Semi-Automatic Manufacturing of Customized Hearing Aids Using a Feature Driven Rule-based Framework

K. Sickel[1], S. Baloch[2], V. Bubnik[2], R. Melkisetoglu[2],
S. Azernikov[2], T. Fang[2], J. Hornegger[1]

[1]Pattern Recognition Lab, Department of Computer Science,
Friedrich-Alexander-University Erlangen-Nuremberg, Martensstr. 3, 91058 Erlangen, Germany
[2]Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, USA
Email: konrad.sickel@informatik.uni-erlangen.de

Abstract

We present a semi-automatic solution to the problem of customized 3-D shape modeling using a rule based system. The solution is illustrated with the use case of hearing aid manufacturing. The current process of manufacturing customized hearing aids consists of many manual steps, all of which require expert knowledge. The idea is to translate this expert knowledge into machine readable rules using a procedural knowledge representation. Together with a set of anatomical features detected on the ear impression, the necessary surface shaping operations are defined and applied. The advantage of this approach is its high flexibility and customisability. It yields better consistency, efficiency and reproducibility of the resulting device. We validate the approach in a real production environment.

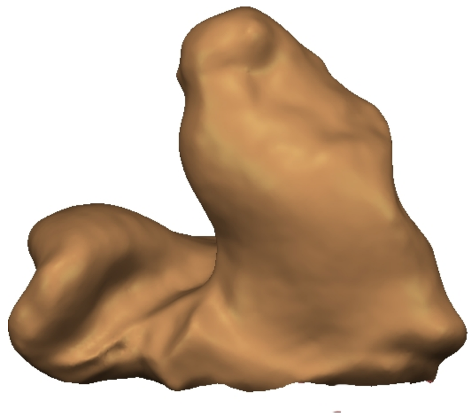

Figure 1: Image of a 3-D raw ear impression.

1 Introduction

This paper addresses the problem of modifying shapes in a rapid prototyping environment. The goal is to transform a raw 3-D impression U of the ear canal (Figure 1) into the shape $D = T(U)$ of a hearing aid (HA), see Figure 2 for examples. The applied transform T is constrained in numerous ways. The first constraint is the preservation of anatomical features to ensure a good fit in the ear of a patient. Second, the impression must maintain enough space inside for the placement of electronic components, such as receiver and faceplate. For wearing comfort, a ventilation tube (vent) has to be integrated. Furthermore the form of the target device type has to be specified. Hence, the current so-called detailing and modeling is a challenging, tedious and time consuming manual process. The complexity of the problem is further increased by the high variability of ear shapes, which has prevented automation of the process so far. Special modeling software [9] provides a workspace to an experienced operator to perform a sequence of operations, such as cutting and rounding. In addition, it offers tools to place the electronic components and the vent. This virtual placement of components ensures that the device is built with electronics suitable for the patient's needs of amplification. The size of the electronic components, in turn, is related to their amplification power. Consequently, patients with major hearing loss may only have larger devices than patients with minor hearing loss.

Recent works on shape modification for hearing aid manufacturing concentrate on the active shape model [1] approach. The active shape model allows

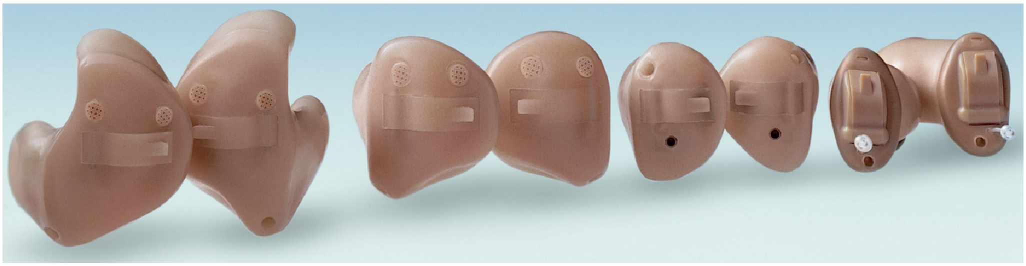

Figure 2: Starting from the left an in-the-ear (ITE), an in-the-canal (ITC), a smaller ITC and a completely-in-the-canal (CIC) hearing aid device is shown.

a compact description of the variation of shapes in a class using statistical methods. One of the first approaches to this problem was proposed by Paulsen et al. [3]. They computed a correspondence field between shapes using a template annotated by experts. To create smoother and denser meshes a Markov field regularization on the correspondence field was introduced in [4]. These works focus on analyzing gender difference in the shape of the ear canal as well as the influence of mandibular movement [5] on the human ear canal.

