AUTOMATED ANALYSIS OF THE FORM ERROR CAUSED BY SPRINGBACK IN METAL SHEET FORMING

Biermann, D.[1]; Surmann, T.[1]; Sacharow, A.[1]; Skutella, M.[2]; Theile, M.[2]

1. Institute of Machining Technology (ISF), TU Dortmund, Germany
2. Combinatorial Optimization & Graph Algorithms Group, TU Berlin, Germany

ABSTRACT
In mechanical engineering, especially in metal sheet forming, the compensation of form errors caused by springback is usually handled by numeric simulation or repeated refinement of a workpiece. But since the simulation of springback can generally only approximate the real process, costly refinements cannot be avoided. Thus, different means of detecting and compensating deformations in manufactured workpieces at the earliest stage in the production chain are required to keep the manufacturing process as efficient as possible. A detailed analysis can assist in generating appropriate rules for error compensation by modifying the tool or changing relevant process parameters. This article presents a new approach combining traditional methods from the fields of production and discrete mathematics to locate and identify form errors.

KEYWORDS: form error, springback, CAD, best fit

1. INTRODUCTION

Free-form metal sheet forming often requires tools with high shape accuracy and long lifetime. Coated tools may meet these requirements but the design, manufacturing, and use of these tools pose a big challenge. On the one hand, processes in the manufacturing chain like milling, coating, or grinding may cause errors, which accumulate to a form error of the tool. On the other hand, springback in the forming process can cause strong deformations of the workpiece. The usual way to handle this problem is the trial-and-error method, in which the workpiece geometry is adapted to the target geometry. Thus, it is necessary to develop rules for an error compensation and an optimization of the whole manufacturing process. Thereby, the numerical simulation of the manufacturing processes can be used to remove the typical trial-and-error iterations and reduce the manufacturing costs. The process of form error compensation is shown in Figure 1.

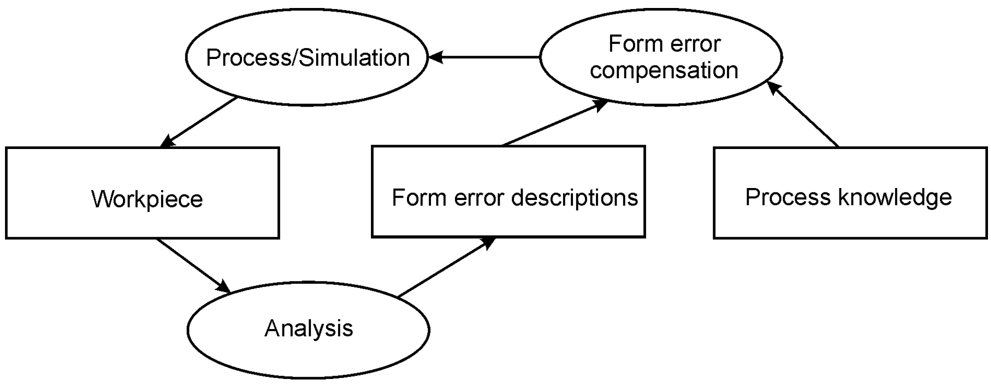


Figure 1: The process of form error compensation.

The analysis of the deformations and the generation of form error descriptions play a major role therein. To detect and describe the deformations, methods have to be found to compare the

actual shape of the workpiece with the desired geometry. In this paper the common method of comparison by the global registration /1,2,3/ is presented as well a new approach using multiple local best fits.
The production setup consists of a CAD model in IGES-format and a manufactured component. The latter can be the result of a production process like deep drawing or milling. It is then scanned by a tactile or an optical scanning device like a laser scanner. A generic and always available output of the scanning process is a point set. In most cases it is also possible to obtain a triangulation from the scanning device although this depends on the particular commercial software, which is provided with the scanning device. However, in the presence of undercuts, it is impossible to generate a triangulation at all. Thus, the main focus will be on point sets as input data for the deformation analysis, but we will also discuss the use of triangulations where appropriate.
In section 2 the pre-processing of the model and the digitized data is discussed, followed by a description of the applied global best-fit algorithms in the next section. However, the output of these algorithms is only the first step required in the analysis of the form error. Thus, a new approach of local best fits is described in section 4, and the construction of a deformation field is detailed in section 5. Finally, the article ends with a discussion of remaining questions.

