Automatic Location of Vertebrae on DXA Images Using Random Forest Regression*

M.G. Roberts, Timothy F. Cootes, and J.E. Adams

Imaging Science Research Group, University of Manchester, U.K.
martin.roberts@manchester.ac.uk

Abstract. We provide a fully automatic method of segmenting vertebrae in DXA images. This is of clinical relevance to the diagnosis of osteoporosis by vertebral fracture, and to grading fractures in clinical trials. In order to locate the vertebrae we train detectors for the upper and lower vertebral endplates. Each detector uses random forest regressor voting applied to Haar-like input features. The regressors are applied at a grid of points across the image, and each tree votes for an endplate centre position. Modes in the smoothed vote image are endplate candidates, some of which are the neighbouring vertebrae of the one sought. The ambiguity is resolved by applying geometric constraints to the connections between vertebrae, although there can be some ambiguity about where the sequence starts (e.g. is the lowest vertebra L4 or L5, Fig 2a). The endplate centres are used to initialise a final phase of Active Appearance Model search for a detailed solution. The method is applied to a dataset of 320 DXA images. Accuracy is comparable to manually initialised AAM segmentation in 91% of images, but multiple grade 3 fractures can cause some edge confusion in severely osteoporotic cases.

1 Introduction

The accurate identification of vertebral fractures is clinically important in the diagnosis of osteoporosis. Typically diagnosis uses a semi-quantitative approach using subjective judgement by a radiologist. Quantitative morphometric methods are not specific and require tedious manual annotation of six points on each vertebra. See [1] for a discusson of diagnosis methods. More sophisticated classification methods based on statistical models have been reported in [1,2], but these require an accurate segmentation method. Active appearance models [3] (AAM) have been used to segment dual energy X-ray absorptiometry (DXA) images in [4], but the method required a manual initialisation on the centre of each vertebra. In applications such as clinical drug trials, it is desirable to eliminate the manual initialisation.

This paper describes a three-phase approach. First we locate putative vertebral endplates using a set of random forest regressors (one per endplate), together with Hough-style tree voting. Modes in the vote image of each vertebral endplate are candidate endplate positions. Secondly, the ambiguity is resolved by

* We are thankful to Arthritis Research UK for funding.

N. Ayache et al. (Eds.): MICCAI 2012, Part III, LNCS 7512, pp. 361–368, 2012.

applying a graphical model of the connections between vertebrae, thus applying geometric constraints. The solution of the graph problem is used to initialise a third phase of AAM search. The novelty of the work lies in providing a fully automatic segmentation method for vertebrae, although there still remains an inevitable ambiguity in the starting level (e.g. is the bottom vertebra detected in Fig 2a L4 or L5), which even experienced clinicians can find difficult to resolve.

1.1 Data

The dataset used consists of 320 DXA Vertebral Fracture Assessment (VFA) images scanned on various Hologic (Bedford MA) scanners, obtained from: a) 44 patients from a previous study [5]; b) 80 female subjects in a epidemiological study of a UK cohort born in 1946; c) 196 females attending a local clinic for DXA BMD measurement, and for whom the referring physician had also requested VFA (as approved by the local ethics committee). The lumbar vertebrae from L4-L1 and the thoracic vertebrae from T12-T7 (Fig 2c) were annotated with detailed point positions (42 points per vertebra) for training AAMs.

2 Methods

2.1 Regression Forests

Background on Regression Trees. Regression trees [6] are an efficient way of predicting continuous output vectors given a complex set of input features. At each branch in the tree the data is split into two subsets based on a threshold on a selected feature using some criterion that seeks to increase the homogeneity of the output vectors in the child branches. A regression forest consists of multiple trees with some degree of randomisation, for example each tree is trained on a bootstrapped subset; and at each branch a random subset of the input features are considered. After training, a new data point can be predicted by dropping the input vector down the trees. Each tree produces a prediction, which is typically the mean of the training outputs at the terminal node. The predictions of each tree in the forest are then aggregated, typically by taking the mean over all trees; or in [7] the distributions of each leaf node are used; or Hough style voting can be used [8]. Unlike Hough forests [8] we do not train object classification; each of our trees cast one vote; and we include votes cast by patches which may be displaced completely outside the object.

