Classification Based 3-D Surface Analysis: Predicting Springback in Sheet Metal Forming

M. Sulaiman Khan, Frans Coenen, Clare Dixon, and Subhieh El-Salhi

Department of Computer Science, Ashton Building, Ashton Street,
Liverpool L69 3BX, United Kingdom
{mskhan,coenen,cldixon,salhi}@liverpool.ac.uk

Abstract. This paper describes an application of data mining, namely classification, with respect to 3-D surface analysis. More specifically in the context of sheet metal forming, especially Asymmetric Incremental Sheet Forming (AISF). The issue with sheet metal forming processes is that their application results in springback, which means that the resulting shape is not necessarily the desired shape. Errors are introduced in a non-linear manner for a variety of reasons, but the main contributor is the geometry of the desired shape. A Local Geometry Matrix (LGM) representation is thus proposed that allows the capture of local 3-D surface geometries in such a way that classifier generators can be effectively applied. The resulting classifier can then be used to predict errors with respect to new surfaces to be manufactured so that some correcting strategy can be applied. The reported evaluation of the proposed technique indicates that excellent results can be produced.

Key words: Classification, Prediction, 3-D Surface Modelling

1 Introduction

Data Mining classification techniques have been applied in many domains using a variety of classifier generators. Much of the original work was directed at the classification of tabular data. Subsequent work focused on more ambitious forms of data such a text, graph and image classification. The current focus is on even more challenging forms of data such as video and 3-D volumes. The work described in this paper is concerned with 3-D surface classification. The challenge with these different forms of classification is not the classification techniques that are used, these tend to be well established, but on the nature of the data preprocessing required to convert the data into a form suited to the application of classifier generators. The data needs to be translated into a format that captures the salient features of the data but at the same time support efficient processing.

In this paper we propose a method for capturing the nature (geometry) of 3-D surfaces in such a way that classification can be applied. More specifically we are interested in data mining techniques for identifying correlations between 3-D surfaces, and then to predict likely correlations with respect to "new" 3-D

surfaces. To act as a focus for the work the investigation is directed at predicting the springback that occurs during Asymmetric Incremental Sheet Forming (AISF); a manufacturing process used to shape sheet metal. The advantages of AISF are that it is comparatively inexpensive and does not require heating of the metal (heating introduces potential fracture points and adds an additional financial overhead). The disadvantage of AISF metal forming is that springback is introduced into the shape. The AISF process commences with a desired *input* shape, defined in terms of a set of 3-D coordinates, and produces an *output* shape which, as a result of the process, is a "variation" of the desired input shape because of the springback that has been introduced. The nature of the resulting output shape can be recorded using an optical measuring system[1] to generate a second set of 3-D coordinates. Thus we have before and after *coordinate clouds* (input and output). Therefore, given a desired shape T, a process P and a result T' we wish to learn the correlation A between T and T' so that given a new shape S we can predict the outcome S' and consequently attempt to redefine S so as to minimise the springback. A simple answer to the problem can be expressed as $A = \frac{T+T'}{2}$. However, the springback introduced by process P is not evenly spread across the entire output shape; it is conjectured by domain experts that the nature of the springback may be dependent on a number of factors such as tool head shape, tool head speed, tool head pitch, lubricant, blank holder, type of alloy, sheet thickness, sheet size, shape geometry and the forming process used. Whatever the case is generally acknowledged that a key influencing factor is the geometry of the desired shape. The nature of the springback (correlation) between T and T' as a result of application of the process P is localised according to the geometry of T (and by extension T').

The proposed technique presented in this paper uses a grid representation for both T and T' so that by registering and superimposing T' over T we can calculate the springback between the two surfaces for each grid point contained in T. We then numerically define the "local surface" surrounding each grid point in T in terms of the change in elevation (the z coordinate) of each of the eight neighbouring grid points compared to the z coordinate of the "centre" grid point. This then gives us a 3×3 Local Geometry Matrix (LGM) for each grid point (except of course at edges and corners) as discussed in Section 5 and shown in table 1 and 3. Any given 3-D surface can then be described in terms of a set of records (one per grid point) such that each record comprises an LGM. If we describe T in this way, and for each record include an error value e obtained by comparing correlated grid points in T and T', we can produce a "training set" set that can be used to train a classifier. The fundamental idea is then, given a new shape S, to use the classifier to predict the springback (S') so that corrective measures can be applied to S to compensate for the springback to give S'' (a corrected definition of S' to be feedback into the AISF process).

For evaluation, a data mining technique is used to predict the springback in sheet metal forming. We evaluated the proposed technique by generating a

[1] In our case the GOM (Gesellschaft fr Optische Messtechnik) optical measuring tool produced by GOM mbH was used.

set of records, using the process described above, and applying a standard Ten-fold Cross Validation (TCV) technique where we built the classifiers using nine tenths of the data and tested on the remaining tenth (using a different tenth as the test set on each occasion). For the evaluation we used a large and a small flat topped square based pyramid. As will be demonstrated later in this paper, the experiments produced excellent results; in some cases a classification best accuracy above 90% was obtained.

