Super-Resolution Under Complex Image Motion

Eran Treister
Javier Turek
Supervisor: Ron Rubinstein

Geometric Image Processing Laboratory
Department of Computer Science
Technion, Israel

April 2007

Introduction

Image resolution depends on the physical characteristics of the digital camera. Some of these physical characteristics may cause distorting effects such as blurring and noise, and therefore, many algorithms for up-sampling images do not achieve satisfactory results. In figure 1, we see an example of the Lanczos algorithm and some of its problems.


Figure 1: Low resolution image (left) and up-sampled image using Lanczos algorithm (right)

However, increasing the image resolution could be done by obtaining more samples of the scene from a *sequence* of displaced pictures. By performing a combination of image registration and image restoration on the set of input images, a higher resolution image can be achieved. Image registration with sub-pixel accuracy is used to place each image in its correct position on the high resolution grid, while image restoration is used to fill-in the holes and correct the errors arising by the previous process. Hence, the entire process is formed by a registration step followed by a restoration step.

In this project, we have implemented a process which follows the above scheme based on the super-resolution algorithm suggested in [1]. In order to calculate the motion between the images, we step beyond [1] by incorporating the optical flow algorithm proposed in [2].

Super-resolution model

The general model which we use in this project assumes the existence of some true high-resolution image *X* which we wish to recover, and a set of known low-resolution inputs which can be represented as the result of a series of transformations applied to *X*, which include

1. Motion
2. Blur
3. Down-sampling
4. Additive noise

We represent this approximate forward model by the following equation:

$$(1) \qquad \underline{Y}(k) = \mathbf{D}(k)\mathbf{H}(k)\mathbf{F}(k)\underline{X} + \underline{V}(k) \qquad k = 1 \ldots N .$$

The vectors *X* and *Y(k)* represent the high resolution frame and the k^{th} low resolution frame, respectively. The matrix **F**(k) is the geometric motion operator between the high and low resolution frames. The blurring effect of the camera is modeled by the blur matrix **H**(k). The matrix **D**(k) represents the down-sampling operator, and the vector *V(k)* is the system noise. Finally, N is the number of available low-resolution frames. The full mathematical model is extensively reviewed in [1], where a more general color model is also considered.

Algorithm outline

The complete algorithm is divided into three main parts (see figure 2):

1. Motion estimation: we compute the motion between a chosen reference image and all other images, using either an optical flow algorithm or a global motion algorithm.

2. Shift & add image: using the motion estimates obtained in the previous part, we construct a new high-resolution image by overlapping all input images. The size of the super-resolved image is determined by a pre-specified magnification factor.

3. Iterative restoration: using the B-TV filter, we apply a restoration process to the resulting S&A image which fills-in remaining holes and reduces noise and blur.

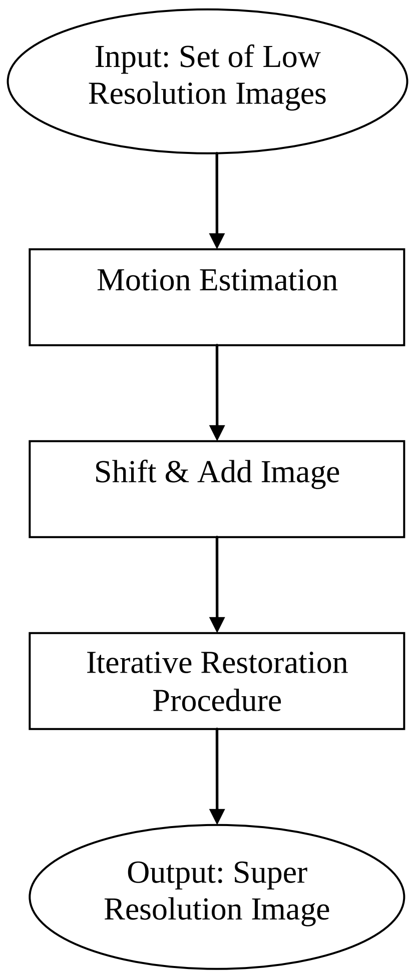


Figure 2: Algorithm flow

1. Motion estimation

A. Global motion estimation

When only camera motion is present, the motion of the scene can be modeled as a linear transformation applied to the scene coordinates. This motion model is also called "global motion". A global motion estimator aims to compute the parameters of this linear transformation.

