Volume 2, Issue 4, April 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Overview on: Image Alignment & Open Issues

Deepika Dubey, Abhishek Jain, Uday Pratap Singh
LNCT-Bhopal, India
sweetdeepika1087@gmail.com

Abstract:- In this paper, we cover the quantity approximated, the warp update rule, and the gradient descent approximation. In future papers we will cover the choice of the norm, how to allow linear appearance variation, how to impose priors on the parameters, and various techniques to avoid local minima. Since the Lucas-Kanade algorithm was proposed in 1981 image alignment has be- come one of the most widely used techniques in computer vision. Applications range from optical flow and tracking to layered motion, mosaic construction, and face coding. Numerous algorithms have been proposed and a wide variety of extensions have been made to the original formulation. We present an overview of image alignment, describing most of the algorithms and their extensions in a consistent framework. We concentrate on the inverse compositional algorithm, an efficient algorithm that we recently proposed. We examine which of the extensions to Lucas-Kanade can be used with the inverse compositional algorithm without any significant loss of efficiency, and which cannot.

Keywords: Image alignment, Lucas-Kanade, a unifying framework, additive vs. compositional algorithms, forwards vs. inverse algorithms, the inverse compositional algorithm, efficiency, steepest descent, Gauss-Newton, Newton, Levenberg-Marquardt.

1. INTRODUCTION

Image alignment consists of moving, and possibly deforming, a template to minimize the difference between the template and an image. Since the first use of image alignment in the Lucas- Kanade optical flow algorithm [13], image alignment has become one of the most widely used techniques in computer vision. Besides optical flow, some of its other applications include track- ing [5, 12], parametric and layered motion estimation [4], mosaic construction [16], medical image registration [7], and face coding [2, 8].

The usual approach to image alignment is gradient descent. A variety of other numerical algorithms such as *difference decomposition* [11] and *linear regression* [8] have also been proposed, but gradient descent is the defacto standard. Gradient descent can be performed in variety of different ways, however. One difference between the various approaches is whether they estimate an additive increment to the parameters (the *additive* approach [13]), or whether they estimate an incremental warp that is then composed with the current estimate of the warp (the *compositional* approach [16].) Another difference is whether the algorithm performs a Gauss-Newton, a Newton, a steepest-descent, or a Levenberg-Marquardt approximation in each gradient descent step.

We propose a unifying framework for image alignment, describing the various algorithms and their extensions in a consistent manner. Throughout the framework we concentrate on the *inverse compositional* algorithm, an efficient algorithm that we recently proposed [2]. We examine which of the extensions to Lucas-Kanade can be applied to the inverse compositional algorithm
without any significant loss of efficiency, and which extensions require additional computation. Wherever possible we provide empirical results to illustrate the various algorithms and their extensions.

We proceed in Section 3 to analyze the quantity that is approximated by the various image alignment algorithms and the warp update rule that is used. We categorize algorithms as either *additive* or *compositional*, and as either *forwards* or *inverse*. We prove the first order equivalence of the various alternatives, derive the efficiency of the resulting algorithms, describe the set of warps that each alternative can be applied to, and finally empirically compare the algorithms. In Section 4 we describe the various gradient descent approximations that can be used in each iteration, *Gauss-Newton*, *Newton*, *diagonal Hessian*, *Levenberg-Marquardt*, and *steepest-descent* [14]. We compare these alternatives both in terms of speed and in terms of empirical performance. We conclude in Section 5 with a discussion. In future papers in this series (currently under prepa- ration), we will cover the choice of the error norm, how to allow linear appearance variation, how to add priors on the parameters, and various techniques to avoid local minima.

2. Background

The original image alignment algorithm was the Lucas-Kanade algorithm [13]. The goal of Lucas- Kanade is to align a template image to an input image , where

is a column vector containing the pixel coordinates. If the Lucas-Kanade algorithm is being used to compute *optical flow* or to *track* an image patch from time to time , the template is an extracted sub-region (a window, maybe) of the image at and is the image .

