velocity and approximately occupies a fixed size rectangle. The path of the vehicle will be parameterized by the two waypoints x_A and x_B. We have initial values for these parameters and we will optimize them. The fixed size vehicle rectangle defines a new vehicle frame of reference that follows the vehicle as it travels. We determine

homographies that map points from this frame of reference to each video frame.

By the constant velocity assumption, in the coordinates of frame i, the vehicle will be at location

$$x_i = H^B_{w,i}(\frac{n_B - i}{n_B - n_A} * {H^B_{w,A}}^{-1} x_A + \frac{i - n_A}{n_B - n_A} * {H^B_{w,B}}^{-1} x_B) \quad (1)$$

This essentially converts the two vehicle waypoints to the world frame of reference, interpolates or extrapolates the points to the time of frame i, and converts the result to frame of reference i. So, the vehicle is moving at a constant velocity in the world frame of reference and is properly tracked even though there is additional frame-to-frame camera motion. Evaluating this equation at $t + \Delta$ also tells us the direction of travel.

We define a vehicle image I_v and vehicle frame of reference that is just large enough to hold the vehicle rectangle. For each frame, the vehicle rectangle is centered at x_i, and oriented according to the direction of travel. The effectively defines vehicle registration homographies $H^V_{v,i}$ that map points on the vehicle from the vehicle frame of reference to same vehicle point in frame i.

Based on the initial vehicle motion parameters we define a background region that is a fixed size rectangle. The rectangle is aligned with the initial vehicle path and sized so that as the vehicle path is optimized it is expected to stay within the background region. Homography $H^B_{w,b}$ maps points from the world frame of reference to the background frame of reference. As the vehicle motion parameters change, the vehicle frame moves with respect to the world frame, but the background frame does not. The purpose of the background region is primarily to reduce computation by operating on a small portion of the image, in the vicinity of the vehicle.

2.5. Vehicle and Background Masks

To more clearly express the cost function process below, we adopt the notation that HI, where H is a homography and I is an image, represents the image warped with bilinear interpolation according to the homography. Further, image multiplication denoted $I_1 * I_2$ and division I_1/I_2 are done pixel-wise. We omit details of image sizes.

A tapered vehicle mask is used so that the cost function will be smoother with respect to the vehicle motion parameters. In the vehicle frame of reference we define a rectangular mask I_{v-mask} that is 1 inside the rectangle, 0 outside, and tapers linearly over a couple of pixels widths at the border.

In the background frame of reference we also define a mask. The background mask excludes the vehicle and is thus frame/time dependent. The background mask is 1 minus the vehicle mask warped to the background frame of reference, using the vehicle position for each frame,

$$I_{b-mask,i} = 1 - H_{w,b} {H^B_{w,i}}^{-1} H^V_{v,i} I_{v-mask} \quad (2)$$

By this equation, the vehicle mask is warped to frame i, then to the world frame of reference, and then to the background frame of reference. In actual implementation, the three warping functions are combined into one so the mask is warped only once. After this warping operation, the vehicle mask is inverted to form the complementary background mask. The use of these masks will result in a cost function that is smooth at small scales.

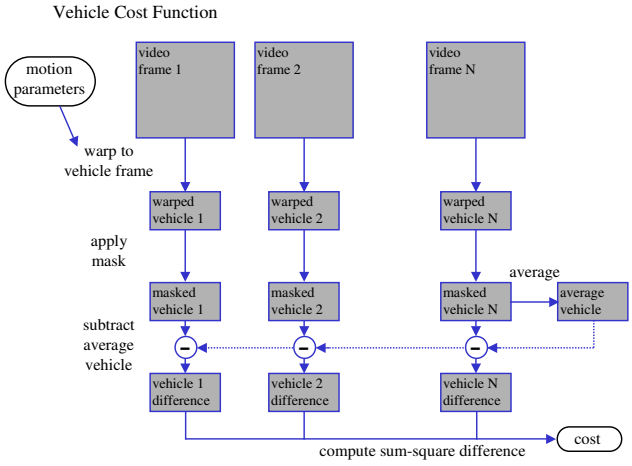

Figure 2. The process for computing the vehicle portion of the cost function, given vehicle motion parameters, is depicted here.

