

Anatomical decomposition of human liver volume to build accurate statistical shape models

Mahdi Delavari¹ · Amir Hossein Foruzan¹

Received: 14 May 2016 / Revised: 16 July 2017 / Accepted: 7 August 2017 / Published online: 21 August 2017
© Springer-Verlag London Ltd. 2017

Abstract Liver is considered as a tissue with large shape variations. Representation of such a complex shape cannot be accurately performed using conventional SSM algorithms. We propose to decompose a human liver into its anatomical parts and use them as a guide when considering point correspondences. To cope with shape complexity, a modified coherent point drift (CPD) algorithm is proposed too. The modified CPD algorithm assigns fuzzy correspondences to points and follows a simulated annealing approach to convert fuzzy correspondences into binary ones. Our modification includes automatic parameter settings which results in robustness of the algorithm. The proposed algorithm was compared to the thin plate spline-robust point matching (TPS-RPM) and minimum description length (MDL) techniques. Our method is twice faster than the MDL algorithm. Compared to the TPS-RPM algorithm, our method improved mean Specificity of the right lobe by 0.09 and Compactness of the model by two less modes.

Keywords Anatomical shape decomposition · Liver shape modeling · Modified coherent point drift · Point correspondences · Statistical shape model

1 Introduction

Statistical shape models (SSMs) have several applications including shape deformation analysis [1] and shape recon-

struction [2]. Establishing correct correspondences between the points of two shapes is a crucial task in developing an SSM, and three types of methods have been proposed [3].

In the 1st group, one shape is registered to the other using a rigid/non-rigid registration scheme. Then, one assigns pairs of the nearest points as corresponding. These techniques are not suitable for non-rigid bodies. Non-rigid multimodality registration is also sensitive to intensity variations [4].

In the 2nd group, one does an exhaustive search on all landmarks which is a tedious task and sometimes impossible [3,4].

The 3rd group uses iterative methods to estimate transformation function and to resolve point correspondences in an (expectation maximization) EM-like [5] approach. Several examples are the ICP (iterative closest point) [6] and its variants, the TPS-RPM (thin plate spline robust point matching) [3] and CPD (coherent point drift) [7]. This class of algorithms has the best performance among others. However, they do not consider morphological/anatomical characteristics of shapes. Involving shape geometry in building an SSM is important especially in cases of highly deformable bodies such as human liver.

2 Previous works

The TPS-RPM method uses the EM scheme to find fuzzy correspondences between shape points. It employs a Gaussian mixture and assigns the set of moving points as the center of the Gaussians. In the E-step, a fuzzy correspondence is assigned to the set of reference points. In the M-step, the transformation matrix is estimated to transfer the moving points to new positions. This method is sensitive to parameter setting.

The CPD method uses Gaussian mixture and the EM algorithm to iteratively register a moving shape to a reference one.

✉ Amir Hossein Foruzan
a.foruzan@shahed.ac.ir

Mahdi Delavari
Md.Delavari@Shahed.ac.ir

¹ Department of Biomedical Engineering, Engineering Faculty, Shahed University, Tehran, Iran

In contrast to the TPS-RPM, it applies a coherency constraint on shape movement that preserves shape topology during registration. Compared to the TPS-RPM, the sensitivity of the CPD method to its parameters is less, and it is faster and more robust.

The minimum description length (MDL) algorithm maps 3D reference and moving shapes onto a sphere [8]. It registers the two shapes by displacement of their points on the sphere using minimization of the model compactness. The method has large computational cost, and its convergence is not guaranteed. Heimann et al. [9] and Xu et al. [10] proposed solutions for non-uniform distribution of points. However, the problems of runtime and convergence remained.

The above methods do not consider natural/anatomical structures of shapes. Some researchers divided a shape into semantic divisions and registered each division individually [11]. Corresponding to each division, they used a Gaussian component. Manually splitting a 3D shape is not a simple task, and employing a rigid transform function is not suitable for soft tissues too. In [12], the authors employed a sequential approach to estimate number of Gaussian components. They employed a smoother objective function and a smaller search space. The algorithm was sensitive to Gaussian parameters.

Some researchers used perception rules to split a shape into semantic parts. Some of these rules have been verified in psychological studies of human being including minima rule [13], short-cut rule [14] and convexity rule [15,16]. The minima rule split a shape based on loci of negative minima of the curves that form a shape. The short-cut rule tries to minimize cut lengths. The convexity rule emphasizes on convexity of a shape to split it into parts. Ma et al. [17] introduced a new rule called “Part-Similarity rule” to distinguish similar parts of a body and combined it with convexity rule to split a shape. Compared with other methods, their results were closer to human perception. However, the outcomes were not acceptable with regard to curved branches and resulted in extra parts.

