Parametric Estimation of Affine Deformations of Planar Shapes [*]

Csaba Domokos [a] Zoltan Kato [a,*]

[a] *Department of Image Processing and Computer Graphics,*
University of Szeged, P.O. Box 652, H-6701 Szeged, Hungary

Abstract

We consider the estimation of affine transformations aligning a known 2D shape and its distorted observation. The classical way to solve this registration problem is to find correspondences between the shapes and then compute the transformation parameters from these landmarks. Here we propose a novel approach where the exact transformation is obtained as the solution of a polynomial system of equations. The method has been tested on synthetic as well as on real images and its robustness in the presence of segmentation errors and additive geometric noise has also been demonstrated. We have successfully applied the method for the registration of hip prosthesis X-ray images. The advantage of the proposed solution is that it is fast, easy to implement, has linear time complexity, works without established correspondences and provides an exact solution regardless of the magnitude of transformation.

Key words: Image registration, shape matching, affine transformation

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 in video

[*] Corresponding author. Fax:+36 62 546 397, Tel:+36 62 546 399

Email addresses: dcs@inf.u-szeged.hu (Csaba Domokos), kato@inf.u-szeged.hu (Zoltan Kato).

URLs: http://www.inf.u-szeged.hu/~dcs (Csaba Domokos), http://www.inf.u-szeged.hu/~kato (Zoltan Kato).

sequences, super resolution, or medical image analysis. In a general setting, one is looking for a transformation which aligns two images such that one image (called the *observation*) becomes similar to the second one (called the *template*). Due to the large number of possible transformations, there is a huge variability of the object signature. In fact, each *observation* is an element of the orbit of the transformations applied to the *template*. Hence the problem is inherently *ill-defined* unless this variability is taken into account. A good survey of registration methods can be found in [1–5]. Basically registration algorithms fall into two main categories: *Feature-based* and *Area-based* methods.

Feature-based methods [6,7] aim at establishing point correspondences between two images. For that purpose, they extract some easily detectable features (*e.g.* intersection of lines, corners, etc.) from the images and then use these points to compute the closest transformation based on a similarity metric. Therefore, to make this approach feasible, the correspondence problem must be solved first. Unfortunately, the solution of this problem is far from trivial and usually relies on the assumption that the deformation is close to identity and that features provide a strong contextual evidence for matching landmark points. Other approaches based on "continuous landmarks", such as curves representing the boundaries of objects, usually yield to a complex, non-convex optimization problem requiring computationally expensive algorithms to solve. Searching for the best transformation usually requires an iterative algorithm like the Iterative Closest Point (ICP) algorithm [8]. The main drawback of these methods is that an optimization procedure has high computational cost. In addition, the reliable solution of the correspondence problem assumes that the transformation is close to identity. The main advantage of these methods is that as long as a sufficient number of point matches are available, one can usually find an optimal aligning transformation implying that these algorithms are less sensitive to occlusions.

Area-based methods [9–13] treat the problem without attempting to detect salient objects. These methods are sometimes called correlation-like methods because they use a rectangular window to gain some preliminary information about the distortion. They search the position in the *observation* where the matching of the two windows is the best and then look for sufficient alignment between the windows in the *template* and in the *observation*. When the distortion is small, often Fourier descriptors [14] or mutual information [15] is used. The drawback of this family of methods is also the high computational cost and the restricted range of distortions.

In many situations, the variability of image features 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 [16]. X-ray images are good examples as they usually exhibit highly nonlinear radiometric distor-

tions [17,18] making registration hard to solve. Therefore binary registration (*i.e.* shape matching) is an important problem for many complex image analysis tasks.

Several techniques have been proposed to address the affine registration problem. By thresholding the magnitude of Fourier Transform of the images Zhang et al. [19] construct affine invariant features, which are insensitive to noise, in order to establish point correspondence. Several Fourier domain based methods [20,21] represent images in a coordinate system in which the affine transformation is reduced to an anisotropic scaling factor, which can be computed using cross correlation methods. Govindu and Shekar [22] develop a framework that uses the statistical distribution of geometric properties of image contours to estimate the relevant transformation parameters. Main advantages of these methods is that they do not need point correspondences across views and images may also differ by the overall level of illumination. A novel one-element voxel attribute, the distance-intensity (DI) is defined in [23]. This feature encodes spatial information at a global level, and the distance of the voxel to its closest object boundary, together with the original intensity information. Then the registration is obtained by exploiting mutual information as a similarity measure on the DI feature space. For matching 2D feature points, [24] reduces the general affine case to the orthogonal case by using the means and covariance matrices of the point sets, then the rotation is computed as the roots of a low-degree complex coefficients polynomial. Another direct approach [25] extends the given pattern to a set of affine covariant versions, each carrying slightly different information, and then extract features for registration from each of them separately. The transformation is parameterized at different scales, using a decomposition of the deformation vector field over a sequence of nested (multiresolution) subspaces in [26]. An energy function describing the interactions between the images is then minimized under a set of constraints, ensuring that the transformation maintains the topology in the deformed image. Manay et al. [27] explore an optimization framework for computing shape distance and shape matching from integral invariants, which are employed for robustness to high-frequency noise. Shape warping by the computation of an optimal reparameterization allows this method to account for large localized changes such as occlusions and configuration changes. In [28] a method for identifying silhouettes from a given set of Radon projections is presented. The authors study how the Radon transform changes when a given 2D function is subjected to rotation, scaling, translation, and reflection. Using these properties, the parameters of the aligning transformation are expressed in terms of the Radon transform. In [9] a computationally simple solution is proposed to the affine registration of *gray level* images avoiding both the correspondence problem as well as the need for optimization. The original problem was reformulated as an *equivalent linear parameter estimation* one having a *unique* and *exact* solution. However, the method relies on the availability of rich radiometric information which is clearly not available in the binary case.

In this paper, as an extension of our previous work [29], we propose a novel method which provides an accurate and computationally simple solution to the affine registration of planar shapes. The main difficulty with binary images is that they do not contain radiometric information, only the foreground pixel coordinates are available for the registration algorithm. We will show how the binary registration problem can be formulated as the solution of a system of polynomial equations obtained by integrating a set of polynomial functions over the shape domains. The proposed method provides a direct solution without established correspondences or optimization. Moreover, the robustness of the resulting algorithm in the presence of an i.i.d. Gaussian noise on the point coordinates as well as segmentation errors are also demonstrated. On the other hand, being an *area-based* method, it is sensitive to occlusions. Comparative tests on partially occluded shapes reveal, however, that other *area-based* state of the art methods cannot cope with occlusion either. Finally, we demonstrate the performance on real images and apply the method to align pairs of hip prosthesis X-ray images.

This paper organizes as follows. In Section 2, we present our approach and the proposed algorithmic solution. Then in Section 3, we will analyze the robustness of the proposed method in the presence of an additive Gaussian noise on the pixel coordinates. Finally, experimental results and comparative tests are presented in Section 4.

2   Estimation of Affine Transformations

Let us denote the homogeneous coordinates of the *template* and *observation* points by $\mathbf{x} = [x_1, x_2, 1]^T \in \mathbb{P}^2$ and $\mathbf{y} = [y_1, y_2, 1]^T \in \mathbb{P}^2$ respectively. The identity relation between the two images is then as follows

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

where $\mathbf{A}$ is the unknown affine transformation that we want to recover:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{A}^{-1} = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that $\mathbf{A}^{-1}$ exists and it is also an affine transformation since $\mathbf{A}$ is affine. Classical landmark-based approaches would now identify at least 3 point pairs $\left(\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n \geq 3} \right)$ and solve the system of linear equations obtained from Eq. (1). However, we are interested in a direct solution without solving the correspon-

dence problem. For that purpose, we will take the Lebesgue integral[1] of both sides of the identity relation

$$\int_{\mathbb{P}^2} \mathbf{x} d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathbb{P}^2} \mathbf{A}^{-1} \mathbf{y} d\mathbf{y}, \tag{2}$$

where the integral transformation $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$, $d\mathbf{x} = d\mathbf{y}/|\mathbf{A}|$ has been applied. The determinant $|\mathbf{A}|$ is the Jacobian which corresponds to the measure of the transformation. Furthermore, let shapes 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. If we denote the *template* by $\mathbb{1}_t$ and the *observation* by $\mathbb{1}_o$, then Eq. (1) implies

$$\mathbb{1}_t(\mathbf{x}) = \mathbb{1}_o(\mathbf{A}\mathbf{x}) = \mathbb{1}_o(\mathbf{y}). \tag{3}$$

