THE FEASIBILITY OF USING FEATURE-FLOW AND LABEL TRANSFER SYSTEM TO SEGMENT MEDICAL IMAGES WITH DEFORMED ANATOMY IN ORTHOPEDIC SURGERY

YAO JUN ZHAO

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

December 2012

ABSTRACT

In computer-aided surgical systems, to obtain high fidelity three-dimensional models, we require accurate segmentation of medical images. State-of-art medical image segmentation methods have been used successfully in particular applications, but they have not been demonstrated to work well over a wide range of deformities. For this purpose, I studied and evaluated medical image segmentation using the feature-flow based Label Transfer System described by Liu and colleagues. This system has produced promising results in parsing images of natural scenes. Its ability to deal with variations in shapes of objects is desirable. In this paper, we altered this system and assessed its feasibility of automatic segmentation. Experiments showed that this system achieved better recognition rates than those in natural-scene parsing applications, but the high recognition rates were not consistent across different images. Although this system is not considered clinically practical, we may improve it and incorporate it with other medical segmentation tools.

DEDICATION

This thesis is dedicated to those who encouraged me and helped me in pursuing the academic knowledge.

ACKNOWLEDGEMENTS

Thanks to Professor Burton Ma. Without his consistent support and advices, my research could not be done and this paper could not be written.

Thanks to Professor James Elder, Professor Parke Godfrey and Professor Keith Schneider. With their patient and kind advices, this paper became more understandable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter One: Introduction

1.1 Computer-aided surgical systems

There are many kinds of computer-aided surgical systems. Two kinds that make use of segmentation of medical images are image-guided surgical systems and patient-specific guides.

1.1.1 Image-guided surgical systems

Perhaps the most common form of computer-aided surgery is the use of an image-guided surgical system which provides real-time virtual visualization for surgical procedures. In such a system, surgeons can observe the virtual view of tools relative to patient anatomy on a computer screen even when the direct physical view is obscured. Computer image-guided surgery may result in improved patient care, shortened surgery time, and easy access for medical education.

Image guided surgical systems consist of four components: anatomical models, real-time position measurement, a registration method, and visualization means. Additionally, pre-surgical planners may be included [20].

- **Anatomical models** are typically 3D computer models derived from a series of 2D images acquired by computed tomography (CT) or magnetic resonance imaging (MRI). These images are segmented and a 3D anatomical model is constructed using the segmentation data. The segmentation procedures usually require human interactions as achieving accurate segmentation automatically is difficult. Hence, it is an active research area.

- **Pre-surgical planning** allows the surgeon to plan or simulate the surgical procedures. Surgeons can navigate or rotate the 3D volume or 3D anatomical model to gain better understanding of the patient anatomy than merely observing stacks of 2D CT or MRI images. Moreover, based on the 3D constructs, a pre-surgical planner can simulate the results of a proposed plan. Fig. 1.1 shows pre-operative planning of one kind of wrist surgery using planning software.

- **Real-time position measurement** is the means to measure an object's position and orientation in real-time. Examples of real-time positioning systems include articulated mechanical arms, ultrasonic acoustic trackers, electromagnetic trackers, and optical trackers. Fig. 1.2 shows a surgery procedure aided by a real-time position measurement system.

- A **registration method** is required to superimpose the moving tool on the computer model. The procedure to find the mathematical transformation between

the patient coordinate frame and the computer model coordinate frame is called registration. Commonly used registration methods can be marker-based, landmark-based, surface-based, or voxel property-based [21]. Because of the invasive nature of marker-based methods, low accuracy of landmark-based methods, and the computational high cost of voxel property-based methods, surface-based registration methods are the most common methods used in modern image-guidance systems. In surface-based registration, the surface of the anatomy of interest is digitized (with the use of a real-time position measurement system) and then the transformation that best maps the digitized points to the computer model of the anatomy is computed. Fig. 1.3 illustrates a surface-based registration that transforms digitized patient anatomy points to pre-operative computer model.

- **Visualization means** often involves high graphic-performance computer workstations. These workstations may be used to provide pre-surgical navigation of patient anatomy, simulate post-surgical results, or even render virtual tool and anatomic models in real-time during surgical operations.

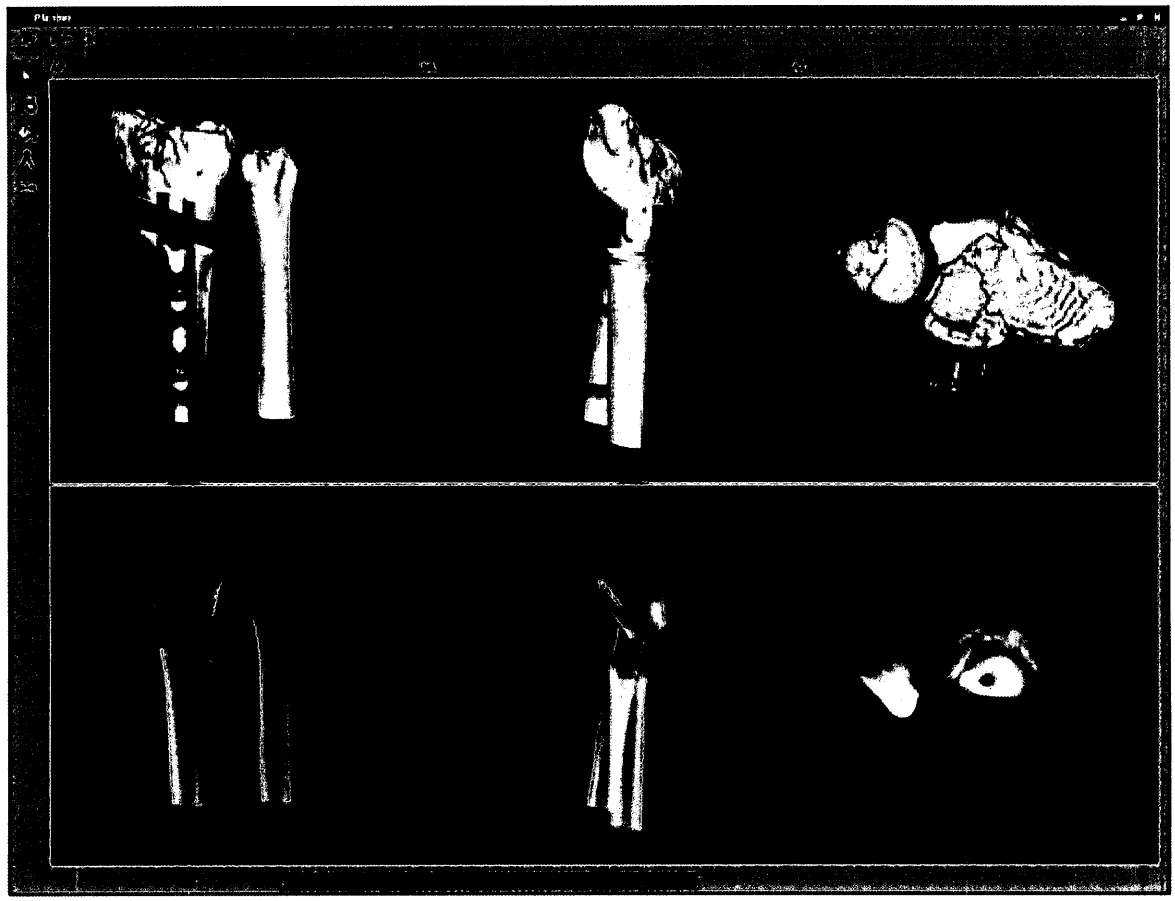


Fig. 1.1 Planning of distal radius osteotomy

This figure illustrates pre-operative planning of a bone deformity procedure called distal radius osteotomy. This particular planner supports operations such as cutting the virtual bone and moving the separate fragments to restore normal anatomical alignment. After restoring the alignment of the bone fragments, a virtual fixation plate can be positioned on the fragments. The planner also generates a synthetic x-ray visualization of the bone fragments.

Fig. 1.2 Intraoperative real-time position measurement

(Left) A three camera optical tracking system that measures real time 3D position is shown. The tracking system measures the location of infrared light emitting diodes. (Right) Two targets containing infrared light emitting diodes are shown. One target is attached to the patient to track the motion of the patient. Another target is attached to a surgical tool to track the motion of the tool; this particular tool is a pointing stylus used to digitize points from the surface of the anatomy.

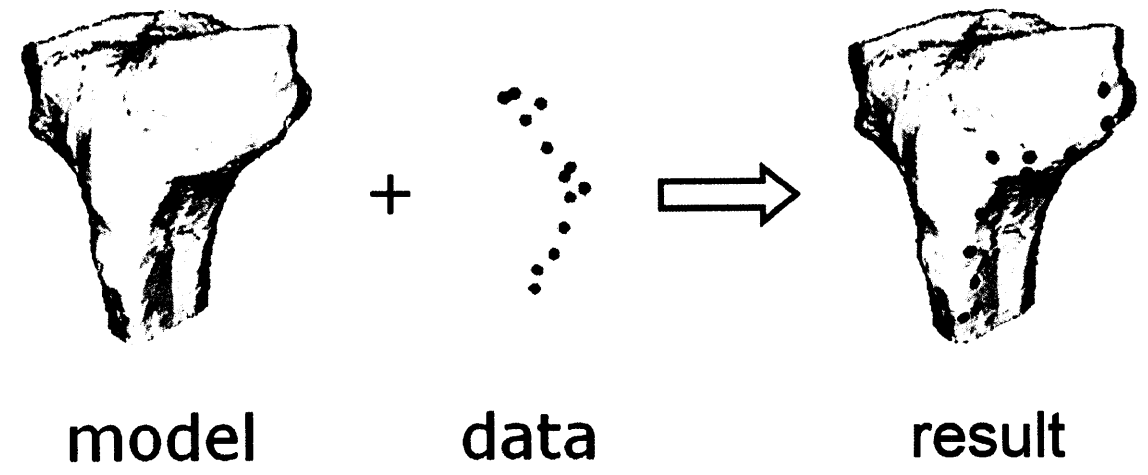


Fig. 1.3 Surface-based registration

In surface-based registration, points digitized from the surface of the patient's anatomy are used to find the transformation that aligns the patient to pre-operative computer model [20].

1.1.2 Patient-specific guides

A second type of computer-aided surgery uses patient-specific guides that are designed to match the anatomy of the patient. Such guides are most commonly used in dentistry and orthopedics where rigid bony surfaces are available on which to mount the guides. Instead of providing virtual navigational guidance using a computer display, patient-specific guides provide mechanical navigational guidance using physical structures incorporated into the body of the guide. An example of a patient-specific guide for wrist surgery is shown in Fig. 1.4 and Fig. 1.5.

Similar to image-guided surgery, the construction of a patient-specific guide requires anatomic models computed from medical images. These models are used in pre-surgical planning to design the guide so that the guide will mount accurately on the patient's anatomy and provide suitable navigational guidance. The guides are then fabricated using computer numerical control machining or three-dimensional printing.

Accurate segmentation of the medical images is especially important for the fabrication of patient-specific guides because the guides are designed to conform to the anatomy of the patient. Errors in the segmentation can result in a guide that does not mount properly to the anatomy of the patient, which in turn will lead to inaccurate navigational guidance.

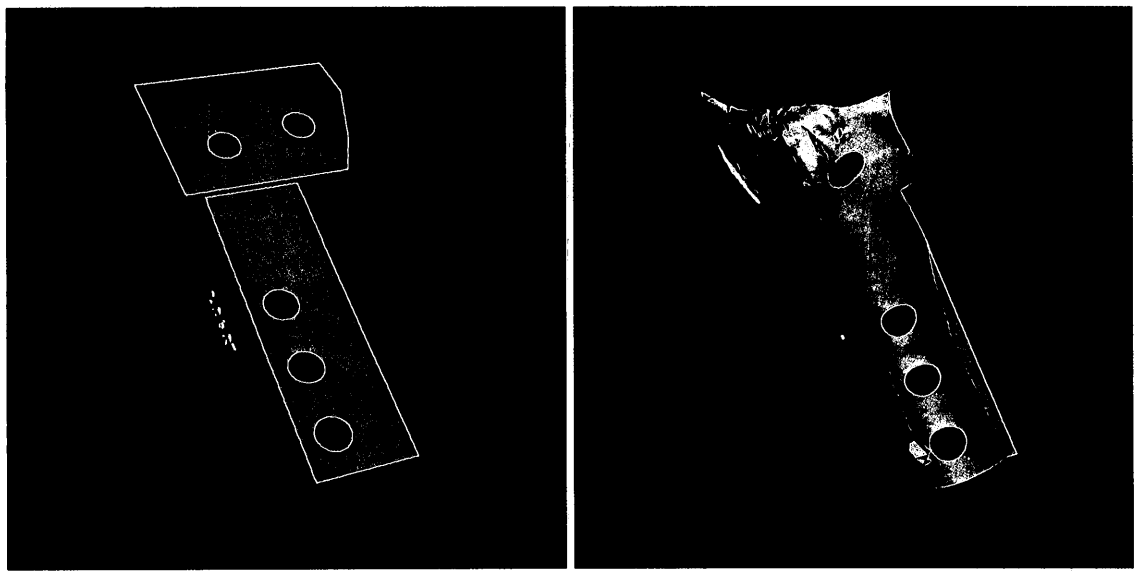


Fig. 1.4 Patient-specific guide for distal radius osteotomy

(Left) View of the top of a computer model of a patient-specific guide. The channels in the guide provide navigation guidance for a surgical drill bit. (Right) View of the bottom surface of the guide; the shape of this surface matches the specific shape of the bone to be operated on.

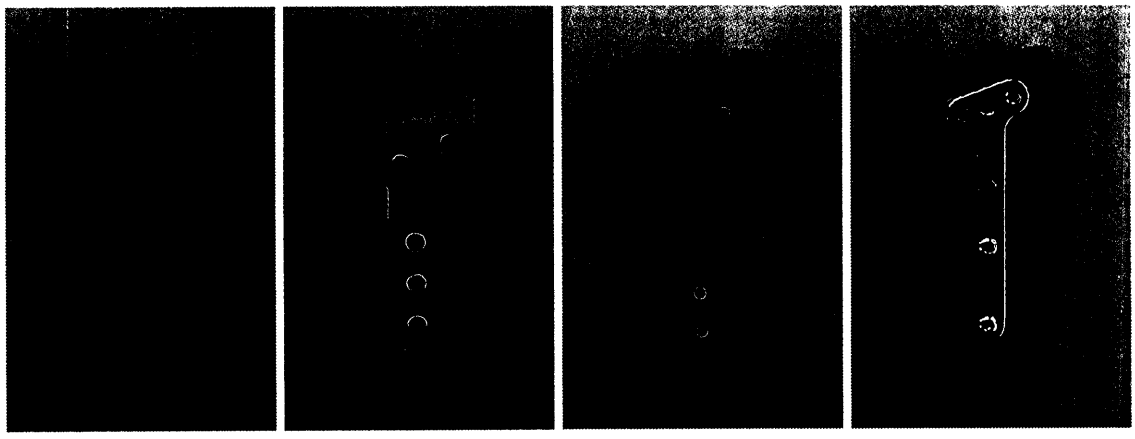


Fig. 1.5 Application of a patient-specific guide for distal radius osteotomy

(Left) A plastic model of a deformed radius that is to be corrected using a patient-specific guide. (Middle-left) The guide is mounted on the deformed radius and pilot holes are drilled into the radius using the channels. (Middle-right) The radius is cut into two fragments after the pilot holes have been drilled. (Right) The fixation plate is fastened to radius using the predrilled pilot holes.

1.2 Medical Image Segmentation

Accurate segmentation of 2D, 3D, and 4D (3D plus time) medical images to extract anatomy of interest for analysis is crucial in many computer-aided surgical systems. Constructing anatomical models relies on segmenting data from patient images. Computer aided visualizations in diagnosis, surgical planning, and simulation heavily depend on segmented image data. In particular, intraoperative planning of surgical procedures requires fast and accurate segmentation to generate real-time visualizations.

1.2.1 Challenges

The medical image segmentation problem is challenging. There is no general or unique solution due to the constantly growing number of highly varied objects of interest, different medical imaging modalities, partial-volume effects, signal inhomogeneity, noise, etc. [22] [23]. For instance, partial-volume effects, the artifacts caused by multiple tissue types contributing to a single voxel, can result in a blurring of intensity across boundaries. Another artifact, signal inhomogeneity, can cause unwanted variations in image intensity. These artifacts can significantly degrade the performance of segmentation methods and have to be taken into account.

1.2.2 Definitions

Image segmentation is the means to divide an image into non-overlapping homogeneous regions such that a more simplified and meaningful representation of the image is generated to be analyzed easily. The sets that contribute to a segmentation must satisfy,

$$\Omega = \bigcup_{i=1}^{n} \Omega_i \tag{1.1}$$

where $\Omega_j \cap \Omega_k = \emptyset \; for \; j \neq k$ and each Ω_i is connected [23]. In general, the homogenous regions are the regions with a well-defined intensity distribution.

For medical image segmentation, the constraint of being connected is usually removed because disconnected regions belonging to same kind of tissue often occur. The segmentation problem of determining the region sets Ω_i without having to be connected is called **pixel classification**. These sets are called classes.

Labeling is the process of assigning a meaningful designation to each region set [23]. Labeling can be performed separately after segmentation, or simultaneously in some segmentation techniques [23].

1.2.3 Segmentation Techniques

Due to diversity of objects of interest, image modalities and problem specific natures, no universal segmentation technique exists. In the following sections, several commonly used segmentation techniques in computer assisted surgery are reviewed.

1.2.4 Rule-based techniques

- **Thresholding** attempts to separate homogeneous classes by specific intensity values, called thresholds. A segmented region, Ω_i, segmented by thresholding is defined as

$$\Omega_i = \{x \in R^2 \mid t_i < I(x) \leq t_{i+1}\} \tag{1.2}$$

where t_i and t_{i+1} are the lower and upper thresholds and $I(x)$ is the intensity value of the pixel at location x. Thresholding is simple and fast, but it does not address topological information and is sensitive to noise and signal inhomogeneity. It is often used as an initial step of image-processing.

- **Region growing** extracts connected image region based on predefined criteria. For example, one criterion to determine if a pixel at location x falls in a region R is that the intensity difference relative to an initial seed point is smaller than a threshold t_s, and that the contrast of the region, $C(R)$, stays smaller than a threshold t_r

$$|I(x) - I(x_{seed})| \leq t_s \tag{1.3}$$

$$C(R) = max_R\big(I(x)\big) - min_R\big(I(x)\big) \leq t_r \tag{1.4}$$

Region growing incorporates topological information but it requires manual interaction to set the initial seed points.

- **Region splitting and merging** first splits the whole image until all regions are homogenous by certain criteria, and then merges small regions as long as merged regions are still homogenous by another criterion. Combining splitting and merging can avoid over-segmentation. Like region growing, this method is also sensitive to initialization and requires user assistance.

1.2.5 Edge-based techniques

Edges correspond to abrupt changes in intensity in images. Ideally, edges are equivalent to the boundaries that separate objects. An edge can be expressed as the local maximum in the first derivative magnitude (basic gradient threshold and Canny detector) or a zero crossing in the second derivative (Laplacian zero-crossing). Edge-based techniques also involve human input for appropriate thresholds and may suffer with boundary discontinuity.

1.2.6 Optimal statistical inference

- **Classifier** (supervised) methods are pattern recognition techniques that partition the feature space derived from an image by using training data with known labels [23]. The feature space is the range space of a function of the image. In the simplest case, feature space can be the intensity alone. A classifier that is nonparametric, such as K-nearest-neighbor classifier and Parzen window classifier, makes no assumption about the statistical structure of data. A parametric classifier, however, assumes that the pixel intensities are independent

samples from a mixture of probability distributions, usually Gaussian. Common parametric classifiers are maximum-likelihood and Bayes classifier. The mixture, called a finite-mixture model, can be expressed by the probability density function (PDF) as,

$$f(I(x);\theta,\pi) = \sum_{n=1}^{N} \pi_n f_n(I(x);\theta_n) \tag{1.5}$$

where $I(x)$ is the intensity of the pixel at location x, f_n is a PDF component parameterized by θ_n, and $\pi_n$is the weight (contribution) of f_n in the mixture. Classifier methods are non-iterative and computationally efficient, but manual interaction is required to obtain training data.

- **Clustering** (unsupervised) methods do not require training data. Instead they self-train with available data, iteratively alternating between segmenting and characterizing class properties. Common clustering algorithms are K-mean (ISODATA), fuzzy c-means and expectation maximization (EM). Although clustering is done without training data, initial segmentation or correct number of classes is required.

Both classifier and clustering methods do not take into account topological information, and hence, are sensitive to noise and signal inhomogeneity. To achieve robustness of noise, Markov Random Field (MRF) modeling is incorporated into these methods under a Bayesian prior model. MRF will be further discussed in Section 2.3.

1.2.7 Atlas-based techniques

In atlas-based techniques, an atlas is generated by compiling information on the anatomy of interest. The atlas typically depicts prototypical locations and shapes of anatomical structures together with their spatial relations [22]. Then this atlas is used as a reference frame for segmenting the target image. Standard atlas-based techniques treat segmentation as a registration problem which attempts to find one-to-one transformations mapping atlas images to target images. These transformations can be linear or nonlinear. Atlas techniques are similar to classifiers except they are implemented in spatial domain rather than in feature space.

Single atlas segmentation is not sufficient since one atlas cannot represent the whole image population. Commonly, a large number of atlases are used to address this issue, which is called multi atlas segmentation; however, this approach can be extremely time consuming. Another issue is, even with multi atlas, it is difficult to find accurate segmentations of complex structures due to natural anatomical variability, as well as variability caused by trauma and disease.

1.2.8 Model-based techniques

In model-based techniques, objects in images are described by model parameters. First, the parameterized prior information, namely the model, is obtained from examples. The prior information can be of object shapes and topology, object appearance, image

formation or expert observations. Then specified constraints are imposed to segment a target image using the model. Deformable models (also called snake or active contour models) and level set models restrict prior information to smoothness constraints, while statistical shape models incorporate prior information heavily [24]. Model-based techniques have become very popular in medical analysis due to their ability to obtain a continuous boundary of an object in spite of shape variations, image noise, image inhomogeneity, and occlusions [22].

1.2.9 Other techniques

- The **Watershed** technique, unlike other edge-based techniques that needs additional mechanisms for joining contours, partitions an entire image into homogenous regions based on object morphology. It is a simple, fast and intuitive method even in poor contrast. However, it may suffer from over-segmentation and a post-processing step is needed to merge separate regions that belong to the same structure.
- The **Graph cut** technique represents the image as a graph with pixels as vertices and pixel-to-pixel connections as weighted edges. The edge weight is determined by the similarity of the two vertices, i.e.,

$$w_{i,j} = e^{-\frac{\left(I(x_i)-I(x_j)\right)^2}{2\sigma^2}} \tag{1.6}$$

Then the segmentation problem becomes a minimization problem of finding the minimal cut for a fully connected graph,

$$\min_{A,B}\left(\sum_{x_i\in A, x_j\in B} w_{i,j}\right) \tag{1.7}$$

- The **Artificial Neural Network (ANN)** resembles biological learning with parallel networks of processing nodes. ANN represents a paradigm for machine learning and can be adapted to various segmentation methods such as classifier, clustering and deformable models.

1.2.10 Evaluating performance in medical image segmentation

To evaluate the segmentation performance in medical applications, we are usually interested in the segmentation accuracy and error rates. To determine segmentation accuracy, or *correctness rate*, we need to find the ratio of the number of correctly segmented pixels to the true number of pixels in this segment. In general, we desire a higher correctness rate; however, correctness rate is not the only metric we need to consider. Error rates may also be used to evaluate the performance of a segmentation method.

There are two kinds of errors we may encounter, type I error and type II error. Consider the following null hypothesis, H_0, and alternative hypothesis, H_1:

H_0: The label for the pixel at location x is 'background'

H_1: The label for the pixel at location x is 'bone'

A type I error occurs when H_0 is true, but is rejected. In our example, a type I error occurs when a pixel is labeled as bone when it is actually background. Type I errors are

also called false positive errors because the alternative hypothesis is incorrectly taken to be true. A type II error occurs when H_0 is false, but it is accepted as true. In our example, a type II error occurs when a pixel is labeled as background when it is actually bone. Type II errors are also called false negative errors because the default (null) hypothesis is incorrectly taken to be true. In this thesis, for a specific segment, the ratio of the number of labeled pixels having type I error to the true number of pixels is called its *false positive rate*, and the ratio of the number of labeled pixels having type II error to the true number of pixels is called its *false negative rate*. One can easily see that the summation of correctness rate and false negative rate is always equal to 1, and that there is no strict correlation between correctness rate and false positive rate. In extreme cases, the false positive rate can be greater than 1, meaning large numbers of pixels are labeled as belonging to a certain segment by mistake. To determine the segmentation performance by the two types of error rates, we need to consider which error rate is more significant.

Which error rate is more significant? In medical image segmentations, the relative importance of type I and type II errors is application specific. For example, when designing a patient-specific guide, as shown in Fig. 1.6, type II errors will miss some bone surface which can lead to incorrect positioning of the guide, whereas type I errors are less likely to cause positioning errors. Another example is when models computed from segmented images are used for surface-based registration; both type I and type II errors are undesirable in these circumstances because the resulting model surface will not accurately match the patient's anatomy.

