AUTOMATIC DETECTION AND SEGMENTATION OF SHOULDER IMPLANTS IN X-RAY IMAGES

A thesis presented to the faculty of
San Francisco State University
In partial fulfilment of
The Requirements for
The Degree

Master of Science
In
Computer Science

by

Maya Belen Cervantes Gautschi Stark

San Francisco, California

May 2018

CERTIFICATION OF APPROVAL

I certify that I have read *AUTOMATIC DETECTION AND SEGMENTATION OF SHOULDER IMPLANTS IN X-RAY IMAGES* by Maya Belen Cervantes Gautschi Stark and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirements for the degree: Master of Science in Computer Science at San Francisco State University.

Kazunori Okada
Associate Professor of Computer Science

Dragutin Petkovic
Professor of Computer Science

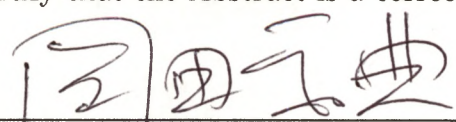
Hui Yang
Associate Professor of Computer Science

AUTOMATIC DETECTION AND SEGMENTATION OF SHOULDER IMPLANTS IN X-RAY IMAGES

Maya Belen Cervantes Gautschi Stark
San Francisco State University
2018

The procedures for repairing or replacing prostheses used in total shoulder arthroplasty (TSA) vary depending on the particular model of the prosthesis. If the model of the prosthesis is unknown, identification is performed by medical professionals based on visual inspection of X-ray images. This process is tedious and time consuming; indicating an unmet need for a tool that will aid surgeons in identifying prostheses quickly and accurately. A preliminary step towards the creation of such a classification tool is the segmentation of the prosthesis. This thesis describes the design and implementation of a software solution to the problem of detection and segmentation of TSA implants in X-ray images. The method implemented uses the Hough transform for circles to locate the implant, followed by segmentation using a seeded region growing method. Validation is performed by comparison with manually segmented ground-truth images, and by visual inspection of the results.

I certify that the Abstract is a correct representation of the content of this thesis.



Chair, Thesis Committee

1/31/2018

Date

ACKNOWLEDGMENTS

Many, many thanks to Professor Okada for his guidance and patience throughout this project. Thank you to Professors Petkovic and Yang for their support and insightful feedback. Thank you to Jen Schwartz for help navigating requirements. Thank you to Poulomi Das for all the help with data collection, analysis, and manual segmentation. Thank you to Nao Funada for help with manual segmentation. Thank you to everyone in the BIDAL group for the feedback and suggestions, and in particular to Jeff Hung for image processing advice, and to Octavian Druela for recommending OpenCV and its implementation of the Hough circle transform for this project. Thank you to Barbara and Peter Cervantes-Gautschi and Isa Gautschi for the encouragement and support. Most of all, to my husband, Tim Stark, thank you so much for everything, I could not have done this without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Total shoulder arthroplasty (TSA) is the surgical replacement of the shoulder joint with an implanted prosthesis. Patients with shoulder prostheses as a result of total shoulder arthroplasty may later require additional surgery to repair or replace the implant. Different prosthesis models require different procedures, and sometimes different equipment, for repair and removal. Patients and surgeons may not know the model or manufacturer of the original prosthesis. This is often the case if the original surgery was performed many years prior to the follow-up surgery, or if the original surgery was performed internationally. Currently, identification is performed based on visual inspection and comparison of X-ray images by medical professionals (see Figure 1.1). This type of workflow is both tedious and time consuming; indicating an unmet need for a tool that will aid surgeons in identifying unknown prostheses

quickly and accurately. This tool would be a system that classifies, by model and manufacturer, TSA prostheses in X-ray images.

In order to build such a classification system, image data with which to train and test the system must be collected and preprocessed. High-resolution digital images are generally considered preferable for image analysis; however, requiring such images for our system may present challenges that negatively affect workflow for potential future users. Ideally, surgeons would be able to use a picture of the X-ray taken with a smartphone camera rather than a large image file that may include sensitive patient information. This would enable them to avoid potential privacy concerns by disassociating the image from digitally stored patient information. Additionally, it would allow surgeons who do not use or have access to high-resolution digital X-ray images to utilize the system.

The resulting system would also need to detect and segment the images prior to classification, ideally with minimal interaction required from the user. Detection and segmentation of the prostheses to be classified are preliminary steps towards the creation of such a classification tool.

1.2 Problem Definition

The end goal of the project is the automatic classification, by model and manufacturer, of shoulder prostheses used in TSA. The goal of the portion of the project proposed in this thesis is the implementation of two of the components of this system:

Figure 1.1: Example of a TSA implant in an X-ray image

the automatic detection and segmentation of TSA implants in X-ray images implemented as a software solution. Given an X-ray image containing such an implant, the prosthesis should be detected with minimal input required from the user. Once the prosthesis has been detected within the X-ray image, it should be automatically segmented sufficiently to be used as input to the classifier.

1.3 Requirements and Constraints

The requirements and constraints for this project are not perfect detection and segmentation. Each of these elements is itself a preprocess; the implant detection phase is a preprocess for the segmentation phase, which itself was designed as a preprocess for a further classification component. Rather, the goal is for each of the components to maximize the performance of the following component in the system. For example, our approach targets only the upper portion of the implant for segmentation, rather than the entire prosthesis. This region is comprised of the

head and proximal end of the stem component, referred to as the body. The body roughly corresponds to the third of the stem closest to the head. The reasons for targeting this region are that the majority of the identifying features of each model are contained in the head and body. The remaining portion of the stem, beginning directly below the body and extending down to the distal end, is referred to here as the distal portion of the stem. The distal portion of the stems offer less information and are often significantly more difficult to segment accurately. For detection of the implant, the goal is to provide a usable mask and seed for the segmentation phase. In order to maintain the usability of our system in a future workflow, the system should be able to accomplish the above using potentially low-quality images.

1.4 Proposed Approach

In this section, we will define the detection and segmentation phases. The overall flow of the system is shown in Figure 1.2. The input is a digital image of an X-ray of a TSA implant, being either the original image or a captured image (such as one taken by a smartphone camera of the original X-ray image). The detection phase consists of preprocessing, implant detection, mask creation and seed selection. The segmentation phase consists of preprocessing, segmentation, and over-segmentation detection. Preprocessing of the original image for detection and segmentation is performed separately because different approaches are used in each of the two phases. Detection refers to implant detection throughout this document unless otherwise

noted. Both the detection phase and the segmentation phase rely on multi-step cascade methods to determine how the image will be preprocessed prior to detection and segmentation. These methods are referred to briefly below, and are described in detail in Chapters 4 and 5.

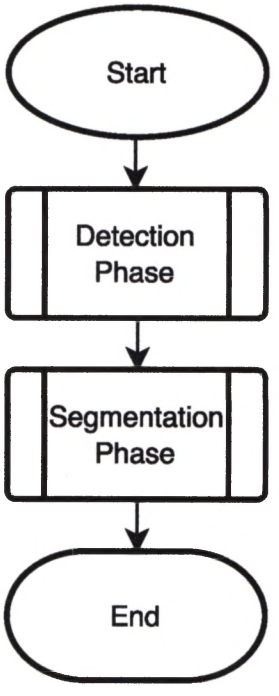


Figure 1.2: High-level system flowchart.

1.4.1 Overview of Detection Phase

The preprocessing step in the detection phase smoothes the image using either the bilateral or median blur filter [9] [3]. The process of choosing which particular filter

will be used is discussed in Chapter 4. The smoothed image is then passed to the implant detection step, which uses the Hough transform for circles to detect the head of the implant [5] [11] [4]. This step provides the location of a circular region of interest containing the head and body of the implant. If no circle is returned by the first attempt at detection, adjustments are made to the smoothing filter in the preprocessing step, and detection is attempted again. The preprocessing and detection steps may be repeated up to a total of three times (initial attempt, possibly followed by one or two further attempts).

In the mask creation step, a mask is created by enlarging the radius of the circle found in the implant detection step. This mask is used during the segmentation phase to isolate the head and body of the implant by excluding the rest of the image from the segmentation process. The center of the circle used to create the mask is used as the seed point for region growing in the segmentation phase. The criteria used to evaluate the results of the detection phase are described in detail in Section 6.3.1, but may be summarized as "correct" results and "usable" results. Detection of the implant is considered "correct" when the detected circle lies along the curve of the head of the implant, referred to as a *Correct-Circle*, and the mask returned completely contains the implant region of interest, referred to as a *Usable-Circle*. A "usable" result does not require correct detection, so long as the result meets the criteria for a *Usable-Circle*, and the seed lies within the implant in the region of interest, referred to as a *Usable-Seed*.

1.4.2 Overview of Segmentation Phase

The segmentation phase begins by preprocessing the original image by smoothing using the bilateral filter. The smoothed image is then passed to the segmentation step, which is performed using seeded region growing, and described in more detail in Section 5.2. The segmentation result is then evaluated for over-segmentation. If over-segmentation detection determines that the image is not likely to have been over-segmented, the segmentation result is returned. Otherwise, the original image undergoes a different preprocessing approach prior to a second attempt at segmentation. The result of this segmentation is returned without being further evaluated. The output of the segmentation phase is a binary image of the segmented head and body of the prosthesis. Overlap with manually segmented ground-truth images is used to judge the segmentation performance quantitatively, combined with qualitative performance evaluation by specific comparison with ground-truth images to evaluate application specific performance.

1.5 Justification for Chosen Solution

Upon inspection of the images, it became apparent that information gathered from the distal portion of the stem would be difficult to use for classification. This is due primarily to two contributing factors. First, because the stems for each distinct model come in different sizes in order to fit different people, there is a considerable

in-class variation of measurable characteristics of the stems, such as length and width. For the majority of models, this portion of the stem lacks characteristics detectable in X-ray images that either distinguish it from other model classes or associate it with other members of its own class. These tendencies indicate that there is likely to be more in-class variation than between-class variation regarding the distal end of the stem. Although the heads also come in different sizes, detectable characteristics tend to be consistent across sizes; for instance, a distinctive shape, or the appearance of an exposed rim where it meets the stem component. The second factor has to do with a tendency for the distal end of the stem to blend in with the bone in the X-rays. In some cases, the contrast was so low that even precise manual segmentation was impossible in this area, as shown in Figure 1.3. These observations led to the decision to perform segmentation only on the region surrounding the head and proximal region of the stem.

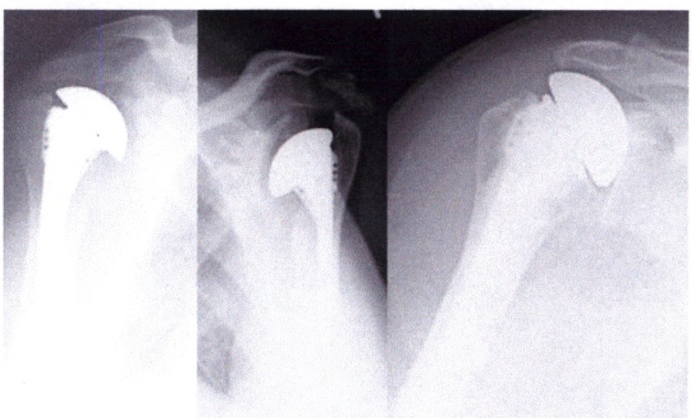


Figure 1.3: TSA implant X-rays with very low contrast between stem and bone.

Another challenge posed by the dataset is that the vast majority of the 605 images are thumbnails collected from the University of Washington Shoulder Site [10], and therefore with significantly low resolution (the longest dimension for most images in the dataset is 250 pixels). Since many of the images already had problematic areas of low contrast, the low resolution increased the difficulty of the problem.

The detection and segmentation approaches proposed in this thesis were chosen as the result of extensive experimental evaluation involving the comparison of several potential variations of our algorithms. The performance of each of these components was assessed individually in addition to analyzing the performance of the overall system. The experimental evaluation process we used to develop our proposed solution is described in Chapter 6.

1.6 Contributions

The main contribution of this work is the creation of a tool that can be used as a preprocessing step prior to implementing and training a classifier to identify the manufacturer and model of the prosthesis. Each component of the system was thoroughly evaluated in order to determine the potential strengths and drawbacks of the proposed approach. The 605 manually segmented ground-truth images can be annotated and used to train the classifier. The automatic detection and segmentation are designed to be used as preprocessing steps in the overall classification system, allowing user input to be limited to providing the X-ray image.

1.7 Outline

This thesis is organized into seven chapters. Chapter 2 is a literature survey of related works. The dataset is described in Chapter 3, and the implant detection and segmentation approaches are described in Chapters 4 and 5, respectively. Chapter 6 describes the evaluation metrics, methods, experimental results, followed by a discussion of the experiments and results. The final chapter includes the conclusions drawn from this project and future work not covered by this thesis.

Chapter 2

Related Works

Based on the results of a literature survey, there has been some interest in incorporating digital image analysis to aid in monitoring patients after joint replacement surgery. Although no literature regarding shoulder arthroplasty was found, related early work has been performed for knee and hip arthroplasty implants.

In [1], the authors presented a proof of concept paper for a project with a goal very similar to that of our future classification system. The goal of this project is identification of prosthesis model in X-ray images of knee arthroplasty implants. The authors propose a system comprised of three components: template image generation, X-ray image segmentation, and template matching. It is difficult to analyze the results the authors achieved because the project was still early in the development phase at the time of publication; and a follow-up paper does not yet appear to have been published.

The first component, template image generation, creates 2D templates from 3D

models of the prostheses. The first step adds the 3D models to a database, then from these models acquires 2D images of each model from different angles to create a comprehensive collection of templates for each model. Thresholding is then used to segment the template images. These templates are later used in the final component for matching with segmentations of the X-rays.

In the segmentation component, the X-ray images are smoothed using the median blur filter, followed by an edge-finding step using the Sobel operator. The user then selects a seed and also a color similarity threshold for the next step, a flood-fill algorithm. In the final step of the segmentation component, dilation is performed to close small holes. The authors do not discuss evaluation of segmentation performance and only provide an example of the results of their algorithm for an X-ray that they note is an exemplary image. They state that the segmentation algorithm currently does not always produce good results, and that they intend to experiment with Canny edge detection and region growing in their future work.

The third step is template matching, the performance of which is also difficult to evaluate at this stage because at the time the article was written the authors only had access to one model. The lack of data prevented evaluation of the system's ability to discriminate between different prosthesis models. However, the authors were able to show as a proof of concept that their algorithm could match a manually segmented image to the correct template.

In [7], the authors use standard deviation of error and ROC curves to evaluate

the quality of different segmentation approaches for total hip arthroplasty (THA) implants. Like our proposed approach, the application presented also prioritizes a specific portion of the implant, in this case the stem rather than the head and body. Unlike our proposed approach, unwanted areas are not masked during segmentation.

The shape of the THA prostheses are somewhat similar to those of TSA prostheses, in that they consist of a stem and semi-spherical head. The authors also note the overlap in intensities between some areas of bone and implant. The example illustrating a typical image shows greater difference between the intensities of the stem and the femur than that between the stem and the humerus for most images in our dataset. The overlap in intensities for the head and socket in their illustrative example appears similar to most images in our dataset. The size and characteristics of the dataset used are not provided. In their evaluation, the authors state that the detection rate shown is for prosthesis stem pixels and that the false positive rate is of non-prosthesis pixels with respect to a manually annotated ground-truth. The paper does not state how prosthesis pixels belonging to regions other than the stem are handled in the evaluation of segmentation quality.