Application-wise our work is closely related to [2], where Unal et al. used statistical shape learning to transform an undetailed (unprocessed) impression into a detailed shell (finished impression). Their system learns the relationship between the two classes without expert involvement. The computed relationship was then used to detail a new sample by carrying out a registration with a mean shape and applying a weighted transformation. The limitation of this approach is that due to its global nature it offers little control for details. Hence, it fails to take the component and vent placement into account. Furthermore the framework can only be trained for one device type. Hearing aid manufactures offer around 3 to 5 different device types each with hundreds of options, which typically define the shape of the HA as shown in Figure 2.

The goal is to automate the entire detailing and modeling process. Due to the complexity of the problem, we present a semi-automatic solution as a first step. In contrast to the previous work, we determine the shape modifications based on knowledge of expert operators.

The operators use written work instructions and their experience to detail and model an impression to its final shape. Our semi-automatic framework is integrated into a modeling software and presents a proposal for every process step. This allows a very detailed control of the applied shape modifications and hence incorporation of different rules for the device types and the numerous options. The presented proposal may be interactively adjusted by the operator, which ensures the quality of the result.

The automation of the process aims at enhancing the efficiency, consistency and repeatability of the detailing and modeling process as well as improving the wearing comfort. The main contribution of this paper is the presentation of a novel approach for the problem of automatic shape modification. Our framework differs significantly from other research by its local step wise nature and direct usage of expert knowledge.

We extend an existing modeling software by incorporating a robust feature detection in Section 2.1, that is specifically adapted to ear impressions. To utilize the expert knowledge we add a simple yet powerful script language parser and interpreter in Section 2.2. The script language provides us with the needed flexibility to compose a rule base, capable of defining the complex shaping process for all device types and options. The framework is general and may be used for other similar shape modification problems by adapting the features and customizing the underlying rule system. Experiments are reported and discussed in Section 3 before concluding with a summary and future directions in Section 4.

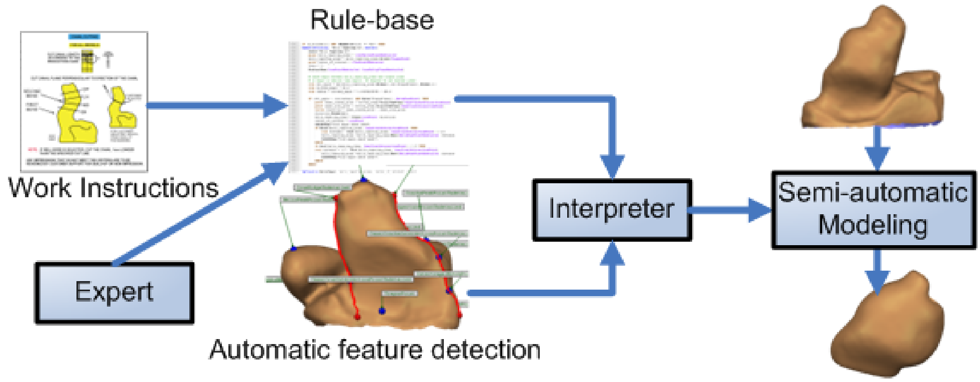

Figure 3: Framework diagram: The rule-base contains the process knowledge of the work instructions and the experts. In combination with the detected features the interpreter executes the rules and controls the modeling software.

2 Method

Currently, HA manufacturing is done by experienced operators who use a modeling software to shape an impression to its final form. The process of shell shaping is carried out by defining various rounding and cutting planes as well as shrinking or growing parts of the impression. This is followed by the placement of several electronic components and a ventilation tube. The vent reduces the occlusion effect, which would be recognized by a completely blocked ear canal.