The Jacobian can then be evaluated by integrating

$$\int_{\mathbb{P}^2} \mathbb{1}_t(\mathbf{x}) d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathbb{P}^2} \mathbb{1}_o(\mathbf{y}) d\mathbf{y} \quad \Leftrightarrow \quad |\mathbf{A}| = \frac{\int_{\mathbb{P}^2} \mathbb{1}_o(\mathbf{y}) d\mathbf{y}}{\int_{\mathbb{P}^2} \mathbb{1}_t(\mathbf{x}) d\mathbf{x}}.$$

Since the characteristic functions take only values from $\{0, 1\}$, we can further simplify the above integrals by making use of the relation:

$$\int_{\mathbb{P}^2} \mathbb{1}_t(\mathbf{x}) d\mathbf{x} \equiv \int_{\mathcal{D}} d\mathbf{x},$$

where the finite domain $\mathcal{D}$ consists of the *template* foreground regions: $\mathcal{D} = \{\mathbf{x} \in \mathbb{P}^2 | \mathbb{1}_t(\mathbf{x}) = 1\}$. Similarly, we can restrict the integral of $\mathbb{1}_o(\mathbf{y})$ to the *observation* foreground regions $\mathcal{F}$. Therefore evaluating the integrals 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* shapes' area

$$|\mathbf{A}| = \frac{\int_{\mathcal{F}} d\mathbf{y}}{\int_{\mathcal{D}} d\mathbf{x}}, \tag{4}$$

which can be directly computed from the input images. The sign ambiguity of the determinant is also easily eliminated: A negative Jacobian would mean that the transformation is not orientation-preserving (*i.e.* flipping of coordinates is allowed). In practice, however, physical constraints will usually prevent such a transformation hence we can assume that $|\mathbf{A}|$ is always positive.

Now multiplying Eq. (2) and Eq. (3) yields a finite integral equation:

---

[1] Although we write these integrals in $\mathbb{P}^2$, they are equivalent to the corresponding Lebesgue integrals in $\mathbb{R}^2$ (*i.e.* integration is actually performed in the corresponding Cartesian coordinate system). This is because by using homogeneous coordinates, the real plane $\mathbb{R}^2$ is mapped to the $w = 1$ plane in real projective space $\mathbb{P}^2$ and affine transformations will never alter the homogeneous component w. One can therefore safely assume that it is always 1 and ignore it.

$$\int_{\mathbb{P}^2} \mathbf{x} \mathbb{1}_t(\mathbf{x}) d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathbb{P}^2} \mathbf{A}^{-1} \mathbf{y} \mathbb{1}_o(\mathbf{y}) d\mathbf{y} \quad \Leftrightarrow$$

$$\int_{\mathcal{D}} \mathbf{x} d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}} \mathbf{A}^{-1} \mathbf{y} d\mathbf{y}. \tag{5}$$

This equation implies that the finite domains $\mathcal{D}$ and $\mathcal{F}$ are also related as $\mathcal{F} = \mathbf{A}\mathcal{D}$, *i.e.* we match the shapes as a whole instead of point correspondences. In fact, Eq. (5) is a linear system of two equations for $k = 1, 2$:

$$|\mathbf{A}| \int_{\mathcal{D}} x_k d\mathbf{x} = q_{k1} \int_{\mathcal{F}} y_1 d\mathbf{y} + q_{k2} \int_{\mathcal{F}} y_2 d\mathbf{y} + q_{k3} \int_{\mathcal{F}} d\mathbf{y}.$$

It is clear that both sides of the equation as well as the Jacobian can be easily computed from the input shapes. Unfortunately, two equations alone are not enough to solve for 6 unknowns.

2.1   Construction of the Polynomial System

In order to generate more equations, let us remark that the identity relation in Eq. (1) remains valid when a function $\omega : \mathbb{P}^2 \to \mathbb{P}^2$ is acting on both sides of the equation [29]. Indeed, for a properly chosen ω

$$\omega(\mathbf{x}) = \omega(\mathbf{A}^{-1}\mathbf{y}). \tag{6}$$

We thus obtain the following integral equation from Eq. (5) and Eq. (6)

$$\int_{\mathcal{D}} \omega(\mathbf{x}) d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}} \omega(\mathbf{A}^{-1}\mathbf{y}) d\mathbf{y}. \tag{7}$$

The basic idea of the proposed approach is to generate enough linearly independent equations by making use of nonlinear ω functions. Note however, that the generated equations contain no new information, they simply impose new linearly independent constraints. Indeed, from a geometric point of view, Eq. (5) simply matches the center of mass of the *template* and *observation* while the new equations of Eq. (7) match the center of mass of the shapes obtained by the nonlinear transformations ω (see Fig. 1). Since ωs are also applied to the unknowns, the resulting equations will be nonlinear. The simplest nonlinear system is a low order polynomial system thus we aim at choosing ω such that Eq. (7) is polynomial. The following proposition states that this is achieved when ω is a polynome.

Proposition 1 *Let $\omega : \mathbb{P}^n \to \mathbb{P}^n$ and $\mathbf{x} \in \mathbb{P}^n$ ($n \in \mathbb{N}$). If the kth coordinate of $\omega(\mathbf{x})$, denoted by $\omega^k(\mathbf{x}) = p_k$ is a real n-variate polynome, $1 \leq k \leq n$, then applying ω in Eq. (7) results in a polynomial system of equations up to a degree of $\deg(p_k)$.*

$$\omega(\mathbf{x}) = \mathbf{x} \quad \omega(\mathbf{x}) = [x_1^2, x_2^2, 1]^T \qquad \omega(\mathbf{x}) = [x_1^3, x_2^3, 1]^T$$

Fig. 1. The effect of the ω functions.

PROOF. See in the Appendix.

It is thus clear that the class of $x^n (n \in \mathbb{N}_0)$ functions are a perfect choice for ω. Hence, we obtain the following polynomial equations for $k = 1, 2$ and $n = 1, 2, 3$:

$$|\mathbf{A}| \int x_k^n = \sum_{i=1}^{n} \binom{n}{i} \sum_{j=0}^{i} \binom{i}{j} q_{k1}^{n-i} q_{k2}^{i-j} q_{k3}^{j} \int y_1^{n-i} y_2^{i-j}. \tag{8}$$

The system of Eq. (8) contains six polynomial equations up to order three which is enough to solve for all unknowns. In fact we have two separate systems for $k = 1, 2$ as shown in Eq. (9)–(11).

$$|\mathbf{A}| \int x_k = q_{k1} \int y_1 + q_{k2} \int y_2 + q_{k3} \int 1, \tag{9}$$

$$|\mathbf{A}| \int x_k^2 = q_{k1}^2 \int y_1^2 + q_{k2}^2 \int y_2^2 + q_{k3}^2 \int 1 + 2q_{k1}q_{k2} \int y_1 y_2 + 2q_{k1}q_{k3} \int y_1$$

$$+ 2q_{k2}q_{k3} \int y_2, \tag{10}$$

$$|\mathbf{A}| \int x_k^3 = q_{k1}^3 \int y_1^3 + q_{k2}^3 \int y_2^3 + q_{k3}^3 \int 1 + 3q_{k1}^2 q_{k2} \int y_1^2 y_2 + 3q_{k1}^2 q_{k3} \int y_1^2$$

$$+ 3q_{k2}^2 q_{k3} \int y_2^2 + 3q_{k1}q_{k2}^2 \int y_1 y_2^2 + 3q_{k2}q_{k3}^2 \int y_2 + 3q_{k1}q_{k3}^2 \int y_1$$

$$+ 6q_{k1}q_{k2}q_{k3} \int y_1 y_2. \tag{11}$$

However, we may get several possible solutions for each unknown q_{ki} due to the cubic polynomial equations. Out of these potential solutions, we can select the right one by dropping the complex roots and selecting the transformation whose determinant matches the Jacobian computed by Eq. (4).

Note that an exact solution always exists, whenever Eq. (3) is satisfied. In practice, however, a solution may not exists due to discretization errors or noise on the point coordinates. We can always check for the existence of a solution by computing the resultant of the system, which is a second order polynome. On the other hand, the solution is not unique (but exists!), when the shape is affine symmetric.

We remark that this method could also be extended to higher dimensions at the price of adding higher order polynoms or restricting the space of admissible transformations. For example, in many medical applications 3-dimensional volume images are used. In this case, we should add a fourth-order equation and solve three systems. This is obviously more difficult as well as the higher powers may compromise numerical stability. Another idea is to use ωs of mixed coordinates (*e.g.* $\omega(\mathbf{x}) = [x_1 x_2, x_2, 1]^T$). This will generate new equations without increasing their order, but we loose the benefit of handling the coordinates separately, *i.e.* we have to solve one large polynomial system instead of three smaller ones.