The rest of this paper is structured as follows. In section 2 a brief overview of some related previous work is presented. Sections 3 and 4 describe respectively our LGM representation and the mechanism to measure springback between T and T'. The processing of the shape representation to produce a training data set from which classifiers can be generated is described in Section 5. The actual generation of our desired classifiers is then considered in Section 6, followed by the evaluation of the proposed technique in Section 7. Finally some conclusions are presented in Section 8.

2 Previous Work

When manufacturing parts using AISF a metal sheet is clamped into a holder and the desired shaped is produced using the continuous movement of a simple round-headed forming tool. A typical AISF machine is shown in Figure 1. The forming tool is provided with a "tool path" generated by a CAD model and the part is "pressed" out according to the co-ordinates of the tool path. However, due to the nature of the metal used and the manufacturing process *springback* occurs, which means that the geometry of the shaped part is different from the geometry of the desired part, i.e. some springback has been introduced. In [1] the authors consider a number of products that could potentially be formed using AISF and demonstrated that the accuracy of the formed part needs to be improved before this process could be used in a large scale production. In [13] the authors considered two drawbacks of the AISF process relating to the metal thickness and the geometric accuracy of the resulting shape.

There has been substantial reported work on dynamic tool path correction in the context of laser guided tools (see for example [5] and [8]). However, AISF requires that the tool path is specified in advance rather than as the process develops. In [2] the authors propose a multi-stage forming technique, i.e. rather than a single pass by the machine tool, several are made so that the process can take into account the springback. As a case study a square based pyramid shape was considered (similar to those considered in this paper). From [2] it is interesting to note that if the initial geometry comprises corner radii larger than the desired radii, and if a number of forming passes are applied, less springback results then would be encountered otherwise.

For several years the Finite Element Method (FEM) has been used as an industry standard for calculating the springback of sheet metal in forming processes [20]. However, the results of FEM calculations are not very accurate

Fig. 1. Asymmetric Incremental Sheet Forming (AISF), the work piece is clamped in position while the tool head "pushes out" the desired shape, on release springback occurs as a result of which the final shape is not the desired shape

because of the involvement of complex non-linear factors [26]. A data mining approach is advocated in the paper. Not unexpectedly data mining techniques have been applied to sheet metal forming. There are many examples of the use of neural networks to support sheet metal forming [7, 14, 16, 17, 19, 22, 25]. Considering one example only, in [22] a neural network is trained to predict springback. Several inputs were used for the neural network to train on; such as: thickness, radius and springback. It was observed that the predictions made by the neural networks were very close to the simulation results. Rule based learning techniques have also been popular. For example in [27] rule based mining was used to extract knowledge from data generated by Finite Element Analysis (FEA). A four phase knowledge discovery model was proposed that included: (i) product design and development, (ii) data-collection, (iii) knowledge discovery and (iv) management and reuse. In the fourth phase the extracted knowledge was filtered with the aim of supporting the design process. Another similar approach

was proposed in [29] for the U-draw bending process where a rule based system was used to extract knowledge from FEA simulation data. The nature of the material, and various process parameters, were considered to study their effect on springback. However, there has been very little reported work on the use of data mining techniques to address the AISF springback problem as formulated in this paper. The approach proposed advocated here is not only concerned with extracting knowledge from the sheet metal forming data, but also with proposing a classification model that can be used to predict and apply springback errors in order to minimise their effect.

3 Grid Representation

The inputs to the proposed procedure are: (i) an input "coordinate cloud" C_{in} (representing T) and (ii) an output coordinate cloud C_{out} (representing T'). Each coordinate cloud comprises a set of N, (x, y, z) coordinate triples, such that $x, y, z \in \mathbb{R}$. The number of coordinates per cm^2 (within the X-Y plane) in each coordinate cloud varies between 120 points per cm^2 to 20 points per cm^2 depending on how the data is generated/collected. The C_{in} coordinate cloud is typically obtained from a tool path specification generated using a CAD model, while C_{out} is collected using an optical measuring system; $|C_{out}|$ is typically less than $|C_{in}|$. Both coordinate clouds must be registered to the same reference origin and orientation.

We first cast C_{in} into a grid representation (Figure 2) such that each grid point is defined by a $\langle x_i, y_j \rangle$ coordinate value pair. The number of grid lines is defined by some grid spacing d. Each coordinate pair $\langle x_i, y_j \rangle$ in the grid has a z value calculated by averaging the z values associated with the part of the input coordinate cloud contained in the $d \times d$ grid square centered on the point $\langle x_i, y_j \rangle$ (Figure 3). We then cast the C_{out} coordinate cloud into the same grid format so that we end up with two grids, G_{in} and G_{out}, describing the before and after surfaces (T and T').

