

# Semi-Automatic Segmentation of the Patellar Cartilage in MRI

Lorenz König<sup>1</sup>, Martin Groher<sup>1</sup>, Andreas Keil<sup>1</sup>, Christian Glaser<sup>2</sup>,  
Maximilian Reiser<sup>2</sup>, Nassir Navab<sup>1</sup>

<sup>1</sup>Chair for Computer Aided Medical Procedures, Technische Universität München

<sup>2</sup>Department of Clinical Radiology, Ludwig-Maximilians-Universität München

**Abstract.** We introduce a software system for semi-automatic segmentation of the patellar cartilage. By providing tools for sub-pixel accurate edge tracing, automatic contour completion, and adequate visualization we achieve a remarkable speed-up of the physician's segmentation process. Our results show the exactness that can be reached for cartilage segmentation if expertise and automation are merged in a meaningful way.

## 1 Introduction

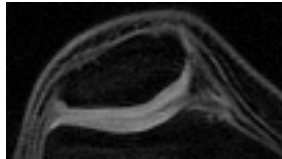
Damage to the articular cartilage is an early and decisive step in the development of osteoarthritis (OA) a major socio-economic burden nowadays. This disease is among the ten leading causes of continued disability world-wide and annual costs associated with OA are estimated to amount up to the equivalent of 1% of total productivity in the USA. This is the motivation to develop and to continuously refine therapies dedicated to cartilage repair contributing to at least postpone and to slow down the development and progression of OA. This, in turn, creates a strong need for non-invasive, accurate, and valid tools to establish appropriate indications for new treatment options, to monitor the disease process and to control therapeutic efficacy. MRI, especially with recent advances in scanner, coil and sequence design, is ideally suited for non-invasively evaluating the cartilage. In this respect and especially in view of statistical discriminatory power, quantitative data are desirable in contrast to more or less subjective semiquantitative evaluation by scoring methods. Such data would be cartilage volume, thickness and the size of the cartilage bone interface, all of these parameters being directly related with the disease process.

## 2 State of the Art

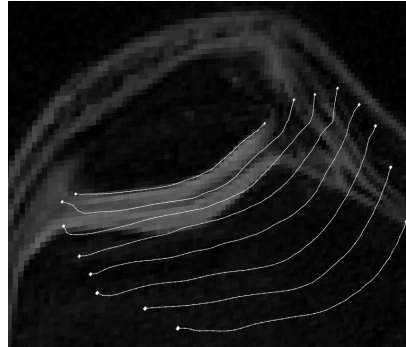
Several methods and tools for segmentation of the patellar cartilage in MR images have been presented so far [1,2,3]. Some of them fully rely on manual segmentation, which makes them cumbersome for daily routine. Some tools perform the segmentation more or less automatically and require the user only to define

a region of interest, for instance. In general, non-manual segmentation methods rely on gradient information from the image.

Depending on the MRI sequence used in the acquisition, however, slices with no contrast between the patellar cartilage and the adjacent femoral cartilage have to be segmented (Fig. 2). In these cases, the radiologists have to rely on their experience for segmenting the cartilage and therefore must be in full control of the segmentation process. Tools not allowing for that will hardly be accepted among physicians.



**Fig. 1.** MRI image of the knee with very little contrast between the patellar (above) and femoral (below) cartilages



**Fig. 2.** perspective visualization of the segmented cartilage bone interface throughout slices

### 3 Method

We present a robust system to support radiologists in semi-automatically segmenting patellar cartilage in MRI images in daily routine, which does not restrict manual control of the segmentation process but at the same time facilitates this task.

We work on three-dimensional one-channel data sets as supplied by the MRI scanners. The data is organized as an array of slices  $s^z(x, y)$ . The ratio of slice distance to in-slice pixel distance is about 5 to 10, and the slices are in most cases oriented perpendicular to the bone cartilage interface and cartilage surface. We therefore decided to do the segmentation in 2D slice by slice. Accordingly, the preprocessing (3.1) is done in 2D.