2.6. Vehicle Average

Each video frame is warped, with bilinear interpolation, to the vehicle frame of reference and these warped frames are averaged. The vehicle mask is applied to the result. This is the average vehicle image and it should match the vehicle appearance in each frame if the motion parameters are correct.

$$I_{v-avg} = I_{v-mask} * \frac{1}{N} \sum_i {H^V_{v,i}}^{-1} I_i \quad (3)$$

2.7. Background Average

In a similar manner, each image frame is warped to the background frame of reference. For each image frame we have a different background mask to exclude the vehicle, and the appropriate mask is applied. The masked warped

Figure 3. Example frames 1 and N=8 with the background frame (large rectangle), vehicle frame at this time (small rectangle), and vehicle positions over time (x-marks), using the initial (not optimized) vehicle motion parameters.

Figure 4. Example of Fig. 3 after optimization.

background images are added, and the background masks themselves are also added. These two sum images are divided to get an appropriately weighted average background image.

$$I_{b-sum} = \sum_i I_{b-mask,i} * H^B_{w,b} {H^B_{w,i}}^{-1} I_i \quad (4)$$

$$I_{b-weight} = \sum_i I_{b-mask,i} \quad (5)$$

$$I_{b-avg} = I_{b-sum} / I_{b-weight} \quad (6)$$

2.8. Cost Function

If the vehicle motion parameters are correct we expect the vehicle in each frame to match the average vehicle image and background of each image to match the average background image. We define a two part cost function expressing this. The vehicle part of the cost function (outlined in Figure 2),

$$J_v = \sum_i \| I_{v-mask} * (I_{v-avg} - {H^V_{v,i}}^{-1} I_i) \| \quad (7)$$

sums, over each frame, the image power of the masked difference between the average vehicle image and warped vehicle for this frame. The background part of the cost function,

$$J_b = \sum_i \| I_{b-mask,i} * (I_{b-avg} - {H^B_{w,i}}^{-1} I_i) \| \quad (8)$$

sums, over each frame, the image power of the masked difference between the average background image and warped background for this frame. The total cost is the sum of these parts,

$$J = J_v + J_b \quad (9)$$

Clearly the foreground portion of the cost function is lower when the vehicle motion parameters align the vehicle in each frame. The effect of the background portion is to force the vehicle rectangle on top of the vehicle. If the vehicle rectangle drifts off of the vehicle, portions of the vehicle become part of the average background image and increase the cost function.

This framework is designed to allow the vehicle mask to take on more arbitrary and realistic shapes and to handle an arbitrary background. We have found, however, that for vehicles on reasonably smooth roadways, keeping the vehicle mask as a fixed rectangle works well. The background in the immediate vicinity of the vehicle is constant, so there is no harm in attributing it to the foreground instead of the background.

2.9. Optimization

To solve for the vehicle motion parameters by minimizing the cost function, we use the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure, implemented by the MATLAB `fminunc` function. About 6–8 iterations, with 70–100 function evaluations, are used to register a moving vehicle in this unconstrained nonlinear optimization problem. The gradient is computed with finite differencing. For the optimization, we can set an intuitive and meaningful termination tolerance on the vehicle motion parameters, waypoints x_A and x_B, since they are all in units of pixel width.

In Figures 3 and 4 we see how the vehicle regions move to align with the path of the vehicle, within the fixed background region, through the optimization process. Note how poorly the vehicles from the 8 frames line up before optimization (Figure 3), and how well they line up after (Figure 4).

3. Super-Resolution

3.1. Image Resolution

It is important to differentiate between the pixel resolution and the actual image content resolution of a super-resolved image. One of the primary goals of image super-resolution is to solve for an image that has accurate spatial frequency content above the Nyquist frequency of the observed images. To represent these high frequencies without aliasing, the super-resolved image typically has twice the pixel resolution of the observed images. Doubling the pixel resolution has many algorithmic and practical advantages. Of course, the super-resolved image will not contain accurate spatial frequencies beyond the diffraction limit of the camera. Image noise, and numerical limitations may further limit the spatial frequency extent of the super-resolved image. So, while the super-resolution result is represented on an image with twice the pixel resolution, this does not imply that the actual image content resolution will double.