In order to estimate the motion parameters, we minimize the squared error defined by $\left\| I_{new} - I \right\|_2^2 \equiv \left\| I_t \right\|_2^2$, as follows:

$$(2) \qquad \min_{a,b,\theta} \sum_{x,y} \left(I_t - aI_x - bI_y - \theta y I_x + \theta x I_y \right)^2$$

This problem has a closed-form solution, and can be solved analytically. The resulting algorithm, known as the *Lucas-Kanade* motion estimation algorithm, is extensively discussed in [5]. In this project we implemented a pyramidal (multiscale) variant of this algorithm, which begins by approximating the motion parameters at the lowest pyramid level, and repeatedly refines them as it advances to the higher levels.

B. Optical flow

The *optical flow* between two images refers to a 2D vector field describing the approximate motion of each pixel from one image to the next. This motion field is a 2D projection of the actual 3D velocities of the image pixels, induced by the combined motion of the camera and the scene. The optical flow model is described by the equation

$$(3) \qquad I_{new}\left(x, y\right) \approx I\left(x + u(x,y),\ y + v(x,y)\right)$$

where $I(x,y)$ and I_{new}(x,y) are the two images, and $u(x,y)$ and $v(x,y)$ are the pixel displacements. A classical algorithm for determining the optical flow between two images is described in [4].
In this project we employ the optical flow algorithm described in [2]. This algorithm suggests representing the flow between a pair of images as a sparse combination of atomic flow models, with a smoothness penalty on the variation of these sparse coefficients.

2. Shift & add image

Assuming all images are acquired using the same camera, it follows that $\forall k: \mathbf{H}(k) \equiv \mathbf{H}$. Moreover, assuming that the operators $\mathbf{H}$ and $\mathbf{F}(k)$ are commutative, we can rewrite (1) as

$$(4) \qquad \underline{Y}(k) = \mathbf{D}(k)\mathbf{F}(k)\mathbf{H}\,\underline{X} + \underline{V}(k) \qquad k = 1 \ldots N.$$

It is shown in [3] that the solution for $\underline{Z}$ is obtained as the weighted mean (or median, depending on the choice of L_2 or L_1 norm, respectively) of all known measurements at a given high-resolution pixel, after proper zero filling the low resolution measurements. This operation is known as *shift-and-add* (S&A) reconstruction.

Using the motion estimations from the first stage, we compute for each input image the locations of its pixels in the final shift & add image. The up-scaling is applied by multiplying these locations by the appropriate factor:

$$(5) \qquad \underline{p}_{SA} = factor \cdot \left(\underline{p}_{orig} + \underline{v}_{motion} \right).$$

The results are rounded to produce the final integer coordinates in the output image.

The shift & add image is computed by applying a median or mean to all values at each specific location. The choice of a median or mean is determined by the error metric used (L_1 or L_2) in the error function $J_0\left(\underline{X}\right) = \left\| \Phi\left(H\underline{X} - Z\right) \right\|_{1\, or\, 2}$. Our experiments have shown the median function to produce better results.

3. Iterative restoration procedure

The error function we focus on is

$$(6) \qquad J_0\left(\underline{X}\right) + \lambda J_1\left(\underline{X}\right) = \left\| \Phi\left(H\underline{X} - Z\right) \right\|_1 + \lambda \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|m|+|l|} \left\| \underline{X} - S_x^l S_y^m \underline{X} \right\|_1 .$$

The data fidelity term measures the similarity between the resulting high-resolution image and the original low resolution ones. Considering the model (1), we define the data fidelity term as

$$(7) \qquad J_0\left(\underline{X}\right) = \rho\left(\underline{Y}, T\underline{X}\right) = \sum_{k=1}^{N} \left\| \mathbf{D}(k)\mathbf{H}(k)\mathbf{F}(k)\underline{X} - \underline{Y}(k) \right\|_{1 or 2} .$$

The regularization cost function we use in the project is given by

$$(8) \qquad J_1\left(\underline{X}\right) = \Gamma\left(\underline{X}\right) = \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|m|+|l|} \left\| \underline{X} - \mathbf{S}_x^l \mathbf{S}_y^m \underline{X} \right\|_1$$

This is a robust regularizer known as the *bilateral-TV* (B-TV) operator, which achieves reconstructed images with sharp edges.