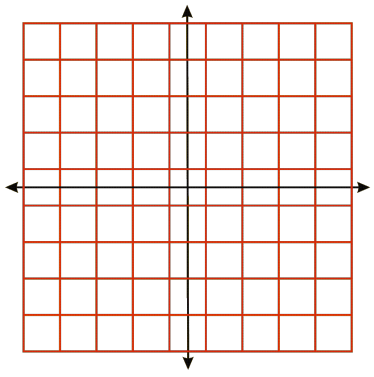

Fig. 2. Example grid referenced to a central origin (grid spacing = d)

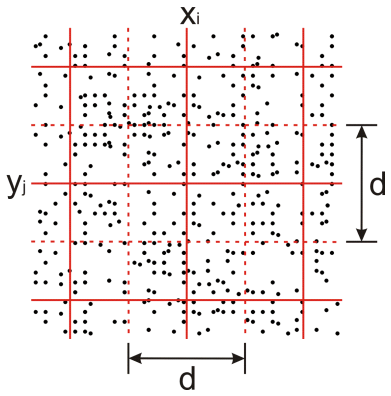

Fig. 3. Coordinate cloud points associated with a grid point $\langle x_i, y_j \rangle$

4 Springback Measurement

A simple mechanism for establishing the degree of springback (e) at a particular grid point is simply to measure difference between the z values in G_{in} and G_{out} (Figure 4). However, a more accurate measure is to determine the length of the surface normal from each grid point in G_{in} to the point where it intersects G_{out}. The distance between any two three dimensional points can be calculated using the point to point Euclidean distance formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{1}$$

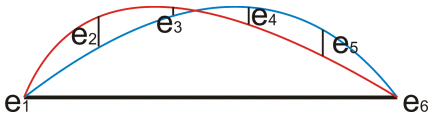

Fig. 4. Cross section at a grid line showing simple vertical springback error calculation between a before and after shape

However, the application of equation (1) first requires knowledge of the x, y, z coordinates of the point where the normal intersects G_{out}. With respect to the work described in this paper we have used the line plane intersection method [9] to determine the length of the normal between two surfaces. Using this approach we find the normal to a plane by calculating the cross product of two orthogonal vectors contained within the plane. Once we have the normal we can calculate the equation for the line that includes the start and end points of the normal and then determine the point at which this line cuts G_{out}. We can then calculate the length of the normal separating the two planes. The process is as follows (with reference to Figure 5):

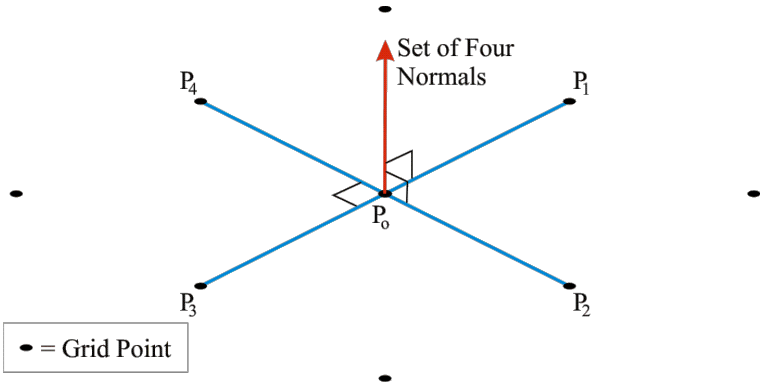

Fig. 5. Error calculation using the line plane intersection method

1. For each grid point in G_{in} first identify the four neighbouring grid points in the X and Y planes as shown in Figure 5 (except at edges and corners where three and two neigbouring grid points will be identified respectively).
2. Define a set of four vectors $V = \{v_1, \ldots, v_4\} = \{\langle p_\emptyset, p_1\rangle, \langle p_\emptyset, p_2\rangle, \langle p_\emptyset, p_3\rangle, \langle p_\emptyset, p_4\rangle\}$, each described in terms of its $x-y-z$ distance from $p_\emptyset$ (the origin for the vector system).
3. Using the four vectors in V, four surface normals are calculated, $N = \{n_1 \ldots n_4\}$, by determine the cross product between each pair of vectors: $v_1 \times v_2$, $v_2 \times v_3$, $v_3 \times v_4$, $v_4 \times v_1$. (Note that to validate a surface normal n_i, the dot product of one of its associated vectors v_j and n_i must be equal to zero, $n_i \cdot v_j = 0$.)
4. For each normal $n_1 \ldots n_4$ calculate the local plane equation in G_{in} that includes $P_\emptyset$ (thus using, in turn, points $\{p_1, p_\emptyset, p_2\}$, $\{p_2, p_\emptyset, p_3\}$, $\{p_3, p_\emptyset, p_4\}$ and $\{p_4, p_\emptyset, p_1\}$). The plane equation is given by Equation 2.

$$ax + by + cz + d = 0 \tag{2}$$

5. For each plane equation identified in (4) determine the parametric equations (a set of equations/functions which describe the x, y and z coordinates of the graph of some line in a plane) [9] of the surface normal as a straight line according to the identities given in equation 3.

$$x = a + i(t), \qquad y = b + j(t), \qquad z = c + k(t) \tag{3}$$

   where t is a constant; a, b and c are the x-y-z coordinates for the point $p_\emptyset$; and i, j and k are the normal components. The constant t is calculated by substituting the parametric equations in plane equation 2 for x, y and z.
6. Once the parametric equations for each surface normal are found, they are then used to compute the points of intersection of each normal with G_{out}.
7. We then use the coordinates for each of the four points of intersection and $p_\emptyset$ to calculate the Euclidean distance (the error) between $p_\emptyset$ and each intersection point to give four error values $E = \{e_1 \ldots e_4\}$
8. We then assign each error a direction (-ve or +ve) based on the direction of the springback. If springback is "downwards", a -ve direction is assigned to the error. Similarly if the springback is "upwards" a +ve direction is assigned to the error. Note that for each point the direction for each of the four errors is same.
9. We now have four error values for each grid point (except at the corners and edges where we will have two or three respectively), we then find the "overall" error e simply by selecting the minimum error that is nearest to zero. The reason for selecting the minimal error is that it gives us the nearest point to the before surface.