2. PRE-PROCESSING

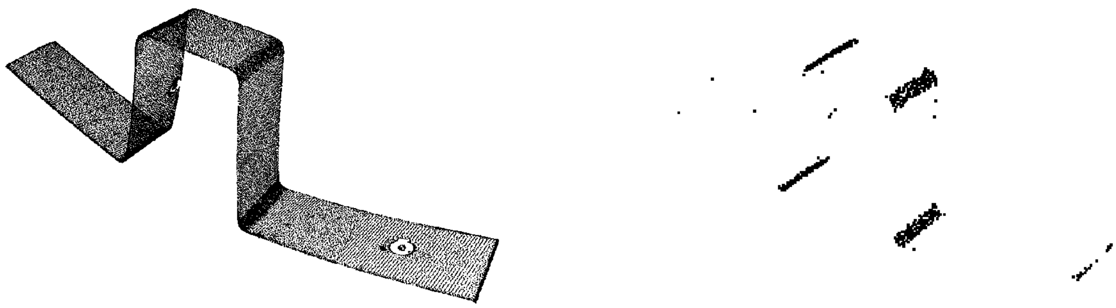
Before the start of the discussion of the data reduction, it should be noted that most best-fit algorithms are working on two point sets, and it is required to extract the shape points from the given CAD model. This can be achieved by scanning the individual surfaces evenly. Depending on the accuracy of the scanning process, the digitized data volume can be very large. This may pose a problem for the algorithms used for the analysis of the form with regard to running time or memory resources. Thus, the data needs to be prepared for further processing, i.e., it has to be reduced without losing relevant information.
In case of a triangulation as input data, a mesh reduction /4/ can be used to reduce the size of the data set without losing too much curvature information. The user specifies a level of approximation to the original geometry by setting up a decimation criterion. A common criterion is the distance of a vertex to its average plane, which is defined by the adjacent vertices. The average plane of a point set has the least square distance to all points from the set and can be calculated by the least square method /5/. Vertices with a smaller distance than the preset threshold are deleted in conjunction with their adjacent triangles followed by a re-triangulation to fill the resulting hole /6/. In the reduced mesh, planes and smooth areas consist of only large triangles and a small number of points, whereas regions of high curvature consist of many small triangles and a large number of data points. By imposing restrictions on boundary vertices, the output of the mesh reduction comprises a data set with vertices along the contour of the object and points within the characteristic, curved areas of the shape.
However, a problem, which may occur, is the poor quality of the initial triangulation mesh. Depending on the triangulation method /7/ used by the scanner software or by a software working directly on point clouds, long and narrow triangles might be generated in regions where the shape has a sharp edge. This could result in a poor reduction because the proximity information is not correct. Thus, the mesh has to be optimized in such critical areas by a re-triangulation before the reduction can be applied. The triangulation of points and a subsequent mesh optimization are quite expensive in terms of computation time. Therefore, an alternative reduction method, which is working directly on the point set, is used.
If the point clouds are to be reduced using this type of reduction, a method to extract these important points has to be developed since a simple point set contains less information about the shape than a triangulation. It lacks the necessary information about the shape contour as well as the proximity relation between points and the normal vectors across the shape, which are required for a computation of the curvature information in the decimation process. The normal vector and the curvature at each point have to be calculated before the reduction can be applied to a point set. The normal vector of a point can be set to the normal vector of its average plane, which is determined by the points from the proximity where the proximity of a

point can be defined by a fixed number of nearest neighbors or by the radius of a sphere around the point.

Next, the curvature of the points is calculated and every point, whose curvature is smaller than a fixed value *minC*, which can be set by the user, is removed. Since the exact curvature calculation for discrete points is impossible, an approximate measure *C(p)* for the curvature of a point *p* is defined:

$$C(p) = \frac{1}{m}\sum_{i=0}^{m-1}\frac{n \times q_i}{1+\|q_i\|},$$

where n is the normal vector of p, q_i is the normalized vector from p to its i-th neighbor and m is the size of the neighborhood of p. It is difficult to choose the right value for m since for small values there may not be enough points for a proper approximation whereas for big values the sphere of influence may be too large, and, therefore, the calculation imprecise. Thus, m can be set to a fixed value, e.g., 10, or a fixed radius can be set for the sphere of proximity. In the second case, the radius must be selected in such a way that the neighborhood of each point contains at least 3 neighbors. The result of a reduction with minimal curvature *minC*=0.015 is shown in Figure 2.



(a) original points (b) points with high curvature

Figure 2: Point reduction via curvature computation.