To minimize the above error function, we use steepest descent optimization with Armijo linear search. The steepest descent iteration is defined by

$$(9) \qquad x_{k+1} = x_k - \beta_k \nabla\left(J_0\left(\underline{X}\right) + \lambda J_1\left(\underline{X}\right) \right) ,$$

where β_k is the step size in the k^{th} iteration. The Armijo linear search method is employed to improve convergence compared to a fixed step size in the steepest descent iterations, while much cheaper than using a full line search.

The entire process is shown in figure 3, where $f(x) = J_0(x) + \lambda J_1(x)$ denotes the target function and d_k denotes the descent direction (the opposite of the gradient). The other parameters of the optimization process are $\underline{X}_0$, α, β_0 and Δ, where

- $\underline{X}_0$ is the initial image fed to the method (the S&A image),
- α is the parameter of the Armijo method for decreasing the step size each line search step (we use 0.25),
- β_0 is the initial step size for the Armijo method (we use 5000), and
- Δ is the gradient norm target (we use 0.15).

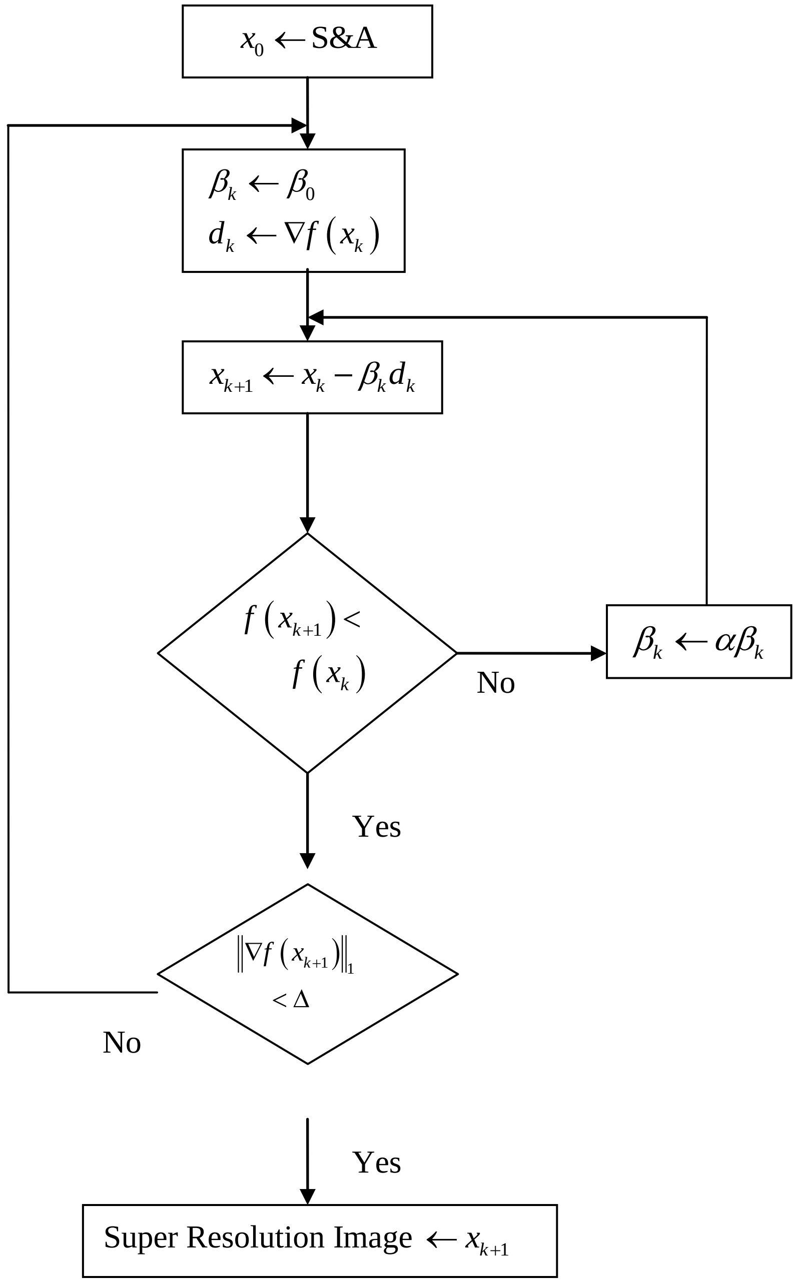