The shaping process is very time consuming and due to the high variation in shape and options, it is prone to process failures and difficult to work consistent or reproducible. Even an expert will design the shell differently if presented with the same task again. Our work aims at removing this bottleneck by introducing a semi-automatic framework, which mimics the experienced operator. This is achieved by using a rule-base to define the shaping as well as the component placement operations. The rules adapt to the shape variations by using on the fly detected features. An overview of the framework is given in Figure 3.

2.1 Feature detection

Our rule based framework works with a vector of features $\Phi = (\mathbf{x_1}, \ldots, \mathbf{x_n})^T$ detected on the impression. The features are composed of the standard anatomical features of the ear as well as some additional features used in the modeling process pointed out by experts (Figure 4). There are point, plane, curve and region features. The resulting feature vector contains 60 elements.

In other words, given a 2-D surface U in the three dimensional Euclidean space, an application of feature functions ϕ_i yields the feature vector $\Phi(U)$.

$$\Phi(U) = (\phi_i(U), \ldots, \phi_n(U))^T \qquad (1)$$

Robust feature detection on ear shapes is a challenging task. Despite their similarity, ear impressions possess a wide variability of shapes. Therefore we have developed a set of general detection algorithms. These algorithms detect concavities, elbows, ridges, peaks and bumps on a surface. These general algorithms, in turn, are adapted to detect an anatomical feature. The details of these algorithms are explained in [6]. The basic ideas for the algorithms are as follows: Peak detection is done by using level sets of a height function in accordance with the *Morse deformation lemma*[11]. Concavity detection is done by generating a surface profile with splines and analyzing this profile with respect to variations in the signed curvature. Elbows are marked by high curvature detected in combination with a region of interest and a skeletal representation of the shape. Ridge detection identifies

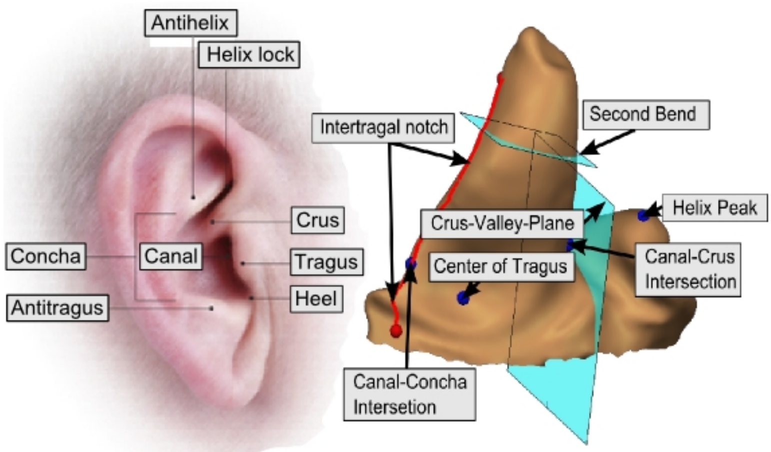

Figure 4: On the left: Picture of the ear with anatomical features [10]. On the right: Impression with some of our detected features displayed.

geodesics of high curvature points. Bump detection also is based on a surface profile using cross-sectional scans. First a rough bump contour is computed which is then refined with an iterative approach.

The validation of the feature detection algorithms was carried out by labeling a set of 130 impressions. The algorithms were tweaked until they reached a detection rate of at least 95 percent and a detection quality of above 80 percent. The detection quality is defined for each feature type separately. E.g. the detected points must be in a certain range of the labeled feature to count as correctly detected. Similar measures are used for the other feature types.

2.2 Rule-base and script language interpreter

The expert knowledge is represented in *if-then-else* rules. This procedural knowledge representation gives us, in contrast to a declarative representation, the possibility to define rules which match the manual process and can be understood by process experts. Despite its simplicity the representation is powerful enough to encode the knowledge for the shell shaping process [7]. To transcribe the rules we developed a simple script language with a context free grammar similar to PASCAL.

The script language supports the standard data types, like booleans, integers, floats, strings and arrays of all types. In addition 3-D points, planes and matrices are added as special data types. For each data type the standard calculation and comparison operators are available, which allow vector and matrix calculations in the script. The script language supports several control structures. It is possible to branch with if-then-else blocks and to include for, while and repeat-until loops. The script parser and interpreter were implemented with the tools bison and flex [8].

