„A Shape-Matching Algorithm for deep drawing applications"

Uli Göhner, Markus Brüchle

University of Applied Sciences Kempten

Abstract:

To get a more precise description of the material behavior of automotive parts in crash scenarios it is important to take into account the production processes of the formed parts [10]. For this purpose different thickness and strain tensor data from the results of the stamping simulation have to be mapped to the car components. Typically different coordinate systems are being used for the stamping and the crash simulation. Also mesh sizes are different in stamping and crash applications. Before the necessary strain interpolation process can be started (*include stamped part), the stamped part and the crash part must be matched. For this purpose different algorithms from computer graphics programming have been compared and one algorithm (Iterative Closest Point) was implemented and tested on different configurations. The work was made by Markus Brüchle as his Diploma Thesis and supervised by Uli Göhner at the University of Applied Sciences Kempten.

Keywords:

Forming simulation, Crash simulation, Mapping, Matching, Iterative Closest Point Algorithm (ICP)

1 Problem

1.1 Application in automotive industry

In stamping applications the drawing direction points into the z-direction. As in the crash application a lot of different parts with different drawing directions are coming together, a matching is necessary before the interpolation process of the stamped data can be performed [10]. The matching can be described as follows: Given two discretized surfaces, namely stamped part and crash part, we look for an appropriate transformation, which minimizes the volume between these two surfaces. Stamped part and crash part are similar, but not necessarily identical, because of different mesh sizes used in stamping and crash applications. Both surfaces are given in discretized form. The element form is not of any importance, as the algorithms we will investigate only need a node set for the description of the surfaces. Such type of matching problems are common in image processing and computer graphics, e.g. for pattern recognition or other purposes. Therefore in the next chapter different algorithms from computer graphics are being compared.

1.2 Mathematical description

Let $\{x_i\}$ and $\{y_i\}$ be the two node sets, which are to be matched. Then we look for the rotation matrix R, the transformation vector t and the scaling factor c, so that the squared sum of the node differences e is minimal:

$$e(R,t,c) = \sum_{i=1}^{n} (y_i - (cRx_i + t))^2 \qquad (1)$$

2 Comparison of different matching algorithms

2.1 Umeyama Algorithm

In 1991 Umeyama [9] presents an algorithm based on the papers of Horn, Arun et al [1], [5], [6] which gives a complete solution of our problem which is also called the “problem of absolute orientation”. We denote with $X=\{x_{1,...}, x_n\}$, $Y=\{y_{1,...}, y_n\}$, $\sum XY$ the covariance Matrix, μ_X and μ_Y the mean values and σ_X and σ_Y the standard deviations of X and Y. Given the singular value decomposition of $\sum XY$ with UDV^T and defining the Matrix S:

$$S = \begin{cases} I, & \text{if } \det(\Sigma XY) \geq 0; \\ diag\{1,....,-1\}, & \text{if } \det(\Sigma XY) < 0. \end{cases}$$

Then if rank($\sum XY$) = n, the solution for R, t and c can be identified by the following formula:

$$\begin{aligned} R &= USV^T \\ t &= \mu_y - cR\mu_x \\ c &= \frac{1}{\sigma_X^2} tr(DS) \end{aligned} \qquad (2)$$

By formula (2) we could directly compute the solution to our problem, however there are 2 conditions, which cannot be ensured in practice: The number of points in X and Y must be the same and the correspondence between the points must be known. In practice the stamped part and the crashed part differ, because of different mesh sizes, so that the dimension of the two vectors X and Y are different, too. Also the correspondence between X and Y requires an identical mesh and identical node numbering within crash and stamped part. Due to that reason, the direct computation of the solution through formula (2) is impossible. Nevertheless, if the meshes in the crash and stamped part are identical and the node numbering is the same, the Umeyama Algorithm gives a very fast and direct solution to our problem.