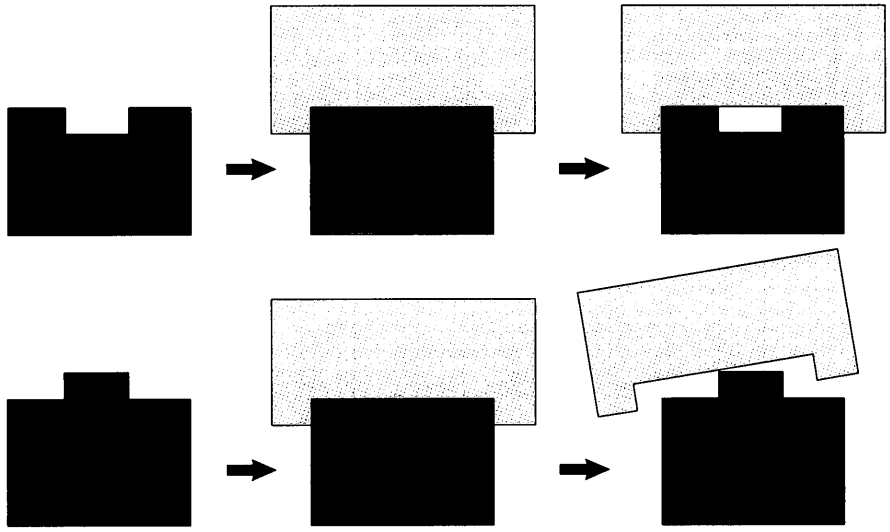


Fig. 1.6 Type I error and type II error in jig customization

First row demonstrates type I error, and second row demonstrates type II error. The green block represents the real shape and the blue block represents the segmented shape. The yellow block is the patient specific guide designed using the segmentation. With type I error, the guide can still sit on the green block firmly, but with type II error, the guide does not conform to the actual anatomic shape.

1.3 Label Transfer System – an object recognition approach for medical image segmentation

Image segmentation is challenging and there is no general solution. The most studied and widely considered advanced medical segmentation methods are atlas-based methods and model-based methods. But they are built on specific or isolated structures and relatively consistent knowledge base. They do not work well in cases that contain high anatomical variability, unpredictable distortion, or deformation. Recently, Liu and colleagues [1] invented an object-recognition-based scene parsing system called the *Label Transfer*

System that simultaneously segments and labels a query image by matching scale and rotation invariant features between the query image and pre-annotated images. They were able to achieve an average labeling accuracy of 76.67% on a series of natural image test cases having widely varying appearance. They also showed a single example of transferring labels from a CT image to and MRI image of a brain from a single patient. It seems worthwhile to assess the performance of this method on other medical image segmentation problems.

1.4 Problem Statement

In this thesis, we assess the performance of Label Transfer System by evaluating its segmentation correctness and error rates on x-ray orthopedic images. We used hip and hand x-ray images, with the images having a wide variation in the shapes and sizes of the anatomic structures, image contrast, and image quality. In searching for improvements to the Label Transfer System, we also altered certain factors in the system and compared the results with the original approach.

Chapter Two: Literature Review

In medical image processing, thresholding methods are often used in combination with manual interaction to obtain segmentation results. Instead of using simple per-pixel intensities, more efficient and effective segmentation can be achieved by using highly distinctive features that are adaptable to variant contexts. In searching such distinctive features, methods such as Harris-corner [25], Harris-Laplace, Hessian-Laplace [26], determinant of Hessian, and Laplacian of Gaussian were studied. Lowe [27] invented the Scale-Invariant Feature Transformation (SIFT) using difference of Gaussian (DoG) to achieve high performance interest point detection. SIFT has been widely adapted in different vision applications and even extended as GLOH [28], PCA-SIFT [29] and so on. More recently, Bay and colleagues [30] created Speeded-up Robust Features (SURF) which uses an approximation of the determinant of Hessian to detect interest points. They claimed that the performance of SURF is comparable to and even superior to SIFT, especially in terms of computation speed and noise resistance.

Image alignment problems have been actively studied for applications such as image stitching, stereo matching, video tracking, object recognition and so on. Traditionally, image alignment has been achieved by aligning sparse corresponding features or by using dense intensity-based optical flow estimation. Recently, Liu and colleagues [31] invented an image alignment method called *SIFT Flow* that incorporates SIFT features and optical flow to obtain dense correspondence between a pair of images. They used a

nonparametric Markov Random Field (MRF) model utilizing cues including SIFT flow correspondence, spatial priors and preserved discontinuity (smoothness) to segment and recognize query image based on a large annotated image database. They called this object recognition and scene parsing method the *Label Transfer System*. In their method, Sequential Belief Propagation (BPS) is used to estimate the dense SIFT flow and optimize the MRF posterior probability because BPS is faster than other optimizers even though it may sacrifice matching or parsing accuracy slightly.

In segmentation of orthopedic images, bony structures usually are of major concern. Because bony structures in medical images present strong edge feature, but the extraction of SIFT or SURF features involves Gaussian operator which suppresses high frequency content, it seems beneficial to accentuate edge feature in pre-processing step. To do so, we can apply edge preserving smoothing filter to the images. Such filters include median filter [32] and bilateral filter [33]. Median filter tends to round up edge corners and filter out thin edges, which may also attenuate SIFT or SURF features. Bilateral filter is a better choice for emphasizing structure edges prior to feature extractions.

In this Chapter, we will discuss SIFT and SURF features, MRF and its optimizers, SIFT flow algorithm, Label Transfer System and bilateral filter in detail.

2.1 Scale-Invariant Feature Transform (SIFT)

In 1999, David Lowe [27] introduced – SIFT – a method that detects and describes highly distinctive image features which are invariant to scale and rotation, robust to noise and changes in illumination or viewpoint; these sparse features were used to achieve reliable matching between objects or scenes in different views. Many applications adapting or extending SIFT have since been developed and shown to be successful. These applications include object recognition, image stitching, gesture recognition and others.

In this section, we will introduce the four major steps in SIFT feature generation: 1) scale-invariant extrema detection; 2) accurate keypoint localization; 3) orientation assignment; 4) keypoint descriptor extraction.

2.1.1 Scale-invariant extrema detection

One can search for stable image features across all possible scales using a continuous function of scale known as scale space. The Gaussian function has been proven to be the only possible scale-space kernel [34] [35], thus, a scale space image function $L(x, y, \sigma)$ can be defined as,

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.1}$$

where G is the Gaussian function with standard deviation σ centered on the pixel location (x, y), I is the input image, and $*$ indicates the convolution operation.

To detect stable keypoint locations in scale space, Lowe searched for extrema in the difference of Gaussian (DoG) image, $D(x, y, \sigma)$, which is defined as,

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.2)$$

The DoG image can be found by subtraction of two nearby scale space images where the scales are separated by a constant multiplicative factor, k. The DoG image provides a close approximation to the scale-normalized Laplacian of Gaussian, $\sigma^2\nabla^2 G$, whose minima and maxima produce the most stable image features compared to other image functions such as gradient, Hessian, and Harris corner function [36]. Thus, by finding the local extrema in the DoG image, we can obtain the location (x, y) and scale (σ) of the potential stable scale-invariant feature points, as shown in Fig. 2.1.

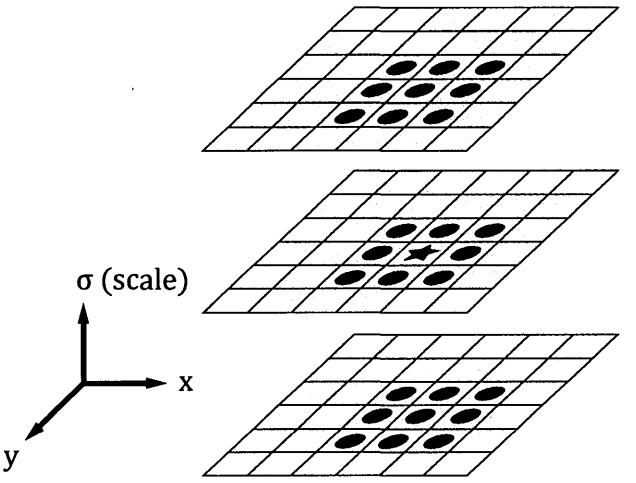


Fig. 2.1 Finding local extrema

The red star is considered an extrema if it is the minima or maxima among the pixels (green dots) at its neighboring location (x, y) and scale (σ)

An efficient way to construct $D(x, y, \sigma)$ is to divide each doubling space of σ, called an octave (e.g., $[\sigma, 2\sigma]$), into s evenly spaced intervals. Then the images in the octave are separated by a constant factor $k = 2^{1/s}$. In the first octave, the Gaussian functions $G(\sigma), G(k\sigma), \ldots, G(k^{s-1}\sigma)$ are convolved with the image I. Afterwards, Gaussian scaled images in consecutive octaves can be obtained by downsampling the corresponding Gaussian scaled images in the previous octave. For example, in octave $L(2\sigma), L(2k\sigma), \ldots L(2k^{s-1}\sigma)$, $L(2\sigma)$ is downsampled from $L(\sigma)$ in the previous octave, $L(2k\sigma)$ from $L(k\sigma)$ and so on.

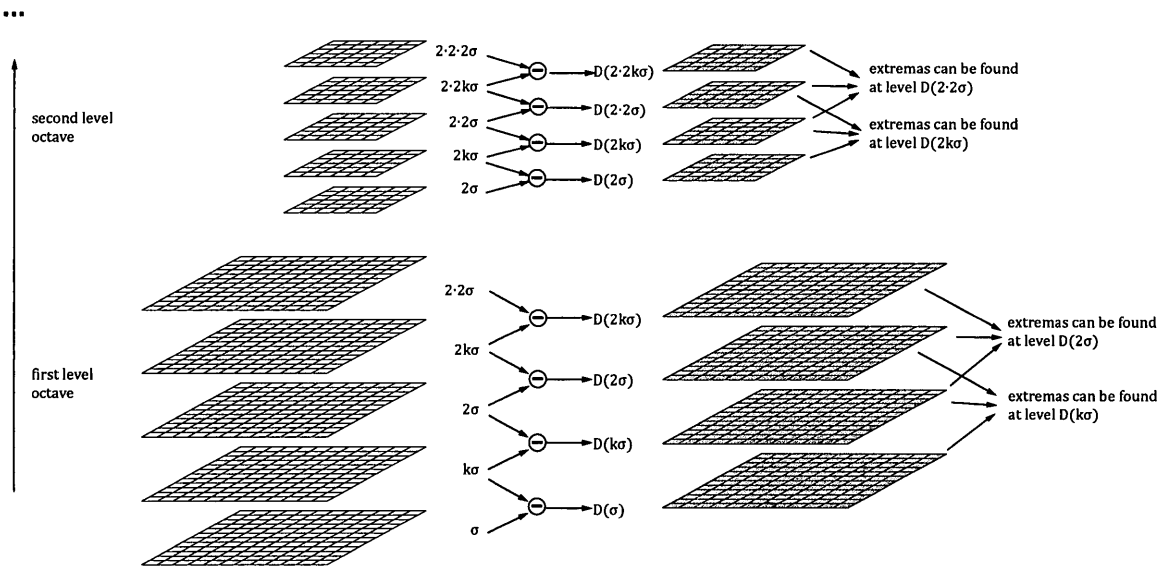


Fig. 2.2 Using octaves to simplify Gaussian convolutions

In the first octave, the initial image is repeatedly convolved with Gaussian functions with standard deviation $k^i\sigma, i = 0 \ldots 4$ to produce scale space images on the left. The adjacent Gaussian images are subtracted to produce DoG images, $D(k^i\sigma), i = 0 \ldots 3$ on the right. Finally two groups of three adjacent DoG images are used to find the local extremas at two DoG levels. After an octave is computed, the Gaussian images are down-sampled by 2 to obtain the images in the next octave.

As shown in Fig. 2.2, where $s = 2\ and\ k = \sqrt{2}$, each octave cover $s + 1 = 3$ Gaussian scaled images, $L(\sigma), L(k\sigma), L(2\sigma)$. To detect extrema in the full octave, namely, on DoG images $D(k\sigma)$ and $D(2\sigma)$, we need the base DoG image $D(\sigma)$ as well as the top one, $D(2k\sigma)$, which requires two extra Gaussian scaled images $L(2k\sigma) and\ L(2 \cdot 2\sigma)$. Hence, the total number of Gaussian scaled images needed to detect extrema in an octave is $s + 1 + 2 = s + 3$. Initially, for the first octave extrema detection, $s + 3$ convolutions have to be done. Then afterwards, there only remain downsamplings to obtain higher octave Gaussian scaled images and subtractions to find DoG. The calculation cost is dramatically reduced while the accuracy is maintained [36].

2.1.2 Improving stability by accurate keypoint localization

In Fig. 2.1, the extrema corresponds to a pixel location in one of the DoG images. To find the location of an extrema with sub-pixel accuracy, Brown and Lowe [37] fit a 3D quadratic to the DoG image location, $D(x, y, \sigma)$, such that,

$$D(\vec{x}) = D + \frac{\partial D^T}{\partial \vec{x}}\ \vec{x} + \frac{1}{2}\vec{x}^T \frac{\partial^2 D}{\partial \vec{x}^2}\vec{x} \tag{2.3}$$

where $\vec{x}$ is the offset from (x, y, σ). The quadratic coefficients are computed by approximating the derivatives with pixel differences of the neighboring points. The accurate keypoint location offset, $\hat{x}$, is then taken as the extremum of the 3D quadratic, which can be found by differentiating $D(\vec{x})$ with respect to $\vec{x}$ and setting the value of derivative to zero; doing so yields,

$$\hat{x} = -\frac{\partial^2 D}{\partial \vec{x}^2}^{-1} \frac{\partial D}{\partial \vec{x}} \tag{2.4}$$

Once $\hat{x}$ is determined, we can find $D(\hat{x})$, which is simplified by the above two equations,

$$D(\hat{x}) = D + \frac{1}{2}\frac{\partial D}{\partial \vec{x}}^T \hat{x} \tag{2.5}$$

If $|D(\hat{x})|$ is less than some threshold value, then the extrema is considered **low contrast** and will be discarded. Otherwise, the result is a keypoint location $P = (x_p, y_p, \sigma_p) = (x, y, \sigma) + \hat{x}$.

The next step is to remove keypoints that have a strong **edge response** but are otherwise poorly localized. To do this, a 2×2 Hessian matrix, H, of a keypoint, is computed using the difference of the neighboring sample points,

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{2.6}$$

Then we examine whether keypoint satisfies,

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{2.7}$$

where $Tr(H) = D_{xx} + D_{yy}$, $Det(H) = D_{xx}D_{yy} - (D_{xy})^2$ are the trace and determinant of H, which resemble the summation and production of eigenvalues of H. r is the magnitude ratio between larger eigenvalue and the smaller one. If the above criterion is not met, the keypoint is considered edge response and will be discarded. In practice, $r = 10$ is chosen.

2.1.3 Rotation invariance by orientation assignment

In SIFT, rotation invariance is achieved by assigning a consistent orientation to each keypoint based on the local image properties. The keypoint is represented relative to this orientation and maintains consistency against different rotations.

For a keypoint with scale σ_p, we select the Gaussian smoothed image, $L(x, y)$, with scale closest to σ_p. For each sample in this image, the gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ is calculated by pixel differences,

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + ((L(x,y+1) - L(x,y-1))^2} \quad (2.8)$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad (2.9)$$

Then, we build an orientation histogram which has 36 bins covering the 360 degree range of orientation. The weighted gradient magnitude of each sample around the keypoint is added to the histogram according to its orientation θ. The weighting window is a Gaussian circle with standard deviation $1.5\sigma_p$. Then the highest peak in the histogram and any other peak within 80% of the highest peak will be chosen as the keypoint orientation. Thus, it is possible to find multiple keypoints at single location but with different orientations. To avoid boundary effects and to address the influence of the neighboring orientation beside a selected peak, a parabola is used to fit the 3 histogram values closest to this peak to interpolate the peak position.

2.1.4 SIFT Descriptor

From the previous step, we find the location, scale and orientation for each stable scale-rotation-invariant keypoint, $P = (x_p, y_p, \sigma_p, \theta)$. The next step is to find a highly distinctive descriptor for the local image region. At the same time, this descriptor should also be invariant to illumination or viewpoint changes.

First, the image gradient magnitudes and orientations around the keypoint location are sampled from the Gaussian blurred image with scale closest to σ_p. Then, the coordinates of the descriptor and gradient orientations are rotated relative to the keypoint orientation to achieve orientation invariance. A Gaussian weighting function is then used to assign a weight to the gradient magnitude of the sample points around the keypoint to avoid sudden changes in the descriptor for small changes in the position of the weighting window, and to give less emphasis to gradients far from the center of the weighting window.

Next, we divide the 16×16 sample region around the keypoint to 4×4 subregions with each subregion containing 4×4 sample points, as shown in Fig. 2.3. The samples in a subregion are accumulated into orientation histograms with 8 orientation bins. To avoid boundary effects, trilinear interpolation is used to distribute the value of each gradient sample into adjacent histogram bins. Thus, a $4 \times 4 \times 8 = 128$ element feature vector is generated as the keypoint descriptor.

Finally, to reduce the influence of changes in illumination, the 128 element vector is normalized. Furthermore, to address non-linear illumination changes, one can remove the large gradients by thresholding the values in the unit vector to no larger than 0.2 and then renormalizing to unit vector.

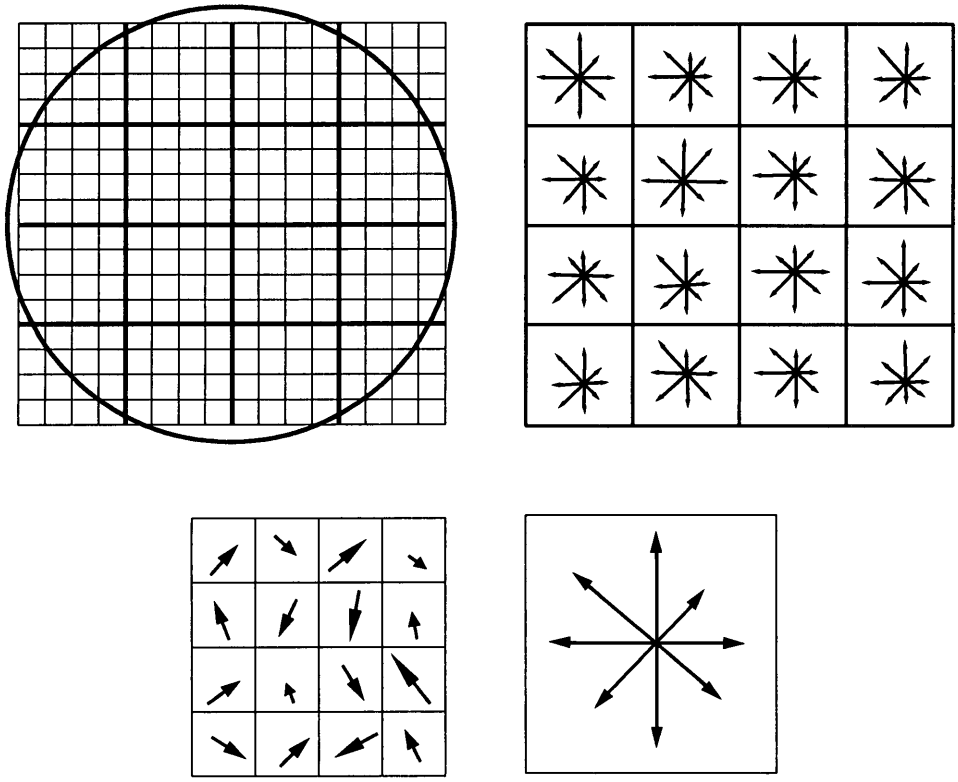


Fig. 2.3 Computing SIFT descriptor

A SIFT descroptor is computed from 16×16 sample area. The sample are wetighted by a Gausain window (the green circle). Then the area is divied into sixteen 4×4 subregieons. In each subregion, gradients are accumulated into an orientation histogramwith 8 bins representing the different orientations. In the end, a $16\times 8 = 128$ element vector is generated.

The SIFT descriptor is not fully affine invariant, though a 50 degree change in viewpoint still gives 50% matching accuracy. Brown and Lowe [37] developed a method to find

corresponding keypoints between images that contain large changes in viewpoint by using groups of interest points to form a geometrically invariant descriptor based on SIFT.

2.2 Speeded-Up Robust Features (SURF)

Speeded-Up Robust Features (SURF) is scale-rotation-invariant detector and descriptor developed by Bay and colleagues [30]. They claimed that SURF has close or better performance than previous descriptors in terms of repeatability, distinctiveness, and robustness. They pointed out that DoG in SIFT, which is an approximation of the Laplacian of Gaussian, increases computation speed but sacrifices accuracy. More importantly, SURF computes and matches faster than its precedents. This is due to the utilization of an integral image in interest point detection and descriptor calculations, and the simplified structure of the SURF descriptor (64 elements compared to 128 in SIFT). They pointed out that although SURF descriptor seems less distinctive in terms of dimensionality, its performance is comparable to or better than its predecessors, making it a superior descriptor that is robust to noise, detection errors, geometric and photometric deformations.

2.2.1 Integral images – the vehicle to speed-up calculation

SURF uses an integral image to increase computation speed. In an integral image, $I_\Sigma(\vec{x})$, the value of an entry $\vec{x} = (x, y)^T$ is the sum of all pixels in the input image I within the rectangular region formed by the origin and pixel $\vec{x}$,

$$I_\Sigma(\vec{x}) = \sum_{i=0}^{x}\sum_{j=0}^{y} I(i,j) \tag{2.10}$$

Integral image computation time is quadratic in the image size (length or width). But it is a onetime calculation which only involves additions as shown in Fig. 2.4. Most importantly, the resulting integral image can be used to perform a box filter calculation in constant time.

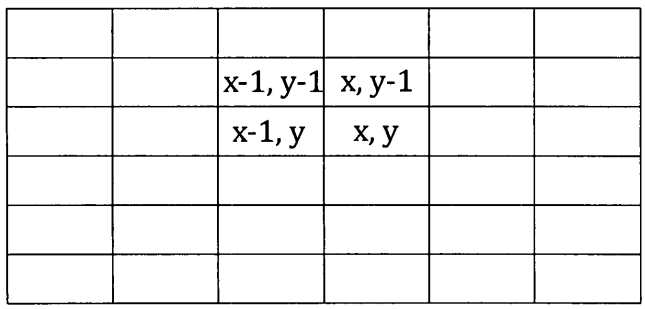


Fig. 2.4 Computing integral image

Given image I, to compute the value at location (x, y) in its integral image I_Σ, we need to do 3 additions: $I_\Sigma(x,y) = I_\Sigma(x,y-1) + I_\Sigma(x-1,y) - I_\Sigma(x-1,y-1) + I(x,y)$. The complexity is linearly proportional to the number of pixels in the image, but quadratic to its size (length or width).

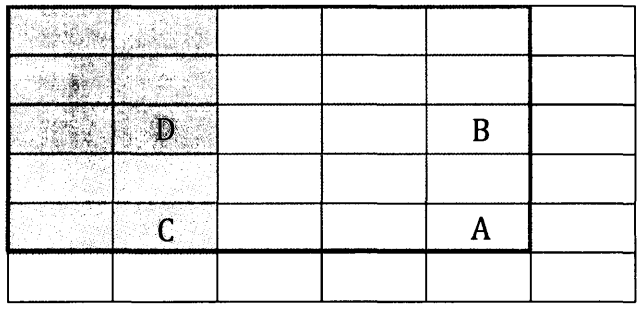


Fig. 2.5 Finding summation of a specific area in the image

Once we have the integral image, to find the summation of an area ABCD (colored in pink), we need 3 additions: 1) subtract area B from area A; 2) then add area D back to reform a complete area C; 3) then subtract area C.

In a box filter calculation, we need the sum of a rectangular neighborhood of pixels. Specifically, to obtain the summation of a rectangular area $Ret(abcd)$ as shown in Fig. 2.5, only three additions are required, namely,

$$\sum_{\vec{x} \in Ret(abcd)} (\vec{x}) = I_{\Sigma}(\vec{a}) - I_{\Sigma}(\vec{b}) - I_{\Sigma}(\vec{c}) + I_{\Sigma}(\vec{d}) \tag{2.11}$$

no matter how big this area is. Box filters, particularly those with large sizes, will be repeatedly used in the following interest point detection and descriptor calculation.

2.2.2 Scale-invariant interest point detection

SURF uses a basic Hessian matrix approximation for interest point detection. Blob-like structures are detected at locations where the determinant of the Hessian matrix is maximal. The scale selection also relies on determinant of the Hessian matrix, as described by Lindeberg [38]. For point $\vec{x} = (x, y)^T$ in an image I, the Hessian matrix $H(\vec{x}, \sigma)$ in $\vec{x}$ at scale σ is define as,

$$H(\vec{x},\sigma) = \begin{bmatrix} L_{xx}(\vec{x},\sigma) & L_{xy}(\vec{x},\sigma) \\ L_{xy}(\vec{x},\sigma) & L_{yy}(\vec{x},\sigma) \end{bmatrix} \tag{2.12}$$

where $L_{xx}(\vec{x},\sigma) = I(\vec{x}) * \frac{\partial^2}{\partial x^2} G_\sigma(\vec{x})$, is the convolution of the image I with the second derivative with respect to x of the Gaussian function with scale σ evaluated at point $\vec{x}$. Similarly, $L_{xy}(\vec{x},\sigma) = I(\vec{x}) * \frac{\partial^2}{\partial x \partial y} G_\sigma(\vec{x})$ and $L_{yy}(\vec{x},\sigma) = I(\vec{x}) * \frac{\partial^2}{\partial y^2} G_\sigma(\vec{x})$.