The decision to develop a new scripting language was made to simplify the integration in the existing modeling software system. It allows us to include any functionality we need and keep it as simple as possible. In addition, extension of the scripting language functionality is straightforward.

The framework is integrated into the shell shaping software in the form of a guide. It guides an operator through the process steps and shows in every step the necessary tools as well as a suggested operation. For example, the first step is the same for all device types. The operator is shown a cutting plane which separates unnecessary material at the bottom from the impression. It also preselects the appropriate cutting tool and the operator only needs to verify the correctness of the plane and press the apply button. This will automatically apply the cut and set up the next guide step. Each step can be done either for all devices, only for a special device type or only for a special option. To support the guide functionality the script includes a number of predefined functions, which can be divided in five groups.

The *tool* functions provide the interface to the shaping and component placement functions of the modeling software. For instance integration of the vent by specifying start and end point. The *geometric* functions allow the modification of points and planes. Examples are moving a plane to a certain point or about a certain distance, projecting a point onto a plane, computing the contour of a plane with the impression. *Parameter* functions allow access to the detected features and to the specific options for a device, like device or vent style. *Visualization* functions allow the script to set up the working environment for an operator, such as switching the transparency of the impression on in cases where components are placed. The last group of functions allows the rules to *communicate* with the operator by giving information about the guide step like the used options or encountered problems. In addition to these functions, it is possible to define functions and procedures in the script language itself. These internal script functions are used to improve the readability of the rule base, for instance by encapsulating the knowledge about the applicableness of a rule. Altogether this allows us to compose our rule base $\mathcal{R}$.

With our rule-base $\mathcal{R} = \{r_1, \ldots, r_s\}$, the given options $\mathcal{O} = \{o_0, \ldots, o_v\}$ and the detected features Φ the transform $D = T(U)$ can be rewritten as a sequence of rule executions on the input impression:

$$\begin{aligned} U_{i+1} &= r_i(U_i, \Phi(U_i), \mathcal{O}),\ i = 1, \ldots, t & (2) \\ D &= U_{t+1} & (3) \end{aligned}$$

The option vector $\mathcal{O}$ contains the information about the target device, like device type, receiver type and vent style. The indexed variable $t \neq s$, because for every device type some rules will not be applicable.

Our rule-base $\mathcal{R}$ is structured in blocks of associated rules. Every rule block is responsible for a guide step. Each block sets up the necessary tools as well as the proposal of the guide. A simplified example rule for the so called crus cutting is given below.

```
01 FUNCTION is_ITC() : bool
02  IF Option(1) == ITC
03  OR Option(1) == LC
04  (OR Option(1) == CIC
05  AND Option(5) == C) THEN
06   RETURN true
07  ENDIF
08  RETURN false
09 ENDFUNCTION
10
11 FUNCTION funcCC(plane input) : plane
12  plane r = input
13  IF Dist(input, CCIPoint) <= 2.0 THEN
14   r = r.Move(HelixPoint, 2.0)
15  ENDIF
16  RETURN r
17 ENDFUNCTION
18
19 IF is_ITC() THEN
20  OpenGuideStep(Shaping, "Crus Cut")
21    plane cutting_plane =  \\
      funcCC(CrusValleyPlane)
22    int level = 3
23  CloseGuideStep(Round, \\
    cutting_plane, level)
24 ENDIF
```

The example shows the interaction between options, features, functions and guiding rules. The first function `is_ITC()` (line 01 – 09) returns true if the current device is an ITC device, which depends on different option code combinations. The second function `funcCC` modifies a given plane if needed – in this case if a distance is below a certain threshold. The modification is done by translating the plane along its normal about 2mm in direction of the HelixPoint. The last part of the example shows a guide step – Crus Cut, which makes use of the previous two functions. The middle sized ITC devices are cut at the crus valley plane, like shown in Figure 5. Therefore a plane is defined and rounding with the plane is applied. The rule will only be activated if the target device is an ITC (line 19). The guide step sets up the correct working environment (line 20). We distinguish between general *shaping* steps and special steps like vent integration and component placement. In the next line (21) the plane is defined with help of the CrusValleyPlane feature and a script function. This plane will be shown to the operator. Most tools of the modeling software have different levels, in this case level 3 will be used (line 22). The level influences the smoothness of the rounding and the affected area (shaded area in Figure 5). The following line (23) finalizes the guide block by specifying the tool and the parameters.

