SILK: SCALE-SPACE INTEGRATED LUCAS-KANADE IMAGE REGISTRATION FOR SUPER-RESOLUTION FROM VIDEO

Joseph Lee and Ricardo Gutierrez-Osuna

CSE Department, Texas A&M University
College Station, TX 77843

S. Susan Young

U.S. Army Research Laboratory
Adelphi, MD 20783

ABSTRACT

Registration between low-resolution images is a crucial step in super-resolution. Conventional methods tend to separate scale estimation from translation and rotation estimation. This is because the scale parameter is inherently related to the image resolution. In this paper, we present an area-based image registration technique that can simultaneously estimate translation, rotation, and scale parameters and also take into account differences in resolution between two images. We first develop a scale-space model that relates each reference pixel to a single observation pixel with a scale parameter. This model is then easily generalized to include x-y shift and rotation parameters. By integrating the scale-space model into a non-linear least squares method, the method can iteratively estimate the transformation (x-y shift, rotation, and scale) in an accurate and efficient manner. We compare our proposed scale-space integrated Lucas-Kanade's method (SILK) against Lucas-Kanade's optical flow and scale-invariant feature transform (SIFT) matching and show that our method is suitable for super-resolution from very low resolution image sequences.

Index Terms— Image registration, super-resolution, scale-space, non-linear least squares

1. INTRODUCTION

Image registration is a crucial step in a variety of computer vision tasks, from image stitching to object recognition and super-resolution (SR). In general, registration aims to align two images (the *reference* image and the *observed* image) by estimating translation, rotation, and scale. Mainly two image registration approaches exist: *feature-based* and *area-based* methods; the former tries to align images using local features, whereas the latter uses the whole image region [1]. Classical examples of each approach are scale-invariant feature transform (SIFT) matching [2] and Lucas-Kanade's (LK) method [3], respectively.

In SIFT matching, scale-invariant local features are extracted from the two images using a *scale-space* model [4] and then corresponding features are matched. The advantages of SIFT are that it can register images with high disparity and that it is robust to occlusion. However, if the image region is small, as is the case of very low resolution images, SIFT becomes problematic since the images will not contain enough features to be matched. SIFT is also computationally intensive because it first builds an explicit *image pyramid* [5] structure, which approximates the scale-space, by downscaling the images and then estimates scale changes at every level. In case of a video, where neighbouring frames have similar scale, estimating scale changes at every level becomes unnecessary. Also, since feature matching is usually followed by RANSAC [6], to reject outlier correspondences, the computational load will increase.

On the other hand, the LK method jointly optimizes the affine registration parameters with an area-based approach. The advantage of this method is that it can efficiently register images by searching in the gradient direction and that it avoids the feature correspondence problem by using the whole image. In the LK method, the relationship between the reference image I_r and observed image I_o is expressed as

$$I_o(\mathbf{x}) = I_r(\mathbf{w}(\mathbf{x};\mathbf{p}))\ , \tag{1}$$

where $\mathbf{w}(\mathbf{x};\mathbf{p})$ warps the two-dimensional coordinate $\mathbf{x}$ with affine transformation parameters $\mathbf{p}$ [7]. The algorithm iteratively estimates $\mathbf{p}$ by minimizing $\|I_o(\mathbf{x}) - I_r(\mathbf{w}(\mathbf{x};\mathbf{p}))\|^2$. A drawback of this approach, however, is that equation (1) does not consider changes in image intensity as in SIFT, which occur when the scale changes (i.e., due to image resolution) [8].

Motivated by the limitation of these two classical problems when dealing with LR video, we present an image registration technique for SR that integrates the scale-space model of SIFT into the LK framework; this is achieved by embedding the Gaussian kernels used in the scale-space model into a non-linear least squares estimation. We adopt the area-based approach since the method has to work on very small image regions without many features, while the scale-space model is adopted to handle scale changes correctly. Additionally, the non-linear least squares framework allows us to avoid building a predetermined pyramid structure and compares the two images only at the scale levels selected by the gradient direction. Thus, the method has the efficiency of LK and can simultaneously estimate sub-pixel translation, rotation, and scale while still addressing the resolution issue of equation (1). Though the latter issue can be somewhat alleviated by interpolation or smoothing, our approach handles this issue naturally by explicitly relating the image scale to image resolution. This allows the method to automatically smooth and downsample (anti-alias) correctly in scale-space for each iteration update.