The paper compares the results of four histogram-based methods and a feature space-based method. The histogram-based segmentation methods evaluated are the Otsu and Fisher thresholding methods, the Lloyd thresholding method, the Kittler thresholding method, and the Pal thresholding method. The feature space-based method uses the Fuzzy C-means (FCM) clustering method to segment the image by

class into prosthesis, bone, and soft-tissue components.

Of the histogram-based segmentation methods, the Lloyd thresholding method showed the best performance, with a detection rate of 98.33% and a false positive rate of 2.24%. The other three histogram-based methods had false positive rates of 51.54% or greater. The FCM approach achieved a detection rate of 98.29% and a false positive rate of 3.22% for the three-class segmentation. The detection and false positive rates by individual class are not given, these statistics reflect the combined performance results for classification of the prosthesis, bone, and soft-tissue components. Due to the presentation of the results in this manner, segmentation performance for the prosthesis component is unknown for the FCM approach.

The authors conclude that the FCM approach produced higher quality segmentation for all three image components, and do not mention in their conclusion that the Lloyd method achieved better performance by their metrics. It is possible that the quality of the FCM segmentation for the prosthesis component class was higher than the overall performance for the three component classes, but this cannot be determined based on the information provided. The authors appeared to experience many of the same challenges that we did with segmentation due to overlapping intensities between the implant and background features. The illustrative example of the typical FCM segmentation shows a high amount of false positives for segmentation of the head of the prosthesis, which indicates that this approach would not be usable for our dataset since no additional qualitative measures of the segmentation

quality are provided.

It appears that segmentation of arthroplasty implants in X-ray images faces similar challenges regardless of the joint in question. The tendency of the implant area to share similar intensity with that of the surrounding bone and tissue areas presents a significant challenge, as does acquiring a large and suitably varied dataset. The approach presented in [1] indicates that classification by model is an achievable final goal. However, they were unable to solve the segmentation problem, which is a necessary preliminary step. One or more of the approaches presented in [7] may offer a viable segmentation solution, but information regarding the performance results is lacking. Furthermore, this approach did not attempt segmentation of the region of interest that we intend to use for classification (i.e., the head and body of the implant).

Chapter 3

Data

This chapter describes the dataset used for the experiments, how it was collected, and particular characteristics of the data that influenced aspects of the proposed algorithms. The generation of our ground-truth images is also described here.

The dataset consists of 605 JPEG images, which were primarily collected between June 2015-September 2015 from [10] on the University of Washington Shoulder Site, a reference site which contains thumbnails of collected TSA X-ray images. A small number of images were acquired from individual surgeons and manufacturers. Any information that could be identified with any individual patient was removed prior to use. Each of the images contains an X-ray of a TSA implant. The X-rays are taken from a variety of angles, as shown in Figure 3.1. The majority of the images are low-resolution thumbnails, although a few are higher resolution.

The dataset contains images of 16 different implant models by four different manufacturers: Cofield, Depuy, Tornier, and Zimmer. The manufacturers were

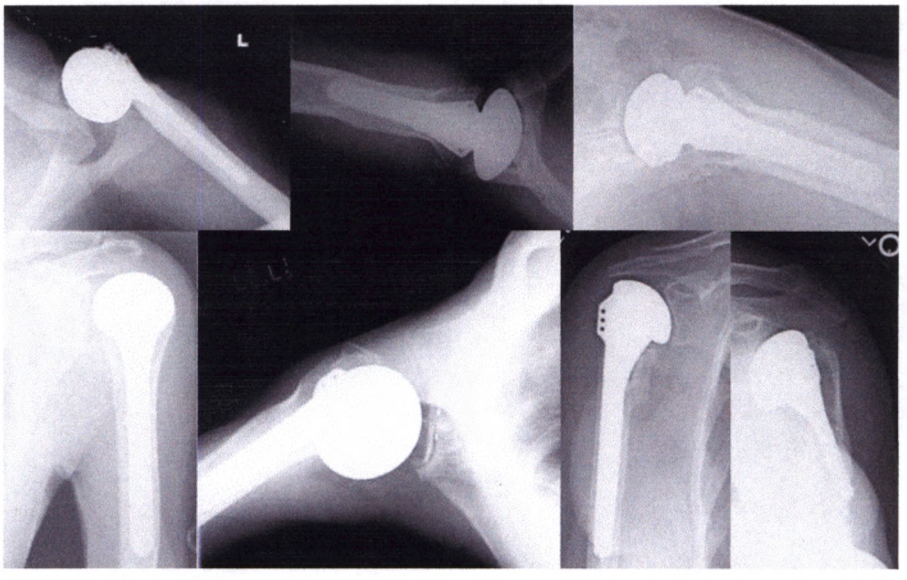


Figure 3.1: TSA implants from different angles in X-ray images.

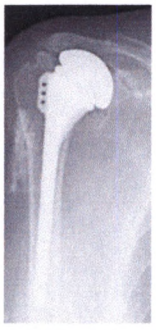
chosen based on their inclusion in a list of common manufacturers provided by Dr. Brian Feeley of the UCSF Department of Orthopaedic Surgery. The contents of the dataset are described in Table 3.1, and exemplary X-ray images of each are model are shown in Figures 3.3, 3.4, 3.5, and 3.6. These particular images are not necessarily representative of typical images in the dataset, but rather were selected because they best illustrate observable characteristics of each model, such as holes and fins. These characteristics are not detectable in all images, nor are the implants themselves always as easily discernible from the background.

The collected images were recorded, sorted, and stored according to model and manufacturer. After collection, the images of each prosthesis model to be included

Model	No. of Images in Dataset
Cofield II Total Shoulder System	62
Cofield Total Shoulder System	26
Depuy Global	129
Depuy Global Advantage	53
Depuy Global AP	2
Depuy Global Fracture	61
Depuy Global HRP	50
Tornier Aequalis	35
Tornier Aequalis Cemented	2
Tornier Aequalis Fracture Shoulder	8
Tornier Aequalis Press Fit	26
Zimmer Anatomical	7
Zimmer Anatomical Combined	6
Zimmer Anatomical Shoulder System	15
Zimmer Bigliani-Flatow	87
Zimmer Select Shoulder	36

Table 3.1: Dataset image count for each manufacturer and model.

in the project were analyzed in order to determine which observable characteristics may be used to classify the individual models. Manual segmentation was completed for each image included in the dataset using ITK-SNAP [12] [13]. An example of an X-ray image from the dataset and the resulting manual segmentation is shown in Figure 3.2. The X-ray images were converted from JPEG to raw format prior to being loaded to ITK-SNAP for manual segmentation. These segmentations were performed by using the polygon tool to select the entire implant in the image, creating the basis of the segmentation, and then edited using the paintbrush tool to refine the segmentation. The manual segmentations were then saved as PNG images.



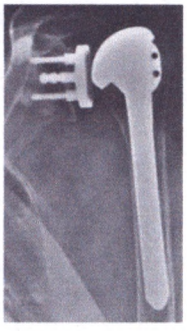
(a) X-ray image of TSA implant.

(b) Manual segmentation.

Figure 3.2: Illustrative example of manual segmentation of a TSA implant in an X-ray image.

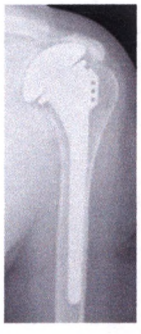
These manual segmentation images make up the set of ground-truth images used to evaluate the segmentation results.

(a) Cofield II Total Shoulder System



(b) Cofield Total Shoulder System

Figure 3.3: Examples of Cofield implant models included in the dataset.

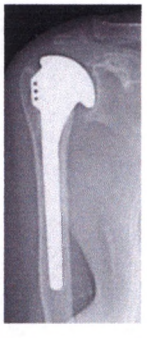


(a) Depuy Global

(b) Depuy Global Advantage

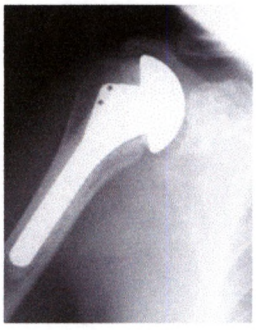
(c) Depuy Global AP

(d) Depuy Global Fracture



(e) Depuy Global HRP

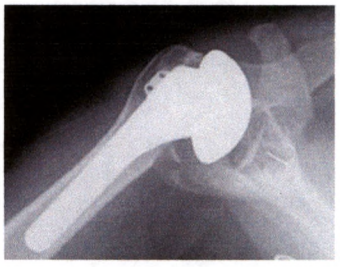
Figure 3.4: Examples of Depuy implant models included in the dataset.



(a) Tornier Aequalis

(b) Tornier Aequalis Cemented

(c) Tornier Aequalis Fracture Shoulder



(d) Tornier Aequalis Press Fit

Figure 3.5: Examples of Tornier implant models included in the dataset.

(a) Zimmer Anatomical

(b) Zimmer Anatomical Combined

(c) Zimmer Anatomical Shoulder System

(d) Zimmer Bigliani-Flatow

(e) Zimmer Select Shoulder

Figure 3.6: Examples of Zimmer implant models included in the dataset.

Chapter 4

Detection of Shoulder Replacement Implants

In this chapter, the design and implementation of the implant detection method is described. In order to minimize the need for interaction with the system by the user, we implemented a component to automatically detect the location of the implant in the image and provide a seed for the seeded region growing method used for segmentation. Depending on the position of the implant in the image, the head of the implant can be approximated by a full or partial circle, as shown in Figure 4.1. We wanted to know if the Hough transform for circles could be used to reliably detect the implant and provide a seed in the form of the center coordinates of the circle, and if so, what combination of preprocessing and parameters would produce the best result. Our proposed approach has three components: preprocessing, implant detection, and mask creation. Section 4.1 briefly describes the overall structure

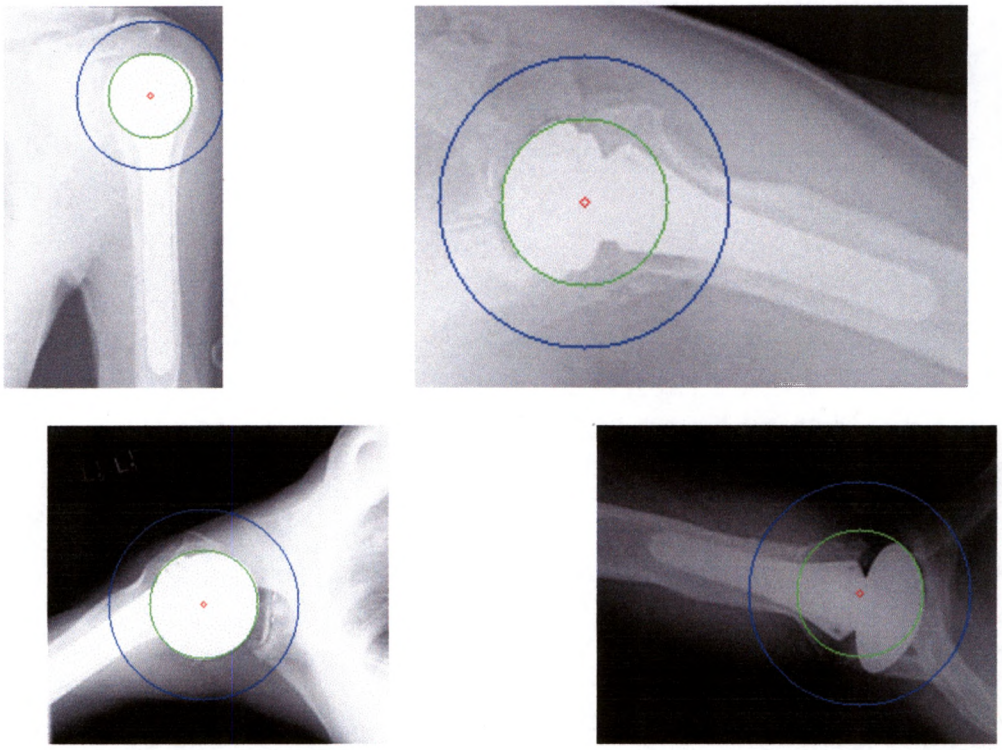


Figure 4.1: Illustrative examples of implants from different angles with detected circle and region of interest shown.

of the detection phase, while the subsequent sections describe each of the three components in greater detail.

4.1 Overview of Detection Phase

As illustrated in Figure 4.2, our detection phase uses a three-step cascade structure. Each of the three steps pairs the implant detection algorithm, referred to hereafter

as *find_circle*, with a different preprocessing approach, referred to in Figure 4.2 as *Detection Preprocesses 1*, *2*, and *3*. If a circle is detected during one of the steps, the subsequent detection attempts are bypassed. If no circle has been detected after the completion of the first preprocessing and detection step, a second attempt is made, this time pairing a different preprocess with the detection algorithm. If still no circle has been detected, a final attempt is made using a third preprocessing approach prior to detection. Once a circle has been detected by one of the first two steps, or the third step has completed (regardless of whether a circle has been detected), the mask is created and returned. When a circle has been detected, the mask covers the image with the exception of a circular region of interest. Otherwise, the mask covers the entire image.

4.2 Preprocessing

There are two different filters used for preprocessing during the detection phase: a bilateral filter [9] and a median blur filter [3]; both are edge-preserving. Our implementation uses OpenCV's *bilateralFilter*() and *medianBlur*() functions, respectively [6] [8]. In the first step, the bilateral filter is used for preprocessing (*Detection Preprocess 1* in Figure 4.2). In the second and third steps, the median blur filter is used for preprocessing, but with two different kernel sizes (*Detection Preprocesses 2* and *3* in Figure 4.2).