2.2 Numerical Implementation

We have constructed our equations in the continuum but in practice we only have a limited precision digital image. This means that the integrals, which are in fact the coefficients of the unknowns in Eq. (9)–(11), can only be *approximated* by a discrete sum over the foreground pixels introducing an inherent, although negligible error into our computation. In [30,31], the effect of such errors on moment computation has been analysed and a number-theoretical estimation of quantization errors is derived. The continuous domains $\mathcal{D}$ and $\mathcal{F}$ are represented as finite sets of foreground pixels

$$\mathcal{D} \approx D = \{\mathbf{d}^i\}_{i=1}^n \quad \text{and} \quad \mathcal{F} \approx F = \{\mathbf{f}^i\}_{i=1}^m,$$

where n and m are the number of foreground pixels of the *template* and *observation* respectively. As a consequence, the integrals are *approximated* by a finite sum over these foreground pixels

$$\int_{\mathcal{D}} x_k d\mathbf{x} \approx \sum_{i=1}^n \mathbf{d}_k^i \quad \text{and} \quad \int_{\mathcal{F}} y_k d\mathbf{y} \approx \sum_{i=1}^m \mathbf{f}_k^i, \quad k = 1, 2$$

where $\mathbf{d}_k^i$ and $\mathbf{f}_k^i$ denote the kth coordinate of the ith foreground pixel. The Jacobian in Eq. (4) reduces to

$$|\mathbf{A}| = \frac{m}{n}. \tag{12}$$

Clearly, the resolution of the images affects the precision of these approximations. As the mesh size tends to zero, the finite sums approximate better the integrals. Therefore, our method performs better on higher resolution images. Experiments show that images with size 500×500 already provide accurate approximations.

Algorithm 1 summarizes the steps of our registration method. It is clear that the solution is obtained in a single pass without any loop or optimization. Although, we have to solve a polynomial system, the complexity of this step is constant and, most importantly, independent of the image size. Matlab is quite efficient in solving our system but other packages are also available, like *PHCpack* [32–34].

Algorithm 1 Pseudo-code of the proposed algorithm.

Input: *template* and *observation* shapes as binary images
Output: Estimated affine transformation $\widehat{\mathbf{A}}$
  1: Estimating the Jacobian $\mathbf{A}$ using Eq. (12)
  2: Evaluating the integrals in Eq. (9)–(11) provides the coefficients of the unknowns.
  3: Solving the system of Eq. (9)–(11) using a standard solver (*e.g.* Matlab).
  4: Choosing the correct transformation based on the Jacobian gives $\widehat{\mathbf{A}}^{-1}$

The images need to be scanned only once, and the integrals in Eq. (9)–(11) as well as the Jacobian can be evaluated during this scan. This step takes $c_1 N$ time, where N is the size of the input images. Once the system is constructed, the rest of the algorithm runs in constant (c_2) time independently of the input size. Thus the overall time complexity of the method is $c_1 N + c_2$, *i.e.* $\mathcal{O}(N)$. Nevertheless, our experiments show that $c_1 N << c_2$, hence the actual running time is dominated by the constant c_2 (see Table 3).

3 Estimation in the Presence of Noise

There are two types of noise which can affect a binary image. One is "radiometric", *i.e.* pixels may randomly take a *foreground* or *background* color. This is typically present in the form of a *salt and pepper* noise, which can be efficiently removed by appropriate morphological filtering. The second type manifests as an uncertainty in the point coordinates, *i.e.* it is a *geometric* noise. More precisely, the observed point coordinates may differ by a random distance from the true coordinates as shown in Fig. 2. This will definitely undermine the identity relation in Eq. (1) yielding an error in our system. In the following section, we will analyze this case and compute the average error caused by a *geometric* noise on the *observation*.

(a) Original   (b) $\sigma = 5$ (c) $\sigma = 10$ (d) $\sigma = 20$
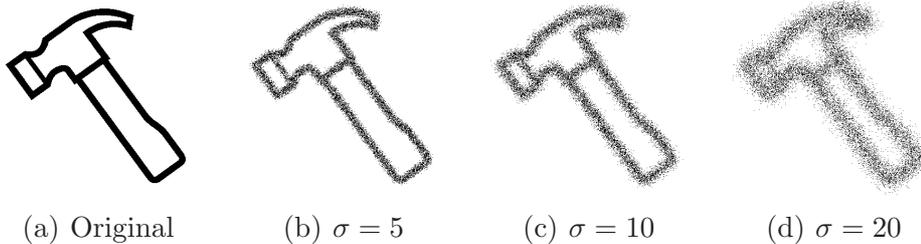
Fig. 2. The noise tolerance of the proposed method has been tested on observations corrupted by additive Gaussian noise on the coordinates.

3.1 Geometric Noise

It is a realistic assumption that the observed point coordinates are around the true ones, hence we will consider an i.i.d. additive Gaussian noise model on the *observation* coordinates. The identity relation Eq. (1) thus becomes

$$\mathbf{y}^* = \mathbf{y} + \varepsilon(\mathbf{y}) = \mathbf{A}\mathbf{x} + \varepsilon^*(\mathbf{y}^*) \Leftrightarrow \mathbf{x} = \mathbf{A}^{-1}\Big(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)\Big),$$

where $\varepsilon(\mathbf{y}) \equiv \varepsilon^*(\mathbf{y}^*) = \left[\varepsilon_1^*(\mathbf{y}^*), \varepsilon_2^*(\mathbf{y}^*), 0\right]^T$ is the noise function which gives a random translation in every point $\mathbf{y}^* = [y_1^*, y_2^*, 1]^T$. We assume that ε_1^* and ε_2^* are independent and normally distributed with 0 means and variances σ_1 and σ_2 respectively. In fact, $\varepsilon^*(\mathbf{y}^*)$ is a sample form a two dimensional, 0 mean Gaussian distribution with a diagonal covariance matrix $\mathrm{diag}(\sigma_1, \sigma_2)$. Thus Eq. (7) becomes

$$\int_{\mathcal{D}} \omega(\mathbf{x})d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}^*} \omega\Big(\mathbf{A}^{-1}\Big(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)\Big)\Big) J(\mathbf{y}^*)d\mathbf{y}^*,$$

where the integral transformation $\mathbf{x} \mapsto \mathbf{A}^{-1}\Big(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)\Big)$, $d\mathbf{x} \mapsto J(\mathbf{y}^*)d\mathbf{y}^*/|\mathbf{A}|$ has been applied. Hereafter, we will omit the integration domains unless they are ambiguous. In fact, the Jacobian becomes $J(\mathbf{y}^*)/|\mathbf{A}|$ in the noisy case with $J(\mathbf{y}^*) = \Big(1 - \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} - \varepsilon_2^*(\mathbf{y}^*)'_{y_2^*} + |\nabla \varepsilon^*(\mathbf{y}^*)|\Big)$ depending on the actual noise. $\varepsilon_1^*(\mathbf{y}^*)'_{y_2^*}$ denotes the partial derivate based on the second variable (y_2^*) and

$$|\nabla \varepsilon^*(\mathbf{y}^*)| = \begin{vmatrix} \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} & \varepsilon_1^*(\mathbf{y}^*)'_{y_2^*} \\ \varepsilon_2^*(\mathbf{y}^*)'_{y_1^*} & \varepsilon_2^*(\mathbf{y}^*)'_{y_2^*} \end{vmatrix}.$$

It is clear that the integrals of ε_i^* give the expected value, which is always 0 in our case. Furthermore, since in practice we always work with discrete pixel coordinates, the partial derivatives of ε^* can be approximated via finite differences, *e.g.* by central differences with $h \in \mathbb{N}$

10

$$\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} \approx \frac{1}{2h}\Big(\varepsilon^*\big([y_1^* + h, y_2^*, 1]^T\big) - \varepsilon^*\big([y_1^* - h, y_2^*, 1]^T\big)\Big),$$

$$\varepsilon_1^*(\mathbf{y}^*)'_{y_2^*} \approx \frac{1}{2h}\Big(\varepsilon^*\big([y_1^*, y_2^* + h, 1]^T\big) - \varepsilon^*\big([y_1^*, y_2^* - h, 1]^T\big)\Big).$$

Therefore the integral of these derivatives will also be approximated by the integral of the finite differences, which thus evaluates to 0.