Because a complete Gaussian filter calculation is not practical in most cases, Bay and colleagues approximated L_{xx}, L_{xy}, and $L_{yy}$with box filters D_{xx}, D_{xy}, D_{yy} as shown in Fig. 2.6, such that,

$$L_{xx} \approx D_{xx} = \sum_{\vec{x}\in(X_1 \cup X_2)} I(\vec{x}) - 2 \times \sum_{\vec{x}\in X_3} I(\vec{x}) \tag{2.13}$$

$$L_{yy} \approx D_{yy} = \sum_{\vec{x}\in(Y_1 \cup Y_2)} I(\vec{x}) - 2 \times \sum_{\vec{x}\in Y_3} I(\vec{x}) \tag{2.14}$$

$$L_{xy} \approx D_{xy} = \sum_{\vec{x}\in(XY_1 \cup XY_2)} I(\vec{x}) - \sum_{\vec{x}\in(XY_3 \cup XY_4)} I(\vec{x}) \tag{2.15}$$

The summations in the above equations can be calculated easily by using the integral image as stated in Section 2.2.1. And the calculation time is independent of the filter size. Then the determinant of Hessian can be approximated as,

$$\det(H) = L_{xx}L_{yy} - {L_{xy}}^2 \approx D_{xx}D_{yy} - (wD_{xy})^2 \tag{2.16}$$

where w is the relative weight of filter responses used to balance the approximation. Although theoretically, w depends on the scale (σ) change, Bay and colleagues observed

that, keeping $w = 0.9$ constant did not have significant impact on the results [30]. Finally, the filter responses at a specific scale (filter size) are normalized with respect to the filter size.

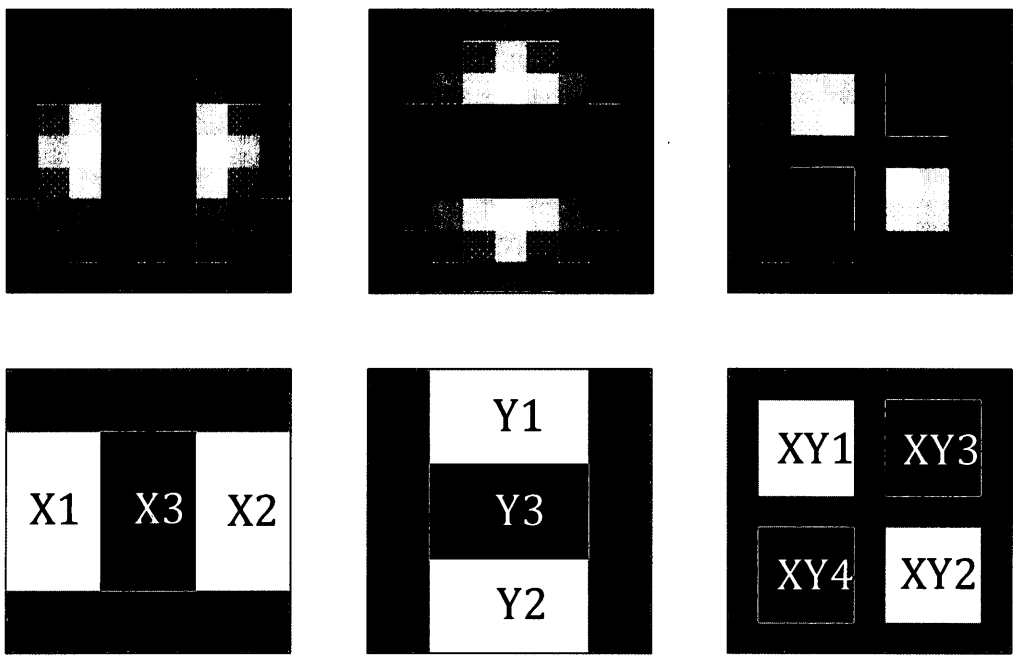


Fig. 2.6 Simple box filter calculations for Gaussian 2nd order derivative convolution

To estimate the convolution of Gaussian 2nd order derivative with an image at a specific location, we only need to find the summations of certain areas around this location from the integral image and do simple additions [30].

The lowest scale in SURF, referred to as scale $s = 1.2$ (the approximation of a Gaussian with $\sigma = 1.2$) are computed using 9×9 box filters. To build the scale space pyramid as in SIFT for scale space analysis (Section 2.1), Bay and colleagues used 9, 15, 21, and 27 as the sizes of box filters in the first octave. The size difference between two successive box filters is 6. For the minimal filter with size 9, a lobe size for calculating the second

derivative is $\frac{1}{3}$ of the filter size, which is 3. To increase the filter size, ensuring the presence of a central pixel for each lobe, the lobe size must increase by at least 2 pixels. Thus, the total filter size must be increased by a minimum of $3 \times 2 = 6$ pixels. For upper scale octaves, they derive a list of box filter sizes as shown in Fig. 2.7, which covers all possible discrete scale spaces.

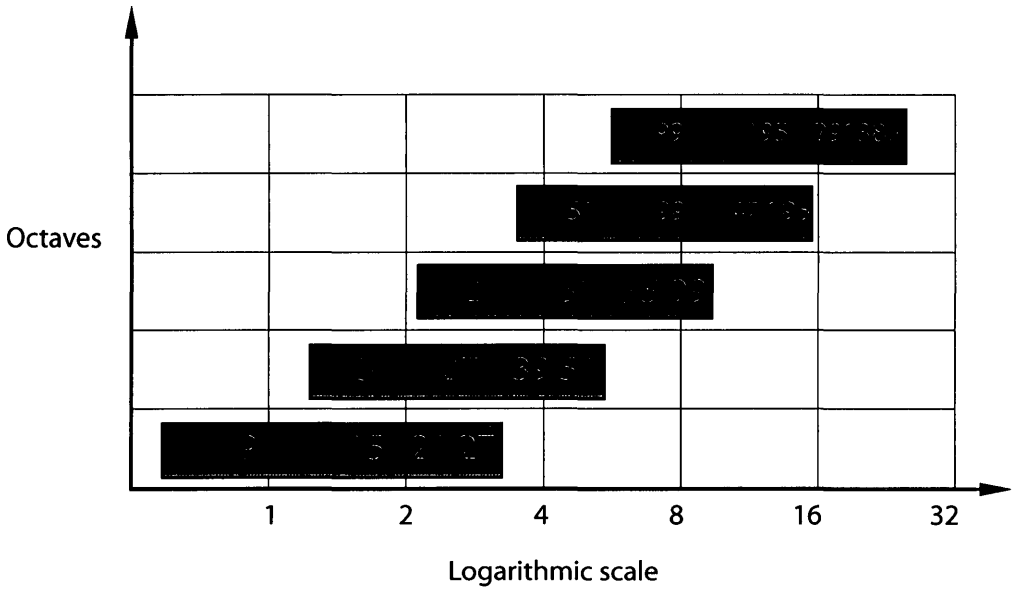


Fig. 2.7 Filter sizes for different level of octaves in discrete scale space

Analogous to SIFT octaves, in each octave, two levels of extremas are found in two groups of adjacent filtered images. In the first octave, extramas in level 15 are found using 9-15-21 images; for level 21, images 15-21-27 are used. Extremas in higher octaves are found in a similar fashion [30].

2.2.3 Interest point localization

For interest point localization over positions and scales, a non-maximum suppression [39] in $3 \times 3 \times 3$ neighborhood is used. Then the interpolations for accurate locations and scales are carried out by the same way as in Section 2.1.2.

2.2.4 Rotation invariance by orientation assignment

Bay and colleagues used first order Harr wavelet responses in x and y direction to describe the distribution of intensity around the interest point, instead of using gradient as in SIFT and its variants. This approach can exploit integral images found in the previous step, thus it can increase the overall speed. Harr wavelet is a sequence of rescaled "square-shaped" functions, which has the form like,

$$\psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \\ 0 & otherwise \end{cases} \tag{2.17}$$

In their case, as shown in Fig. 2.8, the intensity summation of area A, $I(A)$, is given the Harr wavelet function value 1 as coefficient, and the intensity summation of area B, $I(B)$, is given Harr wavelet function value -1 as coefficient. Then the Harr wavelet response with a specific direction (x or y) at a given location (the red star) is the summation, $I(A) - I(B)$. The Harr wavelet responses can be found with seven additions using the integral image as shown in Fig. 2.8.

To find the dominant orientation for an interest point at scale s, the Harr wavelet responses in both x, y directions within circular area of radius $6s$ around the interest point are first found. The side length of the wavelets are $4s$. Then the wavelet responses in both x, y directions are weighted with a Gaussian ($\sigma = 2s$) centered at the interest point.

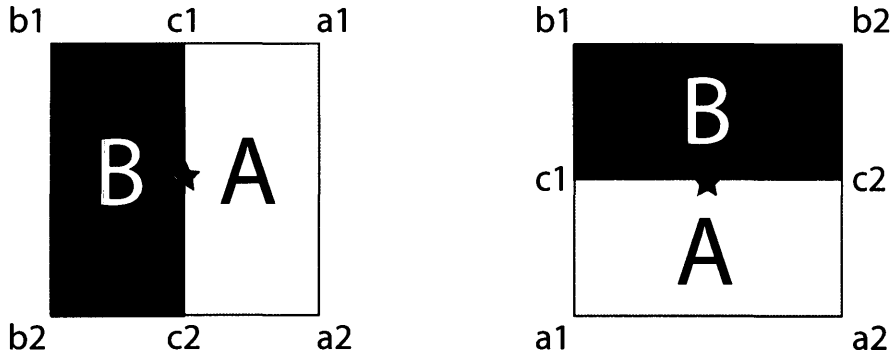


Fig. 2.8 Calculating Harr wavelet responses using integral image

To calculate Harr wavelet response at specific location (x, y), shown as red star in the figure, we need to find the difference between area A and area B, thus 7 additions are needed given the integral image I_Σ, likely,

$$\mathcal{H}(x,y) = I_\Sigma(a_2) - I_\Sigma(a_1) - I_\Sigma(c_2) + I_\Sigma(c_1) - \left(I_\Sigma(c_2) - I_\Sigma(b_2) - I_\Sigma(c_1) + I_\Sigma(b_1)\right)$$

Unlike SIFT, which uses an orientation histogram for accumulating sample magnitudes, SURF uses a $\frac{\pi}{3}$ sliding orientation window to accumulate the weighted x, y Harr wavelet responses. The two summed responses in the window then yield a local orientation vector. Then the longest orientation vector over all windows will define the orientation of the interest point. Bay and colleagues indicated, even without assigning a specific orientation to the interest point, SURF still maintains robustness to rotation of about $\pm 15°$ while giving faster computation and higher distinctiveness.

2.2.5 SURF descriptor

Like in SIFT, the SURF descriptor is a means to describe the properties of the local neighborhood around the interest point. Given the location, scale (s) and orientation θ found from previous steps, the next step is to construct a square region centered at the selected interest point and oriented with direction θ. The size of this window is $20s$.

Then the region is divided into 4× 4 subregions with each subregion containing 5 × 5 sample points as shown in Fig. 2.9. A Gaussian function, with $\sigma = 3.3s$ and centered at the interest point is applied to the wavelet responses in both x, y directions (relative to the interest point orientation) of all samples. The weighted wavelet responses of all samples in one subregion are accumulated as $\sum d_x$ and $\sum d_y$; the absolute value sums are also found as $\sum |d_x|$ and $\sum |d_y|$. Thus, for each subregion, a 4 element vector $\vec{v} = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right)^T$ is defined.

Combing all 4 × 4 subregion vectors, a 64 element descriptor vector is formed. Finally, this vector is normalized to achieve contrast-invariance. To make the descriptor more distinctive, an eight element vector $\vec{v} = \left(\sum d_x^{y-}, \sum d_x^{y+}, \sum d_y^{x-}, \sum d_y^{x+}, \sum |d_x|^{y-}, \sum |d_x|^{y+}, \sum |d_y|^{x-}, \sum |d_y|^{x+}\right)^T$ can be used to represent each subregion, where d_x^{y-} means x direction Harr wavelet response when y direction Harr wavelet response is negative. Because SURF integrates the gradient information within a subpatch, whereas SIFT depends on the orientations of the individual gradients, the result is better noise resistance in SURF compared to its SIFT-like counterparts.

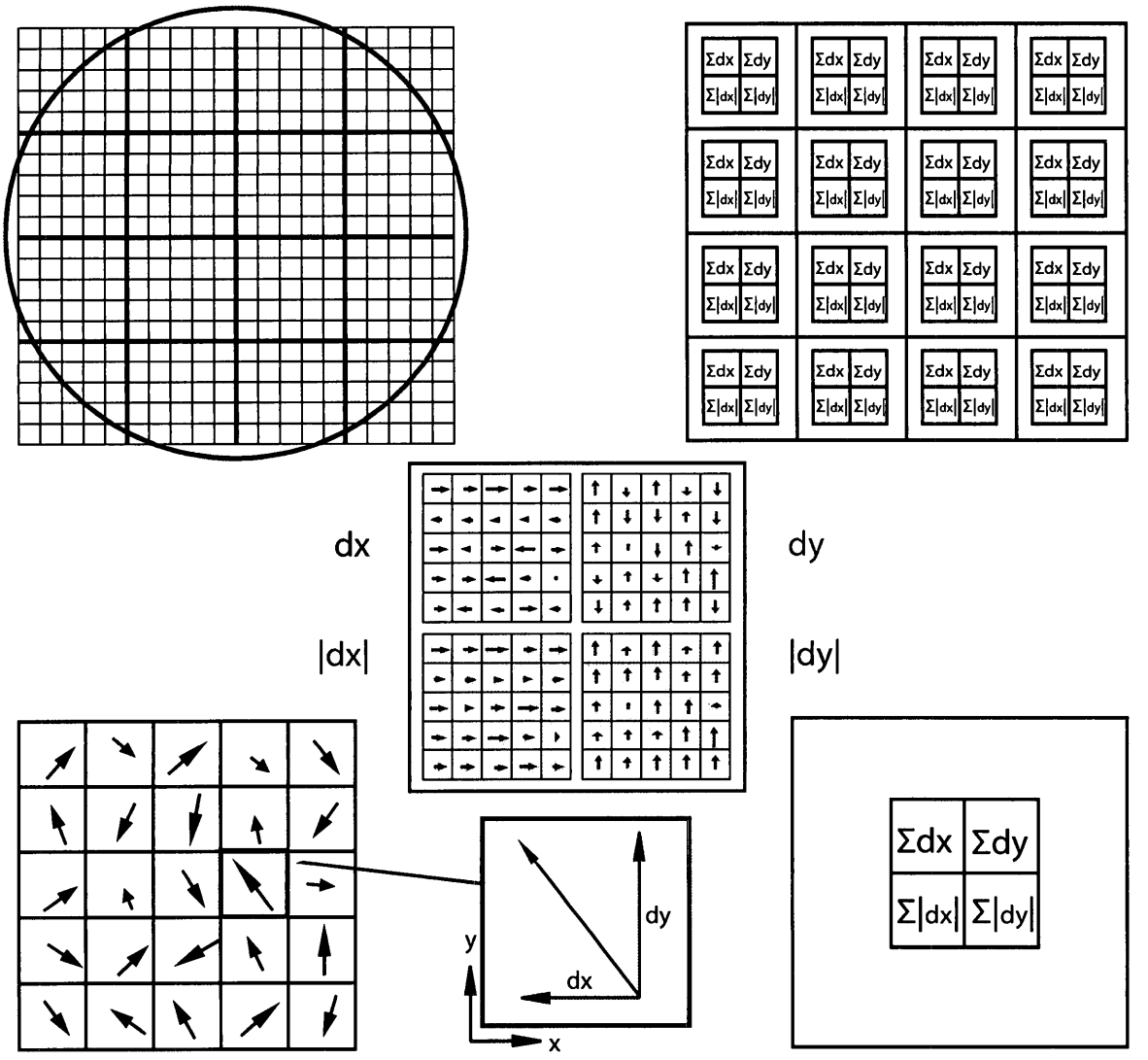


Fig. 2.9 Finding SURF descriptor

First, the Harr wavelet responses (dx, dy) of 20x20 sample points are Gaussian weighted (green circle) centered at the interest point. Then, the sample area is divided into 4x4 subregions. In each 5x5 subregion, wavelet responses and their absolute values are accumulated as $\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|$. In the end, a descriptor vector consists of 4x4x4=64 components.

2.3 Markov random field (MRF) and its optimizers

In early vision, problems involving pixel-labeling tasks (Fig. 2.10) such as stereo matching [40], image stitching [41], image segmentation [42], image denoising and

inpainting [43] [44] can be elegantly expressed as Markov random fields, given the fact that neighboring pixels have interactions among each other. However, MRF energy minimization problems have long been considered intractable; minimization with previous approaches such as iterated conditional modes (ICM) [45]or simulated annealing has been shown to be either ineffective or inefficient [43]. Recently, powerful optimizers such as loopy belief propagation (LBP) and graph cuts have proven to be more accurate and faster [43].

Fig. 2.10 Pixel labeling applications using Markov Random Field

First row, stereo matching results with stereo disparity shown as greyscale images (darker color indicates further position) [43]; second row, segmentation of extracting foreground objects [42]; third row left, photomontage by stitching multiple views into one big image [41]; third row right, denoising and inpainting [43]

2.3.1 Markov random field

A Markov random field, or undirected graphical model, is a set of nodes each of which corresponds to a variable or a group of variables, as well as a set of links each of which connects a pair of nodes. Links in MRF are undirected, and loops can be present. In a MRF, variables or groups of variables maintain the Markov property, namely, the future state of a node depends only on the states of its neighbors despite the sequence of events that preceded it. To be specific, in a MRF model, the following equivalent Markov properties must be satisfied,

1) Any two non-adjacent variables are conditionally independent given all other variables.
2) A variable is conditionally independent of all other variables given its neighbors.
3) Any two subsets of variables are conditionally independent given a separating subset.

In 2D image analysis, it is convenient to assume that a 4-neighbor MRF model is satisfied. The state (label, for example) of a pixel is only affected by its direct neighbors but not any further pixels.

2.3.2 Iterated Conditional Modes (ICM)

Iterated Conditional Modes is a deterministic greedy algorithm for obtaining local minimum. It starts with an estimated initial configuration, then for each pixel, it chooses a new configuration that gives the largest decrease of the energy function, then iterates

until convergence is reached. Although this algorithm guarantees rapid convergence, it is extremely sensitive to the initialization; thus, it is not practical to find global solutions.

2.3.3 Graph Cuts

Graph cuts algorithms repeatedly compute global minimum of binary labeling obeying max-flow/min-cut theorem. They converge rapidly to a strong local minimum. Although graph cuts algorithms can find an exact global solution only for binary labeling problems, they can be also used to obtain near global optimum for problems with more than two labels. Two most popular graph cuts algorithms are the swap-move algorithm and the expansion-move algorithm [43].

2.3.4 Loopy Belief Propagation (LBP)

Belief propagation (BP) is a message passing algorithm that calculates marginal distributions of unobserved nodes, given conditions on the observed nodes. Exact BP algorithm can only be used to find marginal distributions for the simplest form of graph – tree. For a general graph with loops, an approximate BP algorithm called loopy belief propagation (LBP) can be applied by slightly adjusting the message initialization and message passing procedures. There are two approaches of LBP, max-product LBP and sum-product LBP. The former was designed for finding the lowest energy solution, yet the latter one only computes marginal distributions for each node. Szeliski and colleagues compared two different LBP algorithms, BP-M (a max-product LBP) and BP-S (an LBP derived from TRW). They found that in a binary segmentation application, BP-S

outperforms BP-M significantly in terms of speed, yet the minimal energy of BP-S is slightly higher than the one of BP-M [43]. Loopy believe propagation will be further discussed in Section 2.4.6.

2.3.5 Tree-Reweighted Message Passing (TRW)

Tree-reweighted message passing looks similar to LBP, but it can compute a lower bound on the energy. Thus, one can use the lower bound to assess the quality of solutions found by other optimizers. However, the original TRW algorithm does not guarantee that the lower bound always increases with time. Szeliski and colleagues developed an improved TRW algorithm called sequential TRW (TRW-S), in which the lower bound estimate is guaranteed not to decrease [43]. Since the energy can oscillate in practice, one can keep track of the updated lowest energy and return it when the algorithm stops.

2.4 SIFT-flow image alignment algorithm

SIFT flow algorithm is a dense correspondence, pixel-to-pixel image alignment method. It is used to solve a challenging alignment problem, *scene alignment*, in which object categories sharing similar characteristics will be aligned in two images even if they appear differently in the scene. In addition to its success in solving scene correspondence problems, SIFT flow can also give comparable or better results in traditional image alignment applications [1].

2.4.1 Image alignment

Image alignment or image registration, is the process of transforming or integrating multiple sets of data obtained from different measurements into a common coordinate frame such that comparison or analyses across different times or under different conditions can be conducted.

A common approach for solving the image alignment problem involves finding correspondences across different views. In these cases, it is often considered that corresponding pixels have similar intensity after the images are aligned. These applications include image stitching [41] [46] and stereo matching [40] (Fig. 2.10). A more complicated alignment problem is video sequence analysis which is often achieved by estimating optical flow between two temporally adjacent frames [47] [48]. Compared to the geometric parametric motion in image stitching and 1D disparity in stereo matching, the 2D flow vector gives higher level of complexity. In object recognition, image alignment becomes even more difficult because of the variation of possible shapes, sizes, and appearances within an object class.

Liu and colleagues [31] attempted to solve the scene alignment problem in which they tried to align two images from different 3D scenes sharing similar scene characteristics. The objects in the two images may be under different viewpoints, located in different sites, and/or -having different scales. Furthermore, there may be different quantities of objects from the same category in the two images. To solve this challenging problem,

they proposed a method analogous to optical flow called *SIFT Flow dense scene correspondence*.

Instead of simply using pixel intensities as is done in optical flow, SIFT flow relies on the scale-rotation-invariant SIFT descriptor. Unlike optical flow in which adjacent frames in a video sequence are used as closest neighbors for finding meaningful correspondence, the SIFT flow closest neighbors are those with best feature matches, which can become semantically meaningful if the inspected images include large collection of possible scenes in the world.

More importantly, when SIFT flow is applied to the regime of traditional image alignment, comparable or even better results can be obtained [31]. Recently, we used the SIFT-flow algorithm for warping x-ray images, with promising results. Fig. 2.11 and Fig. 2.12 show the warping result of SIFT-flow algorithm on hand and hip x-ray images, respectively.

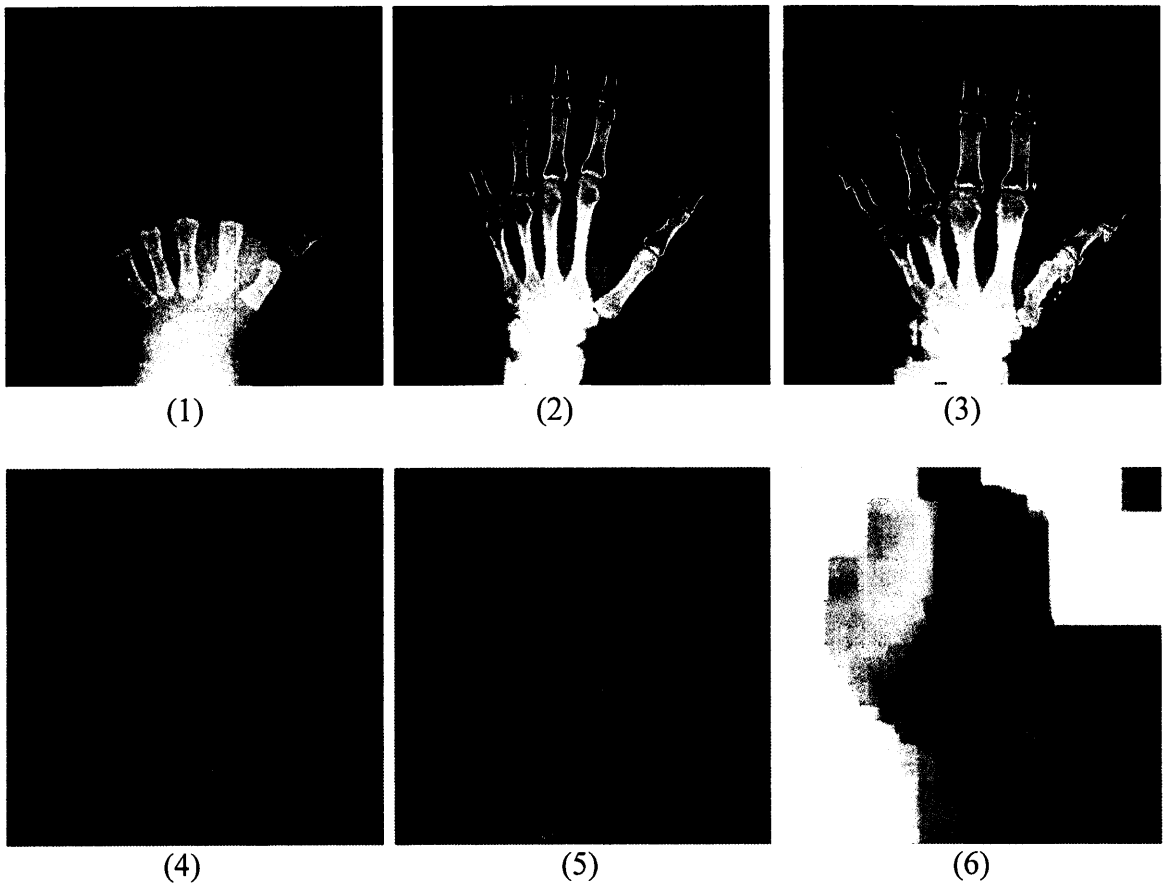


Fig. 2.11 SIFT flow visualization and pixel-to-pixel alignment results on hand images

(1) Query image [2] (2) Template image [2] (3) After template image was warped to query image (4) SIFT feature visualization for query image (5) SIFT feature visualization for template image (6) SIFT flow visualization using color scheme of Baker and colleagues [49]: orientations are characterized by color hue, while magnitudes are represented by saturations. For example, the white parts in (6) means no significant flow field, as the pixels of the query image and template image in these parts tend to be one-to-one similar.