On completion of the process our input grid, G_{in}, will comprise a set of (x, y, z) coordinates describing the N grid points, each with an associated springback (error) value e.

5 Surface Representation (The Local Geometry Matrix)

In this section we describe how local geometries can be represented using the concept of a Local Geometry Matrix (LGM). From the foregoing it has already been noted that the value of e is particularly influenced by the nature of the geometry of the desired surface (shape). We can model this according to the change in the δz value of the eight grid points surrounding each grid point. (Of course along the edges and at the corners of the grid we will have fewer neighbouring grid points). Thus we generate n records (where n is the number of grid points) each typically comprising nine values, eight δz values and an associate e value. We, then coarsen the δz values by describing them using qualitative labels taken from a set L to describe the nature of the "slope" in each of the eight neighbouring directions. Therefore we can describe $|L|^8$ different "local geometries" if we take orientation into consideration. Thus if we have a label set $\{negative, level, positive\}$ we can describe $3^8 = 6561$ different local geometries.

Example 1. Considering the flattened square based pyramid shape in Figure 6 and a section of the surface, measuring 3×3 grid points, covering an edge as shown, then the z coordinate matrix associated with the grid point might be as shown in Table 1. The δz values are then calculated by subtracting the centre z value from each of the surrounding z values in turn. With respect to the example the δz matrix result would be as shown in Table 2 (the centre grid reference point always has a value of 0). We refer to this matrix as a Local Geometry Matrix (LGM). Assuming $L = \{negative, level, positive\}$, and ordering the matrix elements (grid points) in a clockwise direction from the top left, would give us a record of the following form where e is the error value associated with the grid point that the record describes:

$$(positive, positive, positive, level, negative, negative, negative, level, e)$$

where e is the error value.

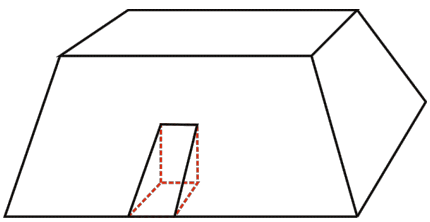

Fig. 6. Square Based Pyramid With Side Section (Example 1)

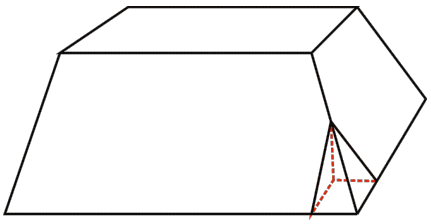

Fig. 7. Square Based Pyramid With Corner Section (Example 2)

Example 2. Again considering a flattened square based pyramid shape but now looking at a section of the surface, measuring 3×3 grid points, located at the corner of the shape as shown in Figure 7, the z coordinates associated with the

20	20	20
10	10	10
0	0	0

Table 1. Z matrix for Example 1

10	10	10
0	0	0
-10	-10	-10

Table 2. LGM for Example 1

grid point might be as shown in Table 3. The LGM would then be as shown in Table 4. Again assuming $L = \{negative, level, positive\}$ the resulting record would be:

$$(positive, level, negative, negative, negative, negative, negative, level, e)$$

20	10	0
10	10	0
0	0	0

Table 3. Z matrix for Example 2

10	0	-10
0	0	-10
-10	-10	-10

Table 4. LGM for Example 2

The proposed representation can be used to capture all local geometries. Given a suitable test shapes (in this paper we have used two flattened square based pyramid shapes, one substantially larger than the other) we can associate an error value with every possible geometry. It should be noted that, at least conceptually, the use of LGMs is akin to the use of Local Binary Patterns (LBPs) as applied in the context of image texture analysis [12, 21].

The set of error values was also discretised using a number of qualitative labels each describing a particular sub-range of error values. The sub-ranges used were of equal size and designed to encompass the full range of error values from the recorded minimum to the recorded maximum.

6 Classifier Generation

There are a number of classification mechanisms that can be applied to data pre-processed in the manner described above, so as to generate a classifier that can be applied to unseen data. In the work described here we favour a classifier that generates rules. Rule base representations offer two principal advantages:

1. Rule representations are intuitive; they are simple to interpret and understand.
2. Because of (1), the validity of rules can be easily verified by domain experts.

It is possible to generate rules using many of the available classifier generation techniques, although some are more suited to rule generation than others. Classification Association Rule (CAR) generators directly generate rule sets. There are a number of well established CAR Mining (CARM) algorithms that can be

adopted: examples include CPAR [28], CMAR [18] and TFPC [3, 4]. Although the principle is the same each of these operates in slightly different manner. It is also fairly straightforward to generate rule sets using decision tree classifiers such as the ID3 Algorithm [24], C4.5 [23] or the MARS Algorithm [11]. Generating rules from Neural Network based classification techniques or Support Vector Machines is less straight forward but can be done [10, 6]. Regardless of the classification algorithm adopted it was assumed that the required input would be in the form of a set of binary valued attributes. Thus for our representation (as describbed above) we will use $|L| \times 8$ attributes plus a number of error attributes. Thus if $|L| = 5$ the input training data will comprise 45 columns, 5×8 attributes plus the class (error) attributes.