3.1.1   Computation of $|\mathbf{A}|$

Since the true coordinates $\mathbf{y}$ are unknown on a noisy *observation*, we have to integrate using $\mathbf{y} = \mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)$. Thus the numerator of Eq. (4) becomes

$$\int d\mathbf{y} = \int \Big(1 - |\nabla\varepsilon^*(\mathbf{y}^*)|\Big)d\mathbf{y}^* = \int d\mathbf{y}^*,$$

because $\int |\nabla\varepsilon^*(\mathbf{y}^*)|d\mathbf{y}^* = 0$ according to our previous deductions. Therefore Eq. (4) remains valid in the noisy case.

3.1.2 $\omega(\mathbf{x}) = \mathbf{x}$

The right hand side of Eq. (9) has to be evaluated on the noisy *observation*. Thus denoting $K := \mathbf{A}^{-1}\big(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)\big)$, i.e. $K_k = q_{k1}(y_1^* - \varepsilon_1^*(\mathbf{y}^*)) + q_{k2}(y_2^* - \varepsilon_2^*(\mathbf{y}^*)) + q_{k3}$, we get for $k = 1, 2$

$$\int \mathbf{A}_k^{-1}\mathbf{y}\,d\mathbf{y} = \int K_k d\mathbf{y}^* - \int K_k \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* - \int K_k \varepsilon_2^*(\mathbf{y}^*)'_{y_2^*} d\mathbf{y}^*$$
$$+ \int K_k |\nabla\varepsilon^*(\mathbf{y}^*)| d\mathbf{y}^*.$$

It is easy to see that all terms, except the first one, evaluates to 0. For example

$$\int K_k \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* = \int \Big(q_{k1}\big(y_1^* - \varepsilon_1^*(\mathbf{y}^*)\big) + q_{k2}\big(y_2^* - \varepsilon_2^*(\mathbf{y}^*)\big) + q_{k3}\Big)\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^*.$$

Evaluating the coefficient of q_{k1} yields

$$\int \big(y_1^* - \varepsilon_1^*(\mathbf{y}^*)\big)\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* = \int y_1^*\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* - \int \varepsilon_1^*(\mathbf{y}^*)\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^*.$$

Using finite differences for $\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*}$ and by iterated integrals, we get for the first term

$$\frac{1}{2h}\int y_1^*\left(\int \varepsilon_1^*\big([y_1^* + h, y_2^*, 1]^T\big)dy_2^*\right)dy_1^* - \frac{1}{2h}\int y_1^*\left(\int \varepsilon_1^*\big([y_1^* - h, y_2^*, 1]^T\big)dy_2^*\right)dy_1^* =$$

$$\frac{1}{2h}\int y_1^* 0\,dy_1^* - \frac{1}{2h}\int y_1^* 0\,dy_1^* = 0,$$

since ε_1 is a 0 mean Gaussian for which $\int \varepsilon_1 = 0$. The remaining terms can be evaluated in a similar way. Finally, the right hand side of Eq. (9) in the case of a noisy *observation* is as follows:

$$\int K_k d\mathbf{y}^* = q_{k1} \int y_1^* d\mathbf{y}^* + q_{k2} \int y_2^* d\mathbf{y}^* + q_{k3} \int d\mathbf{y}^*.$$

Hence Eq. (9) remains valid. This is not surprising as the equation matches the center of mass of the *template* and *observation* shapes. When the *observation* coordinates are corrupted by a 0 mean additive noise, the center of mass will not change as the noise components integrate to 0.

3.1.3 $\omega(\mathbf{x}) = [x_1^2, x_2^2, 1]^T$

Using the noisy *observation*, Eq. (10) becomes

$$|\mathbf{A}| \int x_k^2 d\mathbf{x} = \int \left(\mathbf{A}_k^{-1} \big(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*) \big) \right)^2 d\mathbf{y}^*.$$

It is clear that both first and second order noise statistics will appear in the above equation. While first order statistics are vanishing, second-order moments ($\int \varepsilon_k^*(\mathbf{y}^*)^2 = \sigma_k^2, k = 1, 2$) will affect the equation. It is straightforward to show, using similar considerations as in Section 3.1.2, that in the noisy case Eq. (10) becomes

$$|\mathbf{A}| \int x_k^2 d\mathbf{x} = \int \left(\mathbf{A}_k^{-1}(\mathbf{y}) \right)^2 d\mathbf{y} + q_{k1}^2 \sigma_1^2 + q_{k2}^2 \sigma_2^2.$$

Thus the error introduced by the noisy *observation* depends on the noise variances σ_1^2 and σ_2^2, as well as on the strength of the unknown transformation $\mathbf{A}$.

3.1.4 $\omega(\mathbf{x}) = [x_1^3, x_2^3, 1]^T$

In this case, we will have third order noise statistics involved. Fortunately, any symmetric distribution will have a third central moment of zero, thus using again similar considerations as before, the noisy Eq. (11) is as follows

$$|\mathbf{A}| \int x_k^3 d\mathbf{x} = \int \left(\mathbf{A}_k^{-1}(\mathbf{y}) \right)^3 d\mathbf{y} + 3q_{k1}^2 q_{k3} \sigma_1^2 + 3q_{k2}^2 q_{k3} \sigma_2^2.$$

3.1.5   Summary

In summary, the error caused by an i.i.d. additive Gaussian noise on the point coordinates of the *observation* is as follows:

Templ.	Obs.	Heikkilä [11]	SC [7]	Kannala et al. [10]	Suk et al. [12]	Prop.
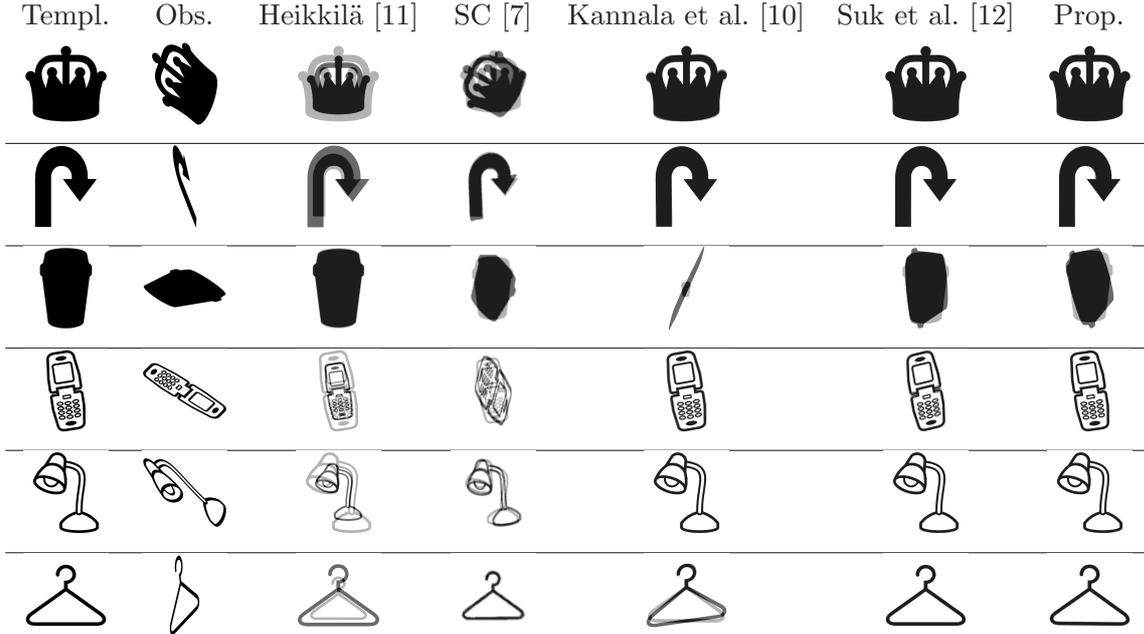
Fig. 3. Registration results on synthetic image pairs. The first two columns show the *template* and its affine distorted *observation* to be matched while the other columns contain the registration result of each considered method. The *template* and its *registered observation* are overlayed such that overlapping pixels are depicted in black while non-overlapping ones are shown in light or dark gray respectively.

Equation	Error term
$|\mathbf{A}|$: Eq. (4)	0
$\omega(\mathbf{x}) = \mathbf{x}$: Eq. (9)	0
$\omega(\mathbf{x}) = [x_1^2, x_2^2, 1]^T$: Eq. (10)	$q_{k1}^2 \sigma_1^2 + q_{k2}^2 \sigma_2^2$
$\omega(\mathbf{x}) = [x_1^3, x_2^3, 1]^T$: Eq. (11)	$3q_{k3}(q_{k1}^2 \sigma_1^2 + q_{k2}^2 \sigma_2^2)$

An experimental analysis of the noisy case is presented in Section 4.2. Our findings suggest that the proposed algorithm can cope with noisy observations up to as high as $\sigma_1 = \sigma_2 = 10$ noise levels.