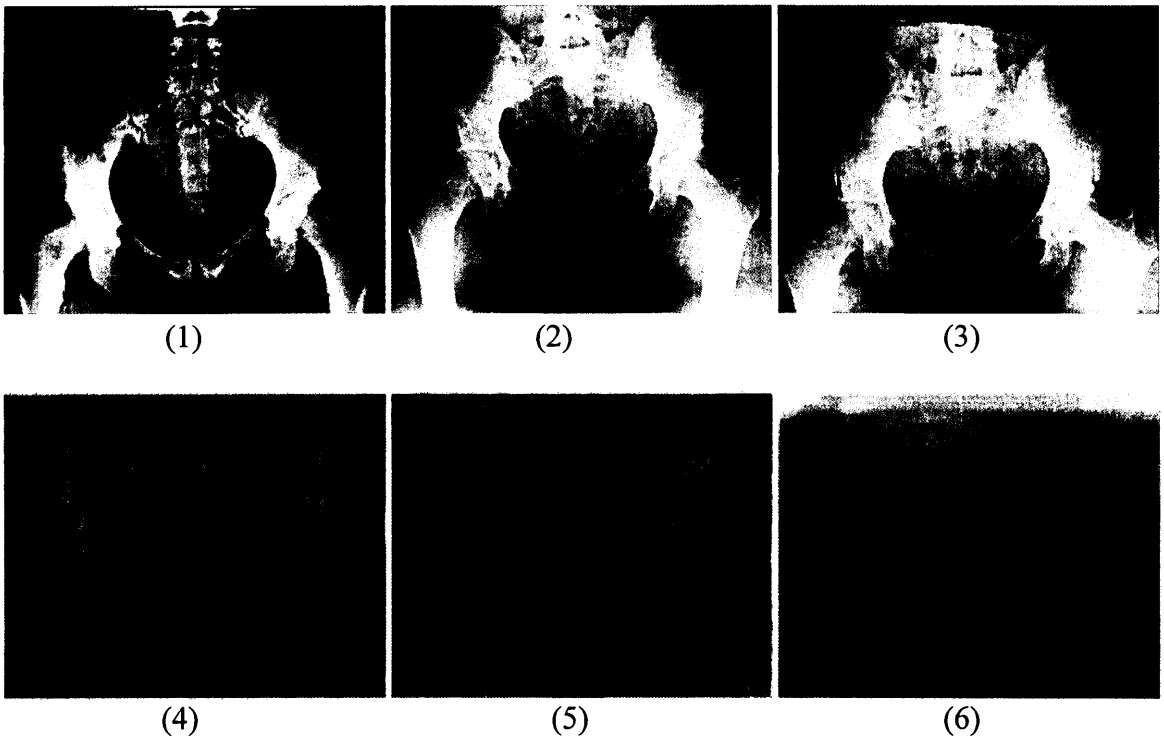


Fig. 2.12 SIFT flow visualization and pixel-to-pixel alignment results on hip images

(1) Query image [7] (2) Template image [50] (3) After temple image was warped to query image (4) SIFT feature visualization for query image (5) SIFT feature visualization for template image (6) SIFT flow visualization using color scheme of Baker and colleagues [49]: orientations are characterized by color hue, while magnitudes are represented by saturations. For example, the homogenous purple color in (6) means the flow movements of most pixels have similar directions (downwards). The fact that the lower part is more saturated indicates the movements of the pixel there are larger.

2.4.2 Optical flow and its estimation

When an object captured in an image sequence moves, its brightness patterns raised in the sequence will move accordingly. The apparent motion of the brightness pattern is called optical flow.

The basic assumption for estimating optical flow is *brightness constancy* which states

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{2.18}$$

where $I(x, y, t)$ is the intensity of an image pixel located at (x, y) at time t, $I(x + u, y + v, t + 1)$ is the intensity of the corresponding pixel in the image taken at time $t + 1$ at position $(x + u, y + v)$ with u, v defined as the offset in x and y direction, respectively. However, in real life, slight changes of brightness may often happen in natural scenes. To address this issue, we can utilize another assumption called *gradient constancy* to allow for small changes in brightness,

$$\nabla I(x, y, t) = \nabla I(x + u, y + v, t + 1) \tag{2.19}$$

where $\nabla = (\partial_x, \partial_y)^T$ is the vector differential operator and $\nabla I(x, y, t)$ is the spatial gradient.

Another assumption is the *smoothness of flow field*, which is used to address the aperture problem (i.e., only the normal or perpendicular direction of a flow is detectable within an aperture window [51]) and the existence of outliers. In general, a piecewise smooth flow field is required because of the fact that boundary discontinuities may occur in optimal flow estimation.

Based on the above assumptions, the energy function for finding optical flow can be stated as,

2.20 $$E(u, v) = E_{data} + \alpha E_{smooth} \tag{2.20}$$

with

$$E_{data}(u,v) = \sum_{\Omega} (|I(\vec{x}+\vec{w}) - I(\vec{x})|^2 + \gamma|\nabla I(\vec{x}+\vec{w}) - \nabla I(\vec{x})|^2) \quad (2.21)$$

$$E_{smooth}(u,v) = \sum_{\Omega} (|\nabla_3 u|^2 + |\nabla_3 v|^2) \quad (2.22)$$

where $\vec{x} = (x, y, t)^T$ is the location (x, y) in time t, $\vec{w} = (u, v, 1)$ is the optical flow vector during unit time interval, $\nabla_3 = (\partial_x, \partial_y, \partial_t)^T$ is the spatio-temporal gradient [52].

As expected, using the above objective function can lead to local minima trapping. To achieve the global minimum, it can be useful to apply a *coarse-to-fine* strategy. Starting with coarsely smoothed problem, the first solution is found and used in the finer level problem as the initialization. After several coarse-to-fine iterations till the original problem is reached, a close to global solution can be obtained (more detail can be seen in Section 2.4.8).

2.4.3 Dense SIFT descriptors

As stated in Section 2.1, SIFT is scale-rotation-invariant. In most applications of SIFT, scale-rotation invariant keypoints at (x, y, σ) are found after extrema detection, localization and orientation assignments. Then SIFT descriptors of the keypoints are calculated. The amount of keypoints is limited (sparse) compared to the pixel count of the image; hence, robust matching and transformation can be performed efficiently.

However, to utilize the SIFT descriptor in optical flow estimation, SIFT descriptors of all pixels in the image are required, which does not involve extrema detection. The per-pixel

SIFT descriptor image s_i $(width \times height \times 128)$ is called the dense SIFT image [31]. Fig. 2.13 shows the dense SIFT images of hand and hip x-ray photographs.

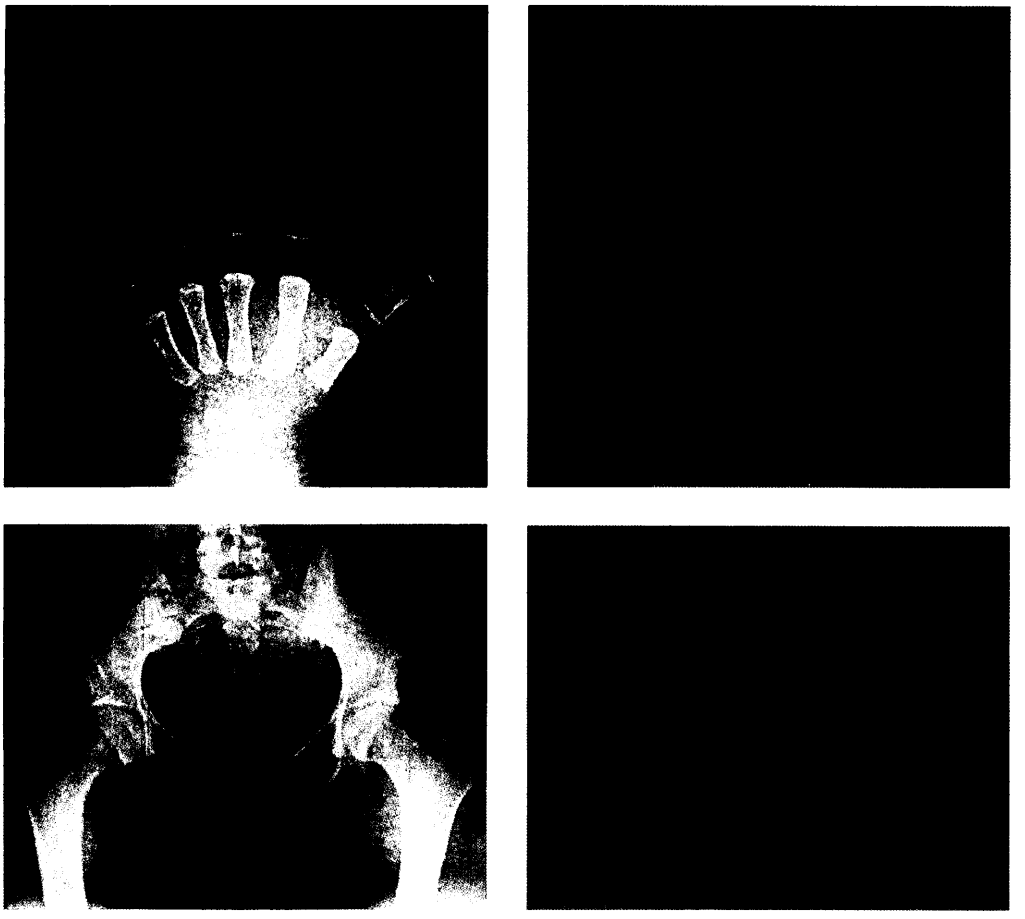


Fig. 2.13 Visualization of dense SIFT images

The 128-D SIFT feature descriptor of each pixel are projected to RGB color space; pixels with similar colors may imply they are with similar structures [2] [50]

2.4.4 SIFT flow estimation objective function

Inspired by the optical flow estimation such as one mentioned in Section 2.4.2 and utilizing dense SIFT image, Liu and colleagues formulated the SIFT flow objective function as following,

$$E(\vec{w}) = \sum_{p} min\left(\left\|s_1(\vec{p}) - s_2\left(\vec{p} + \vec{w}(\vec{p})\right)\right\|_1, t\right) +$$

2.23

$$\sum_{p} \eta(|u(\vec{p})| + |v(\vec{p})|) + \tag{2.23}$$

$$\sum_{(p,q)\in\varepsilon} \left(min\left((\alpha|u(\vec{p}) - u(\vec{q})|), d\right) + min\left((\alpha|v(\vec{p}) - v(\vec{q})|), d\right)\right)$$

where $\vec{p} = (x, y)^T$ is the coordinate of a sample point, $\vec{w}(\vec{p}) = \left(u(\vec{p}), v(\vec{p})\right)$ is the flow vector of $\vec{p}$, $s_1$and s_2 are the per-pixel dense SIFT images of the query image and template image, respectively, ε is the neighborhood (4 neighbors in 2D) of $\vec{p}$. The first term is the *data term* which substitutes the SIFT descriptor for intensity in Equation 2.20. The third term is the *smoothness term*, accordingly. The thresholds t and d are used to address the matching outliers and flow discontinuities. The second term, called the *small displacement term*, is used to constrain the flow vectors to be as small as possible when the data term and smoothness term do not give significant contributions.

2.4.5 Solving SIFT flow estimation

To optimize the objective function above, decoupled sequential loopy belief propagation (BP-S) with distance transformation technique is applied by Liu and colleagues. As commonly used in solving optical flow problems, a coarse-to-fine scheme is used to speed up the global solution search process. In the following sections, dual-layer sequential loopy belief propagation and the coarse-to-fine scheme flow matching scheme are further discussed.

2.4.6 Achieving Loopy Belief Propagation

Consider a simple image, shown in Fig. 2.14, having three pixels p_1, p_2, and p_3. We wish to assign labels f_1, f_2, and f_3 to the pixels where the labels are chosen from a discrete set of values L. The choice of labels depends on two factors: (a) the observed pixel intensities, and (b) a model of how labels vary over an image.

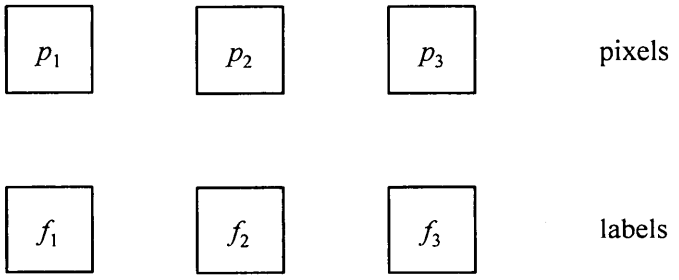


Fig. 2.14 A simple 3-pixel image example for label assignment

The three pixels p_1, p_2, and p_3 are arranged linearly. Their corresponding labels are depicted as f_1, f_2, and f_3 below them.

The relationship between the observed intensity for pixel p_i and its corresponding label f_i can be modeled using an energy cost function $D_i(f_i)$ which is the cost of assigning label f_i to pixel p_i. The function D_i is called the data term because it relates the observed pixel intensity values (the data) to the labels. The relationship between pixel intensities and labels is shown in Fig. 2.15.

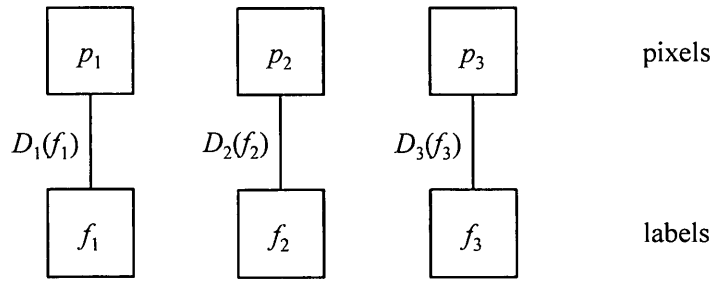


Fig. 2.15 Data term functions in the 3-pixel example

The three pixels p_1, p_2, and p_3 are arranged linearly. Their corresponding labels f_1, f_2, and f_3 are determined by the data term energy cost functions $D_1(f_1)$, $D_2(f_2)$ and $D_3(f_3)$.

In an image where each object has a uniform intensity and different objects have different intensities, we would expect labels within an object to be the same, and we would expect labels to change at the boundaries between objects; i.e., the choice of label for a pixel depends on the labels assigned to the nearby pixels. In a Markov random field model, the label for a pixel depends only on the labels of the immediately adjacent pixels. For our example image, label f_1 depends on label f_2, but not label f_3. Similarly, label f_3 depends on label f_2, but not label f_1. Label f_2 depends on labels f_1 and f_3. The relationship between labels for adjacent pixels p_i and p_j can be modeled using an energy cost function $V(f_i, f_j)$. The function V is called the smoothness term or the discontinuity cost because it models the relationship between labels on adjacent pixels. The relationship between adjacent pixel labels is shown in Fig. 2.16.

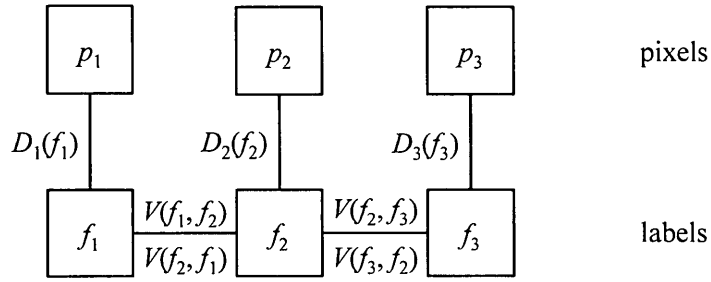


Fig. 2.16 Smoothness term functions in the 3-pixel example

The three pixels p_1, p_2, and p_3 are arranged linearly. Their corresponding labels f_1, f_2, and f_3 are determined not only by the data term energy cost functions $D_i(f_i)$, but also by the smoothness term functions $V(f_i, f_j)$.

For our example image, the total cost of the labeling f all of the pixels is given by

$$E(f) = V(f_1, f_2) + V(f_2, f_1) + V(f_2, f_3) + V(f_3, f_2) + D_1(f_1) + D_2(f_2) + D_3(f_3) \tag{2.24}$$

and the best labeling is the one that minimizes $E(f)$.

Belief propagation is an iterative algorithm that solves the overall energy minimization problem by repeatedly solving energy minimization sub-problems at each label node; the minimized energy for each sub-problem is communicated to adjacent label nodes in a process called message passing. The details for computing an individual message are described in [44]. Fig. 2.17 shows the messages passed at iteration t between nodes for our 3-pixel labeling example. For a Markov network without loops, it can be shown that the messages rapidly converge to a fixed value. The best label for each node can then be computed locally at each node by considering only the messages incoming to the node

and the data cost at the node. In two dimensional images, messages are usually passed between nodes adjacent vertically and horizontally, as shown in Fig. 2.18.

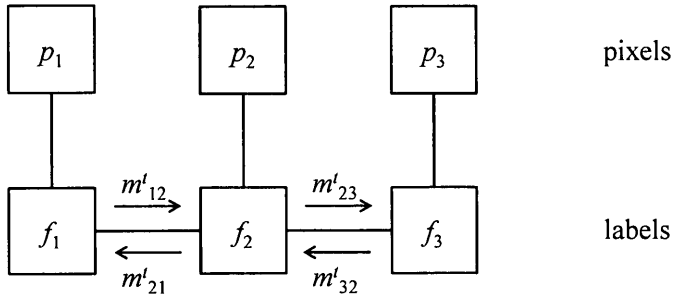


Fig. 2.17 Belief propagation message passing in the 3-pixel example

At iteration t, node f_1 receives message m^t_{21} from node f_2; node f_2 receives message m^t_{12} from node f_1 and message m^t_{32} from node f_3; and node f_3 receives message m^t_{23} from node f_2.

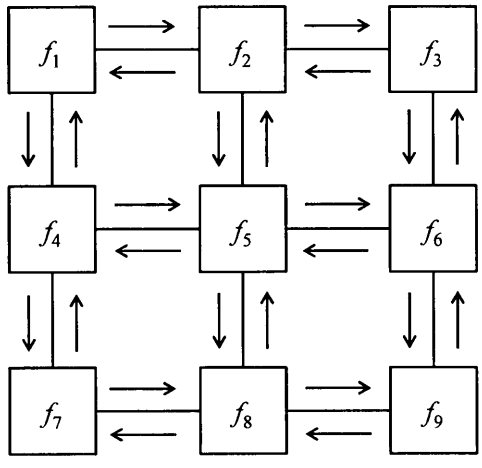


Fig. 2.18 Belief propagation message passing in a 2D image

In two dimensional images, messages are usually passed between nodes adjacent vertically and horizontally.

In the classic belief propagation algorithm, the messages computed at iteration t are independent of one another, and are dependent only on messages computed during the

previous iteration. More precisely, let m_{pq}^{t} be a message from node p to node q at iteration t; then:

- m_{pq}^{t} depends only messages from iteration $t-1$
- m_{pq}^{t} does not depend on message m_{qp}^{t-1}
- m_{pq}^{t} depends on all other messages m_{xp}^{t} where node x is adjacent to node p

The dependencies for m_{56}^{t} are shown in Fig. 2.19. Note that the order in which messages are computed during a single iteration is unimportant.

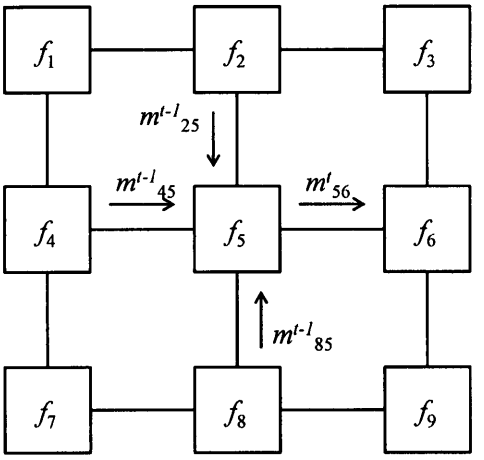


Fig. 2.19 Message passing of an example node in a 2D image images

Node f_5 receives message m_{25}^{t-1} from node f_2, message m_{85}^{t-1} from node f_8 and message m_{45}^{t-1} from node f_4. Then message m_{56}^{t} can be computed based on these messages. Note that message m_{65}^{t-1} from node f_6.

Variations of the classic belief propagation algorithm allow interdependencies between messages passed during a single iteration to increase the speed of convergence. For

example, in our 3-pixel problem, we could compute messages from left to right, and then from right to left: During iteration t, we first compute m_{12}^t; next, we use m_{12}^t to compute m_{23}^t; next, we could compute m_{32}^t; finally, we use m_{32}^t to compute m_{21}^t. If interdependencies between messages passed during a single iteration are allowed, then the order in which messages are computed can affect the final result. The order in which messages are computed is called the message passing schedule. For images, a commonly used schedule is to compute messages row-wise and then column-wise; i.e., messages are computed from left to right, then right to left, then top to down, and finally down to top. This variation is the BP-M algorithm.

Yet another variation for images to compute the messages in scanline order. Starting at the first node on the first row, messages are passed to the right and bottom neighbors. The process is repeated with the second node on the row, and so on, until messages have been computed for every node on the row. Then, the second row is processed, and so on, until the last row is processed. The process then reverses, this time working in reverse scanline order and passing messages to the left and top neighbors for each node. This variation is the BP-S algorithm used by Liu and colleagues [1] to solve the label transfer optimization problem; we refer to this algorithm as BP-S-Liu.

A slight modification of BP-S was proposed by Szeliski and colleagues [43]. Their version uses the message passing schedule of BP-S, but modifies the calculation used to compute the optimal label at a node. The normal calculation used to compute the optimal

label at a node p considers all of the messages incoming to node p. Szeliski's version uses only the messages incoming to node p that originate from nodes that come after p in scanline order. The reason for this modification is that the messages originating from nodes after p already contain information about the messages from nodes prior to p. We refer to this algorithm as simply BP-S.

2.4.7 Dual-layer sequential loopy belief propagation

Due to the topological relationships between neighboring pixels in an image, it is straight forward to consider minimizing the energy function like Equation 2.20 based on graphical models. For the smoothness term and the small displacement term in SIFT flow estimation, one can exploit the property that continuity interaction (smoothness) is separable (sum of horizontal and vertical interactions) and can decouple the energy model as two interacting fields of a scalar variable [53] as shown in Fig. 2.20. The original neighboring graph is transformed to a dual-layer graph, with nodes T becoming nodes V^1 and V^2, edges E becoming $\mathcal{E}^1, \mathcal{E}^2$ and $\mathcal{E}^{12}$. The complexity is reduced from $|T| \cdot |L|^2$ to $2 \cdot |T| \cdot |L|$, where $L = \{d_{min} \dots d_{max}\}$ is the possible displacement of a pixel in either horizontal (d_u) or vertical (d_v) direction. It is easy to see, before decoupling, the possible displacement of a pixel (hence number of nodes in a graph) is $|L|^2$; but after decoupling, the number of nodes to represent a single pixel displacement becomes $2 \cdot |L|$. The factor graph for the energy function in Equation 2.23 can be represented as in Fig. 2.21.

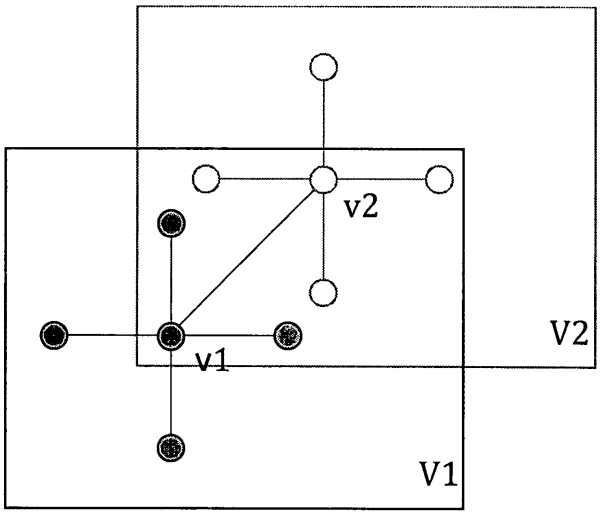


Fig. 2.20 Decoupling horizontal and vertical interactions

Picture shows the neighborhood structure of two vertices v_1 and v_2 and the edges connecting their neighbors [53].

