to be published at SSTD 2011 -- the original paper will be made available at www.springerlink.com

Region of Interest Queries in CT Scans

Alexander Cavallaro[2], Franz Graf[1], Hans-Peter Kriegel[1], Matthias Schubert[1⋆], and Marisa Thoma[1⋆]

[1] Institute for Informatics, Ludwig-Maximilians-Universität München, Oettingenstr. 67, D-80538 München, Germany, {graf,kriegel,schubert,thoma}@dbs.ifi.lmu.de,
[2] Imaging Science Institute, University Hospital Erlangen, Maximiliansplatz 1, D-91054 Erlangen, Germany, Alexander.Cavallaro@uk-erlangen.de

Abstract. Medical image repositories contain very large amounts of computer tomography (CT) scans. When querying a particular CT scan, the user is often not interested in the complete scan but in a certain region of interest (ROI). Unfortunately, specifying the ROI in terms of scan coordinates is usually not an option because an ROI is usually specified w.r.t. the scan content, e.g. an example region in another scan. Thus, the system usually retrieves the complete scan and the user has to navigate to the ROI manually. In addition to the time to navigate, there is a large overhead for loading and transferring the irrelevant parts of the scan.
In this paper, we propose a method for answering ROI queries which are specified by an example ROI in another scan. An important feature of our new approach is that it is not necessary to annotate the query or the result scan before query processing. Since our method is based on image similarity, it is very flexible w.r.t. the size and the position of the scanned region. To answer ROI queries, our new method employs instance-based regression in combination with interpolation techniques for mapping the slices of a scan to a height model of the human body. Furthermore, we propose an efficient search algorithm on the result scan for retrieving the ROI with high accuracy. In the experimental evaluation, we examine the prediction accuracy and the saved I/O costs of our new method on a repository of 2 526 CT scans.

1 Introduction

Radiology centers all over the world currently collect large amounts of 3D body images being generated by various scanners like PET-CT, MRT, x-ray or sonography. Each of these methods generates a three dimensional image of the human body by transforming the echo of a different type of signal allowing a radiologist to examine the inner parts of a human body. In the following, we will particularly focus on CT body scans. However, the methods proposed in this paper are generally applicable to other types of scans as well.

⋆ corresponding authors

Technically, the result of a CT scan is stored as a stack of 2D images representing 3D slices of the human body, i.e. each slice is considered to have a certain thickness. The scans in a radiology center are stored in a centralized picture archiving and communication system (PACS) and they are transferred via LAN to the workstation of a physician. In commercial PACS, querying CT scans is currently restricted to retrieving complete scans being annotated with certain meta information like patient name, date and type of the examination. Therefore, each time a CT scan is queried, the complete scan, potentially comprising several thousand high-resolution images, has to be loaded from the image repository. For example, the data volume of a thorax scan being generated by a modern scanner comprises around 1 GB of data. Considering that several physicians will simultaneously query a PACS, the loading time of a single CT scan is up to several minutes depending on network and server traffic.

However, in many cases it is not necessary to display the complete scan. For example, if a physician wants to see whether a certain liver lesion has improved between two scans, the user primarily requires the portion of both scans containing the liver. Therefore, the physician loses up to several minutes by loading unnecessary information and searching for the liver within both scans. Thus, a system retrieving the parts of both scans containing the liver, would save valuable time and network bandwidth.

Parts of a CT scan can be efficiently loaded by raster databases [2] as long as the coordinates of the ROI are specified. However, in the given context, the ROI is rather defined by the image content. In other words, the coordinates of organs and other anatomical regions may strongly vary because of differences in the patients' heights or in the scanned body region. Thus, raster coordinates cannot be used to align to CT scans w.r.t. the image content.

In this paper, we focus on a query-by-example setting. Therefore, the query is posed by selecting a certain body region in a scan. The result contains the part of the scan showing the corresponding body region in one or multiple result scans. For example, a radiologist could select a certain area in the scan being currently under examination. He or she might want to see the corresponding regions in scans of patients having the same disease or earlier examinations of the same patient.

The most established approach to answer this type of queries is based on landmark detection. [14] A landmark is an anatomically unique location in the human body which is well-detectable by pattern recognition methods. To use landmarks for query processing, it is first of all necessary to detect as many landmarks as possible in the example scan and all result scans. Let us note that landmark detection employs pattern recognition methods and thus, there is a classification error, i.e. some of the predicted landmark positions are error prone. Furthermore, it can happen that some of the landmarks are not detectable due to disturbances while recording the scan. However, having detected a sufficiently large number of landmarks, it is possible to align both scans and afterwards select the area from the target scan corresponding to the query.

An important aspect of this approach is that landmark detection should be done as a preprocessing step. Thus, the example scan and the target scans need to be annotated with the landmark position to allow efficient query processing. However, this causes a problem when allowing example scans not being stored in the same PACS. In this case, the query might not have any landmarks or it is not labelled with the same set of landmarks. If the example scan and the result scan are taken by CT scanners from different companies, the positioning systems might not be compatible. Another drawback of the landmark approach is the size of the scan. CT scans are often recorded for only a small part of the body. Thus, it cannot be guaranteed that the scanned body region contains a sufficiently large set of alignable landmarks. To conclude, a fixed and comparably small set of landmarks is often not flexible enough to align arbitrary scans.

In this paper, we propose a more flexible approach being based on similarity search on the particular slices of a CT scan. Our new method does not rely on any time-consuming preproccesing step, but it can be directly applied on any query and result scan. Whereas landmark-based approaches can only align scans with respect to a limited amount of fixed points to be matched, our new approach can generate the positions in the scan to be matched on the fly. Thus, we can even align scans being labelled with different types of landmarks or scans not having any detectable landmarks at all.

The key idea behind our method is to map single slices of a CT scan to a generalized height model describing the relative distances between concepts w.r.t. the height axis of the human body. The height model is independent of the individual size and proportions of a particular patient. Let us note that it is possible to use width and depth axes as well. However, the height axis is the predominantly used navigation axis for CT scans.

By mapping single slices to the model, we can better adjust to limited information about the scan and we are independent from the distribution of predefined landmark positions. Our prediction algorithm employs instance-based regression based on Relevant Component Analysis [1] and the X-Tree [3] for efficiently answering kNN queries.