Three-dimensional extensions of the above methods are not straightforward. For medical objects, anatomical properties should be considered instead of structural characteristics. Okada et al. [18] divided a human liver shape into sections to improve accuracy of segmentation. Lamecker et al. [19] manually divided the surface of a liver into four patches. Due to the large shape variations, these patches are not easily distinguishable.

Anatomically, a physician divides human liver into four lobes: right, left, caudate and quadrate [20]. In this paper, we propose a new shape model which considers anatomical characteristics of liver. One used the bifurcations of portal veins to divide liver into left and right lobes. Individual models are built for each lobe and the complete hepatic model is the augmented submodels. Our main innovations are to explicitly deal with anatomical structures of the tissue, optimization

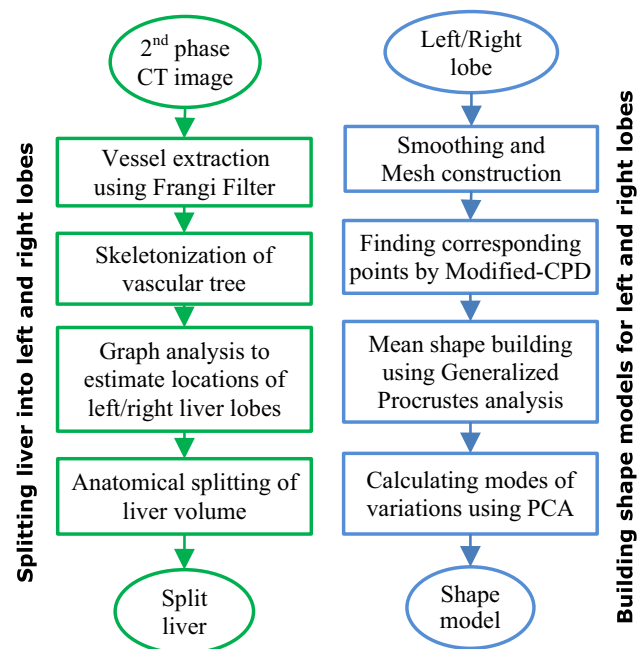


Fig. 1 The flowchart of the proposed method. The *left* column describes splitting a liver into anatomical divisions. The *right* column explains how an SSM is built for each lobe

and conscious choice of registration parameters, decreasing computational cost, and developing a robust algorithm. In Sect. 3, we describe the proposed method. We present the results in Sect. 4 and discuss them in Sect. 5. Section 6 concludes the paper.

3 Proposed method

The flowchart of our method has two branches (Fig. 1): The left column describes splitting a liver into anatomical divisions, and the right column explains how we build an SSM for each lobe.

3.1 Splitting a liver into its left and right lobes

In an earlier research, we built a single model for a liver volume [21]. Consider a liver with a small left lobe that is to be registered on a liver with a large left lobe. In such a case, some points on the right lobe of the first liver are erroneously assigned to the left lobe points of the second liver. Therefore, considering separate models for left and right lobes increases accuracy of the model. A novelty of our method is to use anatomical landmarks to divide a liver into two parts and constructing individual models for a part. A physician divides the hepatic parenchyma into left, right, caudate, and quadrate lobes. Due to small volumes of caudate and quadrate lobes

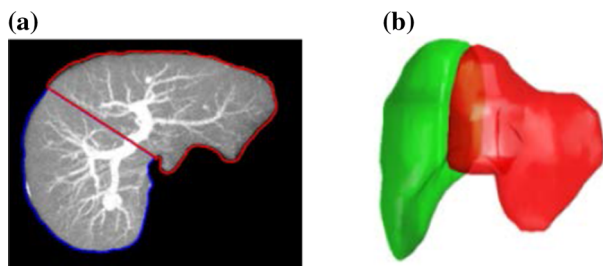


Fig. 2 **a** Dividing a liver into left and right lobes. **b** The two lobe volumes

compared to left/right lobes, we do not consider them when partitioning a liver.

Since separate branches of vessels feed a lobe, we use bifurcation of portal veins that are visible in 2nd phase images to divide the gland into two parts (Fig. 2a). To enhance portal veins more, we use the multiscale filter proposed by Frangi et al. [22].

We segment and extract the skeleton of the enhanced veins based on the method of Lee et al. [23]. Using the skeletons of the portal veins, we use graph analysis to distinguish left and right branches of the vessels. Based on its distance to the nearest vascular branches, each voxel is labeled as either left or right lobe (Fig. 2b). In a case where the contrast between vessels and the liver tissue was not enough, the results of vessel enhancement and skeletonization were not acceptable. A physician manually divided a liver using maximum intensity projection (MIP) of the axial view. A plane was fitted to branching place of portal veins which split the gland into left and right lobes.