Inference of graphical models can be done by techniques such as, belief propagation, graph cuts, Markov chain Monte Carlo (MCMC), simulated annealing and so on. Liu and colleagues [31] used loopy belief propagation as the base algorithm to optimize the objective function in Equation 2.23 because loop structures exist in the model. However, loopy belief propagation does not guarantee convergence at all due to the existence of cycles (loops) [54] [55]. To address this issue, sequential belief propagation (BP-S), which is a loopy belief propagation implementation derived from sequential Tree-Reweighted Message Passing (TRW-S), is used for better convergence [43].

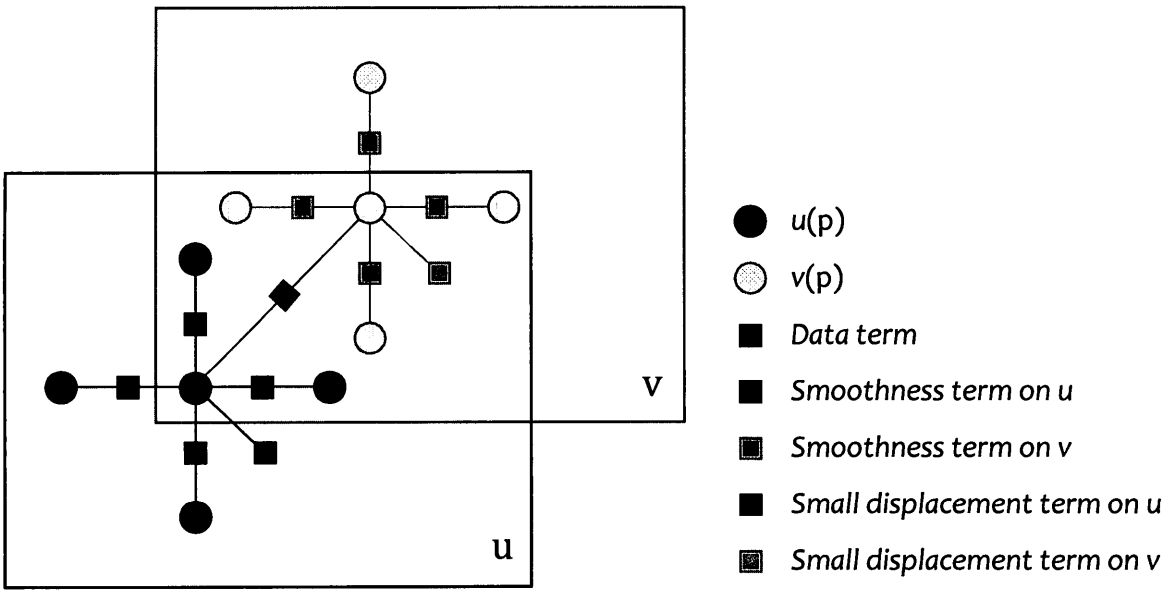


Fig. 2.21 Factor graph – decoupling u, v in SIFT flow Energy function [31]

2.4.8 Coarse-to-fine flow matching

Even though the dual-layer loopy propagation significantly reduces the computational complexity from $|T| \cdot |L|^2$ to $2 \cdot |T| \cdot |L|$, the calculation still scales poorly. When we consider that displacement can happen to any position in an image, $2 \cdot |T| \cdot |L|$ becomes $2|T|^2 = 2|h^2|^2 = 2|h|^4$, where h is the length of the image. Complexity of dual-layer belief propagation is $O(h^4)$.

To speed up performance, Liu and colleagues incorporated the coarse-to-fine idea into their SIFT flow matching technique. First, SIFT pyramids $\{s_1^{(k)}\}$ and $\{s_2^{(k)}\}$ for two SIFT images s_1 and s_2 are generated, where $s_i^{(1)} = s_i$ and $s_i^{(k+1)}$ is smoothed and downsampled from $s_i^{(k)}$. At level k, let p_k be the coordinate of a pixel on $s_1^{(k)}$ to match,

c_k be the offset or centroid of the searching window in $s_2^{(k)}$, and $w(p_k)$ be the best match from belief propagation. An example of 3-level pyramid is shown in Fig. 2.22. At the top level, the search window of p_3 is centered at $c_3 = p_3$, and the window size is $m \times m$, where m is the width or height of $s_1^{(3)}$. The complexity of belief propagation is $O(m^4)$. Once convergence is reached, flow vector $w(p_3)$ is found. In finer level, c_2 is determined by propagating p_2 with $w(p_3)$, but the searching window size is fixed to be 11×11. The process is iterated from $s_i^{(3)}$ to $s_i^{(1)}$ until the flow vector $w(p_1)$ is determined. The complexity of the coarse-to-fine approach is reduced from $O(h^4)$ to $O(h^2 log h)$. To further reduce computational complexity by exploiting the truncated L1 norm in Equation 2.23, a modified distance transform function [44] to cope with coarse-to-fine scheme is developed in Liu's SIFT flow algorithm.

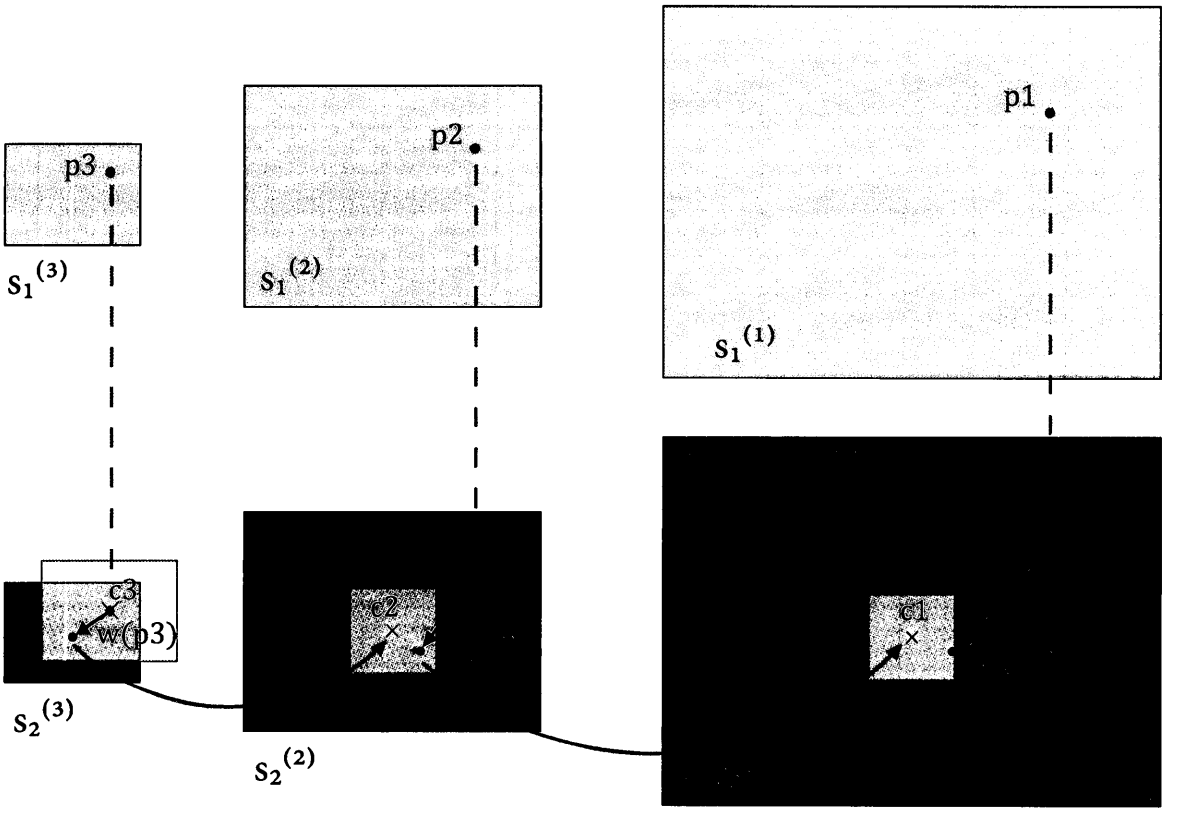


Fig. 2.22 Coarse-to-fine SIFT flow matching pyramid

2.5 Label Transfer System

Even though active research heavily focuses on establishing mathematical models for images, scenes and objects to achieve object recognition and image understanding, Liu and colleagues proposed a nonparametric approach for scene parsing, called Label Transfer. First, the Label Transfer system extracts nearest neighbors for a query image from a large annotated image database. Then the SIFT flow algorithm is used to obtain dense correspondences between the query image and the nearest neighbors. Finally, an MRF model based on the dense correspondence, object location priors, and pair-wised

smoothness is used to segment and recognize the query image. In this section, we first introduce a web-based annotation tool developed by Russell and Torralba [56] for easy image annotation by web service and instant access via Matlab®. Next, the three steps of Label Transfer image parsing will be discussed.

2.5.1 LabelMe web-based tool for image annotation

Seeking to build a large collection of images with ground truth labels for object recognition research, Russell and Torralba developed an easy-access, open and dynamic annotation system called LabelMe. LabelMe takes into account of object-part hierarchy and occlusion, which allow polygons with a high degree of overlapping.

The annotation user interface of LabelMe is shown in Fig. 2.23. First, the image requiring annotation is uploaded to the system. Then the labeler can create polygons to encapsulate objects and name them. During polygon drawing, labelers can make adjustments or start over. After closing the polygons, labelers can relocate vertices, delete or rename the polygons. Once all objects are annotated, the annotation information can be downloaded as a structural XML. In the end, the system will store the pair of image and annotation (XML) files as a new data entry. In the future, researchers or programmers can use the LabelMe Matlab package to retrieve image data and annotation information.

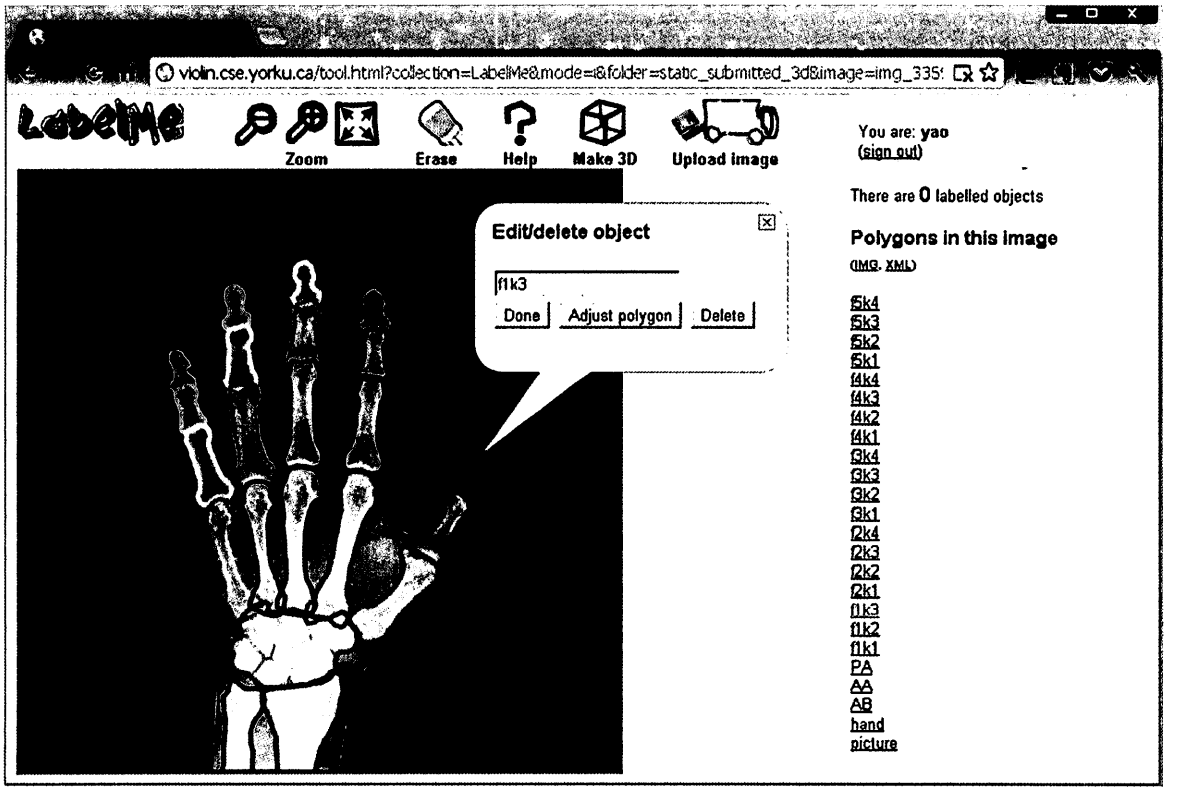


Fig. 2.23 LabelMe annotation user interface

2.5.2 Neighborhood and scene retrieval

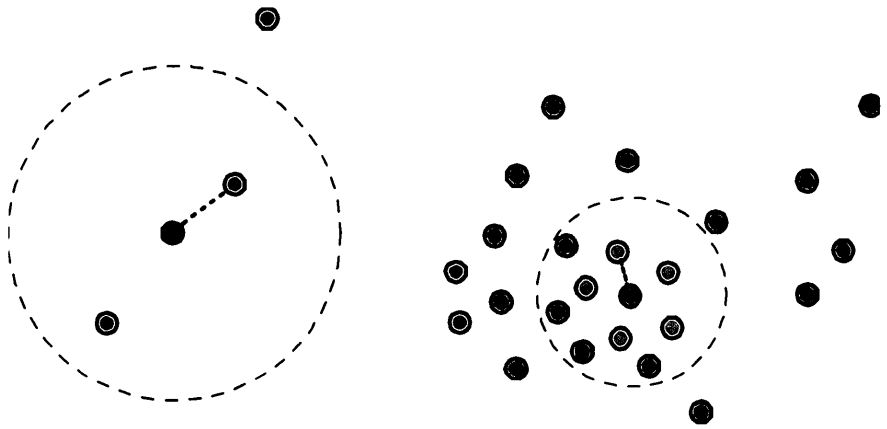
For a query image, before retrieving the most similar images from a large annotated image database, one needs to define the measurement of image similarity. We call this measurement the distance between two images, denoted as $dist(image1, image2)$. There are several descriptors that can be used for measuring the scene distance such as GIST [57] [58], HOG [59], or the dense SIFT described in Section 2.4.3. Although it has been reported that various nearest metrics do not result in significant difference in obtaining nearest neighbors for matching [31], Liu and colleagues found that scene retrieval based on GIST gives best accuracy in the Label Transfer System [1]. The GIST

descriptor is obtained by computing output energy of Gabor-like filters that are tuned into varied orientations and scales [60]. GIST is considered to be able to reliably estimate the dominant spatial structure of a scene, such as naturalness, openness, roughness, expansion and ruggedness [57]. GIST is a low-dimensional representation of a scene, which can be easily computed without identifying specific regions or objects in the image.

To better address the density variation of the neighborhoods, Liu and colleagues claimed that the appropriate nearest neighbors of an instance x should be in the set of its $\langle K, \epsilon \rangle$ – nearest neighbors, or, the $\mathcal{N}eighbor_{\langle K,\epsilon\rangle-NN}(x)$, which is defined as,

$$\begin{gathered} \mathcal{N}eighbor_{\langle K,\epsilon\rangle-NN}(x) = \\ \{\, y_i | dist(x, y_1) \leq \cdots \leq dist(x, y_i) \leq \cdots \leq dist(x, y_K), \\ dist(x, y_i) \leq (1+\epsilon) dist(x, y_1), 1 \leq i \leq K \,\} \end{gathered} \tag{2.25}$$

$\mathcal{N}eighbor_{\langle K,\epsilon\rangle-NN}(x)$ is the intersection of $\mathcal{N}eighbor_{K-NN}(x)$ and $\mathcal{N}eighbor_{\epsilon-NN}(x)$. For $\mathcal{N}eighbor_{K-NN}(x)$, the number of closest neighbors is fixed to K , which will include outliers in a sparse neighborhood (Fig. 2.24a), reducing retrieval reliability. But for $\mathcal{N}eighbor_{\epsilon-NN}(x)$, closest neighbors are those with distance no larger than $(1 + \epsilon)$ times of the minimal distance, which may result in large number of neighbors in a dense neighborhood (Fig. 2.24b), increasing computing time. With $\mathcal{N}eighbor_{\langle K,\epsilon\rangle-NN}(x)$, outlier rejection and number regulation can be well balanced by choosing appropriate ϵ and K, thus, a high performance neighbor retrieval can be achieved.



(a) (b)

Fig. 2.24 $\langle K, \epsilon \rangle - NN$ neighborhood

In this figure, two dots (green and red) are examined for closest neighbors. In (a), taking $K - NN$ $(k = 5)$, the green dot's closest neighbors (light green) include outliers which distances are 5 times of the minimal distance (the straight dash line); In (b), taking $\epsilon - NN(\epsilon = 1)$, the red dot have nine closest neighbors (within the dashed circle), which is too many if the desired number of closest neighbors is consider 5. With $\langle K, \epsilon \rangle - NN(K = 5, \epsilon = 1)$, these conflicts can be resolved, for the greet dot, only the 2 neighbors within the dashed circle are chosen, while only the closest 5 dots (orange) are considered good neighbors of the red dot [1].

2.5.3 SIFT-flow dense correspondence

After nearest neighbors are retrieved by means of matching GIST distances, dense correspondences between a query image and its neighbors can be established using SIFT-flow algorithm (please refer to Section 2.4 for detail of SIFT-flow dense scene alignment).

2.5.4 Label transfer

Once dense correspondences are found, the next step is to transfer the existing annotations to parse a query image. Optimizing SIFT flow energy function 2.23 for the query image (im_q) and its $\langle K, \epsilon \rangle$ nearest neighbor (im_i), we can obtain the SIFT flow

field $\vec{w}_i$ from im_q to im_i and the minimal energy of the function. Then we can sort the $\langle K; \epsilon \rangle$ nearest neighbors by the output energy. The neighbor having the lowest energy is rated the best matching. From the SIFT flow sorted neighbors, we choose the top M $(M \leq K)$ as the candidate set for label transferring.

Shotton and colleagues [61] presented a discriminative conditional random field (CRF) model which exploits texture-layout filters, combining lower-level image features (color, location, edge), to achieve near pixel-specific segmentation (or recognition by pixel labeling) of the image.

Analogous to Shotton and colleague's idea, Liu and colleagues proposed a nonparametric MRF model which only takes voted candidates(M) within nearest neighbors(K) into account with carefully chosen weights (α, β) to achieve label transferring. Replacing texture-layout term with SIFT-flow data term, removing color term, they defined the posterior probability as,

$$\begin{aligned} -logP(c|I,s,\{s_i,c_i,w_i\}) &= \sum_p \overbrace{\psi(c(p);s,\{s_i'\})}^{Likelihood\ (SIFT\ flow\ data)} + \alpha \sum_p \overbrace{\lambda(c(p))}^{location\ prior} \\ &+ \beta \sum_{\{p,q\}\in\varepsilon} \overbrace{\phi(c(p),c(q);I)}^{smoothness\ (edge\ potential)} + \overbrace{logZ}^{normalization} \end{aligned} \tag{2.26}$$

where I is the query image, s is its dense SIFT image, $\{s_i, c_i, w_i\}_{i=1:M}$ is {SIFT image, annotation label image, SIFT flow field from s to s_i} of the i^{th}voted candidate, and Z is

the normalization constant of the probability. This posterior consists of three components, namely, likelihood, prior, and spatial smoothness.

The **likelihood** term is defined as,

$$\psi(c(p)=l)=\begin{cases}\min\limits_{i\in\Omega_{p,l}}\|s(p)-s_i(p+w(p))\|, & \Omega_{p,l}\neq\emptyset\\ \tau, & \Omega_{p,l}=\emptyset\end{cases} \tag{2.27}$$

where $\Omega_{p,l}=\{i;c_i(p+w(p))=l\}, l=1,\dots,L$, is the index set of candidates whose pixel at $p+w(p)$ is with label l. $\tau=\max_{s_1,s_2,p}\|s_1(p)-s_2(p)\|$ is the maximal difference of SIFT feature at pixel p.

The **prior** term $\lambda(c(p)=l)$ represents the prior probability of object class l appearing at pixel p. It can be obtained by counting the occurrence of each object class at each location in the training images:

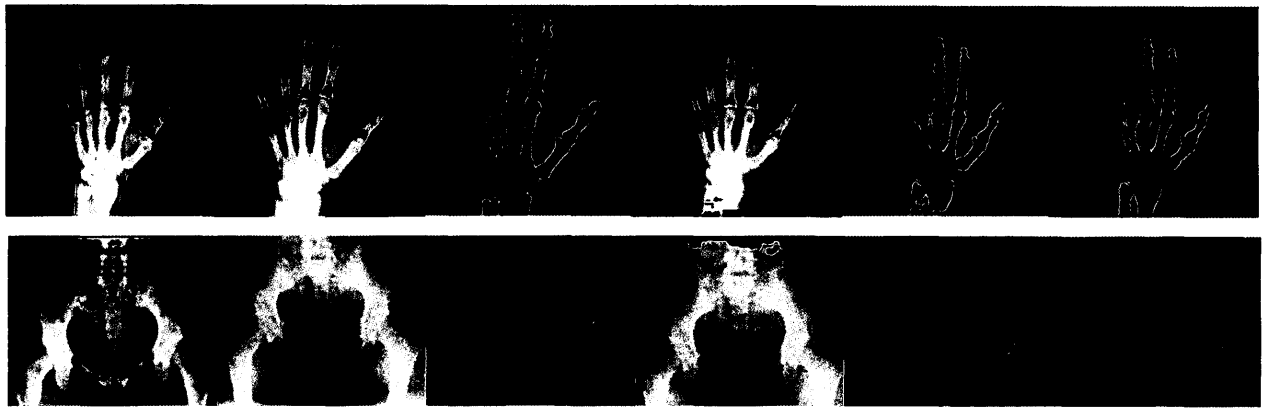
$$\lambda(c(p)=l)=-\log hist_l(p) \tag{2.28}$$

The **smoothness** term has the form of Potts model for pairwise edge potential. When no other information is available, smoothness term can bias the neighboring pixels into taking the same label. Its probability relies on the edge of the image, i.e. the stronger edge contrast, the more possibility of that the neighboring pixels have different labels:

$$\phi(c(p),c(q))=\delta[c(p)\neq c(q)]\left(\frac{\xi+e^{-\gamma\|I(p)-I(q)\|^2}}{\xi+1}\right) \tag{2.29}$$

where $\gamma=(2\langle\|I(p)-I(q)\|^2\rangle)^{-1}$, and $\langle\cdot\rangle$ is the average operation over the whole image.

Compared to Shotton and colleague's approach, Label Transfer System only involves choosing values of the parameters K, M, α, β without training large sets of annotated images for $\boldsymbol{\theta}$. Once the parameters are determined, the BP-S optimizer is used to minimize the energy in solving the above MRF posterior probability. The resulting integer label image $c(I)$ is the desired segmentation (recognition). Fig. 2.25 shows the results of hand and hip x-ray image segmentation using Label Transfer System with parameters $K = M = 1, \alpha = 0.06, \beta = 20$.



a b c d e f

Fig. 2.25 Segmentation results using Label Transfer System for hand and hip x-ray mages
(a) query image [4] (b) candidate image [2] (c) candidate annotation (d) warped candidate pixels on query image (e) resulted segmentation (f) ground truth annotation of query image

2.6 Bilateral filter

The bilateral filter proposed by Tomasi and Manduchi is a nonlinear smoothing filter that can reduce noise and preserve edges at the same time [33]. Bilateral filter not only considers distance as a metric to measure weight of the neighbor points but also considers the intensity similarity as an important metric. It is a combination of a domain filter and range filter. It can be formulated as following,

$$h(\vec{x}) = k^{-1}(\vec{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\vec{\xi}) c(\vec{\xi}, \vec{x}) s\left(f(\vec{\xi}), f(\vec{x})\right) d\vec{\xi} \tag{2.30}$$

where the normalization $k(\vec{x})$ is to maintain the total weights to be one,

$$k(\vec{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\vec{\xi}, \vec{x}) s\left(f(\vec{\xi}), f(\vec{x})\right) d\vec{\xi} \tag{2.31}$$

$c(\vec{\xi}, \vec{x})$ is the measure for geometric closeness and $s\left(f(\vec{\xi}), f(\vec{x})\right)$ is the measure for intensity similarity. In this work, we applied Gaussian function to both c and s. So, in a smooth region where pixel intensities in a small window are similar, $k^{-1}s$ is close to 1, the filter acts as a standard Gaussian filter. However, when the inspected point is on the bright side of a boundary, the similarity function gives high weight to the neighboring pixels having similar intensities and almost zero weight to those pixels having very different intensities. Vice versa, the same can be applied to the points on the dark side. In such a way, the boundary is maintained. Below is the formula for c and s using Gaussian function,

$$c(\vec{\xi},\vec{x}) = \exp\left(-\frac{1}{2}\frac{\|\vec{\xi}-\vec{x}\|^2}{\sigma_d^2}\right) \quad (2.32)$$

$$s\left(f(\vec{\xi}),f(\vec{x})\right) = \exp\left(-\frac{1}{2}\frac{\|f(\vec{\xi})-f(\vec{x})\|^2}{\sigma_r^2}\right) \quad (2.33)$$

where σ_d and σ_r are deviations for the domain filter and range filter, respectively. A larger σ_d blurs more. Pixels with intensity difference smaller than σ_r will be mixed together; otherwise they will not be mixed. Both domain filter and range filter are shift-invariant. The range filter is also insensitive to overall intensity changes [33].

