Chapter 8
Robust Parameter Estimation

In the previous chapters, methods for detecting control points in two images of a scene and methods for determining the correspondence between the control points were discussed. In this chapter, robust methods that use the control-point correspondences to determine the parameters of a transformation function to register the images are discussed. Transformation functions for image registration will be discussed in the following chapter.

Although inaccuracies in the coordinates of corresponding points can be managed if the inaccuracies have a normal distribution with a mean of zero, but presence of even one incorrect correspondence can break down the parameter estimation process. When using image features/descriptors to find the correspondence between control points in two images, presence of noise, repeated patterns, and geometric and intensity differences between the images can result in some incorrect correspondences. Not knowing which correspondences are correct and which ones are not, the job of a robust estimator is to identify some or all of the correct correspondences and use their coordinates to determine the transformation parameters.

In the previous chapter, RANSAC, a robust estimator widely used in the computer vision community was reviewed. In this chapter, mathematically well-known robust estimators that are not widely used in computer vision and image analysis applications are reviewed. As we will see, these estimators can often replace RANSAC and sometimes outperform it.

The general problem to be addressed in this chapter is as follows. Given n corresponding points in two images of a scene:

$$\big\{(x_i, y_i), (X_i, Y_i) : i = 1, \ldots, n\big\}, \tag{8.1}$$

we would like to find the parameters of a transformation function with two components f_x and f_y that satisfy

$$\begin{aligned} X_i &\approx f_x(x_i, y_i), \\ Y_i &\approx f_y(x_i, y_i), \end{aligned} \quad i = 1, \ldots, n. \tag{8.2}$$

A.A. Goshtasby, *Image Registration*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-1-4471-2458-0_8,

If the components of the transformation are independent of each other, their parameters can be determined separately. In such a situation, it is assumed that

$$\{(x_i, y_i, F_i) : i = 1, \ldots, n\} \tag{8.3}$$

is given and it is required to find the parameters of function f to satisfy

$$F_i \approx f(x_i, y_i), \quad i = 1, \ldots, n. \tag{8.4}$$

By letting $F_i = X_i$, the estimated function will represent f_x and by letting $F_i = Y_i$, the estimated function will represent f_y. If the two components of a transformation are dependent, such as the component of a projective transformation, both components of the transformation are estimated simultaneously.

f can be considered a single-valued surface that approximates the 3-D points given by (8.3). If the points are on or near the model to be estimated, f will approximate the model closely. However, if some points are away from the model to be estimated, f may be quite different from the model. The role of a robust estimator is to find the model parameters accurately even in the presence of distant points (outliers).

We assume each component of the transformation to be determined can be represented by a linear function of its parameters. That is

$$f = \mathbf{x}^t \mathbf{a}, \tag{8.5}$$

where $\mathbf{a} = \{a_1, \ldots, a_m\}$ are the m unknown parameters of the model and $\mathbf{x}$ is a vector with m components, each a function of x and y. For instance, when f represents a component of an affine transformation, we have

$$f = a_1 x + a_2 y + a_3, \tag{8.6}$$

and so $\mathbf{x}^t = [x \ y \ 1]$ and $\mathbf{a}^t = [a_1 \ a_2 \ a_3]$. When f represents a quadratic function, we have

$$f = a_1 x^2 + a_2 y^2 + a_3 xy + a_4 x + a_5 y + a_6, \tag{8.7}$$

and so $\mathbf{x}^t = [x^2 \ y^2 \ xy \ x \ y \ 1]$ and $\mathbf{a}^t = [a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6]$.

When the observations given by (8.3) are contaminated, the estimated parameters will contain errors. Substituting (8.5) into (8.4) and rewriting it to include errors at the observations, we obtain

$$F_i = \mathbf{x}_i^t \mathbf{a} + e_i, \quad i = 1, \ldots, n, \tag{8.8}$$

where e_i is the vertical distance of F_i to the surface to be estimated at (x_i, y_i) as shown in Fig. 8.1. This is the estimated positional error in a component of the ith point in the sensed image. Not knowing which correspondences are correct and which ones are not, an estimator finds the model parameters in such a way as to minimize some measure of error between the given data and the estimated model.

In the remainder of this chapter, first the ordinary least squares (OLS) estimation is described. OLS performs well when the errors have a normal distribution. When errors have a long-tailed distribution, often caused by outliers, it performs poorly. Next, robust estimators that reduce or eliminate the influence of outliers on estimated parameters are discussed.

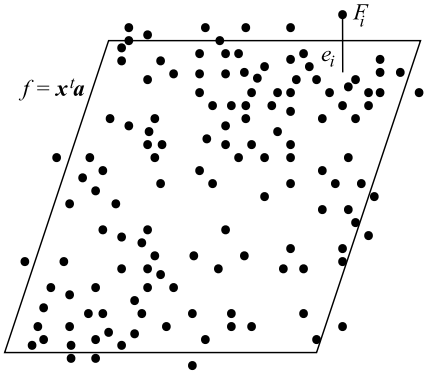

Fig. 8.1 Linear parameter estimation using contaminated data

To evaluate and compare the performances of various estimators, 100 control points detected in each of the coin images in Fig. 6.2 will be used. The coordinates of the points are shown in Table 8.1. The points in the original coin image (Fig. 6.2a) are used as the reference points and denoted by (x, y). The control points detected in the blurred, noisy, contrast-enhanced, rotated, and scaled versions of the coin image are considered sensed points and are denoted by (X, Y).

Correspondence was established between the reference point set and each of the sensed point sets by a graph-based matching algorithm with a rather large distance tolerance ($\varepsilon = 10$ pixels) to allow inaccurate and incorrect correspondences enter the process. The correspondences established between each sensed point set and the reference point set are marked with a '+' or a '−' in Table 8.1. A '+' indicates a correspondence that is correct, while a '−' indicates a correspondence that is incorrect.

The algorithm found 95, 98, 98, 96, and 78 correspondences between the coin image and its blurred, noisy, contrast-enhanced, rotated, and scaled versions, respectively. Among the obtained correspondences, only 66, 60, 68, 48, and 28 are correct. Due to the large distance tolerance used in matching, the process has picked all of the correct correspondences (true positives). However, due to the large distance tolerance, it has also picked a large number of incorrect correspondences (false positives).

Establishing correspondence between points by the closest-point criterion resulted in some reference points being assigned to two or more sensed points. Although multiple assignments are easy to detect and remove, by removing such assignments, we run the risk of eliminating some correct correspondences, something that we want to avoid. Therefore, we keep the contaminated correspondences found by our matching algorithm and use them to determine the parameters of the transformation between each sensed image and the reference image by various estimators. After finding the transformation parameters by a robust estimator, we will then separate the correct correspondences from the incorrect ones.

The parameters of the affine transformations truly relating the blurred, noisy, contrast-enhanced, rotated, and scaled images to the original image are listed in Table 8.2. Knowing the true transformation parameters between each sensed image

Table 8.1 The point sets used to evaluate the performances of various estimators. (x, y) denote the column and row numbers of control points in the reference image, and (X, Y) denote the column and row numbers of control points in a sensed image. A sensed point that is found to correctly corresponds to a reference point is marked with a '+'. The remaining points represent outliers. A sensed point marked with a '−' is a point that is incorrectly assigned to a reference point by the matching algorithm

Point #	Original		Blurred		Noisy		Enhanced		Rotated		Scaled	
	x	y	X	Y	X	Y	X	Y	X	Y	X	Y
1	5	77	5+	76+	5+	77+	5+	76+	13−	105−	5	94
2	7	84	8+	85+	7+	84+	7+	84+	25+	109+	11+	126+
3	8	41	4−	36−	9−	39−	6−	47−	5−	77−	11−	63−
4	9	61	15−	64−	12−	55−	9+	61+	15+	88+	20−	87−
5	9	100	13−	105−	15−	99−	9+	99+	34+	121+	17	139
6	12	33	12+	34+	9−	39−	12+	34+	4+	62+	18+	50+
7	12	94	13+	95+	15−	99−	12+	93+	34+	115+	17−	139−
8	13	105	13+	105+	13+	105+	9−	99−	34−	121−	28−	155−
9	16	47	4	65	16+	47+	16+	47+	15	88	24+	72+
10	18	77	6	48	14−	74−	18+	77+	34−	95−	28+	155+
11	20	23	21+	22+	21+	22+	14−	21−	6−	46−	35−	29−
12	20	87	5	93	15−	85−	20+	87+	20	14	34	161
13	21	105	21−	105−	22+	106+	21+	105+	47+	120+	28−	155−
14	24	115	28−	111−	29−	112−	28−	111−	56−	122−	36	116
15	26	67	26+	67+	26+	67+	26+	67+	32+	85+	39+	102+
16	28	16	28+	16+	27+	17+	33−	20−	9+	39+	40	72
17	28	55	25−	58−	28+	55+	28+	55+	28+	73+	40−	72−
18	28	73	26−	67−	26−	67−	26−	67−	43−	86−	39−	102−
19	29	46	33−	41−	29+	46+	29+	47+	25+	65+	40−	72−
20	30	32	30+	31+	35−	33−	36−	33−	23−	51−	47+	48+
21	32	6	33	121	32+	6+	32+	7+	9−	26−	45−	12−
22	32	21	31+	22+	33+	21+	33+	20+	15+	41+	57−	35−
23	32	114	28−	111−	29−	112−	34−	115−	56−	122−	51−	174−
24	33	121	33+	121+	33+	121+	34+	121+	72−	122−	51−	174−
25	34	101	35+	101+	34+	101+	34+	101+	56+	110+	51+	152+
26	35	85	36+	84+	35+	86+	31−	80−	49+	96+	52−	128−
27	39	16	39+	16+	33−	21−	33−	20−	18+	34+	59+	24+
28	40	49	46−	52−	40+	48+	40+	48+	35+	62+	63−	69−
29	41	62	36−	60−	41+	62+	41−	70−	47−	73−	60−	95−
30	42	105	42+	105+	41+	105+	42+	105+	69−	106−	63	69
31	42	119	42+	119+	45−	120−	41+	119+	72+	122+	62−	179−
32	43	29	42+	30+	43+	29+	43+	29+	28+	43+	67−	43−
33	44	99	44−	105−	41−	105−	42−	105−	69−	106−	69−	148−
34	46	13	47−	7−	47	72	28	111	20−	22−	71−	10−