The bilateral filter is similar to a Gaussian smoothing filter, which replaces pixel

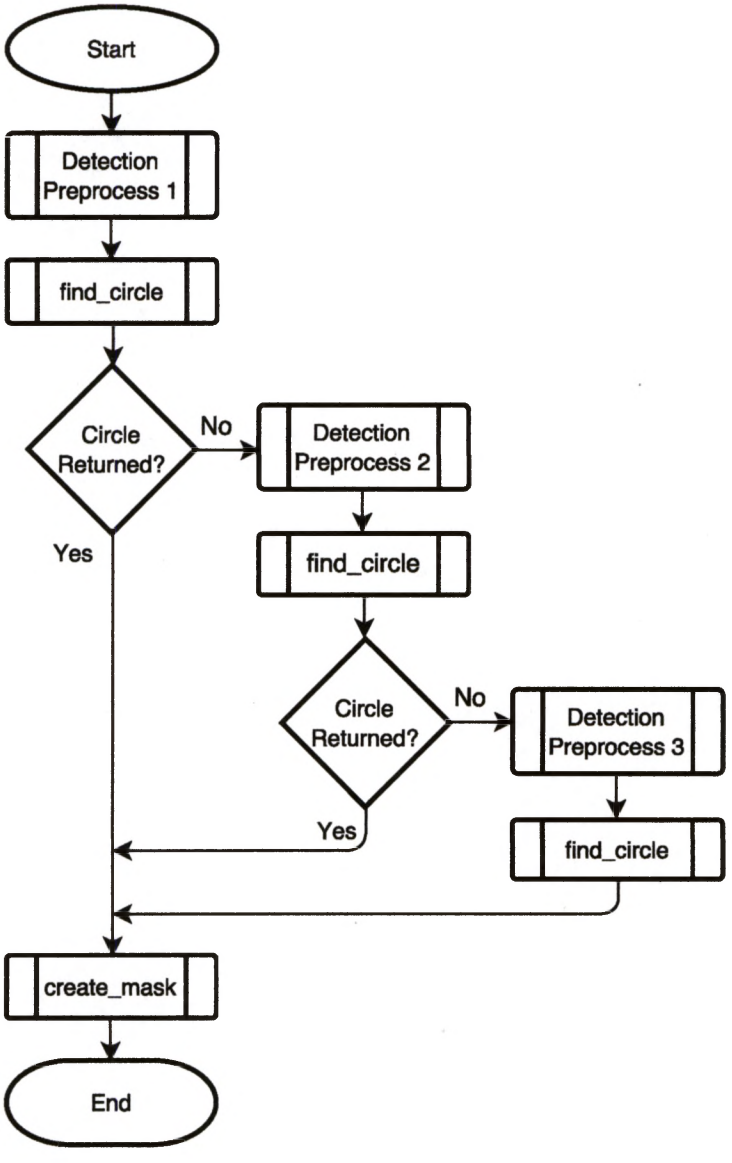


Figure 4.2: Flowchart for the detection phase.

values with the Gaussian weighted average of the pixel's neighborhood. The difference is that in addition to considering spacial relationships between pixels, the bilateral filter also considers similarity in pixel intensity. Only pixels that are close enough in terms of space and intensity value are considered during smoothing, thus preserving edges. The parameters for $bilateralFilter()$ are neighborhood diameter d, $sigmaColor$, and $sigmaSpace$. Larger values in $sigmaColor$ and $sigmaSpace$ allow a higher level of difference for intensities and more distant pixels to be considered for inclusion. The documentation recommends that for real-time applications, a diameter of $d = 5$ be used [8]. For implant detection, our approach uses neighborhood diameter d of 5, and both $sigmaColor$ and $sigmaSpace$ are set to 10. Low sigma values are used because our dataset contains mostly low resolution images, often with poor contrast.

The median blur filter is similar to an averaging filter, differing in that it replaces pixel values with the median, rather than average, intensity value of the pixels in the neighborhood. This means that, unlike an averaging filter, the median blur filter replaces pixel intensity values with intensity values already present in the image. This distinction aids in preserving edges if the size of the neighborhood is appropriately chosen for the image. For *Detection Preprocess 2*, we use a 3x3 kernel for the neighborhood. As with the sigma values used for the bilateral filter, we use a small kernel size for preprocessing during the second step because of the low resolution and contrast of the images in our dataset. For *Detection Preprocess 3*,

we use a 7x7 kernel to catch the implants that are missed by the first 2 steps.

4.3 Implant Detection

In this section, a brief explantation of the *find_circle* algorithm and description of the Hough circle transform is given prior to a detailed description of our implementation. As noted previously, the appearance of the implant heads can be approximated by full or partial circles (see Figure 4.1). Our implant detection algorithm, *find_circle*, makes use of this characteristic by employing the Hough transform for circles to detect the head of the implant. OpenCV's function for the Hough transform for circles, *HoughCircles*(), was used in our implementation [6]. Criteria for success are described in Section 6.3.1, but for the purposes of this section may be thought of as producing a circle that may be used to create a mask that isolates the region of interest and whose center may be used as a seed during the segmentation phase.

The overall structure of *find_circle* is described by the flowchart in Figure 4.3. *find_circle* works by making one or more calls to *HoughCircles*() in order to detect the single circle that is most likely to best approximate the head of the implant in the image. The algorithm makes adjustments to either narrow or broaden the scope of detected circles to be included for consideration based on the results of previous attempts at circle detection. In Figure 4.3, these adjustments and subsequent calls to *HoughCircles*() take place within the *No Circles* and *Multiple Circles* sub-

processes. There are three variables within *find_circle* that are estimated by the algorithm: *min_radius*, *max_radius*, and *acc_th*. The roles of these variables will be discussed in greater detail later in this section, but essentially they are the parameters for the Hough transform that determine the minimum and maximum radii, and the accumulator threshold. After these variables have been set, an initial call is made to *HoughCircles*(). If multiple circles are detected, the results of the call to *HoughCircles*() are passed to the *Multiple Circles* subprocess, which narrows the results down to a single circle. If no circle is detected, the *No Circles* subprocess is used, which loosens the restrictions on circle detection in an attempt to detect the head of the implant. Once a single circle has been identified, the result is returned in the form of the center coordinates and the radius. If it is determined that no circles can be identified using any of the allowed parameters, the origin is returned as the coordinates for the center along with a radius of 0.

4.3.1 Hough Transform

In order to clearly describe our proposed detection algorithm, we will first give a brief description of the Hough transform. The Hough transform for circle detection is an adaptation of the Hough transform for line detection. The Hough transform for line detection as described by Duda and Hart [2] works as follows: An input image is converted to a binary image in which positive values correspond to significant

changes in intensity. The goal is to locate lines in the image described by

$$\rho = x\cos\theta + y\sin\theta, \tag{4.1}$$

using a 2-dimensional array of bins corresponding to each possible pair (ρ, θ) in the image. This array is referred to as the accumulator. At each non-zero pixel (x, y) in the image, (ρ, θ) are calculated, where ρ is the distance from the origin to the point (x, y), and θ is the angle between the x-axis and the line segment beginning at the origin and terminating at the point (x, y). The accumulator bin $[\rho][\theta]$ is then incremented. After all of the pixels have been evaluated in this manner, the accumulator bins for the $[\rho][\theta]$ pairs with the highest values correspond to the lines described by Equation 4.1. An accumulator threshold is used to determine which lines to include in the result.

The Hough transform can be extended to be used for circle detection. The traditional method does so using a 3-dimensional accumulator of possible centers (a, b) and radii, r. These potential centers and radii are represented in the accumulator as $[a][b][r]$. For each non-zero pixel (x, y), increment each accumulator bin corresponding to any pixel that could possibly be the center of the circle described by

$$r^2 = (x - a)^2 + (y - b)^2 \tag{4.2}$$

If the length of the radius is not limited, this involves every point between the

point (x, y) and the edge of the image. The most likely circles are determined by the highest accumulator values, and an accumulator threshold is used to determine the center and radii combinations to include in the result. This implementation can quickly become very computationally expensive and inefficient, so in practice modifications are generally made [5].

HoughCircles() uses the Hough Gradient method [11] [4], which uses the local gradients at each non-zero point and is able to use a 2-dimensional array of possible centers. Although this method works quite well, there are some drawbacks relevant to our implementation. One is that this method is biased towards the selection of large circles over smaller ones. Unlike the traditional Hough circle transform, only one radius may be returned for any center. This means that concentric circles will not be identified, and near-concentric circles whose centers are too close together are sometimes excluded as well. For our dataset, another issue is that the implant may be detected by a circle that lies within the implant rather than enclosing it due to a change in intensity caused by the curvature of the implant, as illustrated by Figure 4.4b.

4.3.2 The *find_circle* Algorithm

The *find_circle* algorithm attempts to detect the implant in the image, and adjusts the values of *min_radius*, *max_radius*, and *acc_th* during the *Multiple Circles* and *No Circles* subprocesses in order to detect the circle most likely to correspond to

the location of the head of the implant. The *min_radius* and *max_radius* variables are the minimum and maximum length thresholds for radii for circles returned by *HoughCircles*(). The initial values for *min_radius* and *max_radius* are set to, respectively, (0.2 * *min_dimen*) and (0.5 * *min_dimen*), where *min_dimen* is the minimum dimension of the image (i.e., $min(imageheight, imagewidth)$). The accumulator threshold, *acc_th*, is the lower bound for the accumulator values used by *HoughCircles*() for the candidate centers; it determines the minimum level of confidence the detection algorithm must have that each result returned is a true circle. The lower it is, the greater the chances of false circles being returned; the higher it is, the greater the chance of false negatives. For our implementation, the initial value for *acc_th* is set to 30. We use a relatively low value for *acc_th* because the implants are not true circles. The values used for *min_radius*, *max_radius*, and *acc_th* were derived experimentally through a pilot study.

After the variables *min_radius*, *max_radius*, and *acc_th* have been initialized, the initial call to *HoughCircles*() is made. There are three possible outcomes: a single circle is returned, no circles are returned, or multiple circles are returned. If only a single circle has been detected, the center coordinates and radius of that circle are returned and the implant detection component of the detection phase is complete. If no circles are detected, the *No Circles* subprocess is activated. If more than one circle is detected, the *Multiple Circles* subprocess is activated.

The *No Circles* subprocess, illustrated in Figure 4.5, attempts to locate the im-

plant by lowering the values of *min_radius* and *acc_th* to allow more candidate circles for consideration by *HoughCircles*(). First, *min_radius* is decreased a single time, from (0.2 * *min_dimen*) to (0.15 * *min_dimen*), followed by one or more adjustments to *acc_th*. The adjustments to *acc_th* are carried out by decrementing *acc_th* by a particular value, then calling *HoughCircles*() again with the adjusted parameters. This process is repeated until either *HoughCircles*() returns one or more circles, or *acc_th* reaches the *minimum allowable accumulator threshold*, referred to as *acc_min*. 5 is used as the value for both the *acc_min* and the decrement amount in our implementation; these values were chosen based on experiments described in Chapter 6 and are discussed in more detail there.

If more than one circle has been detected, the *Multiple Circles* subprocess proceeds as shown in Figure 4.6. First, *max_radius* is decreased by 10% of its current value, and *HoughCircles*() is called again. If decreasing the value of *max_radius* does not changes the number of circles returned, *min_radius* is increased by 5% of its current value prior to the next execution of the loop. Otherwise, the loop continues as before without changing the value stored in *min_radius*. This subprocess continues until either a single circle or an empty list is returned.

4.4 Mask Creation

The final component of the detection phase is mask creation. When a circle is detected at the completion of any step during the cascade, any subsequent steps of

the cascade are skipped. A circular mask is then created by increasing the radius of the detected circle while retaining the original center. The amount by which the radius is increased for the mask is dependent on the length of the original radius r in proportion to *min_dimen*. The mask radius, *mask_rad*, is calculated as follows:

$$mask\_rad = \begin{cases} (1.75)\, r & \text{if } r \leq (0.25)\, min\_dimen, \\ (1.25)\, r & \text{if } r \geq (0.4)\, min\_dimen, \\ (1.5)\, r & \text{otherwise} \end{cases} \tag{4.3}$$

The mask and seed are then passed to the segmentation phase.

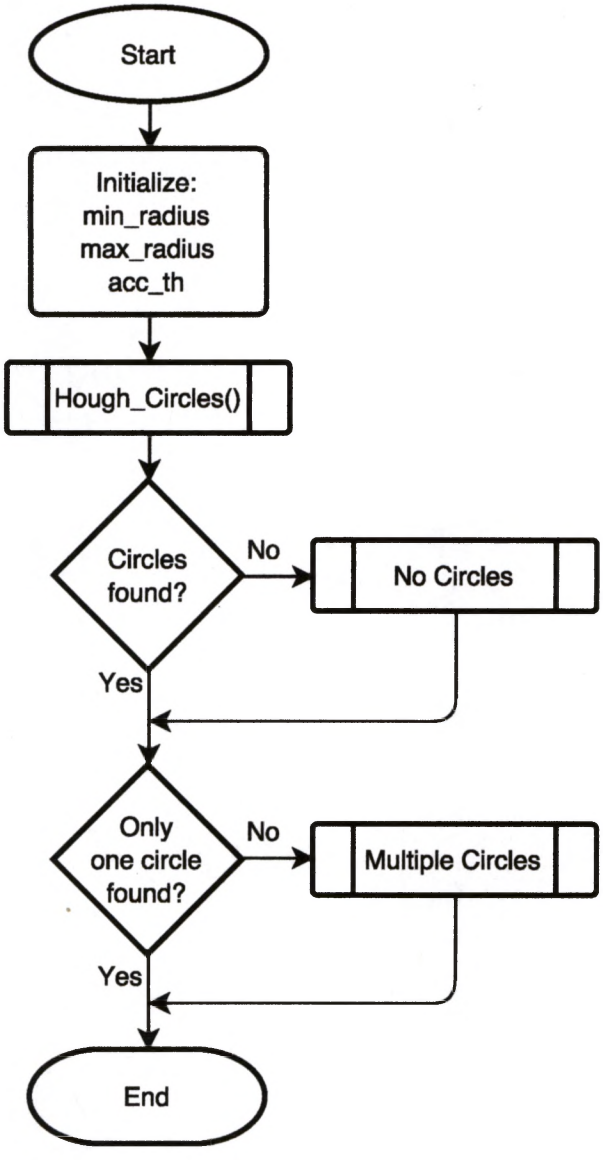


Figure 4.3: Flowchart for the implant detection (*find_circle*) algorithm.

(a) Correct detection of implant.

(b) Implant detected, circle is unusable.

(c) Overly large circle detected.

(d) Incorrect detection of implant.

Figure 4.4: Illustrative examples of detected circle shown in green with region of interest shown in blue.

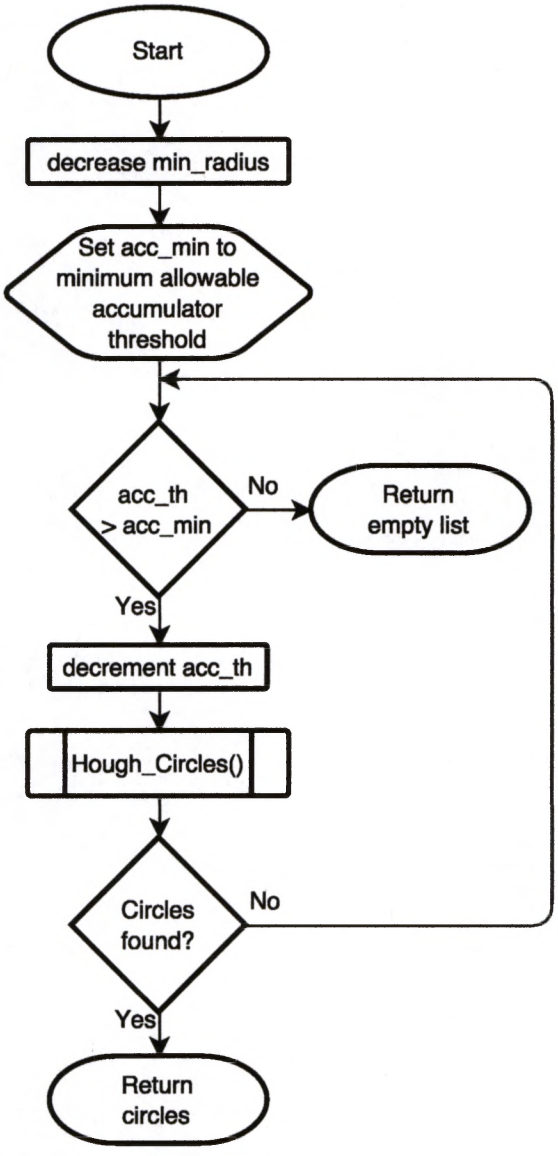


Figure 4.5: Flowchart for the *No Circles* subprocess for implant detection.

