Improvement of an Algorithm for Non-rigid Adjustment of Ear Geometries for Personalised Spatial Hearing

Bachelor thesis from Aufnahmetechnik 1, SE

Dr. Thomas Weiß
Betreuung: Univ.Prof. DI Dr. Alois Sontacchi, Doz. Dr. Piotr Majdak, DI Katharina Pollack

Graz, June 2023

institut für elektronische musik und akustik

Abstract

The acquisition of individual head-related transfer functions is the key step in achieving fully three-dimensional audio perception via headphones, especially localisation in both azimuth and elevation angle. To avoid the necessity for specialised equipment faced by current high-quality acquisition methods, numerical HRTF calculation on the basis of photogrammetrically reconstructed 3D models of the pinna could be a solution. To enhance their insufficient quality, a non-rigid registration (NRR) algorithm is used to map a clean template ear onto the photogrammetrically reconstructed ear. This thesis shows how a state-of-the-art NRR algorithm can be modified to make use of confidence levels, which are an additional output of the photogrammetry algorithm, and aims to evaluate if this could improve the registration result.

Contents

1 Introduction

In recent times plausible spatial audio reproduction via headphones has gained a lot of attention. One of the reasons is the increased interest in virtual reality, where the significance of natural spatiality perception is far higher than in conventional listening situations. Also, augmented reality benefits greatly from the ability to properly place synthetic sound sources at specific directions from the listener.

Lateral localisation, i.e. if a sound is coming from the left or the right, is a result of the differences of incoming sound between the left and the right ear, specifically the interaural time difference (ITD) and the interaural level difference (ILD). Both ITD and ILD can be easily reproduced in headphones, and therefore lateral localisation can be considered sufficiently manageable for virtually all applications. The omnipresence of stereo audio speaks for itself in this regard.

The more intricate spatial effect is localisation in the directions front vs. back and above vs. below. This is mediated primarily by the acoustic filtering of torso, head and pinna through reflection, absorption and shading. It is a monaural phenomenon, i.e. it works even with a single ear, and is particularly effective at frequencies above $3\,\mathrm{kHz}$. The filtering is mathematically described in the form of a "head-related transfer function" (HRTF), which is the totality of directionally dependent transfer functions, varying thus not only with frequency but also azimuth and elevation angle of the incoming sound.

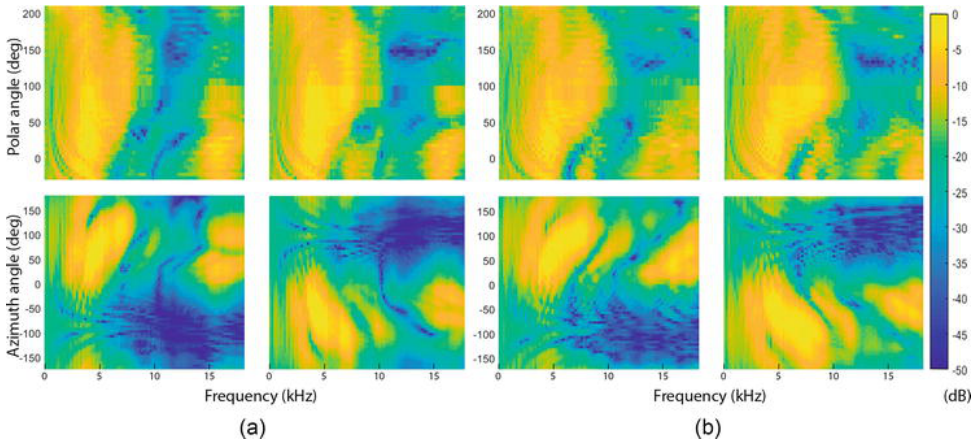


Figure 1: HRTF magnitude spectra for two different listeners (a) and (b), with left and right ear shown. Top: Spectra along the median plane. Bottom: Spectra along the eye-level horizontal plane. Source: [PKM22].

The literature on the role of individual HRTFs for externalisation, i.e. the effect of a sound being perceived as outside the head, shows somewhat inconsistent results [BBL+20]. Room acoustic and binaural features seem to be more important here [HGD16, LLPP16].

HRTFs depend on the individual anatomical geometry especially of the pinna, and are thus listener specific. Consequently, averaged HRTFs do not produce the desired spatial effects for most listeners, instead requiring individual HRTFs. The reason for the lack of

widespread use of HRTFs for spatial audio lies in the significant difficulties in their practical acquisition. Once this problem is solved, current 3D audio formats like classic 5.1 surround or Ambisonics, can be automatically converted to 3D headphone audio via the personalized HRTF. The general availability and usage of spatial audio will consequently increase dramatically. This work aims to make a contribution towards that end.

1.1 Acquisition of Head-related Transfer Functions

The acquisition methods for HRTFs fall in basically three categories [GS18]. The most straightforward method for HRTFs is acoustic measurement. In a very basic setup, a microphone is placed at the entrance of the closed left and right ear canal and a loudspeaker emits a measurement signal from different directions, to obtain an impulse response for discrete set of spatial angles. Systems with loudspeaker arcs or spherical arrays can significantly reduce the time needed for a full HRTF measurement. [LP20]

The second group of methods is numerical calculation of HRTFs based on geometric representations of torso, head and pinna. The effect of head and torso is large for the lower frequency components of the HRTF. These frequencies are however not sensitive to finer structures, so a rough model using only simple shapes of the correct width and depth has been shown to yield acceptable results. The angle-dependent distribution of peaks and notches of the HRTF especially in the higher frequency area is most heavily influenced by the pinna with its many fine reflective structures. Thus it is very important that these geometric details are acquired with high accuracy. [TMK+12] There are different methods of obtaining geometric 3D models of the pinna, each one with its own pros and cons. Among professional imaging techniques there is CT, MRI and Laser scans. All three of them need special, quite expensive equipment, but yield very accurate results in general. The CT would expose a human subject to high amounts of radiation, and is consequently usually done with silicone mouldings of the subject's ears, requiring an additional processing step. MRI scans face the problem that the ears of the subject are usually flattened by the equipment. Finally, with laser scans relying on line-of-sight propagation it is difficult to reach inwardly folded cavities of the pinna. With the improvement of algorithms in recent times, the technique of photogrammetry offers a promising alternative to the methods mentioned above. Thereby, a series of photographs from different angles and distances is taken and transformed by the photogrammetry algorithm to a 3D model. A typical photogrammetry algorithm works in this way: The first step is identifying mutual features of different images, i.e., characteristic image points which can be identified as representing the same point in real life. By the relative position of these points, the exact camera positions can be calculated. Consequently, there are pairs of images resp. camera positions chosen to obtain "depth maps" very much like the two human eyes can perceive depth. The depth maps are finally combined to the resulting 3D point cloud. Various postprocessing steps including noise reduction and smoothing are usually employed. Currently, photogrammetric geometric representations on their own do not meet the quality requirements for useful HRTF calculations. [DPT+08, PMF21]

Apart from these measurement and calculation approaches, one can also obtain an indi-

vidualized HRTF by personalisation of existing non-individual HRTFs. This adaptation can be based on certain geometric information about the listeners anatomy, or on purely subjective information, e.g. by the subject adjusting certain spectral parameters until the desired localisation effect is achieved [IAK22].

1.2 Photogrammetry and Non-rigid Registration Algorithms

Photogrammetric 3D models usually suffer from problems that render them virutally unusable for HRTF calculation. The generated mesh has often holes, which causes the HRTF calculation to throw an error. But also high frequency spatial noise results in very unrealistic reflections of especially high-pitched sounds. In practice one has to resort to elaborate manual correction and denoising to arrive at the geometric quality needed for an accurate HRTF result.

A very promising method in order to avoid this manual processing is a combination of the photogrammetric result with a so-called non-rigid[1] registration (NRR) algorithm [PMF23a]. The basic idea behind this is that we are not trying to geometrically capture a totally unknown object, but instead we know that being an ear it will be geometrically similar to other ears. NRR algorithms are specifically designed to morph one shape into a similar other shape. So it should be possible to fit a very good model of some template ear onto the noisy photogrammetric model of the actual ear. By this procedure, issues like holes or noise are resolved while hopefully maintaining enough geometric structure of the actual ear for the HRTF to be sufficiently individualized.

Attempts so far have shown, that this approach is indeed very promising, but not yet ideally working in practical situations. As is to be expected, the quality of the result depends heavily on the quality of the photogrammetric model. For very elaborate photogrammetric acquisitions the method yields excellent results. However, with a photogrammetry carried out in a customary at-home-situation, the outcomes are to be considered as not usable.

