Fully Automatic Segmentation of the Proximal Femur Using Random Forest Regression Voting

C. Lindner*, S. Thiagarajah, J. M. Wilkinson, The arcOGEN Consortium, G. A. Wallis, and T. F. Cootes

***Abstract*—Extraction of bone contours from radiographs plays an important role in disease diagnosis, preoperative planning, and treatment analysis. We present a fully automatic method to accurately segment the proximal femur in anteroposterior pelvic radiographs. A number of candidate positions are produced by a global search with a detector. Each is then refined using a statistical shape model together with local detectors for each model point. Both global and local models use Random Forest regression to vote for the optimal positions, leading to robust and accurate results. The performance of the system is evaluated using a set of 839 images of mixed quality. We show that the local search significantly outperforms a range of alternative matching techniques, and that the fully automated system is able to achieve a mean point-to-curve error of less than 0.9 mm for 99% of all 839 images. To the best of our knowledge, this is the most accurate automatic method for segmenting the proximal femur in radiographs yet reported.**

***Index Terms*—Automatic femur segmentation, Constrained Local Models (CLMs), femur detection, Hough transform, Random Forests.**

I. Introduction

In clinical practice, plain film radiographs are widely used to assist in disease diagnosis, preoperative planning and treatment analysis. Extraction of the contours of the proximal femur from anteroposterior (AP) pelvic radiographs plays an important role in diseases such as osteoarthritis (e.g., diagnostics and joint-replacement planning) or osteoporosis (e.g., fracture detection and bone density measurements). In addition, accurately segmenting the contours of the proximal femur in radiographs allows monitoring of disease progression.

Manual segmentation of the proximal femur in radiographs is time-consuming and hard to do consistently. Our aim is to automate the segmentation procedure. Fully automatic proximal femur segmentation is challenging for several reasons: 1) The quality of radiographs may vary considerably in terms of contrast, resolution and the region of the pelvis shown. 2) AP pelvic radiographs only give a 2-D projection of what is a 3-D object, and hence are susceptible to rotational issues; the same 3-D shape may yield a different 2-D projection depending on the view point. 3) Plain film radiographs do not provide homogeneous values for the same structure due to overlapping body parts. 4) Deformities of the proximal femur may cause the loss of distinguishable radiographic key features.

Automatically extracting the contours of the proximal femur comprises two key steps. Firstly, the femur is detected in the image and secondly, the contours are segmented. We propose to use Random Forest (RF) regression voting in a sliding window approach to fully automatically segment the proximal femur.

RFs [1] describe an ensemble of decision trees trained independently on a randomized selection of features. They have been shown to be effective in a wide range of classification and regression problems [2]–[4]. Recent work on Hough Forests [5] has shown that objects can be effectively located by training RF regressors to predict the position of a point relative to the sampled region, then running the regressors over a region and accumulating votes for the likely position. To detect the proximal femur, our global search uses a RF regressor that votes for the centre of a reference frame, resulting in a Hough-like [6] response image of accumulated votes. The approximated position is then used to initialize a local search to segment the femur, combining local detectors with a statistical shape model. Following [7], we apply RF regression in the Constrained Local Model (CLM) framework to vote for the optimal position of each model point. In the CLM framework, feature detectors are run independently to generate response images for each point and then a shape model is used to find the best combination of points [8].

Using RF regression voting for both object detection and CLM-based contour extraction yields a robust and fully automatic segmentation system. We use the latter to segment the proximal femur in AP pelvic radiographs. Preliminary outputs of this work were presented in [9]. This paper expands on the latter in that it describes several improvements to the algorithm yielding significantly higher segmentation accuracy. We show additional in-depth experimental results, and all methods are evaluated on a larger data set of 839 images. We demonstrate that results are very accurate and that both the local search and the fully automatic search outperform alternative matching techniques such as Active Shape Models (ASMs) [10], Active Appearance Models (AAMs) [11], and CLMs using normalized correlation and intensity probability-based search. We believe this to be the most accurate fully automatic femur segmentation system yet published.

Manuscript received March 13, 2013; accepted April 05, 2013. Date of publication April 12, 2013; date of current version July 27, 2013. The arcOGEN Consortium is funded by Arthritis Research UK. The work of C. Lindner was supported by the Medical Research Council. *Asterisk indicates corresponding author.*

*C. Lindner is with the Centre for Imaging Sciences, The University of Manchester, M13 9PT Manchester, U.K. (e-mail: claudia.lindner@postgrad.manchester.ac.uk).

S. Thiagarajah and J. M. Wilkinson are with the Department of Human Metabolism, The University of Sheffield, S10 2RX Sheffield, U.K.

G. A. Wallis is with the Wellcome Trust Centre for Cell Matrix Research, The University of Manchester, M13 9PT Manchester, U.K.

T. F. Cootes is with the Centre for Imaging Sciences, The University of Manchester, M13 9PT Manchester, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2013.2258030

II. Related Work

There are different approaches for automatizing the segmentation of contours from medical images such as threshold-based methods, atlas-based techniques or deformable models. Our approach falls into the category of deformable models. Behiels *et al.* [12] have shown that statistical shape models can be successfully used to segment the proximal femur, given that a good initialization for the model is available. Zheng *et al.* [13] have introduced Marginal Space Learning as an effective approach for object detection. This aims to initialize a shape model by estimating pose, orientation, and scale in sequence rather than exhaustively searching the full parameter space. This speeds up the searching procedure but might not find the optimal solution.

The idea of using a Hough-like approach in combination with a deformable model to automatically segment the proximal femur is not new. Pilgram *et al.* [14] as well as Smith *et al.* [15] have suggested to use Hough straight line detection. The latter as well as the atlas-based approach by Ding *et al.* [16] rely on the results of a Canny edge detector [17] for identifying the femoral shafts. However, most published methods make assumptions about the femur pose and were tested on data sets with very similar image quality across the set. Our technique is shown to deal with a wide range of image quality and femur pose.

III. Methods

In the following, we introduce RF regression voting and CLMs both of which are at the core of the fully automated segmentation system to be presented in Section IV.

A. *Voting With Random Forest Regression*

We use RF regression in a similar manner to the Hough Forests approach [5]. However, in our case due to the consistent skeletal anatomy across individuals from one radiograph to the other, almost any part of the image can predict the area we are interested in. Hence, we do not require voting to be dependent on a class label, allowing all image structures to vote. In addition, Hough Forests use RFs whose leaves store multiple training samples. Thus, each sample produces multiple votes, allowing for arbitrary distributions to be encoded. Each leaf of our decision trees only stores the mean offset rather than the training samples.

When training trees for the voting-regression approach, we evaluate a set of points in a grid over a region of interest with displacements within the range $[-d_{\max}, +d_{\max}]$. We generate samples by extracting a set of features $\mathbf{f}(\mathbf{z})$ at each point $\mathbf{z}$. We then train a RF on the pairs $\{(\mathbf{f}_i, \mathbf{d}_i)\}$ learning to predict the most likely position(s) of the target point relative to $\mathbf{z}$. Given the samples at a particular node, we seek to select a feature and a threshold to best split the data. Let f_i be the value of one feature associated with sample i. The best threshold, t, for this feature at this node is the one which minimizes

$$G_T(t) = G(\{\mathbf{d}_i : f_i < t\}) + G(\{\mathbf{d}_i : f_i >= t\}) \quad (1)$$

where $G(S)$ is a function evaluating the set of vectors S, and $\mathbf{d}_i$ is the predicted displacement of sample i. We aim to minimize the entropy in the branches when splitting the nodes using

$$G(\{\mathbf{d}_i\}) = Nlog|\Sigma| \quad (2)$$

where N is the number of displacements in $\{\mathbf{d}_i\}$ and Σ the respective covariance matrix. Note that this approach is somewhat analogous to a recent proposal by Girshick *et al.* [3], who showed that a tree structure generated by minimizing a classification measure can lead to good results for regression. We stop splitting the nodes when either the tree has reached its maximal depth or a minimum sample number per node. Following [18], training can be speeded up by only using a random subset of the available data at each node to select the feature and threshold.