Table 8.1 (Continued)

Point #	Original		Blurred		Noisy		Enhanced		Rotated		Scaled	
	x	y	X	Y	X	Y	X	Y	X	Y	X	Y
35	46	52	46+	52+	46+	52+	40−	48−	42+	61+	71+	10+
36	46	86	46+	85+	46+	86+	46−	86−	59+	91+	75	183
37	47	7	47+	7+	45−	1−	46−	1−	20+	22+	71−	10−
38	52	35	53+	34+	53+	34+	53+	35+	40−	50−	79−	59−
39	53	122	52+	121+	54+	122+	53−	122−	90−	121−	75−	183−
40	54	96	54+	97+	54+	96+	45	125	71+	96+	82+	144+
41	56	21	55+	21+	56+	21+	56+	21+	36−	25−	86+	33+
42	56	72	56+	72+	56+	72+	54−	68−	61+	74+	87−	107−
43	56	114	57+	114+	56+	114+	57+	115+	81−	105−	79−	172−
44	58	12	58+	12+	58+	12+	58+	12+	36−	25−	84−	24−
45	59	52	59+	52+	56−	54−	59+	52+	59−	52−	90−	70−
46	60	5	60+	5+	59+	5+	58−	12−	31+	14+	91+	7+
47	63	78	56−	72−	65−	87−	64+	78+	61−	74−	99−	122−
48	63	104	61	122	63	26	63+	104+	81−	105−	90	70
49	65	52	59−	52−	65+	52+	65+	52+	59+	52+	90	172
50	67	114	68+	114+	68+	113+	67+	114+	90−	115−	98−	178−
51	68	15	68−	15−	64−	11−	74−	12−	42+	19+	112−	18−
52	68	27	67−	21−	63−	26−	68+	27+	54−	27−	111−	42−
53	68	93	69+	93+	69+	93+	68+	93+	82+	86+	102−	148−
54	73	112	68−	114−	68−	113−	73+	113+	104−	105−	111−	169−
55	74	12	74+	12+	74+	12+	74+	12+	46−	7−	112+	18+
56	74	28	81−	22−	75−	21−	74+	28+	54+	27+	111+	42+
57	75	38	75+	38+	76+	38+	76+	38+	61+	35+	101	7
58	75	49	75+	49+	76+	49+	75−	49−	65+	44+	102	148
59	75	90	75+	90+	75+	90+	76−	89−	92−	81−	114−	131−
60	77	61	76+	61+	76+	61+	76+	61+	73+	54+	112−	92−
61	77	121	79−	120−	79−	120−	78+	121+	104+	105+	112	18
62	78	7	78+	7+	80−	7−	78−	1−	46−	7−	120−	10−
63	78	105	78+	105+	79+	106+	77−	101−	73	111	113−	155−
64	81	22	81+	22+	81+	22+	81+	22+	58−	12−	129−	34−
65	83	50	83+	50+	83+	50+	75−	49−	73+	41+	126+	75+
66	85	74	77−	73−	84+	75+	83−	81−	87+	61+	125−	114−
67	87	36	87−	36−	87+	36+	87+	36+	69+	27+	129−	54−
68	87	63	87−	62−	87+	63+	87+	63+	82+	50+	118	97
69	87	110	87+	110+	86+	111+	87+	111+	107+	92+	127	179
70	88	11	88−	11−	89+	12+	87+	11+	58+	5+	133−	15−
71	88	90	88+	90+	88+	90+	80−	94−	97+	73+	133+	134+
72	91	54	91+	53+	91+	53+	91+	53+	77−	36−	137+	81+

Table 8.1 (Continued)

Point #	Original		Blurred		Noisy		Enhanced		Rotated		Scaled	
	x	y	X	Y	X	Y	X	Y	X	Y	X	Y
73	91	121	91−	121−	91+	121+	91+	121+	114+	92+	138−	174−
74	92	6	92+	6+	92+	6+	92+	7+	58−	5−	135	72
75	92	115	93+	114+	91−	121−	92+	115+	114−	92−	138+	174+
76	93	40	88+	36+	94+	39+	93+	40+	77+	27+	146−	52−
77	94	14	98−	17−	89−	12−	94+	14+	58−	5−	130	25
78	95	84	94+	85+	95+	84+	96+	78+	106−	66−	143+	125+
79	96	65	87+	62+	99−	62−	87−	63−	93−	43−	146	52
80	96	78	96+	77+	96+	78+	96−	78−	98+	59+	146+	118+
81	97	93	97+	94+	97+	94+	97+	94+	107+	72+	147+	139+
82	101	115	105−	112−	105−	112−	105−	112−	122−	87−	152	76
83	104	14	104+	14+	104+	14+	103+	14+	78−	7−	155	30
84	104	106	103+	106+	105−	112−	105−	112−	117−	86−	157+	160+
85	106	50	106+	50+	101−	51−	106+	50+	88−	33−	152−	76−
86	106	88	106+	87+	106+	88+	106+	88+	106−	66−	158	88
87	108	70	108+	70+	108−	75−	108+	71+	105	31	157−	111−
88	109	28	111−	21−	114−	25−	111−	30−	84+	9+	163−	38−
89	111	106	111+	106+	108−	101−	105−	112−	125+	77+	171	69
90	112	96	112+	95+	112+	96+	112+	96+	121+	67+	169+	145+
91	115	37	113−	34−	121−	37−	121−	38−	95−	8−	174−	53−
92	115	68	115+	68+	115+	67+	115+	67+	116−	40−	173+	101+
93	116	57	106−	95−	121−	58−	115−	67−	105+	31+	175−	83−
94	117	86	117+	85+	116−	88−	117+	86+	120−	54−	177+	130+
95	118	44	117−	42−	121−	37−	118+	44+	100+	18+	178−	63−
96	119	79	119+	78+	123−	84−	119+	78+	120−	54−	181−	116−
97	121	37	121+	37+	121+	37+	121+	38+	95−	8−	178−	63−
98	121	70	121+	70+	121+	71+	121+	70+	116+	40+	183+	105+
99	123	84	117−	85−	123+	84+	123+	84+	120−	54−	177−	130−
100	124	45	120−	51−	121−	37−	124+	46+	100−	18−	182−	72−

and the reference image, we would like to see how accurately various estimators can find these parameters using the contaminated correspondences shown in Table 8.1

8.1 OLS Estimator

Letting x_{ij} represent the jth element of $\mathbf{x}$ when evaluated at the ith data point, relation (8.8) can be written as

Table 8.2 True linear transformation parameters between the blurred, noisy, contrast-enhanced, rotated, and scaled coin images and the original coin image

Data set	a	b	c	d	e	f
Blurred	1.000	0.000	0.000	0.000	1.000	0.000
Noisy	1.000	0.000	0.000	0.000	1.000	0.000
Enhanced	1.000	0.000	0.000	0.000	1.000	0.000
Rotated	0.866	−0.500	39.94	0.500	0.866	−23.06
Scaled	1.500	0.000	0.000	0.000	1.500	0.000

$$F_i = \sum_{j=1}^{m} x_{ij} a_j + e_i, \quad i = 1, \ldots, n. \tag{8.9}$$

e_i is positive when the given data point falls above the approximating surface, and e_i is negative when the point falls below the surface. Assuming the error at a data point is independent of errors at other data points and the errors have a Gaussian distribution, the ordinary least-squares (OLS) estimator finds the parameters of the model by minimizing the sum of squared vertical distance between the data and the estimated surface:

$$R = \sum_{i=1}^{n} r_i^2, \tag{8.10}$$

where

$$r_i = F_i - \sum_{j=1}^{m} x_{ij} a_j. \tag{8.11}$$

Vertical distance or residual r_i can be considered an estimate of the actual error e_i at the ith point. If the components of a transformation depend on each other, the squared residual at the ith point will be

$$r_i^2 = \left(X_i - \sum_{j=1}^{m_x} x_{ij} a_j \right)^2 + \left(Y_i - \sum_{j=1}^{m_y} x_{ij} b_j \right)^2, \tag{8.12}$$

where $\{a_j : j = 1, \ldots, m_x\}$ are the parameters describing the x-component of the transformation, and $\{b_j : j = 1, \ldots, m_y\}$ are the parameters describing the y-component of the transformation. When the two components of a transformation function are interdependent, some parameters appear in both components. For instance, in the case of the projective transformation, we have

$$X = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1}, \tag{8.13}$$

$$Y = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}, \tag{8.14}$$

or

$$a_7 x X + a_8 y X + X = a_1 x + a_2 y + a_3, \tag{8.15}$$

$$a_7 x Y + a_8 y Y + Y = a_4 x + a_5 y + a_6, \tag{8.16}$$

so the squared distance between the ith point and the transformation function will be

$$r_i^2 = (a_7 x_i X_i + a_8 y_i X_i + X_i - a_1 x_i - a_2 y_i - a_3)^2 + (a_7 x_i Y_i + a_8 y_i Y_i + Y_i - a_4 x_i - a_5 y_i - a_6)^2. \tag{8.17}$$

The linear parameters $a_1, \ldots, a_8$ are estimated by minimizing the sum of such squared distances or residuals.