Relation to prior work: Our work is related to SR registration methods based on the LK algorithm. One of the earliest algorithms was proposed by Keren et al. [9]. To obtain high-accuracy sub-pixel motion, the authors developed a spatial domain technique that iteratively estimates translation and rotation. The classical work by Irani and Peleg [10] also applied this registration method and used a back-projection kernel to update the SR image. Recently, Costa et al. [11] examined the effect of registration errors under x-y translation using the LK method; the authors show that some degree of registration errors can be advantageous for SR since they provide some degree of regularization. These previous studies, however, do not consider the pixel intensities affected by image resolution and scale; which our proposed method takes into account. Besides the LK registration, SIFT matching has also been used recently for SR registration. Vrigkas et al. [12] have used SIFT matching to esti-

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

mate registration parameters and generate higher-resolution images using mutual information. Ferreira et al. [13] present an example-based video super-resolution method with mixed-resolution image sequences. However, the images used in these studies are of sufficient resolution to contain many features. In contrast, our aim is to super-resolve images with a small image size with very low resolution, where a 2-fold increase in resolution can have a significant impact.

This paper is organized as follows. Section 2 describes the proposed model for simultaneous estimation of scale, translation and rotation parameters; the method relates a reference image to an observed image with a single scale parameter and incorporates translation and rotation parameters. Section 3 describes how registration parameters are estimated using non-linear least squares. Experimental results on simulated and real data are shown in section 4. We conclude this study and provide future work in section 5.

2. THE IMAGE MODEL

Consider the problem of registering two images $I_o(x,y)$ and $I_r(x,y)$, defined by their pixel intensities at index (x,y). Namely, we wish to find the registration parameters t_x, t_y, θ, and s that minimize the difference between I_r and I_o where (t_x,t_y) is the x-y translation, θ is the rotation angle, and s is the scale factor.

The scale factor between I_r and I_o determines the inherent image resolution. The relationship between these two images can be expressed as

$$I_o(i,j) = \sum_{x,y} G(x,y;i,j,s) I_r(x,y) \ , \tag{2}$$

where (x,y) are the indices of reference image I_r, and (i,j) are the indices of image I_o, and the scale parameter s determines the width of a two-dimensional Gaussian kernel G:

$$G(x,y;x_m,y_m,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_m)^2+(y-y_m)^2}{2\sigma^2}} \ , \tag{3}$$

where σ is the standard deviation and (x_m,y_m) is the mean vector. By using a Gaussian kernel, we ensure that the number of local extrema of image intensity does not increase at coarser levels [4]. The model in equation (2) represents a single observation pixel as a weighted sum of all the pixels in the reference image.

For convenience, we assume that the pixel width and height of the reference image $I_r(x,y)$ are set to unit length. Thus, if the observed image is scaled by a factor s relative to the reference image, an observed pixel will cover a width of $1/s$ in the reference image. Since most of the mass in a Gaussian density is contained within $\pm 3\sigma$, equating the observation pixel and the Gaussian window yields $6\sigma = 1/s$. Therefore, the standard deviation σ can be expressed in terms of the scaling factor s as $\sigma = 1/(6s)$.