Each leaf of our decision trees stores the mean offset and the standard deviation of the displacements of all training samples that arrived at that leaf. During search, these predictions are used to vote for the best position in an accumulator array. Predictions are made using a single vote per tree yielding a Hough-like response image. To blur out impulse responses we slightly smooth the response image with a Gaussian.

In the following, we use Haar features [19] as they have been found to be effective for a range of applications and can be calculated efficiently from integral images.

B. *Constrained Local Models*

CLMs combine global shape constraints with local models of the pattern of intensities. Based on a number of landmark points outlining the contour of the object in a set of images, we train a statistical shape model by applying principal component analysis (PCA) to the aligned shapes [10]. This yields a linear model of shape variation which represents the position of each landmark point l using

$$\mathbf{x}_l = T_{\boldsymbol{\kappa}}(\bar{\mathbf{x}}_l + \mathbf{P}_l\mathbf{b} + \mathbf{r}) \quad (3)$$

where $\bar{\mathbf{x}}_l$ is the mean position of the point in the reference frame, $\mathbf{P}_l$ is a set of modes of variation, $\mathbf{b}$ are the shape model parameters, $\mathbf{r}$ allows small deviations from the model, and $T_{\boldsymbol{\kappa}}$ applies a global transformation (e.g., similarity) with parameters $\boldsymbol{\kappa}$. Similar to ASMs [10], CLMs combine this model of shape variation with local models which search for each landmark independently.

To match the model to a new image, we seek the shape and pose parameters $\{\mathbf{b}, \boldsymbol{\kappa}\}$ which optimize the fit of the model to the image. To allow for scale and pose variations across the data set, all search operations are done in a standardized reference frame. Given an initial estimate of pose and shape parameters, the region of interest of the image is resampled into the reference frame and an area around each landmark point (with displacements in the range $[-d_{\text{search}}, +d_{\text{search}}]$) is searched. At every position in the search area a response value is computed, indicating the quality-of-fit of the local patch model to the image at that point. These values are stored in a response image $\mathbf{R}_l$.

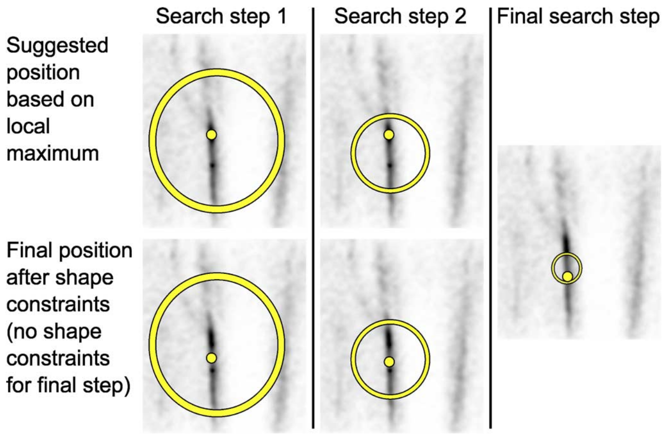

Search step 1
Search step 2
Final search step
Suggested position based on local maximum
Final position after shape constraints (no shape constraints for final step)

Fig. 1. Searching for landmark 0. During search the search area is narrowed down step-by-step, updating shape and pose parameters after each step and relaxing the shape constraints for the final step. This approach uses the shape constraints to help disambiguate multiple local optima and avoid false matches.

This is done for all n landmarks independently. We then find the shape and pose parameters which optimize

$$Q\left(\{\mathbf{b},\boldsymbol{\kappa}\}\right) = \Sigma_{l=1}^{n}\mathbf{R}_l\left(T_{\boldsymbol{\kappa}}(\bar{\mathbf{x}}_l + \mathbf{P}_l\mathbf{b} + \mathbf{r})\right). \quad (4)$$

This is done in an iterative manner applying $k \geq 1$ search steps. Each search step involves: *i*) finding the best response for each landmark in a disk of radius r about the current positions; *ii*) fitting the shape model to the resulting points (solving (3) for $\mathbf{b}, \boldsymbol{\kappa}$ when $\mathbf{r} = \mathbf{0}$); *iii*) updating all landmark positions using (3). Disk radius r is reduced by setting $r = r * 0.6$ and the process is repeated until $r \leq r_{\min}$ (see Fig. 1). In the final search step we relax the shape constraints, replacing *iii*) with a calculation of $\mathbf{r} = T_{\kappa}^{-1}(\mathbf{x} - (\bar{\mathbf{x}} + \mathbf{P}\mathbf{b}))$. We found this relaxation to significantly improve the segmentation result. In all our experiments below, we set the initial disk radius to match d_{search} and we set the number of search steps k such that $d_{\text{search}} * 0.6^k \leq 0.4$ (i.e., $r_{\min} = 0.4$).

IV. Fully Automatic Segmentation System

The fully automatic segmentation system comprises a global search detecting the object and a local search segmenting the contours. Both global and local search use RF regression voting to predict object and point positions. For the local search, we apply RF regression voting in the CLM framework.

Fig. 2 gives a summary of the fully automatic segmentation system. In the following, we describe each step in detail.