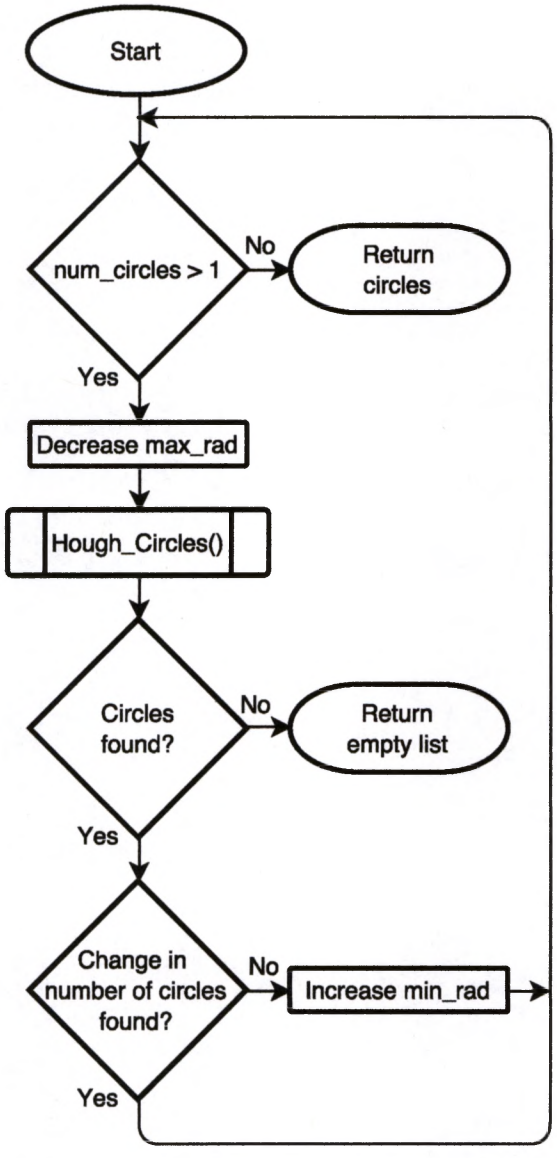


Figure 4.6: Flowchart for the *Multiple Circles* subprocess for implant detection.

Chapter 5

Segmentation of Shoulder Replacement Implants

This chapter discusses the algorithm design of the segmentation phase of our proposed approach. The segmentation phase has three components: preprocessing, over-segmentation detection, and segmentation. The overall structure of the proposed algorithm is described, including the motivation for the particular design choices that were made.

5.1 Overview of Segmentation Approach

The structure of the segmentation approach, described by the flowchart in Figure 5.1, is a two-step cascade, where each of the two steps consists of preprocessing followed by segmentation. After the first step is completed, the resulting segmentation is inspected to determine whether the image is likely to have been over-segmented.

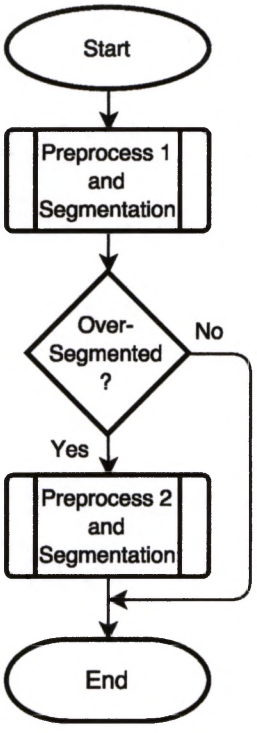


Figure 5.1: Flowchart for the segmentation phase.

If the image is likely to have been over-segmented, the original image is then pre-processed using a different method before segmenting the image again, and the segmentation phase ends. Otherwise, the segmentation phase ends after the first step.

5.2 Seeded Region Growing

The segmentation method itself is a standard seeded region growing approach. The image is preprocessed using one of the two configurations described in the following section. The preprocessed image, along with the mask and seed obtained from the implant detection phase, are then passed to the seeded region growing algorithm (SRG) for segmentation. Beginning at the seed point, the SRG algorithm grows the region using the 4-connected neighbors and the following uniformity predicate:

$$P(R) = \begin{cases} \text{TRUE} & \textit{if} \quad |f(j,k) - f(m,n)| \leq \Delta, \\ \text{FALSE} & \textit{otherwise} \end{cases} \tag{5.1}$$

where $\Delta = 5$. The result is a binary image of the segmented region of interest as determined by the mask (see Figure 5.2).

5.3 Preprocessing and Detection of Over-Segmentation

Over-segmentation is an issue for this algorithm and dataset. Figure 5.3 illustrates typical instances of over-segmentation. Visual inspection of results during preliminary testing showed more desirable segmentation results when smoothing filters were used for preprocessing. Preprocessing the images with a smoothing filter reduced under-segmentation that would otherwise have resulted an inaccurate depiction of the shape of the implant, or in the loss of key features, such as holes and

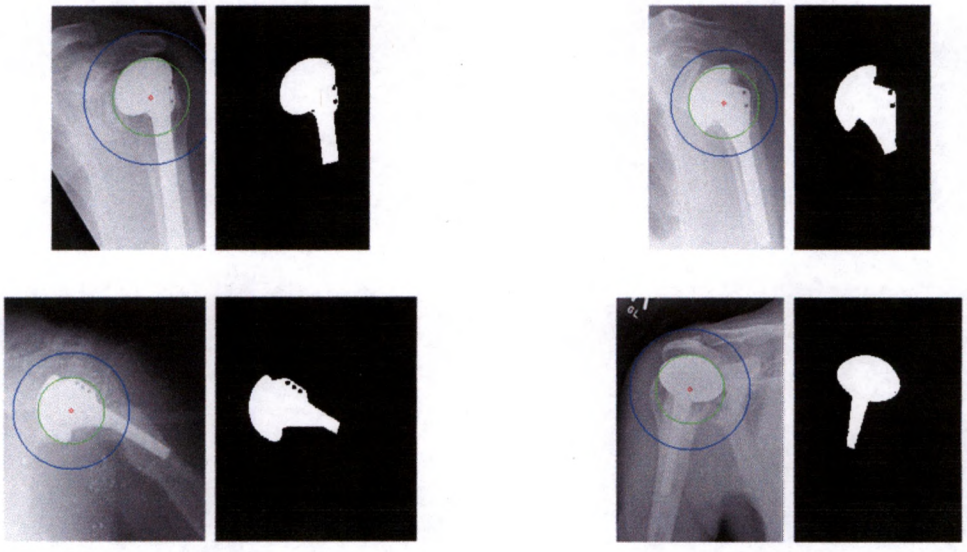


Figure 5.2: Illustrative examples of successful segmentation results.

fins. However, many of the images were over-smoothed, resulting in an increase in over-segmentation, posing a trade-off. Quantitative evaluation of the results also suggested smoothing would be beneficial as a preprocess for our segmentation algorithm. In order to best balance the problem of improving accuracy while reducing over-segmentation, it was necessary to compare different smoothing methods. The mean shift filter and the bilateral filter were found to best address these challenges through experimental evaluation of different preprocessing approaches (described in Section 6.4.3). The evaluation metrics used here are based on overlap with manually segmented ground-truth images and are described in Chapter 6.

Preprocessing the images with the mean shift filter achieves better overall quan-

titative results (see Table 6.13 and Figure 6.10e) and is slightly less prone to over-segmentation. Like the bilateral filter, the mean shift filter considers pixel value in addition spacial proximity during smoothing. The allowable difference in color value is referred to as the color distance. Figure 5.4c illustrates how the overall shape of the implant is retained when preprocessed using the mean shift filter, at the expense of noticeable under-segmentation. Although it is more prone to over-segmentation than the mean shift filter, for images in which over-segmentation is not an issue the bilateral filter produces better results in that finer details, such as holes and fins, are more likely to be preserved. Figure 5.4 illustrates the difference in segmentation results using the different smoothing methods.

To exploit the advantages of both preprocessing methods while balancing their tradeoffs, the segmentation phase uses a two-step cascade approach in which the method of preprocessing the image prior to segmentation is dependent on the likelihood that the image has been over-segmented. The proposed algorithm uses the segmentation result produced by the first step (*Preprocess 1* and *Segmentation* in Figure 5.1) to make a prediction of whether the image has been over-segmented. We treat the image to be over-segmented when the ratio of the pixels within the region of interest that were predicted positive (pp) to the total number of pixels within the

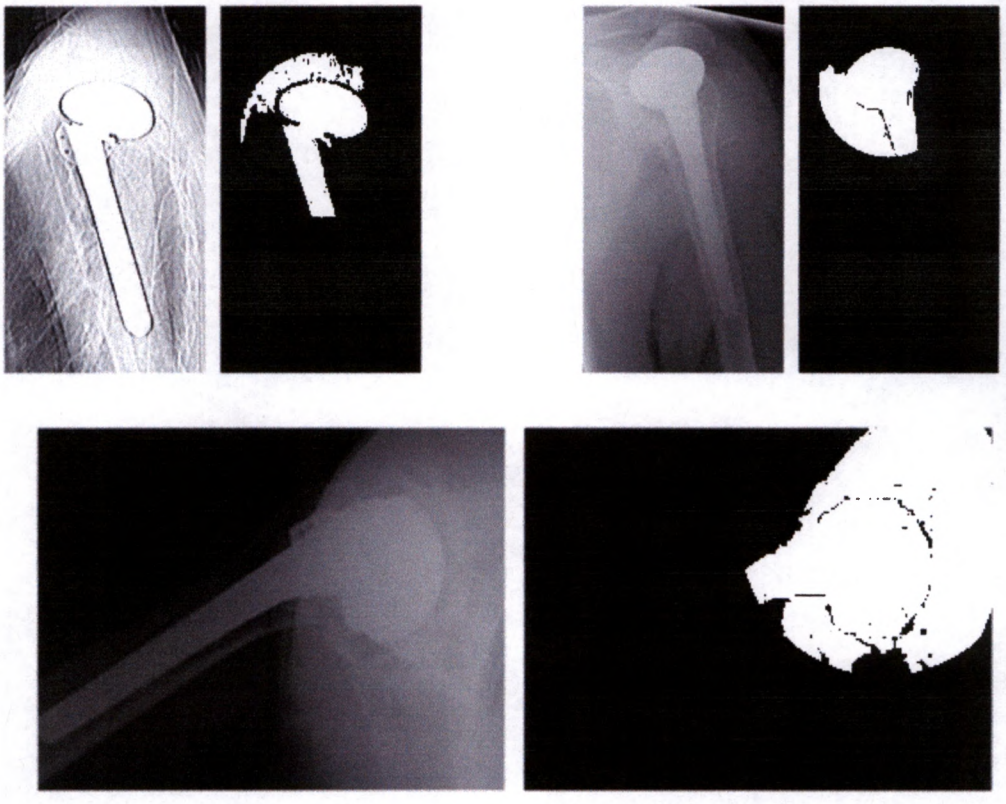


Figure 5.3: Illustrative examples of over-segmented results.

region of interest (*total*), becomes more than 50%.

$$over\_seg = \begin{cases} \text{TRUE} & \text{if } \frac{pp}{total} > 0.5, \\ \text{FALSE} & \text{otherwise} \end{cases} \tag{5.2}$$

If $over\_seg = TRUE$, the second step is activated and a second attempt at segmen-

(a) Manual ground-truth. (b) Bilateral filtering. (c) Mean shift filtering.

Figure 5.4: Illustrative examples of segmentation results with different preprocessing methods.

tation is made using a different preprocessing approach (*Preprocess 2* and *Segmentation* in Figure 5.1), the result is the final segmentation. If $over\_seg = FALSE$, we complete the process as no over-segmentation is detected, resulting in the final segmentation.

Both steps use the same seeded region growing method described in 5.2; the difference between the two is the way the image is preprocessed. *Preprocess 1* represents the bilateral filter in 5.1. In the first step, the image is smoothed using the bilateral filter prior to seeded region growing, as in the implant detection phase. The image is smoothed to a lesser degree for segmentation than for implant detection to preserve features as much as possible, a value of 5 is used for both *sigmaColor* and *sigmaSpace*. After the image has been segmented by the initial step, the segmentation results are evaluated to determine whether the image is likely to have been

over-segmented. If so, histogram equalization is performed on the original image prior to smoothing with the mean shift filter using a spatial radius of 5 and a color distance of 10. Otherwise, the process terminates, resulting in the final segmentation. The successive preprocessing steps of histogram equalization and mean shift filtering are represented by *Preprocess 2* in 5.1. After undergoing the second preprocessing method, the image is again passed to the seeded region growing method for segmentation. The completion of the second preprocessing and segmentation step ends the segmentation phase; the result is the final segmentation.

Chapter 6

Experimental Evaluation

This chapter describes the experimental design used to evaluate the proposed approach to detection and segmentation of implants in X-ray images of total shoulder arthroplasty. Evaluation of each phase will be addressed, as well as evaluation of the overall system. The chapter concludes with a discussion of the results of the experimental evaluation.

6.1 Overview

The overall system is a successive combination of the detection and segmentation phases, and each employs a multi-step cascade algorithm. Our evaluation approach seeks to address this by considering the performance of the individual components within the detection and segmentation phases, as well as the performance of the overall system.

The procedure used for the experimental evaluation of implant detection was

done by visual inspection of each result to determine success. Visual inspection is labor-intensive and time-consuming, which made it impractical to perform an exhaustive grid search of all factors considered in our study. Instead, we created and tested multiple variations of our algorithm by making changes to address failures observed during evaluation of the previous iterations of the design. Evaluation metrics for the detection phase are described in Section 6.3.1.

For segmentation, the performance of different preprocessing/segmentation combinations were tested individually, then compared with the performance of the two-step cascade approach. The proposed methods of detecting likely over-segmentation were evaluated in comparison with a ground-truth based on the results of the first preprocessing/segmentation combination in the cascade. Evaluation metrics and specifics of ground-truth for segmentation and over-segmentation detection are described in Sections 6.4.1 and 6.4.2, respectively.

6.2 Evaluation Metrics

We evaluate the performance of implant detection, over-segmentation detection, and segmentation separately prior to evaluation of the overall system. Precision, sensitivity, F-measure, and Jaccard index were evaluated for all three. Specificity and accuracy are only evaluated for over-segmentation detection and segmentation. Specificity is not useful in evaluating implant detection because the dataset contains no true negatives for that component, i.e., all images contain an implant. For the

same reason, accuracy is made redundant by Jaccard index for datasets with no true negatives.

Precision is the ratio of true positives to predicted positives and is defined

$$\frac{true\ positives}{(true\ positives + false\ positives)}. \tag{6.1}$$

Recall is the ratio of true positives to actual positives and is defined

$$\frac{true\ positives}{(true\ positives + false\ negatives)}. \tag{6.2}$$

Sensitivity is also defined by Formula 6.2. Under this definition, both sensitivity and recall are equivalent to the true positive rate (TPR). In this context, sensitivity and recall refer to the same measure; however, sensitivity is the term that will be used throughout this thesis with the exception of defining F-measure.

