



Application note

Automatic segmentation of Computed Tomography (CT) images of domestic pig skeleton using a 3D expansion of Dijkstra's algorithm

Lars Erik Gangsei ^{a,*}, Jørgen Kongsro ^b^a Animalia, P.O. Box 396 – Økern, N-0513 Oslo, Norway^b Norsvin SA, P.O. Box 504, N-2304 Hamar, Norway

ARTICLE INFO

Article history:

Received 9 September 2015

Received in revised form 4 December 2015

Accepted 10 December 2015

Available online 2 January 2016

Keywords:

Computer Tomography

Pig

Skeleton

Segmentation

ABSTRACT

A 3D expansion of Dijkstra's algorithm used for automatic segmentation and identification of the bones in CT images of live pigs was developed and validated. The major bones in the skeletons of 208 out of 485 live pigs (43%) were segmented and identified from the images without major errors. The segmentation and identification is executed through 8 main operations: (1) identify the full bone structure by a threshold of Hounsfield units, (2) identify forelimbs by voxel connectivity and set landmarks, (3–8) segment out and identify the individual bones in different main parts of the bone structure by the 3D expansion of Dijkstra's algorithm. The algorithms described will constitute an important basis for further work applying CT in pig breeding and management.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Computer Tomography (CT) is a highly suitable method for identifying bone structures inside living and dead organisms containing a calcified skeletal structure, typically mammals and fish. The full skeleton structure might easily be roughly segmented out from the rest of body tissue by adding a threshold value, typically around 200 Hounsfield units (HU) (Fiebich et al., 1999), and by classifying all voxels with larger values as bone.

The ability to segment the carcass into identifiable bones has several purposes. Our ultimate goal is to construct a complete labeled body atlas for pigs. Segmented and identified bones from a plethora of pigs will be indispensable in the process of constructing the atlas, as the skeleton will constitute the framework of the atlas. The atlas made for mice by Dogdas et al. (2007), is an illustrative example for what we want to achieve. The bone segmentation might also be used for other purposes, for instance to diagnose diseases or undesirable qualities in the skeleton.

By a method known as atlas-based segmentation (Baiker et al., 2010; Cuadra et al., 2015), the different parts of the pig, i.e. the CT scan, might be identified as cuts, muscles, organs etc. The highest priority is to be able to identify the main commercial cuts in live animals.

The aim was to describe an approach for automatic segmentation and identification of the larger bones from CT images of live domestic pigs. To our knowledge, this is the first automatic algorithm for segmenting pig skeletons in a volume generated from CT images.

2. Material and methods

2.1. Animals and the CT scan

The material consists of in vivo CT scans from 485 boars at the Norsvin Delta test station for purebred boars (Norsvin SA). The pigs were purebred Norsvin Landrace and Duroc boars. The live weight is as close to 120 kg as practically possible. This is due to 120 kg representing the end point of the testing period for terminal boars, and is regarded as the optimal carcass weight (70–80 kg) in Norway. The samples were selected randomly from the annual breeding stock of 3500 boars tested in the Norsvin breeding program. The CT scanner was a GE Healthcare LightSpeed 32 VCT, and the settings used were 120 kV, slice thickness 1.25 mm and dynamic mA (400–500 mA) adjusting for object thickness.

Prior to CT scanning, the boars were sedated using Azaperone, 8 mg/kg i.m. (Stresnil Vet R, Janssen-Cilag Ltd, Buckinghamshire, UK). Boars were scanned, 45 min after injection, as the sedation was given to help facilitate the scanning procedure and improve image quality. All animals were cared for according to the laws and regulations for keeping pigs in Norway (Aasmundstad et al.,

* Corresponding author at: Animalia, P.O. Box 396 – Økern, N-0513 Oslo, Norway. Tel.: +47 95061231.

E-mail addresses: lars.erik.gangsei@animalia.no (L.E. Gangsei), jorgen.kongsro@norsvin.no (J. Kongsro).

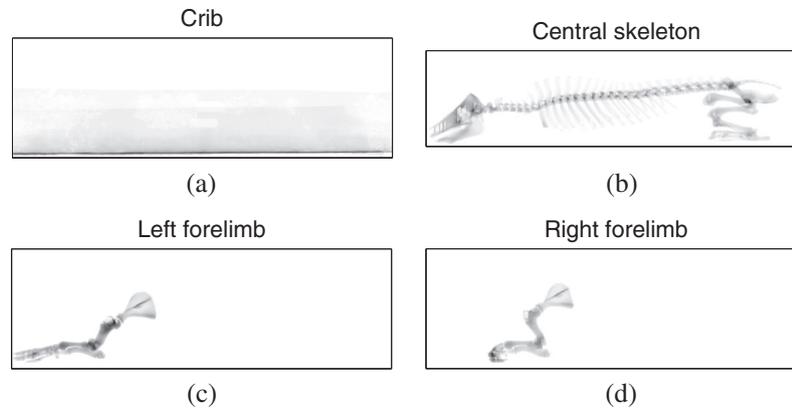


Fig. 1. An example of segmentation through connectivity. The 4 panels represents the 4 largest connected objects, i.e. crib (a), “central skeleton” (b), left forelimb (c), and right forelimb (d). The input is a binary image produced by setting a threshold at 180 HU for a CT image of a random pig.