In terms of cost, bilateral filter is twice as expensive as a non-separable domain filter of the same size. A simple trick for decreasing this cost is to compute all values for $s(f(x))$ first as we do in computing a mask for linear filters. In the case of Gaussian filters, if the image has n intensity levels, there will be $2n + 1$ possible values for $s(f(x))$. Experiments on color images showed that bilateral filter can handle multi-channels much better than other edge-preserving filters such as median filter. Tomasi and Manduchi also showed that if a cartoon-like appearance is desired, we can apply multiple iterations of bilateral filter on the target image such that the edges or object boundaries can stand out more [33]. Fig. 2.20 shows the results of bilateral filter on hand and hip x-ray images. Notice that the bone boundaries are emphasized yet the soft tissues and interior area of the bones are blurred.

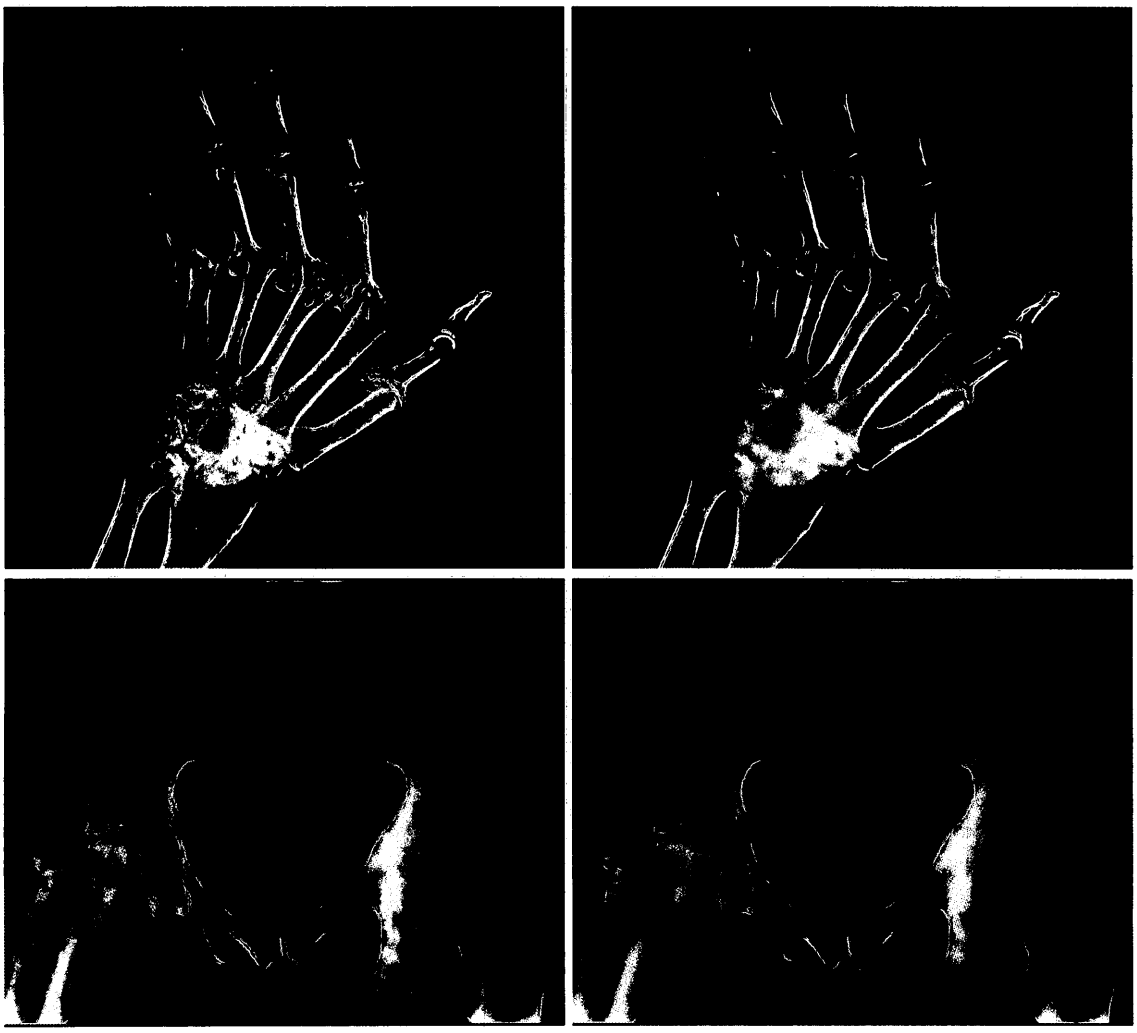


Fig. 2.20 Results of bilateral filter on x-ray images

The images on the left [5] [14] are from the original source. The images on the right are results treated by bilateral filtering. The images on the 1st row were taken from a rheumatoid arthritis hand. The images on the 2nd row were taken from a distorted hip.

Chapter Three: Assessing Segmentation Performance of Label Transfer System on Medical Images

In this chapter, we describe the approach of evaluating the segmentation performance of the Label Transfer System on medical images, particularly, orthopedic x-ray images. We also attempt to alter several factors, such as feature type, preprocessing filter, MRF optimizer and neighborhood system, to find improvements of the Label Transfer System. In the first section of this chapter, we briefly review Liu's Label Transfer System. In the second section, we discuss the choosing of preprocessing filter, neighborhood system, different features and varied optimizers. In the last section, we describe the implementations of our approach which includes assessing procedures, experiment settings, data collection and organization.

3.1 Review of Label Transfer System

Liu and colleagues' Label Transfer System is a nonparametric scene parsing system which does not involve model training to determine parameters. To parse a query image, the system matches the objects in the query image to the pre-annotated images in a database. If the matchings in the pre-annotated images are annotated with object labels and semantically meaningful, then the labels of the pre-annotated images will be simply transferred to the query image. The Label Transfer System consists of three key

algorithmic modules, *scene retrieval*, *dense scene alignment*, and *label transfer* as shown in Fig. 3.1:

1) First, for a given query image, the **scene retrieval** module uses scene retrieval techniques (based on GIST, for example) to find its nearest neighbors in the pre-annotated image database. For example, it first computes the query image GIST feature. Then it calculates the GIST distances between the query image and the images in the database. Finally, it takes the images with the smallest GIST distances as the nearest neighbors of the query image. The number of the nearest neighbors is $\langle K, \epsilon \rangle$ as stated previously.

2) Second, a **dense scene alignment** module establishes dense scene correspondences between the query image and each of the $\langle K, \epsilon \rangle$ nearest neighbors. The M top matching nearest neighbors are chosen as voting candidates, where $M \leq K$. Firstly, this module finds the dense SIFT features of the query image and a nearest neighbor; then it estimates the SIFT-flow between the pair by minimizing the flow energy. Repeated on all $\langle K, \epsilon \rangle$ nearest neighbors, a set of SIFT-flow images representing the dense correspondence with their optimal energies are found. Finally, M neighbors with lowest optimal energies are determined as voting candidates.

3) The **label transfer** module warps the annotations from the voting candidates to the query image by using a Markov random field model to integrate the dense correspondences, multiple object priors, and spatial smoothness constraints.

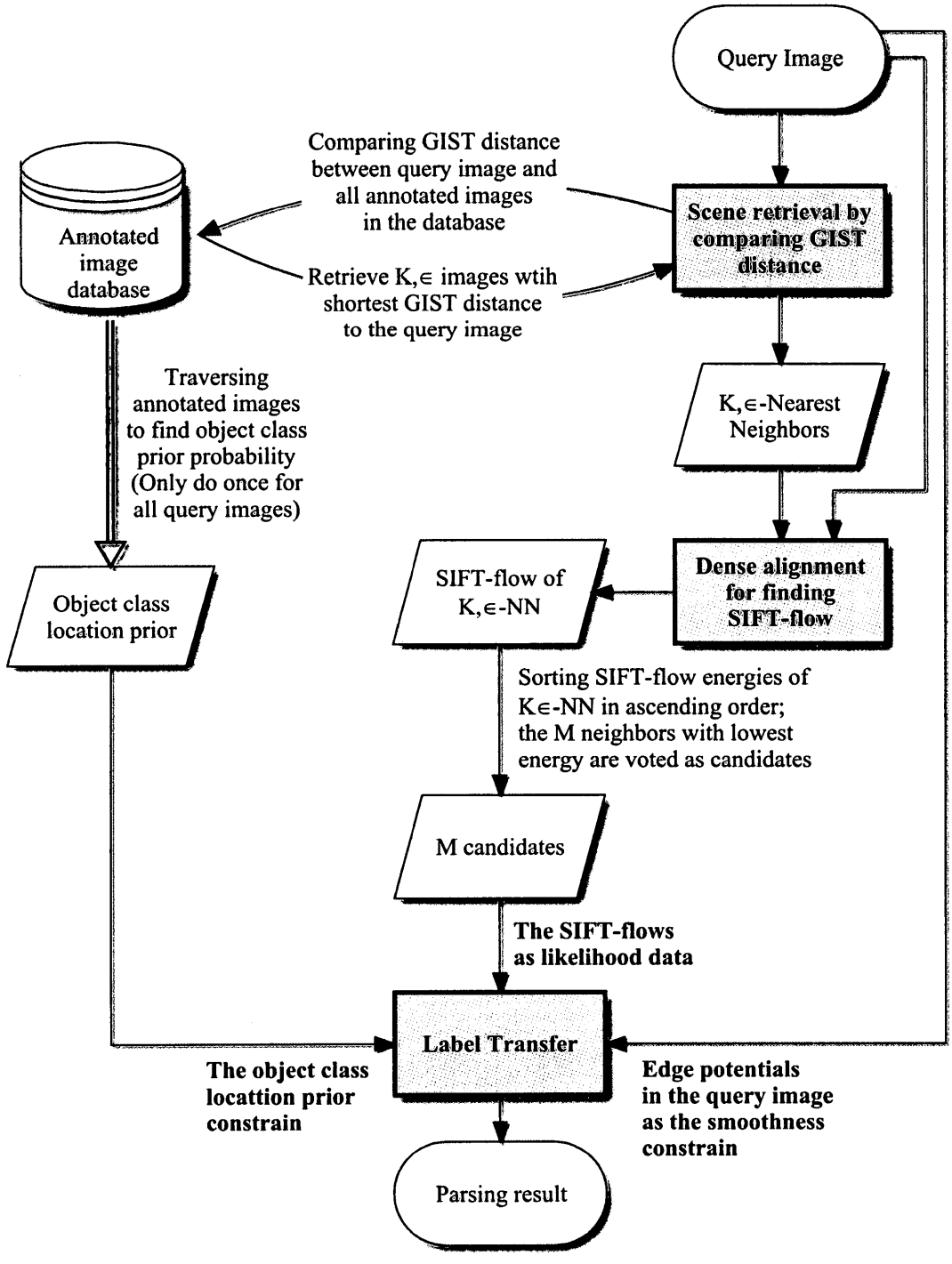


Fig. 3.1 Workflow of Label Transfer System

In this figure, rectangles represent algorithmic processes, parallelograms represent input/output. The three key algorithmic modules, *scene retrieval*, *dense alignment* and *label transfer*, are colored in orange.

3.2 Our assessing approach

As Liu and colleagues pointed out, even though concrete algorithms are chosen for each module in their paper, any algorithm appropriate for a module can be plugged into their parsing system [1]. We hope that altering algorithms or processing mechanisms can yield consistent, substantial performance gains for segmenting medical images, in particular, those with unusual deformations. In our assessment, we concentrate on four major factors: feature descriptors for computing dense correspondence, MRF optimizers in the label transfer module, the neighborhood system, and utilization of an image preprocessing filter. In this section, we elaborate on the rationale of choosing these factors as our focus points.

3.2.1 Choosing feature descriptors

In Liu and colleague's Label Transfer System, dense SIFT descriptors are used to find dense correspondences between points in two different images. They showed that SIFT flow is capable of establishing semantically meaningful correspondence by matching local SIFT descriptors [31] because SIFT features allow robust matching across different object appearance (with varied scales, orientations, viewpoints and illuminations).

However, the dense SIFT descriptors in the Label Transfer System do not address the possibility that multiple keypoints with different orientations may reside at the same location. Lowe pointed out the existence of multiple keypoints at a single location can

significantly improve matching performance even though approximately 15% of locations have multiple orientation keypoints [36]. To address this problem, we propose to use SURF, another scale-rotation-invariant robust feature, to replace SIFT in the system. Compared to SIFT, SURF has superior noise resistance. And more importantly, as Bay and colleagues indicated, even without assigning a specific orientation to the interest point, SURF still maintains robustness to rotation of about $\pm 15°$ while giving faster computation and higher distinctiveness [30]. Considering the fact that rotational variations among most medical images are not larger than 15°, we chose to investigate whether SURF would outperform SIFT when used the Label Transfer System for medical image segmentation. Moreover, the SIFT descriptor is a 128-dimension vector whereas the SURF descriptor has only 64 dimensions. This implies that the Label Transfer System utilizing SURF will run faster than that with SIFT in estimating dense correspondence using feature-flow.

Unlike Lowe's SIFT descriptor that is generated by extrema detection and accurate localization, the SIFT descriptors used in the Label Transfer System are per-pixel SIFT descriptors, which do not involve detection and localization. This dense feature approach sacrifices matching precision, because unlike sparse keypoint matching which locate keypoints at sub-pixel-level, dense feature correspondence considers all locations at pixel-level. And this issue will also occur despite the choice of different feature descriptors. To address this problem, we combine both SIFT and SURF features to

attempt to feed more local structural information into the system. There are two ways to do so:

1) By combining the 128-D SIFT descriptor vector and 64-D SURF descriptor vector into a 196-D descriptor at each pixel. We called this descriptor the STSF descriptor. The dense STSF descriptors then are normalized and used to find the STSF dense correspondence (STSF-flow).
2) By computing both SIFT-flow and SURF-flow in the dense correspondence module. Then, in the label transfer module, we find the likelihood data with SIFT-flow and SURF-flow separately. Further, we combine the likelihoods from SIFT-flow and SURF-flow with respect to a specific object class. We coined this approach as SSLH (SIFT and SURF likelihood).

3.2.2 Comparing MRF optimizers in label transfer

We believe Liu and colleagues implemented their version of sequential loopy belief propagation (BP-S-Liu) as the Markov random field optimizer in label transfer module because BP-S usually converges substantially faster than other MRF optimizing algorithms such as iterated conditional modes (ICM) , simulated annealing , max-product loopy belief propagation (BP-M), graph cuts, and tree-reweighted message passing (TRW), despite the fact that BP-S often does not obtain the lowest MRF energy [43]. For their natural scene parsing application which involves a large amount of pre-labeled images, running time is crucial. However, for applications related to medical images, the tradeoff is obvious: we usually prefer better accuracy over faster computation. We are

interested in replacing BP-S-Liu with Szeliski and colleagues' suite of MRF optimizers and finding substantial accuracy improvement for medical image segmentation.

As Szeliski and colleagues stated, among MRF optimizers, ICM and simulated annealing have been proven to be either ineffective or extremely inefficient [43]. Based on that, we used BP-S, BP-M, two varieties of graph cut (swap or expansion move), and TRW as the replacements of BP-S-Liu in our assessment.

3.2.3 Using the $\langle K, \epsilon \rangle$-NN neighborhood system or a single template

Natural scene parsing is a challenging problem. The compositions of object classes in query images are not always the same; for instance, in one query image, there is sky and sea, but in another image, there exist buildings, a street and vehicles. To parse dramatically different scenes, the Label Transfer System relies on a database with a large amount of varied pre-annotated images, which may cover as many object classes as possible. Then similar images, called nearest neighbors, can be retrieved for a query image. But for medical images, query scenes are usually pre-assumed, i.e., given a query image, we almost always know what anatomical structures will be in the image. Thus, we propose a single template approach which only uses one pre-labeled structure-specific image to parse query images with the same structure. For example, only one pre-labeled hand x-ray image will be used as template to parse other hand x-ray images in the modified Label Transfer System. In this way, we minimize the feature-flow calculation to a single image only, thus reducing the computation time significantly because the dense

correspondence module (flow estimation) is the major time consuming component. In comparison, we call this approach T1 (one template), and denote the $\langle K, \epsilon \rangle$-NN neighborhood system as Tk. Fig. 3.2 demonstrates how the T1 approach eliminates the nearest neighbor search and reduces feature flow calculations.

In our assessment, we test both Tk and T1 approaches. The hypothesis is that T1 approach is sufficient to achieve similar accuracy as the Tk approach.

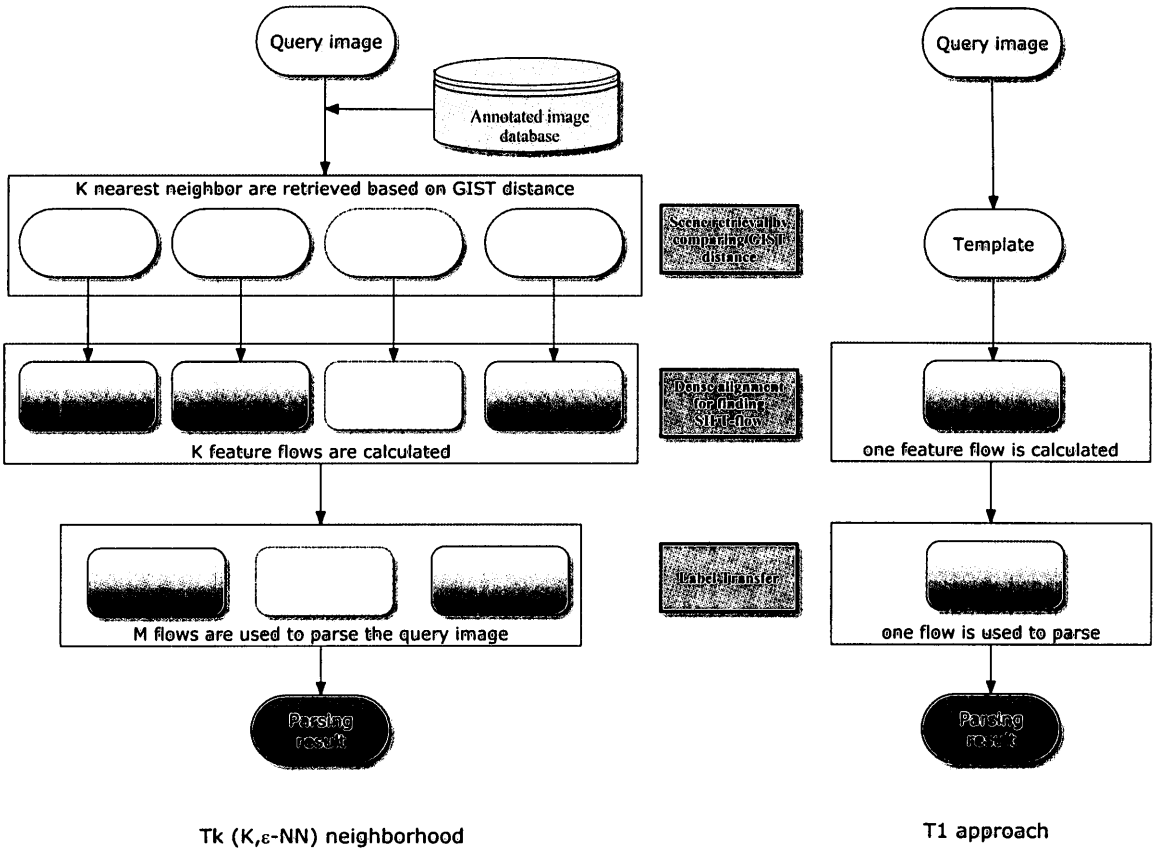


Fig. 3.2 Simplified Label Transfer System for medical image segmentation

The original $\langle \boldsymbol{K}, \boldsymbol{\epsilon} \rangle$-NN system involve K times feature flow computations as of our T1 approach.

3.2.4 Using a multi-image prior or a single-image prior for T1

For Liu and colleagues' natural scene parsing application, the prior probability that an object class appears at a specific location of a query image is found by traversing all pre-labeled images in the database to count the occurrence of this object class at the specific location. Ideally, with a sufficiently large number of images in the database, the object priors reflect the close-to-truth situation. We call this the *multi-image object prior* (MP).

In our assessment, we used their method to find the multi-image object prior for the $\langle \boldsymbol{K}, \epsilon \rangle$-NN system (Tk). For the T1 approach, in addition to utilizing multi-image object prior, we also used a *single-image prior* method (coined as SP) which only involves the single template. Basically, the single-object prior is the normalized pixel intensities of the template image convolved with a broad Gaussian filter. Thus, toggling Tk/T1 and MP/SP, we have three conditions to evaluate: Tk-MP, T1-MP and T1-SP.

3.2.5 Using a preprocessing filter (bilateral filter)

In orthopedic image segmentation, different object classes are usually distinguished by boundaries. Edge features are very important in isolating bony structures. However, SIFT or SURF feature extractions utilize repeated Gaussian processes which could substantially suppress edge features. This may result in unreliable matching in the dense correspondence module for our applications.

To address the possible suppression of edges, we propose to conduct a preprocessing step before executing Label Transfer System. An appropriate preprocessing filter is bilateral filtering. The bilateral filter is an edge-preserving filter that can both accentuate edge features and reduce image noise. We expect that, after bilateral treatments on both the query and training images, Gaussian smoothing in feature-flow may be reduced.

3.3 Implementations

To test the many combinations of feature descriptors, MRF optimizers, neighborhood systems, and preprocessing filters, we modified the Label Transfer System into a cross-factor testing platform. We evaluated its performance in terms of recognition accuracy and running time under conditions of 4 feature descriptors, 6 MRF optimizers, 3 neighborhood/prior approaches, and 2 preprocessing treatments: thus, there are a total of $4 \times 6 \times 3 \times 2 = 144$ conditions for each query image. In this section, we briefly explore how this testing platform is constructed and what data are generated by it. A complete flowchart can be found in Fig. 3.3.

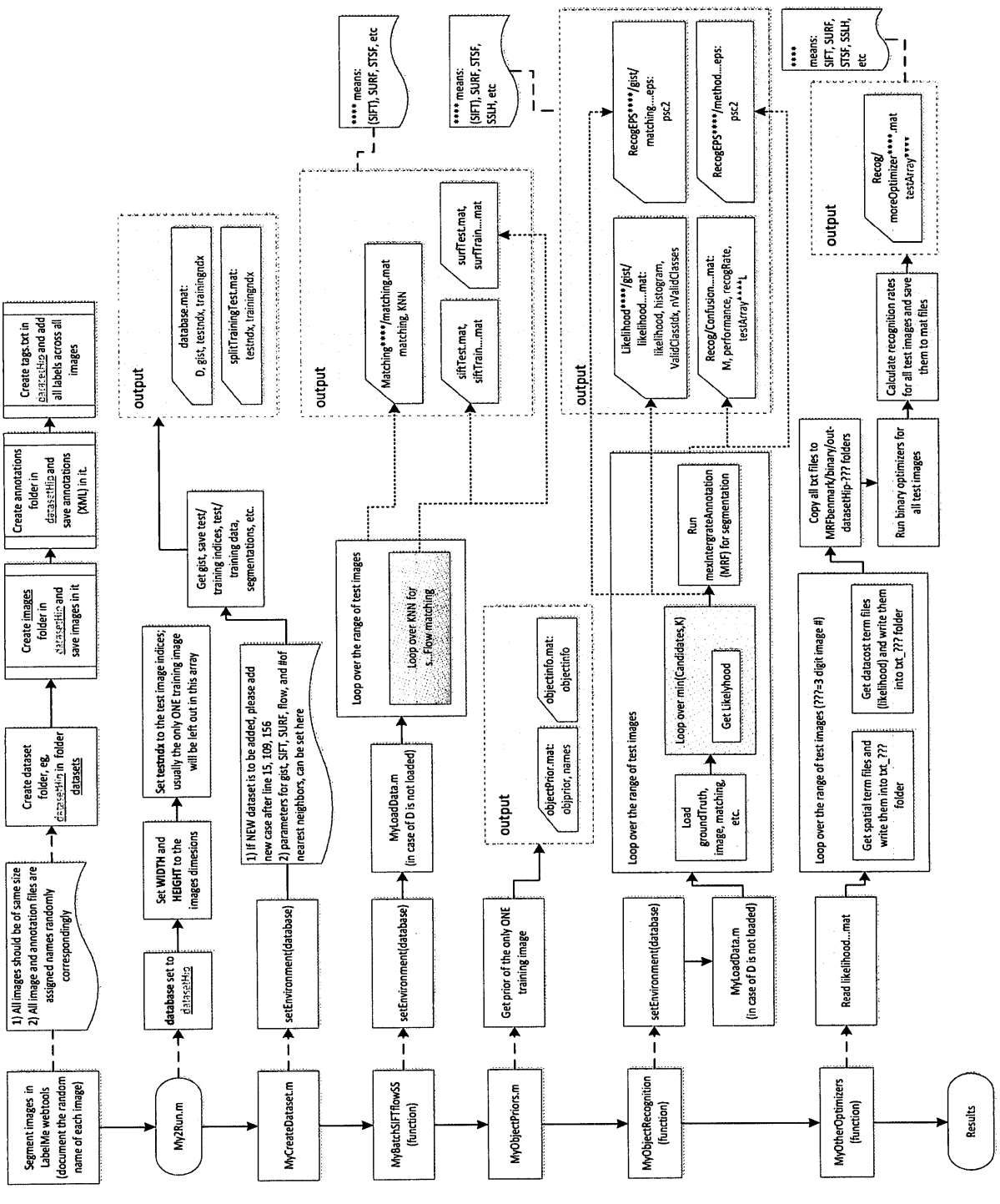


Fig. 3.3 The complete flowchart of the Label Transfer System

3.3.1 Assessing procedures

First, we divided the assessment experiments into 3 big blocks determined by the different combinations of nearest neighborhood system and object prior calculation methods. As stated in previous section, they are T1-SP, Tk-MP, and T1-MP. Then each block is divided into two sub-blocks representing the use of bilateral filter or not. In each sub-block, we generate four different feature-flows (SIFT, SURF, STSF and SSLH) between the query image and training image(s). In the end, the four feature-flows are fed to the label transfer module individually. For each type of feature-flow, the label transfer module utilizes six different MRF optimizers (BP-S-Liu, BP-M, BP-S, Expansion, Swap, and TRW) to parse the query image to obtain segmentation results. A condensed flowchart of these procedures is shown in Fig. 3.4.

