PARAMETRIC ESTIMATION OF AFFINE DEFORMATIONS OF BINARY IMAGES

Csaba Domokos, Zoltan Kato

Image Processing & Computer Graphics Dept.,
University of Szeged, P.O. Box 652, 6701 Szeged, Hungary,
Email: {dcs,kato}@inf.u-szeged.hu

Joseph M. Francos

Electrical and Computer Engineering Department,
Ben-Gurion University of the Negev,
P.O.B 653 Beer-Sheva 84105, Israel

ABSTRACT

We consider the problem of planar object registration on binary images where the aligning transformation is restricted to the group of *affine* transformations. Previous approaches usually require established correspondences or the solution of nonlinear optimization problems. Herein we show that it is possible to formulate the problem as the solution of a system of up to third order polynomial equations. These equations are constructed in a simple way using some basic geometric information of *binary* images. It does not need established correspondences nor the solution of complex optimization problems. The resulting algorithm is fast and provides a direct solution regardless of the magnitude of transformation.

Index Terms— Image registration, affine transformation, binary image.

1. INTRODUCTION

Registration is a crucial step in almost all image processing tasks where images of different views or sensors of an object need to be compared or combined. Typical application areas include visual inspection, target tracking, super resolution, shape modeling, object recognition, or medical image analysis. In a general setting, one is looking for a transformation which aligns two images such that one image (*template*) becomes similar to the second one (*observation*). Due to the large number of possible transformations, there is a huge variability of the object signature. Hence the problem is inherently *ill-defined* unless this variability is taken into account.

A good survey of affine registration methods can be found in [1, 2]. Basically these algorithms fall into two main categories: *feature-based* methods aim at establishing point correspondences between the two images. For that purpose, they extract some easily detectable landmarks (*e.g.* contours, intersection of lines, corners, etc.) from the images and then use these landmarks to establish correspondences based on a similarity metric. A common assumption needed to find good matches is that the unknown transformation is close to identity (*i.e.* the strength of the deformation is limited). On the other hand *area-based* methods treat the problem without attempting to detect salient objects. It searches the position on the *observation* where the matching of the two object is the best and based on this looks for sufficient alignment between the images.

The parametric estimation of two-dimensional affine transformations between two *gray-level* images has been addressed by Hagege and Francos in [3] which provides an accurate and computationally simple solution avoiding both the correspondence problem as well as the need for optimization. The basic idea is to reformulate the original problem as an *equivalent linear parameter estimation* one which can be easily solved. This solution, however, makes use of the radiometric information which is not available in binary images. Herein we propose an extension of these ideas to the binary case.

2. ESTIMATION OF AFFINE TRANSFORMATIONS

Let us denote the points of the *template* and the *observation* by $\mathbf{x}, \mathbf{y} \in \mathbb{P}^2$ respectively (*i.e.* we use homogeneous coordinates). $\mathbf{A}$ is the unknown affine transformation that we want to recover. We can define the identity relation as follows

$$\mathbf{A}\mathbf{x} = \mathbf{y} \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{y}. \tag{1}$$

If we can observe some image features (*e.g.* gray-level of the pixels) that are invariant under the transformation $\mathbf{A}$ then the following equality also holds

$$g(\mathbf{x}) = h(\mathbf{A}\mathbf{x}) \quad \Leftrightarrow \quad g(\mathbf{A}^{-1}\mathbf{y}) = h(\mathbf{y}). \tag{2}$$

Furthermore, the above equations still hold when an *invariant function* $\omega : \mathbb{R}^n \to \mathbb{R}^n$ is acting on both sides of the equations. Indeed, for a properly chosen ω

$$\omega(\mathbf{x}) = \omega(\mathbf{A}^{-1}\mathbf{y}), \quad \text{and} \tag{3}$$

$$\omega(g(\mathbf{x})) = \omega(h(\mathbf{A}\mathbf{x})) = \omega(h(\mathbf{y})). \tag{4}$$

The basic idea of the proposed approach is to generate enough linearly independent equations by making use of the relations in Eq. (1)–(4). Since $\mathbf{A}$ has 6 unknown elements, we need at least 6 equations.