This method is working very efficiently because a special data structure can be used for the nearest-neighbor search. Geometry libraries like the Visualizations Tool Kit (VTK[1]) or CGAL[2] provide the required structure, e.g., an octree or grid structure, which allows fast initialization and access.

A disadvantage of the latter reduction method is that all points from the plane areas and boundaries might be removed and, thus, some shape details might be lost resulting in an inexact comparison of those shapes. Another problem are the deformations of the manufactured workpiece and the output of scanning devices, which can be noisy or suffer from a lack of density. Both cases impede a comparable reduction of model and digitized data. Thus, another method is used in addition to the one described, which samples points from both data sets with a similar density across the whole shape to obtain similarly reduced data sets. This is also necessary for the computation of the deformation field, which is discussed in section 5.

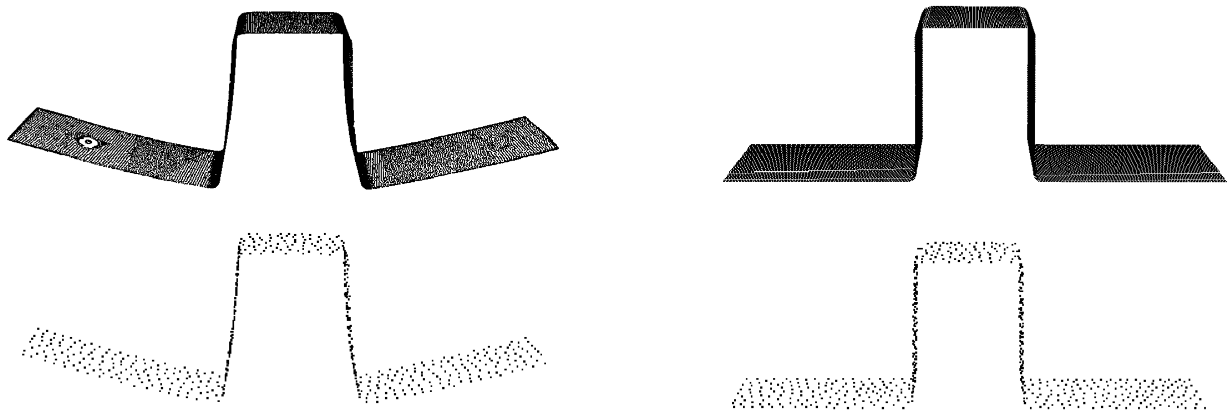
The approach used is the following: either a user-defined amount of points from both data sets or a user-defined threshold for the maximal distance a removed point should have to a point, which is kept in the reduced data set, can be chosen. This notion is captured by a problem known in discrete mathematics as the *k-Center problem*. The problem can be conceived as finding the best place for a preset number of fire brigades such that every house in a city can be reached as fast as possible.

Consider a point set **P** with points from $\mathbb{R}^3$ for each pair of points (p,q) from **P**, their distance d can be computed using the Euclidean distance. The goal is to find a subset **C** of **P** of size k, such that the maximal distance w is minimized, i.e., $w = \max_{p \in \mathbf{P}} \min_{c \in \mathbf{C}} d(p,c)$, which is

[1] http://www.vtk.org/

[2] http://www.cgal.org/

obtained by measuring the distance each point p from P has to its nearest center c from **C** and taking the maximum over all these distances.
For the computation of the centers, a simple method called Plesnik's algorithm /8/ is used, which will iteratively add points to the set of centers **C** until the preset number k is reached. In each iteration the algorithm adds the point to the output data set which is farthest to the last-added point, starting with an arbitrary point. After the algorithm has terminated, the weight w is known, which gives the maximal distance a point $p \in \mathbf{P}$ has to its nearest center $c \in \mathbf{C}$. The algorithm can be adapted to the case that a user sets the weight w as a threshold. In that case, the algorithm starts with an arbitrary point and adds the point farthest to the last point added to **C** as long as the current distance does not fall below the given threshold. The output of both variants is a subset **C** of points from the original set **P**, which is used as the reduced data set.



(a) data points (b) model points

Figure 3: K-Center reduction applied to the profile with weight 10.

For the computation of the deformation field as described in section 5, the k-Center computation is used with a preset threshold w for both, model and digitized data (compare Figure 3 for an example with weight 10). The model point input consists of 16,230 points and is reduced to 594 points whereas the digitized data consists of 25,406 points and is reduced to 604 points. If the number of neighbors within distance w is counted for each of the remaining neighbors, it can be observed that the algorithm creates a similar point distribution in both sets, compare Figure 4. The difference between both sets results from the fact that the manufactured workpiece might consist of some additional material from the specific production process. This means that the size of the reduced data set differs.