3.2 Modified CPD

Since the surface of a segmented liver is not smooth, we employ a $3 \times 3 \times 3$ smoothing filter. The liver images are zero-padded to keep the image size unchanged.

Each lobe is then represented by a mesh using the marching cube algorithm [24]. We sample liver shapes by 1000 points according to our previous work [21]. Regarding the voxel dimensions ($0.625 \times 0.625 \times 1.25 \text{ mm}^3$), the minimum distance between mesh points cannot be less than 0.625 mm. In other words, the minimum surface of a mesh cannot be less than 0.195 mm^2 ($0.625 \times 0.625/2$). We built meshes of all training shapes with 500, 1000, and 2000 points and measured the least surface of the meshes. For the shapes sampled by 2000 points, the least mesh surface of some training shapes were less than 0.195 mm^2 . It means that some voxels are sampled with at least two points which is not meaningful and it just increases computational time. On the other hand, the least mesh surface of some shapes sampled by 500 points was bigger than 0.195 mm^2 which may ignore some critical parts of a liver. However, if we sample a liver shape by

1000 points, mesh surfaces will be of the proper sizes. After splitting a liver, we represent use the larger right lobe by 601 points to and the smaller left lobe by 401. Next, we use the modified CPD algorithm to set up point correspondences.

The modified CPD is another novelty in our method to build statistical models of a shape with large variations. The CPD method simultaneously finds a non-rigid transformation matrix and point correspondences. It imposes a motion coherence constraint [25] on the velocity field of point movements which enforces the neighboring points to move coherently. This preserves shape topology during registration process. It considers registration of two point sets as a problem of estimating the probability density of one point-set using another point-set density. It achieves this goal by maximizing the posterior probability or maximizing the likelihood function equivalently. To do this, the points of the moving mesh (M points) are considered as centers of a Gaussian mixture and they are registered to the fixed mesh points (N points). Considering Y_0 as the initial position of the centers, new positions of the template mesh are obtained by Eq. 1.

$$Y = v(Y_0) + Y_0 \quad (1)$$

In Eq. 1, v is a continuous velocity field of template point movements. Using the Bayes theorem, we calculate the posterior probability and the likelihood function of moving points Y by Eq. 2 and Eq. 3, respectively.

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \quad (2)$$

$$\begin{aligned} p(X|Y) &= \sum_{m=1}^M p(X|y_m) = \prod_{n=1}^N \sum_{m=1}^M p(x_n|y_m) \\ &= \prod_{n=1}^N \sum_{m=1}^M e^{-\frac{1}{2} \left(\left\| \frac{x_n - y_m}{\sigma} \right\|^2 \right)} \end{aligned} \quad (3)$$

In Eq. 2, $p(X)$ and $p(Y)$ are prior probability functions of the reference and template shapes respectively. In Eq. 3, variances of the Gaussian components (σ) are equal. In order to impose a smooth motion constraint, the prior probability of moving mesh is defined in Eq. 4.

$$p(Y) = e^{-\frac{\lambda}{2} \phi(Y)} \quad (4)$$

In Eq. 4, λ is a weighting constant and $\phi(Y)$ is a function which regularizes motion smoothness. Assuming linearly independence and using minus log-likelihood, Eq. 4 is rewritten as Eq. 5.

$$\begin{aligned} -\log(p(Y|X)) &= -\log(p(X|Y)) - \log(p(Y)) \\ &\quad + \log(p(X)) \end{aligned} \quad (5)$$

By omitting independent parts from Y and substituting Eqs. 3 and 4 with Eq. 5, we write the objective function as Eq. 6.

$$E(\mathbf{Y}) = -\sum_{n=1}^N \log \sum_{m=1}^M e^{-\frac{1}{2} \left(\left\| \frac{x_n - y_m}{\sigma} \right\|^2 \right)} - \left(-\frac{\lambda}{2} \phi(\mathbf{Y}) \right) \quad (6)$$

Minimizing the objective function leads to minimizing the fixed and moving point distances which is equivalent to maximizing the posterior probability of the moving mesh.

