Fast Generalized Motion Estimation and Superresolution

Abhijit Sinha and Xiaolin Wu

Electrical and Computer Engineering Dept.
McMaster University, Hamilton, ON L8S 4K1, Canada

***Abstract*— We propose a new superresolution algorithm based on a fast motion estimation technique. Two stages of this algorithm, namely, motion estimation and high-resolution reconstruction, rely on an area-based interpolation scheme that involves intersecting two pixel grids in arbitrary orientation, displacement, and scaling. We develop a fast approximate solution of the above problem, whose exact solution is prohibitively expensive. Also, gradient descent algorithm is used for fast convergence of the motion estimation algorithm. Experimental results demonstrate the good performance of the proposed superresolution algorithm as well its robustness against noise.**

***Index Terms*— Image registration, superresolution, denoising, gradient descent algorithm.**

I. Introduction

In this paper a fast image registration and reconstruction algorithm is proposed which obtains a high-resolution (HR) frame using a number of adjacent low-resolution (LR) frames. It is assumed that there is only a global motion between the current frame and the reference frames. Although such an assumption is simplistic for real videos, algorithms similar to ours provide an important building block for the more generalized superresolution problem with local motions and occlusion [7]. Furthermore, in its current form the algorithm can be applied to any part of video frames for which motion can be described by a single model.

Accurate estimation of generalized motion is critical for superresolution [2] and many other applications. Our main contribution is a fast generalized motion estimation algorithm. Unlike available motion estimators [7], [9], which are based on optical flow, the proposed algorithm directly minimizes the difference between the current frame pixels and the interpolated pixels of the motion-compensated reference frame. Gradient based iterations of the algorithm ensures fast convergence and accuracy. The algorithm is also robust against high noise level in video frames. In this paper we also introduce a fast HR reconstruction algorithm. Both motion estimation and HR reconstruction rely on area-based interpolation, which is approximated to speed-up the algorithms. Interested readers are directed to [1], [2], [3], [10] for detailed surveys on image registration and superresolution.

II. Superresolution Algorithm

In this work superresolution is achieved in two steps: motion estimation and estimation of high-resolution pixel values. The first step involves solving a non-convex optimization problem, while the second step can be formulated as a convex (linear quadratic) optimization problem by proper choice of regularization.

A. Motion Estimation

We perform motion estimation in three steps: feature selection in the current frame, initial translation motion estimation by block matching to narrow the search range, and generalized motion estimation in real-value precision by modified Newton-Raphson method.

1) Feature Selection: Smooth regions of video frames, due to lack of features, do not have discrimination power in motion estimation. Such regions can be safely disregarded in the motion estimation without loss of precision. This significantly reduces the computational complexity. In this work feature selection is performed only in the current frame. Motion estimation is done by matching selected pixels (with features) of the current frame with pixels of the reference frames.

We extract large-scale edges as good reliable features. The edge map is subject to an erosion operation to exclude isolated pixels selected by the high-pass edge detector because there is a high possibility that the corresponding features have noise origin. This is followed by a dilation step to allow pixels adjacent to the edges to participate in motion estimation. The last step is intended to make the algorithm robust against inaccurate edge detection. For video frames with negligible noise the edge detection step can be replaced by inexpensive high-pass filtering and thresholding. We found that less than 1/20 th of the total number of pixels can be selected for very accurate registration of video sequences.

2) Block Matching: The non-convexity of generalized motion estimation problem makes gradient descent algorithms susceptible to the local minima. To avoid this, in our current work, an initial translation motion estimation is performed to ensure that the starting point for the generalized motion estimation is sufficiently close to the global minima. Let $x(i,j)$ be the pixel value at grid coordinates (i,j) in the current frame and $x_k(i,j)$ the pixel value of the k th reference frame. In addition, let I be the set of the pixels of the current frame selected by the feature selection step. Then the block matching problem between the current frame and the k th reference frame can be described as

$$\min_{d_k^1, d_k^2} J_1 = \sum_{i,j \in I} \left(x(i,j) - x_k(i + d_k^1, j + d_k^2) \right)^2 \qquad (1)$$

The cross-search algorithm [5] is applied for fast implementation of block matching.