Our rule base currently consists of 44 manually defined guide steps. To detail and model an impression between 15 to 25 of them are required, depending on device type and chosen options. To be able to improve the rule base in a structured way the result of each step is stored in a xml file. The file contains the setup and the commit phase of each rule. This allows the analysis of the modifications done by an operator for potential future improvements in the rules.

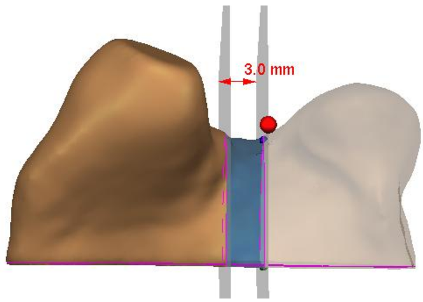

Figure 5: Example of a *Crus Cut* done for all ITC devices.

3 Results and Discussion

For testing the usability and performance of our semi-automatic shell shaping framework we used a set of 39 different ear impressions in combination with 13 different device type option combinations. Six experts processed all 39 samples. The operators were advised to make corrections if necessary. As a result the finished devices are qualitatively well done. Therefore, to analyze the performance of our framework we developed a quality matrix. This matrix enables the expert operators to assign a value for each guide step for each processed sample. The value is 0 for unusable, 1 for usable with modifications, 2 for acceptable without modification and 3 for perfect. The results are given in Table 1.

The overall quality level is very promising. On average the guide proposes an acceptable solution for each step. For full automation it would be necessary to reach at least a mean value above 2 with a small standard deviation. In this first test *automation* was reached for about 14 percent of the test cases (every script step was rated ≥ 2). The median value is ≥ 2 for 75 percent of the steps, which as well shows that an acceptable performance is reached in most cases.

The results are preliminary and need to be interpreted with care. In our feedback sessions with the experts we recognized, that there was a misunderstanding for some steps between what the framework does and what the operator expected. This, in turn, yields to a lower quality level, e.g. optional cuts. Most of the guide steps perform well on average, but need to be improved in terms of the standard deviation, e.g. ITE anti-helix filling. A minor subset of the rules (crus scooping, CIC measured cut and ITE measured cut) suffers from the fact that the training set did not cover the shell variability found in the validation set.

The quality of the resulting device was comparable to a device done completely manually. There were small differences in size and shape. According to the experts a variation in size of about 2mm can be expected between operators. Even in the case that an operator does a device twice. The resulting devices manufactured using our framework were always in this range and visually more consistent between different operators.

The overall feedback by the experts was very positive. It is a great help that the guide points out the needed options and exactly follows the optimal process. This is very well appreciated by the process trainers as well as new operators, which are not familiar with the process yet.

The rule base itself has still a size, which can be handled manually. If an expert suggests a modification or a new rule this can typically be implemented in less than one hour. Due to the high shape variability the testing and refining of the rule to its final form takes approximately four hours (strongly dependent on the rule itself).

4 Summary and Conclusion

We presented a general framework for semi-automatic generation of a target shape from a new input shape via applying rules driven by detected features on the input shape.

As a specific application, our proposed frameworks is able to design a hearing aid device given an undetailed impression of the ear and the chosen device options. The framework guides an operator through the shaping and the component placement process. In each process step the guide presents a suggestion for the current task. The operator may interactively modify or accept the proposal. This semi-automatic approach allows the usage of the automation framework in a real production environ-

Step name	$\mu\,(\pm\sigma)$	$\tilde{x}$	Step name	$\mu\,(\pm\sigma)$	$\tilde{x}$
All steps	**1.96 (0.95)**	**2**	Canal thickening	1.72 (1.13)	2
Canal tip cut	1.74 (0.98)	2	CIC measured cut	1.14 (0.81)	1
Excess material cut	2.21 (0.86)	2	ITC crus cut	2.45 (0.78)	3
ITC measured cut	2.10 (1.01)	2	ITE anti-helix filling	2.25 (1.10)	3
ITE crus scooping	1.07 (1.14)	1	ITE cymba rounding	2.43 (0.77)	3
ITE measured cut	1.58 (0.93)	1	Labeling	1.86 (1.26)	2
Optional cuts	1.68 (0.87)	1	Optional vent cuts	2.60 (0.76)	3
Receiver hole placement	1.97 (1.06)	2	Vent placement	2.04 (0.93)	2
Waxguard cut	2.57 (0.86)	3			