We restrict the velocity field of the moving points \mathbf{Y} to be smooth. A smooth function has less energy in high frequencies. Thus, we obtain a smooth function through multiplying its frequency transform by a high-pass Gaussian filter or equivalently dividing by a low-pass Gaussian filter (Eq. 7).

$$\phi(\mathbf{v}) = \int \frac{|\tilde{v}(s)|^2}{\tilde{G}} ds \quad (7)$$

In Eq. 7, \tilde{v} is the Fourier transform of the velocity field and \tilde{G} is a symmetric, positive-definite and low-pass Gaussian filter that becomes smaller as $\|s\| \rightarrow \infty$. By putting Eq. 7 back into Eq. 6, we write the objective function as Eq. 8.

$$E(\tilde{\mathbf{v}}) = -\sum_{n=1}^N \log \sum_{m=1}^M e^{-\frac{1}{2} \left(\left\| \frac{x_n - y_m}{\sigma} \right\|^2 \right)} + \frac{\lambda}{2} \int \frac{|\tilde{v}(s)|^2}{\tilde{G}} ds \quad (8)$$

Minimizing Eq. 8 leads to minimizing the high-frequency contents of the point movement function which causes the movements to be smooth. By substituting Eq. 1 in Eq. 8 and calculating its derivation with respect to \mathbf{v} , it can be shown that the velocity function which minimizes Eq. 8 is rewritten as Eq. 9.

$$\mathbf{v}(z) = \sum_{m=1}^M w_m \mathbf{G}(z - y_{0m}) \quad (9)$$

In Eq. 9, w_m is a weighting factor and \mathbf{G} is a Gaussian mixture kernel which is different from the Gaussian mixture of points \mathbf{Y} . This kernel goes to zero as $\|s\| \rightarrow \infty$ which satisfies the velocity field smoothness. Also, the smoothness of the velocity function can be controlled by tuning parameters of the Gaussian kernel. By choosing a Gaussian kernel for the velocity field, Eq. 8 becomes similar to MCT (Motion Coherence Theory) [25] as well. The second term of Eq. 8 is equal to a weighted sum of all orders of the squared derivations of the velocity field (Eq. 10).

$$\frac{\lambda}{2} \int \frac{|\tilde{v}(s)|^2}{\tilde{G}} ds \rightarrow \frac{\lambda}{2} \int \sum_{m=1}^{\infty} \frac{\beta^{2m}}{m! 2^m} (D^m \mathbf{v})^2 \quad (10)$$

In Eq. 10, D is the derivative operator defined in Eqs. 11 and 12.

$$D^{2m} \mathbf{v} = \nabla^{2m} \mathbf{v} \quad (11)$$

$$D^{2m+1} \mathbf{v} = \nabla \left(\nabla^{2m} \mathbf{v} \right) \quad (12)$$

By substituting Eq. 10 for Eq. 8, the energy function is rewritten as Eq. 13.

$$E(\mathbf{W}) = -\sum_{n=1}^N \log \sum_{m=1}^M e^{-\frac{1}{2} \left(\left\| \frac{x_n - y_{0m} - \sum_{k=1}^M w_k \mathbf{G}(y_{0k} - y_{0m})}{\sigma} \right\|^2 \right)} + \frac{\lambda}{2} tr(\mathbf{W}^T \mathbf{G} \mathbf{W}) \quad (13)$$

In Eq. 13, \mathbf{W} is the matrix of Gaussian kernel weights of velocity field and \mathbf{G} is a symmetric square matrix. We calculate an element of \mathbf{G} by Eq. 14.

$$g_{ij} = e^{-\frac{1}{2} \left\| \frac{y_{0i} - y_{0j}}{\beta} \right\|^2} \quad (14)$$

Each element of \mathbf{G} has a value between 0 and 1. In Eq. 14, β is the effective radius of each Gaussian component with the center of y_{0i} and it determines the number of neighboring points of y_{0i} which are imposed to move coherently. By utilizing the EM algorithm, an upper bound for Eq. 13 can be found in the form of Eq. 15. Minimizing Eq. 15 leads to minimization of Eq. 13.

$$Q(\mathbf{W}) = \sum_{n=1}^N \sum_{m=1}^M \mathbf{P}^{old}(m|x_n) \frac{\|x_n - y_{0m} - \mathbf{G}(m, \cdot) \mathbf{W}\|^2}{2\sigma^2} + \frac{\lambda}{2} tr(\mathbf{W}^T \mathbf{G} \mathbf{W}) \quad (15)$$

In Eq. 15, $G(m, \cdot)$ is the m th row of the matrix \mathbf{G} and \mathbf{P}^{old} is the posterior probability matrix which is calculated from current places of the reference points with respect to the Gaussian centers. We obtain an element of \mathbf{P}^{old} using Eq. 16.

$$p_{mn} = e^{-\frac{1}{2} \frac{y_m^{old} - x_n}{\sigma}^2} / \left(\frac{(2\pi\sigma^2)^{\frac{d}{2}}}{\alpha} + \sum_{m=1}^M e^{-\frac{1}{2} \frac{y_m^{old} - x_n}{\sigma}^2} \right) \quad (16)$$

In Eq. 16, y_m^{old} is the current location of a Gaussian center. A uniform distribution is included in the denominator of the p_{mn} to model noise and outliers. Regarding the uniform distribution, d is the dimension of the space and α determines the effect of the uniform distribution in p_{mn} .