A comparison of different NRR algorithms, namely *coherent point drift* [MS10] *preserving global and local strctures* [MZY16], *Bayesian coherent point drift* (BCPD) [Hir21a] and *geodesic-based Bayesian coherent point drift* (GBCPD) [Hir22], has shown that the most promising candidates are BCPD and GBCPD [PMF23b].

Since fine tuning of the algorithm parameters cannot improve the results sufficiently, there are new ideas necessary. The approach here is to include additional information that is coming from the photogrammetric reconstruction algorithm. Besides the coordinates of the points in the point cloud, it also outputs a confidence level for each point, which represents the number of depth maps used in construction of this point.

In practice, points that lie further inward, or even behind some folding, are captured by fewer images and thus exhibit a low confidence level and also actually poorer geometric accuracy. By including the confidence level as input for the NRR algorithm, it should

[1] Non-rigid means that the transformation function is not only scaling, rotation and translation, but includes also any kind of non-linear deformation.

be possible to fit outwardly lying structures very accurately while leaving the inner parts more flexible. At these parts, the structure of the template ear will hopefully guess the covert shapes within the pinna folds approximately correct.

1.3 Goal of this Thesis

For our inquiry we chose the algorithm BCPD as one of the prime candidates. In order to avoid unnecessary complexity we favored BCPD over GBCPD, the latter being in fact an extension of BCPD that considers not only euclidic distance but also geodesic distance for motion coherence. However, as will be shown later, the modification of BCPD that is derived and implemented here is 100% compatible with GBCPD, so for future work all results will be usable for both BCPD and GBCPD.

The main goals of this thesis are:

- to show how target shape confidence levels can be mathematically included in the BCPD algorithm,
- to demonstrate a working implementation of this modified BCPD algorithm and,
- to evaluate, if appropriate confidence level input can alleviate some typical problems with fitting template pinna shapes onto photogrammetric pinna shapes by BCPD.

2 Methods

This section is comprised of a basic mathematical formulation of the BCPD algorithm, then the discussion of typically inclined problems with photogrammetric ear reconstructions and their NRR fitting, and finally a description of the synthetic test cases that are carried out and how their results are evaluated.

2.1 Bayesian Coherent Point Drift (BCPD)

Bayesian coherent point drift [Hir21a] is a reformulation and modification of the algorithm "coherent point drift" [MS10], which improves on it in terms of sensitivity to target rotation, kernel choice flexibility, computation speed and mathematical rigor. In this thesis we follow very closely the paper [Hir21a], often restating formulas or propositions, etc., with little or no modification. To avoid redundancies, we do not explicitly cite at each position.

The basic goal of a non-rigid registration algorithm is two-fold: To obtain a correspondence or registration from the source point to the target points, and to find a smooth map that transforms the source shape to the target shape.

The source shape is described by a point set $Y = \{y_1, \ldots, y_M\} \in \mathbb{R}^{3\times M}$ and the target shape by a point set $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^{3\times M}$. The transformation map of the source point set is given by $\mathcal{T}(y_m) = sR(y_m + v_m) + t$. The rigid part of the transformation, i.e. scaling by $s > 0$, rotation by a rotation matrix $R \in \mathbb{R}^{3\times 3}$ and translation by a vector $t \in \mathbb{R}^3$, is parameterized by $\rho = (s, R, t)$. The non-rigid deformation is contained in the vectors v_m, that obey a type of motion coherence.

After convergence we arrive at the desired transformation map $\mathcal{T}$, the adjusted source shape $\hat{Y} = \{\mathcal{T}(y)|y \in Y\}$ and a correspondence matrix $P = (p_{mn})$, defined later, where p_{mn} is the probability that x_n corresponds to y_m.

We will use the following mathematical notation:

- I_K is the identity matrix of size K.
- 1_K is the vector of size K, with all elements being 1.
- $\mathrm{Tr}(\cdot)$, $|\cdot|$, $\mathrm{d}(\cdot)$, $\mathrm{svd}(\cdot)$ are the trace of a matrix, the determinant of a matrix, the operation that converts a vector into its corresponding diagonal matrix, and the singular value decomposition, respectively.
- The symbol $\otimes$ represents the Kronecker product.
- $\phi(x; \mu, \sigma)$ is the density function of a Gaussian random variable with mean μ and covariance matrix σ.

2.1.1 Probabilistic Model

The underlying probabilistic model assumes that the target points x_n are *observations of a Gaussian mixture model* where the centroids are given by the transformed source points $\mathcal{T}(y_m)$. Specifically the generation of target points proceeds in this way:

1. A target point x_n is selected as either a point forming a target shape or an outlier with the outlier probability ω.
2. If a target point x_n is selected as an outlier, x_n is generated from the outlier distribution $p_{out}(x_n)$, which is uniform on some appropriate bounding box.
3. If x_n is a non-outlier, an index $m \in \{1, ..., M\}$, indicating that x_n corresponds to y_m , is sampled with a probability α_m, where $\sum_{m=1}^{M} \alpha_m = 1$.
4. Then, x_n is generated from a 3-dimensional multivariate normal distribution with a mean vector $\mathcal{T}(y_m)$ and a covariance matrix $\sigma^2 I_3$.
5. A target point set X is generated by an N-times repetition of steps 1, 2, 3, and 4.

To properly formalize this, we need indicator variables $c_n \in \{0, 1\}$, where $c_n = 0$ exactly if x_n is an outlier, as well as index variables $e_n \in \{0, \ldots, M\}$, where $e_n = m$ exactly if

x_n was generated by y_m. Using the shorthand $\phi_{mn} = \phi(x_n; \mathcal{T}(y_m), \sigma^2 I_3)$, we can now write down the probability distribution of (x_n, c_n, e_n) given $y, v, \alpha, \rho, \sigma^2$:

$$p(x_n, e_n, c_n | y, v, \alpha, \rho, \sigma^2) = \left\{ (1-\omega) \prod_{m=1}^{M} (\alpha_m \phi_{mn})^{\delta_m(e_n)} \right\}^{c_n} \{\omega p_{out}(x_n)\}^{1-c_n} \quad (1)$$

Following the Bayesian idea of defining priors, the full probability of observing the point cloud X is given by:

$$p(x, y, \theta) = p(\alpha) p(v|y) \prod_{n=1}^{N} p(x_n, e_n, c_n | y, v, \alpha, \rho, \sigma^2) \quad (2)$$

with $\theta = (v, \alpha, c, e, \rho, \sigma^2)$.

The prior of the weights α is a Dirichlet distribution Dir with parameter κ, which is a standard choice in Bayesian models of this kind:

$$p(\alpha) = \mathrm{Dir}(\alpha | \kappa 1_M). \quad (3)$$

The prior of the displacement vectors v is given by a Gaussian distribution:

$$p(v|y) = \phi(v; 0, \lambda^{-1} G \otimes I_3). \quad (4)$$

The matrix $G = (g_{mm'}) \in \mathbb{R}^{M \times M}$ is defined by $g_{mm'} = \mathcal{K}(y_m, y_{m'})$, with $\mathcal{K}$ being a positive definite kernel.

In our case the kernel is Gaussian, based on euclidic distance, with spatial correlation length parameter $\beta > 0$, i.e.:

$$\mathcal{K}(y_m, y_{m'}) = \exp\left(-\frac{1}{2\beta^2} ||y_m - y_{m'}||^2 \right). \quad (5)$$

This means that the displacement vectors v are zero-mean normally distributed random variables, where the correlation between two vectors v_m and $v_{m'}$ increases with euclidic proximity of the points y_m and $y_{m'}$. This ensures the desired smoothness of the non-rigid deformation. The expected length of the deformation vectors can be adjusted by the parameter λ.

The kernel $\mathcal{K}$ can also be chosen as not only based on euclidic distance, but also on the geodesic distance of two source points, i.e. the length of the shortest path on the source surface. This gives rise to the algorithm “geodesic based Bayesian coherent point drift” (GBCPD) [Hir22], which is also a promising candidate for adjusting ear geometries.

2.1.2 Solving the Model

The core concept of the algorithm is a bayesian inference ansatz based on the probability density $p(x, y, \theta)$ of our model defined by equation (2). This means that we

treat the parameters θ as if they were observables and calculate $p(\theta|x,y)$ from $p(x,y,\theta)$. Then, the optimal parameters θ are the ones that maximize $p(\theta|x,y)$. The solution $\tilde{\theta} = (v, \alpha, c, e, \rho, \sigma^2)$ contains all desired information, including the function $\mathcal{T}$.