Table 1: The table shows quality values given by expert operators using the script guide to shape a hearing device. For the sake of clarity and space, similar steps are grouped together. μ is the mean quality value of all operators for the step, σ the standard deviation and $\tilde{x}$ the median.

ment by maintaining the necessary quality of the result. Since recently the framework is in use by one of the major HA manufacturers. It increases the efficiency, consistency, reproducibility and quality of the design while ensuring the processing of each device option. The time to model a HA within our framework was (after a training phase) more or less the same.

Our framework is quite general, every mapping between two classes which can be expressed by a set of feature driven rules can be addressed by it. The framework is currently in its first phase. We are acquiring further data on its performance. For a better evaluation, we plan to develop a tool that analyzes the modifications done by an operator. This will allow a more detailed evaluation of the framework, which will enhance the performance of the rules either by manual modification or automatic inference.

The final target is to fully automate the entire process. Therefore the framework will be extended by expanding the rule base and improving the feature detection. It is questionable if a handcrafted rule base is sufficient to reach full automation in a challenging environment like customized hearing aid manufacturing. Hence, our next steps will be the analysis of the difference between the expert system and an expert and how to use this knowledge to automatically extend and improve the rule base. In addition, we plan to integrate shape clustering to achieve a better adaption to the ear variability and also to employ statistical shape models for steps in the process which cannot be compactly expressed by rules.

Acknowledgments We are grateful to Siemens for supporting this work. We are also grateful to the anonymous reviewers for both suggesting and inspiring some additional notes and numbers which have strengthened this paper.

References

[1] T. F. Cootes and C. J. Taylor and D. H. Cooper and J. Graham. Active shape models and their training and application. *Computer Vision and Image Understanding,* 61(1):38–59, 1995.

[2] G. Unal and D. Nain and G. Slabaugh and T. Fang. Customized Design of Hearing Aids Using Statistical Shape Learning. In *11th Int. Con. on Medical Image Computing and Computer Assisted Intervention (MICCAI),* 518–526, 2008.

[3] R. R. Paulsen and R. Larsen and S. Laugesen and C. Nielsen and B. K. Ersboll. Building and Testing a Statistical Shape Model of the Human Ear Canal. In *5th Int. Con. on Medical Image Computing and Computer Assisted Intervention (MICCAI),* 373–380, 2002.

[4] R. R. Paulsen and K. B. Hilger. Shape Modelling Using Markov Random Field Restoration of Point Correspondences. In *18th Int. Con. on Information Processing in Medical Imaging (IPMI),* 1–12, 2003.

[5] S. Darkner and R. Larsen and R. R. Paulsen. Analysis of Deformation of the Human Ear and Canal Caused by Mandibular Movement. In *10th Int. Con. on Medical Image Computing and Computer Assisted Intervention (MICCAI),* 801–808, 2007.

[6] S. Baloch and K. Sickel and R. Melkisetoglu and S. Azernikov and T. Fang. Automatic detection of anatomical features on 3D ear im-

pressions for canonical representation. To be submitted.

[7] J. C. Giarratano and G. D. Riley. *Expert Systems: Principles and Programming.* Course Technology, 1998.

[8] C. Donnelly and R. M. Stallman. *The Bison Manual, Using the YACC Compatible Parser Generator.* Free Software Foundation, 2003.

[9] 3Shape A/S. 3Shape Hearing Aids, ShellDesigner. http://www.3shape.com/our-products/hearing-instruments/shelldesigner.aspx, 06/2009.

[10] H. Gray. Gray's anatomy of the human body (1918). http://www.bartleby.com/107/229.html, 06/2009.

[11] Y. Matsumoto. *An Introduction to Morse Theory.* American Mathematical Society, 2001.