The segmented regions in each of the slices are described by their sub-pixel accurate boundaries. The boundaries are in turn represented by a sequence of fragments, any of which can be manually drawn (3.2) or semi-automatically computed (3.3) as well as corrected (3.4) at any time.

To support review of the results of segmentation, we provide a view of the current slice superimposed with the segmentation of adjacent slices (using an orthographic projection) as well as a perspective visualization of the segmentation as in fig. 2. In Section 3.5, we introduce a unified projection matrix, which allows for smooth transitions between orthographic and perspective projection without the user getting lost.

### 3.1 Sub-pixel Edge Detection

Sub-pixel edge detection is an important preprocessing step of the proposed system. Edge detection is done in 2D separately for each slice  $s^z(x, y)$  using a 2D facet model [4]. The idea is to find the maxima in the gradient image

$$g(x, y) = |\mathbf{g}(x, y)| = \left| \begin{pmatrix} s_x^z(x, y) \\ s_y^z(x, y) \end{pmatrix} \right| = \left| \begin{pmatrix} \frac{\partial}{\partial x} s^z(x, y) \\ \frac{\partial}{\partial y} s^z(x, y) \end{pmatrix} \right|$$

which is obtained using the Sobel operator. In the following, we describe the algorithm for one pixel at  $(i, j)$ .

Based on the  $3 \times 3$  environment of  $(i, j)$ , the gradient image  $g$  is locally approximated by a second degree polynomial in two variables  $p(x, y)$ . The maximum of  $p$  along the straight line defined by the gradient vector  $\mathbf{g}$  indicates a point  $E$  on the edge in the original image  $s$ . If  $E$  lies within the boundaries of the pixel at  $(i, j)$ , we accept  $E$  as an edge point for further processing. We also label it with the value of  $g(i, j)$  to indicate the sharpness of the edge at this point.

This is done for all the pixels in every slice. The result is a set of edge points and their respective sharpness for each slice.

### 3.2 Manual Tracing of Contours

As a first interaction step, we provide tools to manually trace edges for segmentation. This can be accomplished by drawing with the mouse (also with sub-pixel accuracy), as well as by relying on the previously computed edge points. In the latter case, for every point the mouse cursor passes, the edge points in a window around the cursor are weighted depending on the distance to the cursor and the sharpness of the edge they represent. Thus, the contour made up by these points is close to the trace of the mouse cursor, and is located on a sharp edge of the current slice with sub-pixel accuracy.

### 3.3 Semi-Automatic Tracing and Propagation of Contours

We also provide the user with a tool to semi-automatically find the contours of the patella. User interaction is minimized to the input of a starting point  $A$  and an end point  $B$  of the contour fragment to be traced. Between  $A$  and  $B$ , a guiding contour  $c$  is constructed either from previously segmented contours in adjacent slices, if there exist any. Otherwise, we simply use a straight line. The contour fragment  $f$  to be found is initialized with  $A$ . Let  $E$  denote the last point of  $f$ .

The guiding contour is then discretized into a sequence of control points  $C_i$  at steps of one third of a pixel. For each  $C_i$ , the following steps are performed: (1) A set of edge points having a distance of less than 5 pixels to the end  $E$  of  $f$  is computed. (2) These edge points are weighted with respect to (a) their respective gradient magnitude, (b) their distance from  $E$ , (c) their position relative to  $E$

compared to the direction of  $c$  in the environment of  $C_i$ , (d) their position relative to  $E$  compared to the direction of  $f$  at  $E$ , and (e) whether they are farther from  $C_i$  than  $E$ . The weight for (e) increases as  $C_i$  gets closer to  $B$ . (3) The edge point with the highest weight is appended to  $f$ , if it is different from its current end point  $E$ .

Thus, a contour fragment  $f$  is constructed, that consists of rather uniformly spaced edge points with high magnitude, is shaped smoothly and similar to the guiding contour  $c$ .

### 3.4 Correction and User Interaction

Correction of segmentation results (be they generated automatically or manually) is simply done by re-drawing the part of the contour that is to be corrected. Similar to the manual segmentation, re-drawing can be done either fully manually or based on the previously calculated edge points. For the first and last point of a newly drawn contour fragment, the respective closest point on an existing contour is determined. If they belong to the same contour, the section in-between is replaced with the new contour. Similarly, open contours can be extended and shortened.