ROI queries are answered as follows: In the first step, the user selects a certain region of interest in the example scan. Afterwards, we employ instance-based regression to determine the query position in the generalized height model. In the next step, we need to determine the part in each target scan corresponding to the query interval in the height model. Let us note that this second step is more complicated, since we cannot directly determine the slice belonging to a particular height value. One solution to this problem would be to label all available slices with the height value in the model. However, labelling all DICOM images in an average PACS would cause an enormous overhead in preprocessing. Since the majority of images will never be involved in answering an ROI query, we follow a different strategy. Instead of preprocessing each image in the PACS, determining height values for a given slice is done on the fly. To make this type of processing efficient, we propose a query algorithm that alternates regression and interpolation steps until the queried ROI is found in the result scan.

Let us note that although the solutions proposed in this paper are very problem-oriented, the solution principle can be extended to other data as well. For example, a similar processing scheme can be applied to video streams (e.g. procedure timing in surveillance videos) or text mining (e.g. news tickers, twitter streams, age classification in Internet forums).

The rest of the paper is organized as follows. Sect. 2 surveys methods that are related to our approach or parts of it. In Sect. 3, we formalize ROI queries and give an overview of our system. Afterwards, Sect. 4 introduces our method for predicting height values for particular CT slices. Sect. 5 first describes interpolation methods for aligning CT scans to a generalized height model and then presents our new query algorithm. The results of our experimental evaluation are shown in Sect. 6. The paper concludes with a brief summary and ideas for future work in Sect. 7.

2 Related Work

In medical imaging, there are various localization or registration approaches. Most of them are very domain specific, like the Talairach space brain atlas [15] or the MNI space [5]. Nevertheless, as these atlases are very specific to their domain, they were not designed to cover the entire body and they can thus hardly be used for general ROI queries.

Position mapping via landmark-detector-based approaches like the Theseus Medico system presented in [14] are more appropriate for our purpose. This prototype provides an image parsing system which automatically detects 22 anatomically relevant landmarks, i.e. invariant points, and 9 organs. [13] It is thus possible to query the database directly for ROIs which are equivalent to these automatically-annotated image regions. However, general queries for arbitrarily defined ROIs are not yet supported.

A more general, landmark-based interpolation approach for mapping a volume into a standardized height space has been proposed by [7]. However, it is very patient-specific and dependent on the used landmarks. Another approach that uses partial volumes as query is described in [6]. It localizes query volumes with sizes ranging from 4 cm to more than 20 cm by comparing the partial volume with an implicit height atlas based on Haar-like features. In [4], we presented an alternative method such that only a single query slice is needed in order to achieve comparable results.

In Sect. 5.2, we introduce an iterative interpolation and regression approach. In contrast to established regression methods, [9, 11] we enhance our model with newly generated information after each iteration in order to refine the final model until convergence is reached.

We experimented with several regression methods from the Weka machine learning package [8]. However, simple approaches like linear regression did not yield a sufficient prediction accuracy and more complicated approaches like support vector regression using non-linear kernel functions could not cope with the enormous amount of training data. Therefore, we decided to employ instance-

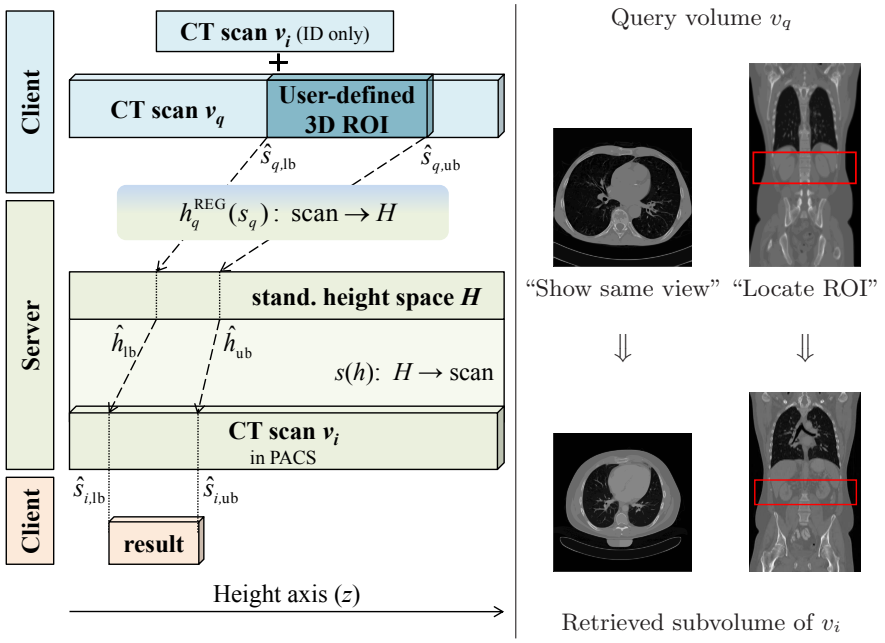

Fig. 1. Workflow of ROI retrieval and two example queries. The first query is specified by a ROI of only one slice, the second is given by a 3D ROI.

based regression which is robust and sufficiently fast when employing techniques of efficiently computing the k-nearest neighbors (k-NN). In particular, we employ k-NN queries being based on the X-Tree [3]. Let us note that there are multiple other index structures [12] for speeding up the same type of query. We decided to employ the X-Tree because it represents an extension of the standard R*-Tree [10] which is better suited for higher dimensionalities.

Current database systems like RasDaMan [2] already support conventional region of interest queries in raster data like CT scans. Nevertheless, the system needs to know the coordinate system in which the query is applied in order to navigate to the requested region. As we do not know the complete coordinate systems of the patients' CT scans in advance and since patients differ in height and body proportions, and thus, locations along the z-axis are not standardized, a globally fixed coordinate system will not be available in our setting. Therefore, our new approach represents a way to bridge the gap between the coordinates in the example scan and the coordinate system of the result scan.

3 Example-Based ROI Queries

In this section, we specify the proposed query process and give an overview of the proposed system. Formally, a dataset consists of n volumes $v_i \in \mathbb{N}^{x(i)\times y(i)\times z(i)}$ with $i \in \{1, \ldots, n\}$ and varying voxel dimensions x, y, z. The height model H is

an interval $[h_{\min}, h_{\max}] \in \mathbb{R}_o^+$ representing the extension of the human body in the z-axis. A mapping function $h_i : \mathbb{N} \to H$ maps slices of volume v_i to a height value $h \in H$. Correspondingly, the reverse mapping function $s_i : H \to \mathbb{N}$ maps a position h in the height model to a slice number s in v_i. A matching point $p = (\mathbf{s}_p, \mathbf{h}_p, \mathbf{w}_p) \in \mathbb{N} \times H \times \mathbb{R}$ is a triple of a slice number, its corresponding height value in H and a reliability weight w. We use $p_{i,j}$ for naming the j^{th} matching point in scan v_i.