Pre-processing and Training. The images are smoothed with a Gaussian filter with σ =1mm for L4-L1 and then reduced according to mean vertebral size in the thoracic spine. The image is locally normalised using mean and variance derived from a sliding exponential filter of standard deviation 25mm. Patches are then sub-sampled with a step size of 1.5σ based on the appropriate endplate centre and aligned to the vertebral axis. For the lumbar vertebrae (L4-L1) the patch extends from the centre of the endplate 18 mm inside the vertebra and

12mm outside, whilst in width it extends to the mean semi-width across the training set, plus a left distance of 18mm and 12mm to the right. The left (posterior) bias is to include further context information from the spine. These distances are downscaled for the smaller thoracic vertebrae. Similar patches are also sampled by randomly displacing the centre up to 30mm in x, 20mm in y (with an x-axis aligned to the true vertebral axis), and randomly rotating by up to 20°. The (axis-relative) displacements in x, y and orientation are stored after normalising.

Input Features and Training. The input features used are all possible Haar-like features [9]. In [7] the features used were the difference in mean intentsity values between two randomly displaced boxes. We also experimented with these, but found that the Haar features performed slightly more accurately and reliably. We believe that broadly similar results can be obtained from random box-comparison features, but the Haar scheme includes more complex comparisons for highlighting ridges and corners.

In order to randomise between trees, and reduce the large Haar set, at each tree branch we pick a random subset of features (mean size 1000). We generated 100 perturbations for each patch in each image and trained the random forest regressors using leave-20-out cross-validation. The splitting criteria used was the total variance (weighted by sub-sample size) summed over the output variables. The selected splitting feature and threshold are those that minimise the weighted total variance, subject to an imposed minimum node sample size of 7. A branch was terminated if the node variance reached a minimum variance of 1% of the total initial variance, or at a depth of 18. We used a forest size of 40 trees. The output is the 3-dimensional vector giving the displacements in x, y (axis-relative) and rotation.

Voting Scheme. Endplate centre candidates are located by sampling the Haar features across a grid of points separated by 4mm in x and 2mm in y, and at a set of 5 orientations $\{\alpha_r^{(t)}\}$, given by the mean vertebral axis and displacements between $-20°$ and $+20°$ at 10° intervals. The grid is sampled over the 3 SD range of positions for that vertebra in the training set together with a 10% bounding region. We initially maintain separate voting accumulators for each of the 5 starting orientations. Each grid position casts votes from each tree in the forest as follows. For tree i starting at grid position $\mathbf{x}_j$ for endplate r, the angular displacement prediction $\hat{\theta}_{ijr}$ is used to update the vertebral axis $\hat{\alpha}_{ijr} = \alpha_r^{(t)} + \hat{\theta}_{ijr}$; then after applying the counter-rotation $\mathbf{R}^T(\hat{\alpha}_{ijr})$ from the tree's axis-relative frame, we obtain the predicted location $\hat{\mathbf{x}}_{ijr} = \mathbf{x}_j + \mathbf{R}^T [\Delta x_{ijr}, \Delta y_{ijr}]^T$. This update scheme also using orientation prediction is more complex than simply predicting $\Delta x, \Delta y$ in a world frame, but since the features are defined relative to a vertebral axis, it should be more accurate; it also allows us to introduce orientation weighting (see below) in the voting scheme.

The predicted location $\hat{\mathbf{x}}_{ijr}$ is used to accumulate a vote array in the neighbourhood of the rounded position in an array of 1x1mm^2 bins, using Gaussian

kernel smoothing with $\sigma = 0.5$ (half a bin). The vote portion accumulated in bin position $\mathbf{b}_{lm}$ is $w_{lm}/\sum_{(l,m)\in N} w_{lm}$ with N the neighbourhood such that $|\mathbf{b}_{lm} - \hat{\mathbf{x}}_{ijr}| \leq 2$, and $w_{lm} = \exp(-|\mathbf{b}_{lm} - \hat{\mathbf{x}}_{ijr}|^2/2\sigma^2)$.

The vote of each tree is further weighted by angular displacement $w_a(\hat{\theta}_{ijr})$, as we expect better accuracy for starting orientations closely aligned to the real orientation; where

$$w_a(\hat{\theta}_{ijr}) = 1 \quad \left|\hat{\theta}_{ijr}\right| <= \theta_0; \qquad w_a(\hat{\theta}_{ijr}) = \exp\Big(-\frac{|\hat{\theta}_{ijr}|-\theta_0}{\sigma_\theta}\Big) \quad \left|\hat{\theta}_{ijr}\right| > \theta_0$$

We use $\theta_0 = 5^\circ$ and $\sigma_\theta = 10^\circ$. The total vote for each accumulator at each point V_{lmr} is obtained by summing across all starting grid points and trees so $V_{lmr} = \sum_i \sum_j w_a(\hat{\theta}_{ijr}) w_{lm}(\hat{\mathbf{x}}_{ijr})$. Each vote accumulator is then treated as a 2D image and further smoothed with SD 1.5mm. All local modes in this smoothed vote image $\breve{\mathbf{V}}$ are located, and the top 20 modes from each orientation are passed on for clustering.