Since a direct calculation of $p(\theta|x,y)$ and its maximum mode is mathematically intractable, the algorithm applies the principle of "Variational Bayesian Inference" (VBI), which functions by approximating $p(\theta|x,y)$ by a product of simpler factors. Without going too much into mathematical details, let us state here the important steps applied to our case:

1. Approximate $p(\theta|x,y) \approx q_1(\theta_1)q_2(\theta_2)q_3(\theta_3) = q_1(v,\alpha)q_2(c,e)q_3(\rho,\sigma^2)$.
2. Initialize q_1, q_2 and q_3.
3. Cycle the update of q_i with the other q_j fixed for each $i \in \{1,2,3\}$ using the equation
$$\ln \hat{q}_i = E_i \left[\ln p(x,y,\theta)\right] + \text{const.}, \tag{6}$$
where E_i denotes the expectation over $\prod_{j\in\{1,2,3\}\setminus\{i\}} q_j(\theta_j)d\theta_j$.
4. Repeat step 3. until convergence.
5. The maximizing parameter set is given by $\tilde{\theta} = (\tilde{\theta}_1, \tilde{\theta}_2, \tilde{\theta}_3)$, where $\tilde{\theta}_i$ is the maximum mode of the converged q_i.

The VBI update rule (6) is explicitly solvable for the model. The solution formulas are subject of propositions 1, 2 and 3 [Hir21a], and are restated with modifications in section 3.1. The BCPD algorithm is a direct result from these propositions.

The BCPD algorithm in its naive form is of computational complexity $\mathcal{O}(MN + M^3)$ and therefore in [Hir21a] a combination of two methods, the Nyström method and a KD tree search, is used to speed up the calculations to $\mathcal{O}\left((M+N)\log(M+N)\right)$. The details are omitted here.

2.2 Adjustment of Pinna Geometries

As indicated before, the quality of photogrammetry results varies from very good to hardly useful. Figure 2 shows the ground truth for pinna NH5 from the database of ARI, Vienna, along with photogrammetric capturings of the same ear. The high quality result has been acquired with elaborate methods, including shaving of the fine pinna hair, application of a matte spray, professional camera and lighting. The low quality one has been made under typical "at home" conditions, without using any equipment except a smartphone camera.

Empirically, there have been found different typical problems of photogrammetric reconstructions of pinnae and their improvement by NRR: Firstly, there is always some noise present. Since the noise is usually not uniformly strong, but more pronounced in the further inside and folded parts of the pinna, the modified NRR algorithm might help improve

(a) Ground truth. (b) High quality photogrammetry. (c) Low quality photogrammetry.

Figure 2: The ground truth and two different photogrammetry results for NH5.

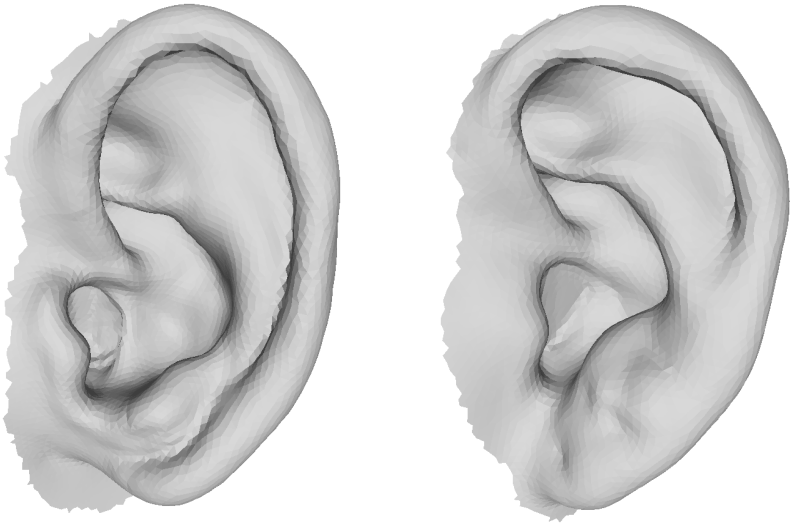
the fitted result, since these parts also exhibit lower confidence level. A second problem is the closing up of cavities: Further inside structures are often not captured by the images, and photogrammetric algorithms commonly close up these hollow spaces. Both of these problems will be adressed in our synthetic testing of the modified BCPD algorithm. Thirdly, if the quality of the photogrammetry is very low, the NRR algorithm might mismatch some parts entirely, resulting in quite unusable results. This is difficult to test in artificially constructed cases, but might be alleviated indirectly by improving the other two types of problems.

2.3 Evaluation of the modified algorithm

The modified algorithm is evaluated using synthetic test cases. As discussed before in improving photogrammetries by NRR, one maps a high quality ground truth ear, called source, onto the low quality photogrammetrically reconstructed ear, called target. The source, i.e. the shape that is deformed by the NRR algorithm, will always be the NH131 pinna (see Figure 3). The target, i.e. the shape that is intented to be matched by the deformed source, will be the NH5 ear, albeit distorted in some way as to artificially create the typical problems discussed in the section above.

For the localised noise, a medium sized area at a more inside part of the NH5 target was selected. Each point of the area was randomly displace following a rotational symmetric Gaussian distribution with mean zero. The experiments were run using six different noise levels in $1\,\mathrm{mm}$ steps, i.e. the standard deviation ranged from $0\,\mathrm{mm}$ to $5\,\mathrm{mm}$. A visual representation of the noisy target with the distance to the undistorted target is shown in Figure 7.

The closing up of cavities is modelled by a manual systematic deformation to the hollow structures of the upper part of the pinna, called helix rim and cymba conchae. Again, we



(a) Pinna NH5, used as target. (b) Pinna NH131, used as source.

Figure 3: The ground truths used as the basis for the synthetic test cases.

choose six grades of deformation, with deformation grade 0 reflecting no deformation. The changes to the target shape are shown in Figure 4.

For each noise level and deformation grade, we at first carried out the original BCPD algorithm and consequently the modified BCPD algorithm.

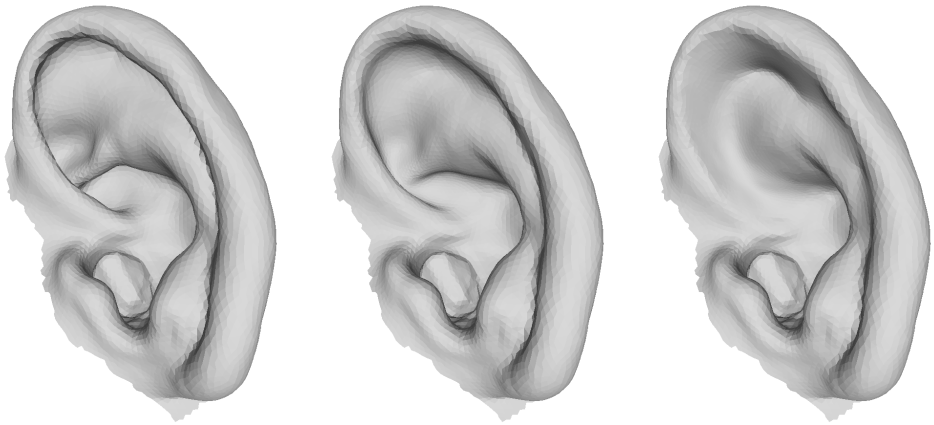
To evaluate the quality of the fit, we choose a statistical point-to-surface distance metric: The reference point set is given by the *undistorted positions* of the target shape points, that were subject to the specific artificial distortion. Then, for each point of the reference point set, its distance to the surface of the adjusted source shape outputted by the algorithm was calculated. The distribution of these distances is finally displayed in the form of a violin plot.

3 Results

3.1 Modification of the BCPD algorithm

As discussed earlier, most of the photogrammetric algorithms output a relative confidence level for each input point x_n. For ease of notation we assume that the input data associated with each x_n is a relative *standard deviation scaling factor*, or *uncertainty level* $\zeta_n > 0$, i.e. inversely related with the confidence level.

From the probabilistic model definition described in Section 2.1.1 there appears a straightforward way of meaningfully incorporating the ζ_n factors. As x_n is assumed to be generated from the points y_m in a Gaussian mixture kind of way, it is sensible to allow points x_n with low confidence, i.e. high ζ_n to deviate further from the Gaussian centroids and



(a) Deformation grade 0, i.e. undeformed Pinna. (b) Deformation grade 3. (c) Deformation grade 5.