Let the parameterized set of allowed warps, where is a vector of parameters. The warp takes the pixel in the coordinate frame of the template and maps it to the sub-pixel location in the coordinate frame of the image. If we are computing optical flow, for example, the warps might be the translations:

where the vector of parameters is then the optical flow. If we are tracking a larger image patch moving in 3D we may instead consider the set of affine warps where there are 6 parameters as, for example, was done in [4]. (There are other ways to parameterize affine warps. Later in this framework we will investigate what is the best way.) In general, the number of parameters may be arbitrarily large and can be arbitrarily complex. One example of a complex warp is the set of piecewise affine warps used in Active Appearance Models [8, 2] and Active Blobs [15]

Algorithms for aligning images and stitching them into seamless photo-mosaics are among the oldest and most widely used in computer vision. Frame-rate image alignment is used in every camcorder that has an "image stabilization" feature. Image stitching algorithms create the high- resolution photo-mosaics used to produce today's digital maps and satellite photos. They also come bundled with most digital cameras currently being sold, and can be used to create beautiful ultra wide-angle panoramas.

An early example of a widely-used image registration algorithm is the patch-based translational alignment (optical flow) technique developed by Lucas and Kanade (1981). Variants of this algorithm are used in almost all motion-compensated video compression schemes such as MPEG and H.263 (Le Gall 1991). Similar parametric motion estimation algorithms have found a wide variety of applications, including video summarization (Bergen *et al*. 1992a, Teodosio and Bender 1993, Kumar *et al*. 1995, Irani and Anandan 1998), video stabilization (Hansen *et al*. 1994), and video compression (Irani *et al*. 1995, Lee *et al*. 1997). More sophisticated image registration algorithms have also been developed for medical imaging and remote sensing—see (Brown 1992, Zitov'aa and Flusser 2003, Goshtasby 2005) for some previous surveys of image registration techniques.

In the photogrammetry community, more manually intensive methods based on surveyed ground control points or manually registered tie points have long been used to register aerial photos into large-scale photo-mosaics (Slama 1980). One of the key advances in this community was the development of bundle adjustment algorithms that could simultaneously solve for the locations of all of the camera positions, thus yielding globally consistent solutions (Triggs et al. 1999). One of the recurring problems in creating photo-mosaics is the elimination of visible seams, for which a variety of techniques have been developed over the years (Milgram 1975, Milgram 1977, Peleg 1981, Davis 1998, Agarwala et al. 2004)

In film photography, special cameras were developed at the turn of the century to take ultra wide angle panoramas, often by exposing the film through a vertical slit as the camera rotated on its axis (Meehan 1990). In the mid-1990s, image alignment techniques started being applied to the construction of wide-angle seamless panoramas from regular hand-held cameras (Mann and Picard 1994, Szeliski 1994, Chen 1995, Szeliski 1996). More recent work in this area has addressed the need to compute globally consistent alignments (Szeliski and Shum 1997, Sawhney and Kumar 1999, Shum and Szeliski 2000), the removal of "ghosts" due to parallax and object movement (Davis 1998, Shum and Szeliski 2000, Uyttendaele *et al*. 2001, Agarwala *et al*. 2004), and dealing with varying exposures (Mann and Picard 1994, Uyttendaele *et al*. 2001, Levin *et al*. 2004b, Agarwala *et al*. 2004). (A collection of some of these papers can be found in (Benosman and Kang 2001).) These techniques have spawned a large number of commercial stitching products (Chen 1995, Sawhney *et al*. 1998), for which reviews and comparison can be found on the Web.

While most of the above techniques work by directly minimizing pixel-to-pixel dissimilarities, a different class of algorithms works by extracting a sparse set of *features* and then matching these to each other (Zoghlami *et al*. 1997, Capel and Zisserman 1998, Cham and Cipolla 1998, Badra *et al*. 1998, McLauchlan and Jaenicke 2002, Brown and Lowe 2003). Feature-based approaches have the advantage of being more robust against scene movement and are potentially faster, if imple- mented the right way. Their biggest advantage, however, is the ability to "recognize panoramas", i.e., to automatically discover the adjacency (overlap) relationships among an unordered set of im- ages, which makes them ideally suited for fully automated stitching of panoramas taken by casual users (Brown and Lowe 2003).

What, then, are the essential problems in image alignment and stitching? For image align- ment, we must first determine the appropriate mathematical model relating pixel coordinates in one image to pixel coordinates in another. Section 2 reviews these basic *motion models*. Next, we must somehow estimate the correct alignments relating various pairs (or collections) of images. Section 3 discusses how *direct* pixel-to-pixel comparisons combined with gradient descent (and other optimization techniques) can be used to estimate these parameters. Section 4 discusses how distinctive *features*

can be found in each image and then efficiently matched to rapidly establish correspondences between pairs of images. When multiple images exist in a panorama, techniques must be developed to compute a globally consistent set of alignments and to efficiently discover which images overlap one another. These issues are discussed in Section 5.