In our system, a region of interest (ROI) query is specified by a set of consecutive slices $(\hat{\mathbf{s}}_{e,\text{lb}}, \ldots, \hat{\mathbf{s}}_{e,\text{ub}}) \subseteq \{0, \ldots, z(e) - 1\}$ from an example scan v_e and it retrieves a consecutive sequence of CT slices $(\hat{\mathbf{s}}_{i,\text{lb}}, \ldots, \hat{\mathbf{s}}_{i,\text{ub}}) \subseteq \{0, \ldots, z(i) - 1\}$ from the result scan v_i.

Fig. 1 illustrates the complete workflow of query processing for example-based ROI queries. A user specifies the ROI query on the client computer by marking a region in an example scan v_e. Additionally, the queried scan v_i has to be identified for the server. Let us note that it is not necessary to transfer the complete marked subset of the example scan. Instead it is sufficient to transfer a scale-reduced version of the first and the last slice of the subset. After receiving the slices, the server performs a feature extraction step generating image descriptors for both slices. As an alternative, the client computer might directly compute the required image descriptors and only transfer the descriptors.

In the next step, the server employs a mapping function to predict height values $\hat{h}_{\text{lb}}$ and $\hat{h}_{\text{ub}}$ to describe the borders of the query interval in the height model H. In our system, $h_e^{\text{REG}}(s)$ is implemented by instance-based regression (cf. Sect. 4). Afterwards, our algorithm starts with aligning the result scan v_i to the height model H by employing the algorithm described in Sect. 5.2. In particular, the algorithm employs $h_i^{\text{REG}}(s)$ for generating matching points P_i which are required for the reverse mapping function $s_i(h)_{P_i}$, an interpolation function described in Sect. 5.1. Once the quality of $s_i(h)_{P_i}$ is satisfying, the server selects the sequence of slices $(\hat{\mathbf{s}}_{i,\text{lb}}, \ldots, \hat{\mathbf{s}}_{i,\text{ub}}) \subseteq \{0, \ldots, z(v_i) - 1\}$ from v_i corresponding to the height interval $[\hat{h}_{\text{lb}}, \hat{h}_{\text{ub}}]$ and returns them to the client as the query result. Let us note that $(\hat{\mathbf{s}}_{i,\text{lb}}, \ldots, \hat{\mathbf{s}}_{i,\text{ub}})$ is extended by the amount of slices corresponding to 90 % of the expected prediction error in order to compensate for the inaccuracy of $h_i^{\text{REG}}(s)$.

Table 1 displays an overview of the defined parameters including some additional annotations that will be introduced in the next sections.

4 Efficient Instance-Based Regression

In this section, we introduce our method for mapping a single slice into the standardized height scale H. We already mentioned that there exist methods for landmark and organ detection which mark slices in the scan with the detected landmarks or organs. [13] Using multiple landmarks detected at slices $\mathbf{s}_{i,j}$, which can be mapped to anatomical concepts with known standardized height positions $\mathbf{h}_j$ and reliabilities $\mathbf{w}_j$, we can infer the standardized height of our queried slice. The landmark detector of [13] being used in our experiments was not

Table 1. Notation of frequently used parameters.

$v_i \in \mathbb{N}^{x(i)\times y(i)\times z(i)}$	one volume ($i \in \{1,\ldots,n\}$)
$H \in \mathbb{R}_o^+$	standardized height space / height model
$\mathbf{h}_j$	one height value in H
$\mathbf{s}_{i,j}$	one slice number of v_i in $\{0,\ldots,z(i)-1\}$
$p = (\mathbf{s}_{i,p}, \mathbf{h}_p, \mathbf{w}_p)$	matching point with reliability weight $\mathbf{w}_j$
P_i	set of matching points of v_i
$h_i^{\mathrm{REG}}(s)$	regression function of $\mathbb{N} \to H$
$s_i(h)_{P_i}$	interpolation function of $H \to \mathbb{N}$ using a set P_i
$(\hat{\mathbf{s}}_{i,\mathrm{lb}},\ldots,\hat{\mathbf{s}}_{i,\mathrm{ub}})$	slice range in v_i
$[h_{\mathrm{lb}}, h_{\mathrm{ub}}]$	interval in H
$F(s_{i,j}) : \mathbb{N} \to \mathcal{F} = \mathbb{R}^d$	image feature transformation of slice j of v_i with $d \in \mathbb{N}$
T_R	training set for regression

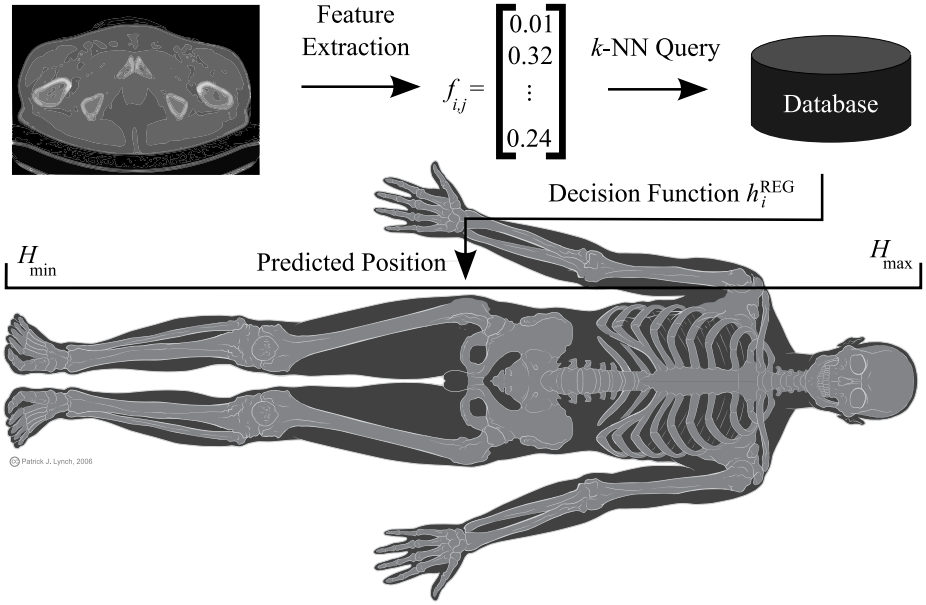

Fig. 2. Overview of content-based matching point generation using instance-based regression on HoGs. (Human model visualization taken from Patrick J. Lynch, medical illustrator and C. Carl Jaffe, MD, cardiologist at http://commons.wikimedia.org/wiki/File:Skeleton_whole_body_ant_lat_views.svg)

able to detect landmarks for all available scans, though. The reason why the detector failed to find landmarks were the following: The image quality is too fuzzy for the detector, the body region covered by the scan is not big enough or only a single slice is available. Further drawbacks of employing landmark detection for generating matching points are the complexity and availability of reliable detectors and that their runtimes are not suitable for interactive query

processing. In order to allow instant query processing on arbitrary scans, a faster and more flexible method should be employed that can efficiently generate a matching point for any given slice in the queried scan.