2.1. Basic solution using polynomial equations

Since binary images do not contain radiometric information, they can be represented by their characteristic function $\mathbb{1} : \mathbb{P}^2 \rightarrow \{0,1\}$, where 0 and 1 correspond to the background and foreground respectively. Therefore in the binary case Eq. (2) becomes

$$\mathbb{1}_t(\mathbf{x}) = \mathbb{1}_o(\mathbf{A}\mathbf{x}) \quad \Leftrightarrow \quad \mathbb{1}_t(\mathbf{A}^{-1}\mathbf{y}) = \mathbb{1}_o(\mathbf{y}), \qquad (5)$$

where $\mathbf{A}$ is the transformation aligning the *template* (denoted by $\mathbb{1}_t$) and *observation* (denoted by $\mathbb{1}_o$). It is clear that a characteristic function is not rich enough so we cannot use Eq. (5) to generate new equations. Hence we have to use the identity relation to impose new constraints by applying $\omega : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ to the coordinates of the points like in Eq. (3). Thus multiplying both sides of Eq. (5) by $\omega(\mathbf{x})$ yields a new equation. Furthermore, because of the lack of point correspondences, we have to integrate over $\mathbb{R}^2$ yielding

$$\int_{\mathbb{R}^2} \omega(\mathbf{x})\mathbb{1}_t(\mathbf{x})d\mathbf{x} = \frac{1}{|\mathbf{A}|}\int_{\mathbb{R}^2} \omega(\mathbf{A}^{-1}\mathbf{y})\mathbb{1}_o(\mathbf{y})d\mathbf{y}, \qquad (6)$$

where we have used the integral transformation $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$, $d\mathbf{x} = |\mathbf{A}|^{-1}d\mathbf{y}$. The Jacobian $|\mathbf{A}|$ (*i.e.* the measure of the transformation $\mathbf{A}$) can then be evaluated through Eq. (5), hence

$$\int_{\mathbb{R}^2} \mathbb{1}_t(\mathbf{x}) = |\mathbf{A}^{-1}| \int_{\mathbb{R}^2} \mathbb{1}_o(\mathbf{y}) \Rightarrow |\mathbf{A}| = \frac{\int_{\mathbb{R}^2} \mathbb{1}_o(\mathbf{y})}{\int_{\mathbb{R}^2} \mathbb{1}_t(\mathbf{x})}. \qquad (7)$$

The sign ambiguity of the determinant can be easily eliminated: A negative Jacobian would mean that the transformation is not orientation-preserving. In practice, however, such transformations are usually excluded by physical constraints.

Since the characteristic functions take only values from $\{0,1\}$, we can further simplify the integrals in Eq. (6):

$$\int_{\mathbb{R}^2} \omega(\mathbf{x})\mathbb{1}(\mathbf{x}) = \int_{\mathcal{F}} \omega(\mathbf{x})\mathbb{1}(\mathbf{x}) = \int_{\mathcal{F}} \omega(\mathbf{x}),$$

where the domain $\mathcal{F}$ consists of the foreground regions: $\mathcal{F} = \{\mathbf{x} \in \mathbb{P}^2 | \mathbb{1}(\mathbf{x}) = 1\}$. Therefore evaluating the integrals in Eq. (7) yields the *area* of the foreground regions. From this point of view, the measure of the transformation $|\mathbf{A}|$ corresponds to the ratio of the *observation* and *template* objects' area. In the remaining part of this paper, we will always integrate over the respective domain $\mathcal{F}$ unless otherwise noted. So we get the simplified version of Eq. (6)

$$|\mathbf{A}| \int \omega(\mathbf{x}) = \int \omega(\mathbf{A}^{-1}\mathbf{y}). \qquad (8)$$