Specificity is the ratio of true negatives to actual negatives and is defined

$$\frac{true\ negatives}{(true\ negatives + false\ positives)}. \tag{6.3}$$

F-measure is the harmonic mean between precision and recall and is defined

$$2 * \frac{(precision * recall)}{(precision + recall)}. \tag{6.4}$$

As noted previously, sensitivity is equivalent to recall as defined by Formula 6.2

and may therefore be used to calculate F-measure.

The Jaccard index is used as a measure of overlap between the predicted outcome and ground-truth and is defined

$$\frac{true\ positives}{(true\ positives + false\ positives + false\ negatives)}. \tag{6.5}$$

Accuracy is the ratio of correctly predicted outcomes to all predicted outcomes and is defined

$$\frac{(true\ positives + true\ negatives)}{(true\ positives + true\ negatives + false\ positives + false\ negatives)}. \tag{6.6}$$

6.3 Implant Detection

This section describes the experimental process and methods of evaluation we used to develop and analyze the implant detection phase. As previously mentioned, design choices were made based on evaluation of pervious iterations of the algorithm. In this section we will describe the versions of the detection algorithm that meaningfully contributed to the final algorithm by influencing decisions such as the smoothing filters chosen, the number of steps in the cascade, and adjustments to the values used within the algorithm itself.

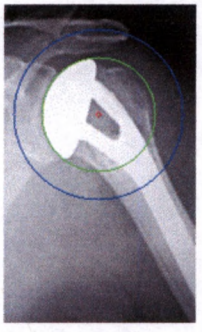
6.3.1 Implant Detection Evaluation Methods

For implant detection, results of the selected method were evaluated in four different categories: *Correct-Detection*, *Usable-Result*, *Usable-Seed*, and *Correct-Usable*. Each category is a combination of one or more of the following three criteria: *Correct-Circle*, *Usable-Circle*, *Usable-Seed*. *Correct-Circle* is defined as one which lies along the curve of the head of implant, indicating that circle detection worked as intended. *Usable-Circle* is defined as a circle which produces a mask that completely contains the implant region of interest (the head and body), regardless of the usability of the seed, or correctness of the circle. *Usable-Seed* is defined as a circle center which lies within the implant region of interest, regardless of the correctness or usability of the circle. It should be noted that circle and seed usability do not necessarily correspond with an ideal result, but simply indicates that the result met at least

minimum requirements for use by the segmentation phase.

The criteria for a true positive result for the four evaluation categories are:

- *Correct-Detection*: *Correct-Circle*, *Usable-Circle* (regardless of usability of seed) (see Figure 6.1a)
- *Usable-Result*: *Usable-Circle*, *Usable-Seed* (regardless of correct detection of circle) (see Figure 6.1b)
- *Usable-Seed*: *Usable-Seed* (regardless of correct detection or usability of circle) (see Figure 6.1c)
- *Correct-Usable*: *Correct-Circle*, *Usable-Circle*, *Usable-Seed* (see Figure 6.1d)



(a) Correct-Detection

(b) Usable-Result

(c) Usable-Seed

(d) Correct-Usable

Figure 6.1: Illustrative examples of criteria for evaluation of implant detection performance.

(a) shows *Correct-Detection* with *Correct-Circle*, *Usable-Circle*, unusable seed. (b) shows *Usable-Result* with *Usable-Circle*, *Usable-Seed*, incorrect detection of circle. (c) shows *Usable-Seed* with incorrect and unusable circle. (d) shows *Correct-Usable* with *Correct-Circle*, *Usable-Circle*, *Usable-Seed*.

6.3.2 Single-Step Detection Algorithm Experiments

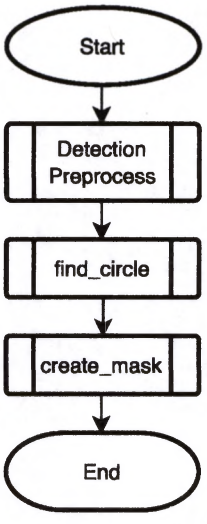


Figure 6.2: Flowchart for single-step version of the detection phase.

One of the goals of the experimental study was to choose an effective smoothing filter to use for preprocessing prior to circle detection, and early versions of our algorithm used a single preprocessing step rather than a cascade. Figure 6.2 shows the structure of these versions. The structure of the *find_circle*() component is the same as that described in Chapter 4 and illustrated by Figure 4.3.

A prominent characteristic of this dataset is that many images have very low contrast between the head of the implant and the background areas. In some cases this is true only on one side of the implant, in others the entire region of interest is affected, as shown by Figure 6.3a and 6.3b, respectively. This tendency towards "washed out" images implied an edge-preserving filter would be preferable, partic-

(a) (b)

Figure 6.3: Illustrative examples of low-contrast TSA implant X-ray images.

ularly since *HoughCircles*() implements edge detection as a preprocess to circle detection. We evaluated the performance of our implant detection algorithm using three different smoothing filters as preprocessing methods. The three filters were the OpenCV implementations of the median blur, mean shift, and bilateral filters [6]. Early on, the mean shift filter was eliminated from consideration for this preprocessing task when it was outperformed by the median blur filter in a preliminary experiment (see Figure 6.4).

Next we evaluated the results of *find_circle*() when using the bilateral filter for preprocessing. A comparison of these results with those using the median blur filter for preprocessing are shown in Table 6.1 and Figure 6.5. The version using median blur appeared to perform better overall, producing more true positives and less than half as many false negatives as the version using the bilateral filter. However, the version using the bilateral filter produced fewer false positives as well as more

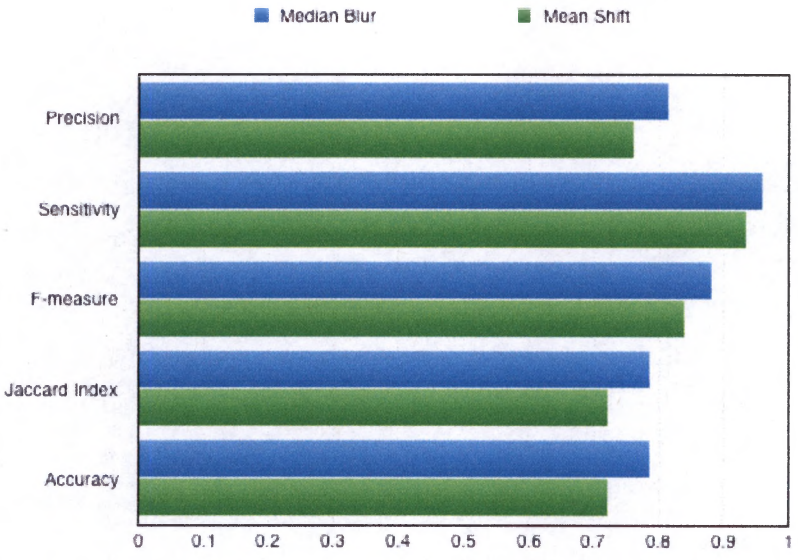


Figure 6.4: *Correct-Detection* evaluation of implant detection results comparing single-step preprocessing methods using the mean shift and median blur filters.

usable results when the correctness of the circle returned is not considered. During evaluation of the results, it was noted that there was little overlap between the false negatives produced by the two versions. The lack of overlap between missed implants suggested that the circle detection approach may benefit from a two-step cascade process in order to maximize the strengths of both preprocessing techniques.

	Filter	Prec.	Sens.	F-meas.	Jaccard Index
Correct-Detection	Median Blur	0.81	**0.97**	**0.88**	**0.79**
	Bilateral	0.81	0.92	0.86	0.76
Usable-Result	Median Blur	0.82	**0.97**	0.89	0.80
	Bilateral	**0.85**	0.93	0.89	0.80
Usable-Seed	Median Blur	0.92	**0.97**	**0.95**	**0.90**
	Bilateral	0.92	0.93	0.93	0.86
Correct-Usable	Median Blur	0.80	**0.97**	**0.87**	**0.78**
	Bilateral	0.80	0.92	0.86	0.75

Table 6.1: Evaluation of implant detection results comparing single-step preprocessing methods using the median blur and bilateral filters.

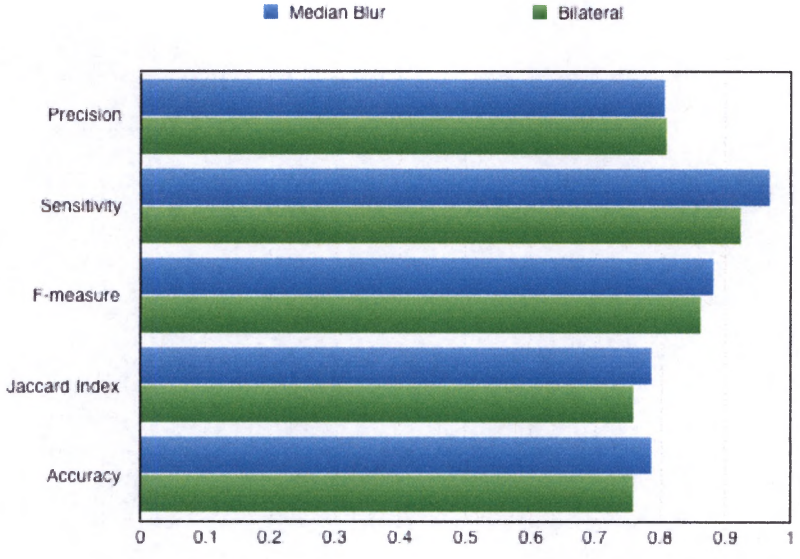


Figure 6.5: *Correct-Detection* evaluation of implant detection results comparing single-step preprocessing methods using the median blur and bilateral filters.

6.3.3 Two-Step Cascade Detection Algorithm Experiments

As illustrated by Figure 6.6, the design of the two-step cascade is similar to the three-step version described in Chapter 4 (see Figure 4.2). The structure of the two-step cascade is as follows: the image is smoothed using the bilateral filter (*Detection Preprocess 1* in Figure 6.6) before being passed to *find_circle*(). If *find_circle*() fails to detect an implant and returns an empty list, the original image is smoothed using the median blur filter (*Detection Preprocess 2* in Figure 6.6) and passed again to *find_circle*(). The bilateral filter was selected as the initial preprocess because the *find_circle*() method produced slightly fewer false positives and more usable results overall when the bilateral filter was used for preprocessing. This suggested that the circles found were slightly more likely to be true positives, or at least usable. False negatives returned by *find_circle*() then have a chance to be corrected by the second attempt using median blur for preprocessing.

In its initial form, the two-step process appears to perform better in some areas in the *Usable-Result* evaluation category than either of the approaches employing a single preprocessing step, with the number of false negatives reduced and an increase to the number of true positives (see Table 6.3). There was little to no improvement shown by using the two-step process when evaluating the other three categories: *Correct-Detection*, *Usable-Seed*, and *Correct-Usable*. In these categories the number of false positives for the two-step approach is higher and the precision slightly lower than for either of the single-step methods (see Tables 6.2 and 6.3).

	Filter	Prec.	Sens.	F-meas.	Jaccard Index
Correct-Detection	Median Blur	**0.81**	0.97	**0.88**	**0.79**
	Bilateral	**0.81**	0.92	0.86	0.76
	Two-step	0.79	**0.99**	**0.88**	**0.79**
Usable-Result	Median Blur	0.82	0.97	0.89	0.80
	Bilateral	**0.85**	0.93	0.89	0.80
	Two-step	0.84	**0.99**	**0.91**	**0.83**
Usable-Seed	Median Blur	**0.92**	0.97	**0.95**	**0.90**
	Bilateral	**0.92**	0.93	0.93	0.86
	Two-step	0.91	**0.99**	**0.95**	**0.90**
Correct-Usable	Median Blur	**0.80**	0.97	**0.87**	**0.78**
	Bilateral	**0.80**	0.92	0.86	0.75
	Two-step	0.78	**0.99**	**0.87**	**0.78**

Table 6.2: Evaluation of single-step preprocessing methods and two-step cascade method for implant detection during algorithm development.

The increase in usable results and the reduction in false negatives produced using the two-step cascade method was compelling, so further testing was conducted to see if the multi-step cascade approach could be improved. To this end, the next experiments were conducted to evaluate the effects of changes to the degree of smoothing and allowable radius size of detected circles.

At this point in testing, we had been using a 5x5 kernel for the median blur filter, and a distance of 5 for both *sigmaColor* and *sigmaSpace* for the bilateral filter. To determine whether increasing the amount of smoothing during preprocessing would produce better results, we increased the kernel size for the median blur filter to 7x7 and the distance for *sigmaColor* and *sigmaSpace* to 10. The results are shown in

	Filter	TP	FP	TN	FN
Correct-Detection	Median Blur	475	114	0	16
	Bilateral	458	**109**	0	38
	Two-step	**476**	124	0	**5**
Usable-Result	Median Blur	484	105	0	16
	Bilateral	483	**84**	0	38
	Two-step	**501**	99	0	**5**
Usable-Seed	Median Blur	544	**45**	0	16
	Bilateral	521	46	0	38
	Two-step	**545**	55	0	**5**
Correct-Usable	Median Blur	469	120	0	16
	Bilateral	452	**115**	0	38
	Two-step	**470**	130	0	**5**

Table 6.3: Evaluation of single-step preprocessing methods and two-step cascade method for implant detection during algorithm development.

Tables 6.4 and 6.5. Performance was not much affected by the change, however the precision and F-measure values were slightly greater for *Correct-Detection* when a higher degree of smoothing was used. For this reason, these values were used in subsequent iterations of the two-step version of the algorithm.

One of the weaknesses observed while evaluating each iteration of the implant detection algorithm was that many times the head of the implant was detected, but the circle was unusable because it was too small. This was particularly common when the X-ray is taken from the position shown in Figure 6.7. Our next experiment was designed to determine whether decreasing the value of *min_radius* during the *No Circles* subprocess (Figure 4.5) actually improves performance, and if so, to establish

	Kernel	*sigma-Color*	*sigma-Space*	Prec.	Sens.	F-meas.	Jaccard Index
Correct-Detection	5x5	5	5	0.79	0.99	0.88	0.79
	7x7	10	10	**0.80**	**0.996**	**0.89**	**0.80**
Usable-Result	5x5	5	5	**0.84**	0.99	**0.91**	0.83
	7x7	10	10	0.83	**0.996**	0.90	0.83
Usable-Seed	5x5	5	5	0.91	0.99	0.95	0.91
	7x7	10	10	0.91	**0.996**	0.95	0.91
Correct-Usable	5x5	5	5	0.78	0.99	0.87	0.78
	7x7	10	10	**0.79**	**0.996**	**0.88**	**0.79**

Table 6.4: Comparing performance of two-step cascade method for implant detection using different amounts of smoothing during algorithm development.

the optimal decreased *min_radius* value. To answer these questions, three versions of the two-step system were compared. The only difference between the three was the value used for *min_radius* during the *No Circles* subprocess: decreasing the minimum radius to (0.1 * *min_dimen*), decreasing the minimum radius to (0.15 * *min_dimen*), and no decrease to the minimum radius. The results are shown in Tables 6.6 and 6.7.

