Pattern Recognition Letters 33 (2012) 349–355

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Least-squares estimation of anisotropic similarity transformations from corresponding 2D point sets

Carsten Steger [*]

MVTec Software GmbH, Neherstraße 1, 81675 München, Germany

ARTICLE INFO

Article history:
Received 21 March 2011
Available online 4 November 2011
Communicated by S. Sarkar

Keywords:
Computer vision
Pose estimation
Least-squares adjustment

ABSTRACT

Pose estimation is a problem that occurs in many applications. In machine vision, the pose is often a 2D affine pose. In several applications, a restricted class of 2D affine poses with five degrees of freedom consisting of an anisotropic scaling, a rotation, and a translation must be determined from corresponding 2D points. A closed-form least-squares solution for this problem is described. The algorithm can be extended easily to robustly deal with outliers.

1. Introduction

Computing a least-squares transformation between corresponding point sets is a task that must be solved in many disciplines, for example to register images in computer vision applications (Zitová and Flusser, 2003; Hartley and Zisserman, 2003; Xiong and Zhang, 2010) or in medical applications (Fitzpatrick et al., 2000; Hill et al., 2001; Bankman, 2009), to compute the absolute orientation of a 3D reconstruction in photogrammetry (McGlone et al., 2004), or for factor analysis in statistics, where the problem is known as the Procrustes problem (Gower and Dijksterhuis, 2004).

The transformation between the point sets is frequently modeled as an affine transformation or a subclass thereof. For many different subclasses of affine transformations, closed-form solutions have been derived:

- 2D rigid transformations (rotation and translation; 3 degrees of freedom), e.g., Haralick et al. (1989) and Umeyama (1991);
- 2D similarity transformations (isotropic scaling, rotation, and translation; 4 degrees of freedom), e.g., Umeyama (1991);
- general 2D affine transformations (6 degrees of freedom), e.g., Lamdan et al. (1990) and Hartley and Zisserman (2003);
- 3D rigid transformations (rotation and translation; 6 degrees of freedom), e.g., Arun et al. (1987), Horn (1987), Horn et al. (1988), Haralick et al. (1989), Umeyama (1991), Walker et al. (1991), Fitzpatrick et al. (2000, Chapter 8.3.1) and Hill et al. (2001);
- 3D similarity transformations (isotropic scaling, rotation, and translation; 7 degrees of freedom), e.g., Umeyama (1991) and Fitzpatrick et al. (2000, Chapter 8.3.2);
- general 3D affine transformations (12 degrees of freedom), e.g., Gower and Dijksterhuis (2004).

In some applications, a subclass of affine transformations that is more restricted than general affine transformations but more flexible than similarity transformations must be determined. In this class, the transformations comprise an anisotropic scaling, a rotation, and a translation. This results in five degrees of freedom for 2D transformations and nine degrees of freedom for 3D transformations. We will call this class of transformations anisotropic similarity transformations. In 3D applications in medicine, this class of transformations occurs, for example, because the voxel size in 3D images may be unknown or not known accurately, e.g., as a result of a miscalibration of the voxel dimensions (Hill et al., 2001).

In 2D machine vision applications, this kind of transformation occurs in different scenarios, in particular when a model must be registered to an image. For example, in print inspection applications it is typically assumed that the pixels of the model of the print are square. When the document is printed, however, the printing process may result in non-square pixels. If such a document is then inspected, it might be visible in the field of view slightly rotated and translated. This results in an overall affine transformation with five degrees of freedom. In other applications, the image of an object to be inspected is acquired with a device that has non-square pixels, which can happen, for example, if the image is acquired with a line scan camera (Steger et al., 2007). Here, the object may be rotated and translated in the world in a fixed 2D plane, and the unknown aspect ratio of the sensor causes an anisotropic scaling. In both kinds of inspection applications, it is undesirable to include the sixth degree of freedom of the full affine transformation (a skew) because this could result in certain defects to be missed in the inspection.

[*] Tel.: +49 89 4576950; fax: +49 89 45769555.
E-mail address: steger@mvtec.com

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.10.015

Algorithms to determine anisotropic similarity transformations from point correspondences are described, for example, in Fitzpatrick et al. (2000, Chapter 8.3.2), Batchelor et al. (2002), Gower and Dijksterhuis (2004, Chapter 8.3), Gower (2010), and Bennani Dosse and Ten Berge (2010). All of these algorithms, however, are iterative minimization algorithms, which are undesirable in machine vision applications for two reasons. First, they can easily converge to local minima (Bennani Dosse and Ten Berge, 2010). Second, they are less efficient than closed-form solutions. In machine vision applications, however, speed is often critical. Therefore, this paper derives a closed-form solution for the problem for determining an anisotropic similarity transformation from 2D point correspondences.

2. Algorithm

The problem to be solved can be stated as follows: given two sets of corresponding points x_n and y_n, $n = 1,\dots,N$, find the anisotropic similarity matrix A and the translation vector t that minimize the weighted sum of residual errors:

$$\varepsilon^2 = \sum_{n=1}^{N} w_n \|y_n - (Ax_n + t)\|^2. \tag{1}$$

The weights w_n must be greater than 0 and sum to 1. They can be used to make the algorithm robust to outliers, as described below. For a regular least-squares algorithm, they can be set to $1/N$.

It is obvious that $N \geqslant 3$ point correspondences in general position are required to solve (1) (i.e., the points x_n and y_n must not all be collinear, respectively).

The matrix A consists of a rotation R and an anisotropic scaling S, given by

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \qquad S = \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix}. \tag{2}$$

Note that R and S do not commute (Fitzpatrick et al., 2000, Chapter 8.2.2.1). Therefore, there are two different classes of anisotropic similarity transformations, corresponding to the two different scenarios described in Section 1: $A = RS$ and $A = SR$. We will denote these classes of transformation as pre-scaling and post-scaling anisotropic similarity transformations, respectively. Note that these classes of transformations are distinct. With the pre-scaling anisotropic similarity transformations, an axis-aligned square can be transformed into a rotated rectangle, but not into a rhomboid. With post-scaling anisotropic similarity transformations, an axis-aligned square can be transformed into a rhomboid, but not into a rotated rectangle. Also note that both classes of transformations do not constitute a subgroup of the group of affine transformations because the composition of two anisotropic similarity transformations generally results in a full affine transformation and because the inverse of a pre-scaling anisotropic similarity transformation is a post-scaling anisotropic similarity transformation and vice versa.

