CURVE SHAPE MATCHING AND DIFFERENCE DETECTION

Jarmo Pirhonen
Institute of Photogrammetry and Remote Sensing
Helsinki University of Technology
Finland

Commission III, Working Group 2

KEY WORDS: Matching, Registration, Change_Detection

ABSTRACT

A method for matching curves and detecting differences under rigid motion transformations is described. After least squares matching the result for a difference detection purpose can be far from satisfactory. A method presented in this paper use basic rigid motion invariants, distance, angle and dot product in a difference detection search process after least squares matching.

Shortly, main parts of the process are follows. First, a least squares match under rigid transformation is computed. Second, for a one of the curves previously mentioned intra curve invariants are calculated and used later to extract good local areas along the curve for final motion computation. Corresponding points from another curve are solved using specified computations, the method does not extract any features there. The key idea is that when first curve is deformed to another curve some parts of the first curve change less than another parts. Our process tries to detect the less deformed parts and use those parts of a curve in a motion computation. A curve is represented as a B-spline curve function and invariant features are computed from the coefficients of the that function not from the actual curve points.

0. INTRODUCTION

Shape matching and difference detection are here considered as problems that arise frequently in digital close range photogrammetry or geometric computer vision tasks. Typical examples are difference detection between CAD-model and measured model or between measured models which acquired at different times from the same object or differences are needed between different objects which have some similar geometric parts.

Main assumption considered here is that difference detection have to be done without corresponding control information. So, the one side of the problem is matching and another side is difference detection.

From the rigid motion transformation point of view differences between models can considered as errors. Papers by (Karras et al., 1993), (Pilgrim, 1991) and (Zhang, 1994) have for example treated differences as (gross) errors in iterative least squares matching problem. In every iteration (gross) errors are localized and rejected from the next iteration round. Usually errors of different size change the weight of an observation. Problem leads to iterative weighting scheme, where not only the parameters of the motion transformation are iterated, but also the weights are iterated too. In this paper we have not used iterative weighting in the least squares estimation problem but it can be also used with the presented method.

Differential features are commonly used in matching problems. Geometric invariant features remain unchanged under considered geometric transformation. Usually features are extracted independently from each data set and then the correspondences between these invariant features are searched. We also use basic rigid motion invariants, distance, angle, and dot product, but a bit different way than usually.

1. B-SPLINE CURVES

We use parametric B-spline curve representation. Given the knot sequence $t_1<t_2<...<t_{n+k}$ a parametric non-rational B-spline curve of order k (of degree k-1) with the end points $a=t_k$ and $b=t_{n+1}$ can be represented as

$$C_k(t)=\sum_{i=1}^{n} P_i B_{i,k}(t) \tag{1}$$

where t is the curve parameter.
P_i is vector of the coefficients or guiding points (dimension is degree of the curve).
$B_{i,k}(t)$ are B-splines of order k, that can be defined (and also computed efficiently) with the recursive Cox-de Boor algorithm (de Boor,1978).

Here it can only bring to notice some important parts of general problems that spline fitting includes. Good and practical references to B-spline algorithms are (de Boor, 1978), (Schumaker, 1981) and (Piegl et al., 1995).

Following basic things need to be considered in least squares B-spline curve fitting problems. Given noisy data points (observations) solve the coefficients of a approximating spline curve. Curve parameter, t, is unknown for every observation, so usually this parametrization problem is solved first and perhaps is improved later if needed. Chord length parametrization is invariant to rigid motions. To get more information on parametrization see (Ma et al., 1995). Specially in matching problems it usually helps a lot if chosen parametrization is invariant to used geometric transformation (for example if affine geometric transformation is used then affine invariant parametrization is good choice). Knots divides the chosen curve parametrization to finite segments. The number of knots and placement is needed. Usually knots are chosen using heuristic rules, such as every n:th data point is a knot (value of a knot is a curve parameter value of that point). Automatic selection of number and positions of the knots see (Cox et al. 1988) and references there. The number of knots defines also the number of spline coefficients (number is not the same, see definitions). Also suitable degree of a curve should be selected. Curve parameters, knots and degree of a curve defines the basis functions, B-splines. Resulting linear least square system is sparse.