4 Experimental Results

The proposed algorithm has been tested on a large database of binary images of size 1000×1000. The dataset consists of 56 different shapes and their transformed versions, a total of ≈ 50000 images. The applied affine transformations were randomly composed of $0°, 10°, \ldots, 350°$ rotations; $0, 0.4, \ldots, 1.2$ shearings; $0.5, 0.7, \ldots, 1.9$ scalings, and $-20, 0, 20$ translations along both axes.

Table 1
Registration results of the proposed method on the benchmark dataset containing 49282 synthetic observations of 56 shapes. There was no solution in 5.47% of the test cases.

	Runtime (sec.)	ϵ (pixel)	δ (%)
Median	0.98	0.51	0.15
Mean	0.94	36.98	3.36
Variance	0.2	154.18	12.55

Table 2
Registration results (medians of error measures) of the proposed method on 32724 filled shapes and 16558 line drawings.

	Runtime (sec.)	ϵ (pixel)	δ (%)	Unsolved (%)
Filled	1	0.49	0.06	4.03
Line drawings	0.95	0.55	0.63	8.32

Some typical examples of these images can be seen in Fig. 3. The original shapes were then used as *template* and the transformed images as the *observation*. The proposed algorithm has been implemented in Matlab 7.2 and ran on a Pentium IV 3.2 GHz under Linux operating system. The average runtime was around 1 second including the computation of the integrals and the solution of the polynomial system. The dataset and a demo implementation of our method are available for download at `http://www.inf.u-szeged.hu/~kato/software/`.

In order to quantitatively evaluate registration results, we defined two kind of error measures. The first one (denoted by ϵ) measures the distance between the true $\mathbf{A}$ and the estimated $\widehat{\mathbf{A}}$ transformation obtained by our algorithm, based on all *template pixels* $\mathbf{p}$. Intuitively, ϵ shows the average transformation error per pixel. Another measure is the absolute difference (denoted by δ) between the *observation* and the *registered* image.

$$\epsilon = \frac{1}{|D|} \sum_{\mathbf{p} \in D} \|(\mathbf{A} - \widehat{\mathbf{A}})\mathbf{p}\|, \quad \text{and} \quad \delta = \frac{|R \triangle O|}{|R| + |O|} \cdot 100\%,$$

where D is the set of *template* pixels, $\triangle$ means the symmetric difference, while R and O denote the set of pixels of the *registered* shape and *observation* respectively. Note that ϵ can only be used when the true transformation $\mathbf{A}$ is also known, while δ can always be computed. On the other hand, ϵ gives a better characterization of the transformation error as it directly evaluates the mistransformation. δ sees only the percentage of non-overlapping area between the *observation* and *registered* shape. Hence the value of δ depends also on the compactness and topology of the shapes. The performance of our algorithm on the benchmark dataset has been evaluated based on these measures. A summary of these results is presented in Table 1 Another important indicator

Table 3
Median of error measures versus resolution of the observation.

Size (%)	Runtime (sec.)	ϵ (pixel)	δ (%)
100	1.04	0.71	0.09
50	0.9	14.68	0.74
25	0.85	21.54	0.81
12.5	0.81	25.39	4.91

is the number of test cases where the algorithm finds no solution. Although all the applied transformations were affine, it is possible that an *observation* is not on the orbit of its *template*. The reason is mainly due to an excessive amount of numerical error in the coefficients caused by a combination of strong deformation and discretization error. The compactness of the shapes also affects the amount of discretization error. Obviously, such errors arise only around the boundaries thus shapes with a longer contour will produce higher discretization error. Our database contains 32724 filled shapes and 16558 line drawings. Table 2 shows a comparison of the registration quality on these shapes. It is clear that registration of line drawings (*i.e.* shapes with longer contours) is slightly less accurate. Finally, the resolution of the images is also important. As we already mentioned in Section 2.2, the discrete sums will approximate better the integrals at higher resolutions. Fortunately, the time complexity of our method is linear hence increasing the resolution will not deteriorate computing time. Table 3 shows the error and CPU time in function of the resolution.

4.1 Comparison to Previous Approaches

Herein, we review some of the most relevant binary registration approaches and, where an implementation was available, evaluate quantitatively the performance of our algorithm with respect to these methods. For that purpose, we have used 1686 randomly chosen images from our database. The results are presented in Table 4 and Fig. 3.

Flusser et al.propose an image registration algorithm based on affine moments in [13]. 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.

Table 4

Median of error measures on 1686 randomly selected images using the method of Heikkilä [11], Shape context [7], the method of Kannala et al. [10], Suk et al. [12] and the proposed algorithm.

	Runtime (sec.)	ϵ (pixel)	δ (%)
Heikkilä [11]	1.15	86.35	39.03
Shape context [7]	24.79	–	27.17
Kannala et al. [10]	32.45	8.89	9.7
Suk et al. [12]	5.62	0.51	0.19
Proposed	0.93	0.5	0.15

Belongie et al.proposed a novel approach for shape matching in [7]. The method first searches for point correspondences between the two objects, then estimates the transformation using these correspondences. The point matches are established using a novel similarity metric, called *shape context*, which consists in constructing a log-polar histogram of surrounding edge pixels. The advantage compared to traditional landmark based approaches is that landmarks need not be salient points nor radiometric information is involved. Basically the method can be regarded as matching two points sets, each of them being a dense sample from the corresponding shape's boundary. Obviously, there is no guarantee that point pairs are exactly corresponding because of the sampling procedure. However, having a dense sample will certainly keep mismatch error at a minimum. The correspondences are simply established by solving a linear assignment problem, which requires time consuming optimization methods. For example, the complexity of the Hungarian method adopted in [7] is $\mathcal{O}(N^3)$. We ran the demo software [35] provided by the authors on our dataset. Although the method uses regularized thin-plate splines as a flexible class of transformations, it is possible to set the regularization parameter to get affine behavior: *beta_init=500*. Other parameters were also set empirically to their optimal value (number of iteration *n_iter=15*; annealing rate *r=5*). Due to its high complexity, we had to reduce the size of the images by a factor of 0.2. Nevertheless, as can be seen in Table 4, CPU time was still over 20 seconds.

Probably the most closely related approach is the binary registration algorithm proposed by Kannala et al. [10,25]. The fundamental difference is that [10] constructs a system of equations by basically 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 ω functions hence we always use all the information available in the images.

16

Table 5
Median of error measures versus σ of the noise on 1377 randomly selected images.

σ	1	2	5	10	15	20
ϵ (pixel)	0.51	0.53	0.73	2.42	5.86	11.91
δ (%)	0.19	0.27	0.79	2.72	6.24	10.64

Another class of related methods consists in object matching based on image moments [12,11]. Suk and Flusser [12] construct affine normalized images by making use of image moments. An affine transformation is decomposed into basic transformations, and then they are successively eliminated by central and complex moments. The aligning transformation of two objects is then obtained by affine normalizing both images. This approach works well on synthetic as well as on real images. We remark that the method in [12] allows mirroring too, which is excluded in our model. In [11], Heikkilä constructs affine descriptors using higher order moments and moment invariants. Similar to the previous method, the transformation parameters are eliminated one by one. However, in many cases this may result in increased registration error as erroneous parameter values are fixed and propagated towards the computation of subsequent parameters. In contrast, our approach solves for *all* parameters hence error is better distributed over the estimated parameters. We have obtained the Matlab implementation from the authors of [10], [12], and [11] and conducted a comparative test. The results presented in Table 4 and Fig. 3 show that our method outperforms these approaches in terms of both quality and computing time. In the case of [12], the registration quality is almost the same thus it is fair to say that both methods give accurate registrations but our algorithm runs faster. On the other hand, our method clearly dominates [12] in terms of robustness as it is demonstrated in the next section.

4.2   Robustness

In Section 3, we derived the errors caused by noisy observations in our polynomial system Eq. (9)–(11). Herein, we will experimentally test the robustness of the proposed method against i.i.d. Gaussian geometric noise. For that purpose, we have used 1377 randomly selected images from our benchmark database. An i.i.d. Gaussian noise with $\sigma = 1, 2, 5, 10, 15, 20$ was added to the extracted pixel coordinates of each *observation* and the registration algorithm got as input the point list of this noisy *observation* and the original *template*. To evaluate registration quality, the recovered transformation has been applied to the original (*i.e.* noiseless) *observation* and the usual error measures ϵ and δ have been computed. It is clear from Table 5 that the proposed algorithm provides good solutions up to us high as $\sigma = 10$ noise levels.