We will derive the algorithm for pre-scaling anisotropic similarity transformations, i.e., for $A = RS$. Our derivation of the algorithm is based on the discussion in Haralick et al. (1989, Section II).

We start be expanding the error term in (1):

$$\varepsilon^2 = \sum_{n=1}^{N} w_n \left[(y_n - t)^T (y_n - t) - 2(y_n - t)^T A x_n + x_n^T A^T A x_n \right]. \tag{3}$$

To compute the translation t, we take the partial derivative of (3) with respect to t and set it to 0:

$$0 = \sum_{n=1}^{N} w_n [-2(y_n - t) + 2Ax_n]. \tag{4}$$

If we define $\bar{x}$ and $\bar{y}$ as the weighted centroids of the two point sets

$$\bar{x} = \sum_{n=1}^{N} w_n x_n \qquad \bar{y} = \sum_{n=1}^{N} w_n y_n. \tag{5}$$

we can see that $\bar{y} = A\bar{x} + t$, i.e., that the translation t is given by

$$t = \bar{y} - A\bar{x}. \tag{6}$$

We can substitute t into (3) to obtain

$$\varepsilon^2 = \sum_{n=1}^{N} w_n \Big[(y_n - \bar{y})^T (y_n - \bar{y}) - 2(y_n - \bar{y})^T A (x_n - \bar{x}) \\ + (x_n - \bar{x})^T A^T A (x_n - \bar{x}) \Big]. \tag{7}$$

Note that the coordinates in (7) are all given with respect to their weighted centroids. To simplify the discussion, we will make the following substitutions: $x'_n = x_n - \bar{x}$ and $y'_n = y_n - \bar{y}$. Additionally, since $A^TA = S^TR^TRS = S^TS = S^2$, (7) can be simplified to

$$\varepsilon^2 = \sum_{n=1}^{N} w_n \left[y'^T_n y'_n - 2 y'^T_n R S x'_n + x'^T_n S^2 x'_n \right]. \tag{8}$$

Finally, we will expand the terms in this sum so that the dependence on the parameters s_1, s_2, and θ is made explicit. If we denote the components of the vectors as $v = (v_1, v_2)^T$, we obtain:

$$\varepsilon^2 = \sum_{n=1}^{N} w_n \Big[y'^2_{n1} + y'^2_{n2} \\ -2\big(x'_{n1}y'_{n1}s_1\cos\theta - x'_{n2}y'_{n1}s_2\sin\theta \\ + x'_{n1}y'_{n2}s_1\sin\theta + x'_{n2}y'_{n2}s_2\cos\theta\big) \\ + x'^2_{n1}s_1^2 + x'^2_{n2}s_2^2 \Big]. \tag{9}$$

To determine s_1, s_2, and θ, the gradient of ε^2 with respect to these parameters must vanish:

$$\frac{\partial\varepsilon^2}{\partial s_1} = -2\sum_{n=1}^{N} w_n \left[x'_{n1}y'_{n1}\cos\theta + x'_{n1}y'_{n2}\sin\theta - x'^2_{n1}s_1 \right] = 0 \tag{10}$$

$$\frac{\partial\varepsilon^2}{\partial s_2} = -2\sum_{n=1}^{N} w_n \left[x'_{n2}y'_{n2}\cos\theta - x'_{n2}y'_{n1}\sin\theta - x'^2_{n2}s_2 \right] = 0 \tag{11}$$

$$\frac{\partial\varepsilon^2}{\partial\theta} = -2\sum_{n=1}^{N} w_n [-x'_{n1}y'_{n1}s_1\sin\theta - x'_{n2}y'_{n1}s_2\cos\theta \\ + x'_{n1}y'_{n2}s_1\cos\theta - x'_{n2}y'_{n2}s_2\sin\theta] = 0. \tag{12}$$

The above equations can be simplified by introducing the following variables:

$$a = \sum_{n=1}^{N} w_n x'_{n1} y'_{n1} \qquad b = \sum_{n=1}^{N} w_n x'_{n1} y'_{n2} \\ c = \sum_{n=1}^{N} w_n x'_{n2} y'_{n1} \qquad d = \sum_{n=1}^{N} w_n x'_{n2} y'_{n2} \\ e = \sum_{n=1}^{N} w_n x'^2_{n1} \qquad f = \sum_{n=1}^{N} w_n x'^2_{n2}. \tag{13}$$

With this, (10)–(12) reduce to:

$$a\cos\theta + b\sin\theta - es_1 = 0 \tag{14}$$

$$d\cos\theta - c\sin\theta - fs_2 = 0 \tag{15}$$

$$as_1\sin\theta + cs_2\cos\theta - bs_1\cos\theta + ds_2\sin\theta = 0. \tag{16}$$

From (14) and (15), we obtain:[1]

$$s_1 = \frac{a}{e}\cos\theta + \frac{b}{e}\sin\theta \tag{17}$$

$$s_2 = \frac{d}{f}\cos\theta - \frac{c}{f}\sin\theta \tag{18}$$

[1] Note that the degenerate cases $e = 0$ and $f = 0$ occur only if all x_n lie on a vertical or horizontal line, respectively. In this case, the x_n are not in general position and consequently the transformation is undefined.