If a fitting problem is formulated selecting for example chord length parametrization a result can be seen in figure 1 (knots have been placed to curve parameter value of every tenth observation, degree of the curve is three). Small circles are observations and line between a circle and the curve is a residual. As mentioned earlier curve parametrization can be improved, see (Guéziec et al.,1994), (Sarkar et al.,1991). We select optimization algorithm that finds minimum length between an observation and the curve by golden section search and parabolic fitting. Optimization algorithm computes the new curve parameter values. With the new curve parameters the least square problem for spline coefficients is solved once again. This may be repeated if needed or solution satisfies the specified criterion, see figure 2.

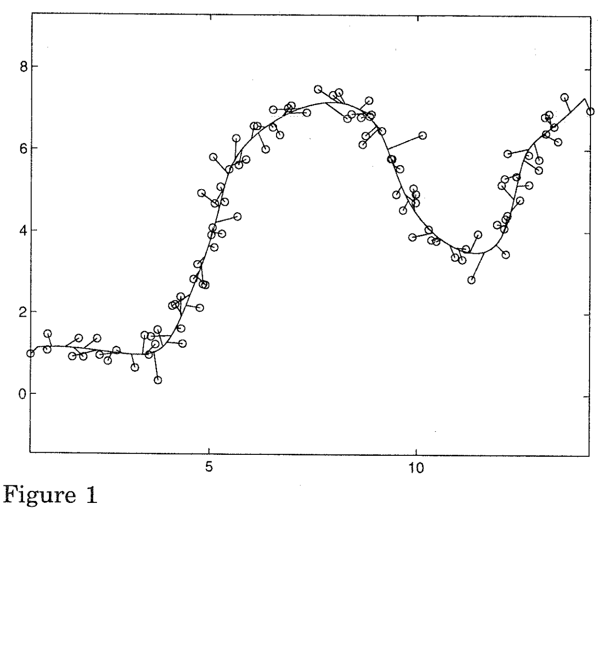


Figure 1

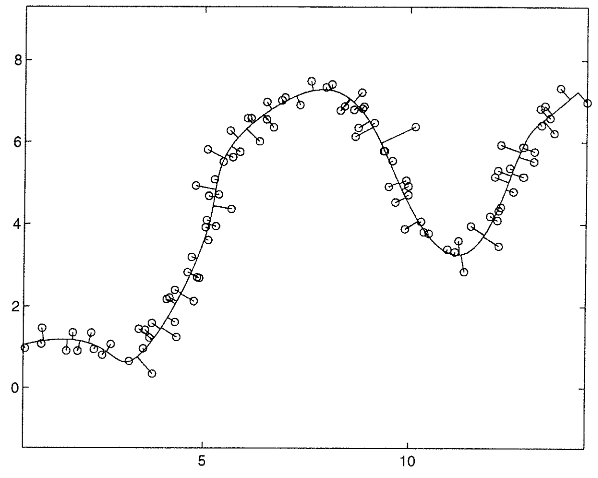


Figure 2

2. MATCHING

The first geometric invariant used here is angle between three consecutive spline coefficient vectors, see figure 3 for a third degree spline curve and figure 4 for a first degree spline curve. An angle can be defined for example clockwise from a first segment to a second segment. This feature is computed at every node but first and last of the coefficient polygon. For a closed curve, the angle can be computed at all nodes. In 3-D case three consecutive coefficient vectors form a local plane and the angle is defined in that plane. So the defined angles does not change if a rigid motion applied to the coefficient vectors.

The second invariant is product of two distances computed between three consecutive coefficient vectors.

The third feature is combination of both previously defined invariants, dot product of two vectors formed by three consecutive spline coefficients. Also absolute value of a vector product is rigid motion invariant but it is not used here.

Three chosen invariants do not need any derivative information as many differential rigid motion invariants do. Invariant measures are computed for coefficients of a curve and effective area of a coefficient depends on the degree, n, of a curve. Effective area (or support area) of one coefficient is n consecutive knot segments. See figures 3 and 4, a curve is changing locally when one coefficient has moved (notice the arrow).