A. Object Detection

1) Training: A reference frame, or bounding box, is set so as to capture the object to be detected. Here, two reference points are used to give the horizontal axis of the reference frame. (In our case, to later use the global search results to initialize the local search, we use two points that are part of the local search shape model.) These two reference points have a fixed position within the reference frame coordinate system. They determine the reference frame's position, orientation and scale and hence the area of the image captured. For each training image, a number of random displacements (in scale, angle, and position) of the bounding box are sampled. This makes the detector

```
OBJECT DETECTION
Input: $\mathbf{I}$, $N_{fits}$
Output: $\mathfrak{r}_1, \mathfrak{r}_2$
Initialise: $\mathfrak{c} \leftarrow \emptyset$, $\mathfrak{Q} \leftarrow \emptyset$
for all $angles\ \theta\ and\ scales\ s$ do
    $\mathbf{R} \leftarrow$ getResponseImage($T_{\theta,s}(\mathbf{I})$)
    $\mathfrak{c} \leftarrow \bigcup T_{\theta,s}^{-1}$(findLocalMaxima($\mathbf{R}$))
$\mathfrak{c} \leftarrow$ sortMaximaBySoV($\mathfrak{c}$)
$\mathfrak{c} \leftarrow$ returnBestFits($N_{fits}, \mathfrak{c}$)
$\mathfrak{r}_1, \mathfrak{r}_2 \leftarrow$ getRefPointPositions($\mathfrak{c}$)

Subroutine: getResponseImage($\mathbf{I}$)
Initialise: $\mathbf{R}(:,:) \leftarrow 0$
for $i \leftarrow 1$ to $width(\mathbf{I})$ step $3$ do
    for $j \leftarrow 1$ to $height(\mathbf{I})$ step $3$ do
        $\mathbf{P} \leftarrow$ samplePatch($\mathbf{I}(i,j)$)
        for all $trees\ t\ in\ forest$ do
            $\mathbf{f} \leftarrow$ getFeatureValues($\mathbf{P}$)
            $(\delta_x, \delta_y, w) \leftarrow$ getLeafData($\mathbf{f}, t$)
            $\mathbf{R}(i+\delta_x, j+\delta_j) \leftarrow + \ w$
$\mathbf{R} \leftarrow$ GaussianSmoothResponseImage($\mathbf{R}$)

OBJECT SEGMENTATION
Input: $\mathbf{I}$, $\mathfrak{r}_1\ \mathfrak{r}_2$, $\bar{\mathbf{x}}$, $\mathbf{P}$, $N_{search}$
Output: $\mathbf{l}_{xy}$, $Q$
$model \leftarrow$initShapeModel($\bar{\mathbf{x}}, \mathbf{P}$)
for all $candidates\ c\ in$ $(\mathfrak{r}_1, \mathfrak{r}_2)$ do
    $\mathbf{l}_{xy}^{c}(:) \leftarrow 0$; $Q^c \leftarrow 0$
    $\mathbf{l}_{xy}^{c} \leftarrow$ initLandmarks($l^{r_1}, l^{r_2}, \mathbf{r}_1^c, \mathbf{r}_2^c$)
    $\mathbf{b}, \kappa \leftarrow$ fitModelToLandmarks($\mathbf{l}_{xy}^{c}, model$)
    $\mathbf{b}, \kappa \leftarrow$ setModelToMeanShape($\mathbf{b}, \kappa, model$)
    for $s \leftarrow 1$ to $N_{search}$ do
        $votes, \mathbf{b}, \kappa \leftarrow$ runSearch($\mathbf{I}, \mathbf{b}, \kappa, model$)
        if $votes > Q^c$ then
            $Q^c \leftarrow votes$
            $\mathbf{l}_{xy}^{c} \leftarrow T_{\kappa}(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b})$
$\mathbf{l}_{xy}, Q \leftarrow$ findMaxQoFCandidate($\{(\mathbf{l}_{xy}^{c}, Q^c)\}$)

Subroutine: runSearch($\mathbf{I}, \mathbf{b}, \kappa, model$)
for all $landmarks\ l$ do
    $\mathbf{S} \leftarrow$ sampleSearchAreaOfLandmark($\mathbf{I}, l$)
    $\mathbf{R}_l \leftarrow$ getResponseImage($\mathbf{S}$)
$\mathfrak{R} \leftarrow \bigcup_{l=1}^{n} \mathbf{R}_l$
$\mathbf{b}, \mathbf{r}, \kappa \leftarrow$ fitModelToResponseImages($\mathfrak{R}, model$)
$votes \leftarrow \Sigma_{l=1}^{n}\mathbf{R}_l(T_{\kappa}(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b} + \mathbf{r}))$
```

Fig. 2. Fully automatic segmentation system for a single image $\mathbf{I}$ using the N_{fits} best fits of the detector to initialize the segmentation. For each candidate obtained from the detector, the local model runs N_{search} search iterations, iteratively updating shape and pose parameters $(\mathbf{b}, \boldsymbol{\kappa})$ after each iteration. The segmentation output are the predicted positions $\mathbf{l}_{xy}$ of all landmarks from the candidate with the best quality-of-fit value Q.

scale and pose invariant within a local range. To train the object detector, for every sample i we extract features $\mathbf{f}_i$ at a set of random positions within the sampled patch and store displacement $\mathbf{d}_i$. The latter defines the difference in x and y (within the reference frame coordinate system) from the original centre of the reference frame to the centre of the displaced sample. We then train a RF on the pairs $\{(\mathbf{f}_i, \mathbf{d}_i)\}$ where all trees are trained independently on a random subset of features.

To train a single tree, we take a bootstrap sample of the training set, and construct the tree by recursively splitting the data at each node as described in Section III-A. The extracted features are a random subset of all possible Haar features and at each node, we choose the feature and associated optimal threshold which minimizes G_T to split the data.

2) Search: To detect the object in an image $\mathbf{I}$, we scan the image at a set of coarse angles and scales in a sliding window approach. The search is speeded up by evaluating only positions on a sparse grid rather than at every pixel. For every angle-scale combination of the bounding box at every position, we obtain the relevant feature values $\mathbf{f}$ and get the RF to make predictions $\mathbf{v} = (\delta_x, \delta_y, w)$ on the relative position of the true centre of the reference frame. Predictions are made using a single weighted vote per tree $w = (1/\sigma_x \sigma_y)$ where σ_x, σ_y are the standard deviations in x and y of the training samples that arrived at the particular leaf. The resulting response image $\mathbf{R}$ is then smoothed with a Gaussian of width 1.5 and searched for local maxima.

Once a response image has been obtained for every angle-scale combination, all maxima (across all angle-scale combinations) are ranked according to the sum of votes, S, at the predicted position. Every maximum is associated with an angle, a scale and a prediction of the reference frame centre. This gives reference frame predictions $\mathfrak{c} = \{\mathbf{c}_i\}$, where $\mathbf{c}_i = (x_i, y_i, s_i, \theta_i, S_i)$. Predictions $\mathbf{c}_i$ for the N_{fits} candidates with the highest sum of votes are then used to get the candidate positions of two reference points $\mathbf{r}_1$ and $\mathbf{r}_2$ of the object. These correspond to the two reference points that were used during training of the object detector. That is, given the position, orientation and scale of the reference frame candidate we know exactly where in the reference frame these two points are.

B. Object Segmentation

In [7] it is shown how RF regression voting produces useful response images for the CLM framework. Here, we summarize the key steps (see also Fig. 2).

1) Training: CLMs in their original form use normalized correlation as quality-of-fit measurement for each response image. In the RF regression approach, we train a regressor to predict the position of a landmark point based on a random set of Haar features. The quality-of-fit values here relate to the votes of the RF.

For each training image and every landmark l, we sample local patches at a number of random displacements $\mathbf{d}_l$ from the true position. Sample patches include displacements in x and y as well as random perturbations in scale and orientation. For every sample, we extract features $\mathbf{f}_l$ and train a RF on the pairs $\{(\mathbf{f}_l, \mathbf{d}_l)\}$. As with the global search, we train every tree taking a bootstrap sample and constructing it recursively by splitting the data at each node as described in Section III-A.

2) Search: To match the RF regression-based CLM to a new image, we run the search as follows: For every landmark l, we sparsely sample local patches in the area around an initial estimate of the landmark's position. We extract the relevant features for each sample and get the RF to make predictions on the true position of the landmark. Predictions are made using a single vote per tree. This yields a response image $\mathbf{R}_l$ for every landmark l. We then aim to combine voting peaks in the response images with the global constraints learned by the shape model via optimising (4). The procedure outlined in Section III-B only describes a single search iteration. The number of search iterations to be applied, N_{search}, depends on the searching range $[-d_{\text{search}}, +d_{\text{search}}]$ as well as the initialization of the model, i.e., how close to the true position the search is starting. In the experiments described below, N_{search} was preset (see also Section VI). Every new search iteration starts from updated landmark positions and hence is using response images $\mathbf{R}$ based on a different search area; the model is iteratively moving towards the contour of the object. If the model starts off very close to the object then a single search iteration might be sufficient. In other cases, though, the contour of the object may not even be part of the response images in the first few iterations.

For the first search iteration, the pose of the model is initialized with estimates from either a detector or from an earlier model. Every following search iteration is initialized with the results from the previous iteration.

C. Combined System

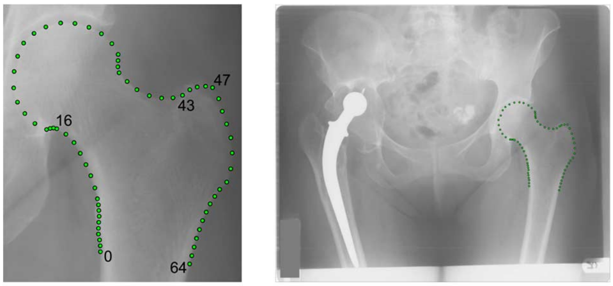
The fully automatic segmentation system first applies a global search at multiple scales and orientations (as in Section IV-A) to produce a number of candidate poses which are ranked by total votes. This is done without any assumptions about the pose of the object, allowing the technique to be universally applied. The candidate poses obtained from the object detector will then be used to initialize the local search to segment the contours of the object (as in Section IV-B).