6.1 Classifier Application

Once we have generated our desired classifier we will wish to apply it to unseen data, i.e. a new shape S so that we can predict S'. To do this the coordinate cloud describing S must be expressed in terms of its components in the same manner as used to define the training data. Thus the coordinate cloud for S must be expressed as a grid using the same values of d as that was used to generate the classifier, which must then be converted in to a set of records comprising $L \times 8$ attributes so as to be compatible with the generated classification rule representation (again there will be some missing data at edges and corners).

7 Evaluation

This section reports on the outcomes of the evaluation, using a small (SP) and a large (LP) square based pyramid (similar to that used in [2] and [15]), of the proposed approach. The two pyramids were constructed using the AISF process (Figure 10). In each case the before cloud was the CAD generated input to the AISF process. The resulting after clouds were obtained using a GOM optical measuring tool. The objective of the evaluations were:

1. To identify the most appropriate value for d, the grid spacing, so as to maximise the descriptive accuracy of the rules.
2. To identify the most appropriate value for $|L|$, the number of qualitative labels used to describe local geometries, again so as to maximise the descriptive accuracy of the rules.
3. To determine the overall effectiveness of the proposed approach, in terms of classification accuracy.

7.1 Datasets

As already noted the experiments were conducted using two geometries (i) a Small Pyramid (SP) and (ii) a Large Pyramid (LP). Figure 8 shows a square

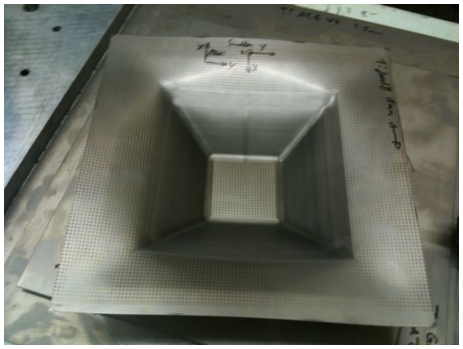

Fig. 8. Square based pyramid (upside down) at the point when it is unclamped after application of the AISF process

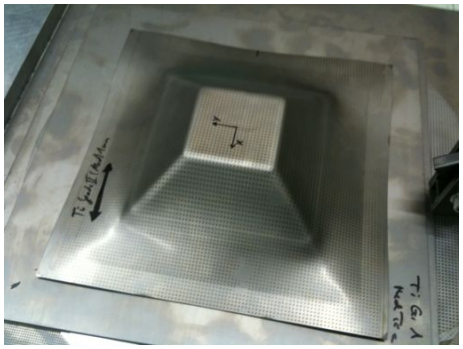

Fig. 9. Square based pyramid (right way up); the markings are used with respect to the GOM optical measuring tool

based pyramid at the point when it was unclamped from the AISF machine, Figure 9 shows the same shape "the right way up". The springback that has been introduced can be observed by inspection of the two figures. The before clouds comprised 24925 and 114888 points respectively. The clouds are shown in Figures 10 and 11. In the case of the large pyramid the surrounding surface that was used to clamp it in the AISF machine was cropped in order to acquire the desired shape as shown in Figure 11. The large pyramid before cropping is shown in Figure 12.

For the reported classification experiments 50 datasets were generated using different combinations of grid sizes d and sets of labels L. Some statistics regarding the size of the resulting data sets are presented in Tables 5 and 6. Table 5 displays the number of records contained in each datasets generated using a range of d values from 1 to 5 (the units are in millimetres). Table 6 shows the number of attributes in each datasets resulting from the use of different $|L|$ values from 3 to 11. The number of record decreases as we increase the grid size because the bigger the grid size the fewer the number of grid points that will be contained within it. Conversely, if the label size (L) increases, the number of attributes increases as shown in Table 5.

7.2 Experiments

In [15] we tested a number of CARM algorithms (CMAR, CPAR, TFPC) and the C4.5 decision tree classifier using Ten-fold Cross Validation (TCV); as a result C4.5 was found to outperform the other classifiers in terms of accuracy. Thus in this paper the reported experiments were conducted using only the C4.5 classification algorithm with TCV. Three sets of experiments were performed to exhibit the applicability of the approach:

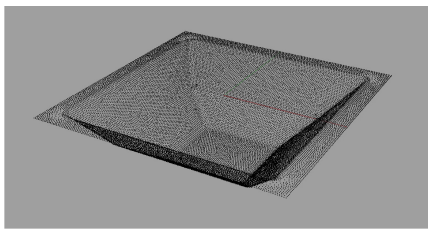

Fig. 10. Before cloud for small square based pyramid

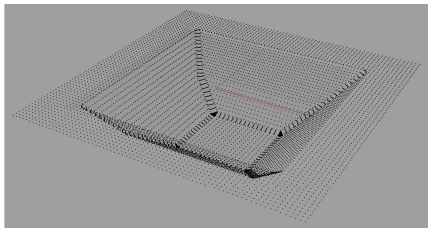

Fig. 11. Before cloud for large square based pyramid (after cropping)