We substitute s_1 and s_2 into (16) and group the terms:

$$\begin{aligned}&\left(\frac{a}{e}\cos\theta+\frac{b}{e}\sin\theta\right)(a\sin\theta-b\cos\theta)\\&+\left(\frac{d}{f}\cos\theta-\frac{c}{f}\sin\theta\right)(c\cos\theta+d\sin\theta)=\\&\left(\frac{a^2-b^2}{e}+\frac{d^2-c^2}{f}\right)\sin\theta\cos\theta\\&+\left(\frac{cd}{f}-\frac{ab}{e}\right)(\cos^2\theta-\sin^2\theta)=\\&\frac{1}{2}\left(\frac{a^2-b^2}{e}+\frac{d^2-c^2}{f}\right)\sin 2\theta+\left(\frac{cd}{f}-\frac{ab}{e}\right)\cos 2\theta=0.\end{aligned} \tag{19}$$

By letting $\phi = 2\theta$ and

$$g=\frac{1}{2}\left(\frac{a^2-b^2}{e}+\frac{d^2-c^2}{f}\right) \qquad h=\left(\frac{cd}{f}-\frac{ab}{e}\right) \tag{20}$$

(19) reduces to:

$$g\sin\phi+h\cos\phi=0. \tag{21}$$

This equation has the following two solutions (Haralick et al., 1989):

$$\cos\phi=\frac{-g}{\sqrt{g^2+h^2}} \qquad \sin\phi=\frac{h}{\sqrt{g^2+h^2}} \tag{22}$$

and

$$\cos\phi=\frac{g}{\sqrt{g^2+h^2}} \qquad \sin\phi=\frac{-h}{\sqrt{g^2+h^2}}. \tag{23}$$

Note that the two solutions correspond to values for ϕ that differ by π, i.e., values for θ that differ by $\pi/2$. Hence, it is clear from the geometry of the problem that only one solution can be correct. In fact, we can show that one solution corresponds to a minimum of ε^2, while the other one corresponds to a maximum.

To see which solution is correct, we can examine the Hessian matrix of ε^2, i.e., the matrix that corresponds to the second partial derivatives of ε^2. At a minimum, the Hessian matrix must be positive definite. A necessary condition for the Hessian matrix to be positive definite is that all its diagonal elements are positive. This is a consequence of Sylvester's criterion (Gilbert, 1991). In particular, the second derivative of ε^2 with respect to θ must be positive. Hence, we require

$$2g\cos 2\theta-2h\sin 2\theta>0 \iff g\cos\phi-h\sin\phi>0. \tag{24}$$

By substituting (22) and (23) into (24), we obtain

$$\frac{-g^2-h^2}{\sqrt{g^2+h^2}}<0 \tag{25}$$

and

$$\frac{g^2+h^2}{\sqrt{g^2+h^2}}>0. \tag{26}$$

Therefore, the solution is given by (23), i.e.,

$$\theta=\frac{1}{2}\operatorname{atan2}(-h,g) \tag{27}$$

is the correct rotation.[2] Here, atan2 denotes the arctangent function that correctly takes the signs of its arguments into account.

To derive an algorithm for post-scaling anisotropic similarity transformations, where $A = SR$, we could proceed as above. However, in this case, the term A^TA is given by $R^TS^TSR = R^TS^2R$, which cannot be simplified and leads to complicated rational polynomial functions in $\cos\theta$ and $\sin\theta$ that are very difficult to solve in closed form. However, as noted above, the inverse of a post-scaling anisotropic similarity transformation is a pre-scaling anisotropic similarity transformation. Therefore, to solve (1) for $A = SR$, we can simply exchange x and y in (1), i.e., use the point correspondences in the opposite order in the above algorithm. We only need to invert the resulting transformation to obtain the desired post-scaling anisotropic similarity transformation. It should be noted, however, that this algorithm minimizes errors that are semantically slightly different since they are calculated in the frame of x, whereas they should ideally be computed in the frame of y, like for pre-scaling anisotropic similarity transformations. However, as the experiments below show, the results are nevertheless unbiased. Therefore, the algorithm will work as desired for almost all practical purposes.

The algorithm derived above is not robust to outliers, which can, for example, be caused by erroneous point matches. To make this algorithm robust to outliers, it can be embedded into any of the well-known algorithms for outlier suppression. For example, the algorithm can be used directly in the iteratively reweighted least-squares (IRLS) algorithm described in Haralick et al. (1989, Section II.C). Here, the weights w_n and the transformation are computed iteratively, where the initial iteration uses $w_n = 1/N$ and the weights in later iterations are computed from the residual errors that result from the transformation computed in the previous iteration using certain cost functions, such as Huber's cost function (Huber, 1981; Stewart, 1999) or Tukey's biweight cost function (Mosteller and Tukey, 1977; Stewart, 1999). Alternatively, for even greater robustness, the algorithm can be embedded into methods such as random sample consensus (RANSAC) (Fischler and Bolles, 1981; Stewart, 1999; Hartley and Zisserman, 2003) or least median of squares (LMedS) (Rousseeuw, 1984; Stewart, 1999; Hartley and Zisserman, 2003). Since the algorithm computes a closed-form solution very rapidly, it is ideally suited to generate hypotheses from three samples as well as to compute the actual transformation in these methods.

To conclude the discussion of the algorithm, Fig. 1 gives an example of the results that can be obtained with the proposed algorithm. Fig. 1(a) shows a noisy square, which was created by adding Gaussian noise to the coordinates of a perfect square. Fig. 1(b) shows a noisy rotated rectangle (solid line), which was created by transforming the perfect square with a pre-scaling anisotropic similarity transformation and then adding Gaussian noise to the transformed coordinates. The result of computing the optimal pre-scaling anisotropic similarity transformation is shown by the dashed line. The transformed noisy square aligns very well with the noisy rectangle. Fig. 1(c) shows a noisy rhomboid (solid line). It was created by transforming the perfect square with a post-scaling anisotropic similarity transformation and then adding Gaussian noise. The dashed line visualizes the result of computing the optimal pre-scaling anisotropic similarity transformation. As is to be expected, there are large residual errors because a pre-scaling anisotropic similarity transformation cannot transform a square into a rhomboid. In fact, this example has been included merely to show that the two classes of transformations are distinct. Nevertheless, the transformed square aligns with the rhomboid as well as possible. Finally, Fig. 1(d) displays the same noisy rhomboid as Fig. 1(c) as well as the result of fitting a post-scaling anisotropic similarity transformation (dashed line). The transformed noisy square aligns very well with the noisy rhomboid.

[2] To be completely precise, $\theta + \pi$ would also be a valid solution. However, in this case we simply obtain negated scale factors s_1 and s_2 from (17) and (18). This defines exactly the same transformation since $\theta + \pi$ as well as $-s_1$ and $-s_2$ both describe rotations by π with respect to the solution obtained from θ, s_1, and s_2, which cancel each other in RS.