The local search will be run from each of the N_{fits} candidate positions, where points $\mathbf{r}_1$ and $\mathbf{r}_2$ are used to initialize the two corresponding landmarks l^{r_1} and l^{r_2} of the local model. This yields N_{fits} segmentation results. Each result will be evaluated using a quality-of-fit measure that is defined by the number of votes at each landmark's position accumulated over all landmarks. The segmentation result with the best quality-of-fit measure, i.e., the best total CLM fit, will be the overall segmentation result of the fully automated system.

To speed up the segmentation process, candidate positions resulting from the global search can be clustered according to their proximity.

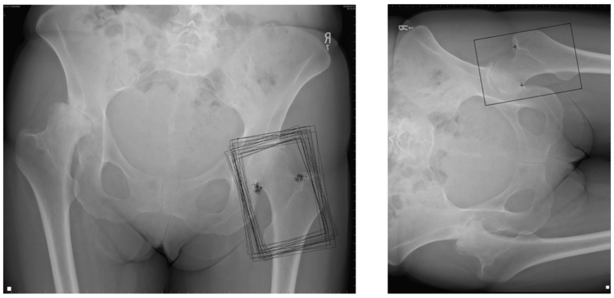
V. Data Set

Our data set comprises AP pelvic radiographs of 839 subjects (527 females and 312 males) suffering from unilateral hip osteoarthritis. All images were provided by The arcOGEN Consortium and were collected under relevant ethical approvals. The images have been collected from different radiographic centres resulting in varying resolution levels (555–4723 pixels wide) and large intensity differences caused by using different X-ray tubes and recording devices. In addition, the radiographs



(a) (b)

Fig. 3. Segmentation of the proximal femur: (a) 65 landmarks outlining the "front-view" femur (ground truth); (b) automatically segmented femur in AP pelvic radiograph.



(a) (b)

Fig. 4. Global search results of the *full detector*: (a) 20 best fits; (b) best fit for arbitrary pose of the femur.

are not guaranteed to have been taken under standardized conditions, and hence the displayed region of the pelvis as well as the pose of the femur in the images vary considerably. For example, in some images the proximal femur is at the top of the image, in others it is at the bottom or somewhere in-between. The data set also contains single side images where only half a pelvis is displayed and hence the proximal femur in these images is almost centred. For these reasons, to develop a fully automatic segmentation system that works for the whole data set, we cannot make any assumptions about the pose of the proximal femur in the image.

VI. Experiments and Evaluation

The aim is to fully automatically segment the femur by putting a dense annotation of 65 landmarks along its contour as demonstrated in Fig. 3 where (a) gives the manual annotation and (b) the result of the fully automated system. We use a *front-view femur* model that excludes the lesser trochanter as well as the greater trochanter and approximates the superior lateral edge (points 43–47) from an anterior perspective. All points were defined using anatomical features mixed with evenly spaced subsets.

In this work, we focused on annotating the left proximal femur. Images where the osteoarthritis unaffected side is the right one were mirrored accordingly. For each image, a dense reference annotation was obtained by manually placing 65 points along the contour of the proximal femur as illustrated in Fig. 3(a). All evaluations in the experiments below use this manual annotation as ground truth. To be able to both train with all images and to also test on all images, we performed two-fold cross-validation experiments. We randomly split the data set in half, trained on one half and tested on the other. We then repeated with the sets switched. Results reported below are the average of the two runs.

A. Global Search: Automatic Femur Detection

In the following, we will first describe the training of the object detector and will then analyze its performance with respect to finding the left proximal femur in AP pelvic radiographs.

1) Training: We set the reference frame so as to capture the proximal femur using points 16 and 43 [see Fig. 3(a)] to give its horizontal axis. In general, any two points would serve this purpose but points 16 and 43 were chosen specifically because *i*) they remain relatively constant with respect to each other allowing to capture about the same area of the hip joint across images, and *ii*) they allow a well placed initialization of the mean shape of the local search CLM providing maximal overlap of the mean shape and the true shape given good estimates of their position. To sample the patches, we used 20 random positions (within $\pm 50\%$ of the reference frame position) for 15 random scale (in the range of $\pm 15\%$) and angle (in the range of $\pm 15^\circ$) displacements. Including the true pose, this resulted in 336 samples for every training image, making the detector scale and pose invariant within that range. We trained two detectors: The *full detector* sampled the whole reference frame, and the *3ROI detector* sampled three regions of interest within the reference frame (shaft, femoral head, and greater trochanter). Both detectors used the exactly same reference frame and hence were trained to predict the same reference frame centre; the difference between both detectors being the area of the reference frame the features were sampled from. We trained a RF consisting of 10 trees for the full detector as well as for each region of the 3ROI detector, and used random subsets of size 500 when there were more than 500 samples to be processed at a node. The stopping criteria for node splitting were either a tree depth of 100 or less than five samples per node. We set up a third detector, the *combined detector*, by combining the outputs of both trained detectors; we found that obtaining a combined detector this way outperforms a detector that is trained on the combined patches.

2) Search: During search, we used the *full detector* and the *3ROI detector* to scan every test image at seven orientations ranging from -30° to 0° and at a range of scales such that the height of the bounding box is 30%–60% of the image height. Each of the two detectors provided the 20 best fits as shown in Fig. 4(a). Note that the restrictions in orientations and scales to be searched were for the sake of speed only. Fig. 4(b) shows an example where no restrictions were imposed to find femurs of arbitrary pose. Each of the fits gave the reference centre, angle and scale of a candidate reference frame. Because points 16 and 43 (see Fig. 3) have a fixed position within the reference frame, we can use the latter information to determine their positions for a given candidate reference frame. These positions can then be used to initialize the local search in the fully automated system. The *combined detector* contained 40 candidate positions for points 16 and 43, combining the candidates from each of the two trained detectors.

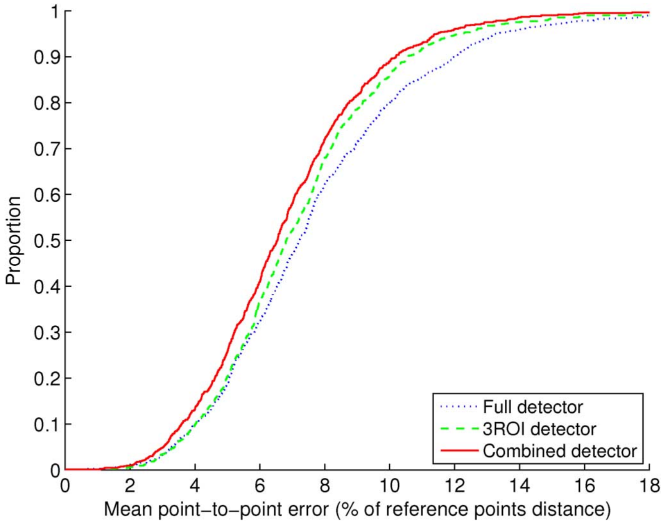


Fig. 5. Global search results for all three detectors, each using the cluster with the minimal mean point-to-point error.

The distance between points 16 and 43 defines a reference length that was used for quantitative evaluation of the global search. Our data set contains 15 calibrated images which we used to estimate this reference length in mm, suggesting an average length of 57 mm. However, this is just an approximation; the distance between points 16 and 43 varies depending on individual shape as well as on subject positioning during image acquisition.