To find the parameters that minimize the sum of squared residuals R, the gradient of R is set to 0 and the obtained system of linear equations is solved. For example, a component of an affine transformation ($m = 3$) is determined by solving

$$\begin{aligned} \frac{\partial R}{\partial a_1} &= -2 \sum_{i=1}^{n} x_i (F_i - a_1 x_i - a_2 y_i - a_3) = 0, \\ \frac{\partial R}{\partial a_2} &= -2 \sum_{i=1}^{n} y_i (F_i - a_1 x_i - a_2 y_i - a_3) = 0, \\ \frac{\partial R}{\partial a_3} &= -2 \sum_{i=1}^{n} (F_i - a_1 x_i - a_2 y_i - a_3) = 0, \end{aligned} \tag{8.18}$$

which can be written as

$$\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & n \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i F_i \\ \sum_{i=1}^n y_i F_i \\ \sum_{i=1}^n F_i \end{pmatrix}. \tag{8.19}$$

In matrix form, this can be written as

$$\mathbf{A}^t \mathbf{A} \mathbf{X} = \mathbf{A}^t \mathbf{b}, \tag{8.20}$$

where $\mathbf{A}$ is an $n \times 3$ matrix with $A_{i1} = x_i$, $A_{i2} = y_i$, and $A_{i3} = 1$; $\mathbf{b}$ is an $n \times 1$ array with $b_i = F_i$; and $\mathbf{X}$ is a 3×1 array of unknowns. Generally, when f is a function of m variables, A_{ij} represents the partial derivative of f with respect to the jth parameter when evaluated at the ith point.

We see that (8.20) is the same as left multiplying both sides of equation

$$\mathbf{A} \mathbf{X} = \mathbf{b} \tag{8.21}$$

by $\mathbf{A}^t$, and (8.21) is an overdetermined system of equations for which there isn't an exact solution. Therefore, OLS finds the solution to this overdetermined system of linear equations in such a way that the sum of squared residuals obtained at the data points becomes minimum.

If (8.20) has full rank m, its solution will be

$$\hat{\mathbf{X}} = \left(\mathbf{A}^t \mathbf{A}\right)^{-1} \mathbf{A}^t \mathbf{b}. \tag{8.22}$$

Matrix $\mathbf{A}^\dagger = (\mathbf{A}^t \mathbf{A})^{-1} \mathbf{A}^t$ is known as the pseudo-inverse of $\mathbf{A}$ [4, 22]. Therefore,

$$\hat{\mathbf{X}} = \mathbf{A}^\dagger \mathbf{b}. \tag{8.23}$$

Table 8.3 Estimated parameters by OLS for the five data sets in Table 8.1. RMSE_a indicates RMSE when using all correspondences (marked with a '+' or a '−') and RMSE_c indicates RMSE when using only the correct correspondences (marked with a '+'). The last column shows computation time in seconds when using all correspondences on a Windows PC with a 2.2 MHz processor

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	1.007	−0.004	0.676	−0.002	0.989	0.665	3.46	1.03	0.0001
Noisy	1.012	0.000	0.899	0.007	1.004	−0.652	3.56	0.88	0.0001
Enhanced	0.998	0.110	−0.353	−0.001	1.000	−0.274	3.70	0.84	0.0001
Rotated	0.872	−0.489	38.35	0.505	0.850	−22.78	4.31	0.83	0.0001
Scaled	1.501	0.017	−1.454	−0.021	1.485	2.899	5.01	1.75	0.0001

The OLS estimator was developed independently by Gauss and Legendre. Although Legendre published the idea in 1805 and Gauss published it in 1809, records show that Gauss has been using the method since 1795 [31]. It has been shown that if (1) data represent random observations from a model with linear parameters, (2) errors at the points have a normal distribution with a mean of zero, and (3) the variables are independent, then the parameters determined by OLS represent the best linear unbiased estimation (BLUE) of the model parameters [1]. Linear independence requires that the components of $\mathbf{x}$ be independent of each other. An example of dependence is x^2 and xy. This implies that when least squares is used to find parameters of functions like (8.7) with $\mathbf{x}$ containing interdependent components, the obtained parameters may not be BLUE.

Comparing the linear model with m parameters estimated by OLS with the first m principal components about the sample mean (Sect. 8.11), we see that OLS finds the model parameters by minimizing the sum of squared distances of the points to the surface vertically, while the parameters predicted by the first m principal components of the same data minimizes the sum of squared distances measured between the points and the surface in the direction normal to the surface. Although the two use the same error measure, OLS treats one dimension of the observations preferentially, while principal component analysis (PCA) treats all dimensions of observations similarly.

In addition to treating one dimension of data preferentially, OLS lacks robustness. A single outlier can drastically change the estimated parameters. The notion of *breakdown point* ε^*, introduced by Hampel [5], is the smallest fraction of outliers that can change the estimated parameters drastically. In the case of OLS, $\varepsilon^* = 1/n$.

Using the 95 points marked with '+' and '−' in Table 8.1 for the blurred image and the corresponding points in the original image, OLS estimated the six linear parameters shown in Table 8.3. The root-mean-squared error (RMSE) obtained at all correspondences and the RMSE obtained at the 66 correct correspondences are also shown. The estimated model parameters and RMSE measures between the noisy, contrast-enhanced, rotated, and scaled images and the original image are also shown in Table 8.3.

Due to the fact that the outliers are not farther than 10 pixels from the surface to be estimated, their adverse effect on the estimated parameters is limited. Since in

image registration the user can control this distance tolerance, outliers that are very far from the surface model to be estimated can be excluded from the point correspondences. Therefore, although the correspondences represent contaminated data, the maximum error an incorrect correspondence can introduce to the estimation process can be controlled. Decreasing the distance tolerance too much, however, may eliminate some of the correct correspondences, something that we want to avoid. Therefore, we would like to have the distance tolerance large enough to detect all the correct correspondences but not so large as to introduce false correspondences that can irreparably damage the estimation process.

Having contaminated data of the kind shown in Table 8.1, we would like to identify estimators that can accurately estimate the parameters of an affine transformation model and produce as small an RMSE measure as possible.

Since points with smaller residuals are more likely to represent correct correspondences than points with larger residuals, one way to reduce the estimation error is to give lower weights to points that are farther from the estimated surface. This is discussed next.

8.2 WLS Estimator

The weighted least-squares (WLS) estimator gives lower weights to points with higher square residuals. The weights are intended to reduce the influence of outliers that are far from the estimated model surface. It has been shown that OLS produces the best linear unbiased estimation of the model parameters if all residuals have the same variance [20]. It has also been shown that when the observations contain different uncertainties or variances, least-squares error is reached when the square residuals are normalized by the reciprocals of the residual variances [2]. If σ_i^2 is the variance of the ith observation, by letting $w_i = 1/\sigma_i$, we can normalize the residuals by replacing $\mathbf{x}_i$ with $w_i\mathbf{x}_i$ and f_i with $w_i f_i$. Therefore, letting $A'_{ij} = A_{ij}w_i$ and $b'_i = b_i w_i$, (8.20) converts to

$$\mathbf{A}'^t\mathbf{A}'\mathbf{X} = \mathbf{A}'^t\mathbf{b}', \tag{8.24}$$

producing the least squares solution

$$\mathbf{X} = \left(\mathbf{A}'^t\mathbf{A}'\right)^{-1}\mathbf{A}'^t\mathbf{b}'. \tag{8.25}$$

If variances at the sample points are not known, w_i is set inversely proportional to the magnitude of residual at the ith observation. That is, if

$$r_i = F_i - \mathbf{x}_i\hat{\mathbf{a}}, \quad i = 1, \ldots, n, \tag{8.26}$$

then

$$w_i = \frac{1}{|r_i| + \varepsilon}, \quad i = 1, \ldots, n. \tag{8.27}$$

ε is a small number, such as 0.01, to avoid division by zero.

Table 8.4 Estimated parameters by WLS for the five data sets in Table 8.1 and the RMSE measures

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	1.001	−0.003	−0.108	0.001	0.998	0.063	3.52	0.79	0.001
Noisy	1.000	0.000	−0.038	0.000	1.000	0.089	3.59	0.75	0.001
Enhanced	0.997	0.005	−0.132	0.000	1.000	−0.043	3.73	0.69	0.001
Rotated	0.872	−0.489	38.36	0.505	0.850	−22.79	4.33	0.83	0.001
Scaled	1.501	−0.001	−0.082	−0.001	1.507	0.134	5.15	1.06	0.001

Since the weights depend on estimated errors at the points, better weights can be obtained by improving the estimated parameters. If (8.26) represents residuals calculated using the model surface obtained by OLS and denoting the initial model by $f_0(\mathbf{x})$, the residuals at the $(k+1)$st iteration can be estimated from the model obtained at the kth iteration:

$$r_i^{(k+1)} = F_i - f_k(\mathbf{x}_i), \quad i = 1, \ldots, n. \tag{8.28}$$

The process of improving the weights and the process of improving the model parameters are interconnected. From the residuals, weights at the points are calculated, and using the weights, the model parameters are estimated. The residuals are recalculated using the refined model and the process is repeated until the sum of square weighted residuals does not decrease noticeably from one iteration to the next.

Using the data in Table 8.1 and letting $\varepsilon = 0.01$, WLS finds the model parameters shown in Table 8.4 between the blurred, noisy, contrast-enhanced, rotated, and scaled images and the original image. Only a few to several iterations were needed to obtain these parameters. The estimation errors obtained by WLS using the correct correspondences are lower than those obtained by OLS. Interestingly, the parameters and the errors obtained by OLS and WLS on the rotated data set are almost the same. Results obtained on contaminated data by WLS are not any better than those obtained by OLS.