Figure 4: Deformation of the target pinna NH5 used for synthetic tests.

vice versa. Thus we would modify a single assumption only, which is point 4.:

4. Then, x_n is generated from a 3-dimensional multivariate normal distribution with a mean vector $\mathcal{T}(y_m)$ and a covariance matrix $\zeta_n^2\sigma^2 I_3$.

The changes in the algorithm are then consequences of this modified assumption only. For improved readability we will continue to highlight the changes in the formulas in color.

Another posibility would have been to adjust the *outlier probability* ω proportionally to ζ_n, but as we do prefer the locational information to be relaxed instead of eliminated, the above approach seemed more sensible.

3.1.1 Derivation of the Algorithm

By the above modification we now have the shorthand $\phi_{mn} = \phi(x_n; \mathcal{T}(y_m), \zeta_n^2\sigma^2 I_3)$.

The derivation of the original BCPD algorithm is contained in the propositions 1, 2 and 3 [Hir21a], being proven in the supplementary document [Hir21b]. To adapt them to the modified model, we will need some new definitions:

$$p^*_{mn} = \zeta_n^{-2} p_{mn}, \qquad \nu^*_m = \sum_n p^*_{mn}, \qquad \nu'^*_n = \sum_m p^*_{mn} \qquad \hat{N}^* = \sum_{m,n} p^*_{mn}. \tag{7}$$

The proofs of the propositions is a moderately complicated adaption of the proofs contained in [Hir21b]. The starting point is always the VBI update equation(6), now of course with the modified probability $p(x, y, \theta)$ given by (2). The calculations follow through without significant problems, by replacing a variable by its starred variant according to (7) if necessary. Detailed proofs are therefore not provided.

Proposition 3.1. *The approximated posterior distribution $\hat{q}_1(v, \alpha)$ is factorized into its marginals, i.e., $\hat{q}_1(v, \alpha) = \hat{q}_\alpha(\alpha)\hat{q}_v(v)$. Furthermore, $\hat{q}_\alpha(\alpha)$ is a Dirichlet distribution and $\hat{q}_v(v)$ is a multivariate normal distribution, which are defined as follows:*

$$\hat{q}_\alpha(\alpha) = \mathrm{Dir}(\alpha|\kappa 1_M + \nu),$$
$$\hat{q}_v(v) = \phi\Big(v; \hat{v}, \tilde{\Sigma}\Big),$$

where $\hat{x} = \mathrm{d}(\tilde{\nu}^*)^{-1}\tilde{P}^* x \in \mathbb{R}^{3\times M}$ is the shape for which $\tilde{T}^{-1}(\hat{x})$ is interpreted as the inverse alignment from x to y, and $\tilde{\Sigma} = \Sigma \otimes I_3 \in \mathbb{R}^{3M\times 3M}$ with $\Sigma = (\lambda G^{-1} + \frac{s^2}{\sigma^2}\mathrm{d}(\nu^*))^{-1} \in \mathbb{R}^{M\times M}$ is the posterior covariance matrix of displacement vector v.

For later reference, we will need the following variable definitions:

$$\hat{x} = (\hat{x}_1^T, \ldots, \hat{x}_M^T)^T = \mathrm{d}(\tilde{\nu}^*)^{-1}\tilde{P}^* x,$$
$$\hat{v} = (\hat{v}_1^T, \ldots, \hat{v}_M^T)^T = \frac{s^2}{\sigma^2}\tilde{\Sigma}\mathrm{d}(\tilde{\nu}^*)(\tilde{T}^{-1}(\hat{x}) - y),$$
$$\hat{u} = (\hat{u}_1^T, \ldots, \hat{u}_M^T)^T = y + \hat{v},$$
$$\hat{y} = (\hat{y}_1^T, \ldots, \hat{y}_M^T)^T = \tilde{T}(y + \hat{v}).$$

The proposition allows us to compute:

$$\langle \alpha_m \rangle = \exp\left\{\psi(\kappa + \nu_m) - \psi(\kappa M + \hat{N})\right\},$$
$$\langle \phi_{mn} \rangle = \phi(x_n; \hat{y}_m, \zeta_n^2\sigma^2 I_3) \exp\left\{ -\frac{s^2}{2\zeta_n^2\sigma^2}\mathrm{Tr}(\sigma_m^2 I_3)\right\}.$$

The next proposition is completely sustained, containing only the implicit change of ϕ_{mn}:

Proposition 3.2. *The approximated posterior distribution $\hat{q}_2(c, e)$ is a combination of a Bernoulli distribution and a categorical distribution, and is defined as follows:*

$$\hat{q}_2(c, e) = \prod_{n=1}^{N}(1 - \nu'_n)^{1-c_n}\left\{\nu'_n \prod_{m=1}^{M}\left(\frac{p_{mn}}{\nu'_n}\right)^{\delta_m(e_n)}\right\}^{c_n},$$

where p_{mn} and ν'_n are defined as

$$p_{mn} = \frac{(1-\omega)\langle \alpha_m \rangle \langle \phi_{mn} \rangle}{\omega p_{\mathrm{out}}(x_n) + (1-\omega)\sum_{m'=1}^{M}\langle \alpha_{m'} \rangle \langle \phi_{m'n} \rangle},$$

and $\nu'_n = \sum_{m=1}^{M} p_{mn}$.

To state the final proposition, the following definitions are needed:

$$
\begin{aligned}
\bar{x} &= \frac{1}{\hat{N}^*} \sum_{m=1}^{M} \nu_m^* \hat{x}_m, \\
\bar{\sigma}^2 &= \frac{1}{\hat{N}^*} \sum_{m=1}^{M} \nu_m^* \sigma_m^2 \\
\bar{u} &= \frac{1}{\hat{N}^*} \sum_{m=1}^{M} \nu_m^* \hat{u}_m, \\
S_{xu} &= \frac{1}{\hat{N}^*} \sum_{m=1}^{M} \nu_m^* (\hat{x}_m - \bar{x})(\hat{u}_m - \bar{u})^T, \\
S_{uu} &= \frac{1}{\hat{N}^*} \sum_{m=1}^{M} \nu_m^* (\hat{u}_m - \bar{u})(\hat{u}_m - \bar{u})^T + \bar{\sigma}^2 I_3.
\end{aligned}
$$

Proposition 3.3. *Suppose the approximated posterior distribution $q_3(\rho, \sigma^2)$ is a Dirac delta function. Given $q_1(v, \alpha)$ and $q_2(c, e)$, the lower bound is maximized by the following equations:*

$$
\begin{aligned}
\hat{R} &= \Phi \mathrm{d}(1, \ldots, 1, |\Phi \Psi^T|) \Psi^T, \\
\hat{s} &= \mathrm{Tr}\big(\hat{R}^T S_{xu}\big) / \mathrm{Tr}\big(S_{uu}\big), \\
\hat{t} &= \bar{x} - \hat{s} \hat{R} \bar{u}, \\
\hat{\sigma}^2 &= \frac{1}{3\hat{N}} \big\{ x^T \mathrm{d}(\tilde{\nu}^{*\prime}) x - 2 x^T \tilde{P}^{*T} \hat{y} + \hat{y}^T \mathrm{d}(\tilde{\nu}^*) \hat{y} \big\} + \frac{\hat{N}^*}{\hat{N}} \hat{s}^2 \bar{\sigma}^2,
\end{aligned}
$$

where Φ and Ψ are the orthogonal matrices of size 3×3 obtained by the singular value decomposition of S_{xu}, i.e., $S_{xu} = \Phi S'_{xu} \Psi^T$ with the diagonal matrix of singular values, $S'_{xu} \in \mathbb{R}^{3\times 3}$.

All of the actual computation steps, which are the direct consequence of these propositions, are summarised in Figure 5.

It is immediately clear from Equation (7) that in the case $\zeta_n = 1$ all starred variables equal their unstarred variants, and consequently all formulas revert to their original version.

To prove our earlier claim, that the modifications of BCPD are fully compatible with GBCPD, notice the difference for GBCPD being only an exchange of the matrix G. As is clear from Figure 5, G is not modified in any way, and also all computations with Σ can be carried out unaltered, simply with the input ν^* instead of ν.

3.1.2 Numerical Optimizations

The numerical optimisations are targeted at the bottleneck computations which are the matrix operations with G, Σ and P. Since G and Σ are not modified, their optimizations can be sustained exactly as is.