3.2. Frequency Domain Super-Resolution

Frequency domain super-resolution models the image formation process in the spatial frequency domain. To do this in a straightforward manner, the registration must be shift-only. When the shift-only registration criteria is met, spatial frequency domain super-resolution works quite well and requires little computation.

In the registration procedure described above, homographies are used to represent the background registration of the whole image. For our target data, the frame-to-frame rotation is small, but for the large images it is still significant. However, over a small region such as a vehicle, the rotation is no longer significant in terms of pixel shift, and shift-only registration is an accurate model.

We apply the spatial frequency domain method described by Kim [7], which we refer to as Frequency Domain Direct (FDD). In summary, we wish to solve for a high-resolution image, X, from the observed images Y_i, that are registered by sub-pixel shifts. The image formation process is modeled, including registration, the Point Spread Function (PSF), and sub-sampling. Typically, X has twice the pixel resolution of the observed images in each dimension. Each component of the image formation process is readily and efficiently handled in the Fourier domain. Shift-only registration becomes a product with the complex exponential of a linear phase function. Convolution with the PSF becomes multiplication by the Optical Transfer Function (transform of the PSF). Finally, sub-sampling becomes a simple operation due to the aliasing rule.

The position of the center of the vehicle rectangle on each frame, X_i, is determined from equation (1). A 32 by 32 region about the integer pixel coordinates $X_i^c = \lfloor X_i + 0.5 \rfloor$ is extracted from each frame and denoted $I_{c,i}$. The residual sub-pixel shift is then $X_i^r = X_i - X_i^c$.

The FDD algorithm is applied to these image chips with sub-pixel shifts X_i^r and a Gaussian PSF with a hand selected width σ. The super-resolved image has twice the pixel resolution of the observed image.

3.3. Spatial Domain Super-Resolution

Frequency domain SR techniques, such as the method used above [7] require shift-only registration. These methods are not suitable when the frame-to-frame vehicle motion is more general. Spatial domain SR frameworks are more appropriate in this situation because of their flexibility with respect to registration and their ability to use robust regularization techniques. In this section we outline the methods of [5, 4], which we will refer to as Robust Super-Resolution (RSR).

We will use the notation of linear algebra, as if all of the pixel values of the observed images and the super-resolution image are elements of very long vectors, and as if linear operations on these vectorized images are products with very large matrices. This is a notational convenience only– helpful for describing and understanding the process. In actual implementation of solutions, these large vectors and matrices are not formed. The solution process is carried out using more computationally practical operations on 2D images.

A high-resolution image, X, is defined, and we form the linear relationship between X and the noisy low-resolution images Y_i. The image X generally has twice the pixel resolution of the observed Y_i images so that it can represent high spatial frequencies recovered through de-aliasing.

The image formation process that models the creation of Y_i from X involves several stages, each a linear operation. First is the motion, or registration compensation, represented by matrix F_i. Image resampling using nearest neighbor, bilinear, or bicubic interpolation is a linear operation. Next is blurring, represented by matrix H_i. The application of a Point Spread Function (PSF) is also a linear operation. The result is subsampled to the resolution of the observed images by a simple sparse matrix D_i. With additive noise represented by vector V_i, the image formation process is then simply,

$$Y_i = D_i H_i F_i X + V_i \tag{10}$$

The warping, blurring and subsampling operation can be combined into a single operation, $A_i = D_i H_i F_i$, making the image formation process,

$$Y_i = A_i X + V_i \tag{11}$$

Solutions for the super-resolved image X are found by optimizing the norm of the difference between the modeled observations and the actual observations,

$$\hat{X} = \underset{X}{\operatorname{argmin}} \left[\sum_i \|A_i X - Y_k\|_1 \right] + \Psi(X) \qquad (12)$$

The additional term $\Psi(X)$ is for regularization, and is discussed below.

The L_1 cost is used here for robustness against errors in the observation modeling process, vehicle registration, and the noise model. If we were exactly right about each of these components of the model, and if the observation noise were additive white Gaussian, then the L_2 norm would be appropriate. Since we are unlikely to achieve perfect modeling, a strong case can be made for the L_1 norm.

The registration term $\Psi(X)$ is required because, without it, in an ill-posed problem such as this, there are many undesirable solutions for X that satisfy equation (12) equally well. This happens primarily because very high spatial frequency components of X are nearly completely filtered out by the PSF so they have no effect on the data fidelity cost. Regularization terms are added to counteract this effect and prevent the solution from having unwanted unobservable components.