If some information about the uncertainties of the point correspondences is available, the initial weights can be calculated using that information. This enables estimating the initial model parameters by WLS rather than by OLS and achieving a more accurate initial model. For instance, if a point in each image has an associating feature vector, the distance between the feature vectors of the ith corresponding points can be used as $|r_i|$ in (8.27). The smaller the distance between the feature vectors of corresponding points, the more likely it will be that the correspondence is correct and, thus, the smaller the correspondence uncertainty will be.

The main objective in WLS estimation is to provide a means to reduce the influence of outliers on the estimation process. Although weighted mean can reduce the influence of distant outliers on estimated parameters, it does not diminish their influence. To completely remove the influence of distant outliers on estimated parameters, rather than using the weight function of (8.27), a weight function that cuts

Table 8.5 Estimated parameters by the weighted least squares with cut-off threshold $r_0 = 2$ pixels

Data set	a	b	c	d	e	f	RMSE$_a$	RMSE$_c$	Time
Blurred	1.001	−0.005	0.026	0.001	0.996	0.295	3.52	0.78	0.001
Noisy	1.000	0.000	0.030	0.000	0.999	0.202	3.60	0.75	0.001
Enhanced	0.998	0.006	−0.276	−0.001	0.999	0.180	3.74	0.69	0.001
Rotated	0.872	−0.489	38.36	0.505	0.850	−22.79	4.33	0.83	0.001
Scaled	1.502	−0.001	−0.067	−0.002	1.507	0.357	5.15	1.03	0.001

off observations farther away than a certain distance to the estimated surface can be used. An example of a weight function with this characteristic is

$$w_i = \begin{cases} \frac{1}{|r_i|+\varepsilon} & |r_i| \leq r_0, \\ 0 & |r_i| > r_0, \end{cases} \tag{8.29}$$

where r_0 is the required distance threshold to identify and remove the distant outliers.

The WLS estimator with a cut-off of $r_0 = 2$ pixels and $\varepsilon = 0.01$ produced the model parameters shown in Table 8.5. The errors when using the correct correspondences are either the same or only slightly lower than those found by the WLS estimator without a cut-off threshold. Removing points with larger residuals does not seem to change the results significantly when using the contaminated data. If the residuals obtained with and without the cut-off threshold both have the same distribution, the same results will be produced by OLS. Because the residuals initially estimated by OLS contain errors, by removing points with high residuals or weighting them lower, the distribution of the residuals does not seem to change, resulting in the same parameters by OLS and by WLS with and without a cut-off threshold distance in this example.

8.3 M Estimator

An M estimator, like the OLS estimator, is a maximum likelihood estimator [12], but instead of minimizing the sum of squared residuals, it minimizes the sum of functions of the residuals that increases less rapidly with increasing residuals when compared with squared residuals. Consider the objective function:

$$\sum_{i=1}^{n} \rho(r_i), \tag{8.30}$$

where $\rho(r_i)$ is a function of r_i that increases less rapidly with r_i when compared with the square of r_i. To minimize this objective function, its partial derivatives with respect to the model parameters are set to 0 and the obtained system of equations is solved. Therefore,

$$\sum_{i=1}^{n} \frac{\partial \rho(r_i)}{\partial r_i} \frac{\partial r_i}{\partial a_k} = 0, \quad k = 1, \ldots, m. \tag{8.31}$$

Since $\partial r_i/\partial a_k = x_{ik}$, and denoting $\partial\rho(r_i)/\partial r_i$ by $\psi(r_i)$, we obtain

$$\sum_{i=1}^{n} \psi(r_i)x_{ik} = 0, \quad k = 1, \ldots, m. \tag{8.32}$$

The residual at the ith observation, $r_i = F_i - \sum_{j=1}^{m} x_{ij}a_j$, depends on the measurement scale, another unknown parameter. Therefore, rather than solving (8.32), we solve

$$\sum_{i=1}^{n} \psi_k\left(\frac{r_i}{\sigma}\right)x_{ik} = 0, \quad k = 1, \ldots, m, \tag{8.33}$$

for the model parameters as well as for the scale parameter σ.

The process of determining the scale parameter and the parameters of the model involves first estimating the initial model parameters by OLS and from the residuals estimating the initial scale. A robust method to estimate scale from the residuals is the *median absolute deviation* [6, 12]:

$$b\, med_i\left\{|r_i - M_n|\right\}, \tag{8.34}$$

where $M_n = med_i\{r_i\}$ for $i = 1, \ldots, n$. To make the estimated scale comparable to the spread σ of a Gaussian distribution representing the residuals, it is required that we let $b = 1.483$.

Knowing the initial scale, the model parameters are estimated from (8.33) by letting $r_i = F_i - \sum_{j=1}^{m} x_{ij}a_j$. The process of scale and parameter estimation is repeated until the objective function defined by (8.30) reaches its minimum value.

A piecewise continuous ρ that behaves like a quadratic up to a point, beyond which it behaves linearly, is [11, 12]:

$$\rho(r) = \begin{cases} r^2/2 & \text{if } |r| < c, \\ c|r| - \frac{1}{2}c^2 & \text{if } |r| \geq c. \end{cases} \tag{8.35}$$

The gradient of this function is also piecewise continuous:

$$\psi(r) = \begin{cases} r & \text{if } |r| < c, \\ c\,\text{sgn}(r) & \text{if } |r| \geq c. \end{cases} \tag{8.36}$$

$\rho(r)$ and $\psi(r)$ curves, depicted in Fig. 8.2, reduce the effect of distant outliers by switching from quadratic to linear at the threshold distance c. To achieve an asymptotic efficiency of 95%, it is required that we set $c = 1.345\sigma$ when residuals have a normal distribution with spread σ.

The gradient of the objective function, known as the *influence function*, is a linear function of the residuals or a constant in this example. Therefore, the parameters of the model can be estimated by solving a system of linear equations. Although this M estimator reduces the influence of distant outliers and produces more robust parameters than those obtained by OLS, the breakdown point of this estimator is also $\varepsilon^* = 1/n$. This is because the objective function still monotonically increases with increasing residuals and a single distant outlier can arbitrarily change the estimated parameters.

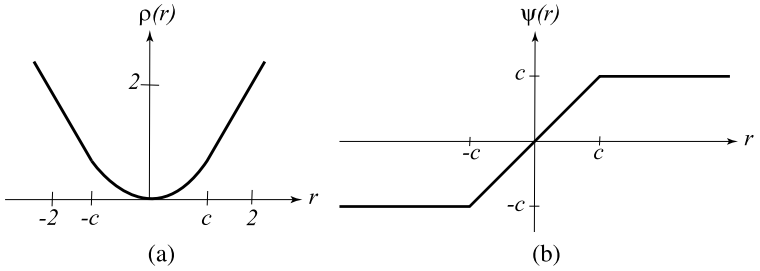

Fig. 8.2 (**a**) The plot of $\rho(r)$ curve of (8.35). (**b**) The plot of $\psi(r)$ curve of (8.36)

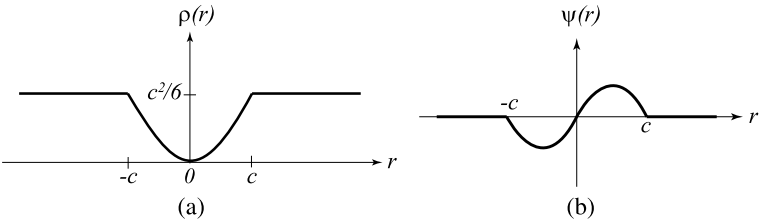

Fig. 8.3 (**a**) The plot of $\rho(r)$ of (8.37). (**b**) The plot of $\psi(r)$ of (8.38)

To further reduce the influence of outliers, consider [28]:

$$\rho(r) = \begin{cases} \frac{r^2}{2} - \frac{r^4}{2c^2} + \frac{r^6}{6c^4} & \text{if } |r| \le c, \\ \frac{c^2}{6} & \text{if } |r| > c. \end{cases} \tag{8.37}$$

This $\rho(r)$ is also a piecewise function. It is a function of degree six in r up to distance c, beyond which it changes to a constant, treating all residuals with magnitudes larger than c similarly. This estimator will, in effect, avoid distant outliers to arbitrarily change the estimated parameters. The gradient of $\rho(r)$ is:

$$\psi(r) = \begin{cases} r[1 - (\frac{r}{c})^2]^2 & \text{if } |r| \le c, \\ 0 & \text{if } |r| > c. \end{cases} \tag{8.38}$$

$\rho(r)$ and $\psi(r)$ curves are plotted in Fig. 8.3. Setting parameter $c = 4.685\sigma$, 95% asymptotic efficiency is reached when residuals have a normal distribution with spread of σ.

Note that the influence function in this example is a nonlinear function of the residuals, requiring the solution of a nonlinear system of equations to estimate the model parameters, which can be very time consuming. The objective function, by assuming a fixed value for residuals larger than a given magnitude, keeps the maximum influence an outlier can have on the estimated parameters under control. In this M estimator, a distant outlier can also adversely affect the estimated parameters, although the effect is not as damaging as the M estimator with the objective and influence curves defined by (8.35) and (8.36).

8.4 S Estimator

The scale (S) estimator makes estimation of the scale parameter σ in an M estimator the central problem [28]. An S estimator has the following properties:

1. The ρ curve in the objective function is continuously differentiable and symmetric, and it evaluates to 0 at 0 (i.e., $\rho(0) = 0$).
2. There exists an interval $[0, c]$ ($c > 0$), where ρ is monotonically increasing, and an interval (c, ∞), where ρ is a constant.
3. $$\frac{E(\rho)}{\rho(c)} = 0.5, \tag{8.39}$$

   where $E(\rho)$ is the expected value of ρ.