In order to minimize the energy function, we calculate the derivation of Eq. 15 with respect to W set it to zero (Eq. 17).

$$\left\{ \text{diag}(\mathbf{P1}) \mathbf{G} + \lambda \sigma^2 \mathbf{I} \right\} \mathbf{W} = \mathbf{P}\mathbf{X} - \text{diag}(\mathbf{P1}) \mathbf{Y}_0 \quad (17)$$

In Eq. 17, $\mathbf{P1}$ is the posterior probability matrix with a column of ones and $\text{diag}(\cdot)$ is a diagonal matrix. Solving Eq. 17 for W is performed in the M-step and calculating the posterior probability matrix \mathbf{P} is done in the E-step of the EM algorithm. If the EM algorithm is converged, the template points will be moved to new locations using Eq. 18.

$$\mathbf{Y} = \mathbf{Y}_0 + \mathbf{G}\mathbf{W} \quad (18)$$

Then, we reduce the coverage of the mixture components (σ) by a deterministic annealing strategy [21] (Eq. 19).

$$\sigma_{\text{new}} = \sigma_{\text{old}} \times r \quad (19)$$

In Eq. 19, r is the annealing factor in the 0.93–0.97 range. After reducing σ , the EM algorithm is repeated. The annealing scheme of σ is continued until each Gaussian component covers a single point. This leads to a one-to-one correspondence between the reference and template point sets. The steps involved in the modified CPD method are shown using Algorithm 1.

Algorithm 1 Finding corresponding points between two shapes.

Inputs: Reference shape vector \mathbf{X} , moving shape vector \mathbf{Y}_0 .

Step 1. Initialize parameters $\beta = 1.87, \sigma = 1, \lambda = 2, r = 0.95$.

Step 2. Construct elements of the \mathbf{G} matrix using Eq. 14.

Step 3. Update correspondence matrix \mathbf{P} using Eq. 16.

Step 4. Solve for \mathbf{W} in Eq. 17.

Step 5. Update moving shape points using $\mathbf{Y} = \mathbf{Y}_0 + \mathbf{G}\mathbf{W}$.

Step 6. If $\sigma > 0.028$ then

Update σ using $\sigma = \sigma_{\text{old}} \times r$.

Goto Step 3

End

Output: Registered shape vector \mathbf{Y} with points corresponding to the reference shape.

3.3 Free parameters

A novelty in our method is conscious selections of non-rigid registration parameters. The free parameters of the CPD algorithm are σ, λ , and β . In Eq. 15, the parameter λ compromises between the point registration and the motion smoothness and we set it to 2. Thus, both parts of the Eq. 15 have the same effect on the objective function. The parameter β in \mathbf{G} (Eq. 14) regularizes interactions among the points in \mathbf{Y} and determines that number of neighboring points in \mathbf{Y} which

are restricted to move together coherently. A small value of β leads to a local transformation, while a large value results in a rigid transformation. We set β to 1.87 so that all elements of \mathbf{G} are larger than 0.36. Consider the effective range of a Gaussian as (mean \pm standard deviation) the standard deviation is 0.36 of its maximum value. Thus, the farthest points in \mathbf{Y} (Gaussian centroids) have a negligible effect on each other, and thus topology of the moving shape is preserved during the registration.

The parameter σ in Eq. 15 is the standard deviation of Gaussian components. It compromises between point registration accuracy and motion smoothness. We decrease this parameter by a constant factor (between 0.93 and 0.97) during registration using the Deterministic Annealing strategy. Since we normalize the shapes before registration, we set the initial value of the parameter to 1. Thus, a point in \mathbf{X} corresponds to any point in \mathbf{Y} at the start of the process. We also set the annealing factor to 0.95.

Due to the complexity of liver shapes, if we decrease the parameter σ less than a threshold, some points in \mathbf{X} may fall far from the Gaussian contours. The transformation matrix is therefore calculated incorrectly and the Gaussian centers are moved to wrong positions. We empirically found that the threshold value for sigma is 0.028. It is equivalent to 70 iterations of the algorithm according to Eqs. 20 and 21.

$$\sigma_{\text{final}} = \sigma_{\text{init}} \times r^n \quad (20)$$

$$n = \frac{\ln \frac{\sigma_{\text{final}}}{\sigma_{\text{init}}}}{\ln r} \quad (21)$$

Proper selection of the final value of σ prevents divergence of the algorithm and helps to develop a robust technique.

3.4 Construction of the shape model

After finding corresponding points, we convert a shape in the training set to a vector and normalize it ($|\mathbf{X}| = 1$) (Eq. 22).