Algorithm: Bayesian Coherent Point Drift with target error levels

- Input: $x \in \mathbb{R}^{3\times N},\ \zeta \in \mathbb{R}^{N},\ y \in \mathbb{R}^{3\times M},\ \omega,\ \lambda,\ \kappa,\ \gamma.$
- Output: $\hat{y} = \tilde{T}(y+\hat{v}) = s(I_M \otimes R)(y+\hat{v}) + (1_M \otimes t).$
- Initialisation:

  $\hat{y} = y, \hat{v} = 0, \Sigma = I_M, s = 1, R = I_3, t = 0,$
  $\langle \alpha_m \rangle = \frac{1}{M}, \sigma^2 = \frac{\gamma}{3NM}\sum_{n=1}^{N}\sum_{m=1}^{M}||x_n - y_m||^2,$
  $G = (g_{mm'})$ with $g_{mm'} = \mathcal{K}(y_m, y_{m'}).$
- Optimisation: Repeat until convergence.
  - Update $P = (p_{mn})$ and related terms.

    $\langle \phi_{mn} \rangle = \phi(x_n; \hat{y}_m, \zeta_n^2\sigma^2 I_3)\exp\left\{-\frac{s^2}{2\zeta_n^2\sigma^2}\mathrm{Tr}(\sigma_m^2 I_3)\right\}$
    $p_{mn} = \frac{(1-\omega)\langle\alpha_m\rangle\langle\phi_{mn}\rangle}{\omega p_{\mathrm{out}}(x_n)+(1-\omega)\sum_{m'=1}\langle\alpha_{m'}\rangle\langle\phi_{m'n}\rangle},$
    $\nu = P1_N, \hat{N} = \nu^T 1_M$
    $\nu^* = P^* 1_N, \nu'^* = P^{*T}1_M, \hat{N}^* = \nu^{*T}1_M$
    $\hat{x} = \mathrm{d}(\tilde{\nu}^*)^{-1}\tilde{P}^* x.$
  - Update Σ, $\hat{v}$, $\hat{u}$, and $\langle\alpha_m\rangle$ for all m.

    $\Sigma^{-1} = \lambda G^{-1} + \frac{s^2}{\sigma^2}\mathrm{d}(\nu^*), \hat{v} = \frac{s^2}{\sigma^2}\tilde{\Sigma}\mathrm{d}(\tilde{\nu}^*)(\tilde{T}^{-1}(\hat{x}) - y),$
    $\hat{u} = y + \hat{v}, \langle\alpha_m\rangle = \exp\{\psi(\kappa+\nu_m) - \psi(\kappa M + \hat{N})\}.$
  - Update s, R, t, σ^2, $\hat{y}$ and related terms.

    $\bar{x} = \frac{1}{\hat{N}^*}\sum_{m=1}^{M}\nu_m^*\hat{x}_m, \bar{\sigma}^2 = \frac{1}{\hat{N}^*}\sum_{m=1}^{M}\nu_m^*\sigma_m^2,$
    $\bar{u} = \frac{1}{\hat{N}^*}\sum_{m=1}^{M}\nu_m^*\hat{u}_m,$
    $S_{xu} = \frac{1}{\hat{N}^*}\sum_{m=1}^{M}\nu_m^*(\hat{x}_m - \bar{x})(\hat{u}_m - \bar{u})^T,$
    $S_{uu} = \frac{1}{\hat{N}^*}\sum_{m=1}^{M}\nu_m^*(\hat{u}_m - \bar{u})(\hat{u}_m - \bar{u})^T + \bar{\sigma}^2 I_3,$
    $\Phi S'_{xu}\Psi^T = \mathrm{svd}(S_{xu}), R = \Phi\mathrm{d}(1,\cdots,1,|\Phi\Psi|)\Psi^T,$
    $s = \mathrm{Tr}(R^T S_{xu})/\mathrm{Tr}(S_{uu}), t = \bar{x} - sR\bar{u}, \hat{y} = \tilde{T}(y+\hat{v}),$
    $\hat{\sigma}^2 = \frac{1}{3\hat{N}}\left\{x^T\mathrm{d}(\tilde{\nu}^{*\prime})x - 2x^T\tilde{P}^{*T}\hat{y} + \hat{y}^T\mathrm{d}(\tilde{\nu}^*)\hat{y}\right\} + \frac{\hat{N}^*}{\hat{N}}\hat{s}^2\bar{\sigma}^2$

Figure 5: Modified BCPD algorithm.

The optimization for P is more involved, as P contains the implicit change through ϕ_{mn}, but also we now have calculations involving P^*. In the early stages of the algorithm, when P is densely populated but close to a low rank matrix, the Nyström method is used to approximate P. Since it relies on P being a Gaussian affinity matrix, the method breaks down in the modified case. The KD tree method on the other hand relies on the fact that later in the algorithm, P is sparse, i.e. most of its entries are very close to zero. The non-zero elements are the ones where x_n and y_m are close to one another which can be efficiently searched for by a KD tree. While we have to abandon the Nyström method, the KD tree method can be sustained by slightly increasing the search ball radius based on the maximum of ζ.

The resulting computation times are very acceptable, in the range of 1 to 2 minutes for the target and source point cloud sizes of around $10k$ points on a conventional home-use laptop. Alternatives to the Nyström method could certainly be found, but are not necessary for our purposes.

3.1.3 Implementation

At the public github repository of O. Hirose[2], a complete `C++` implementation of BCPD is available, which we used as the basis for implementing the modified BCPD algorithm. The source code is structured as follows:

- The file `register/main.c` is the main container function, where the basic setup and all preliminaries are handled, like loading of the input files, memory allocation, checks, normalisation, downsampling and saving of the results.
- `register/main.c` calls the file `register/bcpd.c`, where the core of the algorithm is executed. It includes all of the calculations from Figure 5, sometimes using internal subroutines, which are located in the folder `base/`, and external subroutines from the *LAPACK* package for linear algebra operations.
- `base/gaussprod.c` is responsible for calculating all products involving the matrix P, which is more intricate due to the Nyström and KD tree optimisations.

To avoid excessive restructuring of the code, we make one approximation to the modified BCPD algorithm described in Figure 5, which is replacing ζ_n^2 by $\overline{\zeta^2}$ in the equation for $\langle\phi_{mn}\rangle$:

$$\langle\phi_{mn}\rangle \approx \phi(x_n; \hat{y}_m, \zeta_n^2\sigma^2 I_3) \exp\Big\{ -\frac{s^2}{2\overline{\zeta^2}\sigma^2} \mathrm{Tr}(\sigma_m^2 I_3)\Big\} \tag{8}$$

In this way, the multiplication by the exponential can be kept outside of the function `gaussprod`. For details of the calculations involving P we refer to Section 4.6.2 [Hir21a].

A `git diff` excerpt from the modified version of `register/bcpd.c` compared to the original one is included in Section A. The changes are the following:

[2] https://github.com/ohirose/bcpd

- Mostly, variables are replaced by their starred versions where required, according to Figure 5, notably `w` / ν by `wSt` / ν^*, `PX` / $\tilde{P}x$ by `PStX` / $\tilde{P}^*x$ and `*Np` / $\hat{N}$ by `NpSt` / $\hat{N}^*$. `f` now represents ν'^*, since the unstarred ν' does not appear anymore.
- In the beginning the increase for the KD tree search radius is calculated using `ZetaMax` and `ZetaMean`. It is passed onto the `gaussprod` functions as a multiplier to the variable `dlt`.
- `c` was upgraded to a vector, since it is now ζ_n dependent. For details regarding `c`, `b` and `q`, see c'_0, b and q in Section 4.6.2 [Hir21a].
- The function `gaussprod_batch` has been modified to output ν, ν^* and $\tilde{P}^*x$.

3.2 Evaluation of the modified algorithm

In performing the modified BCPD algorithm, the ζ_n values used were $\zeta_n^2 = 8$ for all x_n that were deformed in any way, and $\zeta_n^2 = 1$ for the undeformed points x_n. All calculations were done using the following parameter set:

Name	Value	Description
β	0.5	Smoothness of the non-rigid deformation map
λ	300	Magnitude of the non-rigid deformation
ω	0.1	Outlier probability
γ	2	How much the initial alignment is considered
K	150	Number of Nyström samples used in approximating G
n	200	Maximum number of loops
c	10^{-7}	Convergence tolerance. If the difference of two consecutive σ is smaller than c, the algorithm stops.

In the following figures, all distances are indicated in the unit mm.