2.2 Iterative Closest Point

Besides the direct approach following Umeyama also iterative procedures have been developed. In 1992 Paul Besl and Neil McKay from General Motors published the Iterative Closest Point (ICP) algorithm [2]. If the node numbering is different in the two parts, it is unknown, which point from the original point set corresponds to which target object. This problem is being solved by an iterative

procedure. For each point in the original point set the target object with the smallest distance ("closest point") will become the partner. Possible target objects could be e.g. points or triangular elements. This could be the wrong choice in the early stages of the iteration processes, but will become better the closer the two parts get.
Let $\{x_i\}$ and $\{y_i\}$ be again two node sets, which should be matched and denote by Z=C(X,Y) the operator, which gives the closest point to a given configuration. Then to this configuration the solution R,c,t can be calculated from (2). Let Q(X,Y) be the operator which gives the registration vector q, the new point set q(X)=cRX+t and the resulting least square error e. Then the algorithm reads as follows:

1. Initialize R=I, t=0, c=1, X_0=X
2. Iterate until $|e_k-e_{k-1}|<\varepsilon$ (given accuracy):
    a. Compute closest Point $Z=C(X_k,Y)$
    b. Compute new point set $X_{k+1}=Q(X_k)$ and corresponding least square error e_k

It can be shown, that this algorithm converges only to a local minimum. By using a good initial configuration X_0 for the applications we want to address in this paper, we can easily get the right solution. The numerical costs are $O(n^2)$ per iteration. As the dimension of X and Y must be still the same, different variants of this ICP are considered in the next chapter

2.3 Trimmed ICP and Fractional ICP

To overcome also the problem arising from different dimensions for X and Y different variants of the ICP are proposed. In the Trimmed ICP algorithm a Least Trimmed Squares (LTS)-approach is followed [3], [8]. LTS means, that only a ("trimmed") subset of all error contribution is used for the minimization process. The FICP [6] is an improved version of the Trimmed ICP. For the FICP it can be shown, that it converges to a local minimum. Also the overall numerical efforts are smaller than in the Trimmed ICP. As a consequence the FICP was chosen in this paper and tested in chapter 4 for the metal stamping application.

3 Mesh simplification

3.1 Reasons for the Mesh simplification

As the numerical effort for the FICP is still considerable and the algorithm has to be started with several different start conditions to ensure a global optimum it was decided to apply a mesh simplification algorithm. After simplifying the mesh, the FICP will be started based on the reduced point (or element) set. Of course, the mesh simplification must have the property, not to change the typical shape of the geometry, so that the transformation which is determined based on the simplified geometry coincides with the transformation on the full mesh. Mesh simplification is a standard procedure, which is applied in computer graphics to reduce the effort for rendering of objects in far distance to the camera. Figure 1 shows a typical mesh simplification of a geometrical object on different simplification levels starting from 5804 triangles reducing to 64 triangles.

Figure 1: Mesh simplification example [4]

A lot of research work has been done in the last years concerning mesh simplification. For our application the Quadric Error Metric Algorithm (QEM) as described by Garland and Heckbert in 1997 [4] was chosen.

QEM is based on pairwise merging of two points. After merging two points v_1 and v_2 the merged point v_1 is moved to a new position v. All degenerated faces and edges will then be deleted. The decisive part for a good simplification method is the selection criteria for the merging process and the determination of the new position v. The selection process is based on a quadratic form $\Delta(v)=v^TQv$, with v denoting a vertex in barycentric coordinates. The 4x4-Matrix Q is based on some heuristic assumptions as shown in [4]. The resultant algorithm can produce good approximations in fairly short amount of time. In figure 2 a terrain model of crater lake is shown. The fine model consists of 199.114

faces, the reduced model contains only 999 faces. Although the model is reduced considerably, the major details of the original model remain.