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{bmatrix} \rightarrow \mathbf{X} = (x_1, y_1, z_1, \dots, x_m, y_m, z_m)^T \quad (22)$$

In Eq. 22, m is the number of shape points. We align the normalized shapes using Generalized Procrustes Analysis and calculate the mean shape ($\bar{\mathbf{X}}$) by Eq. 23.

$$\bar{\mathbf{X}} = \frac{1}{S} \sum_{i=1}^S \mathbf{X}_i \quad (23)$$

In Eq. 23, S is the number of shapes in the training set. Next, we calculate the covariance matrix corresponding to the training shapes (C) by Eq. 24.

$$C = \frac{1}{S-1} \sum_{i=1}^S (X_i - \bar{X})(X_i - \bar{X})^T \quad (24)$$

If the eigenvalues of C are sorted in the descending order, we reconstruct a valid shape using a linear combination of the mean shape and the eigenvalues (Eq. 25).

$$X \approx \bar{X} + \Psi b \quad (25)$$

In Eq. 25, Ψ is the eigenvector matrix corresponding to the ordered eigenvalues and b is a parameter vector.

4 Results

4.1 Dataset

We employed the second phase of 30 computed tomography images of abdominal cavity from 16 males and 14 females in the 20–75 age range. The images were stored in 12-bit DICOM format of $512 \times 512 \times 159$ pixels with a pixel size of $0.625 \times 0.625 \times 1.25$ mm³. The images were captured at Osaka University Hospital, and a radiology specialist manually segmented the livers.

We implemented the codes in MATLAB environment and ran on a personal computer with windows 8.1 operating system, AMD-fx4100 3.6 GHz CPU and 8 GB of dynamic RAM.

Regarding the runtime of our method, it took approximately 16 and 31 s to find corresponding points of the left and right lobes, respectively. However, it took about 68 s to process a complete liver. Regarding computational cost, our anatomical decomposition technique is therefore superior to traditional methods. The time complexity of our algorithm is $O(N)$ where N is the number of the input shape points.

4.2 Evaluation measures

We used Compactness, Generalization ability, and Specificity metrics [14] to evaluate the proposed shape model.

The compactness of a model is the number of modes that are required to reconstruct a valid shape (Eq. 26). A compact model uses fewer modes to reconstruct a shape.

$$C = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^{Nd} \lambda_i} \quad (26)$$

In Eq. 26, λ_i s are eigenvalues of the covariance matrix (Eq. 24) that are sorted in the descending order, M is the total number of the modes, N is the number of points and d

is the shape dimension. The number of required modes (m) is selected such that C reaches a value between 0.9 and 0.98.

Generalization is the ability of a model to reconstruct training and unseen valid shapes. It evaluates over-fitting of the model with respect to the training set. The leave-one-out method is employed to calculate this metric for all training images. The mean value of reconstruction errors is considered as the generalization ability of the model (Eq. 27).

$$G(m) = \frac{1}{M} \sum_{i=1}^M \|x_i - x'_i(m)\|, \quad (27)$$

$$x'_i(m) = \bar{X} + \sum_{i=1}^m \Psi_i b_i$$

In Eq. 27, x_i is the leftout shape, $x'_i(m)$ is the reconstructed shape using m modes, Ψ_i is the i th eigenvector, b_i is the i th shape parameter, and M is the number of training shapes. The reconstructed shape ($x'_i(m)$) is a function of m (number of modes) and the Generalization ability is a function of m as well.

A model is specific if it represents only valid shapes. In other words, any shape that is created by the model should be similar to the training set. To compute this measure, we created 1000 random shapes using random shape vectors (\tilde{b}_A , $1 \leq A \leq 1000$). We constrained the shape vectors in $[-\sqrt{2}\lambda, +\sqrt{2}\lambda]$ range. The minimum distance of a random shape from the training images was calculated in the allowable shape domain (ASD) space. The mean of these distances is the specificity measure (Eq. 28).

$$S(m) = \frac{1}{N} \sum_{A=1}^N \min_i \|y_A(m) - x_i\|, \quad (28)$$

$$y_A(m) = \bar{X} + \sum_{i=1}^m \Psi_i b_{iA}$$

In Eq. 28, y_A is a randomly generated shape, N is the number of random shapes, and x_i is a training shape. The Specificity is a function of m (the number of modes) as well.

4.3 Evaluation of the left and right lobes

The models of the left and right lobes were compared with the complete liver model using compactness, generalization, and specificity. We compared the results qualitatively too (Fig. 3).

The compactness of a complete liver, left lobe and right lobe models are 18, 18 and 16 respectively.