The mean location of the kernel $G(x,y;i,j,s)$ can be expressed in terms of the observed image coordinates (i,j) and the observed image pixel width $1/s$ as

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} x_o + (i-1)/s + 1/(2s) \\ y_o + (j-1)/s + 1/(2s) \end{bmatrix} \ , \tag{4}$$

where (x_o,y_o) are the coordinates of the image origin of the observed image. Substituting expressions σ and (x_m,y_m) in the kernel equation (3) yields

$$G(x,y;i,j,s) = \frac{1}{2\pi\left(\frac{1}{6s}\right)^2} e^{-\frac{\left(x-x_o-\frac{2i-1}{2s}\right)^2+\left(y-y_o-\frac{2j-1}{2s}\right)^2}{2\left(\frac{1}{6s}\right)^2}} \ . \tag{5}$$

The above equation expresses the Gaussian weights as a function of the observation pixel locations and scale parameter. Therefore, when combined with equation (2) we are able to relate I_r and I_o with a single scale parameter s. More importantly, because the scale parameter is a continuous variable, the observed image I_o can be simulated at any scale in the scale-space of the reference image I_r. When compared to equation (1), though, the scale parameter is no longer linear as in the transformation $\mathbf{w}(\mathbf{x};\mathbf{p})$.

In order to generalize equation (2), translation and rotation parameters t_x, t_y, and θ are incorporated into the Gaussian kernel mean location. The mean location (x_m,y_m) of the Gaussian window can be transformed by

$$\begin{bmatrix} x'_m \\ y'_m \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \ . \tag{6}$$

Therefore, we can generalize equation (5) as

$$G(x,y;i,j,t_x,t_y,\theta,s) = \frac{1}{2\pi\left(\frac{1}{6s}\right)^2} e^{-\frac{(x-x'_m)^2+(y-y'_m)^2}{2\left(\frac{1}{6s}\right)^2}} \ . \tag{7}$$

Finally, the model equation (2) for the reference image and the observed image becomes

$$I_o(i,j) = \sum_{x,y} G(x,y;i,j,t_x,t_y,\theta,s) I_r(x,y) \ . \tag{8}$$

As in equation (5), the translation and rotation parameters are no longer linear terms. In contrast with the previous models, though, equation (8) allows us to take into account the physical effects of the scale parameter when representing each observed pixel $I_o(i,j)$ as a weighted sum of the reference pixels $I_r(x,y)$.

3. IMAGE REGISTRATION ALGORITHM

Since the registration parameters t_x, t_y, θ, and s are non-linear terms of the model equation (8), we estimate them by non-linear least-squares using Gauss-Newton method [14]. Namely, we wish to find an estimate $\hat{\mathbf{p}}$ that minimizes the objective function

$$J = \frac{1}{2}[\tilde{\mathbf{y}} - \mathbf{f}(\hat{\mathbf{p}})]^T W [\tilde{\mathbf{y}} - \mathbf{f}(\hat{\mathbf{p}})] \ , \tag{9}$$

where $\tilde{\mathbf{y}}$ is the observed image in raster scan order and $\mathbf{f}(\hat{\mathbf{p}})$ is a transformed image of the reference image with registration parameters $\hat{\mathbf{p}}$; i.e., $\mathbf{f}(\hat{\mathbf{p}})$ is obtained from equation (8). The weight matrix W in equation (9) allows us to emphasize certain observation pixels. For the experiment in this study, we use an identity matrix, which puts equal weight to each observed pixel.

An initial value $\mathbf{p}_c$ for $\hat{\mathbf{p}}$ is required to start the estimation process. Since adjacent frames are close to each other, we assume that a good initial estimate between the reference and observed image is provided. For a given estimate $\mathbf{p}_c$, its goodness is computed by the error term $\Delta\mathbf{y}_c = \tilde{\mathbf{y}} - \mathbf{f}(\mathbf{p}_c)$. The Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$, which expresses the linear change of the predicted image at current state $\mathbf{p}_c$, is computed as

$$\frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \begin{bmatrix} \left.\frac{\partial f_1}{\partial p_1}\right|_{\mathbf{p}_c} & \cdots & \left.\frac{\partial f_1}{\partial p_4}\right|_{\mathbf{p}_c} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial f_m}{\partial p_1}\right|_{\mathbf{p}_c} & \cdots & \left.\frac{\partial f_m}{\partial p_4}\right|_{\mathbf{p}_c} \end{bmatrix} \ , \tag{10}$$

where $[p_1, p_2, p_3, p_4]^T = [t_x, t_y, \theta, s]^T$ and m is the number of observation pixels. Once $\Delta\mathbf{y}_c$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ have been computed, the correction term can be expressed as:

$$\Delta\mathbf{p} = \left[\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^T W \left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)\right]^{-1} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)^T W \Delta\mathbf{y}_c\,. \tag{11}$$