The clustering algorithm forms the minimum spanning tree of all modes pooled from all orientations, and then deletes all arcs of length exceeding 2mm. The remaining connected sub-graphs form clusters and the feature location is taken as the highest scoring mode in the cluster. The vote score $\breve{V}_{kr}$ of mode k is post-processed onto a (0,1) scale to form a quasi-probability $\breve{p}_{kr}$, using a sigmoidal transform parameterised using the inter-quartile range of the successfully located modes in the training set. This transform can be viewed as a biased version of the logistic function commonly used in classification [1], or as an approximation to the CDF of the successful mode vote. The top 10 modes for each endplate location give a candidate set of scores $\{\breve{\mathbf{p}}_r\}$ and positions $\{\breve{\mathbf{x}}_r\}$.

2.2 Inter-Feature Geometric Constraints

Typically the correct endplate position is somewhere in the list of modes, but similar responses are encountered at neighbouring vertebrae (Fig 2a), and the lower endplate detectors can locate neighbouring upper endplates or vice versa (Fig 2b). To resolve the ambiguity, we use a geometric model containing multiple nodes (one per endplate), together with a model of the pairwise geometrical

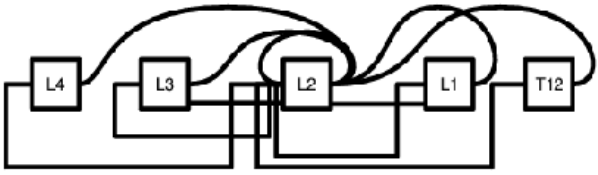

Fig. 1. The arcs connecting L2 endplates to neighbours. Similar connections exist for other vertebrae.

[1] Note that the graph solution (see below) is invariant to constant bias.

relationships between them as in [10]. We can then use graph algorithms to locate the optimal solution for the combination of feature response and geometry. We connect the lower endplate to the upper endplate, which is also connected to the lower endplate immediately above it. This chain can be solved by dynamic programming. However a more complex graph is needed to allow for missing detection cases, and also additional constraints can help resolve ambiguities. So arcs also join each upper/lower endplate to the corresponding endplates of vertebrae above and below including two vertebrae distant (Fig 1). The arc costs are given by the Mahalanobis distance D_{rq} of each pair in the edge set E after aligning into the target vertebral axis frame. The final selection of optimal solution $\tilde{\mathbf{k}}$ from the set of candidate modes $\{\check{\mathbf{x}}_{rk}\}$ is given by finding the minimum sum of node and arc costs over the R endplates:

$$\tilde{\mathbf{k}} = \text{argmin}_{\mathbf{k}} \Big\{ \sum_{r=1}^{R} -\log(\check{p}_r(k_r)) + \lambda \sum_{(r,q)\in E} D_{rq}(k_r, k_q) \Big\} \tag{1}$$

The graph is not a tree and cannot be solved by dynamic programming, but we use loopy belief propagation (LBP) instead [11] [2]. The parameter λ controls the relative weighting on the spatial constraints; we used $\lambda = 0.1$, based on some provisional experiments with related random forest classifiers (i.e. is a box centred on the endplate or not). We found the detection of T8/T7 to be somewhat unreliable (may be obscured by the scapulae, Fig 2d), and so solve the graph problem from L4-T9, and leave T8/T7 to the next phase of AAM fitting.

2.3 Active Appearance Models

We train AAMs [3] for overlapping triplets of vertebrae similar to [4] covering L4-T7. We initialise a global shape model using the endplate centres using a robust M-estimator to allow for some detection failures. Then a set of individual AAMs are initialised for each triplet covering L4 to T9. Initially we concentrate on the more reliable L4-T11 section, and at each iteration perform a tentative fit to all remaining triplets, and then pick the best one (lowest residual sum of squares in AAM texture model) to impose. This affects the re-initialisation of the neighbours. After fitting L4-T11, the remaining triplets are fitted by moving up the spine, and after each triplet fit a global shape model is updated to predict the positions of the remaining vertebrae used in their AAM initialisation.