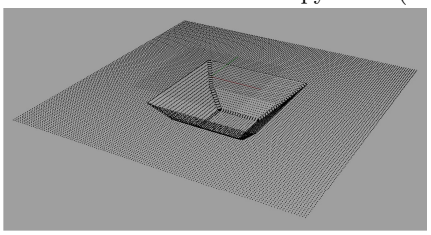

Fig. 12. Before cloud for large square based pyramid before cropping

(d)	Small Pyramid	Large LPyramid
1	19044	37637
2	4624	9216
3	1936	4096
4	1156	2304
5	676	1444

Table 5. Number of records using a range of values for d

(L)	No. of Attributes	No. of Classes
3	24	3
5	40	5
7	56	7
9	72	9
11	88	11

Table 6. Number of attributes using a range of values for L

1. Training and testing the classifier using a single dataset with TCV (for both the small pyramid and the large pyramid datasets).
2. Training the classifier on the small pyramid dataset and testing on the large pyramid datasets.
3. Training the classifier on the large pyramid dataset and testing on the small pyramid datasets.

The last two sets of experiment were conducted to ascertain whether a generically applicable classifier could be produced using the advocated method described in this paper.

With respect to the first set of experiments, Tables 7 and 8 present the classification accuracies and the AUC values for the SP and the LP datasets that were obtained using different combinations of d (1 to 5) and L (3 to 7) values. The results show that the accuracies obtained using the LP datasets are better than those obtained for the SP datasets due to the fact that the

LP datasets featured less springback, thus higher accuracies but lower AUC values. The AUC values indicate that usage of datasets that feature higher error (springback) values results in the generation of classifier with more true positive rules as compared to datasets that feature less springback.

SP	$\|L\|=3$	$\|L\|=5$	$\|L\|=7$	$\|L\|=9$	$\|L\|=11$
	$d=1$				
Accuracy	77.715	68.478	60.875	48.918	53.025
AUC	0.693	0.779	0.772	0.755	0.783
	$d=2$				
Accuracy	74.762	67.645	51.189	56.639	51.384
AUC	0.759	0.847	0.808	0.837	0.816
	$d=3$				
Accuracy	71.281	67.562	58.626	57.283	55.320
AUC	0.717	0.849	0.845	0.853	0.851
	$d=4$				
Accuracy	72.751	73.529	67.734	53.460	54.844
AUC	0.723	0.891	0.870	0.841	0.838
	$d=5$				
Accuracy	71.597	69.231	65.976	63.166	54.734
AUC	0.755	0.865	0.876	0.859	0.869

Table 7. C4.5 TCV Classification Results (Small Pyramid)

LP	$\|L\|=3$	$\|L\|=5$	$\|L\|=7$	$\|L\|=9$	$\|L\|=11$
	$d=1$				
Accuracy	99.939	99.814	99.694	99.529	99.402
AUC	0.454	0.500	0.489	0.494	0.494
	$d=2$				
Accuracy	99.001	98.991	97.536	98.221	97.797
AUC	0.491	0.488	0.495	0.492	0.494
	$d=3$				
Accuracy	99.682	96.899	98.193	96.728	97.094
AUC	0.419	0.491	0.483	0.831	0.495
	$d=4$				
Accuracy	95.529	95.876	95.139	94.791	94.010
AUC	0.490	0.486	0.491	0.496	0.491
	$d=5$				
Accuracy	92.659	91.828	92.245	91.759	91.482
AUC	0.652	0.906	0.906	0.935	0.912

Table 8. C4.5 TCV Classification Results (Large Pyramid)

For the second set of experiments, Table 9 shows the classification accuracy and AUC values obtained when using the SP dataset for training and the LP dataset for testing using different combinations of d (1 to 5) and L (3 to 7) values. Similarly, for the third set of experiments. Table 10 shows the classification accuracy and AUC values obtained when training on the LP dataset and testing on the SP datasets again using different combinations of d (1 to 5) and L (3 to 7) values.

The classification results presented in Tables 9 and 10 demonstrate that high AUC and accuracies values can be achieved for different d and L combinations. From the tables the following can be noted:

1. We can predict the springback (error) to a high level of accuracy (best accuracy of 77% for SP from Table 9, and 99.9% for LP from Table 10).
2. The decision tree classifier worked the best with respect to both pyramids.
3. A high size value for $|L|$ seems to be beneficial (the best value for $|L|$ was $|L|=5$).
4. An argument can be made that a small gird size ($d=3$ or $d=4$) is also beneficial.

The fact that a high value for $|L|$ is beneficial is not suprising because the greater the value of $|L|$ the more expressive the label descriptors. However, if $|L|$ becomes too large there are implications for the runtime complexity of the approach; and, more significantly, may result in "overfitting" of the training

	$\|L\|=3$	$\|L\|=5$	$\|L\|=7$	$\|L\|=9$	$\|L\|=11$
	$d=1$				
Accuracy	99.939	38.067	30.954	23.788	16.027
AUC	0.5	0.359	0.289	0.302	0.395
	$d=2$				
Accuracy	73.459	39.887	23.741	29.514	18.479
AUC	0.475	0.249	0.405	0.3	0.698
	$d=3$				
Accuracy	92.798	96.435	77.807	91.626	65.967
AUC	0.494	0.503	0.552	0.366	0.610
	$d=4$				
Accuracy	6.163	74.523	27.995	10.156	10.937
AUC	0.478	0.5	0.49	0.325	0.764
	$d=5$				
Accuracy	29.917	15.443	62.574	19.598	19.252
AUC	0.440	0.149	0.898	0.638	0.350