Figure 3: The Optimization Process

Experimental results

We created seven sets of low-resolution images for testing the algorithm. Two sets were generated synthetically from a high-resolution image found on the web, on which different affine transforms and down-scalings were applied. The other five sets are real-world sequences captured with a Sony camcorder: a global motion sequence taken from a tall building; a moving object in front of a still background; a sequence combining simultaneous camera and object movement; a global motion sequence featuring various types of movement; and finally, a global motion sequence with translation only. Below, we show some of these results.

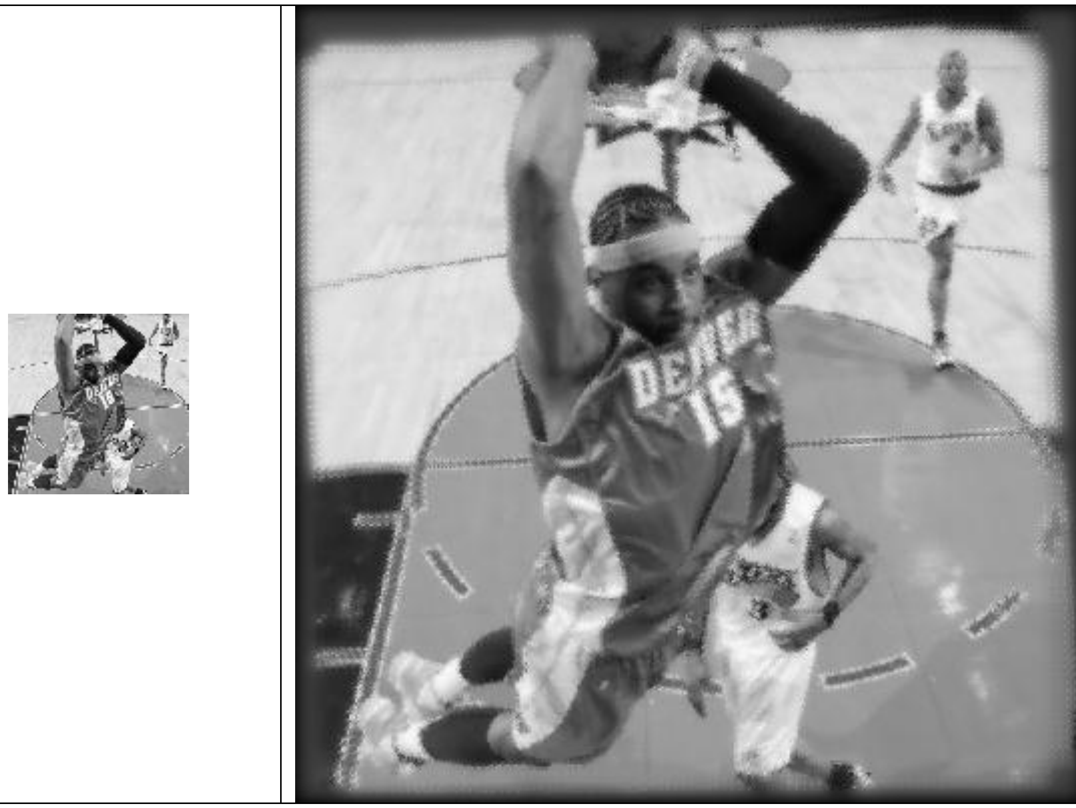


Figure 4: The synthetic *Anthony* sequence results. On the left is a sample input image. On the right is the up-scaled super-resolution result, with a scaling factor of 4.

Figure 5: the upper image is the S&A output using the global motion estimator; the limits

of the motion model are visible. The lower image is the super-resolution result using the optical flow algorithm.