2.4 Experiments

We tested the algorithms using leave-20-out cross-validation on the 320 images, and calculated point-to-line errors against the gold standard manual annotation. There is a fundamental ambiguity in determining the vertebral levels. The lowest

[2] The max-product variant, equivalent to max-sum with log probabilities.

visible vertebra may be L4 or L5 (even L3) (Fig 2a); false positives can be detected on the sacrum; or there can even be an L6. The optimum graph solution can correspond to the correct solution shifted up or down by one vertebra (occasionally even two). We have not yet addressed how to resolve this ambiguity. In order to produce meaningful overall accuracy statistics we examine the top solution, and other solutions after removing the lowest nodes in the former, up to 5 possible solutions. If any of these identify the vertebral centres L4-T12 to within 4mm in Y and 6mm in X we consider it a success, and proceed with the highest ranked such solution to the AAM stage.

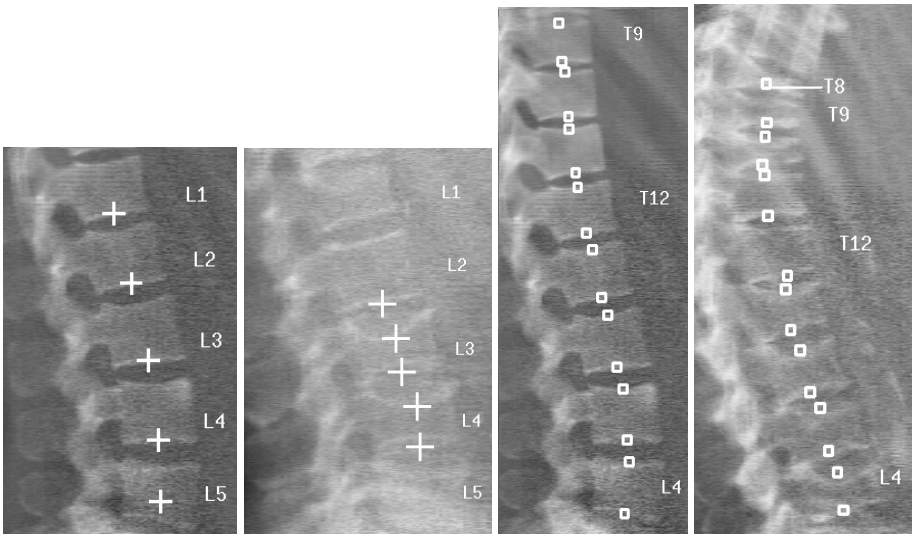

Fig. 2. Detected positions for bottom of L2 and overall LBP graph solutions used to initialise AAM. a) Top 5 L2 (bottom) regressor mode positions on bottoms of L5-L1; b) Similar regressor modes on bottoms of L2-L4 plus confounders at the tops of L3, L4; note L1,L3,L4 fractures; c) Good image with accurate LBP solution for all vertebrae; d) Severely osteoporotic case - most fractures are located by the LBP solution but the top of T9 is mis-located on the bottom of the extremely fractured T8 (indicated).

3 Results

Figure 2 (a-b) shows two examples of the top 5 located positions for the L2 lower endplate. These typically include responses from neighbours, but there are responses on the upper endplates which can produce some confusing shifts. Figure 2 (c-d) also shows examples of the solution for L4-T9 used to initialise the AAM. A successful initialisation was obtained in 308 images out of 320 (96.2%). The failures were mostly severely osteoporotic cases with many fractures (hence unusual geometries), or severe disc disease fusing vertebrae. In the 308 successful

Table 1. Search error statistics (point-to-line) for AAM by vertebra fracture status. Bracketed figures are after excluding the 15 images (5%) where 3 or more vertebrae suffered edge confusion with neighbours.

Vertebra Status	%ge of Sample	Automatic Initialisation Search Error Statistic			Manual Initialisation Search Error Statistic		
		Mean (mm)	Median (mm)	%ge errors >2mm	Mean (mm)	Median (mm)	%ge errors >2mm
Normal	84.9%	0.66(0.55)	0.41(0.40)	3.6(2.5)%	0.55(0.55)	0.40(0.40)	2.5(2.3)%
Grade 1	5.9%	0.92(0.75)	0.52(0.50)	7.7(5.4)%	0.70(0.70)	0.49(0.49)	4.8(4.8)%)
Grade 2	5.1%	1.12(0.88)	0.61(0.58)	11.4(9.3)%	0.92(0.88)	0.61(0.59)	10.2(9.3%)
Grade 3	4.1%	1.94(1.18)	0.80(0.67)	23.9(15.0)%	1.19(1.07)	0.72(0.68)	16.5(14.0)%

images, 60% have the best solution at the correct vertebral level, with 27% and 13% shifted up or down by one vertebra respectively, and a single case is shifted by two.