To speed up the subsequent segmentation process, for every detector all candidates for landmarks 16 and 43 were clustered using a cluster radius, r_c, of 10% of the reference length, i.e., of the distance between the estimated positions of points 16 and 43 in the image.[1] We evaluate the mean point-to-point error as a percentage of the reference length, and give results for the *best* (minimal mean error) cluster only. Fig. 5 demonstrates the performance of all three detectors. This shows that the *combined detector* works best, which indicates that each of the trained detectors works particularly well for a different subset of images (depending on shape and appearance). By combining their best fits we get the best of both. When averaging over both reference points, the *combined detector* yields an error of less than 11.5% for 95% of all 839 images. This relates to a global search error of less than 6.6 mm for 95% of all images.

B. Local Search: Accurate Femur Segmentation

In the following, we will first describe the training of the local search models and will then analyze their performance with respect to segmenting the proximal femur when initializing the models with the mean shape at the correct pose.

1) Training: We found that excellent results can be achieved by following a coarse-to-fine two-stage approach for the local search. That is we trained two RF regression-based CLMs to be run in sequence: a *basis model* and a *refinement model*. Table I gives details on the training parameters for each of the models. For both the models, we trained a RF consisting of 10 trees for every landmark, and used random subsets of size 500 when there were more than 500 samples to be processed at the node. The stopping criteria for node splitting were either a tree depth of 500 or less than five samples per node.

TABLE I
PARAMETER SETTINGS FOR TRAINING THE LOCAL SEARCH MODELS. ALL VALUES REFER TO THE REFERENCE FRAME AND ARE GIVEN IN PIXELS IF NOT STATED OTHERWISE

	Basis model	*Refinement model*
Reference frame width*	200	500
Patch size (width x height)	20x20	20x20
Number of patches sampled	15	15
Range of random displacem. in x and y (d_{max})	20	15
Range of random perturbations in scale	5%	5%
Range of random perturbations in orientation	6°	6°
Range of search area displacements (d_{search})	±35	±15
Number of search steps (k)	9	7

*For both models, the reference frame captures the exactly same area, i.e., a higher frame width corresponds to a higher resolution of the area captured.

To compare the performance of the RF regression-based CLMs presented in this paper with alternative techniques, we trained a correlation-based CLM, a PDF-based CLM, an ASM [10], and an AAM [11]. The correlation-based CLM uses normalized correlation as quality-of-fit measurement for each response image, and the PDF-based CLM uses a PDF of the normalized intensities to measure the quality-of-fit. For both the CLMs, we used the same settings as for the *basis model* of the RF regression-based CLM where the local search area (as in Fig. 1) was optimized for best search performance. We did not apply the refinement step for the correlation-based CLM and the PDF-based CLM as experiments showed that this would decrease their segmentation accuracy. The settings used for the ASM and the AAM resulted from extensive experiments to select the parameters that show the highest segmentation accuracy. All models were trained to explain 95% of the shape variation given by the training set. We found that a 95% model generally performs best for searching the proximal femur. However, in the two-stage approach the *refinement model* explains 99% of shape variation to allow more freedom to fit to the shape given by the image; training the other models to explain more than 95% of shape variation yielded lower segmentation accuracy.

2) Search: We tested all models on 839 images and started searching from the mean shape at the correct pose running five search iterations. We applied four search iterations using the *basis model* and applied a single search iteration using the high resolution and high shape variation explaining *refinement model*. Experiments showed that applying a single refinement search iteration performs best. Any additional refinement search iteration would allow the model to drift off the correct pose. This is because the *refinement model* is not very shape restrictive (explaining 99% of shape variation) and only contains fairly local information (due to the high resolution and relatively small patch size). Table I gives details on the parameters used to optimize (4) for both the models.

Fig. 6 shows the mean point-to-curve error as a percentage of the shaft width. We define the latter as the distance between landmarks 0 and 64 (see Fig. 3). We used this as a reference

[1] The best fit defined the first pair of cluster centres. Candidate pairs were considered in order of their quality-of-fit, starting a new pair of clusters if any of the two candidate points was beyond r_c of any existing cluster pair.

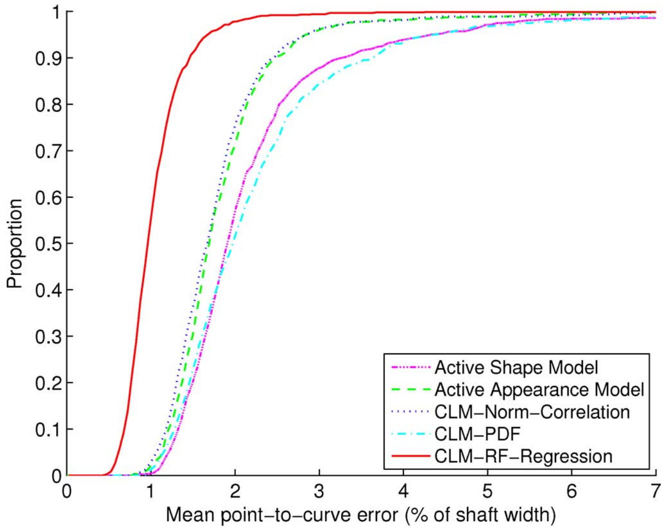


Fig. 6. Local search results starting search from mean shape at true pose.

length to evaluate segmentation performance as it tends to be relatively constant across individuals; our calibrated subset suggests an average length of 37 mm. Results show that the RF regression-based CLM performs best with a mean point-to-curve error of within 1.7% for 95% of all images, which relates to a local search accuracy of within 0.6 mm.

C. Full Search: Accurate Automatic Femur Segmentation

In this section, we will first provide details on how we set up the fully automated system before describing a series of experiments. We will start by looking at the overall performance of the system comparing the fully automatic search results to those obtained when initializing the local search models with the mean shape at the correct pose. Here, we will also analyze the performance improvement that can be achieved when following a two-stage approach for the local search. We will then compare the performance of the *basis model* suggested in this paper to the performance of the model presented in [9], pointing out the importance of patch size and search area to the performance of the system. Finally, we will investigate the effect gender may have on the segmentation performance of the system, aiming to identify whether gender-specific male and female models achieve higher accuracy.

For the fully automated system, we used the clustered candidates obtained via the global search to initialize the local search. Based on the results of our global search performance experiments as demonstrated in Fig. 5, we applied the *combined detector* and linked its outcome to a two-stage RF regression-based CLM with the settings as in Section VI-B. Every clustered candidate was used to predict the positions of points 16 and 43. This initialized the scale and pose of the *basis model* of the RF regression-based CLM. We tested all clustered candidates for every image and ran 20 search iterations using the *basis model* (starting from the initialized mean model) plus a single search iteration using the *refinement model*. We chose the candidate that gave the best final quality-of-fit value to give the fully automatic segmentation result.

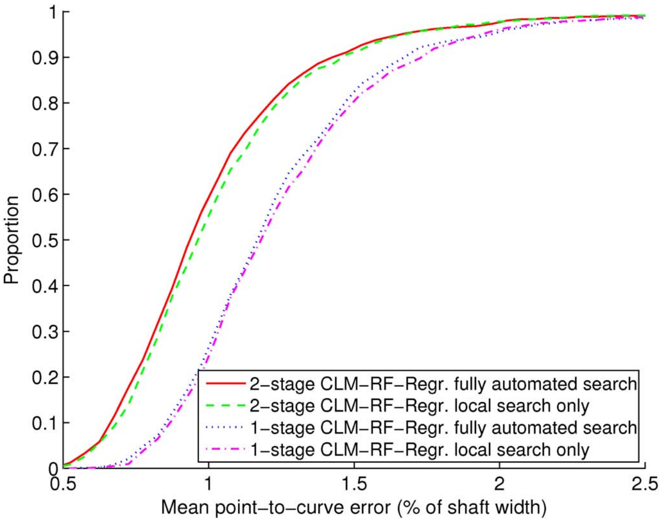


Fig. 7. Fully automated search results (showing results for the best clustered candidate) compared to local search results for the two-stage approach using the *basis model* followed by a single search iteration of the *refinement model* as well as for the one-stage approach using the *basis model* only.