3.2.1 Target with Noise

The resulting violin plots for how close the target points in the region of interest are to the algorithm output are shown in Figure 6. For noise level 0, the original algorithm is clearly superior, as is to be expected. For noise level 1 and 4, the original algorithm also outperforms the modified one. Noise level 2 is solved better by the modified algorithm, the other two cases can be considered a tie. All in all the modified algorithm does not meet the expectations of improving the fitted result. A sensible explanation of this behaviour would be, that the symmetrically distorted target points still carry the correct information on average, thus using the information coming from these points fully has still an advantage over attenuating the influence of these points. The original BCPD algorithm seems to have enough robustness to handle this localized noise even without the confidence level adjustment. A visualisation of the results for noise level 3 are shown in Figure 7.

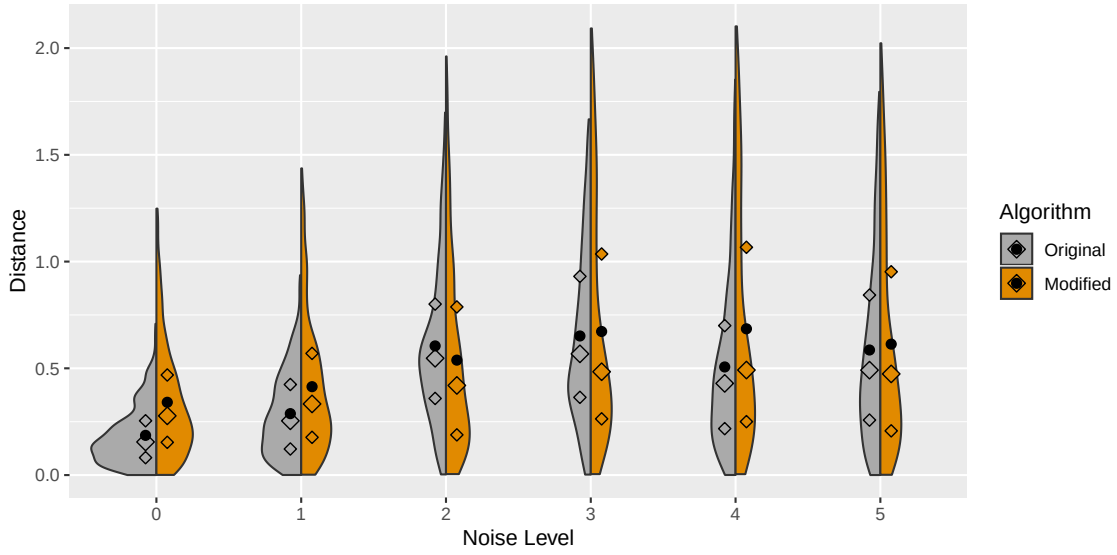


Figure 6: Violin plot of the distance distribution of the undeformed target points in the region of interest to the adjusted source point surface for different noise levels. Mean (dot) and quartiles (diamonds) indicated.

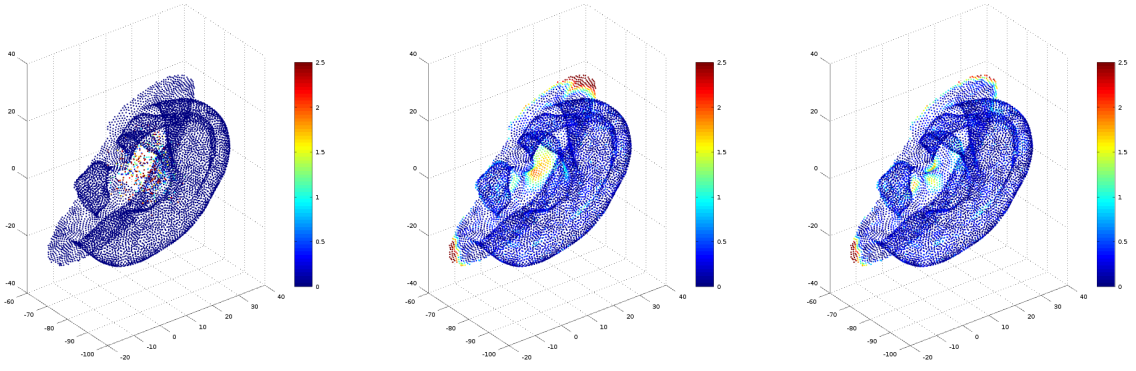


Figure 7: Left: The distorted target shape at noise level 3, colorcoded with distances to the original target shape. Middle: The original target shape, colorcoded with the distance to the adjusted source using the *original* BCPD. Right: The original target shape, colorcoded with the distance to the adjusted source using the *modified* BCPD.

3.2.2 Target with Systematic Deformation

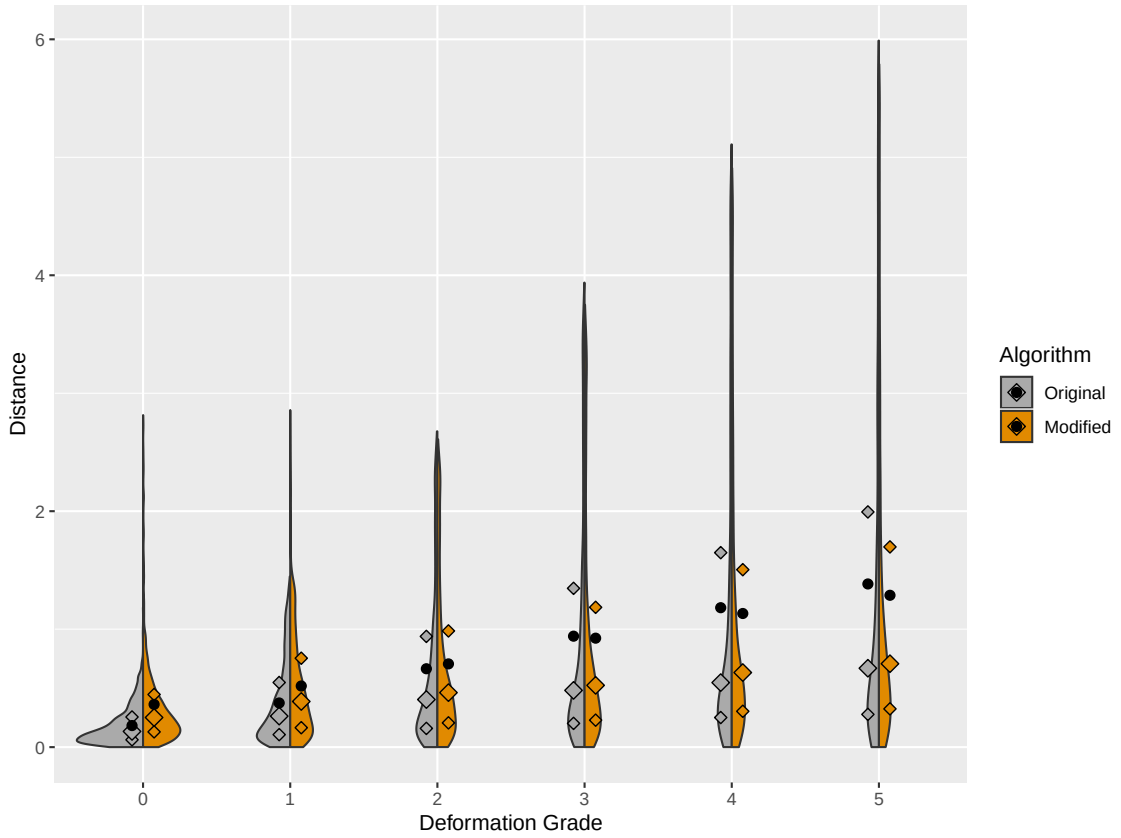


Figure 8: Violin plot of the distance distribution of the undeformed target points in the region of interest to the adjusted source point surface for different deformation grades. Mean (dot) and quartiles (diamonds) indicated.

For the target shape being systematically deformed, the resulting violin plots for how close the target points in the region of interest are to the algorithm output are shown in Figure 3.2.2. For deformations up to grade 2, the modified algorithm is clearly outperformed by the original algorithm. In the case of larger deformations, the modified algorithm improves on the upper quartile and the mean, but degrades on the lower quartile and median. If applying a worst-case metric, i.e. above all avoiding large deviations, this could be considered a tiny advantage of the modified algorithm, otherwise it has to be considered a tie. One weakness of this synthetic test, is the choice of ζ being uniform on all deformed points. In practice the confidence levels would smoothly decline from the outside to the inside of the pinna. Adapting the ζ values in this way could potentially improve upon the performance of the modified BCPD a bit. A visualisation of the results for deformation grade 3 are shown in Figure 9.