(a) Original   (b) $\delta = 2.03\%$ (c) $\delta = 8.2\%$ (d) $\delta = 17.12\%$
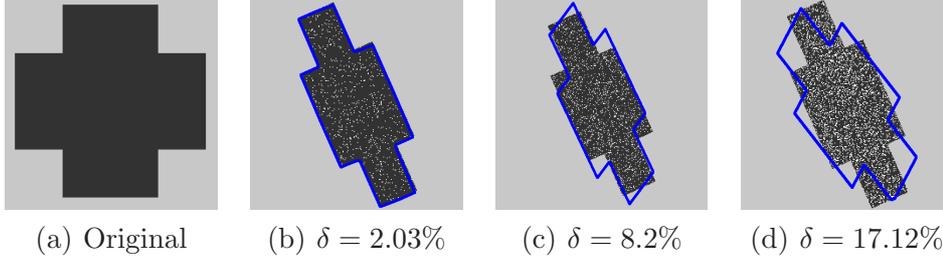
Fig. 4. The robustness of the proposed approach in case of incomplete objects has been evaluated on images where we have randomly removed 5%, 10%, and 20% of the foreground pixels ((b)-(d)) of the original image (a). The registration results are shown as overlayed contours of the *registered* shape.

We also considered the robustness of the proposed approach in case of incomplete objects. For that purpose, we have used 1289 randomly selected images from our benchmark database where we have randomly removed 5%, 10%, 15%, and 20% of the foreground pixels before registration (see Fig. 4). Clearly, such incomplete observations will cause errors in the original system of equations Eq. (9)–(11). Table 6 shows that our method is quite robust while the error rate of other state of the art methods considerably increases even for as low as 5% missing pixels. However, we have to note, that all of these methods are less robust against the same amount of occlusion (*i.e.* when missing pixels are not uniformly distributed over the whole region). This is presented in Table 7, where we show the results on 794 randomly selected images with occlusions of size 2.5%, 5% and 10% of the input shape's area. Clearly, even relatively small occlusions yield a rather high error rate for both the proposed as well other state of the art methods. This is because they are relying on quantities obtained by integrating over the whole object area. Thus large missing parts would drastically change these quantities resulting in false registrations. Nevertheless, in many application areas one can take images under controlled conditions which guarantees that observations are not occluded (*e.g.* medical imaging, industrial inspection).

4.3   Real Images

The performance of our method has also been evaluated on real images. Fig. 5 shows some examples of these images with overlayed contours. For segmentation, we have used classical thresholding as well as active contours [36]. The main challenges are the segmentation errors (*e.g.* see the fifth image in Fig. 5) and slight projective distortion between the image pairs. In summary, when reasonably good segmentations are available and the true transformation is close enough to an affine one then our method performs quite well, as it is shown by the δ error values and difference images displayed below each image pairs.

18

Table 6
Median of error measures versus the ratio of removed foreground pixels on 1289 randomly selected images.

Measure	Method	5%	10%	15%	20%
ϵ (pixel)	Heikkilä [11]	76.29	75	71.86	70.38
	Kannala et al.	8.69	12.07	13.55	14.42
	S. & F. [12]	8.09	16.12	24.73	33.09
	Proposed	1.95	3.56	9.25	8.86
δ (%)	Heikkilä [11]	40.5	40.64	41.05	39.63
	Kannala et al.	9.42	10.82	12.54	14.41
	S. & F. [12]	4.44	8.78	12.98	17.7
	Proposed	1.93	3.78	10.96	10.9

Table 7
Median of error measures versus the size of occlusion on 794 randomly selected images.

Measure	Method	2.5%	5%	10%
ϵ (pixel)	Heikkilä [11]	120.36	153.38	215.81
	Kannala et al.	76.49	137.34	227.63
	S. & F. [12]	42.61	91.88	175.32
	Proposed	47.44	162.34	251.4
δ (%)	Heikkilä [11]	53.77	55.65	60.98
	Kannala et al.	55.77	68.68	79.75
	S. & F. [12]	23.91	37.16	47.65
	Proposed	38.12	51.1	58.67

4.4   Registration of Hip Prosthesis X-ray Images

Hip replacement [17,37] is a surgical procedure in which the hip joint is replaced by a prosthetic implant. In the short term post-operatively, infection is a major concern. An inflammatory process causes bone resorption and subsequent loosening or fracture often requiring revision surgery. In current practice, clinicians assess loosening by inspecting a number of post-operative X-ray images of the patient's hip joint, taken over a period of time. Obviously, such an analysis requires the registration of X-ray images as shown in Fig. 6. Even visual inspection can benefit from registration as clinically significant prosthesis movement can be very small [17,37].

There are two main challenges in registering hip X-ray images: One is the highly non-linear radiometric distortion [18] which makes any graylevel-based method unstable. Fortunately, the segmentation of the prosthetic implant is quite straightforward [38] so binary registration is a valid alternative here. Herein, we used active contours [36] to segment the implant. Let us remark that our binary registration method is not directly affected by illumination changes as there is no radiometric information available to the algorithm. On the other hand, illumination variations may cause segmentation errors which affects our method (see Section 4.2). However, such errors arise only in extreme cases as modern segmentation algorithms are quite robust to illumination variations. The second problem is that the true transformation is a projective one which depends also on the position of the implant in 3D space. Indeed, there is a rigid-body transformation in 3D space between the implants, which becomes a projective mapping between the X-ray images. Fortunately, the affine assumption is a good approximation here as the X-ray images are taken in a well defined *standard position* of the patient's leg. Some registration results are presented in Fig. 6.

5 Conclusions

In this paper, we have presented a novel approach for planar shape alignment. The fundamental difference compared to classical image registration algorithms is that our model works without any landmark, feature detection or optimization by adopting a novel idea where the transformation is obtained as a solution of a set of polynomial equations. It uses all the information available in the input images, but there is no need for established correspondences. Our algorithm is simple to implement and runs quite fast almost independently of the image size. Experimental results show that the proposed method provides good alignment on both real and synthetic images. Furthermore, it is robust in the case of noisy observations. Comparative tests show the efficiency and accuracy of our model compared to state of the art methods.

A Proof of Proposition 1

Let $1 \leq k \leq n$ arbitrary and fixed. We assume that $\omega^k(\mathbf{x})$ is polynomial, *i.e.* there exists an n-variate real polynome p_k with $\deg(p_k) \geq 1$, such that

$$\omega^k(\mathbf{x}) = p_k(x_1, \ldots, x_n) = \sum_{i=1}^{u_k} s_i (\mathbf{A}_1^{-1}\mathbf{y})^{\alpha_{i1}} \ldots (\mathbf{A}_n^{-1}\mathbf{y})^{\alpha_{in}}, \qquad (A.1)$$

20

where $u_k = \binom{\deg(p_k)+n}{\deg(p_k)}$, and $\mathbf{A}_j^{-1}$ denotes the jth row of $\mathbf{A}^{-1}$. One term of Eq. (A.1) can be expanded by making use of the *Multinomial theorem* [39]. For a given i and for all $1 \leq j \leq n$, we get

$$(\mathbf{A}_j^{-1}\mathbf{y})^{\alpha_{ij}} = (q_{j1}y_1 + \cdots + q_{jn}y_n + q_{j(n+1)})^{\alpha_{ij}}$$

$$= \sum_{\substack{\beta_{ij1},\ldots,\beta_{ij(n+1)} \in \mathbb{N}_0 \\ \beta_{ij1} + \cdots + \beta_{ij(n+1)} = \alpha_{ij}}} \frac{\alpha_{ij}!}{\beta_{ij1}!\ldots\beta_{ij(n+1)}!} q_{j1}^{\beta_{ij1}} \ldots q_{jn}^{\beta_{ijn}} q_{j(n+1)}^{\beta_{ij(n+1)}} y_1^{\beta_{ij1}} \ldots y_n^{\beta_{ijn}},$$

hence we get an $(n+1)$-variate real polynome. In fact, we should compute the sum of the product of n pieces of $(n+1)$ - variate polynoms in Eq. (A.1). Let $m = n(n+1)$ and consider these products as m-variate polynoms. Furthermore, the sum of m-variate polynoms is also an m-variate polynome. Integrating and using this observation we can rewrite Eq. (A.1) as