Fig. 7 gives the mean point-to-curve error of the fully automated system (using either the basis model followed by a refinement step or the basis model only) compared to the local search performance (as in Fig. 6); results are given as a percentage of the shaft width. This shows that the global search works sufficiently well for the fully automated system to be very accurate with the two-stage approach achieving mean point-to-curve errors of less than 1.7% for 95% of all 839 images, relating to 0.6 mm. It also shows that the two-stage approach using the *basis model* followed by a single search iteration of the *refinement model* significantly outperforms the one-stage approach that uses the *basis model* only. The overlapping plots indicate that the fully automated system yields equally high accuracy as a local search starting from the mean shape at the correct pose.

The *basis model* of the two-stage approach described in Section VI-B is similar to the approach in [9]. However, we increased the patch size from 15 × 15 to 20 × 20 pixels and extended the local search area of the model, d_{search}, from 20 to 35 pixels. These adaptations to the model proved useful in order to overcome segmentation difficulties around the lesser trochanter. Fig. 8 demonstrates two cases where the model in [9] does not fit properly to the shaft. We found that these segmentation issues along the femoral shaft are linked to the size or visibility of the lesser trochanter. Fig. 8(a) and (b) represents cases where the lesser trochanter is very well visible and appears quite *big* (the automatic annotation tends to not stretch down enough), and Fig. 8(c) and (d) represents cases where the lesser trochanter is well hidden behind the shaft (the automatic annotation tends to stretch down too far).[2] Results show that the improved settings of the *basis model* helped to overcome these segmentation issues. The new settings work very well for cases

[2] The visibility of the lesser trochanter in AP pelvic radiographs depends on the internal or external rotation of the proximal femur during image acquisition. Hence, differences in radiographic lesser trochanter size are not necessarily due to differences in anatomical shape. However, in [20] we showed that the within-person variation is small compared to the overall shape variation of the proximal femur given by AP pelvic radiographs.

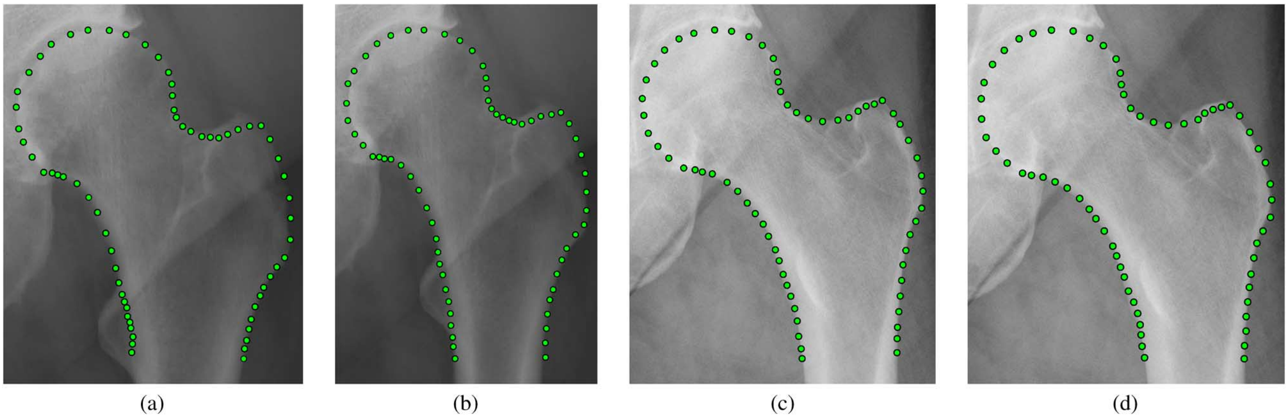


Fig. 8. Segmentation issues when the lesser trochanter (a), (b) is very well visible or (c), (d) hardly visible. (a) and (c) are obtained using the fully automated system presented in [9]. (b) and (d) use the *basis model* described in Section VI-B. (All searches were run on full pelvic images).

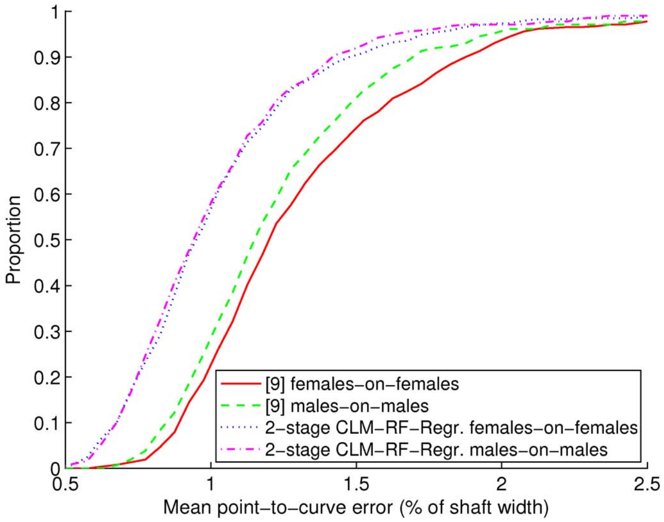


Fig. 9. Fully automated search results compared to previously best published results (showing results for the best clustered candidate).

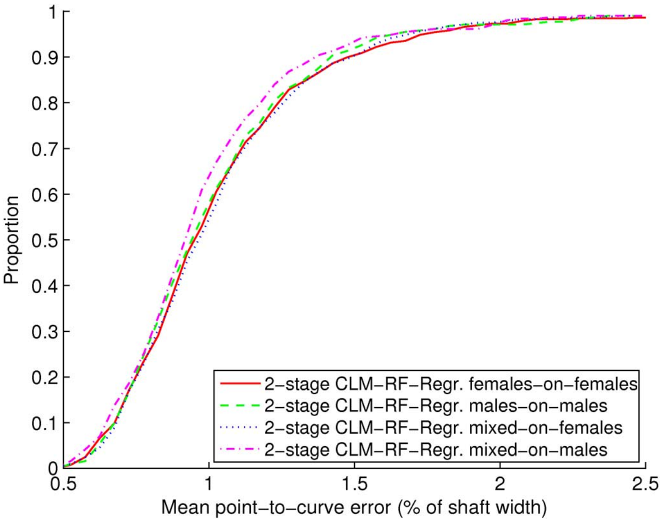


Fig. 10. Fully automated search results comparing gender-specific and mixed-gender local models (showing results for the best clustered candidate).

as in Fig. 8(b) but achieve only slight improvements for cases as in Fig. 8(d). Note that the results in Fig. 8(b) and (d) are based on 20 search iterations of the *basis model*; the *refinement model* was not applied to allow evaluation of the impact of the parameter settings of the *basis model* on the segmentation performance. Results can be further improved by applying a refinement search iteration.