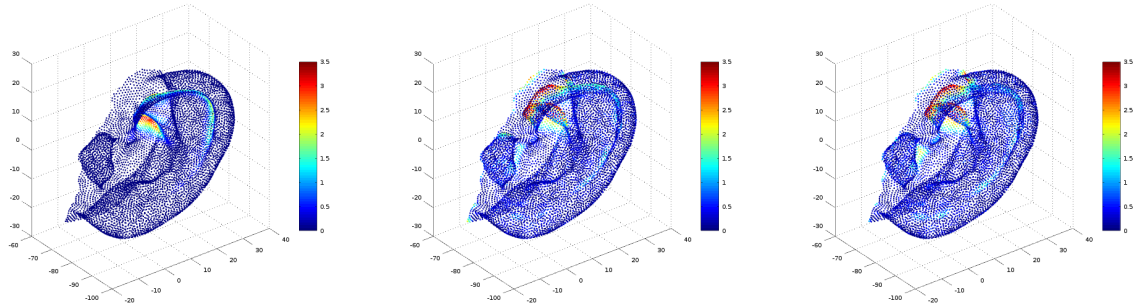


Figure 9: Left: The target shape with deformation grade 3, colorcoded with distances to the original target shape. Middle: The original target shape, colorcoded with the distance to the adjusted source using the *original* BCPD. Right: The original target shape, colorcoded with the distance to the adjusted source using the *modified* BCPD.

4 Conclusions

The results of the synthetic tests reveal the performance of the modified BCPD algorithm to be rather mixed. In both tests we carried out, the modified algorithm outperformed its original variant in half of the cases at best. Nonetheless, the modified algorithm is an improvement if considered from the view of having additional input choices to affect the outcome in a desired way. Other practical applications are probable and could be analysed in future research. One specific idea would be to create a two-step process, where in the first step the modified algorithm is used to ensure a correct rough fitting, i.e. a fitting where at least the parts with high confidence are matched correctly - avoiding the third type of problem explained in Section 2.2. In the second step, the original algorithm would adjust the fine structures, greatly supported by the parts further out being already matched.

A Source Code Excerpt

This section contains a `git diff` excerpt from `register/bcpd.c`, where the main parts of the modified BCPD calculations take place. As usual, newly added lines are preceeded by "+", and removed lines by "-".

```
@@ -131,6 +135,10 @@ int bcpd(
   /*-----------------------------------------------------------------o
   |   initialization                                                 |
   o-----------------------------------------------------------------*/
+  /* Zeta KD Tree search radius modifier */
+  for(n=0;n<N;n++) ZetaMax = Zeta[n]>ZetaMax?Zeta[n]:ZetaMax;
+  ZetaKDrad = sqrt(sqrt(ZetaMax*ZetaMean));
+  printf("  zeta^2 has a maximum of %f. Setting KD tree search radius factor to %f \n\n",ZetaMax,ZetaKDrad);
   /* a */
   for(m=0;m<M;m++) a[m]=1/(double)M;
   for(m=0;m<M;m++) sgm[m]=0;
@@ -171,18 +179,20 @@ int bcpd(
     |   update: x, w                                                 |
     o-----------------------------------------------------------------*/
     local=(*r<pm.btn); flg=(local&&(opt&PW_OPT_LOCAL))?GRAM_FLAG_LOCAL:0;
-    c=(pow(2.0*M_PI*SQ(*r),0.5*D)*omg)/(vol*(1-omg)); /* c */
-    for(m=0;m<M;m++) b[m]=a[m]*exp(-(D/2)*sgm[m]*SQ((*s)/(*r)));
```

```
+    for(n=0;n<N;n++) c[n]=(pow(2.0*M_PI*SQ(*r)*Zeta[n],0.5*D)*omg)/(vol*(1-omg)); /* c */
+    for(m=0;m<M;m++) b[m]=a[m]*exp(-(D/2)*sgm[m]*SQ((*s)/(*r))/ZetaMean);

-    gaussprod (q,wgd,wgi,y,X,b,Ty,D,M,N,J,*r,dlt,lim,flg|GRAM_FLAG_TRANS|GRAM_FLAG_BUILD);/* Kt1 */
-    for(n=0;n<N;n++) q[n]=1.0/(q[n]+c); /* q */
-    for(n=0;n<N;n++) f[n]=1.0-(q[n]*c); /* f */
-    gaussprod_batch (w,PX,wgd,wgi,y,X,q,Tx,D,M,N,J,*r,dlt,lim,flg);
+    gaussprod (q,wgd,wgi,y,X,Zeta,b,Ty,D,M,N,J,*r,dlt*ZetaKDrad,lim,flg|GRAM_FLAG_TRANS|GRAM_FLAG_BUILD);/* Kt1 */
+    for(n=0;n<N;n++) q[n]=1.0/(q[n]+c[n]); /* q */
+    for(n=0;n<N;n++) f[n]=(1.0-(q[n]*c[n]))/Zeta[n]; /* f */ /* This is actually fSt */
+    gaussprod_batch (w, wSt, PStX,  wgd,wgi,y,X,Zeta,q,Tx,D,M,N,J,*r,dlt*ZetaKDrad,lim,flg|GRAM_FLAG_STARR);

     for(m=0;m<M;m++){w[m]*=b[m];w[m]=w[m]<reg?reg:w[m];}
-    for(d=0;d<D;d++)for(m=0;m<M;m++) PX[m+M*d]*=b[m];
-    *Np=0;for(m=0;m<M;m++) *Np+=w[m]; /* Np */
-    for(m=0;m<M;m++)for(d=0;d<D;d++) x[d+D*m]=PX[m+M*d]/w[m]; /* x */
+    for(m=0;m<M;m++){wSt[m]*=b[m];wSt[m]=wSt[m]<reg?reg:wSt[m];}
+    for(d=0;d<D;d++)for(m=0;m<M;m++) PStX[m+M*d]*=b[m];
+    *Np=0; for(m=0;m<M;m++) *Np +=w[m]; /* Np */
+    NpSt=0;for(m=0;m<M;m++) NpSt+=wSt[m];
+    for(m=0;m<M;m++)for(d=0;d<D;d++) x[d+D*m]=PStX[m+M*d]/wSt[m]; /* x */
     /*-------------------------------------------------------------o
     |   save trajectory                                            |
     o-------------------------------------------------------------*/
@@ -195,7 +205,7 @@ int bcpd(
     for(m=0;m<M;m++)for(d=0;d<D;d++){ix[d+D*m]=0;for(i=0;i<D;i++){ix[d+D*m]+=R[i+D*d]*(x[i+D*m]-t[i]);}
         ix[d+D*m]/=*s;}
     for(m=0;m<M;m++)for(d=0;d<D;d++) E[m+M*d]=ix[d+D*m]-Y[d+D*m];
     if(K){/* CASE: low-rank, NOTE: C=d(P1)*Q, E=ix-Y */
-      for(m=0;m<M;m++)for(k=0;k<K;k++) C[m+M*k]=w[m]*Q[m+M*k];
+      for(m=0;m<M;m++)for(k=0;k<K;k++) C[m+M*k]=wSt[m]*Q[m+M*k];
       for(k=0;k<K;k++)for(d=0;d<D;d++){B[k+K*d]=0;for(m=0;m<M;m++) B[k+K*d]+=C[m+M*k]*E[m+M*d];}
       #pragma omp parallel for private (j) private (m) private(val)
       for(i=0;i<K;i++)for(j=i;j<K;j++){val=0;for(m=0;m<M;m++){val+=Q[m+M*i]*C[m+M*j];} A[i+K*j]=A[j+K*i]=val;}
@@ -208,9 +218,9 @@ int bcpd(
       if(db){tr=0;
         #pragma omp parallel for private (i) private (j) private(val)
         for(m=0;m<M;m++){val=0;for(i=0;i<K;i++)for(j=0;j<K;j++){val+=A[i+K*j]*Q[m+M*i]*Q[m+M*j];} sgm[m]=val/lmd;}
-        for(m=0;m<M;m++){tr+=w[m]*sgm[m];} tr*=D;
+        for(m=0;m<M;m++){tr+=wSt[m]*sgm[m];} tr*=D;
       }
-      for(m=0;m<M;m++)for(d=0;d<D;d++) W[d+D*m]=w[m]*E[m+M*d];
+      for(m=0;m<M;m++)for(d=0;d<D;d++) W[d+D*m]=wSt[m]*E[m+M*d];
       for(m=0;m<M;m++)for(d=0;d<D;d++)for(k=0;k<K;k++) W[d+D*m]-=C[m+M*k]*B[k+K*d];
       for(m=0;m<M;m++)for(d=0;d<D;d++) W[d+D*m]/=cc;
       /* y */
@@ -219,7 +229,7 @@ int bcpd(
     }
     else{/* CASE: full-rank */ //char trs='t'; double one=1;
       #pragma omp parallel for private (j)
-      for(i=0;i<M;i++)for(j=i;j<M;j++) G1[i+M*j]=G1[j+M*i]=G[i+M*j]+(i==j?cc/w[i]:0);
+      for(i=0;i<M;i++)for(j=i;j<M;j++) G1[i+M*j]=G1[j+M*i]=G[i+M*j]+(i==j?cc/wSt[i]:0);
       dposv_(&uplo,&M,&D,G1,&M,E,&M,&info); if(info!=0){goto err06;}
       for(m=0;m<M;m++)for(d=0;d<D;d++) u[d+D*m]=Y[d+D*m];
       #pragma omp parallel for private (d) private (i)
@@ -227,7 +237,7 @@ int bcpd(
       if(db){tr=0;
         for(i=0;i<M;i++)for(j=i;j<M;j++) G2[i+M*j]=G2[j+M*i]=G[i+M*j];
         dpotrs_(&uplo,&M,&M,G1,&M,G2,&M,&info);if(info!=0){goto err07;}
-        for(m=0;m<M;m++){sgm[m]=G2[m+M*m]*SQ(*r/(*s))/w[m]; tr+=sgm[m];} tr*=D;
+        for(m=0;m<M;m++){sgm[m]=G2[m+M*m]*SQ(*r/(*s))/wSt[m]; tr+=sgm[m];} tr*=D;
       }
     }
     /*-------------------------------------------------------------o
@@ -239,10 +249,10 @@ int bcpd(
     if(pm.opt&PW_OPT_NOSIM) goto skip;
     /* xb, yb */
```

```
-    for(d=0;d<D;d++){val=0;for(m=0;m<M;m++){val+=u[d+D*m]*w[m];} ub[d]=val/(*Np);}
-    for(d=0;d<D;d++){val=0;for(m=0;m<M;m++){val+=x[d+D*m]*w[m];} xb[d]=val/(*Np);}
+    for(d=0;d<D;d++){val=0;for(m=0;m<M;m++){val+=u[d+D*m]*wSt[m];} ub[d]=val/NpSt;}
+    for(d=0;d<D;d++){val=0;for(m=0;m<M;m++){val+=x[d+D*m]*wSt[m];} xb[d]=val/NpSt;}
     /* Sxv */
-    for(i=0;i<D;i++)for(j=0;j<D;j++){val=0;for(m=0;m<M;m++){val+=w[m]*(x[i+D*m]-xb[i])*(u[j+D*m]-ub[j]);}
-        Sxu[i+D*j]=val;}
+    for(i=0;i<D;i++)for(j=0;j<D;j++){val=0;for(m=0;m<M;m++){val+=wSt[m]*(x[i+D*m]-xb[i])*(u[j+D*m]-ub[j]);}
+        Sxu[i+D*j]=val;}
     for(i=0;i<D;i++)for(j=0;j<D;j++){val=0;for(d=0;d<D;d++){val+=Sxu[d+D*i]*Sxu[d+D*j];} psi[i+D*j]=val;}
     /* R */
     dsyev_(&jobz,&uplo,&D,psi,&D,dS,wk,&lwork1,&info);
@@ -255,7 +265,7 @@ int bcpd(
     /* s */
     c1=c2=0; if(db) c2+=tr/(*Np);
     for(i=0;i<D;i++)for(j=0;j<D;j++) c1+=R[i+D*j]*Sxu[i+D*j];
-    for(d=0;d<D;d++)for(m=0;m<M;m++) c2+=w[m]*(SQ(u[d+D*m]-ub[d]));
+    for(d=0;d<D;d++)for(m=0;m<M;m++) c2+=wSt[m]*(SQ(u[d+D*m]-ub[d]));
     *s=c1/c2;
     /* t */
     for(d=0;d<D;d++){val=0;for(i=0;i<D;i++)val+=R[d+D*i]*ub[i];t[d]=xb[d]-(*s)*val;}
@@ -267,8 +277,8 @@ int bcpd(
     o-----------------------------------------------------------------*/
     rold=*r; *r=0; if(db) *r+=SQ(*s)*tr;
     for(n=0;n<N;n++)for(d=0;d<D;d++) *r+=SQ(X[d+D*n])*f[n];
-    for(m=0;m<M;m++)for(d=0;d<D;d++) *r+=SQ(y[d+D*m])*w[m];
-    for(m=0;m<M;m++)for(d=0;d<D;d++) *r-=2*PX[m+M*d]*y[d+D*m];
+    for(m=0;m<M;m++)for(d=0;d<D;d++) *r+=SQ(y[d+D*m])*wSt[m];
+    for(m=0;m<M;m++)for(d=0;d<D;d++) *r-=2*PStX[m+M*d]*y[d+D*m];
     if(*Np<0){goto err05;}
     *r/=(*Np)*D; *r=fabs(*r); *r=sqrt(*r);
     diff=fabs(rold-*r);
```