$$\int \sum_{i=1}^{u_k} s_i (\mathbf{A}_1^{-1}\mathbf{y})^{\alpha_{i1}} \ldots (\mathbf{A}_n^{-1}\mathbf{y})^{\alpha_{in}} \equiv \int \sum_{i=1}^{v_k} t_i q_1^{\gamma_{i1}} \ldots q_m^{\gamma_{im}} y_1^{\delta_{i1}} \ldots y_n^{\delta_{in}},$$

where $v_k = \binom{\deg(p_k)+m}{\deg(p_k)}$. It is obvious from the above equation that the system of equation has a degree of up to $\deg(p_k)$. Furthermore, by making use of the basic properties of the Lebesgue integral, we get

$$\int \sum_{i=1}^{v_k} t_i q_1^{\gamma_{i1}} \ldots q_m^{\gamma_{im}} y_1^{\delta_{i1}} \ldots y_n^{\delta_{in}} = \sum_{i=1}^{v_k} \int t_i q_1^{\gamma_{i1}} \ldots q_m^{\gamma_{im}} y_1^{\delta_{i1}} \ldots y_n^{\delta_{in}}$$

$$= \sum_{i=1}^{v_k} t_i q_1^{\gamma_{i1}} \ldots q_m^{\gamma_{im}} \int y_1^{\delta_{i1}} \ldots y_n^{\delta_{in}} = \sum_{i=1}^{v_k} w_i q_1^{\gamma_{i1}} \ldots q_m^{\gamma_{im}}.$$

The last term is indeed a real polynome r_k with variables $q_1, \ldots, q_m$ yielding

$$\int \omega^k(\mathbf{x}) \equiv r_k(q_1, \ldots, q_m).$$

Hence the system of equations is polynomial which completes the proof.

Acknowledgements

References

[1] B. Zitová, J. Flusser, Image registration methods: A survey, Image and Vision Computing 21 (11) (2003) 977–1000.

[2] L. G. Brown, A survey of image registration techniques, ACM Computing Surveys 24 (4) (1992) 325–376.

[3] J. Salvi, C. Matabosch, D. Fofi, J. Forest, A review of recent range image registration methods with accuracy evaluation, Image and Vision Computing 25 (5) (2007) 578–596.

[4] J. B. A. Maintz, M. A. Viergever, A survey of medical image registration, Medical Image Analysis 2 (1) (1998) 1–36.

[5] D. L. G. Hill, P. G. Batchelor, M. Holden, D. J. Hawkes, Medical image registration, Physics in Medicine and Biology 46 (3) (2001) R1–R45.

[6] R. A. McLaughlin, J. Hipwell, D. J. Hawkes, J. A. Noble, J. V. Byrne, T. Cox, A comparison of 2D-3D intensity-based registration and feature-based registration for neurointerventions, in: Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention, Vol. 2489 of Lecture Notes in Computer Science, Springer, Tokyo, Japan, 2002, pp. 517–524.

[7] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape context, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 509–522.

[8] A. W. Fitzgibbon, Robust registration of 2D and 3D point sets, Image and Vision Computing 21 (13) (2003) 1145–1153.

[9] R. Hagege, J. M. Francos, Parametric estimation of multi-dimensional affine transformations:an exact linear solution, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, IEEE, Philadelphia, USA, 2005, pp. 861–864.

[10] J. Kannala, E. Rahtu, J. Heikkilä, M. Salo, A new method for affine registration of images and point sets, in: Proceedings of Scandinavian Conference on Image Analysis, Vol. 3540 of Lecture Notes in Computer Science, Pattern Recognition Society of Finland, Springer, Joensuu, Finland, 2005, pp. 224–234.

[11] J. Heikkilä, Pattern matching with affine moment descriptors, Pattern Recognition 37 (9) (2004) 1825–1834.

[12] T. Suk, J. Flusser, Affine normalization of symmetric objects, in: J. Blanc-Talon, W. Philips, D. Popescu, P. Scheunders (Eds.), Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems, Vol. 3708 of Lecture Notes in Computer Science, Springer, Antwerp, Belgium, 2005, pp. 100–107.

[13] J. Flusser, T. Suk, A moment-based approach to registration of images with affine geometric distortion, IEEE Transactions on Geoscience and Remote Sensing 32 (2) (1994) 382–387.

[14] T. M. Lehmann, A two-stage algorithm for model-based registration of medical images, in: Proceedings of International Conference on Pattern Recognition, Vol. 1, IEEE, Brisbane, Australia, 1998, pp. 344–351.

[15] J. Kim, J. A. Fessler, Intensity-based image registration using robust correlation coefficients, IEEE Transactions on Medical Imaging 23 (11) (2004) 1430–1444.

[16] K. M. Simonson, S. M. Drescher, F. R. Tanner, A statistics-based approach to binary image registration with uncertainty analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 112–125.

[17] M. Downing, P. Undrill, P. Ashcroft, D. Hukins, J. Hutchison, Automated femoral measurement in total hip replacement radiographs, in: Proceedings of International Conference on Image Processing and Its Applications, Vol. 2, IEEE, Dublin, Ireland, 1997, pp. 843–847.

[18] C. Florea, C. Vertan, L. Florea, Logarithmic model-based dynamic range enhancement of hip x-ray images, in: Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems, Vol. 4678 of Lecture Notes in Computer Science, Springer, Delft, Netherlands, 2007, pp. 587–596.

[19] Y. Zhang, C. Wen, Y. Zhang, Recognition of symmetrical images using affine moment invariants in both frequency and spatial domains, Pattern Analysis & Applications 5 (3) (2002) 316–325.

[20] M. P. Kumar, S. Kuthirummal, C. V. Jawahar, P. J. Narayanan, Planar homography from fourier domain representation, in: Proceedings of International Conference on Signal Processing and Communications, IEEE, Bangalore, India, 2004, pp. 560–564.

[21] A. Kadyrov, M. Petrou, Affine parameter estimation from the trace transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1631–1645.

[22] V. Govindu, C. Shekhar, Alignment using distributions of local geometric properties, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (10) (1999) 1031–1043.

[23] R. Gan, A. C. S. Chung, Distance-intensity for image registration, in: Proceedings of Computer Vision for Biomedical Image Applications, Vol. 3765 of Lecture Notes in Computer Science, Springer, Beijing, China, 2005, pp. 281–290.

[24] J. Ho, M.-H. Yang, A. Rangarajan, B. Vemuri, A new affine registration algorithm for matching 2D point sets, in: Proceedings of IEEE Workshop on Applications of Computer Vision, IEEE, Austin, Texas, USA, 2007, p. 25.

[25] E. Rahtu, M. Salo, J. Heikkilä, J. Flusser, Generalized affine moment invariants for object recognition, in: Proceedings of International Conference on Pattern Recognition, Vol. 2, IEEE, Hong-kong, 2006, pp. 634–637.

[26] O. Musse, F. Heitz, J.-P. Armspach, Topology preserving deformable image matching using constrained hierarchical parametric models, IEEE Transactions on Image Processing 10 (7) (2001) 1081–1093.

[27] S. Manay, D. Cremers, B.-W. Hong, A. Y. Jr., S. Soatto, Integral invariants for shape matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1602–1618.

[28] F. Hjouj, D. W. Kammler, Identification of reflected, scaled, translated, and rotated objects from their Radon projections, IEEE Transactions on Image Processing 17 (3) (2008) 301–310.

[29] C. Domokos, Z. Kato, J. M. Francos, Parametric estimation of affine deformations of binary images, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, IEEE, Las Vegas, Nevada, USA, 2008, pp. 889–892.

[30] R. Klette, J. Žunić, Multigrid convergence of calculated features in image analysis, Journal of Mathematical Imaging and Vision 13 (3) (2004) 173–191.

[31] R. Klette, J. Žunić, Towards experimental studies of digital moment convergence, Tech. Rep. CITR-TR-61, Computer Science Department of The University of Auckland CITR at Tamaki Campus, Auckland, New Zealand (June 2000).

[32] J. Verschelde, P. Verlinden, R. Cools, Homotopies exploiting newton polytopes for solving sparse polynomial systems, SIAM Journal on Numerical Analysis 31 (3) (1994) 915–930.

[33] J. Verschelde, Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation, ACM Transactions on Mathematical Software 25 (2) (1999) 251–276.