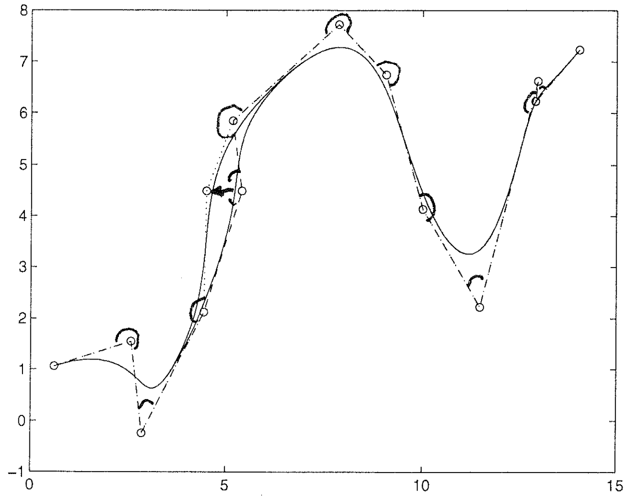


Figure 3

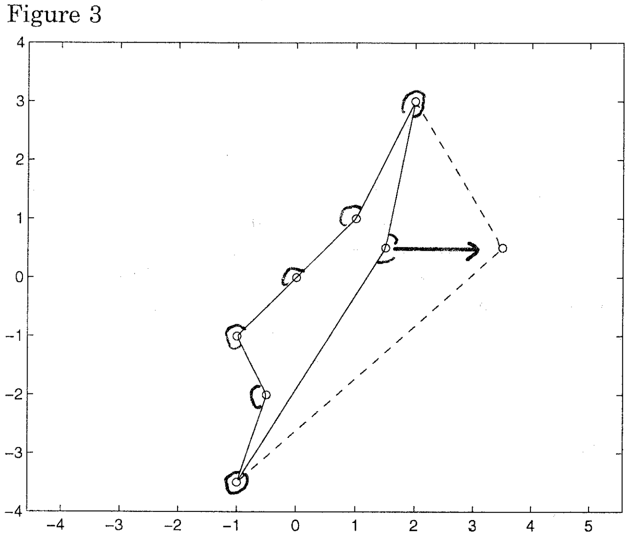


Figure 4

Matching procedure can described as follows:

- Compute the least square match under rigid motion transformation between the curves. We use the algorithm from (Pirhonen et al.,1994). The algorithm there is for 2-D affine case but can be replaced by a rigid transformation.

- For the first curve compute the invariants. These can be computed also before matching because they remain unchanged under rigid motions.

- Compute local least squares match between curves. By local match we mean no predefined global transformation is computed between curves. Just the coefficients of the first curve are changed so that first curve fits to the second curve. Local match needs corresponding points. These are selected for example by the closest point algorithm (Besl et al., 1992) or by curve parameter transformation (Pirhonen et al.,-1994) which one we have selected. The curve parameter transformation assumes that curve parametrization is computed using rigid invariant curve parametrization. Curves are usually approximated from data points and for this reason the curve parameter transformation can not be very accurate. So also combination of these methods might be useful, especially because closest point algorithm needs an initial value.

- For the first curve compute the invariants once again.

- Rank the differences between invariants computed before and after local match. Select some absolute or relative criterion that cluster the ranked coefficients to two groups. First group includes the coefficients where the change in invariant measures have been below the criterion. For the final motion computation curve points from the first curve can be chosen from the support areas of the coefficients that belong to the first group. Corresponding points from the second curve are selected same way as in local matching phase.