### 3.5 Unified Projection Matrix

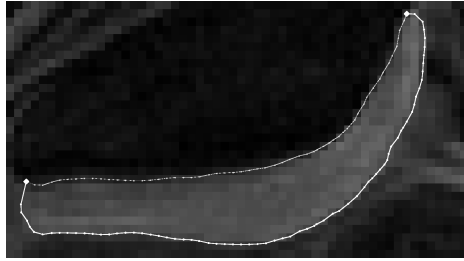
Standard OpenGL projection matrices are limited either to orthographic projection or perspective projection. We therefore propose a unified projection matrix depending on a parameter  $p$  to control perspective foreshortening and on parameters  $d_x, d_y$  to control perspective displacement (Fig. 3.5). If  $p = 0$ , the unified projection matrix is of the same shape as an orthographic projection matrix. If  $p > 0$ , perspective foreshortening is enabled and the projection becomes perspective.

$$\begin{array}{ccc}
 \text{(a)} \begin{bmatrix} * & 0 & 0 & * \\ 0 & * & 0 & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & 1 \end{bmatrix} & 
 \text{(b)} \begin{bmatrix} * & 0 & * & 0 \\ 0 & * & * & 0 \\ 0 & 0 & * & * \\ 0 & 0 & 1 & 0 \end{bmatrix} & 
 \text{(c)} \begin{bmatrix} * & 0 & d_x & 0 \\ 0 & * & d_y & 0 \\ 0 & 0 & * & * \\ 0 & 0 & -p & 1 \end{bmatrix}
 \end{array}$$

**Fig. 3.** Standard OpenGL orthographic projection matrix (a) and perspective projection matrix (b) compared to unified projection matrix (c).

## 4 Results

Tools and algorithms were implemented in C++ using a Qt-based User Interface and OpenGL as graphics engine. We segmented a set of MR images of healthy patellae, which were acquired using a FLASH sequence.



**Fig. 4.** Segmentation of the patellar cartilage in one slice with sub-pixel accuracy. The contour fragments show the cartilage bone interface (gray, top) and cartilage surface (white, bottom) respectively. The nodes in the contour indicate the edge points that were determined in the preprocessing step.

Figure 4 shows the result of the segmentation of the patellar cartilage in one slice.

Semi-automatic tracing and propagation of contours turned out to be highly effective on the cartilage bone interface. This technique was also successfully conducted on the cartilage surface in some cases, whereas in cases like the one in fig. 2 manual segmentation was required.

## 5 Discussion

We introduced a semi-automatic segmentation system for patellar cartilage extraction. Its slice-by-slice approach corresponds to the way radiologists are used to look at anatomy and thus enables radiologists to directly transfer their day-to-day experience in the segmentation process without any relevant training period. The focus of the system lies on user-friendliness instead of purely automatic procedures. We believe this to be more vital for a speed-up in segmentation, which is expected by clinicians to save 50 % of the time used on the segmentation task nowadays, using fully manual tools.

Integration of tools for volumetric measurements and registration will continue this work in the future.

## References

1. Stammberger, T.: Entwicklung von Bildverarbeitungsmethoden zur quantitativen Analyse des Gelenkknorpels in der Magnetresonanztomographie. PhD thesis, Institut für Radiologische Diagnostik der Ludwig-Maximilians-Universität München (1999)
2. Pirnog, C.D.: Articular Cartilage Segmentation and Tracking in Sequential MR Images of the Knee. PhD thesis, ETH Zürich (2005)
3. Cohen, Z., McCarthy, D., Kwak, S., Legrand, P., Fogarasi, F., Ciaccio, E., Ateshian, G.: Knee cartilage topography, thickness, and contact areas from MRI: in-vitro calibration and in-vivo measurements. *Osteoarthritis Cartilage* **7** (1999) 95–109
4. Haralick, R.M., Shapiro, L.G.: The Facet Model. In: *Computer and robot vision*. Volume 1. Addison-Wesley, Reading, Massachusetts, USA (1992)