Once the correction term $\Delta\mathbf{p}$ has been calculated, the current state estimate $\mathbf{p}_c$ is updated as $\mathbf{p}_c = \mathbf{p}_c + \Delta\mathbf{p}$. The above process continues iteratively until the maximum number of iterations is reached or when the stopping criteria given by $\frac{|J_i - J_{i-1}|}{J_i} < \frac{\varepsilon}{\|W\|}$ is satisfied, where $J_i = \Delta\mathbf{y}_c^T W \Delta\mathbf{y}_c$ is the predicted residual at the i-th iteration and ε is a small value that determines the tolerance [14].

How does the proposed update equation compare to the LK method? In the original LK algorithm, the parameter $\mathbf{p}$ is updated according to:

$$\Delta\mathbf{p} = \left[\sum_{\mathbf{x}} \left(\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{p}}\right)^T \left(\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{p}}\right)\right]^{-1} \sum_{\mathbf{x}} \left(\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{p}}\right) \Delta\mathbf{y}_c\,, \tag{12}$$

where $\Delta\mathbf{y}_c = I_o(\mathbf{x}) - I_r(\mathbf{w}(\mathbf{x}; \mathbf{p}_c))$. When compared to our update equation (11), the LK update in equation (12) multiplies the gradient image $\nabla I_r = (\frac{\partial I_r}{\partial x}, \frac{\partial I_r}{\partial y})$ with the Jacobian warp to compute the steepest descent image $\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{p}}$. The LK update has the advantage that the gradient image is computed only once, and only the Jacobian warp needs to be computed in each iteration. However, the LK model does not correctly represent how the images are affected by scale changes. Instead of using only the gradient image ∇I_r in the x-y directions, our method directly computes the gradient $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ with respect to x-y translation, rotation, and scale in scale-space.

To compare the computational cost with LK, assume n registration parameters and an image I_o of size m. For each iteration, the computational complexity of computing $\nabla I_r \frac{\partial \mathbf{w}}{\partial \mathbf{p}}$ in equation (12) is $O(mn)$ [7]. In our proposed method, when the Gaussian kernel size is k at an iteration, the algorithm will take time $O(kmn)$ to compute $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ according to equation (10).

4. EXPERIMENTAL RESULTS

We tested our proposed method on registering LR license-plate images. The registered images were then used for SR. We first demonstrate the proposed technique on synthetic sequences of LR frames where the ground-truth is known. Then we use real image sequences to generate SR images. For each experiment, we compared our proposed method against both LK optical flow and SIFT matching. The LK optical flow implementation in OpenCV [15] and Lowe's SIFT implementation[1] [16] were used for comparison.

4.1. Synthetic image sequences

For the simulated experiment, we use the 91 images in the license-plate category of the Caltech-256 [17] dataset. We manually cropped the license-plate regions from each image and generated 100 LR frames for each sequence by applying translation, rotation, and scale. For translation, the x-y translation followed a spiral trajectory $(x, y) = (at\cos(2t), at\sin(2t))$ where a is the amplitude and t is the frame index. Throughout the frame sequence, the license-plate moved within ±2 pixels in the LR domain. For rotation, a random value within ±5 degrees was applied to each frame. The downscaling was performed by pixel averaging (as described in [18]) so that each LR sequence was no greater than 40x40 pixels.