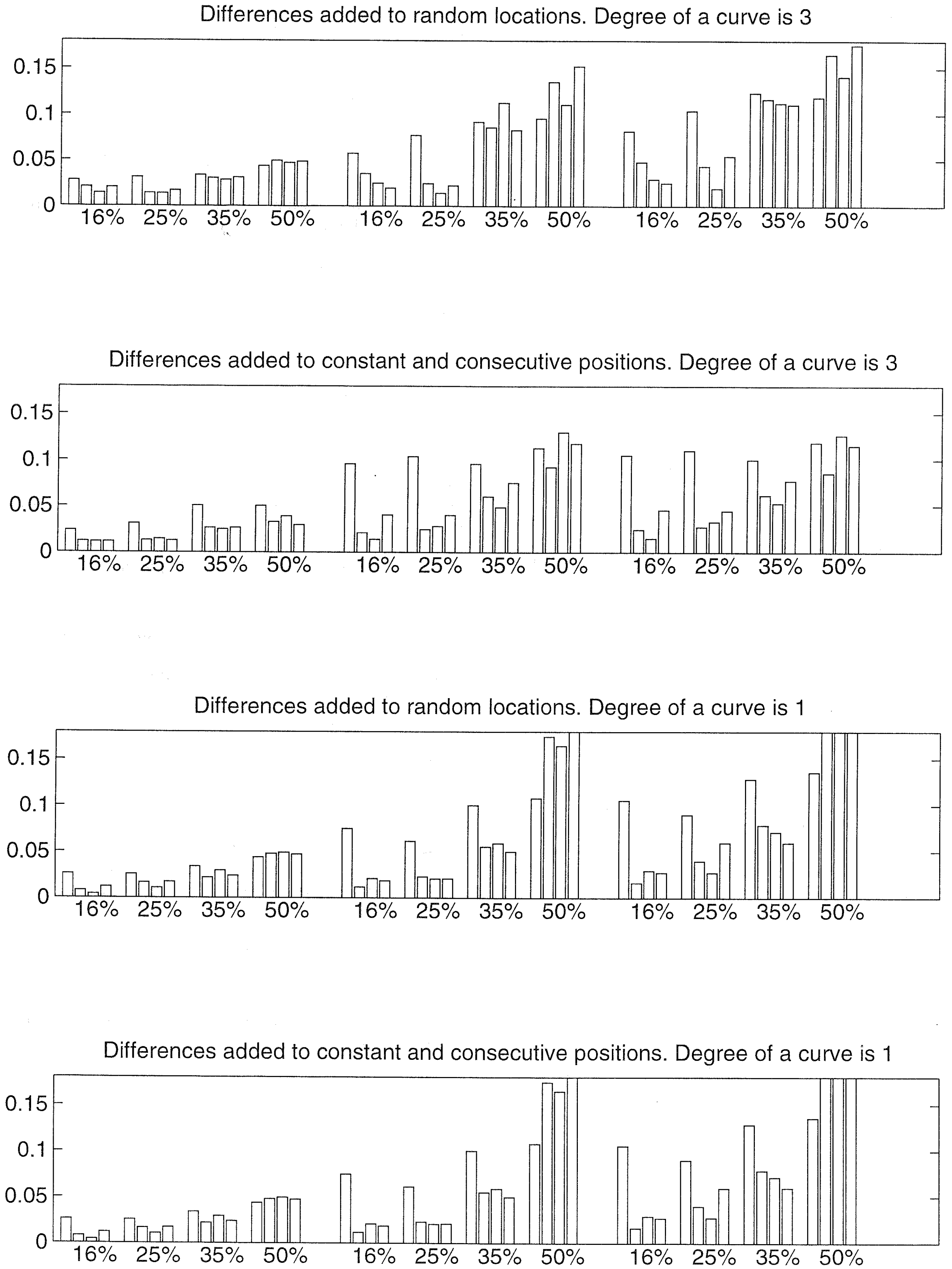
The algorithm does not handle curve identification problem at all. If corresponding curves are not known that problem must be solved first.

3. RESULTS

Sixteen test cases were generated. In the first basic group of 8 cases we added random differences to random places of the second curve. Four groups that have different number of differences were included, 16%, 25% ,35% and 50% of the coefficients of the second curve changed. First degree curve and third degree curve cases was chosen. The reason for this was that coefficients of the first degree curve have smaller support area than coefficients of the third degree curve. The rest 8 cases random differences were added to constant and consecutive coefficients. Otherwise these cases was generated same way as in the first basic group. Each of the 16 cases were generated 50 times. The rigid motion under these test is 2-D transformation, so it includes two shifts and one rotation.

Results can be seen in figure 5. Each bar graph has 48 bars. First 16 bars are for x-coordinate shift, second 16 bars are for y-coordinate shift and rest 16 bars are for the rotation angle. Each bar defines how much the mean value of the 50 computations deviates from zero. If there is no deviation the original rigid transformation is recovered exactly. The subgroups that includes four bars are: First bar defines deviation just after least squares matching. The rest three bars defines deviation after final motion computation when features were dot product, angle, and product of distances respectively. It is noticed that this number of cases and computations does not produce purely robust statistical information.

When 50% of coefficients have differences the method does not have any positive effect. In all 16% and 25% cases the method have positive effect. From the first degree curves the method recovered the original motion better than from the third degree curves. That was expected because in the first degree curve the invariants


Differences added to random locations. Degree of a curve is 3
0.15
0.1
0.05
0
16% 25% 35% 50% 16% 25% 35% 50% 16% 25% 35% 50%
Differences added to constant and consecutive positions. Degree of a curve is 3
0.15
0.1
0.05
0
16% 25% 35% 50% 16% 25% 35% 50% 16% 25% 35% 50%
Differences added to random locations. Degree of a curve is 1
0.15
0.1
0.05
0
16% 25% 35% 50% 16% 25% 35% 50% 16% 25% 35% 50%
Differences added to constant and consecutive positions. Degree of a curve is 1
0.15
0.1
0.05
0
16% 25% 35% 50% 16% 25% 35% 50% 16% 25% 35% 50%
Figure 5

are more local. Same relative clustering criterion was used in all 800 computations and it was 20% of the number of a curve coefficients. Two examples are given in figures 6 and 7. In figure 6 both curves are third degree curves and 35% of the coefficients of the second curve have differences in randomly chosen locations. In figure 7 both curves are first degree curves and 35% of the coefficients of the second curve have differences in randomly chosen locations.