In order to obtain a polynomial system from Eq. (8), the applied ω functions should be carefully selected. We found that if there exists a set of three-variate polynoms $\{p_i | p_i \in \mathbb{R}[x_1, x_2, x_3]$ and $\deg(p_i) \geq 1, i = 1, \ldots, n\}$ and a set of functions $\{f_i\}_{i=1}^n : \mathbb{P}^2 \rightarrow \mathbb{R}$ such that

$$\int \omega^{(k)}(\mathbf{x}) = \sum_{i=1}^{n} p_i(\mathbf{A}_k^{-1}) \int f_i(\mathbf{y}), \quad k = 1, 2$$

is satisfied then the resulting equations are polynomials. $\omega^{(k)}(\mathbf{x})$ is the kth coordinate of $\omega(\mathbf{x})$ and $\mathbf{A}_k^{-1}$ is the kth row of $\mathbf{A}^{-1}$. It is easy to see that the class of $x^n (n \in \mathbb{N})$ functions and their linear combinations achieve this property. Thus setting $\omega^{(k)}(\mathbf{x}) = x_k^n$, Eq. (8) becomes

$$|\mathbf{A}| \int x_k^n = \sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} q_{k1}^{n-i} q_{k2}^{i-j} q_{k3}^{j} \int y_1^{n-i} y_2^{i-j}, \qquad (9)$$

where $k = 1, 2$; $n = 1, 2, 3$ and q_{ki} denotes the unknown elements of the inverse transformation $\mathbf{A}^{-1}$. Note that in the coefficients we can recognize the first, second and third order moments of the *template* and *observation*.

The system of Eq. (9) contains six polynomial equations up to order three which is enough to solve for all unknowns. In fact there are two independent systems ($k = 1, 2$), each consisting of three equations. Unlike in [3], however, this is not a linear system thus we have up to six possible solutions for each unknown q_{ki} due to the cubic polynomial equations. Out of these potential solutions, we can get the right one by dropping the complex roots and selecting the transformation whose determinant matches the Jacobian computed in Eq. (7). The *uniqueness* of the solution is then guaranteed as long as the observation and *template* are not affine symmetric [3]. We also remark that the integrals in Eq. (9) need to be evaluated only once hence the complexity of the algorithm depends linearly on the size of the objects.

3. NUMERICAL IMPLEMENTATION

We have constructed our polynomial equations in the continuum but in practice we only have a limited precision digital image. Hence the integral over the domain $\mathcal{F}$ can only be *approximated* by a discrete sum over the foreground pixels, for example

$$\int y_1 \approx \sum_{i=1}^{n} y_1^{(i)},$$

where n is the number of pixels and $y_1^{(i)}$ is the first coordinate of the ith point. Clearly, the resolution of the images affect the precision of this approximation. As the mesh size tends to zero, the finite sums approximate better the integral. Therefore, our method performs better on higher resolution images. However, due to its linear time complexity, the algorithm runs quite fast on huge images thus we do not have to compromise quality when CPU time is critical.

The numerical error caused by large pixel coordinates can be an issue, especially in our case where these coordinates

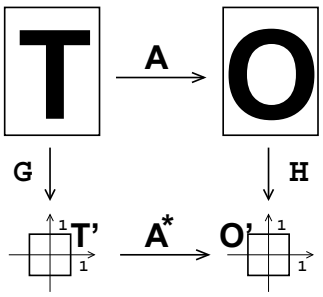


Fig. 1. Normalization of the *template* (T) and *observation* (O). First the origin is translated into the center of the image then the coordinates are scaled to $[-1,1]\times[-1,1]$.

need to be raised to power three. A standard technique to minimize this error is to normalize the image by transforming it into $[-1,1]\times[-1,1]$. Such a transformation is composed of a translation of the origin into the center of the image followed by an appropriate scaling. As shown on Fig. 1, applying the normalizing transformation $\mathbf{G}$ to the *template* and $\mathbf{H}$ to the *observation*, the algorithm will now recover the transformation $\mathbf{A}^*$ which aligns the normalized images O' and T'. Since our equations are based on integral transform, the normalization has to be taken into account as it affects the measure of integrals. Indeed, $\mathbf{x}' = \mathbf{T}\mathbf{x}$ and $d\mathbf{x}' = |\mathbf{T}|d\mathbf{x}$, where $|\mathbf{T}|$ is the Jacobian of the transformation ($\mathbf{T}$). Hence

$$\int \omega(\mathbf{x}')d\mathbf{x}' = |\mathbf{T}| \int \omega(\mathbf{T}\mathbf{x})d\mathbf{x},$$

where the left hand side of the equation is the integral on the normalized image based on the modified measure. Therefore when normalizing the images, the left and right hand sides of the equations in Eq. (9) have to be multiplied by $|\mathbf{G}|$ and $|\mathbf{H}|$ respectively and the Jacobian of the transformation is

$$|\mathbf{A}^*| = \frac{|\mathbf{H}| \int \mathbb{1}_o(\mathbf{y})}{|\mathbf{G}| \int \mathbb{1}_t(\mathbf{x})} = \frac{|\mathbf{H}|}{|\mathbf{G}|}|\mathbf{A}|.$$