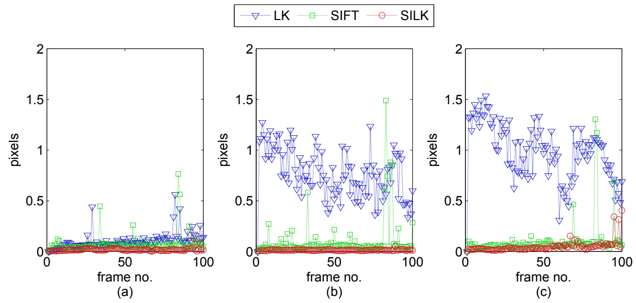

Fig. 1. RMS error between ground-truth and estimated parameters for the three synthesized sequences: (a) translation only, (b) translation and rotation, (c) translation, rotation, and scale. For each scenario, SIFT includes only (a) 18%, (b) 11%, and (c) 10% of the 91 image sequences.

We illustrate the performance of the method on three image registration scenarios: image sequences with (1) x-y translation only, (2) x-y translation and rotation, and (3) x-y translation, rotation, and scaling. For the LK optical flow method, we used a 3x3 kernel to smooth the images beforehand so that the method does not suffer from aliasing effects, and we used a 7x7 window to compute the flow vectors [19]. After computing the dense optical flow of LK, we estimate the global transformation using the flow vectors. Our proposed method did not require any manual smoothing and worked directly on the images.

Fig. 1 shows the root-mean-square (RMS) error of the length $\sqrt{t_x^2 + t_y^2}$ between the ground-truth and estimated parameters (t_x, t_y). In particular, these results indicate that the LK optical flow method can work relatively well when there is only translation. However, as rotation and scale changes are introduced, the LK optical flow becomes less accurate. In addition, and as shown in Fig. 1(b) and (c), registration errors of the LK optical flow can be more than a pixel wide, when sub-pixel accuracy is required for SR. Although SIFT matching appears to be able to effectively estimate the registration in each scenario, Fig. 1 only includes the cases when SIFT was able to find more than 5 feature matches between the reference and observed images. Overall, SILK is almost always more accurate than SIFT matching unless SIFT can find enough features, in which case both algorithms perform comparably.

We also compared SIFT and SILK in terms of run time; although SIFT is known to be computationally intensive for typical sizes of images, our experiment involves only a small image region. Results are summarized in Fig. 2, which shows the average run time over the 9 image sequences in the third scenario of Fig. 1(c) (translation, rotation, and scale). On a dual-core 2.6GHz processor with 6MB cache, SIFT takes an average of 0.20 seconds per frame. In contrast, our method takes 0.08 seconds per frame or twice as fast as the SIFT feature extraction. However, because SILK uses a gradient search, the registration requires more iterations as the images become more separated, as shown in Fig. 2.

Finally, we compared the three algorithms in terms of their ability to produce accurate registrations for SR. Namely, given the estimated registration data from each image sequence, we construct a linear system $\mathbf{y} = H\mathbf{x}$ as in [20] where $\mathbf{x}$ is the unknown high-resolution pixels and H is the weight matrix that relates the high-

[1] http://www.cs.ubc.ca/~lowe/keypoints/

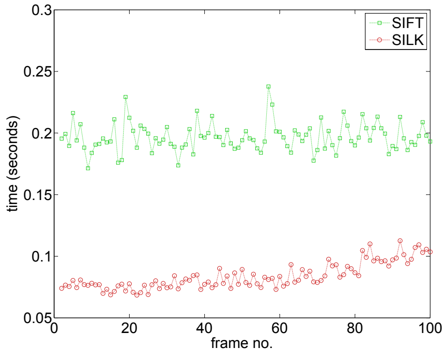

Fig. 2. Speed comparison between SIFT keypoint detection and SILK. Each data point represents the average run time of the third scenario in Fig. 1(c). As shown, SIFT keypoint detection takes approximately constant time for each image. In contrast, registration time for SILK depends on the number of iterations performed for each image match.

resolution pixels to the observation pixels $\mathbf{y}$. Since the matrix H is sparse, we use LSQR [21] to super-resolve the reference image [22]. Fig. 3 shows some examples of the super-resolved images. The SR results are consistent with the previous plots: (1) LK performs well when there is only translation, and (2) SIFT is robust to rotation and scale.