Fig. 9 compares the results of the fully automated two-stage approach presented in this paper to the fully automated system in [9]. All error values are mean point-to-curve errors given as a percentage of the shaft width. Note that the difference between these two fully automated systems lies in the local search, the global search part is the same in both systems. Results show that the improved settings of the *basis model* in combination with a very high resolution refinement search iteration significantly outperforms previous results. In addition, we investigate the effect gender may have on the segmentation performance. For the results in Fig. 7, all cross-validation experiments were based on mixed-gender data sets. Fig. 9 gives the results of the fully automated system using a two-stage RF regression-based CLM with the settings as in Section VI-B trained and tested on cross-validation single-gender data sets. These results show that the fully automated system using the new two-stage model for the local search works equally well for pelvic images of males and females. The latter is not the case for the system in [9] where performance was significantly better on male images compared to female images. In addition to male images benefiting from better contrast due to increased bone density, the male images in our data set seem to be of slightly better quality than the female images. Moreover, shape analysis suggests that the anatomy of male and female proximal femurs differ to a certain extent.

The results in Fig. 9 suggest that following the two-stage approach for the local search a mixed-gender model may work equally well as gender-specific male and female models. We therefore trained gender-specific male and female cross-validation models, and compared their performance to the performance of the mixed-gender models presented above. All models were tested on gender-specific data sets. Fig. 10 gives the mean point-to-curve error of the fully automated system for female and male gender-specific as well as mixed-gender trained models. It can be seen that mixed-gender models and gender-specific models perform equally well.

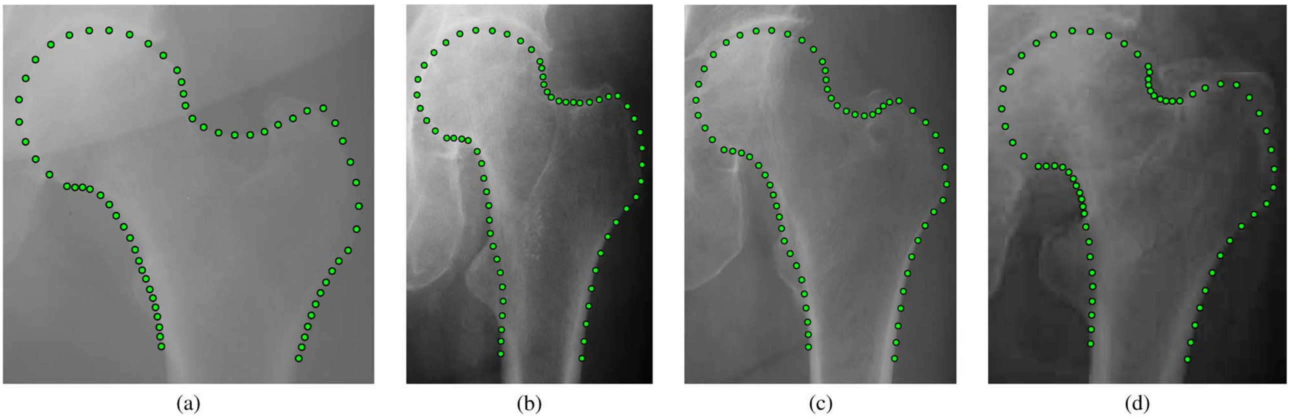


Fig. 11. Examples of segmentation results of the fully automated system (sorted by the mean point-to-curve percentiles): (a) median; (b) 91.2%, based on highest global search error; (c) 99.0%; (d) maximal overall error. (Due to space we only show the proximal femur; all searches were run on full pelvic images).

Note that running a mixed-gender model on the male data set performs slightly better. We assume this to be, on the one hand, due to the better quality of the male images and on the other hand, because the gender-specific male models were trained on 156/156 images whereas the mixed-gender models were trained on 419/420 images. The mixed-models may therefore contain additional shape information that allows them to perform better than the gender-specific male models.

Fig. 11 shows various segmentation results of the fully automated system, ranked according to mean point-to-curve percentiles: (a) gives the median result (50% of the images have a mean error of less than 0.4 mm); (b) is based on the highest global search error yielding a mean segmentation error of 0.6 mm; (c) shows the 99%ile achieving an accuracy of 0.9 mm; (d) gives the highest overall segmentation error corresponding to a mean error of 2.7 mm. These results show that the global search had a success rate of 100% in that it worked sufficiently well to initialize the local search.

A direct comparison to other reported automatic proximal femur segmentation results is difficult as most findings are either given qualitatively, or are not easy to interpret in more general terms. The best reported results appear to be those by Pilgram *et al.* [14] with a point-to-curve error of within 1.6 mm for 80% of the 117 test cases (estimated on the basis of a likely average shaft width of 209 pixels relative to the stated image width of 2320 pixels).

The fully automated segmentation system was developed using C++. All experiments were run on a 3.3-GHz Intel Core Duo PC in a VMware using 2 GB RAM. No parallel computing was implemented. The global search took on average 15 s per image, and the local search 10 s per image and cluster. For the results in Section VI-C, we searched on average ten clusters. Note that running times vary depending on image size and search settings.

VII. Discussion and Conclusion

We have presented a system to segment the proximal femur in AP pelvic radiographs which is fully automatic, does not make any assumptions about the femur pose, and is very accurate.[3] The segmentation system comprises two steps, femur detection and femur segmentation, both of which are based on a RF regression-based voting approach. For the segmentation step, we incorporated the latter into the CLM framework.

We have shown that the system achieves excellent performance in both its steps when tested on a data set of 839 images of mixed quality. The femur detector, achieving an accuracy of a mean point-to-point error of less than 8.4 mm for 99% of all images, works sufficiently well to initialize the local model used for segmentation. This is also reflected in the fully automated system yielding equally high accuracy as a local search initialized with the mean shape at correct pose, as seen by the overlapping plots in Fig. 7. In our experiments, the fully automatic segmentation system achieved an overall mean point-to-curve error of less than 0.9 mm for 99% of all 839 images. We believe that this is the most accurate fully automatic system for segmenting the proximal femur in AP pelvic radiographs so far reported.

In all the experiments above, the performance of the system depends on the random samples used to train the RFs. For different training sets, there will be a slight change in performance. However, our cross-validation experiments demonstrate that this difference is very small. For example, for the fully automatic system using a two-stage local search model (as in Fig. 7) the difference in medians between the two single experiments (i.e., when switching the randomly assigned training and test sets) is 0.005 mm.

In Appendix A, we discuss our choices of parameters and provide guidance to their setting.

A limiting factor for higher accuracy is the quality of the manual annotation. This is not only because the manual annotation is used to train the models but also because using the manual annotation as ground truth for performance evaluation can result in wrong error measurements. To estimate the intra-observer variability of the manual annotation, 20 images (10 females and 10 males) where reannotated after a period of 10 months. The difference in point-to-curve errors between the intra-observer variability and the fully automated system amounts to 0.1 mm for the medians and to 0.2 mm for the 95%iles. Inter-observer variability is likely to be larger.

[3] For access to the segmentation tool, please contact the authors.

We have examined the performance of mixed-gender and gender-specific models showing that mixed-gender models perform well on male and female data sets. We therefore conclude that there is no need to train gender-specific models. This is important when taking the segmentation system into practice as it means that radiographs would not need to be preselected. It also allows a *richer* model as it can be trained on a bigger mixed data set.

Overall the fully automated segmentation system is fast but for applications where running time is crucial efficiency could be improved by adjusting the number of search iterations as well as the number of clusters searched. This could be done in an adaptive manner based on some kind of reliable quality-of-fit value or using a learned threshold. Alternatively, running times may potentially be improved by following a Marginal Space Learning approach as in [13]. However, we do not expect to achieve accuracy improvements by using the latter as we have shown that the fully automated system performs equally well as a local search initialized with the mean shape at the true pose. The presented fully automated system is sufficiently general to be applied to other medical segmentation problems.

Appendix A
Choice of Parameters

The system is generally insensitive to most of the parameter settings. Applying the same system (without any change in parameter values) to the segmentation of the bones of the knee joint from AP radiographs showed similar good results as for the proximal femur achieving a mean point-to-curve error of within 0.7 mm for 95% of 500 images. This shows that the system generalizes well across application areas. However, in the following we will give some guidance on individual parameter settings for each the RF training, the global search and the local search.