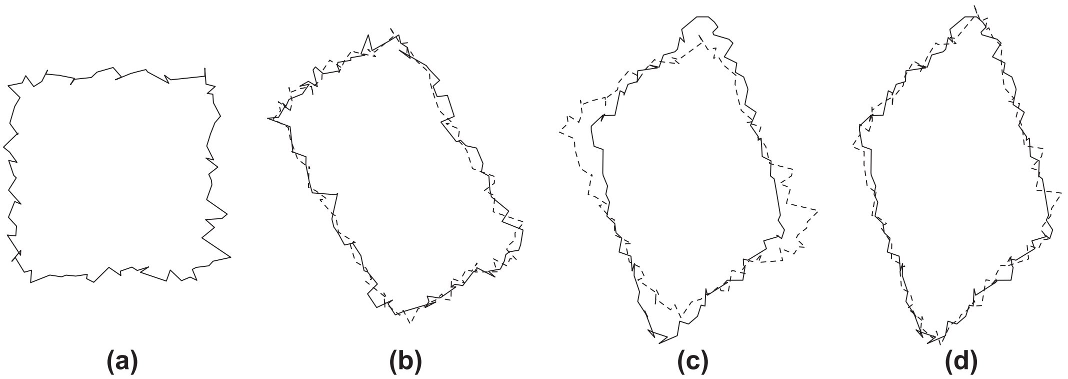


Fig. 1. Example of the results that are obtained by fitting an anisotropic similarity transformation. (a) A noisy square. (b) A noisy rectangle (solid line) and the result of transforming the noisy square in (a) with the optimal pre-scaling anisotropic similarity transformation (dashed line). (c) A noisy rhomboid (solid line) and the result of transforming the noisy square in (a) with the optimal pre-scaling anisotropic similarity transformation (dashed line). (d) A noisy rhomboid (solid line) and the result of transforming the noisy square in (a) with the optimal post-scaling anisotropic similarity transformation (dashed line).

3. Performance evaluation

To evaluate the performance of the algorithm, the following experiment was performed: N points $\tilde{x}_n$ were selected randomly, where $N \in \{3, 10, 30, 100, 300, 1000\}$. Note that $N = 3$ tests the minimal number of point correspondences. The N random points were chosen uniformly within the square $[-1000, 1000] \times [-1000, 1000]$. Anisotropic scaling factors $\tilde{s}_1$ and $\tilde{s}_2$ were chosen randomly from the interval $[0.25, 4]$, rotation angles $\tilde{\theta}$ were chosen randomly from the interval $[-90\,°, 90\,°]$, translations $\tilde{t}_1$ and $\tilde{t}_2$ were chosen randomly from the interval $[-500, 500]$, and a pre-scaling anisotropic similarity transformation $\tilde{T}$ was computed from these five random values. The random points were transformed with $\tilde{T}$, yielding a set of random points $\tilde{y}_n$ in perfect correspondence with $\tilde{x}_n$. The coordinates of the point correspondences were then modified with Gaussian noise of varying standard deviations $\sigma \in [0, 5]$, yielding noisy corresponding points x_n and y_n. A pre-scaling anisotropic similarity transformation T was computed from the noisy point correspondences. For each N and σ, the computation of the transformation was repeated 1000 times. From these results, various statistics were derived.

The first kind of statistics that characterize the performance of the algorithm are the root-mean-square (RMS) residual error and the RMS estimation error. We define the RMS residual error similar to Hartley and Zisserman (2003, Chapter 5.1):

$$\varepsilon_{\mathrm{res}} = \frac{1}{\sqrt{4N}} \left(\sum_{n=1}^{N} d(Tx_n, y_n)^2 + \sum_{n=1}^{N} d\left(x_n, T^{-1} y_n\right)^2 \right)^{\frac{1}{2}} \tag{28}$$

Here, $d(x, y)$ denotes the Euclidean distance of x and y. Furthermore, we define the estimation error based on the perfect point correspondences and the computed transformation as follows:

$$\varepsilon_{\mathrm{est}} = \frac{1}{\sqrt{4N}} \left(\sum_{n=1}^{N} d(T\tilde{x}_n, \tilde{y}_n)^2 + \sum_{n=1}^{N} d\left(\tilde{x}_n, T^{-1} \tilde{y}_n\right)^2 \right)^{\frac{1}{2}} \tag{29}$$

Note that the symmetric nature of the above two errors, which use T and T^{-1}, implies that the performance evaluation also tests the performance of post-scaling anisotropic similarity transforms.

For each N and σ, the mean residual error and the mean estimation error were computed. Fig. 2 shows the two kinds of errors plotted as a function of N and σ. It can be seen that the error plots have the shape predicted by Hartley and Zisserman (2003, Result 5.2): $\varepsilon_{\mathrm{res}}$ increases with increasing N, while $\varepsilon_{\mathrm{est}}$ decreases with increasing N. Furthermore, both kinds of error increase linearly with σ.

The precision of the transformation parameters was computed from the standard deviation of the errors in the estimated transformation parameters. The standard deviations are shown in Fig. 3. It can be seen that the scale factors s_1 and s_2 and the rotation angle θ are determined very precisely. The translations t_1 and t_2, on the other hand, might seem to be determined less precisely. Note, however, that s_1, s_2, and θ are dimensionless quantities, whereas t_1 and t_2 have a length dimension. Therefore, the standard deviations of t_1 and t_2 scale with the scale of the input data (which is in the order of 1000). As the RMS residual and estimation errors in Fig. 2 show, the estimated transformation parameters are consistent and correct. Thus, the relatively large variations of the translation parameters are caused by the choice of the scale of the data in the experiments.

To verify that the algorithm produces accurate results, the estimated transformation parameters were compared to their true values. For each N and σ, the mean values of $s_1 - \tilde{s}_1, s_2 - \tilde{s}_2, \theta - \tilde{\theta}$, $t_1 - \tilde{t}_1$, and $t_2 - \tilde{t}_2$ were computed. Fig. 4 shows these errors. To check that the algorithm computes unbiased results, the mean errors in Fig. 3 should be compared to the standard error of these means, which are given by dividing the standard deviations in Fig. 3 by the square root of the number of tries, i.e., by $\sqrt{1000} = 31.62$. It can be seen that the mean errors are smaller than (for $N > 3$) or at most as large as (for $N = 3$) the standard error of the means. Consequently, it can be seen that the algorithm computes unbiased estimates of the transformation parameters.