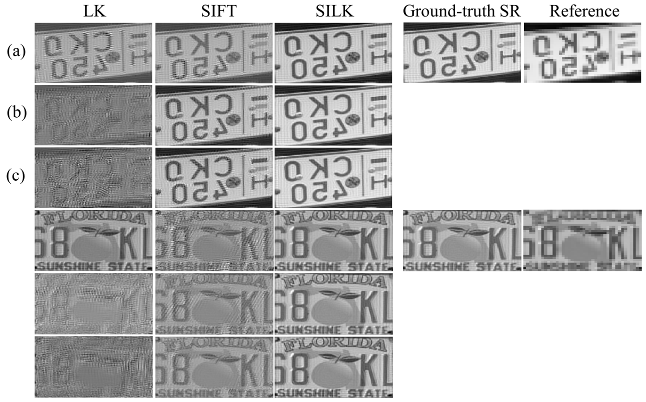

Fig. 3. SR results from synthetic sequences of images; the image sequence was obtained by applying the three transformation scenarios shown in Fig. 1. The reference image (*rightmost*) was super-resolved after aligning the frame sequence. For each sample, the row corresponds to each scenario: (a) translation only (b) translation and rotation (c) translation, rotation, and scale. The first three columns show the reconstruction results from LK optical flow, SIFT matching, and SILK. The fourth column shows the SR result with ground-truth alignment.

4.2. Real image sequences

For our second experiment, we applied the method to register real LR image sequences. We used a web-camera and captured a video of stationary cars from a distance so that the text on the license-plate would be difficult to read. The license-plate regions were smaller than a 20x40 image; see Fig. 4. The web-camera was hand held so that a jittering motion could be applied. For each sample sequence, we captured 300 frames. We also took an image of the license plate at a closer distance to verify the characters on the license plate after super-resolving an image.

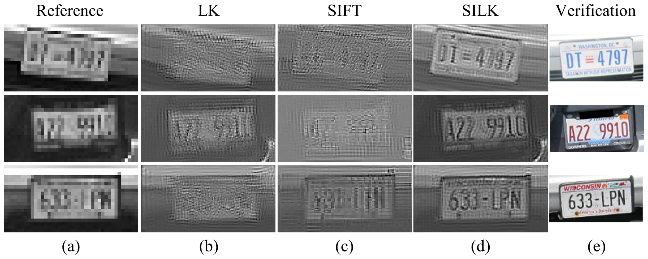

Fig. 4. SR results from registration parameters of real image sequences. The reference image (a) is reconstructed using (b) LK optical flow, (c) SIFT matching, and (d) proposed SILK. Column (e) are the verification images. For each sample sequence, SIFT was able to match 16, 5, and 3 frames out of 300 frames.

The same image fusion method was used as in the previous experiment for SR. Reconstruction results are shown in Fig. 4. After registering the LR images with SILK, we were able to obtain a super-resolved image where the characters on the license-plate become readable. In contrast, SIFT matching failed to find enough features for most of the frames; less than 6% of the frames had more than 5 matches). As shown in the figure, LK optical flow gave poor reconstruction results as well.

5. CONCLUSIONS

In this paper, we have presented an image registration technique (SILK) that can simultaneously estimate translation, rotation, and scale between images and also take into account intensity differences caused by discretization. Compared to SIFT matching, our method assumes that a rough initial estimate is given. However, our method can estimate scale changes as accurately as SIFT and more efficiently in the continuous scale-space. Compared to the original LK method, the improvements in accuracy comes at the price of higher computational complexity; however, when run on LR image regions (e.g., 40x40) the algorithm can run at frame rates on a contemporary office workstation. Our experimental results show that SILK can outperform LK optical flow and SIFT matching for the purpose of SR from very low resolution image sequence when translation, rotation, and scaling are present. Our future work will involve extending our method to use a coarse-to-fine strategy to avoid local minima and develop performance metrics of the proposed registration method.