A. Random Forest Training

Displacement range $d_{\max}$: Search results seem to be fairly robust to the choice of the displacement range. Choosing $d_{\max}$ within a half and a full size of the area captured (global search: bounding box size, local search: patch size) seems to be sufficient.

Range of scale and orientation displacements: These depend on the amount of variation in scale and orientation to be expected across the test data set.

Number of trees in the RF: We investigated the effect of the number of trees in the RF and found that the biggest improvement can be achieved when moving from a single tree to a few trees. Only minimal improvements can be achieved by increasing the number of trees beyond 5. We settled on 10 trees as it gives a slight improvement from five trees, but we did not find any additional number of trees worth the additional computation time.

Number of random Haar features used to train the trees: Search results seem to be fairly insensitive to this number. In all the experiments described above, we considered about 700 random Haar feature values for building a single tree.

B. Global Search

Number of candidates N_{fits}: The number of global search candidates to be considered for the local search depends on the performance of the global search which links back to how similar the test images are to the training images. We found that for most test images five candidates would be sufficient to achieve equally good results. However, this number might need to be increased to guarantee that the system works for all images.

Number of reference points to initialize the shape model: The number of reference points used to initialize the mean shape model of the local search depends on the complexity of the shape and the amount of shape variation across the data set. For a simple structure such as the femur or the knee joint that only varies modestly across individuals, two reference points are generally sufficient. However, for very complex structures more than two reference points may become necessary. The aim is to choose the reference points such that they well define the mean shape and maximize its overlap with the object when based on true point positions.

C. Local Search

Number of samples used for training: We found that a training data set of 150 images achieves comparable results to training data sets of 260 or 420 images. However, in all our experiments we adapted the number of samples taken per image to the overall number of training images available. To train a model based on 420 training images we use 135 samples (15 samples at each of three scales [95%, 100%, 105%] and three orientations $[-6^\circ, 0^\circ, +6^\circ]$) per image, and we increase/decrease this number accordingly if the number of training images is smaller/larger. In an initial set of experiments, we found that around 45 000 samples in total allow to achieve high accuracy with only slight accuracy improvements when using more.

Reference frame width: We chose the reference frame width of the *refinement model* to match the area captured (ROI) in our high resolution images such that for images where high resolution information is available no down sampling takes place. The reference frame width of the *basis model* is more arbitrary but we recommend it to be between a quarter and a half of the reference frame width of the *refinement model*. The aim is that when down sampling your images so that the reference frame width of the *basis model* corresponds to the width of the ROI the structures to be segmented should still be clearly visible.

Patch size: The patches need to be big enough to contain structural information that is relevant for the segmentation. The patches should cover a large enough area such that patches for different landmarks are distinguishable; if necessary making use of structural information that is not part of the model (e.g., in our case the trochanters and pelvis). This is particularly important for the *basis model* which may start searching from far off the true object contour. The patch size of the *refinement model* can be smaller (with respect to the particular reference frame width) as this model is expected to start searching from very close to the true answer—though the patches should be big enough to capture the object contour as well as some background. Note that the patch size influences the runtime and size of the model.

Search range d_{search}: For the *basis model*, the search range should be big enough to capture enough structural information to be able to guide the model towards the object contour. In our case, we chose $d_{\text{search}} = 35$ as this allows points along the shaft to include information about the lesser trochanter which for most of the shaft points is the only nearby structure that has a distinguishable relative position to each of the points. For the *refinement model*, a rather small search range (with respect to the particular reference frame width) is sufficient as this model is expected to start searching from very close to the true answer—the *refinement model* is meant to only refine the result and not to drift off the already found object contour.

Number of samples used to create response images $\mathbf{R}$: In [7] we have shown that it is sufficient to sample the search range on a sparse grid. Across our experiments, we found a step size of 3 (i.e., to only take one sample in each 3×3 square) to achieve good accuracy while significantly increasing runtime.

Optimization parameters for landmark updates: As described in Section III-B, we set the initial disk radius to match search range $d_{\text{search}}, r_{\min} = 0.4$ and the number of search steps k such that $d_{\text{search}} * 0.6^k \leq r_{\min}$. The crucial point is to set the initial disk radius to equal the search range. The search results seem to be fairly insensitive to the choice of $r_{\min}$.

References

[1] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[2] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu, "Regression forests for efficient anatomy detection and localization in CT studies," in *Medical Computer Vision Workshop*. New York: Springer, 2010, vol. 6533, Lecture Notes in Computer Science, pp. 106–117.

[3] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 415–422.

[4] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 54.1–54.10.

[5] J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1022–1029.

[6] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

[7] T. Cootes, M. Ionita, C. Lindner, and P. Sauer, "Robust and accurate shape model fitting using random forest regression voting," in *ECCV 2012*. New York: Springer, vol. 7578, Lecture Notes Computer Science, pp. 278–291.

[8] D. Cristinacce and T. Cootes, "Automatic feature localisation with constrained local models," *J. Pattern Recognit.*, vol. 41, no. 10, pp. 3054–3067, 2008.

[9] C. Lindner, S. Thiagarajah, and J. Wilkinson, arcOGEN ConsortiumG. Wallis and T. Cootes, "Accurate fully automatic femur segmentation in pelvic radiographs using regression voting," in Proc. MICCAI. New York, Springer, vol. 7512, Lecture Notes Computer Science, pp. 353–360, "Accurate fully automatic femur segmentation in pelvic radiographs using regression voting," in *Proc. MICCAI*, 2012.

[10] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models—Their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, 1995.

[11] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," in *ECCV*. New York: Springer, 1998, vol. 1407, Lecture Notes in Computer Science, pp. 484–498.

[12] G. Behiels, F. Maes, D. Vandermeulen, and P. Suetens, "Evaluation of image features and search strategies for segmentation of bone structures in radiographs using active shape models," *Med. Image Anal.*, vol. 6, no. 1, pp. 47–62, 2002.

[13] Y. Zheng, B. Georgescu, and D. Comaniciu, "Marginal space learning for efficient detection of 2-D/3-D anatomical structures in medical images," in *IPMI*. New York: Springer, 2009, vol. 5636, Lecture Notes in Computer Science, pp. 411–422.

[14] R. Pilgram, C. Walch, M. Blauth, W. Jaschke, R. Schubert, and V. Kuhn, "Knowledge-based femur detection in conventional radiographs of the pelvis," *Comput. Biol. Med.*, vol. 38, pp. 535–544, 2008.

[15] R. Smith, K. Najarian, and K. Ward, "A hierarchical method based on active shape models and directed Hough transform for segmentation of noisy biomedical images; application in segmentation of pelvic X-ray images," *BMC Med. Informat. Decision Making*, vol. 9, pp. S2–S12, 2009.

[16] F. Ding, W. Leow, and T. Howe, "Automatic segmentation of femur bones in anterior-posterior pelvis X-ray images," in *CAIP*. New York: Springer Verlag, 2007, vol. 4673, Lecture Notes in Computer Science, pp. 205–212.

[17] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[18] S. Schulter, C. Leistner, P. Roth, L. V. Gool, and H. Bischof, "On-line Hough forests," in *Proc. Br. Mach. Vis. Conf.*, 2011, pp. 128.1–128.11.

[19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2001, pp. 511–518.

[20] C. Lindner, S. Thiagarajah, and J. Wilkinson, arcOGEN ConsortiumG. Wallis and T. Cootes, "Short-term variability of proximal femur shape in anteroposterior pelvic radiographs," in *Proc. Med. Image Understand. Anal.*, 2011, pp. 69–73.