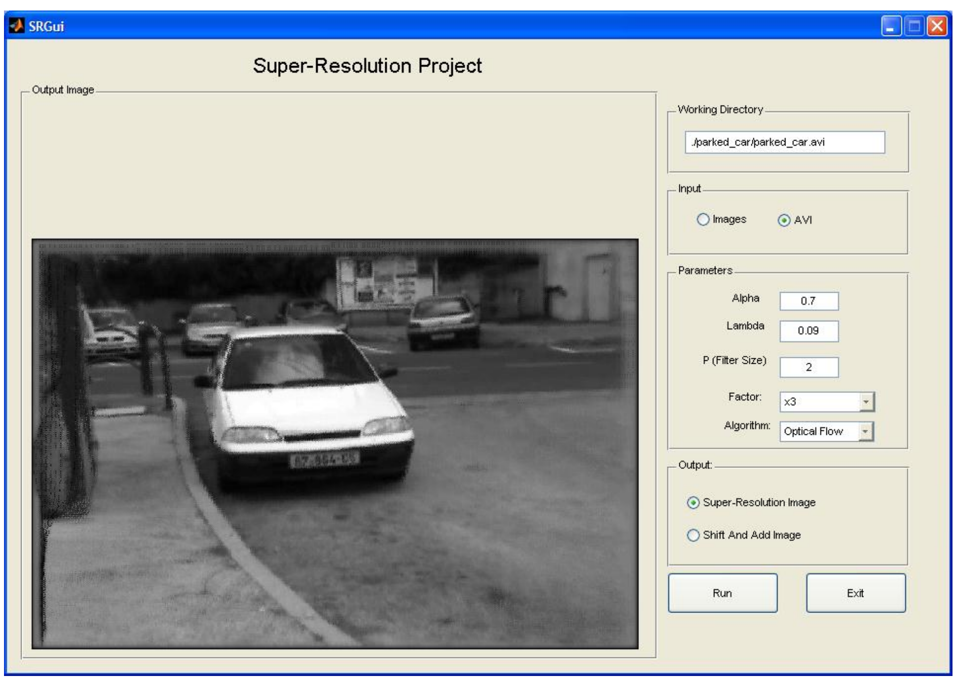
Summary and conclusions

In the results we have seen, we recognize problems mainly along the edges of the objects. This is caused by errors in the motion estimation, which must provide high-level sub-pixel accuracy, especially since every small error is multiplied and becomes more significant in the super-resolution result. Camera movement produces better results than object movement. However, when a small detail is completely lost in the low resolution images, a super-resolution process will never be able to reveal it.

The above algorithm is an extension to the algorithm in [1], capable of constructing super-resolution images from a sequence of low resolution frames. It provides high quality result for global motion sequences and reasonable results for object motion sequences. The algorithm combines a motion estimation algorithm and an iterative reconstruction method, which together produce a robust process with a visible gain in detail.

Software package

The algorithm was developed and implemented in Matlab (the optical flow code was kindly provided by Dr. Tal Nir). The project includes a Matlab command line procedure and a graphical user interface. The low resolution images for the command line procedure are specified as a set of gray-level frames in a three dimensional matrix.



Acknowledgments

We would like to thank Dr. Tal Nir for sharing his optical flow software with us, as well as Prof. Michael Elad and Mr. Ron Rubinstein for their guidance and assistance throughout the project.

References

[1] M. Elad, S.Farsiu and P. Milanfar, "Multiframe demosaicing and super-resolution of color images", IEEE transactions on image processing, vol. 15, No. 1, January 2006.
[2] T. Nir, R. Kimmel and A.M. Bruckstein, "Over-Parameterized Variational Optical Flow", Int. J. of Computer Vision, Vol. 76, No. 2, February 2008.
[3] S. Farsiu, D.Robinson, M.Elad and P. Milanfar, "Fast and robust multiframe super-resolution", IEEE Trans. on Image Processing, Vol. 13, No. 10, pp. 1327-1344, October 2004.
[4] B.K.P. Horn and B.G. Schunck, "Determining Optical Flow", MIT, April 1980.
[5] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision", Proc. of Imaging Understanding Workshop, pp. 121-130, 1981.