4. Applications

To give an impression about an application area for which the proposed algorithm is useful, we will discuss a print inspection application in this section. To keep the discussion simple and the length of the paper reasonable, the application has been stripped down to its essence, leaving out all the engineering details that must be solved in a real-world application (lighting, image acquisition, image resolution, etc.; see, e.g., (Steger et al., 2007) for more information).

Print inspection applications are typically solved by specifying an image of the ideal print that is expected to be seen in the image (the "model image"). In many applications, the model image is simply specified by presenting the inspection system with a sample of a known good print. In other applications, the model image is generated directly from a PDF file. The task to find any relevant differences between the model image and an image of the print to be inspected (the "test image"). Often, this task is simply solved by comparing the model image and the test image. First, the test image is aligned with the model image. In the aligned images, areas

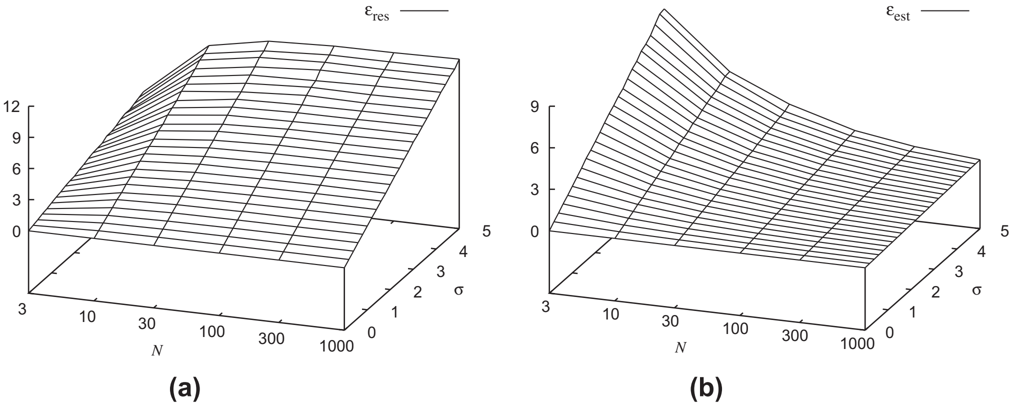


Fig. 2. (a) Mean residual errors computed using (28). (b) Mean estimation errors computed using (29).

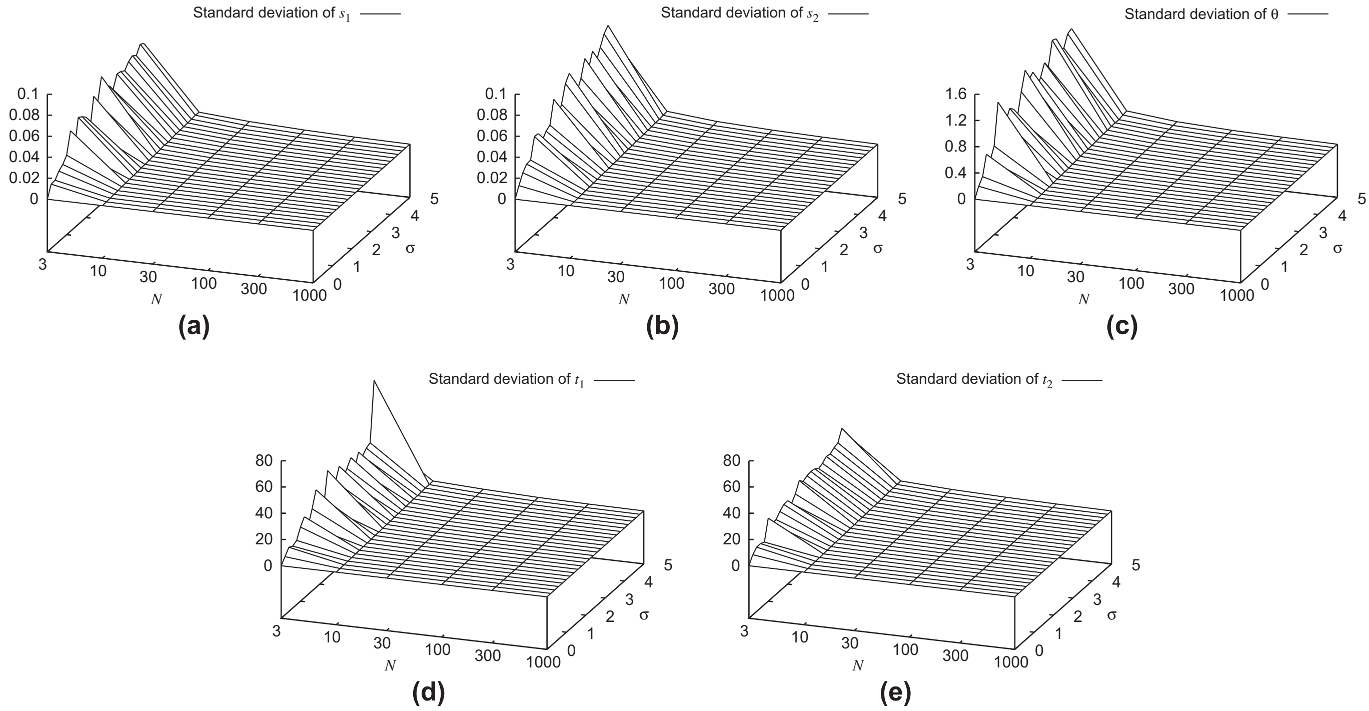


Fig. 3. Standard deviation of the errors of the transformation parameters: (a) s_1, (b) s_2, (c) θ (in degrees), (d) t_1, and (e) t_2.

for which the gray value differences are too large are marked as errors. More sophisticated techniques take the potential variability of acceptable prints into account (Steger et al., 2007, Chapter 3.4.1), but in essence are still image comparison techniques. For these kinds of applications, obviously the brightness and contrast of the model and test images must be matched before the images are compared. A different method to inspect prints is to require that the edges in the model and test images are identical (Perkins, 1983). The advantage of this strategy is that it is robust to brightness and contrast changes. It is also possible to combine both types of inspection strategies (Perkins, 1983).