The idea behind our method is to represent the slice of interest $\mathbf{s}_{i,j}$ of the scan v_i by an image descriptor and to employ regression techniques to predict its height value in H. To train the regression function, we employ a training set of height annotated CT scans where each slice is labelled by a height value $h \in H$. An overview of this approach can be see in Fig. 2.

We use a 7 bins histogram of oriented gradients (HoG) which is applied on the cells of a 5x5 grid of the image's sub-regions. In contrast to the image descriptor introduced in [4], we omit the additional global extraction cell. Thus, the concatenated HoGs form a descriptor of size $d = (5 \cdot 5) \cdot 7 = 175$. Since our query algorithm requires multiple online feature extraction steps per query, we down-scale the images to a 100x100 pixels resolution before feature extraction for speeding up feature generation. We denote by $F(\mathbf{s}_{i,j}) : \mathbb{N} \rightarrow \mathbb{R}^d$ the feature transformation of the $\mathbf{s}_{i,j}^{\text{th}}$ slice of volume v_i to the final, d-dimensional feature space.

A d-dimensional feature vector $f_{i,j}$ corresponding to the j^{th} slice of the scan v_i can be mapped to the height model H with any given regression function. However, in our experiments the majority of standard regression methods either required extensive training times on our large training datasets of up to 900 000 training examples or they did not yield the required prediction quality. Therefore, we employ an instance-based approach to regression determining the k-nearest neighbors ($k-$NN) of the given feature vector $f_{i,j}$ in the training set T_R, consisting of image features r with existing labels $h(r) \in H$, w.r.t. Euclidean distance. Afterwards, the height of slice $\mathbf{s}_{i,j}$ in scan v_i is predicted using the following decision function:

$$h_i^{\text{REG}}(\mathbf{s}_{i,j}) = \text{median}_{\{r \in T_R \mid r \in k\text{-NN of } F(\mathbf{s}_{i,j})\}} \{h(r)\} \ . \tag{1}$$

Although instance-based regression does not suffer from extensive training times, the cost for large example datasets has to be spent at prediction time. However, the prediction rule does only require to process a single kNN query and thus, allows us to use optimization methods for this well-examined problem.

In order to allow efficient query processing, we transform the high-dimensional feature space of the proposed image features into a lower-dimensional space which can be indexed by suitable spatial index structures. For this paper, we use an X-Tree [3], which is well-suited for data of medium dimension.

We reduce dimensionality d in a supervised way employing Relevant Component Analysis (RCA) [1] with the goal of maintaining the principal information of the original feature vectors $r \in \mathbb{R}^d$. RCA transforms the data into a space minimizing the co-variances within subsets of the data, which are supposed to be similar, the so-called chunklets. Chunklets can be defined by matching a set of class labels or by using clusters. In our setting, we sort the data points used for training the feature transformation according to their height labels and retrieve a pre-defined number (150 chunklets performed well) of equally-sized data subsets. For our datasets, using a 10-dimensional feature representation turned out

to be a viable trade-off between prediction time and accuracy. On the average, a query took 22 ms while yielding an average prediction error of only 1.88 cm.

When using positions $\hat{h}_{i,j}^{\mathrm{REG}}$ computed with $h_i^{\mathrm{REG}}(\mathbf{s}_{i,j})$ as matching points for answering ROI queries, we are also interested in how reliable they are. One way to determine a position's reliability is to use the variance of the k-nearest neighbors, with a low variance indicating a reliable prediction. [4] However, in our setting, the best predictions could be observerd with $k = 1$ or $k = 2$. Since building a deviation on 1 or 2 samples does not make any sense, we had to develop an alternative approach for approximating the prediction quality.

We thus perform an additional pre-processing step assigning weights to all instances r in the training database T_R. The weight $w(r)$ of instance r is determined in a leave-one-out run of $h_i^{\mathrm{REG}}(r)$ on T_R. The predicted height value $\hat{h}_r^{\mathrm{REG}}$ is compared to the true position $h(r)$, resulting in the weight $w(r)$:

$$w(r) = 0.1 / \left(0.1 + \left|\hat{h}_r^{\mathrm{REG}} - h(r)\right|\right) \ . \tag{2}$$

The reliability of a predicted value $\hat{h}_{i,j}^{\mathrm{REG}}$ is now approximated by the average weight $w(x)$ over all k-nearest neighbors x of the queried instance r.

5 Answering ROI Queries

In the following, we define a method for retrieving an ROI in a volume v_i for which no matching points are yet available. As mentioned before, the first step of an ROI query is to determine the query interval $[h_{\mathrm{lb}}, h_{\mathrm{ub}}]$ in the standardized space H corresponding to the ROI in the example scan v_e. We employ instance-based regression as proposed in the previous section for predicting the height of the lower and upper bound of the marked ROI of the example scan.

Once such a query interval is defined, we need to collect a set of matching points $p \in P_i$ for being able to interpolate from the standardized height space H to the volume space of a slice $\mathbf{s}_{i,j} \in \mathbb{N}$ of volume v_i. We will now introduce the interpolation approach used for this purpose.

5.1 Interpolation using Matching Points

For mapping model positions $h_j \in H$ to slices $\mathbf{s}_{i,j} \in \mathbb{N}$, we use an interpolation approach based on a linear function. However, due to varying body proportions, the patient's position on the scanner table and the imperfect reliability of the used matching points P_i, a strictly linear model is not sufficient. Therefore, we additionally consider a non-linear component in our function which adds an instance-based off-set, comparable to an approach introduced in [7].

The mapping function $s_i(\mathbf{h}_q)_{P_i}$ for mapping the height value $\mathbf{h}_q$ to a slice number $\mathbf{s}_{i,q} \in \mathbb{N}$ is dependent on the scan v_i and the set of matching points P_i. We approximate the slice spacing δ_i describing the thickness of a slice in the target space H as the median slice spacing over all pairs of matching points in P_i as $\hat{\delta}_i = \mathrm{median}_{p,p' \in P_i, \mathbf{s}_{i,p} \neq \mathbf{s}_{i,p'}} \left|\mathbf{h}_p - \mathbf{h}_{p'}\right| / \left|\mathbf{s}_{i,p} - \mathbf{s}_{i,p'}\right|$.