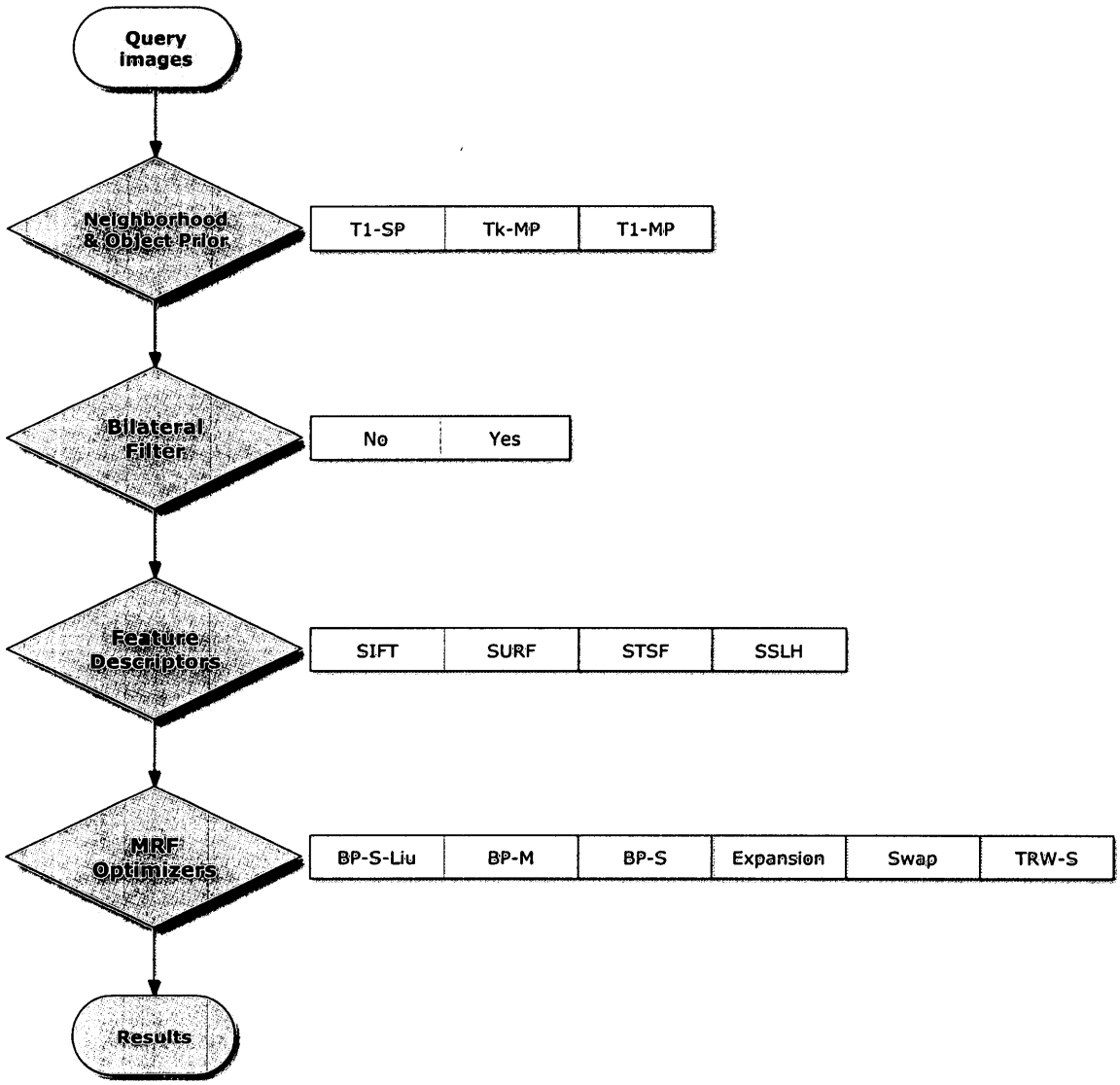


Fig. 3.4 Workflow of our assessment platform

Our platform tests query images under 3 neighborhood/prior approaches and 2 preprocessing filter treatments, using 4 feature descriptors and 6 MRF optimizers. The total number of conditions is $3 \times 2 \times 4 \times 6 = 144$.

3.3.2 Software and hardware environment

Our assessment platform is built on the Label Transfer System code from Liu and colleagues [62]. The platform is implemented in Matlab 2011b on Ubuntu 64bit 12.04 LTS operating system. For higher efficiency, some computationally intense algorithms, such as dense SIFT extraction, feature flow estimation, and label transfer MRF optimization are coded in C++ mex-files. Parallel computation on all CPU cores is also used in the SURF descriptor calculation and feature flow estimation using the Matlab Parallel Computation Toolbox.

The same system was used to run all of the experiments. The hardware configuration of this system was a PC architecture workstation with one 2.93GHz Intel i7 Core Quad CPU, and 12GB memory.

3.3.3 Data collection

To evaluate segmentation performance of the Label Transfer System on medical images, we focus on the measurements of recognition accuracy (correctness rate and error rates) and algorithm running time. For each query image, the 144 sets of recognition correctness rates, false positive ($false^{POS}$) rates, false negative ($false^{NEG}$) rates, and run times were calculated and recorded. All flow estimation times were also recorded to compare the efficiencies of feature flow algorithm using different feature combinations.

For a query image under a specific condition, one confusion matrix is recorded for accuracy analysis. In the confusion matrix, each column represents the pixels in a predicted class, while each row represents the pixels in an actual class. From this matrix, we can obtain the table of confusion of each object class, which contains its true positive pixel count, false negative pixel count, false positive pixel count and true negative count. Furthermore, we can calculate the correctness rate and error rates as,

$$correctness\ rate\ (accuracy) = \frac{true\ positives}{true\ positives + false\ negatives} \tag{3.1}$$

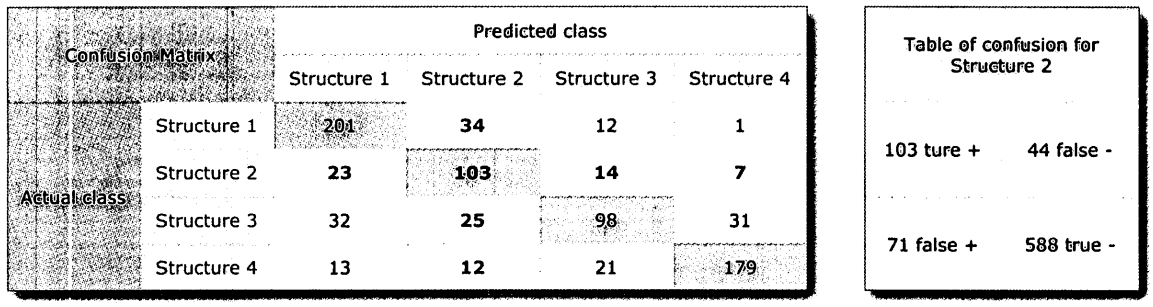
$$false^{POS}\ error\ rate\ (type\ I) = \frac{false\ positives}{true\ positives + false\ negatives} \tag{3.2}$$

$$false^{NEG}\ error\ rate(type\ II) = \frac{false\ negatives}{true\ positives + false\ negatives} \tag{3.3}$$

Also, from the confusion matrix, we can obtain an overall recognition rate for this query image as,

$$overall\ correctness\ rate = \frac{\sum_i true\ positives\ of\ class\ i}{\sum all\ values} \tag{3.4}$$

An example of the confusion matrix and its utilization for calculating recognition accuracy is demonstrated in Fig. 3.5. Notice that, for an object class C_i, (correctness rate + false^{NEG} rate) = 1, but the false^{POS} rate can be any non-negative value (possibly larger than 1 if the recognition algorithm fails to distinguish pixels belonging to other classes from pixels of class C_i).



Confusion Matrix		Predicted class			
		Structure 1	Structure 2	Structure 3	Structure 4
Actual class	Structure 1	201	34	12	1
	Structure 2	23	103	14	7
	Structure 3	32	25	98	31
	Structure 4	13	12	21	179

Table of confusion for Structure 2	
103 ture +	44 false -
71 false +	588 true -

Fig. 3.5 Confusion matrix and recognition rates

In confusion matrix (left), columns represent pixel counts in predicted classes while rows give pixel counts in actual classes. For instance, value at row 2 column 1 represent the amount of pixels that actually below to structure 2 but mistakenly predicted as part of structure 1. In table of confusion for structure 2 (right), four counts are present and rates can be easily obtained as, correctness rate = 103/(103+44) = 70.1%, false^{NEG} rate = 44/(103+44) = 29.9%, and false^{POS} rate = 71/(103+44) = 48.3%. The overall recognition rate is, (201+103+98+179)/sum_of_all = 72.1%.

Chapter Four: Evaluation and Discussion

In this chapter, we present the results of assessment experiments and analysis of these results. In the first section, we describe the hand and hip x-ray images used to test the Label Transfer System. In particular, we point out the specific challenges among automatic segmentations of these images. Then in the next two sections, we describe and discuss the experimental results, respectively.

4.1 Description of test images

The two sets of x-ray images were all collected from the internet and [2]. There were eight images in the hand image set and fourteen images in hip image set. The images were all in grayscale. The size of each hand image was 691×691 pixels. For each hip image, the width was 500 pixels and the height was 400 pixels. The twenty four images were manually segmented and annotated by us using LabelMe we-based tool mentioned in Section 2.5.1.

In each set, we picked one clear image with complete and normal anatomy as the template image and the rest were tested in our assessment platform. The template images and their ground truth segmentations are shown in Fig. 4.1 (hand) and Fig. 4.3 (hip). The

test images are shown in Fig. 4.2.1, Fig. 4.2.2 for hand, and Fig. 4.4.1, Fig. 4.4.2, Fig. 4.4.3 for hip.

The hand images were taken from left hands with thumbs positioned on the right side. There were 24 labels in the hand images including 5 distal phalanges, 4 middle phalanges, 5 proximal phalanges, 5 metacarpals, carpal bones taken as a single whole unit, ulna, radius, the overall hand shape and picture background. The hip images contained 13 labels, including left/right pelvis, left/right femurs, left/right sockets, left/right femurs in socket, left/right pelvic holes, tail bones, the overall body shape, and picture background. Labels in both hand and hip images can be found in Fig. 4.1 and Fig. 4.3.

4.1.1 Hand images

The hand images were taken from left hands with thumbs positioned on the right side (Fig. 4.1). There were 24 labels in the hand images including 5 distal phalanges, 4 middle phalanges, 5 proximal phalanges, 5 metacarpals, the carpal bones grouped as a single label, the ulna, the radius, the soft tissues of the hand, and the image background. The bones of the wrist (the carpal bones) were grouped as a single label because of the large amount of overlap between the bones in a typical x-ray image.

We picked the 18 year old female hand as the template (Fig. 4.1) because it had complete and normal anatomy, and clear structural contrast. The remaining 7 images (Fig. 4.2.1 and Fig. 4.2.2) included a baby hand, two small-scaled hands, a hand with where the

angle between the thumb and index finger was small compared to the template image, a rheumatoid arthritis hand, a 90 degree rotated version of the 3rd hand and a hand where the tips of the thumb and middle finger were cropped at the edges of the image.

One challenge in the baby hand is that the intensity of the bone structure is more homogenous than the adult counterparts because the tissue differentiation in baby bones has not taken place yet. Another issue is there are large gaps between bones in baby hands because the bone structures are not fully grown. For example, the carpals are two tiny dots instead of the eight bones present in the fully formed wrist. In this particular image, the baby hand is significantly larger in size compared to the adult ones.

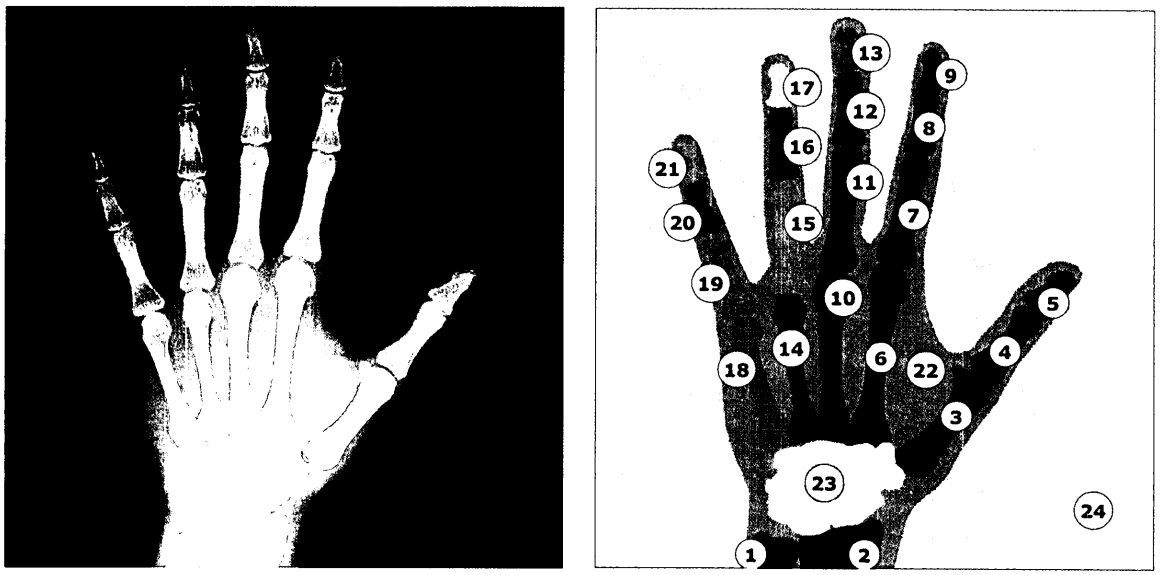
In the 2nd and 3rd images, the hands are significantly smaller in scales than in the template images. We predicted that the result would be good because the feature flows are considered scale invariant. However, we observed that there were contrast changes in the 2nd image that might affect the segmentation results.

The 4th image is taken from an 18 year old male and is very similar to the template image except that the thumb is angled closer to the index finger. We thought that this should not have a significant effect on the segmentation accuracy because the system can preserve discontinuity and endure minor rotations.

The most challenging image was the 5th image which was taken from a rheumatoid arthritis patient. Compared to the template, there were significant clockwise rotations of the bottom half of the hand and obvious counter-clockwise rotation of the upper half of the hand. Also, joints between the proximal phalanges and metacarpals were fused or distorted. Because of the rotation-invariant aspect of feature flow, we predicted that the system might achieve a reasonable (although not perfect) segmentation in this case.

The 6th image is a counter-clockwise rotated version of the 3rd image. This image was used to evaluate the rotation-invariant characteristic of the Label Transfer System in an extreme case of rotation and displacement.

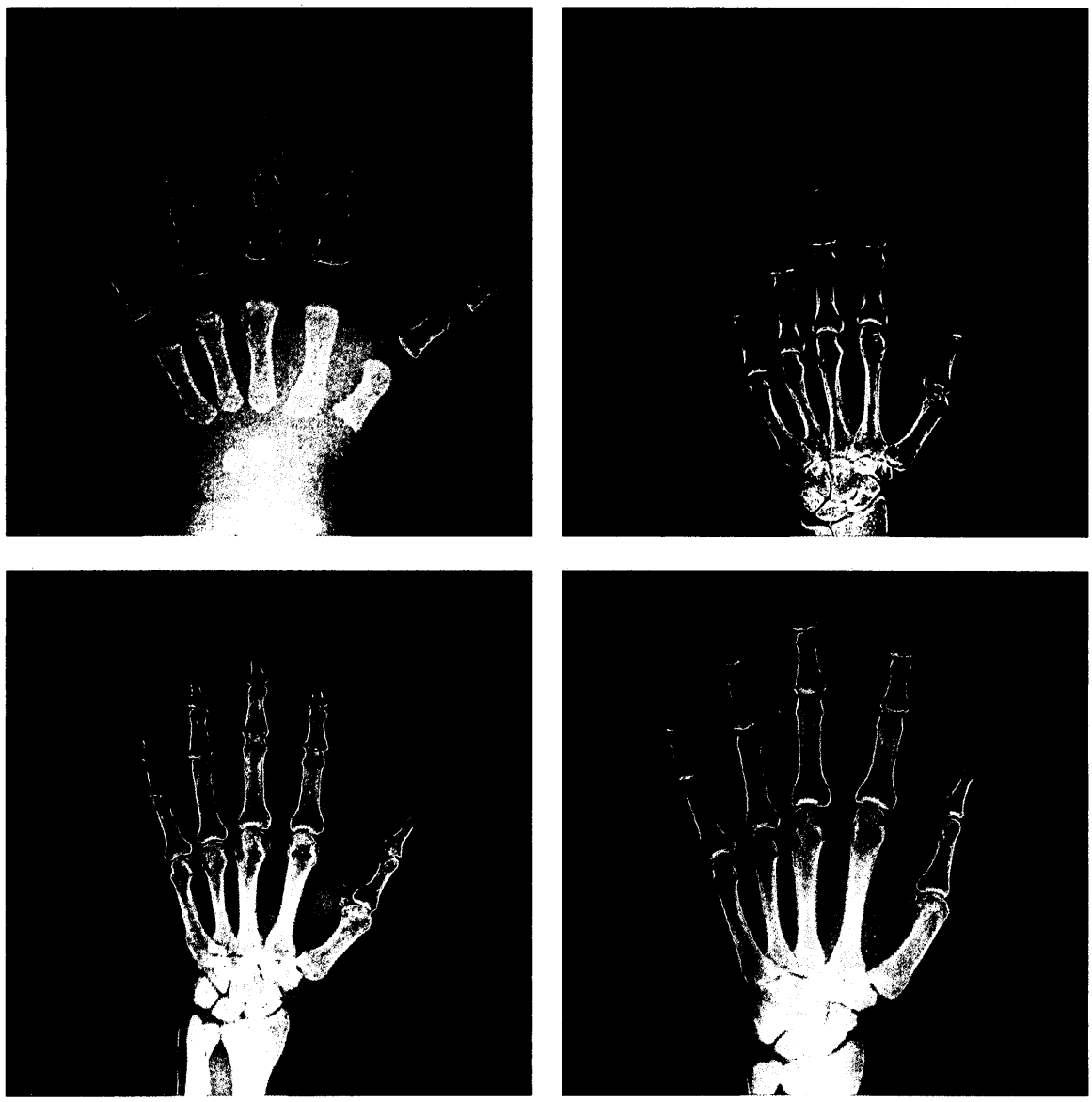
The last image is slightly larger in scale than the template which leads to cropping of the tips of the 3rd distal phalange and the 1st distal phalange. Also, signal inhomogeneity artifacts exist and cause contrast isotropy, particularly at the finger tips.



(01) Ulna
(02) Radius
(03) 1st metacarpal
(04) 1st proximal phalange
(05) 1st distal phalange
(06) 2nd metacarpal
(07) 2nd proximal phalange
(08) 2nd middle phalange
(09) 2nd distal phalange
(10) 3rd metacarpal
(11) 3rd proximal phalange
(12) 3rd middle phalange
(13) 3rd distal phalange
(14) 4th metacarpal
(15) 4th proximal phalange
(16) 4th middle phalange
(17) 4th distal phalange
(18) 5th metacarpal
(19) 5th proximal phalange
(20) 5th middle phalange
(21) 5th distal phalange
(22) Hand
(23) Carpals
(24) Picture
Unlabeled

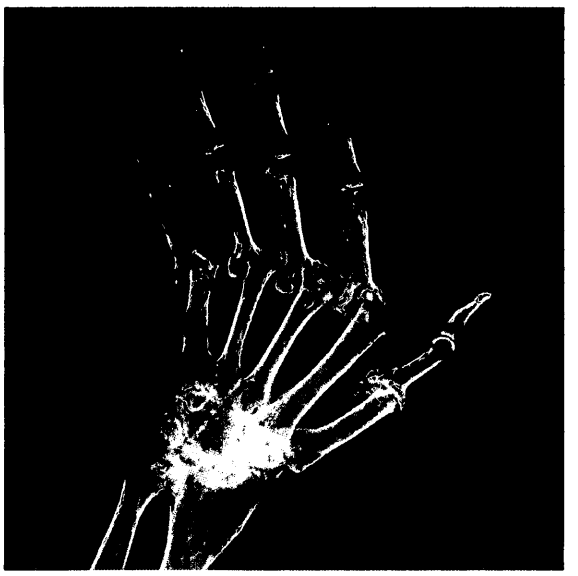
Fig. 4.1 Template image in hand image set and its segmentation ground truth

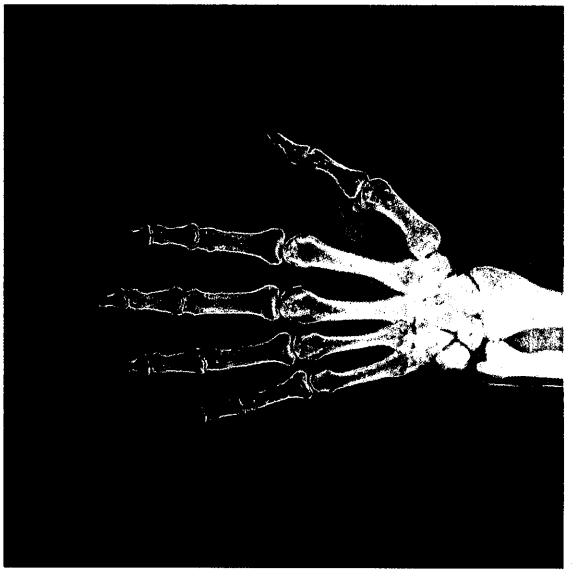
The top left image [2] is the x-ray photograph with 691 x 691 pixels in dimensions. The top right image shows the manually labeled structures. The bottom of this figure lists the names of the labeled structures with corresponding colors in the top right image.

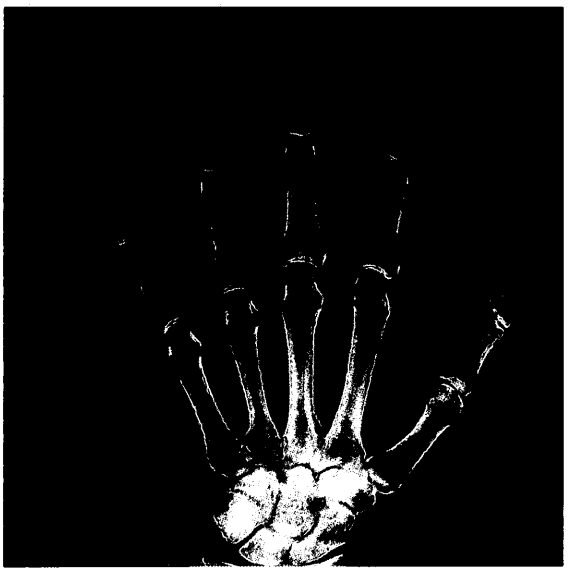


1	2
3	4

Fig. 4.2.1 Test images (1-4) in hand image set [2] [3] [4] [2]







5	6
7	-

Fig. 4.2.2 Testing images (5-7) in hand image set [5] [4] [6]

4.1.2 Hip images

The hip images contained 13 labels, including both sides of the pelvis, the femurs, the overlap of the femoral heads and pelvis, the joint space between the acetabulum and femoral head, the pelvic holes, tail bone, the soft tissues, and the image background (Fig. 4.3).

	Size and displacement	Contrast and quality	Signal inhomogeneity	Abnormality	Interfering structures
1			upper part is dimmer		intestines
2	bigger; loss of periphery parts		upper part is dimmer; joints are stronger		
3	moved up; loss of pelvis; long femurs				intestines
4	moved up; long femurs				male organ
5		low contrast			intestines
6				broken left femur	
7		low contrast	upper pelvis are inhomogeneous		male organ
8		bad quality		right joint fusion; asymmetric	
9		low contrast	lower part of image is strong		
10	moved up; loss of pelvis;		upper part is dimmer		male organ
11			upper part is dimmer		
12	bigger; loss of periphery parts		upper part is dimmer	right joint fusion	intestines
13				both joints fusion	

Table 4.1 Overview of challenges of hip image segmentations

We chose a hip x-ray image taken from a female as the template image (Fig. 4.3). It presents complete and normal hip anatomy. The remaining 13 images (Fig. 4.4.1, Fig.