Table 9. Classification Results (Training on Small Pyramid and Testing on Large Pyramid)

	$\|L\|=3$	$\|L\|=5$	$\|L\|=7$	$\|L\|=9$	$\|L\|=11$
	$d=1$				
Accuracy	77.667	49.291	35.538	27.972	22.768
AUC	0.5	0.5	0.5	0.5	0.5
	$d=2$				
Accuracy	56.012	42.257	25.562	28.071	19.578
AUC	0.5	0.5	0.5	0.5	0.5
	$d=3$				
Accuracy	55.785	47.675	33.109	31.301	24.483
AUC	0.5	0.576	0.5	0.528	0.5
	$d=4$				
Accuracy	34.429	41.609	22.318	16.263	13.495
AUC	0.349	0.5	0.5	0.433	0.489
	$d=5$				
Accuracy	44.082	40.384	22.337	16.272	14.793
AUC	0.613	0.407	0.473	0.49	0.495

Table 10. Classification Results (Training on Large Pyramid and Testing on Small Pyramid)

data. Overall it can be seen that some very good accuracies and AUC values were obtained. These were very encouraging results. The experiments indicate that we can generate classifiers (as demonstrated) for given shapes, and that this classification approach can provide a sound AI platform for (say) an Intelligent Process Model (IPM) that may be applied in the context of AISF.

8 Conclusions and Perspectives

In this paper we have described a mechanism for discovering correlations between 3-D surfaces. More specifically we have described a mechanism for discovering local correlations between a target shape T and a shape T' produced as a result of the application of an AISF process. We have demonstrated that the mechanism we have proposed to represent local geometries, using the LGM concept, can be used to generate accurate classifiers to predict (and consequently apply) errors in shapes produced using AISF. More generally we have described a 3-D surface representation that accurately describes local geometries in such a way that they are compatible with the effective and efficient generation of classifiers that may be used for prediction purposes.

Given the above it is suggested that classification is an appropriate technology for building Intelligent Process Models (IPMs) for use in AISF (and similar processes). However, we believe our current representation still needs further refinement. Firstly the ranging mechanisms used to discretize LGM values may not be the most appropriate if we wish to apply a classifier built using one shape to another type of shape. It may also be the case that the current representation needs to be augmented with additional information regarding the proximity of grid points to edges and/or corners. The reason for this is that it is conjectured

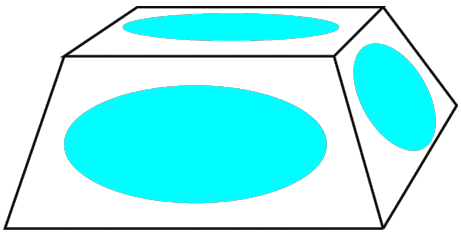

Fig. 13. Areas of greatest springback in a flattened square based pyramid shape

that the error magnitude of the springback increases as we move away from edges (Figure 13). This means that the errors should be greater in the large pyramid than in the small pyramid. Two possible mechanisms whereby we may augment our current representation are suggested. The first involves using two or more d values so that we capture both the "big picture" as well as the "small picture". Alternatively we can include an edge/corner proximity measure (p). Currently we describe shapes using a grid. For each grid point (except at edges and corners) we have eight surrounding grid points. We have established that local geometry can be described by the difference in z values between the center grid points and the surrounding eight points. In each case this gives a 3×3 Local Geometry Matrix (LGM) describing the δz values (with the value 0 at the center representing the grid point). Some of these LGM configurations will indicate the presence of edges and corners provided that the grid distance (d) is sufficient to capture this. Given a "bank" of LGMs describing edge and corner configurations we can use pattern matching to identify the corners and edges in any given piece. We can then use this knowledge to determine values for p for each grid point. The long term goal is to produce a generally applicable classifier that can be applied to any shape (of course other influencing factors such as material and tool head speed must be kept constant).

Currently errors are defined as the distance along the normal from the before surface to where it intersects the after surface. We calculate four normals for each grid point and consequently four error values are obtained. The specific error associated with a grid point is then the minimum of these four error values. To produce a new coordinate cloud, S'', we can simply reverse these errors. The reverse errors can either be applied to the before grid points or directly to the before coordinate cloud. If we apply the error to the coordinate cloud and if there is a significant difference between the error associated with adjacent grid points, we may get a "stepping" effect (especially if d is large); in which case some sort of smoothing may be required. If we apply the error to the grid coordinates we may not have sufficient points to allow a new shape to be manufactured. We will therefore need to use small values of d, $d = 1$ seems to be a good value. It should also be noted that we believe that simply reversing the error is unlikely to produce a good S'', we therefore propose to apply a factor f to the errors. The intention is that the nature of f will be dependent on the local geometry

as defined so far, but augmented by the additional work on representing local geometries (as described above) that we intend to undertake.