Let us note that the median is used to achieve a higher stability against outliers caused by unreliable matching points. We define $s_i : H \to \mathbb{N}$ as:

$$s_i(\mathbf{h}_q)_{P_i} = \frac{\mathbf{h}_q}{\delta_i} - \frac{\sum\limits_{p \in P_i} \mathbf{w}_p \cdot \min\left(1, |\mathbf{h}_q - \mathbf{h}_p|^{-1}\right) \cdot \left(\frac{\mathbf{h}_p}{\delta_i} - \mathbf{s}_{i,p}\right)}{\sum\limits_{p \in P_i} \mathbf{w}_p \cdot \min\left(1, |\mathbf{h}_q - \mathbf{h}_p|^{-1}\right)} \,. \tag{3}$$

In order to avoid the case of $\mathbf{h}_p = \mathbf{h}_q$, we limit the maximal contribution of a matching point p with the minimum terms. Other, more complex interpolation models usually performed less stable and are thus omitted from this paper.

5.2 Retrieval Algorithm

The quality of the mapping $s_i(\mathbf{h})_{P_i}$ directly depends on the quality of the matching points $p \in P_i$. Having a large set of matching points increases the mapping quality because it increases the likelihood that reliable matching points being close to $[h_{\text{lb}}, h_{\text{ub}}]$ are available. Furthermore, having more matching points decreases the impact of low-quality matching points. However, increasing the amount of matching points is connected with generating costs for feature transformation, dimension reduction and regression.

Thus, we want to employ a minimal number of matching points while achieving high interpolation quality. The core idea of our method is to start with a minimal set of matching points and to measure the quality of the induced mapping function. As long as this quality is significantly increasing, we select slices in the queried scan and induce additional matching points using the regression method proposed in Sect. 4. This process is illustrated in Fig. 3.

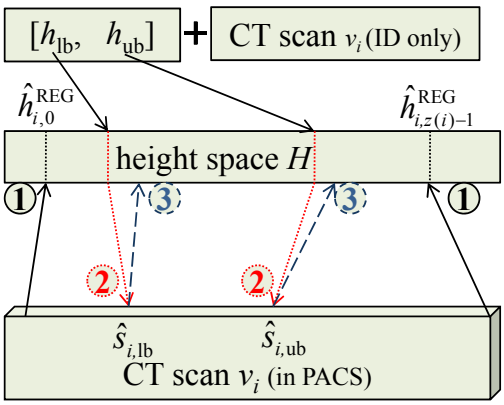

Fig. 3. First steps of Algorithm 1: a query range $[h_{\text{lb}}, h_{\text{ub}}]$ is to be found in a scan v_i. In the initial step (1), the seed slices forming the matching points P_i are selected and mapped to H with h_i^{REG}. In step (2), P_i is used for interpolating a result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$ in v_i. Step (3) validates this result range using h_i^{REG} and decides whether a new range should be tested.

Algorithm 1 ROI Query

Input: v_i: Query volume, $[h_{\text{lb}}, h_{\text{ub}}]$: query interval in H, h_i^{REG}: height regression function $\mathbb{N} \to H$, s_i: interpolation function $H \to \mathbb{N}$, ϵ: tolerated result range deviation

1: **function** ROI_QUERY($v_i, [h_{\text{lb}}, h_{\text{ub}}], h_i^{\text{REG}}, s_i, \epsilon$)
2: $P_i = (\mathbf{s}_i, \mathbf{h}, \mathbf{w}) \leftarrow \text{INIT}(v_i, h_i^{\text{REG}})$ ▷ Initialize P_i
3: $\{\text{err}_{\text{lb}}, \text{err}_{\text{ub}}\} \leftarrow \{\infty, \infty\}$ ▷ Errors for lb and ub
4: $\{\hat{s}_{\text{lb}}, \hat{s}_{\text{ub}}\} \leftarrow \text{NULL}$ ▷ Resulting slice numbers
5: **while** $\text{err}_{\text{lb}} > \epsilon$ **or** $\text{err}_{\text{ub}} > \epsilon$ **do**
6: $\{\hat{s}^*_{\text{lb}}, \hat{s}^*_{\text{ub}}\} \leftarrow \{s_i(h_{\text{lb}}), s_i(h_{\text{ub}})\}$ ▷ Interpolation
7: $\{\hat{h}^{\text{REG}}_{\text{lb}}, \hat{h}^{\text{REG}}_{\text{ub}}\} \leftarrow \{h_i^{\text{REG}}(\hat{s}^*_{\text{lb}}), h_i^{\text{REG}}(\hat{s}^*_{\text{ub}})\}$ ▷ Regression
8: $\{\text{err}^*_{\text{lb}}, \text{err}^*_{\text{ub}}\} \leftarrow \left\{\left|\hat{h}^{\text{REG}}_{\text{lb}} - h_{\text{lb}}\right|, \left|\hat{h}^{\text{REG}}_{\text{ub}} - h_{\text{ub}}\right|\right\}$
9: **if** $\text{err}_{\text{lb}} > \text{err}^*_{\text{lb}}$ **then** ▷ New lower bound
10: $\hat{s}_{\text{lb}} \leftarrow \hat{s}^*_{\text{lb}}$; $\text{err}_{\text{lb}} \leftarrow \text{err}^*_{\text{lb}}$
11: **if** $\text{err}_{\text{ub}} > \text{err}^*_{\text{ub}}$ **then** ▷ New upper bound
12: $\hat{s}_{\text{ub}} \leftarrow \hat{s}^*_{\text{ub}}$; $\text{err}_{\text{ub}} \leftarrow \text{err}^*_{\text{ub}}$
13: Get weights $\hat{w}^{\text{REG}}_{\text{lb}}, \hat{w}^{\text{REG}}_{\text{ub}}$ of new matching points
14: $P_i.\text{APPEND}\left((\hat{s}^*_{\text{lb}}, \hat{h}^{\text{REG}}_{\text{lb}}, \hat{w}^{\text{REG}}_{\text{lb}}), (\hat{s}^*_{\text{ub}}, \hat{h}^{\text{REG}}_{\text{ub}}, \hat{w}^{\text{REG}}_{\text{ub}})\right)$ ▷ Extend P_i
15: **return** $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$

Output: Result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$

We use the mechanism of manually generating matching points via regression $h_i^{\text{REG}}(\mathbf{s})$ for measuring the quality of a predicted result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$. The error of a prediction $\hat{s}_c$ for a value $\mathbf{h}_c$ is thus defined as: $|h_i^{\text{REG}}(\hat{s}_c) - \mathbf{h}_c|$.