In print inspection applications, it often happens that the printing process changes the aspect ratio of the ideal print for various reasons. The interested reader is invited to print a square of, say, size 15 cm × 15 cm on a laser printer. The printed square will frequently be a rectangle.[3] The deviation is often small, typically in the order of 0.5–1%, but is definitively noticeable for an inspection system, especially if resolutions of 300 DPI or more are used. Thus, aspect ratio changes must be handled correctly. Furthermore, if the model image is generated from synthetic data, an overall scale change must also be handled. This can be modeled as two independent scale factors. Furthermore, the print often is seen slightly rotated and translated in the test image because a perfect mechanical alignment is very difficult to ensure. Overall, this results in a pre-scaling anisotropic similarity transformation that must be determined when the model and test images are aligned.

Fig. 5(a) shows an example of a model image (a German tax form). This image was directly created from a PDF file at a resolution of 300 DPI. To simulate a test image, four errors were edited into the PDF file. Furthermore, the PDF file was scaled by 99.5% in the vertical direction and rotated by 0.5°. The test image, shown in Fig. 5(b), was created from the modified PDF file. Thus, as stated above, for the purposes of this section, we ignore most of the engineering issues of this application and simply simulate the test image. The alignment can be performed by template matching,

[3] This problem is also discussed in the context of camera and hand-eye calibration by Strobl and Hirzinger (2008).

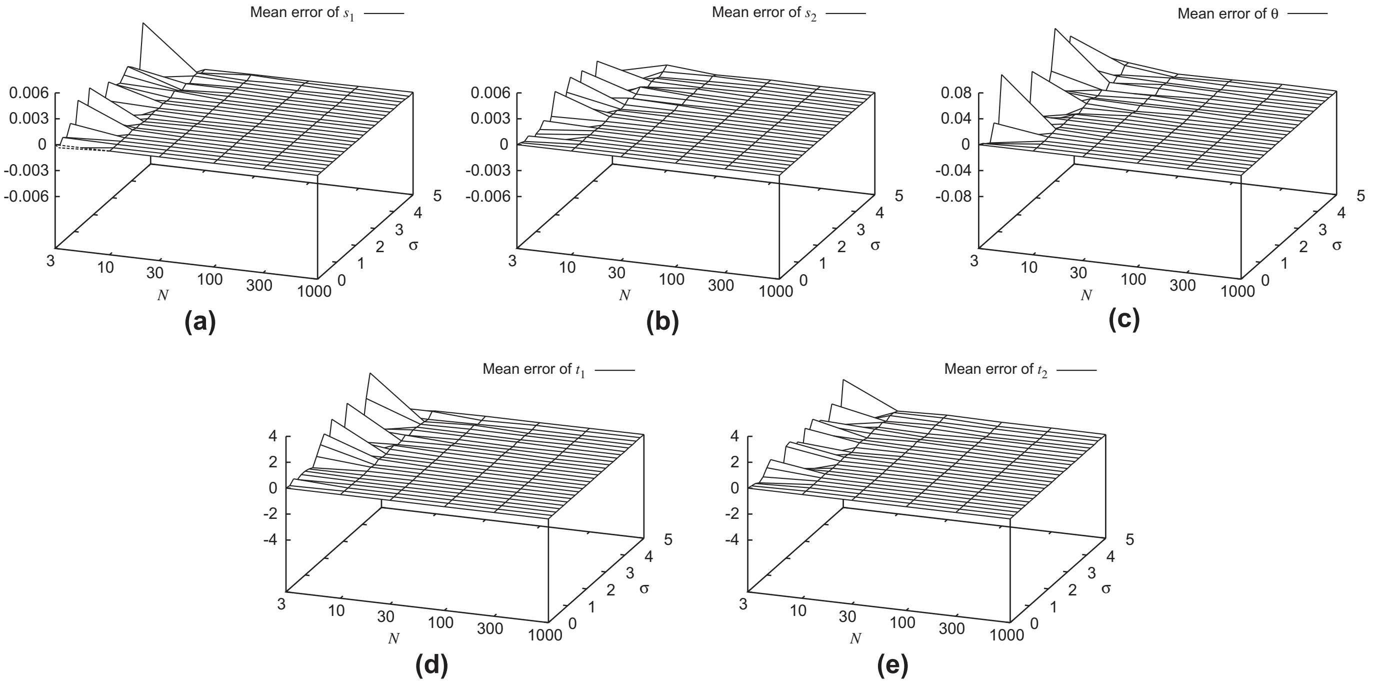
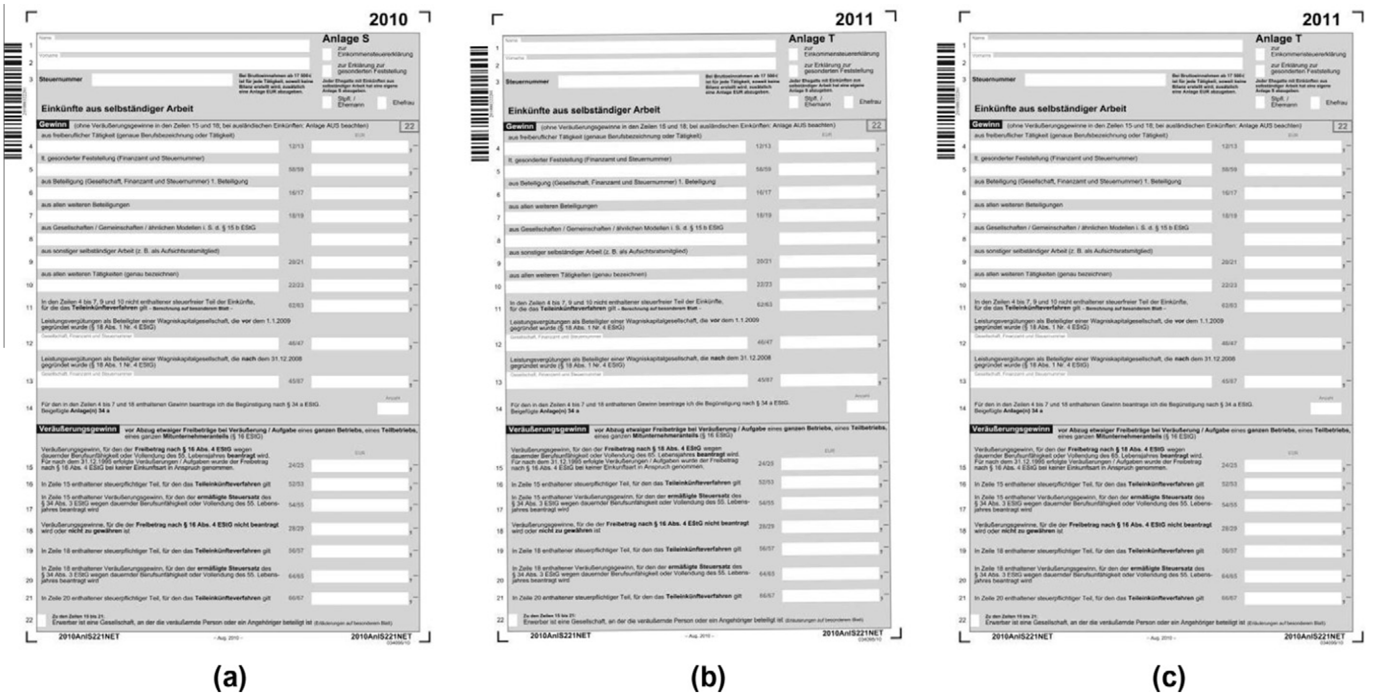