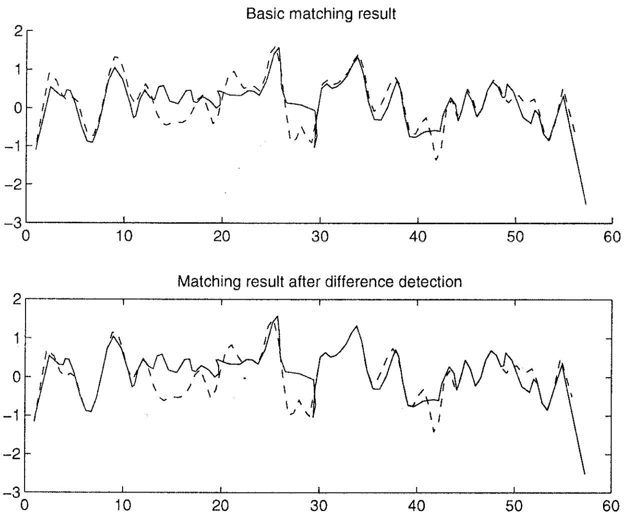


Figure 6

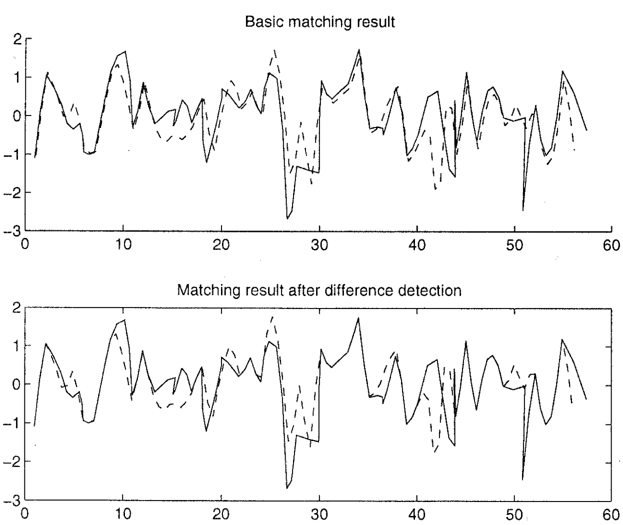


Figure 7

4. REFERENCES

Besl. P. J., McKay. N. D., 1992. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence 14 pp. 239-256.

de Boor, C., 1978. A practical Guide to Splines. Springer-Verlag, New York.

Cox. M. G., Harris. P. M., Jones. H. M., 1988. A knot placement strategy for least squares spline fitting based on the use of local polynomial approximations. Algorithms for Approximation II, ed. Mason J. C. and Cox M. G., Chapman and Hall, London.

Guéziec. A., Ayache. N., 1994. Smoothing and matching of 3-D space curves. International Journal of Computer Vision 12, pp. 79-104.

Karras. G. E., Petsa. E., 1993. DEM matching and detection of deformation in close-range photogrammetry without control. Photogrammetric Engineering & Remote Sensing 59, pp. 1419-1424.

Ma. W., Kruth. J. P., 1995. Parametrization of randomly measured points for least squares fitting of B-spline curves and surfaces. Computer-Aided Design 27 pp. 663-675.

Piegl. L., Tiller. W., 1995. The NURBS Book. Springer-Verlag, Berlin.

Pilgrim. L. J., 1991. Simultaneous three dimensional object matching and surface difference detection in a minimally restrained environment. Research Report no. 066.08.1991 The University of Newcastle, New South Wales, Australia

Pirhonen. J. Inkilä K., 1994. Shape matching by splines. International Archives of Photogrammetry and Remote Sensing 30 pp. 678-682.

Sarkar. B., Meng. C-H., 1991. Parameter optimization in approximating curves and surfaces to measurement data. Computer Aided Geometric Design 8 pp. 267-290.

Schumaker. L. L., 1981 Spline functions: Basic Theory. John Wiley & Sons, New York.

Zhang. Z., 1994. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision 13 pp. 119-152.