We performed experiments with the MDL and the TPS-RPM algorithms using our available dataset. Regarding the run-time of the code, our method took about 1400 s to prepare a model while the MDL method took more than 3600 s to do the task. Our method is more robust compared to the MDL

as well. With respect to the TPS-RPM algorithm, our method improved the mean Specificity of the right lobe by 0.09 and the Compactness of the model by two less modes.

5 Discussion

In this paper, we built individual models for left and right lobes. It prevents the surface of a liver to be represented by a non-uniform mesh. In Fig. 4, a typical shape of a liver was

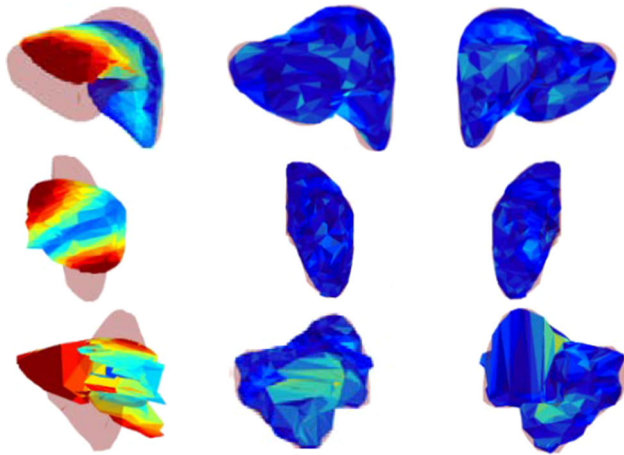


Fig. 3 Top row A typical complete liver. Middle row Left lobe. Bottom row Right lobe. Left column Before registration. Middle and right columns After registration. The reference shape is shown in lower opacity. Warm/cold colors are larger/smaller distances respectively

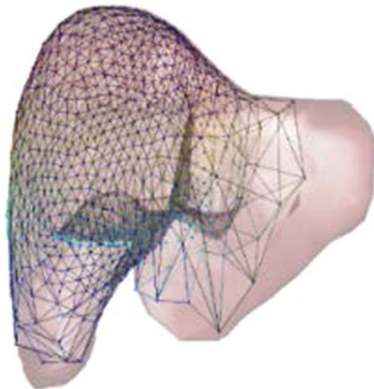


Fig. 4 Representation of a complex shape by a single model makes the corresponding mesh to be non-uniform

represented by a single model using the MDL method. As it is seen, some parts of the left lobe are not covered completely. This happens since the modes of variation corresponding to the right lobe control the left modes as well.

Based on our previous research [21], we used the modified CPD to build left and right lobe models. Regarding the results shown in Fig. 3, the moving liver shape is clearly different from the reference shape. The left lobe of the moving shape is smaller than that of the reference shape. However, the left and right lobes of the moving liver were registered to the reference liver well.

The results of the compactness reveal that the right lobe model with 16 modes is more compact compared to the left lobe and complete liver models. This means that right lobes have simpler shapes compared with the left lobe and a complete liver.

The generalization index (Table 1) shows that the complete liver model had the least reconstruction error and the reconstruction error of the right lobe model was about 0.7 mm higher. The left lobe model had the worst result due to its complex shape.

The specificity measure (Table 1) shows that the right lobe model is more specific than both the left lobe and complete liver models. Therefore generated random shapes have lower distances from the training images.

The results reveal that decomposing a shape may affect the evaluating measures and thus lead to a more accurate model. Also, taking anatomical information into account may prevent assigning wrong correspondences in biological tissues.

6 Conclusion and future works

In this paper, we presented a robust point correspondence algorithm, decomposed a complex shape into simple parts, and built individual models for each part. The proposed approach helps us to represent the surface of a complex shape with uniform meshes and to build more accurate shape models. Regarding human liver, typical applications of an accurate shape model include precise segmentation of the hepatic volume [26] and detection of liver shape deformation which is considered as a kind of disease [27].

Table 1 Comparison of compactness, generalization, and specificity of left/right lobe models with the complete liver model

Method	Complete liver	Left lobe	Right lobe
Number of modes to achieve 98% Compactness	18	18	16
Mean generalization (mm)	3.78 ± 0.95	5.56 ± 1.95	4.31 ± 1.25
Mean specificity	0.21 ± 0.06	0.34 ± 0.08	0.12 ± 0.0007

In future, we plan to divide a liver shape into more simple parts based on anatomical landmarks and develop a hierarchical model to represent complex shapes.

Acknowledgements The authors would like to thank Prof. Yoshinobu Sato and Prof. Masatoshi Hori, Osaka University, Osaka, Japan, for the use of their images in this study.