An example of such an estimator is [28]:

$$\rho(r) = \begin{cases} \frac{r^2}{2} - \frac{r^4}{2c^2} + \frac{r^6}{6c^4} & \text{if } |r| \le c, \\ \frac{c^2}{6} & \text{if } |r| > c, \end{cases} \tag{8.40}$$

with influence curve

$$\psi(r) = \begin{cases} r[1 - (\frac{r}{c})^2]^2 & \text{if } |r| \le c, \\ 0 & \text{if } |r| > c. \end{cases} \tag{8.41}$$

The third property is achieved in this example by letting $c = 1.547$ [28].

Given residuals $\{r_i : i = 1, \ldots, n\}$ and letting $\hat{\mathbf{a}}$ be the model parameters estimated by OLS, the scale parameter σ is estimated by solving

$$\frac{1}{n} \sum_{i=1}^{n} \rho\big(r_i(\hat{\mathbf{a}})/\hat{\sigma}\big) = K, \tag{8.42}$$

where K is the expected value of ρ. If there is more than one solution, the largest scale is taken as the solution, and if there is no solution, the scale is set to 0 [28]. Knowing scale, $\mathbf{a}$ is estimated, and the process of estimating σ and $\mathbf{a}$ is repeated until dispersion among the residuals reaches a minimum.

A robust method for estimating the initial scale is the *median absolute deviation* described by (8.34) [6, 12]. An alternative robust estimation of the scale parameter is [25]:

$$1.193\, med_i\big\{med_j\big\{|r_i - r_j|\big\}\big\}. \tag{8.43}$$

For each r_i, the median of $\{|r_i - r_j| : j = 1, \ldots, n\}$ is determined. By varying $i = 1, \ldots, n$, n numbers are obtained, the median of which will be the estimated scale. The number 1.193 is to make the estimated scale consistent with the scale σ of the Gaussian approximating the distribution of the residuals.

If ρ possesses the three properties mentioned above, the breakdown point of the S estimator will be [28]:

$$\varepsilon^* = \frac{1}{n}\left(\left\lfloor \frac{n}{2} \right\rfloor - m + 2\right). \tag{8.44}$$

As n approaches ∞, the breakdown point of the S estimator approaches 0.5. This high breakdown point of the S estimator is due to the second property of the ρ curve that is required to have a constant value beyond a certain point. This will stop a single outlier from influencing the outcome arbitrarily. Note that although an outlier in the S estimator is not as damaging as it can be, an outlier still adversely affects the estimated parameters and as the number of outliers increases, the estimations worsen up to the breakdown point, beyond which there will be a drastic change in the estimated parameters.

To summarize, an S estimator first determines the residuals using OLS or a more robust estimator. Then the scale parameter is estimated using the residuals. Knowing an estimation $\hat{\sigma}$ to the scale parameter, r_i is replaced with $r_i/\hat{\sigma}$ and the influence function is solved for the parameters of the model. Note that this requires the solution of a system of nonlinear equations. Having the estimated model parameters $\hat{\mathbf{a}}$, the process of finding the residuals, estimating the scale, and estimating the model parameters is repeated until a minimum is reached in the estimated scale, showing minimum dispersion of the obtained residuals.

8.5 RM Estimator

The repeated median (RM) estimator works with the median of the parameters estimated by different combinations of m points out of n [32]. If there are n points and m model parameters, there will be overall $n!/[m!(n-m)!]$ or $O(n^m)$ combinations of points that can be used to estimate the model parameters.

Now consider the following median operator:

$$M\{\tilde{\mathbf{a}}(i_1,\ldots,i_m)\} = med_{i_m}\{\tilde{\mathbf{a}}(i_1,\ldots,i_{m-1},i_m)\}, \tag{8.45}$$

where the right-hand side is the median of parameters $\tilde{\mathbf{a}}(i_1,\ldots,i_{m-1},i_m)$ as point i_m is replaced with all points not already among the m points. Every time the operator is called, it replaces one of its m points with all points not already in use. By calling the operator m times, each time replacing one of its points, the median parameters for all combinations of m points out of n will be obtained. The obtained median parameters are taken as the parameters of the model.

$$\hat{\mathbf{a}} = M^m\{\tilde{\mathbf{a}}(i_1,\ldots,i_m)\}, \tag{8.46}$$

$$= med_{i_1}\big(\cdots\big(med_{i_m-1}\big(med_{i_m}\tilde{\mathbf{a}}(i_1,\ldots,i_m)\big)\big)\cdots\big). \tag{8.47}$$

The process of estimating the model parameters can be considered m nested loops, where each loop goes through the n points except for the ones already in use by the outer loops and determines the parameters of the model for each combination of m points. The median of each parameter is used as the best estimate of that parameter.

When n is very large, an exhaustive search for the optimal parameters will become prohibitively time consuming, especially when m is also large. To reduce computation time without significantly affecting the outcome, only point combinations that are sufficiently far from each other in the (x, y) domain is used. Points distant

Table 8.6 The parameters estimated by the RM estimator along with RMSE measures and computation time for the five data sets in Table 8.1

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	1.000	0.000	0.000	0.000	1.000	0.000	3.57	0.79	133
Noisy	1.000	0.000	0.000	0.000	1.000	0.000	3.50	0.75	162
Enhanced	1.000	0.000	0.000	0.000	1.000	0.000	3.75	0.69	164
Rotated	0.871	−0.485	38.68	0.501	0.853	−22.70	4.32	0.79	144
Scaled	1.504	0.008	−0.049	−0.014	1.496	1.964	5.13	1.30	41

Table 8.7 Results obtained by the fast version of the RM estimator using only the convex-hull points in parameter estimation

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	0.999	0.000	0.000	0.000	1.000	0.000	3.38	0.83	0.035
Noisy	1.000	0.000	0.000	0.009	1.000	0.000	3.63	0.84	0.021
Enhanced	0.972	0.005	1.558	0.008	1.000	−0.497	3.65	1.10	0.009
Rotated	0.809	−0.485	41.64	0.507	0.845	−22.71	5.27	2.17	0.028
Scaled	1.458	0.034	1.712	0.003	1.474	0.039	4.91	2.90	0.011

from each other result in more accurate parameters as they are less influenced by small positional errors. For instance, points describing the convex hull of the points can be used. By discarding points inside the convex hull of the points, considerable savings can be achieved.

To evaluate the performance of the RM estimator on the data sets in Table 8.1 when using the full combination of 3 correspondences out of the marked correspondences in the table, the parameters listed in Table 8.6 are obtained. The RMSE measures and computation time required to find the parameters for each set are also shown.

The results obtained by the fast version of the RM estimator, which uses only the convex hull points in the reference image and the corresponding points are shown in Table 8.7. The fast RM estimator achieves a speed up factor of more than 1000 by introducing only small errors into the estimated parameters. The difference between the two is expected to reduce further with increasing n.

Although the RM estimator has a theoretical breakdown point of 0.5, we see that in the scaled data set there are only 28 true correspondences from among the 78 marked correspondences in Table 8.1, showing that more than half of the correspondences are incorrect. However, since all residuals are within 10 pixels, the RM estimator has been able to estimate the parameters of the model.

Table 8.8 The parameters estimated by the LMS estimator using the data sets in Table 8.1

Data set	a	b	c	d	e	f	RMSE$_a$	RMSE$_c$	Time
Blurred	1.000	−0.003	−0.097	−0.001	0.996	0.319	3.52	0.79	0.004
Noisy	1.012	0.000	−0.889	0.007	1.004	−0.562	3.56	0.88	0.003
Enhanced	0.997	0.11	−0.353	−0.001	1.001	−0.274	3.71	0.84	0.001
Rotated	0.869	−0.499	39.32	0.502	0.860	−23.54	4.37	0.58	0.001
Scaled	1.507	−0.007	−0.015	−0.005	1.509	0.612	5.18	1.02	0.001

8.6 LMS Estimator

The least median of squares (LMS) estimator finds the model parameters by minimizing the median of squared residuals [24]:

$$\min_{\hat{\mathbf{a}}} \left\{ med_i \left(r_i^2 \right) \right\}. \tag{8.48}$$

When the residuals have a normal distribution with a mean of zero and when two or more parameters are to be estimated ($m \geq 2$), the breakdown point of the LMS estimator is [24]:

$$\varepsilon^* = \frac{1}{n} \left(\left\lfloor \frac{n}{2} \right\rfloor - m + 2 \right). \tag{8.49}$$

As n approaches ∞, the breakdown point of the estimator approaches 0.5.

By minimizing the median of squares, the process, in effect, minimizes the sum of squares of the smallest $\lfloor n/2 \rfloor$ absolute residuals. Therefore, first, the parameters of the model are estimated by OLS or a more robust estimator. Then, points that produce the $\lfloor n/2 \rfloor$ smallest magnitude residuals are identified and used in OLS to estimate the parameters of the model. The process is repeated until the median of squared residuals reaches a minimum.

Using the data sets shown in Table 8.1, the results in Table 8.8 are obtained. The process in each case takes from a few to several iterations to find the parameters. The LMS estimator has been able to find parameters between the transformed images and the original image that are as close to the ideal parameters as the parameters estimated by any of the estimators discussed so far.