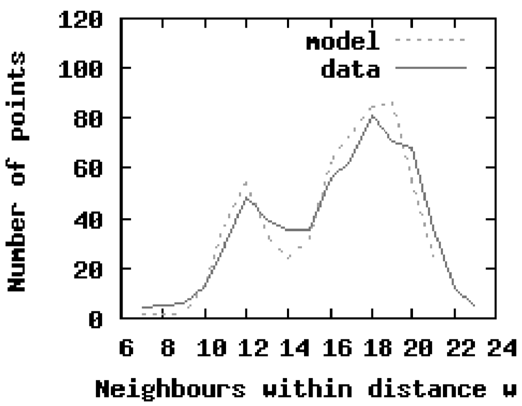


Figure 4: Histogram of the k-Center reduction with weight 10 applied.

Although this reduction does not use information about the implicitly underlying surface or additional information about other characteristic areas, the algorithm is successful in solving the correspondence problem described in section 5. Additionally, it can be easily combined with any other reduction technique by providing a base sample of points from the input to which more specialized information can then be added.

3. GLOBAL REGISTRATION

The global registration (best fit) is the first step in the comparison of the digitized data with the CAD model. At this point, both data sets have to be transformed to the same coordinate system, and the best alignment between model and data has to be found. This problem poses a great challenge, especially in case of deformed objects. This means that the digitized shape differs from the target geometrical shape given by the CAD model, and it is thus very difficult to find the optimal alignment. As a first step a transformation for the digitized data has to be calculated, which minimizes the error metric based on the distance between data and model. Given a mapping between points of model and date, the average square distance E between two point clouds P and Q can be defined as the normalized sum of square distances between corresponding points (Euclidian metric):

$$E = \frac{1}{n}\sum_{i=0}^{n-1}(p_i - q_i)^2 , \qquad (1)$$

where n is the number of corresponding pairs and (p_i,q_i) is the i-th pair.
Thus, the first goal is to find the corresponding point in the model for every data point. The usual way is the mapping of each data point to its closest model point. In the next step, for a given set of corresponding points a rigid transformation can be computed, which minimizes the distance between the shapes with respect to the error metric. This popular algorithm is also known as ICP algorithm (Iterative Closest Point) /1,2/. It is an iterative process, which exhibits linear convergence in the general case /9/. The termination condition can be given by a convergence criterion like the error or an error reduction, or by the number of iterations. There are two types of ICP. The fist version uses the point-to-point metric of Besl and McKay /1/, which minimizes the Euclidian distance between corresponding points. The other version is the point-to-plane ICP, which uses the metric of Chen and Medioni /2/ based on the distance between a point and the tangential plane of the corresponding point.
The disadvantage of the ICP is that it does not guarantee to find the global optimum. Depending on the initial position, the ICP may only find a local optimum. Therefore, it is only useful if the data is already close to the model. This is generally not the case and, thus, it is necessary to first move the data close to the model, i.e., find a good initial alignment. For this purpose an algorithm is used, which is based on the principal component analysis (PCA) /10/. The main principle of this method is to transform the data so that the principal components and the centroids of both shapes correspond. To calculate the principal components of a 3D point cloud, the shape first has to be translated to the origin. The principal components of the centered point cloud are the eigenvectors of the covariance matrix $C=AA^T$. A is a $3×n$-matrix, where n is the number of points, and the i-th column vector contains the coordinates of the i-th point. The covariance matrix C is a symmetric, positive definite $3×3$-matrix and the eigenvectors can be easily calculated by the Jacobi-Davidson method /11/.
Let P and Q be $3×3$-matrices, which contain the coordinates of the principal components of the data and the model, respectively. The rotation R, which transforms P into Q, can be calculated by using $R*P=Q$ and $R=Q*P^{-1}$ since P and Q are orthogonal. Thus, the data is rotated by R followed by translating it to the centroid of the model. The results of this transformation are shown in Figure 5.