Table 1 gives point-to-line accuracy results for these 308 successful cases, together with corresponding figures for a manually initialised AAM search. In these 308 cases the overall mean segmentation error was 0.74mm, increasing with fracture grade. After running the AAM there were 15 images (5%) with edge confusions on 3 or more vertebrae in succession - typically caused by successive severe fractures (such as Fig 2d). Removing these images from the statistics substantially reduces the mean error, which is skewed by the larger error tail on these partial failures. The errors on the images without these substantial edge confusions are given in brackets in table 1, which also shows errors from manually initialised AAM fits for comparison. If the 15 edge confusion images are also considered failures, then the overall success rate is 91.2%, and in these cases the accuracy is comparable to that obtained using a manual initialisation (overall mean 0.59mm vs 0.58mm manual).

4 Discussion and Conclusions

Although there is some failure of the automatic initialisation process, the algorithm sucessfully locates a plausible set of vertebrae in over 91% of cases, with most failures on extremely osteoporotic cases. The fundamental ambiguity of vertebral levels is still an unsolved problem, but in some triage applications (e.g. detect any patient with possible fracture) this may not matter; or the user can be presented with the best solution plus two shifted versions to choose from. Good accuracy is obtained for normal (unfractured) vertebrae, but there are larger errors for severely fractured vertebrae, due to edge confusions between a vertebra and its neighbours. The overall mean error of 0.74mm compares well to other methods (e.g. [12], mean error 1.4mm, or [13], mean errors 1.22 or 1.34mm). Our initialisation failure of 3.8% on L4-T9 appears comparable to the 2% on L4-L1 for the baseline data in [13], when scaled by the number of vertebrae. The final failure rate of 8.8% appears higher, but [13] deals only with the lumbar, whereas

our additional failures are due to multiple severe fractures in the thoracic spine. The thoracic vertebrae are harder to segment because the vertebrae are closer together (especially when affected by disc disease), resulting in more edge confusions between neighbouring vertebrae; and there is more overlaying structure from the ribs and scapulae. If the algorithm were being used in triage to pick up patients with any fractures, then even the additional 15 image "failures" would still result in successful patient referrals, as all are cases like Fig 2d with some fractures being successfully located before edge confusion occurs.

References

1. Roberts, M.G., Cootes, T.F., Pacheco, E.M., Adams, J.E.: Quantitative vertebral fracture detection on DXA images using shape and appearance models. Academic Radiology 14, 1166–1178 (2007)
2. de Bruijne, M., Lund, M., Tanko, L., Pettersen, P., Nielsen, M.: Quantitative vertebral morphometry using neighbour-conditional shape models. Med. Image Anal. 11, 503–512 (2007)
3. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 681–685 (2001)
4. Roberts, M.G., Cootes, T.F., Adams, J.E.: Vertebral morphometry: semi-automatic determination of detailed shape from DXA images using active appearance models. Investigative Radiology 41(12), 849–859 (2006)
5. McCloskey, E.V., et al.: Effects of clodronate on vertebral fracture risk in osteoporosis: a 1-year interim analysis. Bone 28(3), 310–315 (2001)
6. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
7. Criminisi, A., Shotton, J., Robertson, D., Konukoglu, E.: Regression Forests for Efficient Anatomy Detection and Localization in CT Studies. In: Menze, B., Langs, G., Tu, Z., Criminisi, A. (eds.) MICCAI 2010 Workshop MCV. LNCS, vol. 6533, pp. 106–117. Springer, Heidelberg (2011)
8. Gall, J., Lempitsky, V.: Class-specific hough forests for object detection. In: Proc of CVPR 2009, pp. 1022–1029. IEEE Computer Society (2009)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc of CVPR 2001, pp. 511–518. IEEE Computer Society (2001)
10. Donner, R., Micusik, B., Langs, G., Bischof, H.: Sparse MRF appearance models for fast anatomical structure localisation. In: Rajpoot, N., Bhalerao, A. (eds.) Proc. of BMVC 2007, pp. 1080–1089. BMVA (2007)
11. Weiss, Y., Freeman, W.: On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. IEEE Trans. Inf. Theory 47, 736–744 (2001)
12. de Bruijne, M., Nielsen, M.: Image segmentation by shape particle filtering. In: Proc. of ICPR 2004, pp. 722–725. IEEE Computer Society (2004)
13. Petersen, K., Ganz, M., Mysling, P., Nielsen, M., Lillemark, L., Crimi, A., Brandt, S.: A bayesian framework for automated cardiovascular risk scoring on standard lumbar radiographs. IEEE Trans. Med. Imag. 31(3), 663–676 (2012)