2013). After sedation, the pigs were transported to the CT scanner using a crib made of fiberglass.

The software MATLAB (MATLAB, 2014), including the Image Processing Toolbox, was used for the segmentation. The code for landmark identification and the 3D expansion of Dijkstra’s algorithm was written from scratch. MATLAB code for a function conducting the 3D expansion of Dijkstra’s algorithm will be provided by authors upon request.

2.2. General methods for segmentation

We applied three main principles for identifying and segmenting the individual bones; (i) segmentation by connectivity, (ii) identification of points and lines and (iii) a 3D expansion of Dijkstra’s algorithm.

Segmentation by connectivity was primarily involved at the start of the process. A binary 3D image was constructed by applying a threshold value for bone ($HU > 180$). The different connected objects in this 3D binary image were labeled by the MATLAB function “bwconncomp”, which takes a binary image as input and returns an image where each connected object is labeled with a specific value. The connected objects inside the 3D binary image were identified by ranking their volumes or mass center points. The 4 largest connected objects from a randomly selected pig are shown in Fig. 1.

To identify landmarks, 2D projections of the skeleton were used extensively. The principles are explained through an example illustrated in Fig. 2. The steps were as follows:

1. A 2D image containing the sums of bone-voxels perpendicular to the sagittal plane (in this example) was constructed, i.e. high intensity areas represent areas with thick bones. One point, approximately corresponding to *Trochanter major*, was already known from similar techniques, as illustrated in Fig. 2a.
2. Based on the known point, a new region of interest (ROI) was defined based on euclidian distances. Points closer than 50 mm, a figure set a priori based on experience, from the known point were “masked out”, (Fig. 2b).
3. A new point was set at the point having maximal intensity inside the new ROI (Fig. 2c).
4. Steps similar to steps 2–3 were redone to find the rest of the points of interest resulting in a set of four “known fixed points” as illustrated in Fig. 2d.
5. A variant of this method is to define the ROI as a line, for instance the upper part of the spine, by using the MATLAB function “bwboundaries”. The points on top of each vertebra and points between vertebrae are set successively at local maximum and minimum sagittal values for this line.

Most of the bones had to be segmented and treated one by one. Dijkstra’s algorithm (Dijkstra, 1959), a basic method of image analysis, is well known for identifying the cheapest path between two nodes, and is thereby a suitable method for separating different objects in a 2D image. The method we applied might be viewed as an expansion of Dijkstra’s algorithm in order to separate a 3D, not 2D, object.

The first step was to identify a ROI (3D array) that encapsulates the surface in which the two bones were connected. These ROIs were automatically constructed based on the identified landmarks and prior information on bone dimensions and the spatial orientation of bones. As an example, the ROI illustrated by the box in Fig. 3 was constructed via a landmark in the top of the overarm, i.e. the lower right corner of the box. The horizontal distance for the ROI was set by prior knowledge. The two remaining dimensions of the ROI were easily set by the extreme values for bone voxels within the horizontal limitation.

In the next paragraphs, we will view the 3D binary input array (bone/ not bone) inside the ROI as a stack of 2D layers. For each layer, a virtual cut is constructed in the “cutting direction”. The sum of these cuts constitutes the segmentation surface. The directions of the layers and the cutting direction varied with the main direction of the segmentation surface, and were set based on prior knowledge. For instance, vertebrae were split approximately parallel to the transverse plane, pelvis was segmented from spine approximately parallel to the sagittal plane etc. When the segmentation surface is approximately parallel to the transverse plane as shown in Fig. 3, the layer direction is from top to bottom and cutting direction is from the right to left side of the pig. The two bones to be segmented had to penetrate the two opposite sides of the input array orientated perpendicular to both cutting- and layer-direction.

The first step of Dijkstra’s algorithm (2D) is to construct a cost matrix, i.e. a matrix defining the minimum cumulative cost of including any voxel to the virtual cut. The cost matrix for all layers was constructed by the standard Dijkstra’s algorithm. For the first layer, the input to the algorithm was the first layer of the binary input array. For the next layers the input consisted of the sum of the cost matrix from the previous layer and the corresponding layer of the input array. The final 3D cost array was the array of stacked 2D cost matrices.

The next step in Dijkstra’s algorithm is to construct the cheapest path, i.e. the segmentation surface. For all layers, these paths were constructed via the standard 2D Dijkstra’s algorithm based on the corresponding layers from the 3D cost array. The algorithm started with the last layer. After the path in one layer was identified, a region of “possible paths”, i.e. points connected to the identified path, was identified for the next layer. All points outside this pos-

Download English Version:

<https://daneshyari.com/en/article/84026>

Download Persian Version:

<https://daneshyari.com/article/84026>

[Daneshyari.com](https://daneshyari.com)