8.7 LTS Estimator

The least trimmed squares (LTS) estimator [26] is similar to the LMS estimator except that it uses fewer than half of the smallest squared residuals to estimate the parameters. LTS estimates the parameters by minimizing

$$\sum_{i=1}^{h} \left(r^2 \right)_{i:n}, \tag{8.50}$$

Table 8.9 Parameters estimated by the LTS estimator with $h = n/4$ using the data sets in Table 8.1

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	1.000	0.000	0.000	0.000	1.000	0.000	3.57	0.79	0.002
Noisy	1.000	0.000	0.000	0.000	1.000	0.000	3.60	0.75	0.001
Enhanced	1.000	0.000	0.000	0.000	1.000	0.000	3.75	0.69	0.002
Rotated	0.873	−0.496	38.90	0.503	0.857	−23.18	4.35	0.65	0.001
Scaled	1.510	−0.002	−0.579	−0.009	1.505	0.932	5.12	1.08	0.002

where $m \leq h \leq n/2 + 1$ and $(r^2)_{i:n} \leq (r^2)_{j:n}$, when $i < j$. The process initially estimates the parameters of the model by OLS or a more robust estimator. It then orders the residuals and identifies points that produce the h smallest residuals. Those points are then used to estimate the parameters of the model. The squared residuals are recalculated using all points and ordered. The process of selecting points and calculating and ordering the residuals is repeated. The parameters obtained from the points producing the h smallest residuals are taken as estimates to the model parameters in each iteration. The process is stopped when the hth smallest squared residual reaches a minimum.

The breakdown point of the LTS estimator is [26]:

$$\varepsilon^* = \begin{cases} (h - m + 1)/n & \text{if } m \leq h < \lfloor \frac{n+m+1}{2} \rfloor, \\ (n - h + 1)/n & \text{if } \lfloor \frac{n+m+1}{2} \rfloor \leq h \leq n. \end{cases} \tag{8.51}$$

When n is not very large and if the number of parameters m is small, by letting $h = n/2 + 1$ we see that the breakdown point of this estimator is close to 0.5. When n is very large, by letting $h = n/2$, we see that irrespective of m a breakdown point close to 0.5 is achieved. Note that due to the ordering need in the objective function, each iteration of the algorithm requires $O(n \log_2 n)$ comparisons.

By letting $h = n/4$ and using the data in Table 8.1, we obtain the results shown in Table 8.9. Obtained results are similar to those obtained by the LMS estimator when using all the correspondences. When using only the correct correspondences, results obtained by the LTS estimator are slightly better than those obtained by the LMS estimator.

When the ratio of correct correspondences over all correspondences falls below 0.5, the parameters initially estimated by OLS may not be accurate enough to produce squared residuals that when ordered will place correct correspondences before the incorrect ones. Therefore, the obtained ordered list may contain a mixture of correct and incorrect correspondences from the very start. When the majority of correspondences is correct and there are no distant outliers, the residuals are ordered such that more correct correspondences appear at and near the beginning of the list. This enables points with smaller squared residuals to be selected, allowing

more correct correspondences to participate in the estimation process, ultimately producing more accurate results.

8.8 R Estimator

A rank (R) estimator ranks the residuals and uses the ranks to estimate the model parameters [13]. By using the ranks of the residuals rather than their actual values, the influence of very distant outliers is reduced. By assigning weights to the residuals through a scoring function, the breakdown point of the estimator can be increased up to 0.5. Using a fraction α of the residuals in estimating the parameters of the model, Hossjer [9] reduced the influence of the $1-\alpha$ largest magnitude residuals in parameter estimation. It is shown that a breakdown point of 0.5 can be achieved by letting $\alpha = 0.5$.

If R_i is the rank of the ith largest magnitude residual $|r_i|$ from among n residuals and if $b_n(R_i)$ is the score assigned to the ith largest magnitude residual from a score generating function, then the objective function to minimize is

$$\frac{1}{n}\sum_{i=1}^{n} b_n(R_i) r_i^2, \tag{8.52}$$

which can be achieved by setting its gradient to zero and solving the obtained system of linear equations. Therefore,

$$\sum_{i=1}^{n} b_n(R_i) r_i x_{ik} = 0, \quad k = 1, \ldots, m. \tag{8.53}$$

This is, in effect, a WLS estimator where the weight of the residual at the i point is $b_n(R_i)$.

Given ranks $\{R_i : i = 1, \ldots, n\}$, an example of a score generating function is

$$b_n(R_i) = h\big(R_i/(n+1)\big), \tag{8.54}$$

which maps the ranks to (0, 1) in such a way that

$$\sup\big\{u; h(u) > \alpha\big\} = \alpha, \quad 0 < \alpha \leq 1. \tag{8.55}$$

For example, if $\alpha = 0.25$ and letting $u = R_i/(n+1)$, then when $R_i/(n+1) \leq \alpha$ the score is u, and when $R_i/(n+1) > \alpha$ the score is 0.25. This scoring function, in effect, assigns a fixed weight to a certain percentage of highest magnitude residuals. Therefore, when $\alpha = 0.25$, the highest 75% residuals are given a fixed weight that is lower than what they would otherwise receive. The scoring function can be designed to assign decreasing scores to increasing residuals from a point and to assign a score

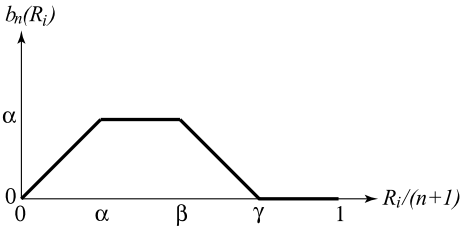

Fig. 8.4 Plot of the scoring function of (8.56)

of 0 to a percentage of the largest magnitude residuals. For example, consider the scoring function depicted in Fig. 8.4 with $0 < \alpha \leq \beta \leq \gamma \leq 1$,

$$b_n(R_i) = \begin{cases} R_i/(n+1), & \text{if } R_i/(n+1) \leq \alpha, \\ \alpha, & \text{if } \alpha < R_i/(n+1) \leq \beta, \\ \alpha[\gamma - R_i/(n+1)]/(\gamma - \beta), & \text{if } \beta < R_i/(n+1) \geq \gamma, \\ 0, & \text{if } R_i/(n+1) > \gamma. \end{cases} \tag{8.56}$$

This scoring function discards the 100γ percentage of the points that produce the largest magnitude residuals. By discarding such points, the process removes the outliers. Hössjer [9] has shown that if the scoring function is nondecreasing, the process has a single global minimum. However, if the scoring function decreases in an interval, there may be more than one minima, and if the initial parameters estimated by OLS are not near the final parameters, the R estimator may converge to a local minimum rather than the global one.

To summarize, estimation by an R estimator involves the following steps:

1. Design a scoring function.
2. Estimate the model parameters by OLS or a more robust estimator and calculate the residuals.
3. Let initial weights at all points be $1/n$.
4. Rank the points according to the magnitude of the weighted residuals.
5. Find the score at each point using the scoring function, and let the score represent the weight at the point.
6. Find the model parameters by the WLS estimator.
7. Estimate the new residuals at the points. If a minimum is reached in the sum of weighted square residuals, stop. Otherwise, go to Step 4.

Using the nondecreasing scoring function in (8.54), the results shown in Table 8.10 are obtained for the data sets in Table 8.1. Using the scoring function (8.56) with $\alpha = 0.5$, $\beta = 0.75$, and $\gamma = 1.0$, the results shown in Table 8.11 are obtained for the same data sets.

Similar results are obtained by the two scoring functions. Comparing these results with those obtained by previous estimators, we see that the results by the R estimator are not as good as those obtained by some of the other estimators when using the data sets in Table 8.1. By using the ranks of the residuals rather than their magnitudes, the process reduces the influence of distant outliers. The process, however, may assign large ranks to very small residuals in cases where a great portion of

Table 8.10 Parameter estimation by the R estimator when using the scoring function of (8.54) with $\alpha = 0.5$ and the data sets in Table 8.1

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	1.000	0.002	−0.256	−0.006	1.000	0.036	3.55	0.95	0.001
Noisy	1.010	−0.004	−0.120	0.005	0.992	−0.044	3.61	0.92	0.001
Enhanced	0.996	0.003	0.038	−0.002	0.994	0.057	3.74	0.91	0.001
Rotated	0.872	−0.489	38.36	0.505	0.850	−22.79	4.33	0.83	0.001
Scaled	1.497	0.002	−0.249	−0.013	1.5001	0.604	5.13	1.59	0.001

Table 8.11 Parameter estimation by the R estimator when using the scoring function of (8.56) with $\alpha = 0.5$, $\beta = 0.75$, and $\gamma = 1.0$ and the data sets in Table 8.1

Data set	a	b	c	d	e	f	RMSE_a	RMSE_c	Time
Blurred	0.996	0.007	−0.220	−0.003	0.995	0.055	3.58	1.01	0.001
Noisy	1.009	−0.007	−0.053	0.003	0.994	−0.033	3.60	0.89	0.001
Enhanced	0.987	0.008	0.143	0.000	0.999	−0.070	3.75	0.90	0.001
Rotated	0.872	−0.489	38.36	0.505	0.850	−22.79	4.33	0.83	0.001
Scaled	1.484	0.012	−0.109	−0.007	1.500	0.438	5.13	1.67	0.001

the residuals are very small. This, in effect, degrades the estimation accuracy. Therefore, in the absence of distant outliers, as is the case for the data sets in Table 8.1, the R estimator does not produce results as accurate as those obtained by LMS and LTS estimators.

8.9 Effect of Distant Outliers on Estimation

If a correspondence algorithm does not have the ability to distinguish inaccurate correspondences from incorrect ones, some incorrect correspondences (outliers) may take part in estimation of the model parameters. In such a situation, the results produced by different estimators will be different from the results presented so far. To get an idea of the kind of results one may get from the various estimators in the presence of distant outliers, the following experiment is carried out.