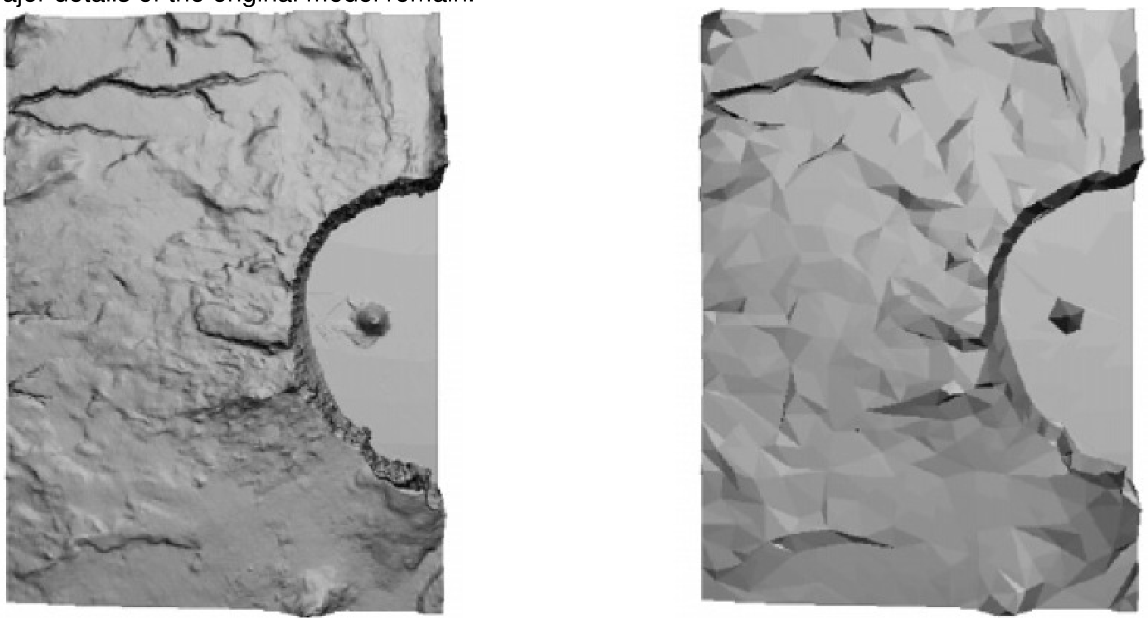

Figure 2: Terrain model of crater lake [4]

4 Testsoftware

To test the algorithms for the purpose of mapping data from stamped parts an appropriate test software was developed. This software can read in LS-DYNA keyword files and calculates the necessary matrices for the transformation. Different parameter settings can be tried and the performance can be measured on different geometries.

4.1 User interface

The only purpose was to investigate different algorithms on a couple of test configurations and to test the performance of the algorithm on geometries with different omplexity. Therefore a simple graphical user interface was designed as shown in figure 3:

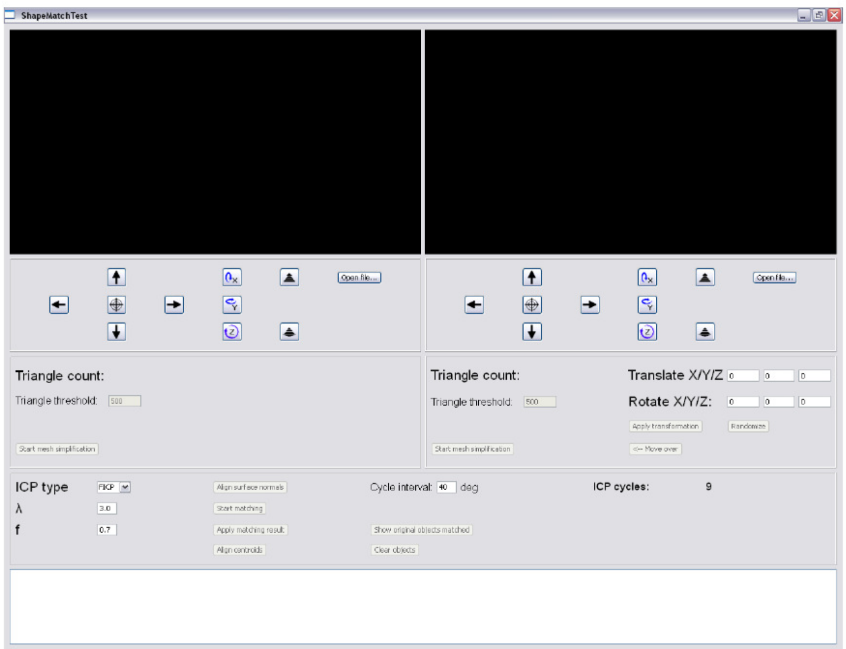

Figure 3: User Interface of ShapeMatchTest

The User interface is divided horizontally into three different sections: The load and display section, the reduce section and the match section. There are two graphic windows in the vertical direction for both the initial geometry and the geometry which is to be matched. The crash part could be loaded and displayed e.g. on the left window, the stamped part on the right window. In the reduce section the geometry could be simplified for performance purposes. To do this the number of triangles, which the simplified mesh should contain as maximum could be put in and the mesh reduction process could be started. A sample geometry of a car floor is shown in figure 4:

Figure 4 Sample geometry of a car floor

After the mesh reduction process is finished, a transformation of the geometry could also be performed by hand. This is only for testing purposes. By clicking “Move over” the two objects will be played into the same graphics window as shown in figure 5:

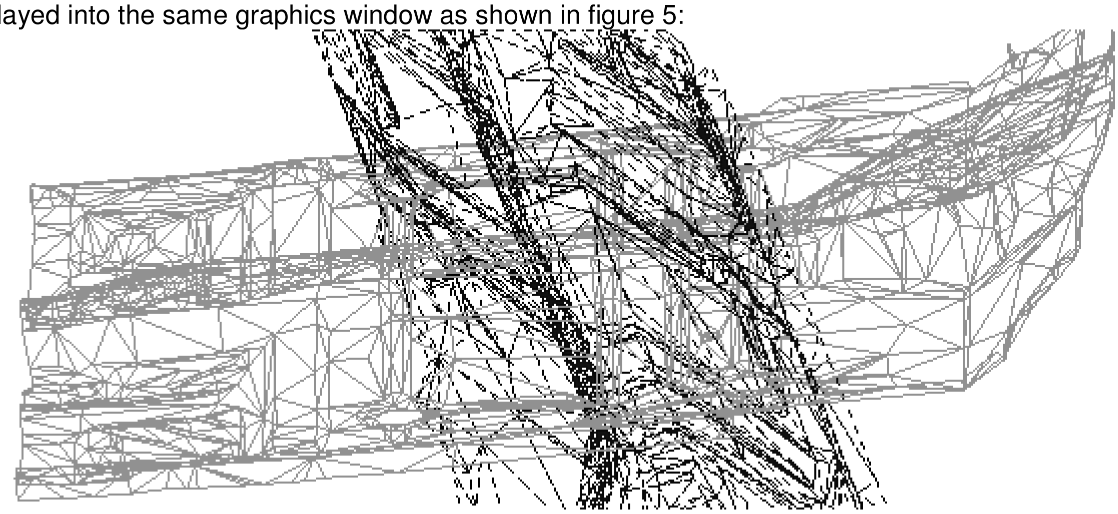

Figure 5: Simplified geometry before matching

To get a good starting point for the ICP algorithm the center of gravity and the mean normal vector of the two objects are being aligned by choosing “align surface normal”. Then the matching process could be started. The result of the matching process could be visibly checked and data about the computing time used for the different steps are being output in the window in the bottom. The results with simplified and detailed geometry are shown in figures 6 and 7:

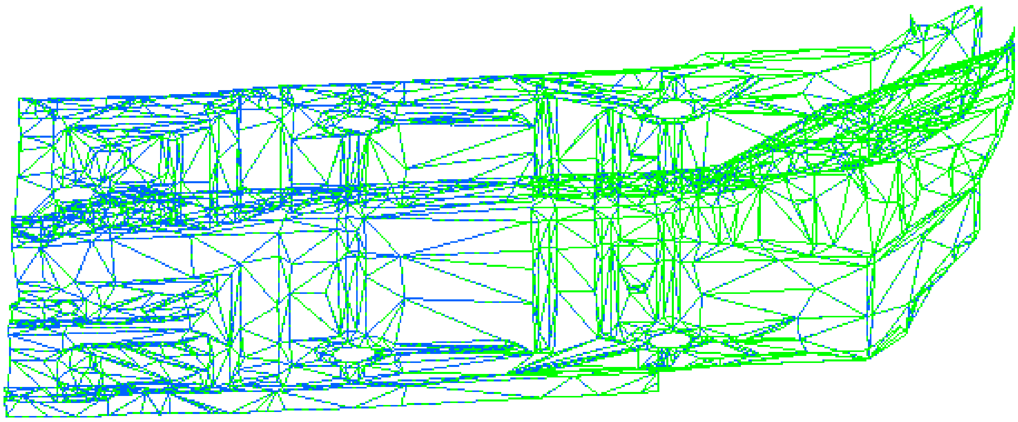

Figure 6 Simplified mesh after matching

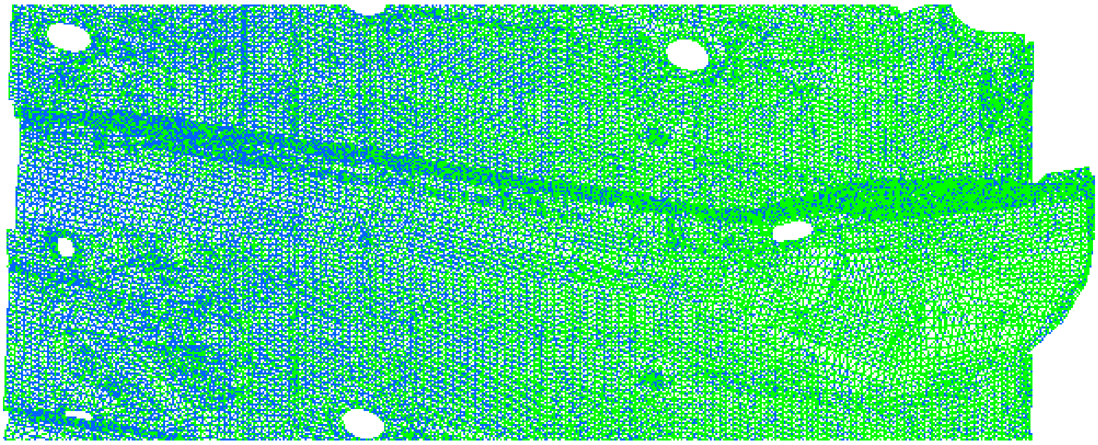

Figure 7: Detailed geometry after matching

4.2 Accuracy

Of course the accuracy is strongly dependent on the simplification level, which is used for the matching algorithm. Also it is important to use at least six different initial configurations to really get to the right local minimum in the underlying optimization process. This means, that the angle must not be chosen larger than 60 degrees. For our tests the number of triangles for the simplified geometry was chosen to 2000 triangles, which is a good compromise between computing time and accuracy.

4.3 Performance

Different geometries have been tested. First the performance of the mesh simplification algorithm was investigated. For an example with 34.808 triangles the following CPU-time was detected:

Reduction to	CPU time in ms
20.000	1.359
10.000	1.406
1.000	1.406
100	1.469

Table 1: CPU time for the mesh simplification of a sample geometry with 34.808 triangles

Now the performance of the matching algorithm itself was tested. For an angle segment of 40 degrees the matching will be started with nine different initial configurations. For the same sample problem the following CPU-time was evaluated:

Number of triangles	CPU time in ms
10.000	117.359
5.000	26.328
1.000	1.015
500	531

Table 2: CPU time for the matching with different simplification levels

As mentioned in section 2.2 the computing costs grow with quadratic order $O(n^2)$, with n the number of triangles. This again confirms the necessity of a proper mesh simplification before applying the matching algorithm.

5 Summary

A shape matching algorithm has been introduced and applied to the problem of mapping thickness and residual strains of stamped parts to crash models. The special algorithm uses mesh simplification and gets good accuracy paired with good performance and can be used for this purpose in future.

6 Literature

[1] Arun, K. S., Huang, T. S. Blostein, S. D.: “Least-squares fitting of two 3d point sets”, IEEE Transactions on Pattern Analysis and Machine Intelligence 9, 1987, pp. 698-700.
[2] Besl, P., McKay, N.: “A Method for Registration of 3-D Shapes”, IEEE Transactions on Pattern Analysis and Machine Intelligence 14, 1992, pp. 239-256.
[3] Chetverikov, D., Svirko, D., Stepanov. D., Krsek, P.: “Robust Euclidean alignment of 3D point sets: the Trimmed Iterative Closest Point algorithm” Image and Vision Computing, 23, 2005, pp. 299-309.
[4] Garland, M., Heckbert, P. S.: “Surface Simplification Using Quadric Error Metrics”, SIGGRAPH 97, 1997.
[5] Horn, B. K. P. “Closed form solutions of absolute orientation using orthonormal matrices”, J. Opt. Soc. Amer. A, vol. 5, No. 7, 1987, pp. 1127-1135.
[6] Horn, B. K. P. “Closed form solutions of absolute orientation using unit quarternions”, J. Opt. Soc. Amer. A, vol. 4, No. 4, 1987, pp. 629-642.
[7] Phillips, J. M. Liu, R., Tomasi, C.: “Outlier Robust ICP for Minimizing Fractional RMSD”, Duke University Technical Report, 2006.
[8] Rusinkiewicz, L., Levoy, M.: „Efficient Variants of the ICP Algorithm”, Third International Conference on 3D Digital Imaging and Modeling (3DIM), 2001.
[9] Umeyama, S.: “Least-Squares Estimation of Transformation Parameters between two Point Patterns”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 1991, p.376-380.
[10] Zöller, A., Frank, Th., Haufe, A.: “Berücksichtigung von Blechformergebnissen in der Crashberechnung”, 3. LS-DYNA Anwenderforum, 2004.