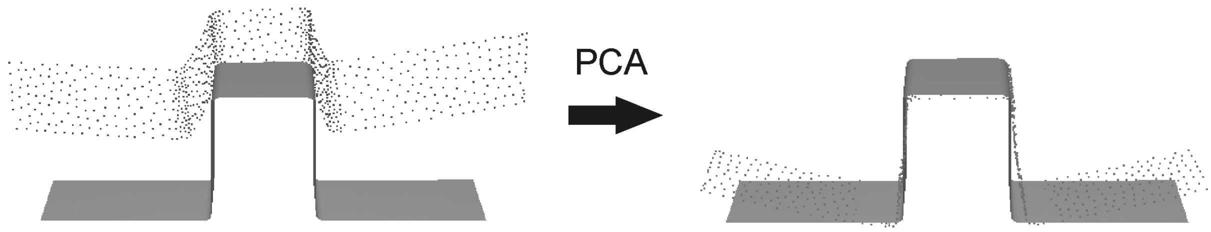
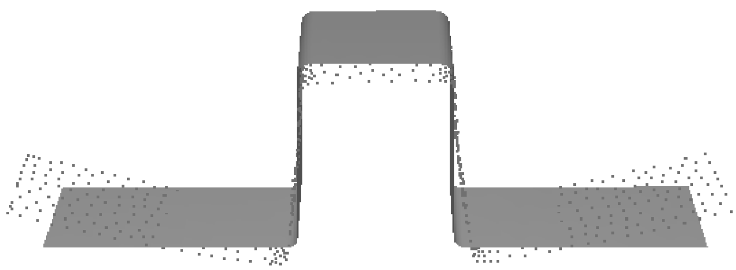
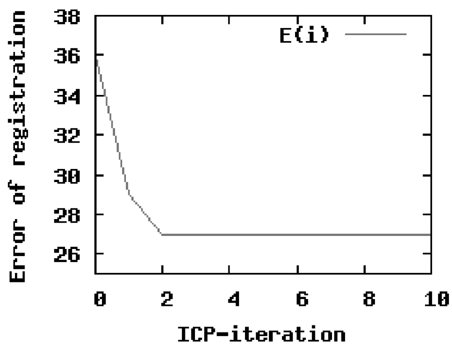


Figure 5: Result of registration based on principal component analysis.

The distance between data and model, i.e., the error, is given by the point-to-point metric defined in equation (1). In this example, it was reduced from 417.951 to 36.289366. After this, 10 iterations of point-to-point ICP are used to improve the alignments. The final result of the global registration is shown in Figure 6a.



(a)



(b)

Figure 6: Final result of the global registration and analysis of the error reduction.

The progression of the error reduction is shown in Figure 6b. The quotient *E(i)/E(i-1)* is close to 1.0 starting with the second iteration, i.e., there were no relevant changes during the last 8 iterations and so the error did not improve. Due to strong deformations of the workpiece, it is difficult to find a better registration, which minimizes the distance *E* defined in (1). In this case it is also exceedingly difficult to compare both shapes and to classify the deformations. From the error and the computed transformation, it could be deduced that the whole workpiece was deformed. In fact, the cause for the deformations in this example is a bending at the rounded edges. Thus, some subareas of the workpiece are rotated, and it is impossible to fit all the subsets of data at the same time. The global registration minimizes the total error between the entire shapes, and this can entail that the corresponding subareas cannot be matched even though the local deformations of these areas are not strong. In this case, we obtain no relevant information about the cause for the form error.

If the springback is not strong, the global best fit is more successful and simple form errors like overmeasure can be deduced from the following visualization: the distance of data points to the model can be encoded by color. Thus, points, which are located below or above the model, become visible. Therefore, it can be deduced that there are deformations in the colored areas. An example of such deformations is shown in Figure 7. The transformation of data was calculated by principal component analysis followed by 40 iterations of ICP. The progression of the error reduction is also shown in Figure 7. To get more information about the kind and measure of deformation, the technique of multiple local best fits is used, which will be introduced in next section.