Since $h_i^{\text{REG}}(\hat{s}_c)$ is fixed during query processing, the only possible way to reduce the error is to improve the quality of the matching points. This can happen by either updating their weights $\mathbf{w}_j$ or by adding further matching points. Even though it is sensible to update weights in special cases, the core component of our algorithm involves the second improvement variant.

For a given query interval $[\mathbf{h}_{\text{lb}}, \mathbf{h}_{\text{ub}}]$ our method proceeds as follows (see also Algorithm 1): We select g equally-spaced seed slices $\mathbf{s}_i \subset \{0, \ldots, z(i) - 1\}$ to generate an initial set of matching points by predicting their positions as $\hat{\mathbf{h}} \in H^g$ using instance-based regression $h_i^{\text{REG}}(\mathbf{s}_i)$. Using the weights obtained in the regression procedure we can induce an initial set of matching points $P_i = (\mathbf{s}_i, \hat{\mathbf{h}}, \mathbf{w})$. We are now free to make our first prediction of the result range.

We interpolate $\hat{s}^*_{\text{lb}} = s_i(h_{\text{lb}})_{P_i}$ and $\hat{s}^*_{\text{ub}} = s_i(h_{\text{ub}})_{P_i}$ in the queried scan using the current set of matching points P_i. Next, we employ $h_i^{\text{REG}}(\mathbf{s})$ on $(\hat{s}^*_{\text{lb}}, \ldots, \hat{s}^*_{\text{ub}})$ and determine the prediction error estimate. If the lower or upper bound ($\hat{s}^*_{\text{lb}}$ or $\hat{s}^*_{\text{ub}}$) has been improved compared to the minimal error observed so far, we update the corresponding lower or upper bound ($\hat{s}_{\text{lb}}$ or $\hat{s}_{\text{ub}}$). Finally, we augment the set of matching points P_i by the regression prediction $h_i^{\text{REG}}(\mathbf{s})$ for the boundaries of the target range $\hat{s}^*_{\text{lb}}$ and $\hat{s}^*_{\text{ub}}$. The algorithm terminates if the improvement on both sides of the target range is less than ϵ.

For simplicity reasons, this algorithm omits a number of special cases. Since the derivation of matching points via regression is expensive due to the overhead of feature generation, the algorithm has to ensure that no slice number of v_i is tested multiple times. The search procedure should stop, once there is no more change to the set of matching points P_i because this usually means that the volume is not well enough resolved for perfectly matching the target range. It is also beneficial to test for both bounds whether a new matching point generated for the opposite bound is better suited. Additionally, if only one bound has been established in an acceptable quality, but it remains stable over a couple of iterations, one should refrain from trying to further improve this bound by costly regression calls and only update the opposite bound.

Furthermore, a number of exceptions should be handled: both s_i and h_i^{REG} can be mapped outside of their allowed ranges. In the case of s_i, this may be an indication that the query range is not contained in the volume. Repeated range violations should thus terminate the algorithm with an indication of a mismatch or a partial match. If $h_i^{\mathrm{REG}}(\mathbf{s})$ goes astray, this can either be noise in the regression function or it can be a reason for down-weighting the current set P_i and for seeking further matching points.

6 Experimental Validation

In the following, we present the results of our experimental evaluation by measuring the quality of the retrieval system and by demonstrating the improved query time of our complete system. All of our experiments were performed on subsets of a repository of 4 479 CT scans provided by the Imaging Science Institute of the University Hospital Erlangen for the Theseus Medico project. The scans display various subregions of the human body, starting with the coccyx and ending with the top of the head.

For generating a ground truth of height labels, we used the landmark detector of [13] annotating each scan with up to 22 landmarks. This restricted the dataset to 2 526 scans where landmarks could be detected. The complete repository contains more than a million single CT slices comprising a data volume of 520 GB.

We implemented our prototype in JAVA 1.6 and stored the scans and their annotations in a MySQL database. To simulate the distributed environment of a radiology center, we employed the LAN and the workstations in our lab consisting of common workstations of varying type and configuration being connected by a 100 Mb Ethernet.

6.1 Prediction via Regression

In the following section, we first examine the used image features on their suitability for k-NN regression. Afterwards, we describe the beneficial effects of reducing the original image feature space using RCA.

Regression Quality For these experiments, we have to provide height labels $h(r)$ as ground truth for all entries of the required regression database, i.e. for each scan in the training dataset T_R. Basically, there are two methods for generating these labels. The first is to manually mark the highest and the lowest point in all scans of a database and to linearly interpolate the height values. [4] We refer to this method as manual labelling.

Since instance-based regression profits from a larger database, we also use an automatic labelling method. It assigns height labels to the slices of a volume with the inverse interpolation approach introduced in Sect. 5.1, using standardized landmark positions as matching points. In our experiments, we use the 22 landmarks of [13], marking meaningful anatomical points, which could be detected in 2 526 of our CT scans. These landmarks are time-expensive to compute and their computation fails in the remaining 1 953 scans of our dataset. We will refer to the height labels generated with this interpolation procedure as automatic labelling.

In our first test, we measure the regression performance of the original image descriptors, which have not yet been transformed by RCA. We first examine a manually annotated dataset of 33 CT scans with a total of 18 654 slices. The average leave-one-out prediction errors are displayed in the first row of Table 2. Let us note that leave-one-out in these experiments means that only slices from other scans than the current query scan are accepted as k-nearest neighbors in order to exclude distorting effects of within-scan similarities. When testing k parameters between 1 and 5, we found $k = 1$ to be the best setting for all experiments using the original slice descriptors.

Table 2. Leave-one-out validation (LOO) errors [in cm] of k-NN slice mapping $h_i^{\mathrm{REG}}(\mathbf{s}_{i,j})$ for two database sizes of n CT scans with m slices.

Ground Truth	n	m	Error [cm]	Time / Query [ms]
manual	33	18 654	4.285	18
automatic	33	18 654	3.584	18
automatic	376	172 318	1.465	200

The next row of Table 2 displays the error of the same dataset, which has been labelled automatically. Our experiments show that the average registration error of 4.3 cm of the manual labelling is even lowered to 3.6 cm when using the automatic labelling. Thus, we can safely test our regression method on larger datasets, which have been automatically annotated. This allows to fully exploit the strength of the proposed instance-based regression approach. For smaller databases, alternative regression approaches should be considered, however, with the wealth of information available, our lazy learner is very hard to beat.