Both versions that decrease the *min_radius* value outperformed the control version that did not. The version that decreased the minimum radius to (0.15 * *min_dimen*) performed better overall than the version that decreased the value of *min_radius* to (0.1 * *min_dimen*), with the exception of producing two more false negatives. Additionally, using the larger value of *min_radius* resulted in fewer occurrences of incorrect circle selection within the implant, with 13 occurrences as

	Kernel	*sigmaColor*	*sigmaSpace*	TP	FP	TN	FN
Correct-	5x5	5	5	476	124	0	5
Detection	7x7	10	10	**481**	**122**	0	**2**
Usable-	5x5	5	5	**501**	**99**	0	5
Result	7x7	10	10	499	104	0	**2**
Usable-	5x5	5	5	545	55	0	5
Seed	7x7	10	10	**548**	55	0	**2**
Correct-	5x5	5	5	470	130	0	5
Usable	7x7	10	10	**476**	**127**	0	**2**

Table 6.5: Comparing performance of two-step cascade method for implant detection using different amounts of smoothing during algorithm development.

Decreased *min_radius*	Prec.	Sens.	F-meas.	Jaccard Ind.
0.1 * *min_dimen*	0.80	**0.996**	0.89	0.80
0.15 * *min_dimen*	**0.82**	0.992	**0.90**	**0.81**
No change	0.79	0.98	0.87	0.78

Table 6.6: Comparing performance of two-step cascade method for implant detection using different values for decreased *min_radius* during algorithm development.

opposed to 25 occurrences for the version that used (0.1 * *min_dimen*). Based on this information, the value of *min_radius* is decreased during the *No Circles* subprocess to (0.15 * *min_dimen*). The slight increase in the number of false negatives is a tradeoff for improved detection of true positives and fewer false positives.

The other adjustment made to the detection phase at this stage of the experiments was to adjust the way *mask_rad*, the radius of the unmasked circular region of the mask, was calculated. Until this point, *mask_rad* was calculated by increasing the radius of the circle returned by *find_circle*() by 50%. Since the heads of TSA im-

Decreased *min_radius*	TP	FP	TN	FN
0.1 * *min_dimen*	481	122	0	**2**
0.15 * *min_dimen*	**493**	**108**	0	4
No change	470	127	0	8

Table 6.7: Comparing performance of two-step cascade method for implant detection using different values for decreased *min_radius* during algorithm development.

plants can vary greatly in size, the ROI for an implant with a very small or very large head may not be well represented by this one-size fits all approach. For instance, an overly small circle may exclude too much of the body, whereas an overly large one can include unneeded areas that make segmentation more difficult. To test this, we modified the mask creation method to include the calculation described by Equation 4.3. This change made an improvement in three of the four evaluation categories, with the greatest improvement to *Usable-Result*, and no change to *Usable-Seed*. The results are shown in Tables 6.8 and 6.9.

	Radius Increase	Prec.	Sens.	F-meas.	Jaccard Ind.
Correct-Detection	50%	0.82	0.99	0.898	0.81
	Adjusts	0.82	0.99	**0.90**	**0.82**
Usable-Result	50%	0.87	0.992	0.926	0.86
	Adjusts	**0.88**	**0.993**	**0.934**	**0.88**
Usable-Seed	50%	0.90	0.99	0.95	0.90
	Adjusts	0.90	0.99	0.95	0.90
Correct-Usable	50%	0.81	0.99	0.893	0.807
	Adjusts	**0.82**	0.99	**0.895**	**0.810**

Table 6.8: Comparing performance of two-step cascade method for implant detection using different methods of determining radius of mask during algorithm development.

	Radius Increase	TP	FP	TN	FN
Correct-Detection	50%	493	108	0	4
	Adjusts	**495**	**106**	0	4
Usable-Result	50%	522	79	0	4
	Adjusts	**530**	**71**	0	4
Usable-Seed	50%	542	59	0	4
	Adjusts	542	59	0	4
Correct-Usable	50%	488	113	0	4
	Adjusts	**490**	**111**	0	4

Table 6.9: Comparing performance of two-step cascade method for implant detection using different methods of determining radius of mask during algorithm development.

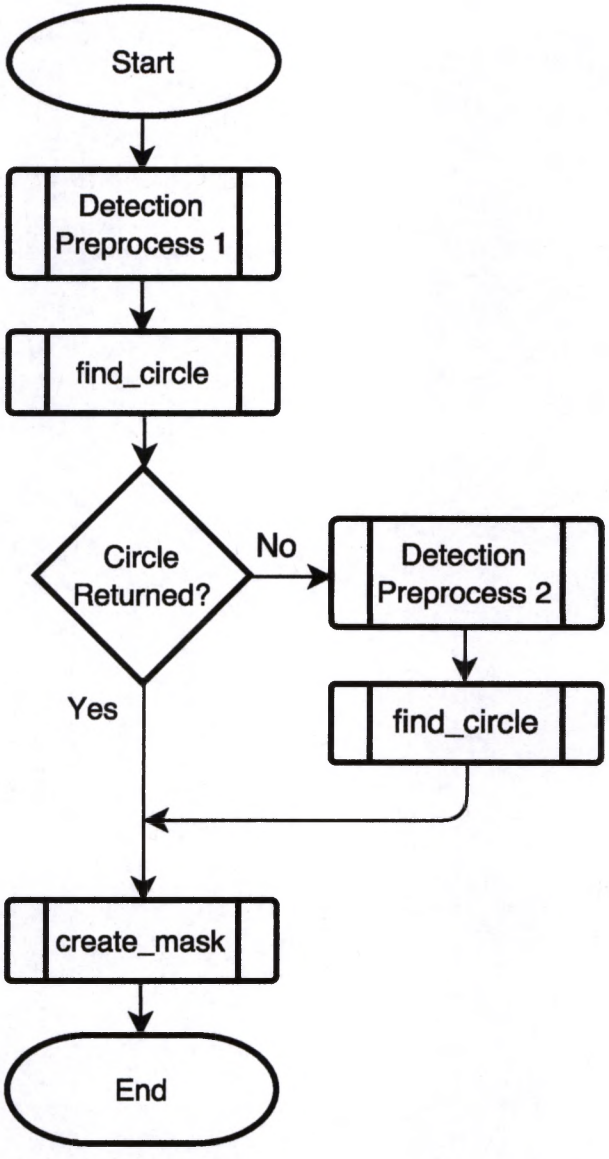


Figure 6.6: Flowchart for two-step version of the detection phase.

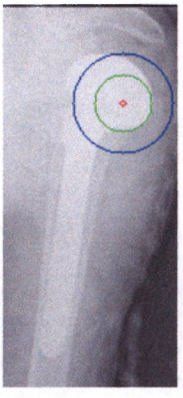


Figure 6.7: Illustrative example of correct detection of implant with unusable circle.

6.3.4 Three-Step Cascade Detection Algorithm

The two-step cascade had improved the number of usable results over the single-step versions, but the number of false positives was still high. Despite the lack of overlap between false negative cases for the two preprocessing filters, perhaps the two steps were too similar in terms of the degree to which the images were smoothed. This lead to our next question: would using a three-step cascade process in which the second preprocessing step used a smaller kernel value for median blur produce better results than using a two-step cascade?

In order to determine which kernel size should be used, two versions of the system using the three-step cascade were compared: one using a 3x3 kernel in the second step, the other using a 5x5 kernel in the second step. Both used a 7x7 kernel in the third step. The results of our test showed no difference in performance between the two versions. We chose to use a kernel size of 3x3 in the second step in order to create a greater difference between the second and third steps.

Another question was whether false circles were being returned because *acc_min*, the minimum allowable accumulator threshold, was too low. The value of *acc_th*, the accumulator threshold, is decremented in $find\_circle()$ when no circles are returned by the initial call to $HoughCircles()$ until either at least one circle is detected, or *acc_th* reaches *acc_min*. For typical circle detection (where the objects are near true circles) the lower this value is, the higher the likelihood of false circles being returned, but if it is too high, the circles may not be detected at all. To determine

	acc_min	Decrement Amt.	Prec.	Sens.	F-meas.	Jaccard Ind.
Correct-Detection	10	2	0.82	1	0.89	0.82
	5	5	**0.83**	1	**0.94**	**0.83**
Usable-Result	10	2	0.86	1	0.89	0.86
	5	5	**0.88**	1	**0.95**	**0.88**
Usable-Seed	10	2	0.88	1	0.95	0.88
	5	5	**0.9**	1	0.95	**0.9**
Correct-Usable	10	2	0.81	0.99	0.89	0.81
	5	5	**0.82**	0.99	0.89	**0.82**

Table 6.10: Evaluating effects of decrementing the accumulator threshold less drastically during algorithm development.

whether higher values for the *acc_th* would result in fewer false positives for implant detection, we increased the value of *acc_min* from 5 to 10. To increase the chances of a higher value being used, we decreased the decrement amount from 5 to 2. Contrary to our expectation, the version using the higher *acc_min* value had slightly poorer performance than the version that reduced the *acc_min* value more drastically (see Table 6.10).

6.3.5 Implant Detection Results

The top performing detection phase approaches are similar both in substance and performance; one is essentially the three-step version of the other. The structure of the two- and three-step approaches are compared in Table 6.11, and the results are compared in Table 6.12. The performance of the three-step version is very slightly higher than that of the two-step version for our dataset, for this reason we used the

three-step approach in our proposed algorithm.

No. of Steps	1st Filter	2nd Filter	Kernel	3rd Filter	Kernel
2	Bilateral	Median Blur	7x7	-	-
3	Bilateral	Median Blur	3x3	Median Blur	7x7

Table 6.11: Comparing structures of two- and three-step cascade implant detection approaches.

	No. of Steps	Prec.	Sens.	F-meas.	Jaccard Ind.
Correct-Detection	2	0.82	0.992	0.90	0.82
	3	**0.83**	**0.996**	0.90	0.82
Usable-Result	2	0.88	0.993	0.93	**0.88**
	3	0.88	**0.996**	0.93	0.87
Usable-Seed	2	0.90	0.993	0.95	0.90
	3	0.90	**0.996**	0.95	0.90
Correct-Usable	2	0.82	0.992	0.89	0.81
	3	0.82	**0.996**	**0.90**	0.81

Table 6.12: Comparing performance of two- and three-step cascade implant detection approaches.

6.4 Segmentation Evaluation

This section describes the experimental evaluation and algorithm development for the segmentation phase. Sections 6.4.1 and 6.4.2 describe the evaluation methods used, Section 6.4.3 describes the algorithm development process, and Section 6.4.4 discusses the results of the experimental evaluation for the segmentation phase.

6.4.1 Segmentation Evaluation Methods

We evaluated the performance of the segmentation phase based on the amount of overlap between the ground-truth images described in Chapter 3 and the output of the segmentation phase. The automatic segmentation results were compared pixel-by-pixel with the manually segmented ground-truth images. Because the segmentation phase relies on the output of the detection phase, we evaluated segmentation performance using the subset of detection phase output determined to be usable by the *Usable-Result* criteria (*Usable-Result* true positives). For simplicity, we will refer to this as the *Usable-Result subset.* This allows us to evaluate the performance of our proposed segmentation algorithm as distinct from complete system performance, which considers all results from the full dataset. Evaluation of the performance of the full system is discussed in Section 6.5.

Development of the detection and segmentation phases occurred semi-concurrently; for this reason, the results discussed in this section use masks and seeds from a slightly different version of the implant detection algorithm than is described in

Chapter 4.

6.4.2 Over-Segmentation Evaluation Method

In order to evaluate over-segmentation detection performance, the results of the first preprocessing/segmentation combination in the cascade were evaluated by visual inspection to determine which images were over-segmented. Each image was classified as either over-segmented or not over-segmented. This classification was used as the ground-truth for evaluating the results of the proposed over-segmentation detection methods.

6.4.3 Segmentation Experiments

The seeded region growing algorithm itself was chosen based on the experimental results of a pilot study. The experimental evaluation used during algorithm development for the segmentation phase focused on identifying an optimum preprocessing method. The following preprocessing approaches were considered:

- No smoothing.
- Mean shift filter, using a spatial radius and color distance of 10.
- Median blur filter, using a 3x3 kernel.
- Bilateral smoothing, using a *sigmaSpace* of 5, and a *sigmaColor* of 5.

- Histogram equalization followed by no smoothing.
- Histogram equalization followed by mean shift filter, using a spatial radius and color distance of 10.
- Histogram equalization followed by median blur filter, using a 3x3 kernel.
- Histogram equalization followed by bilateral smoothing, using a *sigmaSpace* of 5, and a *sigmaColor* of 5.

Each of these preprocessing approaches was evaluated using a single-step version of our proposed algorithm, as illustrated by Figure 6.8. The results are shown in Table 6.13.

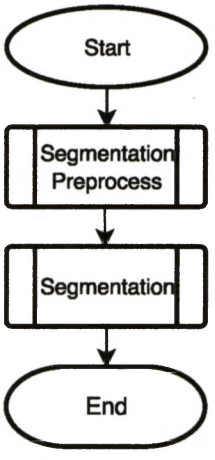


Figure 6.8: Flowchart for single-step version of the segmentation phase.

As mentioned in Section 5.3, segmentations produced using the mean shift filter for preprocessing are less prone to over-segmentation at the expense of the loss of

Preprocess	Prec.	Sens.	Spec.	F-Meas.	Jaccard Ind.	Accuracy
No Histogram Equalization						
No Smoothing	0.51	0.64	0.91	0.57	0.40	0.87
Mean Shift	**0.81**	0.64	**0.98**	**0.71**	**0.55**	**0.93**
Median Blur Filter	0.50	0.65	0.90	0.56	0.39	0.86
Bilateral Filter	0.50	**0.65**	0.90	0.57	0.40	0.87
Preceded by Histogram Equalization						
No Smoothing	0.77	0.46	0.98	0.58	0.41	0.91
Mean Shift Filter	**0.90**	0.55	**0.99**	**0.68**	**0.51**	**0.93**
Median Blur Filter	0.76	0.51	0.97	0.61	0.44	0.91
Bilateral Filter	0.77	**0.59**	0.97	0.67	0.50	0.92

Table 6.13: Evaluation of segmentation results comparing potential one-step preprocessing approaches.

some detail, whereas those produced using the bilateral filter retain finer details but are more susceptible to over-segmentation. The issue of over-segmentation prompted the question of whether results could be improved using a two-step cascade that would determine which preprocessing steps to use according to whether the image was likely to have been over-segmented. Ideally, the bilateral filter would be used when possible (*Segmentation Preprocess 1*), and another approach (*Segmentation Preprocess 2*) would be used for images identified as likely to be over-segmented. Four criteria for determining the likelihood of over-segmentation were tested as potential methods for the *Over-Segmentation Detection* step. Each of these was tested in combination with four different methods of preprocessing for images that are identified as likely to be over-segmented.

The potential criteria for flagging an image as likely to be over-segmented are as

follows:

- Mean intensity of pixels in region of interest > 130
- IQR of intensity of pixels in region of interest < 100
- Standard deviation of pixels in region of interest < 60
- Segmentation result $> 50\%$ of pixels in region of interest

The thresholds for the first three criteria were determined by analyzing the corresponding values for segmentation results that fell into the following categories: not over-segmented, over-segmented, fully over-segmented, and partially over-segmented. The threshold for the fourth criteria was determined by visual analysis of the segmented images. The first three approaches to *Over-Segmentation Detection* do not require that a segmentation result exist to make a prediction. Therefore, a slightly different structure than that shown in Figure 5.1 is employed. This alternate structure is illustrated in Figure 6.9.