For image stitching, we must first choose a final compositing surface onto which to warp and place all of the aligned images (Section 6). We also need to develop algorithms to seamlessly blend overlapping images, even in the presence of parallax, lens distortion, scene motion, and exposure differences (Section 6). In the last section of this survey, I discuss additional applications of image stitching and open research problems.

2.1 motion models

Before we can register and align images, we need to establish the mathematical relationships that map pixel coordinates from one image to another. A variety of such *parametric motion models* are possible, from simple 2D transforms, to planar perspective models, 3D camera rotations, lens distortions, and the mapping to non-planar (e.g., cylindrical) surfaces (Szeliski 1996).

To facilitate working with images at different resolutions, we adopt a variant of the *normalized device coordinates* used in computer graphics (Watt 1995, OpenGL-ARB 1997). For a typical (rectangular) image or video frame, we let the pixel coordinates range from [−1, 1] along the longer axis, and [−a, a] along the shorter, where a is the inverse of the *aspect ratio*.

3. Types of image alignment

Once we have chosen a suitable motion model to describe the alignment between a pair of images, we need to devise some method to estimate its parameters. One approach is to shift or warp the images relative to each other and to look at how much the pixels agree. Approaches that use pixel-to-pixel matching are often called *direct methods*, as opposed to the *feature-based methods* described in the next section.

3.1 Direct (pixel based) alignment

To use a direct method, a suitable *error metric* must first be chosen to compare the images. Once this has been established, a suitable *search* technique must be devised. The simplest technique is to exhaustively try all possible alignments, i.e., to do a *full search*. In practice, this may be too slow, so *hierarchical* coarse-to-fine techniques based on image pyramids have been developed. Alternatively, Fourier transforms can be used to speed up the computation. To get sub-pixel precision in the alignment, *incremental* methods based on a Taylor series expansion of the image function are often used. These can also be applied to *parametric motion models*. Each of these techniques is described in more detail below.

3.1.1 Error metrics

The simplest way to establish an alignment between two images is to shift one image relative to the other. Given a *template* image $I_0(x)$ sampled at discrete pixel locations $\{x_i = (x_i, y_i)\}$, we wish to find where it is located in image $I_1(x)$. A least-squares solution to this problem is to find the minimum of the *sum of squared differences* (SSD) function.

3.1.2 Hierarchical motion estimation

Now that we have defined an alignment cost function to optimize, how do we find its minimum? The simplest solution is to do a *full search* over some range of shifts, using either integer or sub- pixel steps. This is often the approach used for *block matching* in *motion compensated video compression*, where a range of possible motions (say ±16 pixels) is explored.[9] To accelerate this search process, *hierarchical motion estimation* is often used, where an image pyramid is first constructed, and a search over a smaller number of discrete pixels (corresponding to the same range of motion) is first performed at coarser levels (Quam 1984, Anandan 1989, Bergen *et al.* 1992a). The motion estimate from one level of the pyramid can then be used to initialize a smaller *local* search at the next finer level.

3.1.3 Fourier-based alignment

When the search range corresponds to a significant fraction of the larger image (as is the case in image stitching), the hierarchical approach may not work that well, since it is often not possible to coarsen the representation too much before significant features get blurred away. In this case, a Fourier-based approach may be preferable. Fourier-based alignment relies on the fact that the Fourier transform of a shifted signal has the same magnitude as the original signal.

3.1.4 Incremental refinement

The techniques described up till now can estimate translational alignment to the nearest pixel (or potentially fractional pixel if smaller search steps are used). In general, image stabilization and stitching applications require much higher accuracies to obtain acceptable results.

To obtain better *sub-pixel* estimates, we can use one of several techniques (Tian and Huhns 1986). One possibility is to evaluate several discrete (integer or fractional) values of (u, v) around the best value found so far and to *interpolate* the matching score to find an analytic minimum.

3.1.5 Parametric motion

Many image alignment tasks, for example image stitching with handheld cameras, require the use of more sophisticated motion models, as described in §2. Since these models typically have more parameters than pure translation, a full search over the possible range of values is impractical. Instead, the incremental Lucas-Kanade algorithm can be generalized to parametric motion models and used in conjunction with a hierarchical search algorithm (Lucas and Kanade 1981, Rehg and Witkin 1991, Fuh and Maragos 1991, Bergen *et al*. 1992a, Baker and Matthews 2004).