The 28 correct corresponding points in the original and scaled images marked with '+' in Table 8.1 are taken. These correspondences are connected with yellow lines in Fig. 8.5a. In this correspondence set, points in the original set are kept fixed and points in the scaled set are switched one at a time until the breakdown point for each estimator is reached. To ensure that the outliers are far from the estimating model, the farthest points in the scaled set are switched. The correct correspondences, along with the outliers tested in this experiment, are shown in Figs. 8.5b–i. Red lines connect the incorrect correspondences and yellow lines connect the correct correspondences. Using point correspondences connected with yellow and red

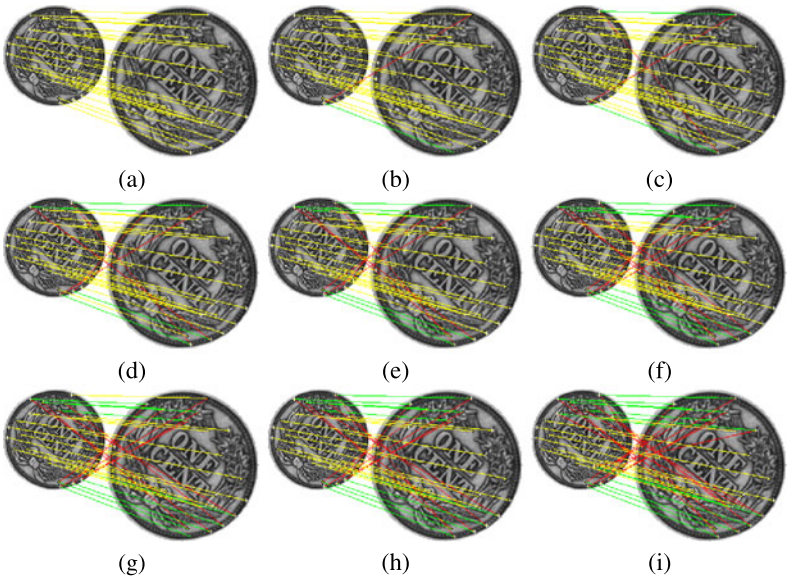

Fig. 8.5 (**a**) 28 corresponding points in the coin image and its scaled version. (**b**)–(**i**) Introduction of 1, 2, 4, 6, 8, 9, 10, and 14 outliers into the correspondence set of (**a**). *Red lines* are the outliers (false positives) and *green lines* are the missed correspondences (false negatives). The *yellow lines* are the correct correspondences (true positives). The *points* connected with the *yellow* and *red lines* are used as corresponding points in the experiments

lines, the results shown in Table 8.12 are obtained by the various estimators. The green lines indicate the correct correspondences that are not used in the estimation process.

From the results in Table 8.12, we can conclude the following:

1. For the data set in Fig. 8.5a where no outliers are present and data are simply corrupted with random noise, OLS performs as good as any other estimator by finding the maximum likelihood estimation of the parameters.
2. Because OLS can break down with a single distant outlier, the estimators that depend on OLS to find the initial residuals or initial parameters can also break down with a single distant outlier. WLS and R-1 estimators have exhibited this characteristic when using the data sets containing one or more outliers.
3. To improve the accuracy of the estimators, a means to either eliminate some of the distant outliers, as done by R-2, or to estimate the initial model parameters more robustly is required.
4. When using the data sets in Fig. 8.5, the clear winner is the R-2 estimator, which uses the scoring function in (8.56). By effectively removing some of the outliers, ordering the rest, and using points with low squared residuals, this estimator has been able to find correct model parameters from data containing up to 50% of distant outliers (Fig. 8.5i). LTS with $h = n/4$ and LMS have also been able to perform well under distant outliers.

Table 8.12 Breakdown points for various estimators in the presence of distant outliers. Table entries show RMSE at the correct correspondences. The point at which a sharp increase in RMSE is observed while gradually increasing the number of outliers is the breakdown point. WLS-1 and WLS-2 imply WLS estimation without and with a cut-off threshold of 2 pixels, RM-1 and RM-2 imply the regular and the fast RM estimators, and R-1 and R-2 imply the R estimator with the non-decreasing scoring function of (8.54) with $\alpha = 0.5$ and the decreasing scoring function of (8.56) with $\alpha = 0.25$, $\beta = 0.5$, and $\gamma = 0.75$, respectively. The numbers in the top row show the number of distant outliers used in a set of 28 corresponding points

Estimator	0	1	2	4	6	8	9	10	14
OLS	0.97	11.14	20.31	33.01	42.53	48.04	50.73	50.86	51.75
WLS-1	0.97	11.14	20.31	33.01	42.53	48.04	50.73	50.86	106.2
WLS-2	0.97	11.14	20.31	33.01	42.53	48.04	50.73	50.86	51.75
RM-1	0.98	1.01	1.06	1.19	5.88	67.07	47.46	47.63	58.30
RM-2	1.15	0.56	1.06	44.04	44.04	59.51	54.74	50.06	45.27
LMS	1.01	1.10	1.20	1.09	1.18	1.05	50.89	50.86	59.16
LTS	1.01	1.36	1.39	1.28	1.17	1.20	1.14	55.65	51.75
R-1	1.02	15.95	22.58	42.98	53.26	52.06	70.52	67.40	84.85
R-2	1.01	1.04	1.07	1.25	1.10	1.10	1.07	1.11	1.21

8.10 Additional Observations

For the data sets in Table 8.1, all tested estimators were able to find the parameters of the affine transformation to register the images with acceptable accuracies. These data sets do not contain distant outliers and errors at the points have distributions that are close to normal with a mean of 0. Among the estimators tested, RM, LMS, and LTS estimators produce the highest accuracies. Considering the high computational requirement of RM estimator, LMS and LTS stand out among the others in overall speed and accuracy in estimating model parameters when using the data sets of the kind shown in Table 8.1.

For the data sets of the kind depicted in Fig. 8.5, where distant outliers are present, results in Table 8.12 show that R estimator with the scoring function given in (8.56) is the most robust among the estimators tested, followed by LTS and LMS estimators. The OLS and WLS estimators are not to be used when the provided data contains distant outliers.

Although some estimators performed better than others on the limited tests performed in this chapter, it should be mentioned that one may be able to find a data set where any of the estimators can perform better than many of the other estimators. When data sets represent coordinates of corresponding points obtained by a point pattern matching algorithm, it is anticipated that the R-2 estimator will perform better than others when distant outliers are present, and LTS and LMS estimators will perform better than other estimators when the correspondences do not contain distant outliers.

When the ratio of outliers and inliers is small and the outliers are distant from the model, methods to remove the outliers have been developed. Hodge and Austin [8]

provided a survey of such methods. Outlier detection, however, without information about the underlying model is not always possible especially when the number of outliers is nearly the same as the number of inliers, or when outliers are not very far from the model to be estimated. Robust estimators coupled with the geometric constraint that hold between images of a scene can determine model parameters in the presence of a large number of outliers and without use of outlier detection methods.

The list of estimators discussed in this chapter is by no means exhaustive. For a more complete list of estimators, the reader is referred to excellent monographs by Andrews et al. [3], Huber [12], Hampel et al. [7], Rousseeuw and Leroy [27], and Wilcox [36].

8.11 Principal Component Analysis (PCA)

Suppose feature vector $\mathbf{x} = \{x_0, x_1, \ldots, x_{N-1}\}$ represents an observation from a phenomenon and there are m such observations: $\{\mathbf{x}^i : i = 0, \ldots, m-1\}$. We would like to determine an $N \times N$ matrix $\mathbf{A}$ that can transform $\mathbf{x}$ to a new feature vector $\mathbf{y} = \mathbf{A}^t\mathbf{x}$ that has a small number of high-valued components. Such a transformation makes it possible to reduce the dimensionality of $\mathbf{x}$ while maintaining its overall variation.

Assuming each feature is normalized to have mean of 0 and a fixed scale, such as 1, then the expected value of $\mathbf{y}\mathbf{y}^t$ can be computed from

$$\begin{aligned} E(\mathbf{y}\mathbf{y}^t) &= E(\mathbf{A}^t\mathbf{x}\mathbf{x}^t\mathbf{A}) \\ &= \mathbf{A}^t E(\mathbf{x}\mathbf{x}^t)\mathbf{A} \\ &= \mathbf{A}^t \Sigma_x \mathbf{A} \end{aligned} \tag{8.57}$$

where

$$\Sigma_x = \begin{bmatrix} E(x_0x_0) & E(x_0x_1) & \ldots & E(x_0x_{N-1}) \\ E(x_1x_0) & E(x_1x_1) & \ldots & E(x_1x_{N-1}) \\ \cdot & \cdot & \ldots & \cdot \\ E(x_{N-1}x_0) & E(x_{N-1}x_1) & \ldots & E(N-1x_{N-1}) \end{bmatrix} \tag{8.58}$$

is the covariance matrix with its ijth entry computed from

$$E(x_ix_j) = \frac{1}{m}\sum_{k=0}^{m-1}(x_i^k x_j^k). \tag{8.59}$$

By letting the eigenvectors of Σ_x represent the columns of $\mathbf{A}$, $\mathbf{A}^t\Sigma_x\mathbf{A}$ will become a diagonal matrix with diagonal entries showing the eigenvalues of Σ_x.