[34] J. Verschelde, PHCpack: a general-purpose solver for polynomial systems by homotopy continuation, software (July 1999).
URL http://www.math.uic.edu/~jan/download.html

[35] S. Belongie, J. Malik, J. Puzicha, Matching with shape contexts, software (April 2002).
URL http://www.eecs.berkeley.edu/Research/Projects/CS/visio%n/shape/sc_digits.html

[36] T. Boudier, The snake plugin for ImageJ, software.
URL http://www.snv.jussieu.fr/~wboudier/softs/snake.html

[37] K. Hardinge, M. L. Porter, P. R. Jones, D. W. L. Hukins, C. J. Taylor, Measurement of hip prostheses using image analysis. the maxima hip technique, Journal of Bone and Joint Surgery 73-B (5) (1991) 724–728.

[38] A. Oprea, C. Vertan, A quantitative evaluation of the hip prosthesis segmentation quality in x-ray images, in: Proceedings of International Symposium on Signals, Circuits and Systems, Vol. 1, IEEE, Iasi, Romania, 2007, pp. 1–4.

[39] R. Merris, Combinatorics, 2nd Edition, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, 2003.

Author Biography

Csaba Domokos received the M.Sc. degrees in computer science from the University of Szeged, Hungary in 2006. He is currently pursuing the Ph.D. degree in the Institute of Informatics at University of Szeged. His current research interests include image registration, image transformations and shape matching.

Zoltan Kato received the M.S. degree in Computer Science from the University of Szeged, Hungary in 1990, and the Ph.D. degree from the University of Nice doing his research at INRIA Sophia Antipolis, France in 1994. Currently, he is head of the Department of Image Processing and Computer Graphics at the University of Szeged. His research interests include statistical image models, MCMC methods, and shape modeling.

$\delta = 4.15\%$ $\delta = 3.00\%$ $\delta = 3.14\%$ $\delta = 2.19\%$

$\delta = 4.30\%$ $\delta = 1.07\%$ $\delta = 8.09\%$ $\delta = 2.42\%$
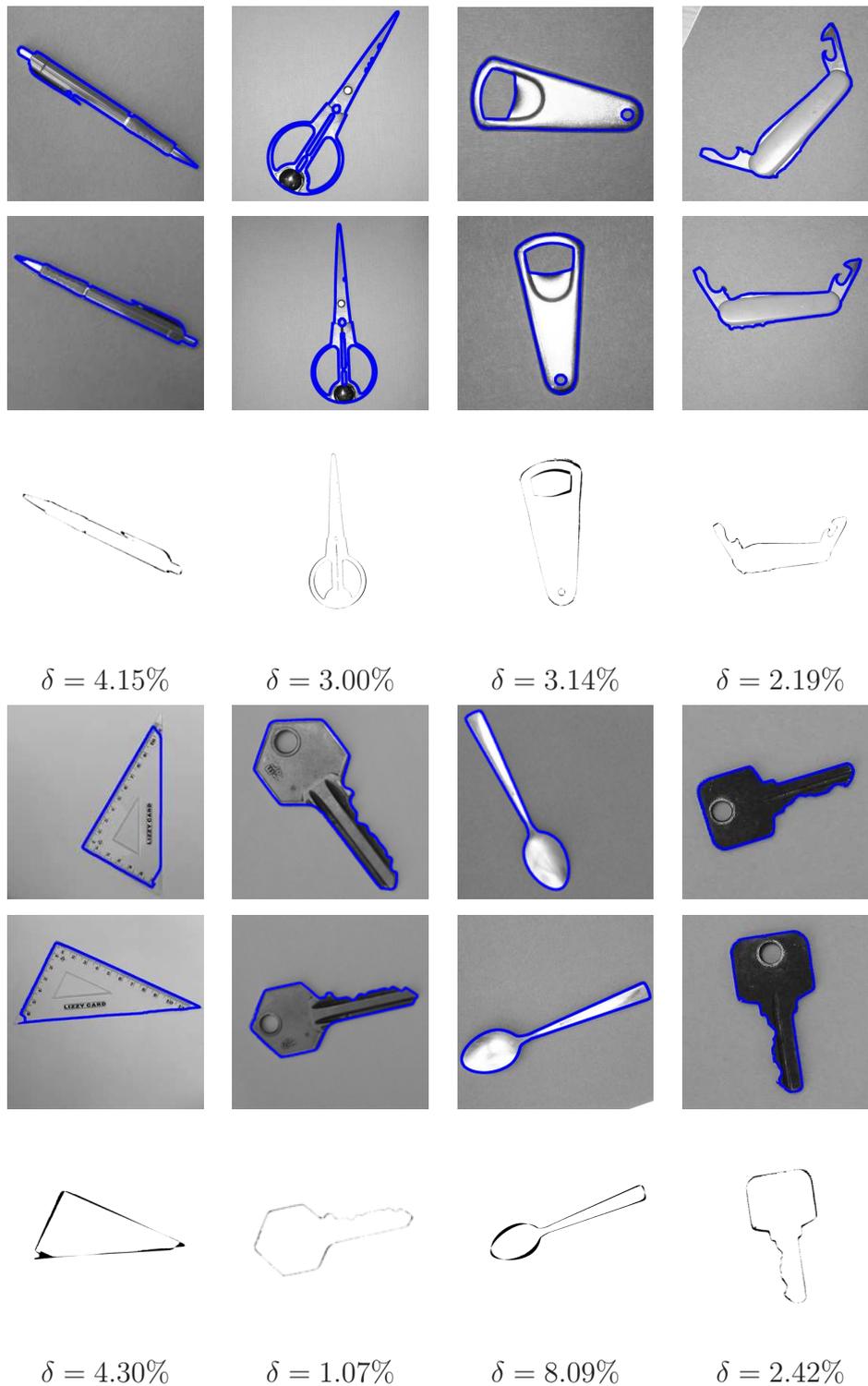
Fig. 5. Registration results on real images. For each image pair, the first two rows contain the *template* and *observation* with overlayed contours of the segmented silhouettes, while the third row shows the difference between the *registered* shapes and in the last row the evaluated error measure δ is given. Note that this value is related to the overlapping area hence it depends also on segmentation errors as segmented regions will never match perfectly.

27

$\delta = 3.69\%$ $\delta = 7.62\%$ $\delta = 5.94\%$ $\delta = 4.13\%$ $\delta = 1.45\%$
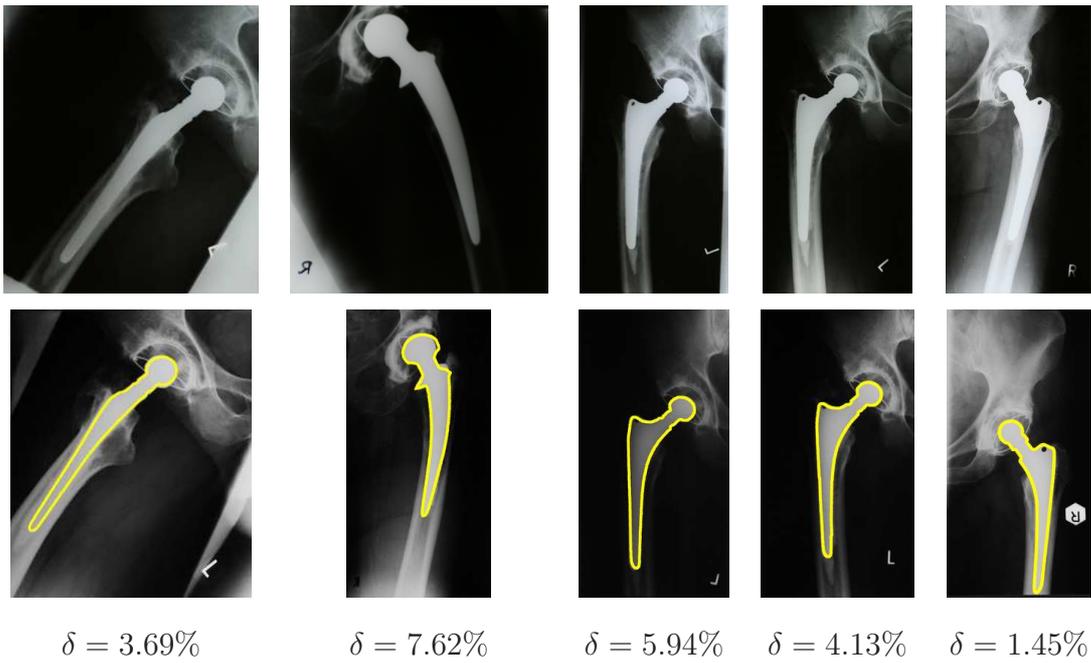
Fig. 6. Registration of hip prosthesis X-ray images. Each image pair has been taken over a period of time about the same patient. The overlayed contour in the second row shows the aligned contour of the corresponding image in the first row. For each pair, we have also evaluated the δ measure.