We observe a steady improvement of the empirical errors for increasing database sizes, however, this comes at the price of longer runtimes. For a dataset

of 376 volumes consisting of 172 318 slices, a single query performed as sequential scan in main memory requires 200 ms. The additional cost of keeping the complete training database in main memory poses an additional drawback. The following section evaluates our method of runtime optimization, by using an efficient indexing scheme.

Speed-up via RCA and Indexing In order to speed up regression, we index the training data in an X-Tree [3] after reducing the dimensionality via RCA. We tested the target dimensions $5, 10, 25$ and 50. Using an index, we could now employ the complete dataset of 2 526 scans. We used a subset of 697 scans (= 163 525 slices) as training set for the RCA and tested the performance on the remaining 2 104 scans (901 326 instances). Table 3 shows the average leave-one-out (LOO) errors and query runtimes (excluding the time for feature generation) for the indexes generated from the test set.

Table 3. LOO regression errors [in cm] for RCA-transformed data with query times [in ms] in an X-Tree representing 2104 scans.

Dimension	Error [cm]	Time / Query [ms]
5	2.764	4
10	1.881	22
25	1.343	440
50	1.209	2966

As can be been seen in Table 3, the curse of dimensionality causes the X-Tree to lose much of its effectiveness for increasing dimensions. Additionally, the error does only moderately increase for smaller dimensions. Based on these observations, we consider the 10 dimensional data set as the best trade-off, having a prediction error of 1.88 cm and a query time of 22 ms. We use this dataset for all following experiments. The total runtime required for feature generation is combined from the actual feature generation for a down-scaled version of the query slice (20 ms) and the time required for RCA transformation (0.1 ms). Thus, our selected query configuration results in a total prediction time of 42 ms.

Next, in order to validate the performance of the proposed ROI query workflow we will first analyze the accuracy of the retrieved ROIs and then proceed with an examination of retrieval times.

6.2 Precision of ROI Queries

We could again use automatically detected landmarks for defining a ground truth of lower and upper bounds, however, we cannot guarantee for the correctness of these matching points.

Therefore, we generated a new set of annotation points with five new landmark types: “lower plate of the twelfth thoracic vertebra”, “lower bound of

coccyx", "sacral promontory", "cranial sternum" and "lower xiphoid process". These landmarks were hand-annotated by a medical expert for providing a set of markers which have been verified visually.

In Table 4, we show the results of predicting all visible intervals with ROI queries formed by pairs of these landmarks in the dataset of 33 manually annotated volumes. As not all landmarks were visible in all volumes, only 158 intervals could be tested. Since the annotation error – the deviation of these markers from their expected positions – is at 2.579 cm, we cannot expect the queries to produce more reliable predictions.

Using Algorithm 1 with varying grid sizes g for the initial matching points P_i provides good predictions. We observe, however, that using a larger number of seed points only mildly improves the accuracy of the predictions, but it greatly increases the number of matching points being generated by regression (q). We conclude that two seed points are sufficient for our simple optimization scheme. Any more sophisticated optimization procedures should rather involve an intelligent screening of the proposed result range $(\hat{\mathbf{s}}_{\mathrm{lb}}, \ldots, \hat{\mathbf{s}}_{\mathrm{ub}})$ than use more seed points.

Table 4. Average deviation [in cm] of the result ROI of Algorithm 1 from the manually marked ROIs with the number of regression queries q and the runtime per query.

Error Measure [cm]: $\mathrm{err}(\hat{\mathbf{s}}_{\mathrm{lb}}) + \mathrm{err}(\hat{\mathbf{s}}_{\mathrm{ub}})$

Annotation Error: 2.579 cm

	ROI prediction with Algorithm 1		
g	Error [cm]	q	Time / Query [ms]
2	2.655	6.8	1 273
5	2.549	9.2	1 951
10	2.430	15.2	3 032
25	2.573	30.0	5 946
50	2.385	55.5	10 081

In Fig. 4 we see the cumulative distribution function $F(\text{error} \leq x\,\text{cm})$ for the analyzed query intervals. The 'Annotation' bars show the performance of the annotated ground truth landmarks, and the 'Algorithm 1' bars represent our ROI query algorithm using two seed points. There is almost no difference between the quality of the ground truth and our algorithm. The probability that the total prediction error $(\mathrm{err}(\hat{\mathbf{s}}_{\mathrm{lb}}) + \mathrm{err}(\hat{\mathbf{s}}_{\mathrm{ub}}))$ is at most 2 cm lies at 50 %. Again, with a height spacing of 5 mm, this means that in half of the cases, the retrieved range deviates by only two slices for each the lower and upper bound. When thus extending the returned query range by our pre-defined safety range, most returned subvolumes will completely contain the requested ROI.

We thus conclude that ROI queries can be efficiently answered by using Algorithm 1 with two initial matching points. The query time for grid size 2 is 1.5 seconds. Thus, our final experiments will show that the benefit of reducing volume queries to a region of interest strongly outweighs this cost.

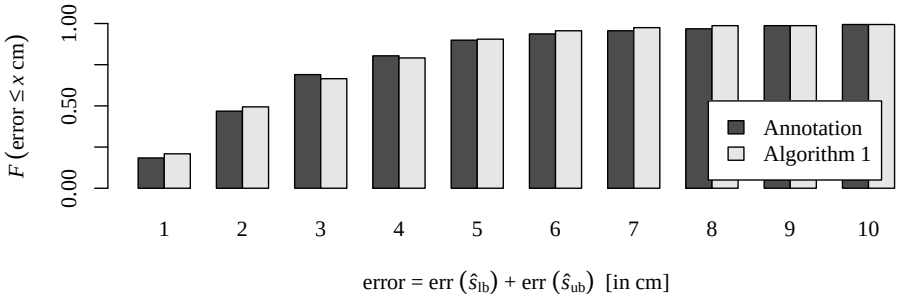

Fig. 4. Cumulative distribution function: $F(\text{error} \leq x\,\text{cm})$ (the steeper the better). It compares the error of Algorithm 1 with the quality of the used annotation points.

6.3 Runtime of ROI Queries

For our last experiment, we chose a random set of 20 volumes from the database and tested them against four ROI queries defined in an example scan. Two queries are aimed at organs ("Left kidney" and "Urinary bladder"), one query ranges from the top of the hip bone to the bottom of Vertebra L5 and the final query is only interested in the view of the arch of aorta. The four hereby defined query ranges have heights of 16.8, 9.6, 4.7 and 0.9 centimeters In Fig. 5, we display the retrieval times of the resulting ROIs and their fraction of the complete dataset of 12 240 slices. Loading the complete 20 volumes from the server takes 1 400 seconds, whereas transferring only the ROIs induced by the given concepts takes 60 to 400 seconds, including the computation overhead for finding the ROI.