Each of the proposed *Over-Segmentation Detection* methods was tested in combination with four different potential methods for *Segmentation Preprocess 2*. *Segmentation Preprocess 1* is the bilateral filter as described in Section 5.3 for all versions of the two-step approach. The methods of preprocessing likely to be over-segmented images are:

- Histogram equalization followed by no smoothing.

- Histogram equalization followed by mean shift filter, using a spatial radius and color distance of 10.
- Histogram equalization followed by median blur filter, using a 3x3 kernel.
- Histogram equalization followed by bilateral smoothing, using a *sigmaSpace* of 5, and a *sigmaColor* of 5.

6.4.4 Segmentation Results

Tables 6.13 and 6.14 and the the accompanying charts in Figure 6.10 show the results for each configuration evaluated in these experiments. The results of the *Over-Segmentation Detection* experiments are shown in Table 6.15 and the accompanying chart in Figure 6.11.

Based on the results shown in Table 6.15, segmentation result $> 50\%$ of pixels in the region of interest for *Over-Segmentation Detection* outperforms the other three approaches. This is also reflected in the segmentation results shown in Table 6.14, where the top performing version of the two-step approach employs this method of *Over-Segmentation Detection*, in combination with the bilateral filter for *Segmentation Preprocess 1* and histogram equalization followed by the mean shift filter for *Segmentation Preprocess 2*. The results shown in Table 6.13 show histogram equalization followed by the mean shift filter to be the top performing preprocess for the single-step approach.

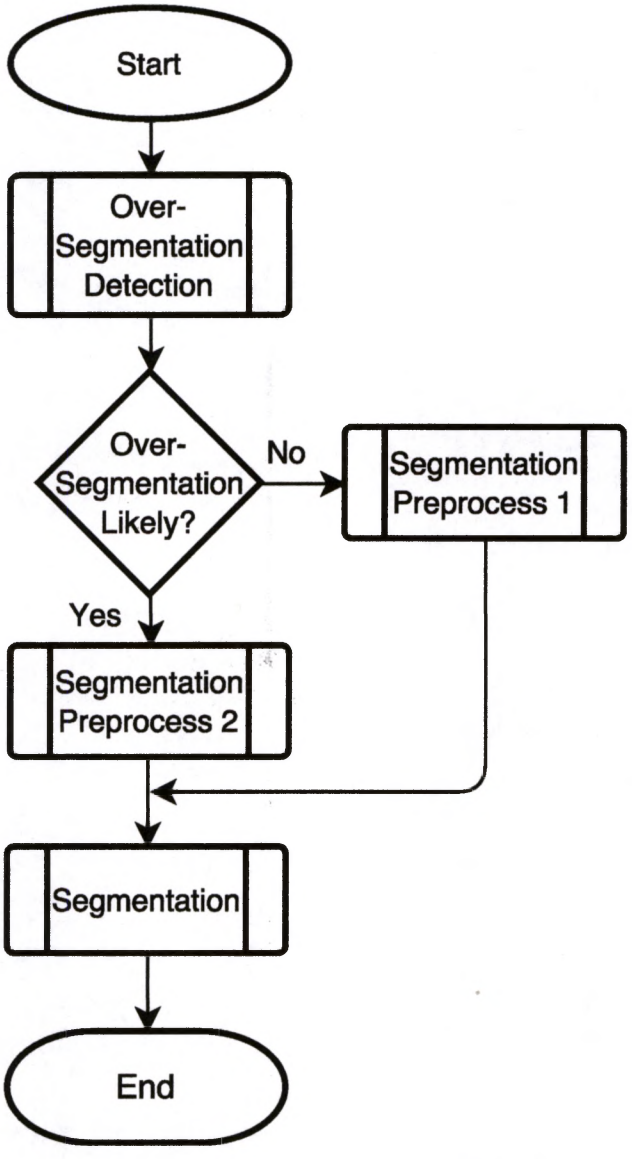


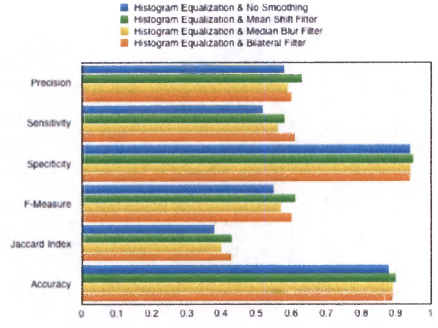
Figure 6.9: Flowchart for alternate two-step version of the segmentation phase.

Segmentation Preprocess 2		Prec.	Sens.	Spec.	F-Meas.	Jaccard Index	Acc.
Mean Intensity > 130							
Hist. Eq.	No Smoothing	0.58	0.52	0.94	0.55	0.38	0.88
Hist. Eq.	Mean Shift	**0.63**	0.58	**0.95**	**0.61**	**0.43**	**0.90**
Hist. Eq.	Median Blur	0.59	0.56	0.94	0.57	0.40	0.89
Hist. Eq.	Bilateral	0.60	**0.61**	0.94	0.60	**0.43**	0.89
IQR < 100							
Hist. Eq.	No Smoothing	0.64	0.54	0.95	0.59	0.42	0.90
Hist. Eq.	Mean Shift	**0.71**	0.60	**0.96**	**0.65**	**0.48**	**0.91**
Hist. Eq.	Median Blur	0.64	0.58	0.95	0.61	0.44	0.90
Hist. Eq.	Bilateral	0.65	**0.62**	0.95	0.64	0.47	0.90
Std. Deviation < 60							
Hist. Eq.	No Smoothing	0.66	0.52	0.96	0.58	0.41	0.90
Hist. Eq.	Mean Shift	**0.74**	0.59	**0.97**	**0.66**	**0.49**	**0.92**
Hist. Eq.	Median Blur	0.66	0.56	0.95	0.61	0.44	0.90
Hist. Eq.	Bilateral	0.68	**0.62**	0.95	0.64	0.48	0.91
Segmentation result > 50% of pixels in ROI							
Hist. Eq.	No Smoothing	0.81	0.57	0.98	0.67	0.50	0.92
Hist. Eq.	Mean Shift	**0.90**	0.60	**0.99**	**0.72**	**0.56**	**0.94**
Hist. Eq.	Median Blur	0.79	0.57	0.98	0.66	0.50	0.92
Hist. Eq.	Bilateral	0.79	**0.62**	0.97	0.69	0.53	0.93

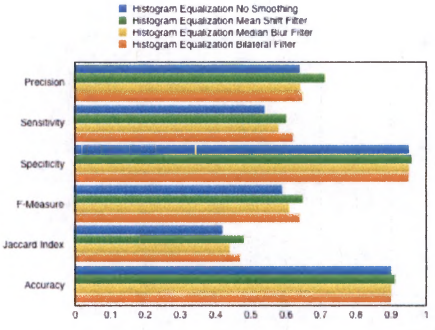
Table 6.14: Evaluation of segmentation results comparing potential two-step approaches using different methods to detect over-segmentation.

Detection Method	Prec.	Sens.	Spec.	F-Meas.	Jaccard Ind.	Acc.
Mean Intensity > 130	0.47	0.54	0.49	0.50	0.34	0.51
IQR < 100	0.64	0.78	0.62	0.70	0.54	0.69
Std. Deviation < 60	0.56	0.83	0.44	0.67	0.50	0.62
Seg. result > 50% of ROI	**0.98**	**0.95**	**0.99**	**0.97**	**0.94**	**0.97**

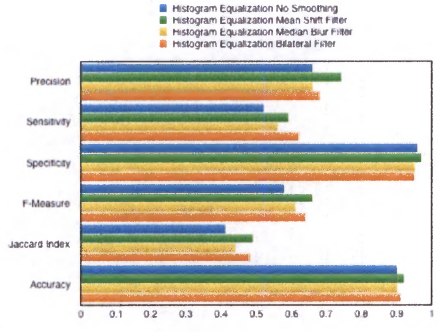
Table 6.15: Evaluation of over-segmentation detection results comparing potential approaches.



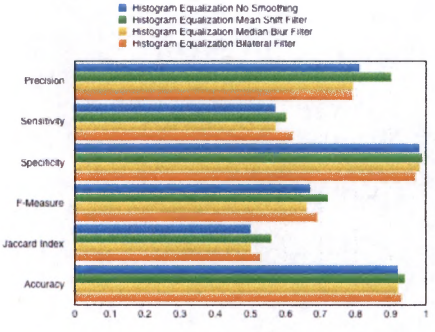
(a) Mean Intensity > 130



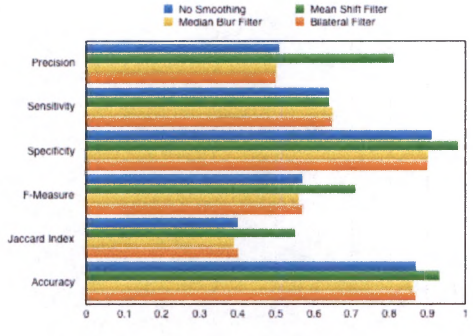
(b) IQR < 100



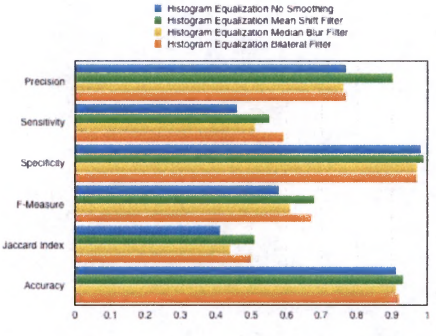
(c) Std. Deviation < 60



(d) Seg. result > 50% of pixels in ROI



(e) Single-step, smoothing only



(f) Single-step, hist.eq. and smoothing

Figure 6.10: Evaluation of segmentation results comparing potential one- and two-step approaches.

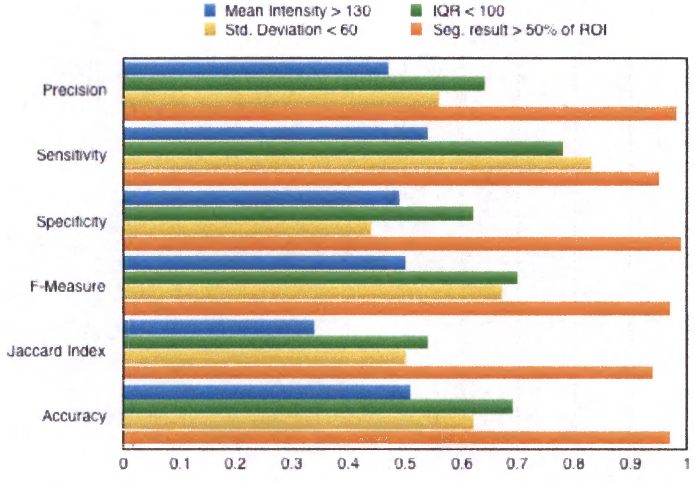


Figure 6.11: Evaluation of over-segmentation detection results comparing potential approaches.

6.5 Evaluation of Complete System

This section describes the experimental evaluation of the complete system using the most promising implant detection and segmentation approaches identified in Sections 6.3 and 6.4. Only the three-step approach discussed in 6.3 is used for implant detection in the experiments described in this section, although several variations of the segmentation phase algorithm are compared. This is because there was greater variation in the output of the different segmentation approaches than there was between the detection approaches.

6.5.1 Complete System Evaluation Methods

To assess performance quantitatively, the implant detection and segmentation phases are evaluated individually as described above. We then evaluate the overall performance of our proposed approach based on the final segmentation results for the entire dataset. We include the results for the *Usable-Result subset* as well in order to assess the performance of the segmentation phase. To determine the qualitative performance of our system, we classify each final segmentation result as correct or incorrect on the basis of a visual evaluation. A segmentation is considered correct if the resulting image is one that could be used as input to a future classification component. This metric is application specific and seeks to evaluate the system as a preprocess for the aforementioned future classification component.

6.5.2 Experiments

The following approaches to the segmentation phase were selected for further evaluation:

- Two-step: bilateral smoothing filter for *Segmentation Preprocess 1*, histogram equalization followed by mean shift filter for *Segmentation Preprocess 2*. This is our proposed algorithm described in Chapter 5.
- Two-step: bilateral smoothing filter for *Segmentation Preprocess 1*, histogram equalization followed by median blur filter for *Segmentation Preprocess 2*
- Two-step: bilateral smoothing filter for *Segmentation Preprocess 1*, histogram equalization followed by bilateral smoothing filter for *Segmentation Preprocess 2*
- Single-step: mean shift filter for *Segmentation Preprocess*
- Single-step: histogram equalization followed by mean shift filter for *Segmentation Preprocess*
- Two-step: bilateral smoothing filter for *Segmentation Preprocess 1*, mean shift filter for *Segmentation Preprocess 2*

All of the two-step approaches use segmentation result $> 50\%$ of pixels in the region of interest for *Over-Segmentation Detection*. The two-step approach using histogram equalization followed by mean shift filter for *Segmentation Preprocess*

2 was tested multiple times using different values for the spatial radius and color distance. The two most successful versions of this approach are included in the results and discussion.

The first five approaches were chosen because they appeared the most promising based on the outcome of the experiments described in Section 6.4. Of the quantitative measures, precision and F-measure were weighted the most heavily in evaluating performance. As its name implies, precision measures how exact the results are: it quantifies the quality of predicted positive results by determining what percentage are actual positive results. This makes precision a useful tool in determining segmentation quality, and one that will reflect over-segmentation. However, precision does not reflect the completeness of a segmentation. Sensitivity, or recall, provides this measure and is therefore useful in considering under-segmentation. For our purposes, some minor under-segmentation was preferable to over-segmentation, so sensitivity was not given the same weight as precision in our evaluation of performance. Because F-measure considers both precision and sensitivity, it provides a measure that reflects our priorities more closely than sensitivity alone. Due to the relatively high numbers of true negative pixels in the segmentation results, specificity and accuracy tended to be less reflective of segmentation quality than the other measures. Jaccard index was the most stringent of the measures, and would have been given greater weight had perfect segmentation been our goal. For these reasons, precision and F-measure were given precedence over the other measures.

Visual evaluation of the results also played a role in the selection of these approaches, particularly in the cases of the two-step methods using the median blur filter and bilateral smoothing filter for *Segmentation Preprocess 2*. Although the one- and two-step versions using histogram equalization followed by the mean shift filter for preprocessing performed better statistically in terms of pixel-by-pixel overlap, the segmentations output by these versions are not necessarily more usable. Unlike over-segmentation, some under-segmentation, such as that shown in Figure 5.4c, does not negatively affect the usability of the segmentation result for our purposes. However, an excessive amount of under-segmentation, like that illustrated by Figure 6.12, is often associated with poor segmentations produced by the versions using histogram equalization followed by the mean shift filter for *Segmentation Preprocess 2*. This under-segmentation issue was the reason for introducing the sixth method, which uses the mean shift filter without histogram equalization for *Segmentation Preprocess 2*, and was not previously evaluated in Section 6.4. Under-segmentation was also the reasoning for re-testing the two-step approach using histogram equalization followed by mean shift filter for *Segmentation Preprocess 2* with different values for the spatial radius and color distance. Initially, a value of 10 was used for both, as in the experiments described in Section 6.4. Like the other parameter values used for the filters in the segmentation experiments, these were derived experimentally from a pilot study. However, the pilot study did not focus on images prone to over-segmentation, as was the case for the images now being smoothed

using the mean shift filter. After performing qualitative evaluation on the results of this approach, further testing was performed and a second version of this method using a spatial radius of 5 and a color distance of 10 was also included.