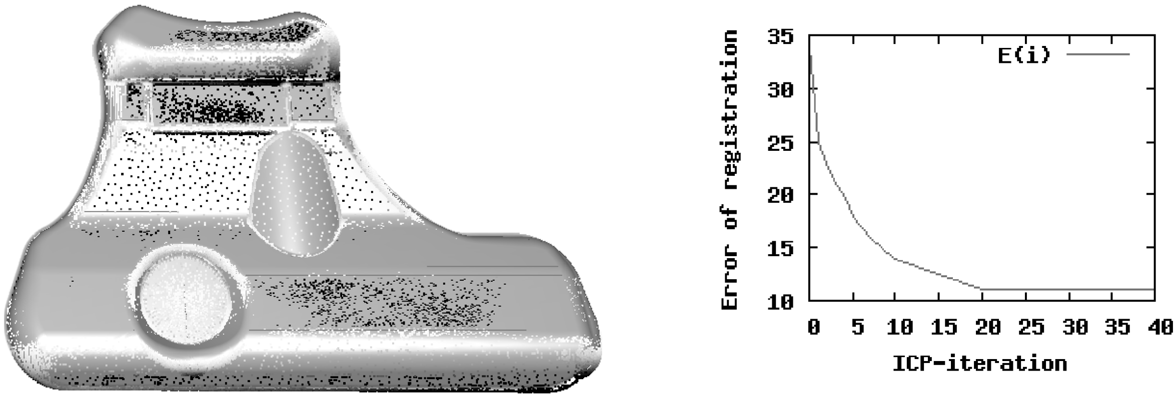


Figure 7: Global registration and coloring of the points with respect to distance.

4. LOCAL REGISTRATION

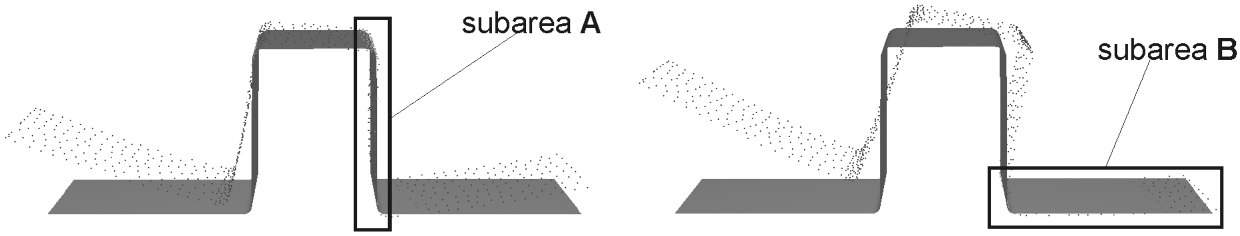
The results of the global registration cannot always be used to detect and describe the deformations of a workpiece. Especially in case of strong deformations, it is impossible to align all subareas of the data to corresponding subareas of the model by using the ICP technique of nearest neighbours. Thus, this matching delivers no information about the kind of deformation, but rather a measure of the form deviation. Nevertheless, the global registration transforms the data as close to the model as possible and establishes a good basis for further analysis.

In the next step, subareas of the workpiece are analyzed in order to get more information about local deformations in these areas and relative transformations between adjacent subareas. The user is able to select a subset of data points and fit these points locally to the model by using the ICP algorithm. Thus, the local registration computes a transformation, which minimizes the distance between a selected subset of the data and the entire model. The advantage of this method is that the subset will be fit to the model in the best possible way without considering the alignment of other data parts. Thus, local deformations inside the subareas will become apparent.

The described method can be used to detect the rotation or bending between two subsets of the workpiece. Let **A** and **B** be two adjacent subsets of data (see Figure 8). The angle between these subsets in the model is 90°. Now, the corresponding angle in the data shall be determined, i.e., the rotation axis and rotation angle of one subset relative to the other shall be calculated. First, **A** is fitted into model. In Figure 8a, it can be seen that there are no strong deformations in this part and the match is almost exact. In the next step, a best fit for the subset **B** is computed (see Figure 8b), which also fits well into the model. The local errors for the subareas **A** and **B** are 1.9724 and 2.42, respectively. The rotation matrix *R* of the last transformation (from a to b) describes the displacement of **B** relative to **A**. The eigenvalue decomposition of *R* yields the rotation axis *v*. It is the eigenvector to the eigenvalue of 1.0. Since *R* is a rotation matrix, it has always only the one eigenvalue of 1.0. The rotation angle α is given by:

$$\cos(\alpha) = \frac{\langle w, R \cdot w \rangle}{\|w\| \cdot \|R \cdot w\|},$$

where *w* is a vector perpendicular to the rotation axis and $\langle\,,\rangle$ denotes the scalar product of two vectors.



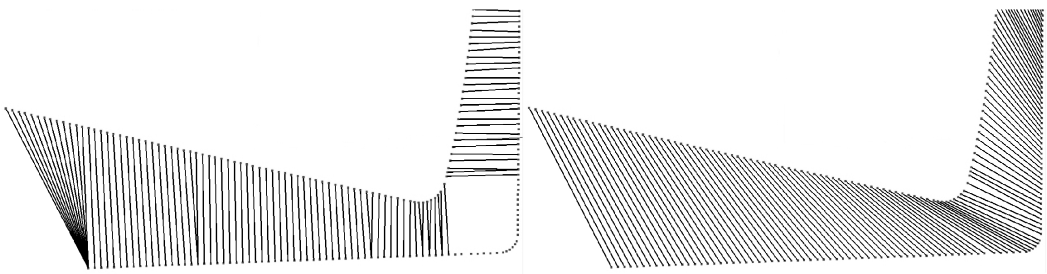
(a) best fit in subarea **A** (b) best fit in subarea **B**

Figure 8: Local best fits.