3.2 Feature based registration:-

As I mentioned earlier, directly matching pixel intensities is just one possible approach to image registration. The other major approach is to first extract distinctive *features* from each image, to match these features to establish a global correspondence, and to then estimate the geometric transformation between the images. This kind of approach has been used since the early days of stereo matching (Hannah 1974, Moravec 1983, Hannah 1988) and has more recently gained popularity for image stitching applications (Zoghlami *et al*. 1997, Capel and Zisserman 1998, Cham and Cipolla 1998, Badra *et al*. 1998, McLauchlan and Jaenicke 2002, Brown and Lowe.

3.2.1 Key point detector

As we saw in §3.4, the reliability of a motion estimate depends most critically on the size of the smallest eigenvalue of the image Hessian matrix, λ_0 (Anandan 1989). This makes it a reason- able candidate for finding points in the image that can be matched with high accuracy. (Older terminology in this field talked about "corner-like" features (Moravec 1983), but the modern usage is *keypoints*, *interest points*, or *salient points*.) Indeed, Shi and Tomasi (1994) propose using this quantity to find *good features to track*, and then use a combination of translational and affine-based patch alignment to track such points through an image sequence.

3.2.2 Feature matching

After detecting the features (keypoints), we must *match* them, i.e., determine which features come from corresponding locations in different images. In some situations, e.g., for video sequences (Shi and Tomasi 1994) or for stereo pairs that have been *rectified* (Loop and Zhang 1999, Scharstein and Szeliski 2002), the local motion around each feature point may be mostly translational. In this case, the error metrics introduced in §3.1 such as E_{SSD} or E_{NCC} can be used to directly compare the intensities in small patches around each feature point. (The comparative study by Mikolajczyk and Schmid (2005) discussed below uses cross-correlation.) Because feature points may not be ex- actly located, a more accurate matching score can be computed by performing incremental motion refinement as described in §3.4, but this can be time consuming, and can sometimes even decrease.

3.2.3 Geometric registration

Once we have computed a set of matched feature point correspondences, the next step is to estimate the motion parameters p that best register the two images. The usual way to do this is to use least squares, i.e., to minimize the sum of squared residuals given by The above least squares formulation assumes that all feature points are matched with the same accuracy. This is often not the case, since certain points may fall in more textured regions than others. If we associate a variance estimate σ^2 with each correspondence, we can minimize *weighted least squares* instead.

4. Direct vs. feature-based alignment

Given that there exist these two alternative approaches to aligning images, which is preferable?

I used to be firmly in the direct matching camp (Irani and Anandan 1999). Early feature-based methods seemed to get confused in regions that were either too textured or not textured enough. The features would often be distributed unevenly over the images, thereby failing to match image pairs that should have been aligned. Furthermore, establishing correspondences relied on simple cross-correlation between patches surrounding the feature points, which did not work well when the images were rotated or had foreshortening due to homographies.

Today, feature detection and matching schemes are remarkably robust, and can even be used for known object recognition from widely separated views (Lowe 2004). Features not only respond to regions of high "cornerness" (Förstner 1986, Harris and Stephens 1988), but also to "blob-like" regions (Lowe 2004), as well as uniform areas (Tuytelaars and Van Gool 2004). Furthermore, because they operate in scale-space and use a dominant orientation (or orientation invariant de- scriptors), they can match images that differ in scale, orientation, and even foreshortening. My own recent experience in working with feature-based approaches is that if the features are well distributed over the image and the descriptors reasonably designed for repeatability, enough correspondences to permit image stitching can usually be found (Brown *et al*. 2005).

The other major reason I used to prefer direct methods was that they make optimal use of the information available in image alignment, since they measure the contribution of *every* pixel in the image. Furthermore, assuming a Gaussian noise model (or a robustified version of it), they properly weight the

contribution of different pixels, e.g., by emphasizing the contribution of high-gradient pixels. (See Baker *et al*. (2003a), who suggest that adding even more weight at strong gradients is preferable because of noise in the gradient estimates.) One could argue that for a blurry image with only slowly varying gradients, a direct approach will find an alignment, whereas a feature detector will fail to find anything. However, such images rarely occur in practice in consumer imaging, and the use of scale-space features means that some features can be found at lower resolutions.