Once the transformation $\mathbf{A}^*$ aligning the normalized images is obtained, the original transformation $\mathbf{A}$ is recovered by unnormalizing $\mathbf{A}^*$:

$$\mathbf{A} = \mathbf{H}^{-1}\mathbf{A}^*\mathbf{G}.$$

4. EXPERIMENTAL RESULTS

The proposed algorithm has been tested on a large database of binary images of size 1000×1000. The dataset consists of 39 different shapes and their transformed versions, a total of more than 40000 images. Some typical examples of these images can be seen in Fig. 2. In order to quantitatively evaluate the results, we have defined two kind of error measures. The first one (denoted by ϵ) measures the distance between the true transformation $\mathbf{A}$ and $\widehat{\mathbf{A}}$ obtained by our algorithm.

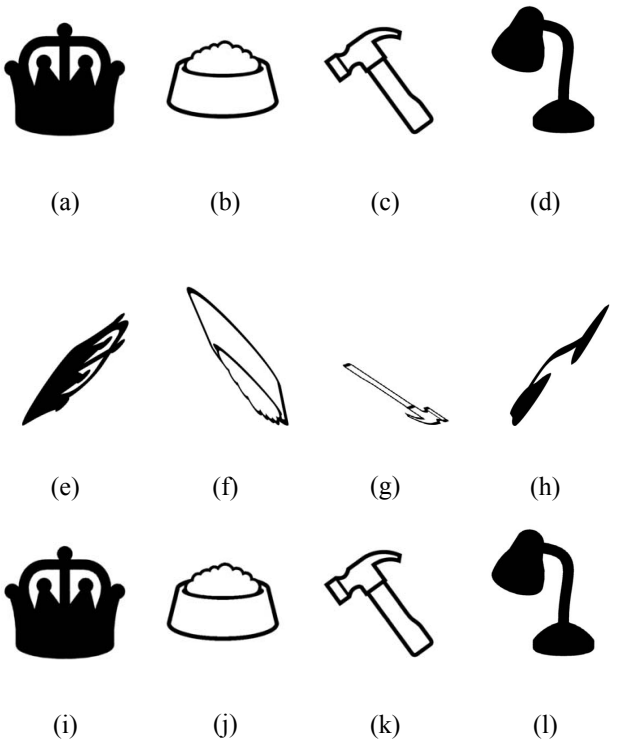


Fig. 2. Registration results of the proposed algorithm. (a)–(d) *Template* images; (e)–(h) *Observation*s of the first row; (i)–(l) Images obtained by applying the inverse of the recovered transformation to the *observations* in (e)–(h).

Another measure is the absolute difference (denoted by δ) between the *observation* and the *registered* image.

$$\epsilon = \frac{1}{m}\sum_{\mathbf{p}} \frac{\|(\mathbf{A}-\widehat{\mathbf{A}})\mathbf{p}\|}{\|\mathbf{A}\mathbf{p}\|}, \quad \text{and} \quad \delta = \frac{|R \triangle O|}{|R|+|O|},$$

where m is the number of *template* pixels (denoted by $\mathbf{p}$), $\triangle$ means the symmetric difference, while R and O denote the set of pixels of the *registered* image and *observation* respectively. The smaller these numbers are, the better is the matching. The median of these errors on the whole database was $\epsilon = 2.91$ pixels and $\delta = 0.46\%$. The algorithm has been implemented in Matlab 7.2 and ran on a SunFire V490 under Solaris 10 operating system. The average runtime was around one second including the normalization, the computation of the integrals and the solution of the polynomial system.