References

[BBL+20] Virginia Best, Robert Baumgartner, Mathieu Lavandier, Piotr Majdak, and Norbert Kopco. Sound externalization: A review of recent research. *Trends in Hearing*, 24:2331216520948390, 2020.

[DPT+08] Matteo Dellepiane, Nico Pietroni, Nicolas Tsingos, Manuel Asselot, and Roberto Scopigno. Reconstructing head models from photographs for individualized 3dâaudio processing. *Computer Graphics Forum*, 27, 2008.

[GS18] Corentin Guezenoc and Renaud Seguier. HRTF Individualization: A Survey. *Journal of the Audio Engineering Society*, October 2018.

[HGD16] Henrik Gert Hassager, Fredrik Gran, and Torsten Dau. The role of spectral detail in the binaural transfer function on perceived externalization in a reverberant environment. *The Journal of the Acoustical Society of America*, 139(5):2992–3000, 2016.

[Hir21a] Osamu Hirose. A bayesian formulation of coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2269–2286, 2021.

[Hir21b] Osamu Hirose. Supplementary document to: A bayesian formulation of coherent point drift, 2021. https://www.dropbox.com/s/pkgw2xxd0f3anfk/bcpd-appendix.pdf?dl=1.

[Hir22] Osamu Hirose. Geodesic-based bayesian coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2022.

[IAK22] Kazuhiro Iida, Tsubasa Aizaki, and Takeshige Kikuchi. Toolkit for individualization of head-related transfer functions using parametric notch-peak model. *Applied Acoustics*, 189:108610, 02 2022.

[LLPP16] Thibaud Leclere, Mathieu Lavandier, Kevin Perreaut, and Fabien Perrin. Externalization of sounds with non-individualized head-related transfer functions and without visual information. *The Journal of the Acoustical Society of America*, 140(4 Supplement):3276–3276, 2016.

[LP20] Song Li and Jürgen Peissig. Measurement of head-related transfer functions: A review. *Applied Sciences*, 10(14), 2020.

[MS10] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[MZY16] Jiayi Ma, Ji Zhao, and Alan L. Yuille. Non-rigid point set registration by preserving global and local structures. *Trans. Img. Proc.*, 25(1):53â64, jan 2016.

[PKM22] K. Pollack, W. Kreuzer, and P. Majdak. *Modern Acquisition of Personalised Head - Related Transfer Functions: An Overview*, pages 25–61. IntechOpen, London, 2022.

[PMF21] K. Pollack, P. Majdak, and H. Furtado. Evaluation of pinna point cloud alignment by means of non-rigid registration algorithms. 2021.

[PMF23a] K. Pollack, P. Majdak, and H. Furtado. Application of non-rigid registration to photogrammetrically reconstructed pinna point clouds for the calculation of personalised head-related transfer functions. pages 1094 – 1098, Hamburg, 2023.

[PMF23b] K. Pollack, P. Majdak, and H. Furtado. Evaluation of various non-rigid registration algorithms to increase accuracy of photogrammetric reconstruction of human pinnae by means of geometric error. 2023.

[TMK+12] Hironori Takemoto, Parham Mokhtari, Hiroaki Kato, Ryouichi Nishimura, and Kazuhiro Iida. Mechanism for generating peaks and notches of head-related transfer functions in the median plane. *The Journal of the Acoustical Society of America*, 132(6):3832–3841, 12 2012.