Fig. 4. Mean errors of the transformation parameters: (a) s_1, (b) s_2, (c) θ (in degrees), (d) t_1, and (e) t_2.


2010

Name
1
Vorname
2
3 Steuernummer

Bei Bruttoeinnahmen ab 17 500 € ist für jede Tätigkeit, soweit keine Bilanz erstellt wird, zusätzlich eine Anlage EÜR abzugeben.

Anlage S

□ zur Einkommensteuererklärung

□ zur Erklärung zur gesonderten Feststellung

Jeder Ehegatte mit Einkünften aus selbständiger Arbeit hat eine eigene Anlage S abzugeben.

□ Stpfl. / Ehemann □ Ehefrau

Einkünfte aus selbständiger Arbeit

Gewinn (ohne Veräußerungsgewinne in den Zeilen 15 und 18; bei ausländischen Einkünften: Anlage AUS beachten) 22

aus freiberuflicher Tätigkeit (genaue Berufsbezeichnung oder Tätigkeit) EUR

4 12/13 ,–

lt. gesonderter Feststellung (Finanzamt und Steuernummer)

5 58/59 ,–

aus Beteiligung (Gesellschaft, Finanzamt und Steuernummer) 1. Beteiligung

6 16/17 ,–

aus allen weiteren Beteiligungen

7 18/19 ,–

aus Gesellschaften / Gemeinschaften / ähnlichen Modellen i. S. d. § 15 b EStG

8 ,–

aus sonstiger selbständiger Arbeit (z. B. als Aufsichtsratsmitglied)

9 20/21 ,–

aus allen weiteren Tätigkeiten (genau bezeichnen)

10 22/23 ,–

11 In den Zeilen 4 bis 7, 9 und 10 nicht enthaltener steuerfreier Teil der Einkünfte, für die das **Teileinkünfteverfahren** gilt – Berechnung auf besonderem Blatt – 62/63 ,–

Leistungsvergütungen als Beteiligter einer Wagniskapitalgesellschaft, die **vor** dem 1.1.2009 gegründet wurde (§ 18 Abs. 1 Nr. 4 EStG)

Gesellschaft, Finanzamt und Steuernummer

12 46/47 ,–

Leistungsvergütungen als Beteiligter einer Wagniskapitalgesellschaft, die **nach** dem 31.12.2008 gegründet wurde (§ 18 Abs. 1 Nr. 4 EStG)

Gesellschaft, Finanzamt und Steuernummer

13 45/87 ,–

14 Für den in den Zeilen 4 bis 7 und 18 enthaltenen Gewinn beantrage ich die Begünstigung nach § 34 a EStG. Beigefügte **Anlage(n) 34 a** Anzahl

Veräußerungsgewinn **vor Abzug etwaiger Freibeträge bei Veräußerung / Aufgabe** eines **ganzen Betriebs**, eines **Teilbetriebs**, eines ganzen **Mitunternehmeranteils** (§ 16 EStG)

15 Veräußerungsgewinn, für den der **Freibetrag nach § 16 Abs. 4 EStG** wegen dauernder Berufsunfähigkeit oder Vollendung des 55. Lebensjahres **beantragt** wird. Für nach dem 31.12.1995 erfolgte Veräußerungen / Aufgaben wurde der Freibetrag nach § 16 Abs. 4 EStG bei keiner Einkunftsart in Anspruch genommen. EUR 24/25 ,–

16 In Zeile 15 enthaltener steuerpflichtiger Teil, für den das **Teileinkünfteverfahren** gilt 52/53 ,–

17 In Zeile 15 enthaltener Veräußerungsgewinn, für den der **ermäßigte Steuersatz** des § 34 Abs. 3 EStG wegen dauernder Berufsunfähigkeit oder Vollendung des 55. Lebensjahres beantragt wird 54/55 ,–

18 Veräußerungsgewinne, für die der **Freibetrag nach § 16 Abs. 4 EStG nicht beantragt** wird oder **nicht zu gewähren** ist 28/29 ,–

19 In Zeile 18 enthaltener steuerpflichtiger Teil, für den das **Teileinkünfteverfahren** gilt 56/57 ,–

20 In Zeile 18 enthaltener Veräußerungsgewinn, für den der **ermäßigte Steuersatz** des § 34 Abs. 3 EStG wegen dauernder Berufsunfähigkeit oder Vollendung des 55. Lebensjahres beantragt wird 64/65 ,–

21 In Zeile 20 enthaltener steuerpflichtiger Teil, für den das **Teileinkünfteverfahren** gilt 66/67 ,–

22 □ Zu den Zeilen 15 bis 21: Erwerber ist eine Gesellschaft, an der die veräußernde Person oder ein Angehöriger beteiligt ist (Erläuterungen auf besonderem Blatt)

2010AnlS221NET – Aug. 2010 – 2010AnlS221NET

Fig. 5. (a) Model image (a German tax form). (b) Simulation of a test image (a German tax form with several errors, scaled by 99.5% in the vertical direction and rotated by 0.5 °). (c) The test image in (b) aligned with the model image in (a) using template matching of the four L-shaped fiducials and computing a pre-scaling anisotropic similarity transformation from the resulting four point correspondences.