6. REFERENCES

[1] Barbara Zitová and Jan Flusser, “Image registration methods: a survey,” *Image and Vision Computing*, vol. 21, pp. 977–1000, 2003. 1

[2] D.G. Lowe, “Object recognition from local scale-invariant features,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, vol. 2, pp. 1150 –1157 vol.2. 1

[3] Bruce D. Lucas and Takeo Kanade, “An iterative image registration technique with an application to stereo vision (darpa),” in *Proceedings of the 1981 DARPA Image Understanding Workshop*, April 1981, pp. 121–130. 1

[4] Tony Lindeberg, *Scale-Space*, John Wiley & Sons, Inc., 2007. 1, 2

[5] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, “1984, Pyramid methods in image processing,” *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984. 1

[6] Martin A. Fischler and Robert C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981. 1

[7] Simon Baker and Iain Matthews, “Lucas-kanade 20 years on: A unifying framework,” *Int. J. Comput. Vision*, vol. 56, no. 3, pp. 221–255, Feb. 2004. 1, 3

[8] Göksel Dedeoğlu, Simon Baker, and Takeo Kanade, “Resolution-aware fitting of active appearance models to low resolution images,” in *Proceedings of the 9th European conference on Computer Vision - Volume Part II*, Berlin, Heidelberg, 2006, ECCV’06, pp. 83–97, Springer-Verlag. 1

[9] D. Keren, S. Peleg, and R. Brada, “Image sequence enhancement using sub-pixel displacements,” in *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR ’88., Computer Society Conference on*, jun 1988, pp. 742 –746. 1

[10] Michal Irani and Shmuel Peleg, “Improving resolution by image registration,” *CVGIP: Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, Apr. 1991. 1

[11] G.H. Costa and J.C.M. Bermudez, “Are registration errors always bad for super-resolution?,” in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, april 2007, vol. 1, pp. I–569 –I–572. 1

[12] Michalis Vrigkas, Christophoros Nikou, and Lisimachos P. Kondi, “On the improvement of image registration for high accuracy super-resolution.,” in *ICASSP*. 2011, pp. 981–984, IEEE. 1

[13] Renan U. Ferreira, Edson M. Hung, and Ricardo L. de Queiroz, “Video super-resolution based on local invariant features matching,” in *IEEE International Conference on Image Processing (ICIP)*, Orlando, USA, September 2012. 2

[14] John L. Crassidis and John L. Junkins, *Optimal Estimation of Dynamic Systems (Chapman & Hall/Crc Applied Mathematics & Nonlinear Science)*, Chapman & Hall/CRC, Apr. 2004. 2, 3

[15] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000. 3

[16] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. 3

[17] G. Griffin, A. Holub, and P. Perona, “The Caltech-256,” Tech. Rep., California Institute of Technology, 2007. 3

[18] Simon Baker and Takeo Kanade, “Limits on super-resolution and how to break them,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1167–1183, 2002. 3

[19] Hongche Liu, Tsai-Hong Hong, Martin Herman, Ted Camus, and Rama Chellappa, “Accuracy vs efficiency trade-offs in optical flow algorithms,” *Comput. Vis. Image Underst.*, vol. 72, no. 3, pp. 271–286, Dec. 1998. 3

[20] Zhouchen Lin and Heung-Yeung Shum, “Fundamental limits of reconstruction-based superresolution algorithms under local translation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 83–97, Jan. 2004. 3

[21] Christopher C. Paige and Michael A. Saunders, “Lsqr: An algorithm for sparse linear equations and sparse least squares,” *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982. 4

[22] Nhat Xuan Nguyen, *Numerical Algorithms for Image Superresolution*, Ph.D. thesis, Stanford University, Scientific Computing and Computational Mathematics, July 2000. 4