9 Acknowledgements

The research leading to the results presented in this paper has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 266208. The authors would particularly like to thank Markus Bambach, Babak Taleb and David Bailly, from RWTH-IBF (Germany) for their support in the preparation and provision of the test data used to evaluate the proposed mechanism described in this paper. The authors would also like to thank Mariluz Penalva, Asun Rivero, Antonio Rubio and Boto Sanchez Fernando from Tecnalia-IS (Spain) for comments on an earlier draft of this paper; and Nicolas Guegan from AIRBUS (France) and Joachim Zettler from EADS (Germany) for their extremely helpful advice on various aspects of the work described.

References

1. J. M. Allwood, G. P. F. King, and J. Duflou. A structured search for applications of the incremental sheet-forming process by product segmentation. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(2):239–244, 2005.
2. M. Bambach, B. Taleb Araghi, and G. Hirt. Strategies to improve the geometric accuracy in asymmetric single point incremental forming. *Production Engineering Research and Development*, 3(2):145–156, 2009.
3. F. Coenen and P. Leng. Obtaining best parameter values for accurate classification. In *Proc. IEEE Int. Conf. on Data Mining (ICDM' 05)*, pages 597–600, 2005.
4. F. Coenen, P. Leng, and L. Zhang. Threshold tuning for improved classification association rule mining. In *Proc. PAKDD 2005, Springer LNAI3158*, pages 216–225, 2005.
5. G. Dearden, S.P. Edwardson, E. Abed, K. Bartkowiak, and K.G. Watkins. Correction of distortion and design shape in aluminium structures using laser forming. In *25th International Congress on Applications of Lasers and Electro Optics(ICALEO 2006)*, pages 813–817, 2006.
6. J. Diederich. *Rule extraction from support vector machines*, volume 80. heidelberg, 2008.
7. S. Dunston, S. Ranjithan, and E. Bernold. Neural network model for the automated control of springback in rebars. *IEEE Expert: Intelligent Systems and Their Applications*, pages 45–49, 1996.
8. S.P. Edwardson, K.G. Watkins, G. Dearden, and J. Magee. Generation of 3D shapes using a laser forming technique. In *Proceedings of ICALEO'2001*, pages 2–5, 2001.
9. P.A. Egerton and W W. Hall. *Computer graphics: Mathematical first steps*. Simon and Schuster International, 1998.
10. A.E. Elalfi, R. Haque, and M.E. Elalami. Extracting rules from trained neural network using GA for managing e-business. *Applied Soft Computing*, 4(1):65–77, 2004.

11. J.H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.
12. G. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.
13. G. Hirt, J. Ames, M. Bambach, R. Kopp, and R. Kopp. Forming strategies and process modelling for CNC incremental sheet forming. *CIRP Annals - Manufacturing Technology*, 53(1):203–206, 2004.
14. M. Inamdar, P.P. Date, K Narasimhan, S.K. Maiti, and U.P. Singh. Development of an articial neural network to predict springback in Air Vee bending. *International Journal of Advanced Manufacturing Technology*, 16(5):376–381, 2000.
15. M. S. Khan, F. Coenen, C. Dixon, and S. El-Salhi. Finding correlations between 3-d surfaces: A study in asymmetric incremental sheet forming. In *Proc. Machine Learning and Data Mining in Pattern Recognition (MLDM'12), Springer LNAI 7376*, pages 336–379, 2012.
16. D.J. Kim and B.M. Kim. Application of neural network and FEM for metal forming processes. *International Journal of Machine Tools and Manufacture*, 40(6):911–925, 1999.
17. B. Kinsey, J. Cao, and S. Solla. Consistent and minimal springback using a stepped binder force trajectory and neural network control. *Journal of Engineering Materials and Technology*, 122(1113):113–118, 2000.
18. W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proc. IEEE Int. Conf. on Data Mining (ICDM' 05)*, pages 369–376, 2001.
19. K. Manabe, M. Yang, and S. Yoshihara. Artificial intelligence identification of process parameters and adaptive control system for deep drawing process. *Journal of Materials Processing Technology*, 80-81:421–426, 1998.
20. N. Narasimhan and M. Lovell. Predicting springback in sheet metal forming an explicit to implicit sequential solution procedure. *Finite Elements in Analysis and Design*, 33(1):29–42, 1999.
21. T. Ojala, M.P. Inen, and T. Maéenpaé. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.
22. K.K. Pathak, S. Panthi, and N. Ramakrishnan. Application of neural network in sheet metal bending process. *Defence Science Journal*, 55(2):125–131, 2005.
23. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.
24. J.R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
25. R. Rufni and J. Cao. Using neural network for springback minimization in a channel forming process. *Journal of Materials and Manufacturing*, 107(5):65–73, 1998.
26. J. Xu, Z. Zhang, and Y. Wu. Application of data mining method to improve the accuracy of springback prediction in sheet metal forming. *Journal of Shanghai University (English Edition)*, 8(3):348–353, 2004.
27. J.L. Yin and D.Y. Li. Knowledge discovery from finite element simulation data. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pages 1335–1340, 2004.
28. X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *SIAM Int. Conf. on Data Mining (SDM'03)*, pages 331–335, 2003.

29. S. Zhang, C. Luo, Y.H. Peng, D.Y. Li, and H.B. Yang. Study on factors affecting springback and application of data mining in springback analysis. *Journal of Shanghai Jiaotong University*, E-8(2):192–196, 2003.