References

1. Sonat, F.A., Ercan, I., Ozdemir, S.T., et al.: Statistical shape analysis of the rat hippocampus in epilepsy. *Anat. Sci. Int.* **84**, 298 (2009)
2. Unal, G., Nain, D., Slabaugh, G., Fang, T.: Customized design of hearing aids using statistical shape learning. *Med. Image Comput. Comput. Interv.* **2008**, 518–526 (2008)
3. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.* **89**, 114–141 (2003)
4. Maes, F., Vandermeulen, D., Suetens, P.: Medical image registration using mutual information. *Proc. IEEE*. **91**, 1699–1722 (2003)
5. Moon, T.K.: The expectation-maximization algorithm. *IEEE Signal Process. Mag.* **13**(6), 47–60 (1996)
6. Besl, P.J., McKay, H.D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256 (1992)
7. Myronenko, A., Song, X.: Point set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 2262–2275 (2010)
8. Davies, R.H., Twining, C.J., Daniel Allen, P., Cootes, T.F., Taylor, C.J.: Building optimal 2D statistical shape models. *Image Vis. Comput.* **21**, 1171–1182 (2003)
9. Heimann, T., Wolf, I., Meinzer, H.: Optimal landmark distributions for statistical shape model construction. *Proc. SPIE Med. Imaging Image Process.* **6144**, 518–528 (2006)
10. Xu, R., Zhou, X., Hirano, Y., Tachibana, R., Kido, S., Fujita, H.: Improvement of MDL method by adaptive sampling on spherical parameter space. *IEICE* **111**(389), 173–178 (2012)
11. McNeill, G., Vijayakumar, S.: Part-based probabilistic point matching using equivalence constraints. In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pp. 969–976. MIT Press (2006)
12. Titsias, M.K., Williams, C.K.I.: Unsupervised learning of multiple aspects of moving objects from video. In: *Proceedings of the 10th Panhellenic conference on Advances in Informatics*, pp. 746–756. Springer (2005)
13. Hoffman, D.D., Richards, W.A.: Parts of recognition. *Cognition* **18**, 65–96 (1984)
14. Singh, M., Seyranian, G.D., Hoffman, D.D.: Parsing silhouettes: The short-cut rule. *Atten. Percept. Psychophys.* **61**, 636–660 (1999)
15. Latecki, L.J., Lakämper, R.: Convexity rule for shape decomposition based on discrete contour evolution. *Comput. Vis. Image Underst.* **73**, 441–454 (1999)
16. Walker, L.L., Malik, J.: Can convexity explain how humans segment objects into parts? *J. Vis.* **3**, 503 (2003)
17. Ma, C., Dong, Z., Jiang, T., Wang, Y., Gao, W.: A method of perceptual-based shape decomposition. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 873–880 (2013)
18. Okada, T., Linguraru, M.G., Yoshida, Y., Hori, M., Summers, R.M., Chen, Y.-W., Tomiyama, N., Sato, Y.: Abdominal multi-organ segmentation of CT images based on hierarchical spatial modeling of organ interrelations. In: *International MICCAI Workshop on Computational and Clinical Challenges in Abdominal Imaging*, pp. 173–180 (2011)
19. Lamecker, H., Lange, T., Seebae, M.: Segmentation of the liver using a 3d statistical shape model. *ZIB Tech. Rep. Zuse Institute, Berlin* (2004)
20. Couinaud, C.: Liver anatomy: Portal (and suprahepatic) or biliary segmentation. *Dig. Surg.* **16**, 459–467 (1999)
21. Delavari, M., Foruzan, A.H., Chen, Y.-W.: Improvement of statistical shape models for soft tissues using modified-coherent point drift. *IFAC-PapersOnLine* **48**, 36–41 (2015)
22. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 130–137 (1998)
23. Lee, T.-C., Kashyap, R.L., Chu, C.-N.: Building skeleton models via 3-D medial surface axis thinning algorithms. *CVGIP Graph. Model. Image Process.* **56**, 462–478 (1994)
24. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Comput. Graph.* **21**, 163–169 (1987)
25. Yuille, A.L., Grzywacz, N.M.: The motion coherence theory. In: *Second International Conference on Computer Vision*, pp. 344–353 (1988)
26. Kumar, S.S., Moni, R.S., Rajeesh, J.: Automatic liver and lesion segmentation: A primary step in diagnosis of liver diseases. *Signal Image Video Process* **1**(7), 163–172 (2013)
27. Kohara, S., Tateyama, T., Foruzan, A.H., Furukawa, A., Kanasaki, S., Wakamiya, M., Chen, Y.-W.: Application of statistical shape model to diagnosis of liver disease. In: *2010 2nd International Conference on Software Engineering and Data Mining (SEDM)*, pp. 680–683. (2010)