3) Generalized Motion Estimation: The problem of estimating motion parameters between the current frame and k th reference frame can be expressed as

$$\min_{v} J_2 = \sum_{i,j \in I} \left(x(i,j) - x^s \left[f_k(i,j,v), v \right] \right)^2 \qquad (2)$$

where $f_k(i,j,v)$ defines the location of pixel (i,j) of the current frame on the grid of the k th reference frame. This mapping depends on the value of motion parameter vector v. Dimension of v depends on the global transformation model. For example, v consists of four parameters if rigid body motion is assumed, six parameters is affine model is assumed, and eight parameters if perspective projection model is assumed. The function $x^s \left[f_k(i,j,v), v \right]$ is an estimate (or interpolation) of $x(i,j)$ using the pixels of the k th reference frame, $f_k(i,j,v)$ and v. Dependence of $x^s[.,.]$ on v is explained as follows. Let us assume two candidate motion parameter vectors v_1 and v_2 with the scaling parameter in v_1 being twice of that in v_2. The estimate of $x(i,j)$ should not be the same in these two cases even if $f_k(i,j,v_1) = f_k(i,j,v_2)$ because v_1 and v_2 suggests different degree of blurring.

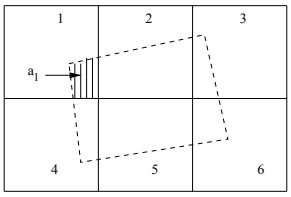

Fig. 1. Location of a pixel of the current frame on the grid of a reference frame.

Different interpolation functions $x^s[.,.]$ may be considered. In this work we choose area-based interpolation. This interpolation can be explained by Fig. 1 which shows the position of one pixel of current frame on the grid of a reference frame. If x_i is the intensity value of pixel i shown in Fig. 1 and a_i is its area of overlap with the pixel of the current frame, then the estimated intensity value of the pixel of the current frame is given by

$$\hat{x} = \frac{1}{A} \sum_{i=1}^{6} a_i x_i \qquad (3)$$

where A is the area of the pixel of the current frame on the grid of the reference frame. This interpolation method is optimal if box point spread function (PSF) and no correlation between neighboring pixels are assumed. Similar area-based interpolation is considered in [8] but for translational motion only.

However, for generalized motion which may consist of rotation, scaling, and shear, the computation of the areas of overlap, a_i in Fig. 1 is nontrivial and expensive. We propose an approximation algorithm for computing a_i. Consider only translational motion as shown in Fig. 2. In this case given the center of the pixel in question, it is inexpensive to compute the areas of overlap and, in turn, the estimated pixel value.

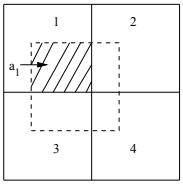

Fig. 2. Location of a pixel of the current frame on the grid of a reference frame assuming only translation.

In fact, the area-based interpolation is same as the bilinear interpolation for upright pixel placement. It can be argued that since the motion between two neighboring video frames can be assumed to be small, the difference between bilinear interpolation and area-based interpolation is negligible when only rigid body motion is considered. For example Fig. 3(a) shows the current frame pixels on a reference grid when motion between the frames is translation and rotation. Fig. 3(b) shows our approximation for interpolation. Unlike the popular block-based motion estimation approach, the proposed method uses different interpolation coefficients for adjacent pixels to account for motions that are more complex than translation. The new technique strikes a good balance between complexity and estimation accuracy by finding a middle ground between oversimplified block-based translational motion estimation and the expensive exact solution. In addition, for non-rigid body

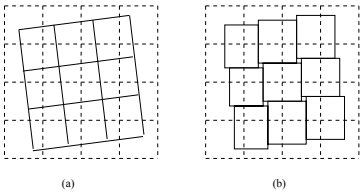

Fig. 3. Translated and rotated current-frame pixels on a reference grid; (a) true pixel location, and (b) our approximation.

motion, approximate height and width of the projected pixels can be computed for more accurate results.