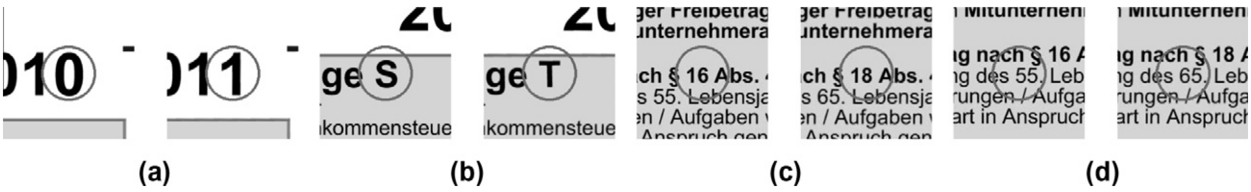


Fig. 6. The four errors detected in the test image in Fig. 5(c).

using the four L-shaped fiducials in the corners of the form (Steger, 2002). This results in four corresponding point pairs in the model and test images. These can be used to determine a pre-scaling anisotropic similarity transformation with the algorithm described in Section 2. The resulting transformation can then be used to align the test image to the model image, as shown in Fig. 5(c). The actual inspection is performed by extracting all pixels for which the images differ by more than a threshold (Steger et al., 2007, Chapter 3.4.1), performing an opening operation with a small rectangle to remove extraneous segmentation results that are caused by transforming the test image, performing a closing operation to merge closely spaced errors, and then computing connected components to isolate the errors. The detected errors are displayed in Fig. 6 by drawing a circle around the center of gravity of the detected error regions. As can be seen, all four errors have been detected correctly.

5. Conclusion

This paper has presented a closed-form solution for the problem of determining a 2D anisotropic similarity transformation from 2D point correspondences. The derivation of the algorithm shows that the problem has a unique solution that can be computed efficiently. A performance evaluation shows that the algorithm provides unbiased results.

References

Arun, K.S., Huang, T.S., Blostein, S.D., 1987. Least-squares fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 698–700.

Bankman, I.N. (Ed.), 2009. Handbook of Medical Image Processing and Analysis, second ed. Academic Press, Inc., Burlington.

Batchelor, P.G., West, J.B., Fitzpatrick, J.M., 2002. Scalings and similarity transforms: Maximum likelihood estimations. In: Houstin, A., Zwiggelaar, R. (Eds.), Medical Image Understanding and Analysis 2002. BMVA Press, pp. 129–132.

Bennani Dosse, M., Ten Berge, J., 2010. Anisotropic orthogonal Procrustes analysis. Journal of Classification 27, 111–128.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24, 381–395.

Fitzpatrick, J.M., Hill, D.L.G., Maurer, C.R., 2000. Image registration. In: Sonka, M., Fitzpatrick, J.M. (Eds.), Handbook of Medical Image Processing, Volume 2: Medical Image Processing and Analysis, SPIE Press, Bellingham, pp. 447–513.

Gilbert, G.T., 1991. Positive definite matrices and Sylvester's criterion. The American Mathematical Monthly 90, 44–46.

Gower, J.C., 2010. Procrustes methods. Wiley Interdisciplinary Reviews: Computational Statistics 2, 503–508.

Gower, J.C., Dijksterhuis, G.B., 2004. Procrustes Problems. Oxford University Press, Oxford.

Haralick, R.M., Joo, H., Lee, C.N., Zhuang, X., Vaidya, V.G., Kim, M.B., 1989. Pose estimation from corresponding point data. IEEE Transactions on Systems, Man, and Cybernetics 19, 1426–1446.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, Cambridge.

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

Horn, B.K.P., 1987. Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A 4, 629–642.

Horn, B.K.P., Hilden, H.M., Negahdaripour, S., 1988. Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America A 5, 1127–1135.

Huber, P.J., 1981. Robust Statistics. John Wiley & Sons, New York, NY.

Lamdan, Y., Schwartz, J.T., Wolfson, H.J., 1990. Affine invariant model-based object recognition. IEEE Transactions on Robotics and Automation 6, 578–589.

McGlone, J.C., Mikhail, E.M., Bethel, J. (Eds.), 2004. Manual of Protogrammetry. fifth ed. American Society for Photogrammetry and Remote Sensing, Bethesda.

Mosteller, F., Tukey, J.W., 1977. Data Analysis and Regression. Addison-Wesley, Reading, MA.

Perkins, W.A., 1983. INSPECTOR: A computer vision system that learns to inspect parts. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 584–592.

Rousseeuw, P.J., 1984. Least median of squares regression. Journal of the American Statistical Association 79, 871–880.

Steger, C. 2002. Occlusion, clutter, and illumination invariant object recognition, in: International Archives of Photogrammetry and Remote Sensing, vol. XXXIV, part 3A, pp. 345–350.

Steger, C., Ulrich, M., Wiedemann, C., 2007. Machine Vision Algorithms and Applications. Wiley-VCH, Weinheim.

Stewart, C.V., 1999. Robust parameter estimation in computer vision. SIAM Review 41, 513–537.

Strobl, K.H., Hirzinger, G., 2008. More accurate camera and hand-eye calibrations with unknown grid pattern dimensions. In: International Conference on Robotics and Automation, pp. 1398–1405.

Umeyama, S., 1991. Least-squares estimation of transformation parameters between two point patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 376–380.

Walker, M.W., Shao, L., Volz, R.A., 1991. Estimating 3D location parameters using dual number quaternions. Computer Vision, Graphics, and Image Processing: Image Understanding 54, 358–367.

Xiong, Z., Zhang, Y., 2010. A critical review of image registration methods. International Journal of Image and Data Fusion 1, 137–158.

Zitová, B., Flusser, J., 2003. Image registration methods: a survey. Image and Vision Computing 21, 977–1000.