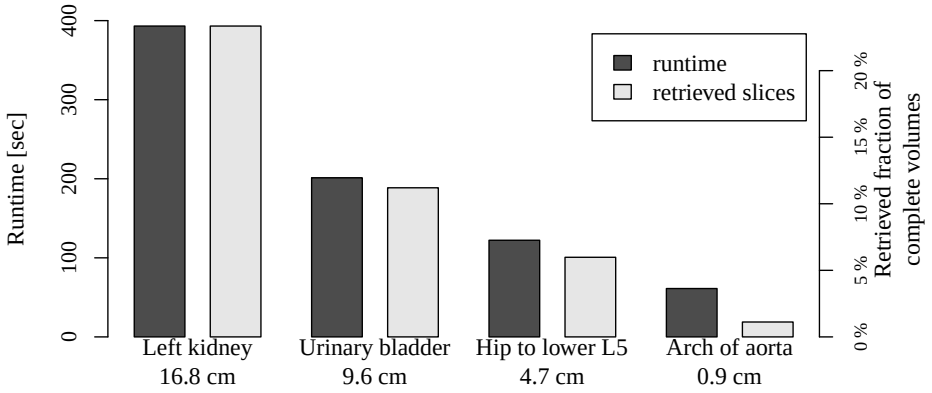

Fig. 5. Average runtimes (ten repetitions) and volume size reduction using ROI queries with Algorithm 1. Each experiment tests 20 volumes with a total of 12 240 slices. Loading the *complete* dataset takes 1 400 seconds.

To conclude, employing our system for answering ROI queries saved between $77 - 99\,\%$ of the loading time compared to the retrieval of the complete scan. Thus, in a clinical routine our system is capable to save valuable time as well as hardware resources.

7 Conclusion

In this paper, we proposed a method for processing region of interest (ROI) queries on a repository of CT scans. An ROI query is specified by giving an example ROI in another CT scan. Since CT scans are usually stored as stacks of 2D images representing a slice in the scan, the answer of an ROI query on a CT scan is a subset of the slices in the target scan representing an ROI which is equivalent to the query ROI.

After query specification, the system maps the ROI to a general height model of the human body. Then, the query region in the height model is mapped to a subregion of the queried scan containing the ROI. Technically, our system is based on an interpolation function using so-called matching points linking a CT scan to the height model. To guarantee the availability of matching points even for unannotated CT scans, we propose a method using content-based image descriptors and regression that can generate matching points for arbitrary slices in a scan. Finally, we propose a query algorithm for finding a stable mapping while deriving a minimal amount of matching points.

In our experimental evaluation, we validated the accuracy of our approach on a large database of 2 526 CT scans and displayed experiments for the reduced transfer volume of ROI queries being processed by our system.

For future work, we plan to extend our system to restrict ROIs in all three dimensions. We also aim to apply our retrieval solution to other types of 3D objects being stored in raster databases and to further, more general regression or interpolation problems.

Acknowledgements

The authors would like to thank Sascha Seifert et.al. for providing the executables of [13] and Robert Forbrig for generating the anatomical annotations used in Sect. 6.2. This research has been supported in part by the Theseus program in the Medico and CTC projects. They are funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020. The responsibility for this publication lies with the authors.

References

1. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC. pp. 11–18 (2003)

2. Baumann, P., Furtado, P., Ritsch, R., Widmann, N.: The RasDaMan approach to multidimensional database management. In: Proceedings of the 12th ACM Symposium on Applied Computing (ACM SAC), San Jose, CA. pp. 166–173. ACM (1997)
3. Berchtold, S., Keim, D.A., Kriegel, H.P.: The X-Tree: An index structure for high-dimensional data. In: Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB), Bombay, India (1996)
4. Emrich, T., Graf, F., Kriegel, H.P., Schubert, M., Thoma, M., Cavallaro, A.: CT slice localization via instance-based regression. In: Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA, USA. p. 762320 (2010)
5. Evans, A.C., Collins, D.L., Mills, S.R., Brownand, E.D., Kelly, R.L., Peters, T.M.: 3D statistical neuroanatomical models from 305 MRI volumes. In: IEEE Nuclear Science Symposium and Medical Imaging Conference (1993)
6. Feulner, J., Zhou, S.K., Seifert, S., Cavallaro, A., Hornegger, J., Comaniciu, D.: Estimating the body portion of CT volumes by matching histograms of visual words. In: Proceedings of the SPIE Medical Imaging 2009 Conference (SPIE), Lake Buena Vista, FL, USA. vol. 7259, p. 72591V (2009)
7. Haas, B., Coradi, T., Scholz, M., Kunz, P., Huber, M., Oppitz, U., André, L., Lengkeek, V., Huyskens, D., van Esch, A., Reddick, R.: Automatic segmentation of thoracic and pelvic CT images for radiotherapy planning using implicit anatomic knowledge and organ-specific segmentation strategies. Physics in Medicine and Biology 53(6), 1751–71 (2008)
8. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. ACM SIGKDD Explorations 11(1), 10–18 (2009)
9. Holland, P., Welsch, R.: Robust regression using iteratively reweighted least-squares. Communications in Statistics-Theory and Methods 6(9), 813–827 (1977)
10. Kriegel, H.P., Seeger, B., Schneider, R., Beckmann, N.: The R*-tree: An efficient access method for geographic information systems. In: Proceedings of the International Conference on Geographic Information Systems, Ottawa, Canada (1990)
11. Nguyen, T.: Robust estimation, regression and ranking with applications in portfolio optimization. Ph.D. thesis, Massachusetts Institute of Technology (2009)
12. Samet, H.: Foundations of Multidimensional and Metric Data Structures. Morgan Kaufmann, San Francisco (2006)
13. Seifert, S., Barbu, A., Zhou, S.K., Liu, D., Feulner, J., Huber, M., Suehling, M., Cavallaro, A., Comaniciu, D.: Hierarchical parsing and semantic navigation of full body CT data. In: Proceedings of the SPIE Medical Imaging 2009 Conference (SPIE), Lake Buena Vista, FL, USA. vol. 7259, p. 725902 (2009)
14. Seifert, S., Kelm, M., Möller, M., Mukherjee, S., Cavallaro, A., Huber, M., Comaniciu, D.: Semantic annotation of medical images. In: Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA, USA. vol. 7628, p. 762808 (2010)
15. Talairach, J., Tournoux, P.: Co-Planar Stereotaxic Atlas of the Human Brain 3-Dimensional Proportional System: An Approach to Cerebral Imaging. Thieme Medical Publishers (1988)