In general cost function J_2 in (2) is non-convex. However, if the initial guess is close to the global minima a gradient descent algorithm can still be applied. Since the motion between two neighboring video frames can be assumed to be small, the block matching step can generate a good initial guess. In this work a modified Newton-Raphson method is applied for solving the optimization problem in (2). If $\nabla_v(J_2)$ is the gradient of the cost function w.r.t. motion parameter vector v and $\nabla_v^2(J_2)$ is the corresponding Hessian, then one iteration of the modified Newton-Raphson algorithm is given by

$$v_{n+1} = v_n - \alpha_{n+1} \left(\nabla_{v_n}^2(J_2) \right)^{-1} \nabla_{v_n}(J_2) \qquad (4)$$

where v_n and v_{n+1} are the estimated motion vectors in the n th and $(n+1)$ th iteration, respectively, and α_{n+1} is a constant such that $0 < \alpha_{n+1} \leq 1$. The algorithm starts with initial estimate v_0 obtained by block-matching. The following steps are required in each iteration

1. Compute $J_2(v_0)$, $\nabla_{v_0}(J_2)$ and $\nabla_{v_0}^2(J_2)$.

2. Set $n = 0$ and $\hat{\alpha} = 1$.
3. Compute $\hat{v} = v_n - \hat{\alpha}\, \nabla^2_{v_n}(J_2)^{-1} \nabla_{v_n}(J_2)$. Reduce $\hat{\alpha}$ if $\hat{v}$ violates any bound on parameters and re-compute $\hat{v}$.
4. If absolute value of $\hat{v} - v_n$ is too small for all parameters, set v_n as the solution. Done.
5. Compute $J_2(\hat{v})$, $\nabla_{\hat{v}}(J_2)$ and $\nabla^2_{\hat{v}}(J_2)$.
6. If $J_2(\hat{v}) < J_2(v_n)$, set $v_{n+1} = \hat{v}$ and $\alpha_{n+1} = \hat{\alpha}$. Go to step 9.
7. $\hat{\alpha} = \hat{\alpha}/4$.
8. Compute $\hat{v} = v_n - \hat{\alpha}\, \nabla^2_{v_n}(J_2)^{-1} \nabla_{v_n}(J_2)$ and go to step 4.
9. If $J_2(v_n) - J_2(v_{n+1}) < t_1$ and $\max\left\{\text{abs}\left(\nabla^2_{v_n}(J_2)\right)\right\} < t_2$, set v_{n+1} as the solution. Done.
10. $n = n + 1$.
11. $\hat{\alpha} = 2\alpha_n$. If $\hat{\alpha} > 1$, then set $\hat{\alpha} = 1$. Go to step 3.

The thresholds t_1 and t_2 are predefined. The algorithm converges within less than 10 function evaluations for all video sequences we experimented on.

B. Superresolution Restoration

Let x^l_k be the pixel values of the kth reference frame in lexicographic order and x^l_0 be the pixel values of the current frame in lexicographic order. We define a vector

$$\mathbf{x}^l = \left[(x^l_0)', (x^l_1)', \cdots, (x^l_N)'\right]' \tag{5}$$

where N is the number of reference frames. Let $\mathbf{x}^h$ denote the HR pixel values, which are to be estimated, in lexicographic order and let F denote a linear interpolation matrix. Each row of F consists of the linear interpolation coefficients that are used along with $\mathbf{x}^h$ to estimate the same row element in $\mathbf{x}^l$. F is a function of the motion between each LR frame w.r.t. the HR frame. Given the above notations and assuming that the interpolation error for each pixel in $\mathbf{x}^l$ be zero mean, independent identically distributed and Gaussian, the optimization problem for superresolution reconstruction is given by

$$\min_{\mathbf{x}^h} J_3 = \left(F\mathbf{x}^h - \mathbf{x}^l\right)' \left(F\mathbf{x}^h - \mathbf{x}^l\right) + \lambda \left(\mathbf{x}^h\right)' B'B\mathbf{x}^h \tag{6}$$

where $\lambda \left(\mathbf{x}^h\right)' B'B\mathbf{x}^h$ is the regularization term. This problem formulation is similar to that in [4] where separate decimation operator, blur matrix and warp matrix are used instead of their combined form given by F. Next, we discuss the choice of F and the iterative procedure for estimating $\mathbf{x}^h$.

1) Fast Interpolation Algorithm: The area-based interpolation method, based on box PSF assumption, is used in this work for estimating each pixel value in each of the LR frames using the pixels of the HR frame. The interpolation coefficients define the rows of F. As discussed in this section, accurate area-based interpolation is computation-intensive and unsuitable for video processing applications. Next, we discuss an approximate interpolation algorithm similar to the one used in motion estimation.