In Figure 8 the rotation from (a) to (b) is given by the rotation axis v=(-0.2614, 0.1958, 0.9452) and the angle α=8.8448°, which means that the angle between the subsets **A** and **B** in the data is 8.8448° smaller then in the model. Similarly, the other subset can be tested for the relative rotations. A disadvantage of this method is that the user has to select the subsets manually, and it is not necessarily evident which subset should be selected. But the global registration and coloring of the data points according to the distance to the model may ease the selection of subsets. With local best fits, the local deformations in discrete subareas can be analyzed more precisely. The correspondences computed by the local best fits are also more precise than the ones obtained by global registration. The improved mapping can be used to set up a deformation field across the model, which describes the workpiece deformation. This is presented in the next section.

5. DEFORMATION FIELD

The deformation field is a common method to describe the deformations of a workpiece. It is composed of a set of vectors across the model shape, which represents the transformations of discrete model points. The information gained from this can be used to change parameters of a numerical simulation or to adapt the NC program of a milling process. The field can be built up by the correspondences of the best-fit algorithms. In the presented case the alignments of the last ICP iteration from the global registration are used. But since the ICP assigns a point on the model shape with the smallest distance to every data point, it cannot be guaranteed that the entire model shape will be covered by this deformation field. In Figure 9a it can be seen that there are subareas on the model with undefined deformation vectors.



(a) (b)

Figure 9: Nearest-Neighbour matching (a) and matching by the bipartite weighted matching method (b) applied to the profile.

This problem can in general not be avoided although it is possible to obtain a complete cover of the model by assigning each sampling point of the model to its nearest neighbor from the data. However, the correspondences could be incorrect and some points of the data are not matched. The computation of the correspondence pairs by the nearest-neighbor relation is, due to its greedy nature of taking the nearest neighbor, not capable to match points on edges correctly.

One possibility is to use the information from the local best fits. The mapping of the corresponding points, which is calculated for the subareas, is more exact than the mapping obtained by global registration. Therefore, these local sets of correspondences are combined into one total set. For the areas of the model shape between the subareas, the correspondences can be interpolated. The disadvantage of this method is that it is a manual and very complex procedure. Furthermore, the interpolation may also cause errors.

In general, the goal is a computation of the correspondence pairs such as to establish a one-to-one relation between points from both data sets. Thus, a method is needed, which can take all such pairs into account at the same time and choose the best global matching as opposed to a successive pairing with the nearest-neighbor method.

Such an approach is known as weighted bipartite matching in discrete mathematics. Given two point sets **P** and **Q**, with the size of **P** being less or equal to the size of **Q**. The distances between every two points from these sets are known and in this case are given as the Euclidean distance. The goal is to find a one-to-one mapping from the points in **P** to points in **Q** such that each point from **P** has a neighbor and each point from Q has at most one neighbor from the point set P. Among all possible bipartite matchings, the cheapest one should be computed, i.e., the one which minimizes the sum of squared Euclidean distances:

$$\sum_i (p_i - q_i)^2 ,$$

where for a given point p_i point q_i is the corresponding point from **Q** given by the matching. An implementation from the LEDA library /12/ was used for the presented computations.

The algorithm is used only after both data sets have been reduced by the k-Center method with a specific weight set by the user. It is important that the data points follow a similar distribution since this gives the necessary constraints over the entire shape. This allows mapping features like edges correctly. If one data set is much larger than the other set, the computation of the cheapest weighted bipartite matching resembles the nearest-neighbor computation, which does not always work. In Figure 9 a correspondence computation by the nearest-neighbor and the weighted bipartite matching is shown. In order to better illustrate the success, two-dimensional data instead of three-dimensional data is used. The result of the weighted bipartite matching approach applied to the profile data sets, which have been reduced by the k-Center algorithm with weight 10, is presented in Figure 10.