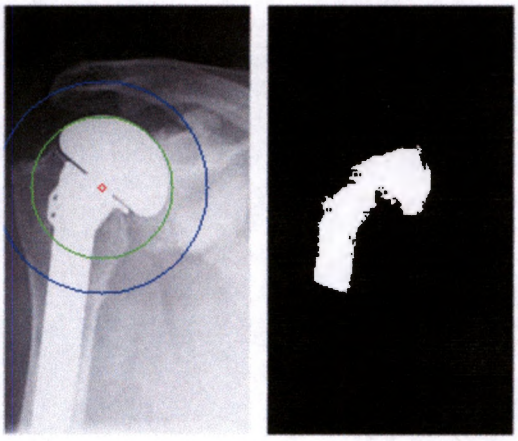


Figure 6.12: Illustrative example of excessive under-segmentation.

6.5.3 Results

The results of the quantitative evaluation of these experiments are shown in Table 6.16 and in the accompanying charts in Figure 6.13. Segmentation performance for the complete system is evaluated for both the *Usable-Result subset* and the full set of results. The versions that performed best for precision were the two-step approach that used histogram equalization followed by mean shift filter using a spatial radius and color distance of 10 for *Segmentation Preprocess 2*, and the single-step version using histogram equalization followed by mean shift filter for *Segmentation Prepro-*

cess. The precision for these were 0.91 and 0.9, respectively, for the *Usable-Result subset.* When all implant detection results are included, precision was 0.77 for both. Precision for the other versions ranged from 0.78 to 0.89 for the *Usable-Result subset*, and from 0.65 to 0.75 for the full dataset. For F-measure, the two-step versions that used the bilateral smoothing filter for *Segmentation Preprocess 1*, and the mean shift filter for *Segmentation Preprocess 2*, had the highest scores. These were both of the two-step approaches using histogram equalization prior to applying the mean shift filter, and the version that does not use histogram equalization. The F-measure for all three was 0.72 for the *Usable-Result subset.* The F-measure for both approaches using histogram equalization followed by mean shift filter for *Segmentation Preprocess 2*, was 0.64 for the full set. The version without histogram equalization had an F-measure of 0.63 for the full set.

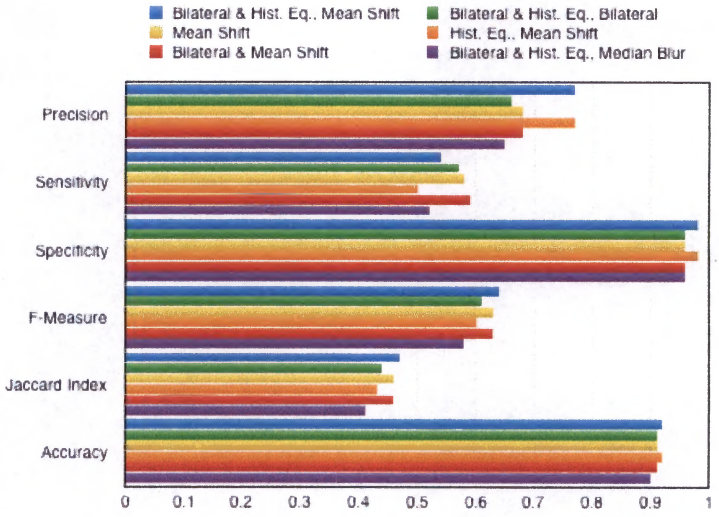
If we prioritize either precision or F-measure as the most important measure of performance, the difference between that of our proposed algorithm and the next best performing approach are negligible. However, based on these quantitative evaluation metrics, our proposed algorithm using histogram equalization prior to applying the mean shift filter outperforms the others when both measures are considered.

Using the method described in Section 6.5.1, we evaluated the qualitative performance of the four versions with the best quantitative performance, the results are shown in Table 6.17. Our proposed algorithm and the version of the two-step

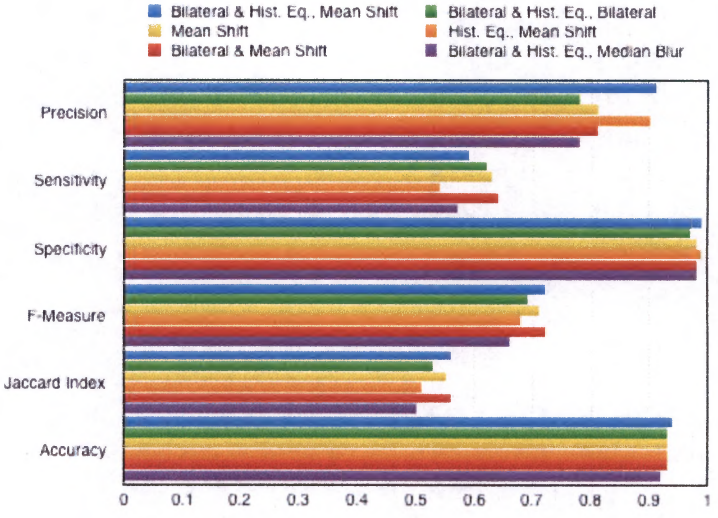
approach using the bilateral smoothing filter for *Segmentation Preprocess 1* and the mean shift filter for *Segmentation Preprocess 2*, had the highest scores. The version of our proposed algorithm using a spatial radius of 5 had the highest percentage of usable results, with 72% overall and 81% for the *Usable-Result subset.* For the version of our proposed algorithm using a spatial radius of 10, the percentage of usable results was 70% overall, and 79% for the *Usable-Result subset.* The percentage of usable results for other two-step approach was 69% overall, and 78% for the *Usable-Result subset.* The results for the other approach we evaluated for qualitative performance, the single-step version using histogram equalization followed by mean shift filter for *Segmentation Preprocess*, were 45% overall and 50% for the *Usable-Result subset.*

Preprocess 1	Preprocess 2	Prec.	Sens.	Spec.	F-Meas.	Jaccard Index	Acc.
Full Dataset							
Bilateral	Hist. Eq., Mean Shift ($sr = 10$)	**0.77**	0.54	**0.98**	**0.64**	**0.47**	**0.92**
Bilateral	Hist. Eq., Mean Shift ($sr = 5$)	0.75	0.55	0.97	**0.64**	**0.47**	**0.92**
Bilateral	Hist. Eq., Bilateral	0.66	0.57	0.96	0.61	0.44	0.91
Mean Shift	-	0.68	0.58	0.96	0.63	0.46	0.91
Hist. Eq., Mean Shift	-	**0.77**	0.50	**0.98**	0.60	0.43	**0.92**
Bilateral	Mean Shift	0.68	**0.59**	0.96	0.63	0.46	0.91
Bilateral	Hist. Eq., Median Blur	0.65	0.52	0.96	0.58	0.41	0.90
Usable-Result Subset							
Bilateral	Hist. Eq., Mean Shift ($sr = 10$)	**0.91**	0.59	**0.99**	**0.72**	0.56	**0.94**
Bilateral	Hist. Eq., Mean Shift ($sr = 5$)	0.89	0.61	**0.99**	**0.72**	**0.57**	**0.94**
Bilateral	Hist. Eq., Bilateral	0.78	0.62	0.97	0.69	0.53	0.93
Mean Shift	-	0.81	0.63	0.98	0.71	0.55	0.93
Hist. Eq., Mean Shift	-	0.90	0.54	**0.99**	0.68	0.51	0.93
Bilateral	Mean Shift	0.81	**0.64**	0.98	**0.72**	0.56	0.93
Bilateral	Hist. Eq., Median Blur	0.78	0.57	0.98	0.66	0.50	0.92

Table 6.16: Evaluation of complete system comparing potential one- and two-step approaches.



(a) Full Dataset



(b) Usable-Result Subset

Figure 6.13: Evaluation of complete system comparing potential one- and two-step approaches.

Preprocess 1	Preprocess 2	Percentage of Usable Results	
		Full Dataset	Usable-Result Subset
Bilateral	Hist. Eq., Mean Shift ($sr = 10$)	70%	79%
Bilateral	Hist. Eq., Mean Shift ($sr = 5$)	**72%**	**81%**
Bilateral	Mean Shift	69%	78%
Hist. Eq., Mean Shift	-	45%	50%

Table 6.17: Qualitative evaluation of full-system segmentation results comparing top three approaches.

6.6 Discussion

We chose the components of our proposed approach based on the usability of the results that were produced. The difference between the segmentation results for the best performing single-step version and the best performing two-step versions was that the former had a stronger tendency towards under-segmentation. As discussed in Sections 5.3 and 6.4.3, the motivation for using a two-step cascade approach to select the best preprocessing method for each image was to take advantage of the strengths of both the bilateral filter and the mean shift filter while minimizing the weaknesses of each. For our application, some degree of under-segmentation is not detrimental, and is generally preferable to over-segmentation because less information is lost about the shape of the implant. However, a severely under-segmented result does not preserve enough information for classification and is unusable.

Although using the mean shift filter alone in *Segmentation Preprocess 2* produced similar results qualitatively, we chose to include histogram equalization prior to applying the mean shift filter in *Segmentation Preprocess 2* because the quantitative results were better. For our proposed approach, using a smaller spatial radius for the mean shit filter in *Segmentation Preprocess 2* produced slightly higher scores for the qualitative evaluation measures. Because this version produced a greater number of usable results, these parameter values are used in our proposed system.

One of the challenges of using a seeded region growing approach is the importance of seed selection on the outcome of the segmentation. We wanted to automate

our system as much as possible, and therefore chose to develop an algorithm that would detect the region of interest and select a seed without user interaction. By doing so, the segmentation performance became dependent on the success of the implant detection algorithm and the seed it provided. Precision for the segmentation algorithm drops from 0.91 for the *Usable-Result subset*, to 0.77 for the full dataset. The difference in performance illustrates the tradeoff between increased automation and segmentation performance. Additionally, because we define a usable seed as one which lies within the boundaries of the implant in the region of interest, a usable seed is not necessarily an ideal seed. When seeds are manually chosen, the user has the ability to select a seed that they believe will result in the best segmentation outcome. It is probable that performance would be improved by making our system semi-automatic by allowing the user to select the seed with a single click if the initial segmentation is unsatisfactory.

Another challenge was handling the prevalence of over-segmentation. Although we chose a method that improved our quantitative results, it introduced the issue of under-segmentation that was severe enough in some cases to affect the usability of the results.

All of the approaches we considered performed better on certain models than on others. All of the approaches performed well for implant detection but poorly for seed selection for the Tornier Aequalis Fracture Shoulder model (see Figure 6.1a). This particular model has a very distinctively shaped body with a large central

gap located directly below the head of the implant. When the X-ray shows the image in profile, the center of the detected circle is often within this gap. Excluding models with fewer than 10 images in the dataset, the models that appeared easiest for our algorithm to detect were the Depuy Global Fracture and HRP models, and the Zimmer Select Shoulder model. It is not apparent why performance for these models was higher; the images are not of higher quality, nor does there appear to be a correlation between detection and the number of images that were identified as likely to be over-segmented.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis proposes a system to automatically detect and segment TSA implants in X-ray images using a detection algorithm based on the Hough transform for circles and a seeded region growing algorithm for segmentation. The purpose of the system is as a preprocessing component for classification in a system that classifies TSA implants by model and manufacturer. Our proposed approach attempts to solve this problem automatically using low-resolution images of varying quality in anticipation of a realistic workflow for future users.

At this time, we are aware of no other system that attempts to detect and segment TSA implants in X-ray images with the specific goal of producing usable segmentations for classification by manufacturer and model. The approach we have proposed is tailored to preprocessing for classification by focusing on the feature-

rich head and body, rather than on the distal portion of the stem. Although two projects with similar goals for other total joint arthroscopy implants have published their findings, one is less suited to our classification needs [7] and the other is still in early development stages without a fully designed segmentation solution [1].

7.2 Future Work

One of our goals was to minimize user interaction, however, it may not be possible to construct a fully automatic segmentation and detection tool using this approach. Although our proposed approach worked well for much of our dataset, improvement of the seed selection approach could be of significant benefit to the performance of the overall system. Additionally, a means of handling cases of unusable detection outcomes, as well as those of over- and under-segmentation, would need to be incorporated in the final system. Making our system semi-automatic by including functionality that would allow the user to select a seed with a single click if they are unsatisfied with a segmentation result is a possible means of addressing both of these challenges. This is an area of future work that could significantly improve segmentation results while preserving the usability of the system.

Our dataset reflected our intention to solve the detection and segmentation problems using difficult images, but it was almost entirely made up of thumbnail images. It is possible that as a result of this limitation, adjustments will need to be made to increase the adaptability of the proposed algorithm. An expanded dataset with a

greater variety of image sizes and qualities should be included in further development of the system.

The next major step in the development our proposed system is the implementation of the classification component. This process may also provide insight to adjustments that should be made to the current system, such as which features should be prioritized in assessing segmentation performance. Upon success for TSA analysis, we plan to extend the framework to address arthroplasty implants for other joints, such as the knee and hip.

Bibliography

[1] J. Bredow, B. Wenk, R. Westphal, F. Wahl, S. Budde, P. Eysel, and J. Oppermann, *Software-based matching of x-ray images and 3d models of knee prostheses*, Technology and Health Care **22** (2014), no. 6, 895–900.

[2] R. Duda and P. Hart, *Use of the hough transformation to detect lines and curves in pictures*, Communications of the ACM **15** (1972), no. 1, 11–15.

[3] T. Huang, GJTGY Yang, and G. Tang, *A fast two-dimensional median filtering algorithm*, IEEE Transactions on Acoustics, Speech, and Signal Processing **27** (1979), no. 1, 13–18.

[4] A. Kaehler and G. Bradski, *Learning opencv 3: Computer vision in c++ with the opencv library*, O'Reilly Media, 2016.

[5] C. Kimme, D. Ballard, and J. Sklansky, *Finding circles by an array of accumulators*, Communications of the ACM **18** (1975), no. 2, 120–122.

[6] OpenCV, *Open source computer vision library*, `https://github.com/opencv/opencv/releases/tag/2.4.10`, 2014.

[7] A. Oprea and C. Vertan, *A quantitative evaluation of the hip prosthesis segmentation quality in x-ray images*, Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on **1** (2007), 1–4.

[8] OpenCV Dev Team, *Image filtering - opencv 2.4.10.0 documentation*, `http://docs.opencv.org/2.4.10/modules/imgproc/doc/filtering.html`.

[9] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, Computer Vision, 1998. Sixth International Conference on, IEEE, 1998, pp. 839–846.

[10] *Common us shoulder prostheses*, `http://faculty.washington.edu/alexbert/Shoulder/CommonUSShoulderProstheses.htm`.

[11] HK Yuen, J. Princen, J. Illingworth, and J. Kittler, *Comparative study of hough transform methods for circle finding*, Image and vision computing **8** (1990), no. 1, 71–77.

[12] P. Yushkevich and G. Gerig, *Itk-snap (version 3.2) [software]*, `http://www.itksnap.org/pmwiki/pmwiki.php`, 2014.

[13] P. Yushkevich, J. Piven, H. Hazlett, R. Smith, S. Ho, J. Gee, and G. Gerig, *User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability*, Neuroimage **31** (2006), no. 3, 1116–1128.