Given that the motion between neighboring video frames is small, the boundaries of each pixel of low resolution frames may be assumed to form a square on the high resolution grid with sides in perfect alignment with the grid axes (see Fig. 4).

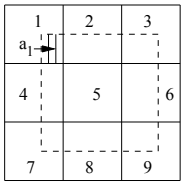

Fig. 4. Approximate boundary of a LR pixel on a HR grid.

Note that for non-rigid motion a rectangle will be a better approximation. The interpolation coefficients corresponding to a LR pixel are proportional to its areas of overlap with HR pixels. The resulting computation is much simpler than computing exact coefficients for area-based interpolation. In Fig. 4 approximate LR pixel location is shown on a HR grid which corresponds to a factor-two scale-up in resolution.

To compute the approximate interpolation coefficients, for each LR pixel, in the simplest case, one needs to compute the position of its center on the high-resolution grid. Note that motion estimation is done on a small fraction of the total number of pixels, whereas for HR reconstruction all of the pixels are needed to be considered. Hence, the computation of interpolation coefficients in the reconstruction stage is the bottleneck of this superresolution algorithm. Computation of F can be parallelized for fast implementation. Furthermore, computational cost can be significantly reduced by assuming fixed interpolation coefficients for a block of pixels when the resolution of the LR video frames is to be scaled up by an integer factor. In this case interpolation coefficients are computed only for a pixel near the center of the block. Other pixels in the block are assumed to have the same coefficients in their shifted overlap region. Size of the block is defined by the maximum acceptable error in computing the location of a pixel. In our work 5×5 pixel blocks are used without any significant reconstruction performance degradation.

2) Iterative HR Frame Estimation Procedure: The optimization problem in (6) has a closed-form solution given by

$$\mathbf{x}^h = (F'F + \lambda B'B)^{-1} F'\mathbf{x}^l \tag{7}$$

However, it involves computationally-expensive large-dimensional matrix inversion. Hence, in this work we use iterative steepest descent method [4], [6]. The steps of the algorithm are given by

1. Set $n = 0$.
2. Initialize $\mathbf{x}^h_0$ with results from bilinear interpolation.
3. $Z = F'\mathbf{x}^l - (F'F + \lambda B'B)\mathbf{x}^h_n$.
4. $\mu = \frac{Z'Z}{Z'(F'F+\lambda B'B)Z}$.
5. $\mathbf{x}^h_{n+1} = \mathbf{x}^h_n + \mu Z$.
6. If $\max\left\{\text{abs}\left(\mathbf{x}^h_{n+1} - \mathbf{x}^h_n\right)\right\} > t_3$, set $n = n + 1$ and go to step 3.
7. Set $\mathbf{x}^h_{n+1}$ as the result. Done.

III. Experimental Results

Representative samples of our superresolution results are shown in Fig. 5-8. Fig. 5 shows that the approximation to the area-based interpolation used in registration and superresolution reconstruction does not change the quality of the output.

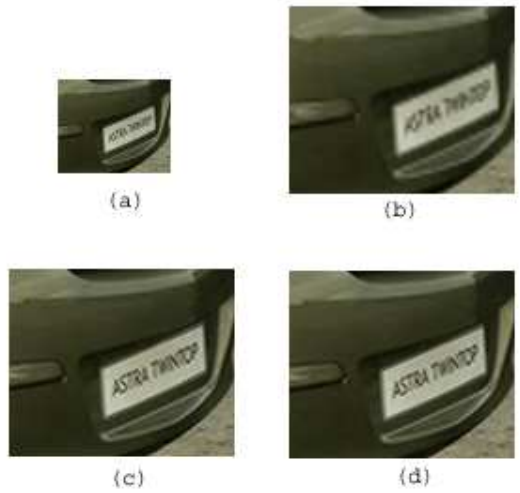

Fig. 5. Result on Car sequence; (a) original, (b) bicubic interpolation, (superresolution result using exact area-based interpolation, and (d) superr olution result using our approximation.

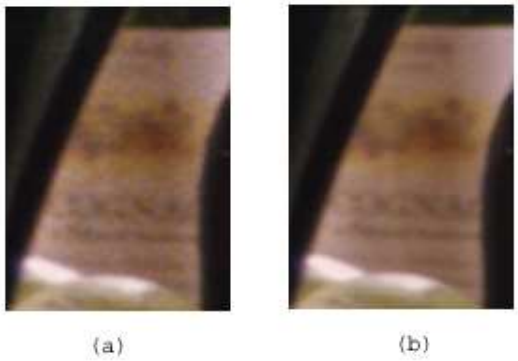

Fig. 6. Results on Italian Wedding sequence; (a) bicubic interpolation, and (b) our superresolution result.