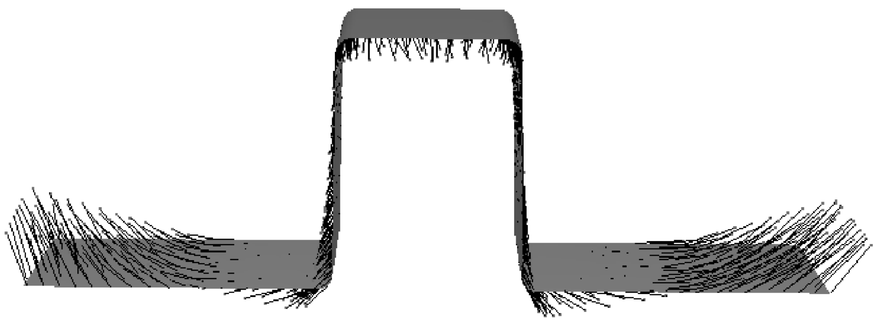


Figure 10: Correspondences computed by a bipartite weighted matching algorithm.

The bending-up deformation is visualized by the correspondence relation, which is computed by the weighted bipartite matching. Moreover, points located on the edges of both sets have been correctly matched to each other. With this approach, it is also possible to perform a visual deformation analysis, as has been shown in section 3 and Figure 7. A more automated deformation analysis is obviously needed, which will be discussed in the final section of this article.

6. CONCLUSIONS AND OUTLOOK

In this article a new approach for the analysis of the deformation of manufactured workpieces has been described. The approach consists of several steps ranging from a pre-processing of the data and using a global best fit for an initial alignment to the application of multiple local best fits and the computation of a deformation field. Ideas from computer graphics and discrete mathematics are brought together in the preprocessing step and for the computation of a deformation field. Especially the idea to compute an improved deformation field has proven to be very successful for mapping features between the digitized data and the model. Moreover, the use of multiple local best fits for the deformation analysis has never been tried before. It was possible to successfully derive the local bending-up of the profile.
However, several open issues remain. The possibility to apply multiple local best fits to the workpiece, which has been used in Section 4, shows that it is possible to gain new information on the deformations in an automated computation step. Although the automation of the analysis has not yet been obtained with the weighted bipartite matching approach from section 4, it should be possible to automate the computation of multiple local best fits by using the information gained from the deformation field. This step will then lead directly to an approach for the automated analysis of deformations of a workpiece. Another open issue, which remains, is the exploitation of triangulations in the deformation analysis. This approach will open up new possibilities to compute a global best fit on the one hand. On the other hand it will help with the segmentation of the digitized data into components that can be mapped to features in the model. This in turn will lead to new methods of a deformation analysis.

7. ACKNOWLEGMENT

This paper is based on investigations of the collaborative research center SFB708, which is kindly supported by the German Research Foundation (DFG).

8. REFERENCES

1. Besl, P. J.; McKay, N. D.: A Method for Registration of 3-D Shapes. IEEE Trans. Pattern Analysis and Machine Intelligence 14 (2) (1992) 239-256.
2. Chen, Y.; Medioni, G.: Object modelling by registration of multiple range images. Proc. IEEE Conf. on Robotics and Automation (1991) 2724-2729.
3. Gelfand, N.; Mitra, N.J.; Guibas, L.J.; Pottmann, H.: Robust Global Registration. Eurographics Symposium on Geometry Processing (2005) 197-206.
4. Schroeder, W. J.; Zarge, J. A. Lorensen, W. E.: Decimation of Triangle Meshes. Computer Graphics (Proceedings of SIGGRAPH 92) (1992) 65-70.
5. Hanson, R. J. Lawson C. L.: Solving least squares problems. SIAM, Philadelphia. (1995)
6. Toussaint, G.: Efficient Triangulation of Simple Polygons. The Visual Computer 7 (1991) 280-295.
7. Preparata, F. P.; Shamos, M. I.: Computational Geometry – an Introduction. Springer-Verlag, New York. (1985)
8. Plesnik, J.: A heuristic for the *p*-center problem in graphs. Discrete Appl. Math. 17 (3) (1987) 263-268.
9. Pottmann, H.; Huang, Q. X.; Yang, Y. L.; Hu, S. M.: Geometry and convergence analysis of algorithms for registration of 3D shapes. Int. J. Computer Vision 67 (3) (2006) 277-296.
10. Jolliffe, I. T.: Pincipal component analysis. Springer-Verlag, New York. (1986)
11. Sleijpen, G. L. G.; Van der Vorst, H. A.: A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems. SIAM Journal on Matrix Analysis and Applications 42 (2) (2000) 267-293.
12. Mehlhorn K; Näher S.: The LEDA Platform of Combinatorial and Geometric Computing. (1999) Cambridge University Press.