Although we have to solve a nonlinear system of equations, the proposed method provides the result without any iterative optimization step or correspondence. The time complexity is $\mathcal{O}(N)$, where N is the number of the pixels of the images. Clearly, most of the time is spent on parsing the foreground pixels. Note however, that all the integrals can be computed in a single pass over the images.

Table 1. Registration results on 1000 randomly chosen images using the method of Kannala *et al.* [6] and the proposed algorithm.

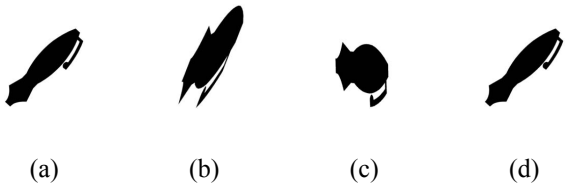
	Runtime (sec.)	ϵ (pixel)	δ (%)
Kannala *et al.* [6]	100.87	16.61	13.41
Proposed	1.06	2.87	0.42

4.1. Comparison to previous approaches

Flusser *et al.* propose an image registration algorithm based on affine moments in [4]. First they extract some representative regions and compute their moments, then the regions from the template and observation are matched based on the similarity of their moments. Then point correspondences are established as the centers of the region pairs and the transformation is recovered in a classical way by solving a system of equations constructed from the point correspondences. While both methods make use of moments, the fundamental difference is that our method provides a direct solution without any point correspondences.

Another recent approach for binary registration of images has been presented in [5]. In fact, the method addresses the registration of images taken under very different lighting conditions or in different seasons. Hence it is not possible to directly measure an invariant image feature as shown in Eq. (2). To overcome this difficulty, the authors extract edges from the images and compute some statistics of the edges which are used as a similarity metric for matching features. Although we address a different problem, this approach demonstrates the importance of the registration of *binary* images. In many cases, the variability of the object signatures is so complex that the only feasible way to register such images is to reduce them to a binary representation and solve the registration problem in that context.

Probably the most closely related approach is the binary registration algorithm proposed by Kannala *et al.* [6]. The fundamental difference is that [6] constructs a system of equations by looking at the images at 3 different scales. Although the resulting system is linear, the solution is inherently less precise as in each equation they can only use part of the available information. On the other hand, our approach constructs the equations by making use of the invariant functions ω hence we always use all the information available in the images. We have obtained the Matlab implementation from the authors and conducted a comparative test on 1000 randomly chosen images from our database. The results presented in Table 1 and Fig. 3 show that our method outperforms [6] in both quality and computing time.



(a) (b) (c) (d)

Fig. 3. (a) *Template* image; (b) The *observation*; (c) Registration result of [6]; (d) Result obtained by the proposed method. The latter two images has been obtained by applying the inverse of the recovered transformation ($\widehat{\mathbf{A}}^{-1}$) to the *observation*.

5. CONCLUSIONS

In this paper, we have presented a novel approach for binary image registration. The fundamental difference compared to classical image registration algorithms is that our model works without any landmark extraction, correspondence, or iterative optimization. It makes use of all the information available in the input images and constructs a polynomial system of equations which can be solved exactly. The complexity of the algorithm is linear hence it is potentially capable of registering images at near real-time speed.

6. REFERENCES

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

[2] Barbara Zitová and Jan Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, October 2003.

[3] Rami Hagege and Joseph M. Francos, "Linear estimation of sequences of multi-dimensional affine transformations," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006, IEEE, vol. 2, pp. 785–788.

[4] Jan Flusser and Tomàš Suk, "A moment-based approach to registration of images with affine geometric distortion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 2, pp. 382–387, March 1994.

[5] Katherine M. Simonson, Steven M. Drescher, and Franklin R. Tanner, "A statistics-based approach to binary image registration with uncertainty analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 112–125, January 2007.

[6] Juho Kannala, Esa Rahtu, Janne Heikkilä, and Mikko Salo, "A new method for affine registration of images and point sets," in *Proceedings of Scandinavian Conference on Image Analysis*, Joensuu, Finland, June 2005, vol. 3540 of *Lecture Notes in Computer Science*, pp. 224–234, Springer.