The biggest disadvantage of direct techniques is that they have a limited range of convergence. Even though they can be used in a hierarchical (coarse-to-fine) estimation framework, in practice it is hard to use more than two or three levels of a pyramid before important details start to be blurred away. For matching sequential frames in a video, the direct approach can usually be made to work. However, for matching partially overlapping images in photo-based panoramas, they fail too often to be useful. Our older systems for image stitching (Szeliski 1996, Szeliski and Shum 1997) relied on Fourier-based correlation of cylindrical images and motion prediction to automatically align images, but had to be corrected by hand for more complex sequences. Our newer system (Brown *et al*. 2004, Brown *et al*. 2005) uses features and has a good success rate at automatically stitching panoramas without any user intervention.

Is there no rôle then for direct registration? I believe there is. Once a pair of images has been aligned with a feature-based approach, we can warp the two images to a common reference frame and re-compute a more accurate estimate using patch-based alignment. Notice how there is a close correspondence between the patch-based approximation to direct alignment given in (103–104) and the inverse covariance weighted feature-based least squares error metric (123).

In fact, if we divide the template images up into patches and place an imaginary "feature point" at the center of each patch, the two approaches return exactly the same answer (assuming that the correct correspondences are found in each case). However, for this approach to succeed, we still have to deal with "outliers", i.e., regions that don't fit the selected motion model due to either parallax (§5.2) or moving objects (§6.2). While a feature-based approach may make it somewhat easier to reason about outliers (features can be classified as inliers or outliers), the patch- based approach, since it establishes correspondences more densely, is potentially more useful for registration.

5. Conclusion

A new technique for subpixel image registration,based on the correlation coefficient maximization, is proposed that pro- vides a closed form solution. Extensive simulation results have shown that the performance of the proposed technique com- pares very favorably with respect to existing ones. The optimum subpixel translation is found with a slight increase in the computational cost, but using the new efficient scheme for pixel level registration as a pre-processing step, the computational complexity of the whole problem is significantly reduced.

REFRENCES

[1] L. Brown, "A survey of image registration TEchniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.

[2] B. Zitová and J. Flusser, "Image registration methods: A sur- vey," *Elsevier Image and Vision Comptuting*, vol. 21, pp. 977–1000, 2003.

[3] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons

[4] H. Foroosh (Shekarforoush), J. B. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp.188–200, Mar. 2002.

[5] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with appli- cation to super-resolution," Accepted to Eurasip JASP (SI on Super-Resolution Imaging: Analysis, Algorithms and Appli- cations).

[6] D. Keren, S. Peleg, and R. Brada, "Image sequence enhance- ment using sub-pixel displacement," in *Proc. of IEEE ICCV- PR*, Jun. 1988, pp. 742–746.

[7] R. J. Althof, M. G. J Wind, and J. T. Dobbins, "A rapid and au- tomatic image registration algorithm with subpixel accuracy," *IEEE Trans. on Med. Imaging*, vol. 16, no. 3, pp. 308–316, Jun. 1997.

[8] S. Periaswamy and H. Farid, "Elastic registration in the pres- ence of intensity variations," *IEEE Trans. on Med. Imaging*, vol. 22, no. 7, pp. 865–874, Jul. 2003.

[9] E. Z. Psarakis and G. D. Evangelidis, "An enhanced correlation-based method for stereo correspondence with sub- pixel accuracy," in *Proc. of 10th IEEE ICCV*, Oct. 2005, Beijing, China.

[10] E. Z. Psarakis and G. D. Evangelidis, "An ENCC based self- weighted similarity measure tailored to the stereo correspon- dence problem," To be submitted to IEEE Trans. on PAMI.

[11] D. I. Barnea and H. F. Silverman, "A class of algorithms for fast digital image registration," *IEEE Trans. on Comp.*, vol. C-21, pp. 179–186, Feb. 1972.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

[13] B. Marcel, M. Briot, and R. Murrieta, "Calcul de translation et rotation par la transformation de Fourier," *Traitement du Signal*, vol. 14, no. 2, pp. 135–149, 1997.

[14] L. Lucchese and G. M. Cortelazzo, "A noise-robust frequency domain technique for estimating planar roto-translations," *IEEE Transactions on Signal Processing*, vol. 48, no. 6, pp.1769–1786, Jun. 2000.
[15] H. Shekarforoush, M. Berthod, and J. Zerubia, "Subpixel im- age registration by estimating the polyphase decomposition of cross power spectrum," *Computer Vison Pattern Recognition*, pp. 532–537, Jun. 1994.