Suppose the eigenvalues of Σ_x are ordered so that $\lambda_i \geq \lambda_{i+1}$ for $0 \leq i < N-1$ and eigenvectors corresponding to the eigenvalues are $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{N-1}$, we can then write

$$y_i = \mathbf{v}_i^t\mathbf{x}, \quad i = 0, \ldots, N-1. \tag{8.60}$$

If transformed features are known, the original features can be computed from

$$\mathbf{x} = \sum_{i=0}^{N-1} y_i \mathbf{v}_i . \tag{8.61}$$

An approximation to $\mathbf{x}$ using eigenvectors of Σ_x corresponding to its n largest eigenvalues is obtained from

$$\hat{\mathbf{x}} = \sum_{i=0}^{n-1} y_i \mathbf{v}_i . \tag{8.62}$$

Squared error in this approximation will be [23, 34]

$$\begin{aligned} E\left(\|\mathbf{x} - \hat{\mathbf{x}}\|^2\right) &= \sum_{i=n}^{N-1} \mathbf{v}_i^t \lambda_i \mathbf{v}_i \\ &= \sum_{i=n}^{N-1} \lambda_i \end{aligned} \tag{8.63}$$

for using $y_0, y_1, \ldots, y_{n-1}$ instead of $x_0, x_1, \ldots, x_{N-1}$.

Since the eigenvalues depend on the scale of features, the ratio measure [23]

$$r_n = \sum_{i=n}^{N-1} \lambda_i \Big/ \sum_{i=0}^{N-1} \lambda_i \tag{8.64}$$

may be used as a scale-independent error measure to select the number of principal components needed to achieve a required squared error tolerance in approximation.

To summarize, following are the steps to reduce the dimensionality of feature vector $\mathbf{x}$ from N to $n < N$ using a training data set containing m observations:

1. Estimate Σ_x from the m observations.
2. Find eigenvalues and eigenvectors of Σ_x. Order the eigenvalues from the largest to the smallest: $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1}$. Note that eigenvalue λ_i has an associating eigenvector, $\mathbf{v}_i$.
3. Find the largest n such that $\sum_{i=n}^{N-1} \lambda_i < \varepsilon$, where ε is the required squared error tolerance.
4. Given a newly observed feature vector $\mathbf{x}$, project $\mathbf{x}$ to the n-dimensions defined by the eigenvectors corresponding to the n largest eigenvalues of Σ_x. That is compute $y_i = \mathbf{v}_i^t \mathbf{x}$ for $i = 0, \ldots, n-1$. $\mathbf{y}$ represents a point in $n < N$ dimensions, thereby, reducing the dimensionality of $\mathbf{x}$ while ensuring the squared approximation error stays below the required tolerance.

PCA was first used by Pearson [21] to find the best-fit line or plane to high dimensional points. The best-fit line or plane was found to show the direction of most uncorrelated variation. Therefore, PCA transforms correlated values into uncorrelated values, called principal components. The components represent the direction of most uncorrelated variation, the direction of second most uncorrelated variation, and so on.

PCA is also called Karhunen–Loève (K–L) transform and Hotelling transform. Given a feature vector containing N features, in an attempt to create $n < N$ new features that carry about the same variance from the linear combinations of the features, Hotelling [10] (also see [16, 17]) found the linear coefficients relating the original features to new ones in such a way that the first new feature had the largest variance. Then, the second feature was created in such a way that it was uncorrelated with the first and had as large a variance as possible. He continued the process until n new features were created. The coefficients of the linear functions defining a new feature in terms of the original features transform the original features to the new ones.

Rao [23] provided various insights into the uses and extensions of PCA. Watanabe [35] showed that dimensionality reduction by PCA minimizes average classification error when taking only a finite number of coefficients in a series expansion of a feature vector in terms of orthogonal basis vectors. He also showed that PCA minimizes the entropy of average square coefficients of the principal components. These two characteristics make PCA a very efficient tool for data reduction. The dimensionality reduction power of PCA using artificial and real data has been demonstrated by Kittler and Young [18]. For a thorough treatment of PCA and its various applications, see the excellent monograph by Jolliffe [16].

Since PCA calculates a new feature using all original features, it still requires high-dimensional data collection. It would be desirable to reduce the number of original features while preserving sufficient variance in collected features without changing the number of principal components. Jolliffe [14, 15] suggested discarding features that contributed greatly to the last few principal components, or selecting features that contributed greatly to the first few principal components. Therefore, if

$$\mathbf{y} = \mathbf{A}^t \mathbf{x}, \tag{8.65}$$

or

$$y_i = \sum_{j=0}^{N-1} A_{ji} x_j, \quad i = 0, \ldots, N-1, \tag{8.66}$$

where A_{ji} denotes the entry at column i and row j in matrix $\mathbf{A}$, then magnitude of A_{ji} determines the contribution of x_j to y_i.

Since this method finds ineffective features in the original set by focusing on the principal components one at a time, the influence of an original feature on a number of principal components is not taken into consideration. Mao [19] suggested finding the contribution of an original feature on all selected principal components. The significance of an original feature on the selected n principal components is determined by calculating the squared error in (8.63) once using all features and another time using all features except the feature under consideration. The feature producing the least increase in error is then removed from the original set and the process is repeated until the squared error among the remaining features reaches a desired tolerance.

Since each transformed feature in PCA is a linear combination of the original features, the process detects only linear dependency between features. If dependency between features is nonlinear, nonlinear approaches [29, 30, 33] should be used to reduce the number of features.

References

1. Abdi, H.: Least squares. In: Lewis-Beck, M., Bryman, A., Futing, T. (eds.) The Sage Encyclopedia of Social Sciences Research Methods, Thousand Oaks, CA, pp. 1–4 (2003)
2. Aitken, A.C.: On least squares and linear combinations of observations. Proc. R. Soc. Edinb. **55**, 42–48 (1935)
3. Andrews, D.F., Bickel, P.J., Hampel, F.R., Huber, P.J., Rogers, W.H., Tukey, J.W.: Robust Estimates of Location: Survey and Advances. Princeton University Press, Princeton (1972)
4. Golub, G., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. J. SIAM Numer. Anal., Ser. B **2**(2), 205–224 (1965)
5. Hampel, F.R.: A general qualitative definition of robustness. Ann. Math. Stat. **42**(6), 1887–1896 (1971)
6. Hampel, F.R.: The influence curve and its role in robust estimation. J. Am. Stat. Assoc. **69**(346), 383–393 (1974)
7. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A.: Robust Statistics: The Approach Based on Influence Functions. Wiley, New York (1986)
8. Hodges, V.J., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 85–126 (2004)
9. Hossjer, P.: Rank-based estimates in the linear model with high breakdown point. J. Am. Stat. Assoc. **89**(425), 149–158 (1994)
10. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**, 417–441 (1933), also see pp. 498–520
11. Huber, P.J.: Robust regression: Asymptotics, conjectures and Monte Carlo. Ann. Stat. **1**(5), 799–821 (1973)
12. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
13. Jaeckel, L.A.: Regression coefficients by minimizing the dispersion of the residuals. Ann. Math. Stat. **43**(5), 1449–1458 (1972)
14. Jolliffe, I.T.: Discarding variables in a principal component analysis. I: Artificial data. J. R. Stat. Soc., Ser. C, Appl. Stat. **21**(2), 160–173 (1972)
15. Jolliffe, I.T.: Discarding variables in a principal component analysis. II: Real data. J. R. Stat. Soc., Ser. C, Appl. Stat. **22**(1), 21–31 (1973)
16. Jolliffee, I.T.: Principal Component Analysis. Springer, New York (2002)
17. Kendall, M.G.: A Course in Multivariate Analysis, 4th Impression. Hafner, New York (1968)
18. Kittler, J., Young, P.C.: A new approach to feature selection based on the Karhunen–Loève expansion. Pattern Recognit. **5**, 335–352 (1973)
19. Mao, K.Z.: Identifying critical variables of principal components for unsupervised feature selection. IEEE Trans. Syst. Man Cybern., Part B, Cybern. **35**(2), 334–339 (2005)
20. McElroy, F.W.: A necessary and sufficient condition that ordinary least-squares estimators be best linear unbiased. J. Am. Stat. Assoc. **62**(320), 1302–1304 (1967)
21. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Mag. **2**(6), 559–572 (1901)
22. Penrose, R.: A generalized inverse for matrices. Math. Proc. Camb. Philos. Soc. **51**(3), 406–413 (1955)
23. Rao, C.R.: The use of interpretation of principal component analysis in applied research. Indian J. Stat., Ser. A **26**(4), 329–358 (1964)
24. Rousseeuw, P.J.: Least median of squares regression. J. Am. Stat. Assoc. **79**(388), 871–880 (1984)
25. Rousseeuw, P.J., Croux, C.: Alternatives to the median absolute deviation. J. Am. Stat. Assoc. **88**(424), 1273–1283 (1993)
26. Rousseeuw, P.J., Hubert, M.: Recent developments in PROGRESS. In: Lecture Notes on L_1-Statistical Procedures and Related Topics, vol. 31, pp. 201–214 (1997)
27. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley, New York (1987)

28. Rousseeuw, P., Yohai, V.: Robust regression by means of S-estimators. In: Franke, J., Hördle, W., Martin, R.D. (eds.) Robust and Nonlinear Time Series Analysis. Lecture Notes in Statistics, vol. 26, pp. 256–274. Springer, New York (1984)
29. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
30. Scholkopf, B., Smola, A., Muller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**, 1299–1319 (1998)
31. Seal, H.L.: Studies in the history of probability and statistics XV: The historical development of the Gauss linear model. Biometrika **54**(1–2), 1–24 (1967)
32. Siegel, A.F.: Robust regression using repeated medians. Biometrika **69**(1), 242–244 (1982)
33. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
34. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press, San Diego (2009), pp. 602, 605, 606
35. Watanabe, S.: Karhunen–Loève expansion and factor analysis theoretical remarks and applications. In: Trans. Prague Conf. Information Theory, Statistical Decision Functions, Random Processes, pp. 9–26 (1965)
36. Wilcox, R.R.: Introduction to Robust Estimation and Hypothesis Testing. Academic Press, San Diego (1997)