The well-known Tikhonov regularization function,

$$\Psi(X) = \alpha^2 \|\Gamma X\|_2 \qquad (13)$$

uses the L_2 norm on the gradient or a high-pass filtered version of the image. This has the effect of adding cost for images with sharp edges, which is counterproductive to our super-resolution goals. More desirable for image restoration is total variation regularization,

$$\Psi(X) = \alpha \|\nabla X\|_1 \qquad (14)$$

Here, the L_1 norm is used on the gradient of the image. With this regularization function, isolated edges do not have a higher cost than smoothed versions of the edges.

Bilateral Total Variation (BTV) [5] is a generalization of total variation regularization that extends the neighborhood at which absolute difference constraints are applied

$$\Psi(X) = \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|m|+|l|} \left\| X - S_x^l S_y^m X \right\|_1 \qquad (15)$$

Parameter P controls the size of the neighborhood, and typically $P = 2$. With $P = 1$, BTV is very similar to total variation.

The super-resolved image X is solved for using a steepest descent optimization with an analytic gradient of the cost function [5]. Typically, using about 20–30 fixed-step-size iterations works well.

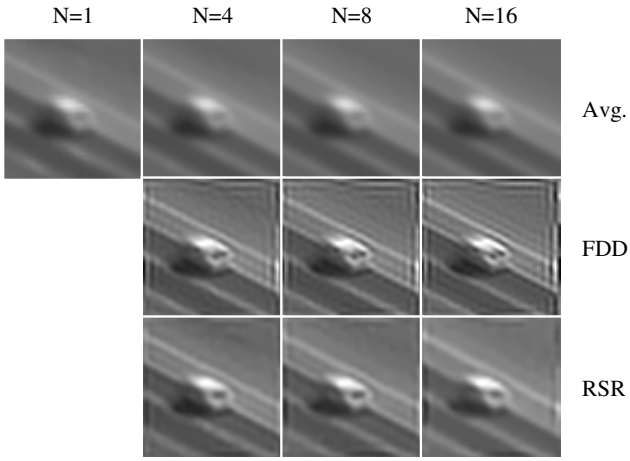

Figure 5. Sample results for each method on the "interchange" sequence.

4. Results

In this section we show results on two different data sets. The first, "interchange", shows a highway interchange with a large number of moving cars. Frames are 4096 by 4096 pixels, and 12-bit grayscale. The second data set, "balloon", is a 24-bit color sequence showing cars on a city road. The "balloon" video was collected with an interlaced camera. Interlacing is not explicitly handled by either of the super-resolution algorithms used. However, as we will see, there is some robustness to the effects of interlacing. For each test sequence, over short durations and in small regions, actual frame-to-frame scene motion is nearly shift-only.

For a number of cars from each data set we have applied the moving vehicle registration process and super-resolved the vehicle using both the Frequency Domain Direct (FDD) and Robust Super-Resolution (RSR) methods. For all test data, and both super-resolution methods, a Gaussian PSF with $\sigma = 0.8$ was found to work well.

Figures 5 and 6 show results for the "interchange" data set and Figure 7 show a result for the "balloon" data set. In each case we see images of the result of warping and averaging the vehicle from $N = 1$, 4, 8 and 16 consecutive frames. The image for $N = 1$ is simply a chip from a single original video frame and is useful as a baseline for comparison. Also shown are the FDD and RSR super-resolution results for $N = 4$, 8 and 16 consecutive frames.

For the "interchange" video the general trend is that the FDD result for $N = 8$ or $N = 16$ frames seems to be the most clear. Sometimes the $N = 8$ FDD result looks better than the $N = 16$ result. This may be due worsening registration, perhaps if the vehicle is changing speed or slightly turning in a way that is not handled by the vehicle motion