In each case, nine neighboring frames are used for upconverting the current frame by a factor of two. A rigid body motion is assumed and corresponding four parameters are estimated in the registration stage. The true motions are as follows; translation and scaling in the Car sequence, translation in Italian Wedding, rotation and translation in Foreman sequence, and translation and scaling in Calender sequence. Fig. 6 shows robustness against noise of the registration algorithm and denoising capability of the superresolution reconstruction algorithm.

IV. Conclusions

A fast-motion-estimation based superresolution algorithm is proposed for video frames. The results show that temporal processing produces far superior results compared to single frame approach. In addition, robustness of the superresolution algorithm is shown against noise. In future we will compare our algorithm with other superresolution algorithms available in the literature for performance and computational complexity. In addition, we will augment the algorithm to handle multiple local motions and occlusion.

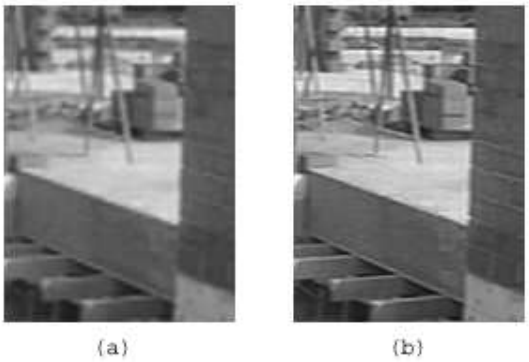

Fig. 7. Results on Foreman sequence; (a) bicubic interpolation, and (b) our superresolution result.

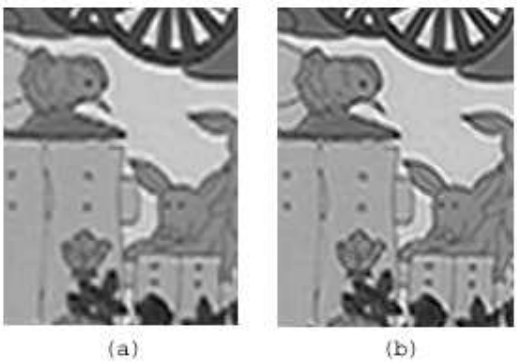

Fig. 8. Results on Calender sequence; (a) bicubic interpolation, and (b) our superresolution result.

References

[1] S. Borman, and R.L. Stevenson, "Spatial Resolution Enhancement of Low-Resolution Image Sequences - A Comprehensive Review with Directions for Future Research," Lab. Image and Signal Analysis, University of Notre Dame, Tech. Rep., 1998.

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

[3] L. G. Brown, "A Survey of Image Registration Techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325376, Dec. 1992.

[4] M. Elad, and A. Feuer, "Superresolution Restoration of an Image Sequence: Adaptive Filtering Approach," *IEEE Trans. on Image Processing*, vol. 8, no. 3, pp. 387-395, March 1999.

[5] M. Ghanbari, "The Cross-Search Algorithm for Motion Estimation," *IEEE Trans. on Communications*, vol. 38, no. 7, July 1990.

[6] L. A. Hageman, and D. Young, *Applied Iterative Methods*, 1 st ed., New York: Academic, 1981.

[7] M. Irani, and S. Peleg, "Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *Journal of Visual Communication and Image Representation*, Vol. 4, No. 4, pp. 324-335, Dec. 1993.

[8] X. Li, and C. Gonzales, "A Locally Quadratic Model for the Motion Estimation Error Criterion Function and Its Application to Subpixel Interpolations," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 123126, Feb. 1996.

[9] R. R. Schultz, L. Meng, and R. L. Stevenson, "Subpixel Motion Estimation for Super-Resolution Image Sequence Enhancement," *Journal of Visual Communication and Image Representation*, Vol. 9, No. 1, pp. 38-50, March 1998.

[10] B. Zitová, and J. Flusser, "Image registration methods: A survey," *Image and Visual Computing*, vol. 21, pp. 977-1000, 2003.