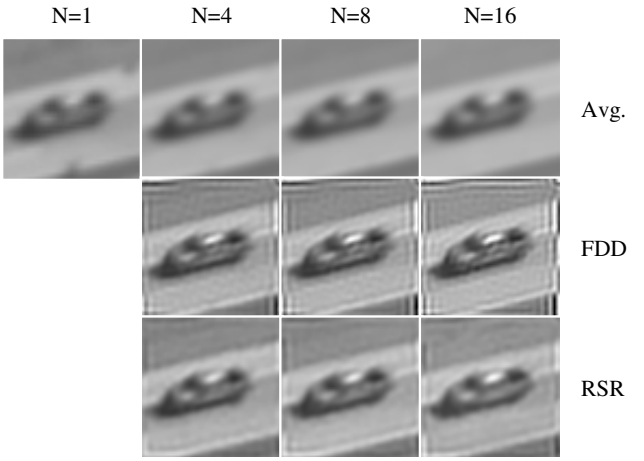

Figure 6. Sample results for each method on the "interchange" sequence.

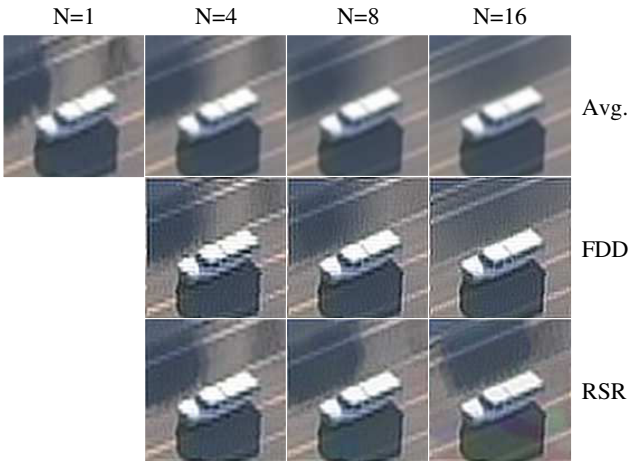

Figure 7. Sample results for each method on the "balloon" color sequence.

model.

The FDD result for the rear end of the van in Figure 5 is especially striking. Here, the super-resolved result clearly shows that the rear of this van has two windows, not one. This cannot be determined from the original video. In this case, super-resolution processing has not just made the image more crisp, but has revealed new and potentially useful information.

For the "balloon" video results we often see harsh interlacing artifacts. However, the FDD result for $N = 16$ frames seems to be free of interlacing artifacts.

5. Conclusions

We have presented a robust vehicle registration and super-resolution technique.

Earlier implementations of this system used a vehicle motion model that was parameterized by a starting location (x, y), a scalar velocity (v), and an angular direction (a) of travel. The implementation described here uses a different parameterization, the two waypoints. While functionally the same, using the two waypoints has some advantages. In this case, each parameter has a similar effect on the system, similar to the homography estimation approach taken in [1].

In a larger system, it would be straightforward to automatically select vehicles based on adherence to the constant velocity assumption, or to extend the vehicle motion model to handle turning vehicles. Vehicles are selected that are on a straight stretch of roadway and we would expect that in any application, there are frequent intervals where any particular vehicle is going in a straight direction.

References

[1] S. Baker, A. Datta, and T. Kanade. Parameterizing homographies. Technical Report CMU-RI-TR-06-11, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2006.

[2] S. Borman. *Topics in Multiframe Superresolution Restoration*. PhD thesis, University of Notre Dame, Notre Dame, IN, May 2004.

[3] S. Chaudhuri, editor. *Super-Resolution Imaging*. Kluwer Academic Publishers, 3rd edition, 2001.

[4] S. Farsiu, M. Elad, and P. Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing*, 15(1):141–159, January 2006.

[5] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super-resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, October 2004.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] S. P. Kim, N. K. Bose, and H. M. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Transactions Acoustics, Speech, and Signal Processing*, 38(6):1013–1027, June 1990.

[8] K. R. Liu, M. G. Kang, and S. Chaudhuri, editors. *IEEE Signal Processing Magazine, Special edition: Super-Resolution Image Reconstruction*, volume 20, no. 3. IEEE, May 2003.

[9] M. Ng, T. Chan, M. G. Kang, and P. Milanfar, editors. *EURASIP Journal on Applied Signal Processing (JASP) Special Issue on Super-Resolution Enhancement of Digital Video*. Hindawi Publishing Corporation, 2006.

[10] J. Shi and C. Tomasi. Good features to track. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, Washington, USA, June 1994.