4.4.2 and Fig. 4.4.3) vary in size, contrast and signal inhomogeneity. There are also structural differences, such as arthritis, broken bones, and occlusions caused by non-bony structures. Table 4.1 gives an overview of the segmentation challenges we may face in each test image using the Label Transfer System.

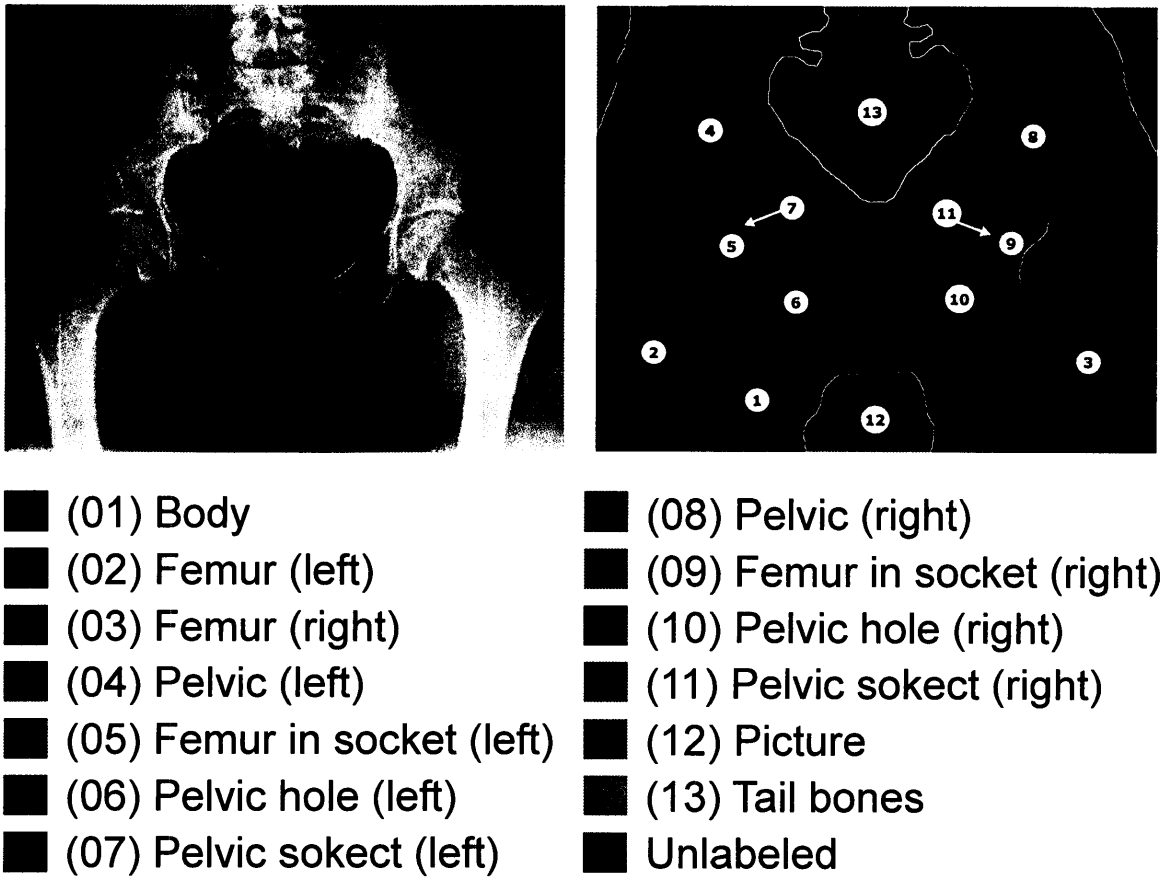
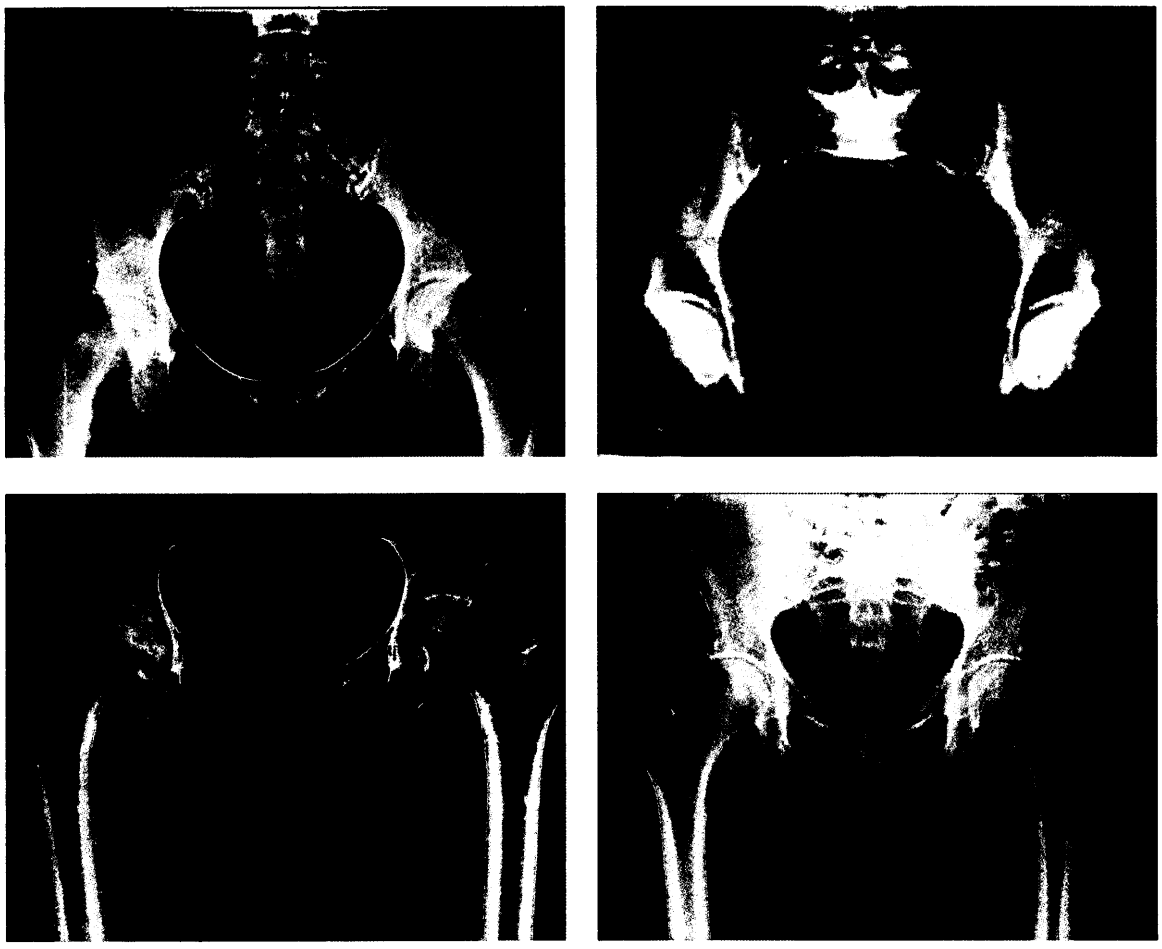


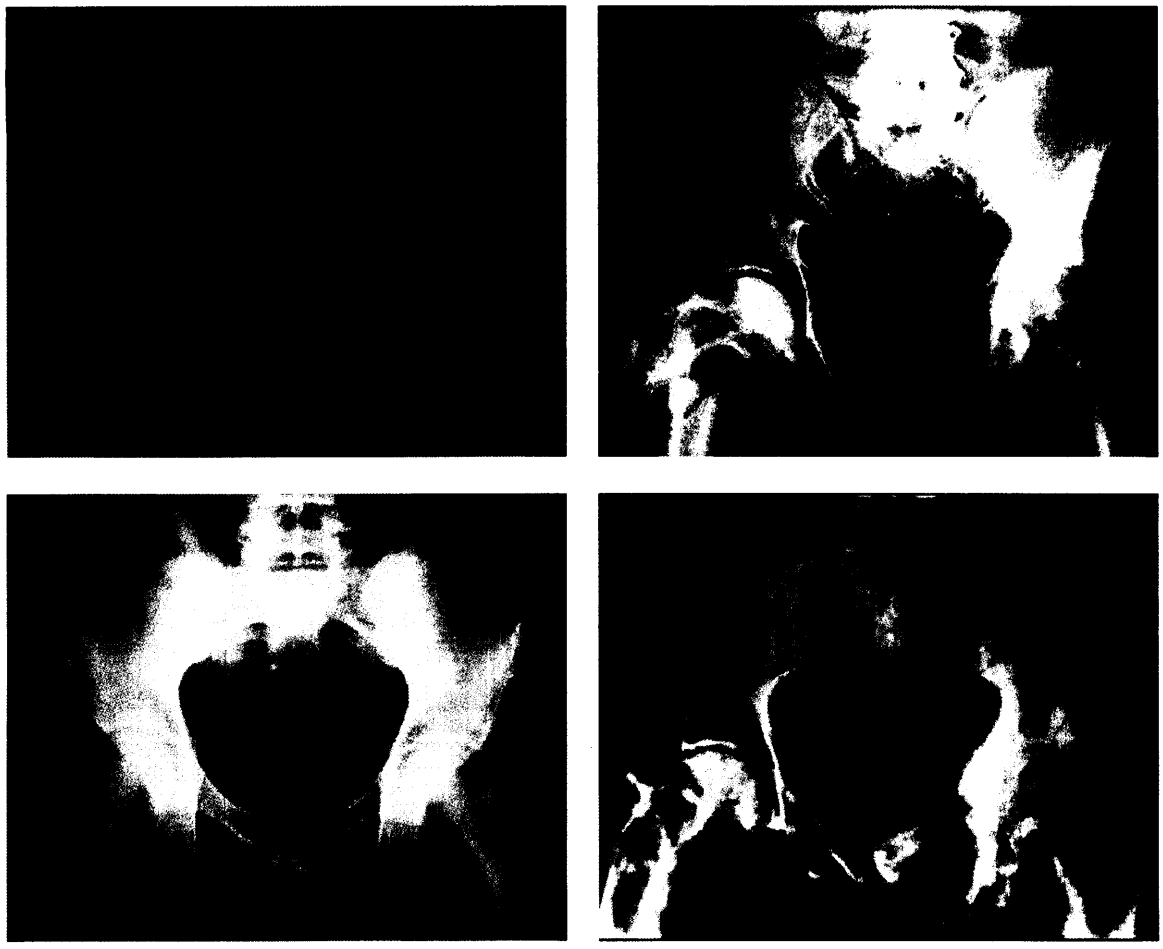
Fig. 4.3 Template image in hip image set and its segmentation ground truth

The top left image [50] is the x-ray photograph with 500 pixels in width and 400 pixels in height. The top right image shows the manually labeled structures. The bottom of this figure lists the names of the labeled structures with corresponding colors in the top right image.



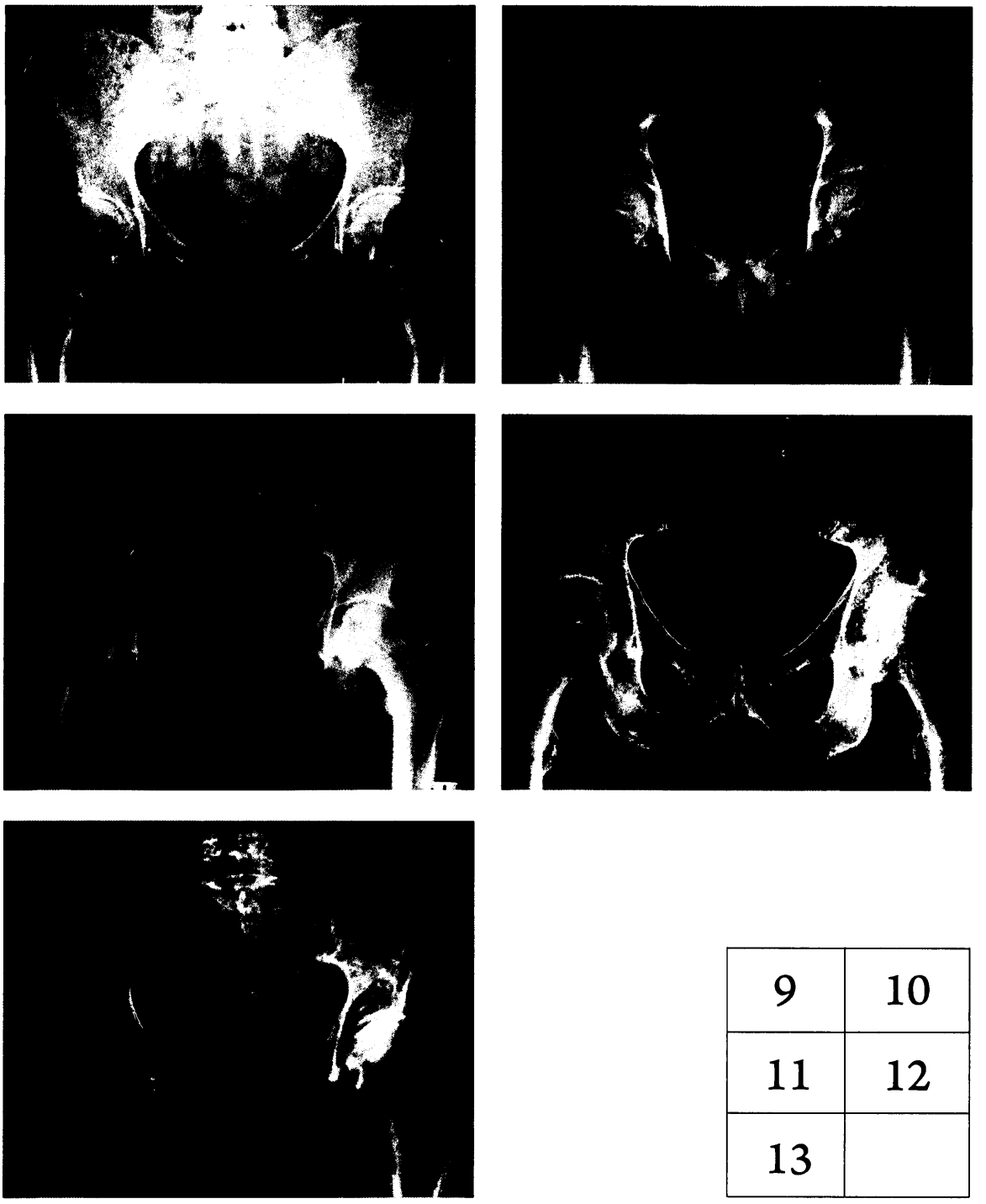
1	2
3	4

Fig. 4.4.1 Testing images (1-4) in hip image set [7] [8] [9] [10]



5	6
7	8

Fig. 4.4.2 Testing images (5-8) in hip image set [11] [12] [13] [14]



9	10
11	12
13	

Fig. 4.4.3 Testing images (9-13) in hip image set [15] [16] [17] [18] [19]

4.2 Experiment results

7 hand images and 13 hip images were tested using the assessment procedure described in Section 3.2.1. There are 4 features, 6 optimizers, 2 filter treatments and 3 neighborhood/prior configurations as major factors to consider in our assessing platform. Hence, we have 144 specific conditions for each test image. For referring convenience, we code different factors with a capitalized letter followed by one digit as shown in Table 4.2. For a specific condition, we have a unique code. For instance, F2-O2-B1-P1 means that the test condition uses (1) SURF as the feature descriptor, (2) BP-M as the label transfer optimizer, (3) no pre-filtering with a bilateral filter, and (4) one template image with the prior computed from the template image.

Features		Optimizers		Bilateral treatments		Neighborhood system & prior configurations	
F1	SIFT	O1	Liu's BP	B1	No	P1	T1SP
F2	SURF	O2	BP-M	B2	Yes	P2	TkMP
F3	STSF	O3	BP-S			P3	T1MP
F4	SSLH	O4	Expansion				
		O5	Swap				
		O6	TRW-S				

Table 4.2 Notations of factor options

For simplicity, in the following subsections, we only present full results under condition F1-O1-B1-P1, which is close to the original setting in Liu and colleague's paper [1]. Section 4.2.1 shows results for hand image set and Section 4.2.3 is for hip image set.

4.2.1 Results on hand images

The resulting segmentations compared to ground truth of hand images, under condition F1-O1-B1-P1are shown in Fig. 4.5.

Qualitatively, the best results were achieved in image 3, 4, and 7. In the 3rd image, the tip of middle finger is over-extended. In the 4th image, there are clear errors in the thumb. In the 7th image, parts of the background are incorrectly labeled as soft tissue.

In the baby hand image, the small (pinky) finger is not recognized at all and the ring figure is mistaken for the small finger. The middle finger is incorrectly labeled as two fingers (ring and middle finger). The joint spaces are not preserved and are labeled as bony structures. The carpals, ulna and radius are over-labeled.

In the 2nd image, the middle finger is labeled as ring finger; hence there are two middle fingers in the result. The thumb and index finger are mixed up as well.

In the 5th image, the rheumatoid arthritis hand, most of the labeling is incorrect. Only the 1st, 2nd, 3rd, and 4th phalanges are partially recognized, leaving all other structures scattered in the scene.

The worst result occurs in the 6th image, which is the 90 degree rotated version of 3rd image. The result does not seem rotated at all and oddly resembles the orientation of the template.

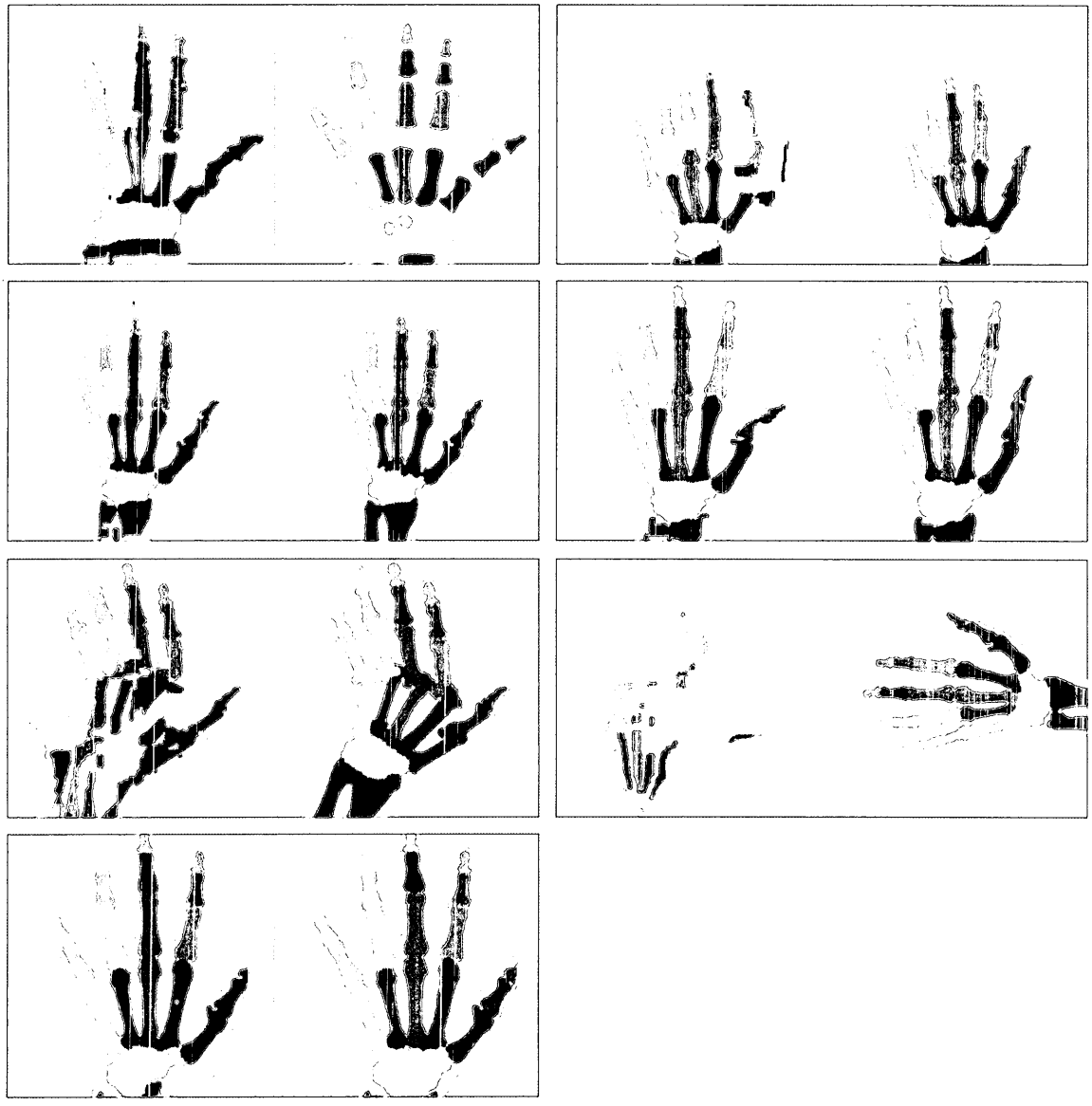


Fig. 4.5 Segmentation under F1-O1-B1-P1and ground truth on hand images

There are 7 pairs of resulted segmentation and corresponding ground truths are shown in this figure. In each pair, the left image is the resulted label map and the right is its ground truth.

The overall correctness rates and structure specific correctness rates of all 7 images are shown in Table 4.3. The 2nd, 3rd, 4th and 7th images resulted in over 90% correctness. But if we exclude classes 'Hand' and 'Picture', which occupy large portion of the images, only the 3rd, 4th and 7th images yields correctness rate over 90%. For the 1st and 5th images, the overall correctness rates without considering 'Hand' and 'Picture' are not satisfactory. The 6th image gives 0 correctness rate in terms of bony structures.

Among bony structures, the worst recognized structures are 'Ulna', 'Radius', '1st distal phalange' and '5th distal phalange'.

The boxplots of correctness rates, false negative rates and false positive rates across images are shown in Fig. 4.6. Due to the poor result of image 6, we remove image 6 from the boxplot generation.

The median false negative rates vary from 0.02 to 0.4 among bony structures, but are generally under 0.2. The median false positive rates vary from 0.02 to 0.3, and are also generally under 0.2.

The variations of correctness rates, false positive rates, and false negative rates are generally large, which indicates that the recognition rates vary a lot widely among different images.

	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Mean	Mean excluded image 6
Overall 1	81.9	93.8	98.1	95.8	75.4	70.5	94.0	87.1	89.8
Overall 2	54.9	72.9	92.9	91.5	28.9	0.0	91.8	61.8	72.1
Struct 1	0.0	90.7	71.3	71.0	0.0	0.0	84.7	45.4	52.9
Struct 2	1.6	96.5	86.5	74.7	0.0	0.0	0.0	37.0	43.2
Struct 3	78.0	90.4	97.4	98.7	90.3	0.0	96.3	78.7	91.9
Struct 4	95.0	1.5	96.0	46.7	73.4	0.0	93.9	58.1	67.7
Struct 5	99.6	0.0	88.7	0.0	52.9	0.0	67.9	44.1	51.5
Struct 6	89.8	90.3	90.6	95.5	2.7	0.0	93.9	66.1	77.1
Struct 7	99.4	31.7	98.1	98.1	59.6	0.1	95.0	68.9	80.3
Struct 8	72.5	0.0	98.6	97.8	90.6	0.0	96.8	65.2	76.1
Struct 9	94.2	0.0	94.6	95.7	83.2	0.0	92.3	65.7	76.7
Struct 10	56.4	94.8	97.8	96.1	0.0	0.0	95.4	62.9	73.4
Struct 11	65.9	38.0	97.1	99.0	63.2	0.0	95.8	65.6	76.5
Struct 12	53.0	0.0	96.1	97.3	94.4	0.0	95.2	62.3	72.7
Struct 13	90.0	0.0	83.8	95.8	83.8	0.0	89.3	63.2	73.8
Struct 14	17.3	86.7	89.5	92.8	0.0	0.0	94.9	54.4	63.5
Struct 15	0.0	96.6	96.9	99.4	41.1	0.0	96.2	61.4	71.7
Struct 16	0.0	96.3	97.1	97.8	95.9	0.0	95.4	68.9	80.4
Struct 17	0.0	80.3	93.5	96.1	81.3	0.0	89.0	62.9	73.4
Struct 18	83.6	94.2	95.6	97.7	0.0	0.0	94.7	66.5	77.6
Struct 19	0.0	96.0	97.3	98.8	0.0	0.0	96.6	55.5	64.8
Struct 20	0.0	94.5	98.1	98.3	0.0	0.0	89.3	54.3	63.4
Struct 21	0.0	76.4	93.6	94.5	0.0	0.0	83.0	49.7	57.9
Struct 22	56.3	91.4	94.8	87.2	69.9	18.8	88.4	72.4	81.3
Struct 23	100.0	95.6	94.2	97.7	0.0	0.0	81.8	67.0	78.2
Struct 24	99.7	97.2	99.6	98.8	95.2	92.2	96.9	97.1	97.9
Mean of structures	52.2	64.1	93.6	88.6	44.9	4.6	87.6		

Table 4.3 Recognition correctness rates on hand images under F1-O1-B1-P1

The overall rate 1 represents correctness rate considering all existing structures in an image. The overall rate 2 only considers structures excluding those labeled "Picture" and "Hand" (colored in darker gray).
Note: Figures in this table are in unit percentage.

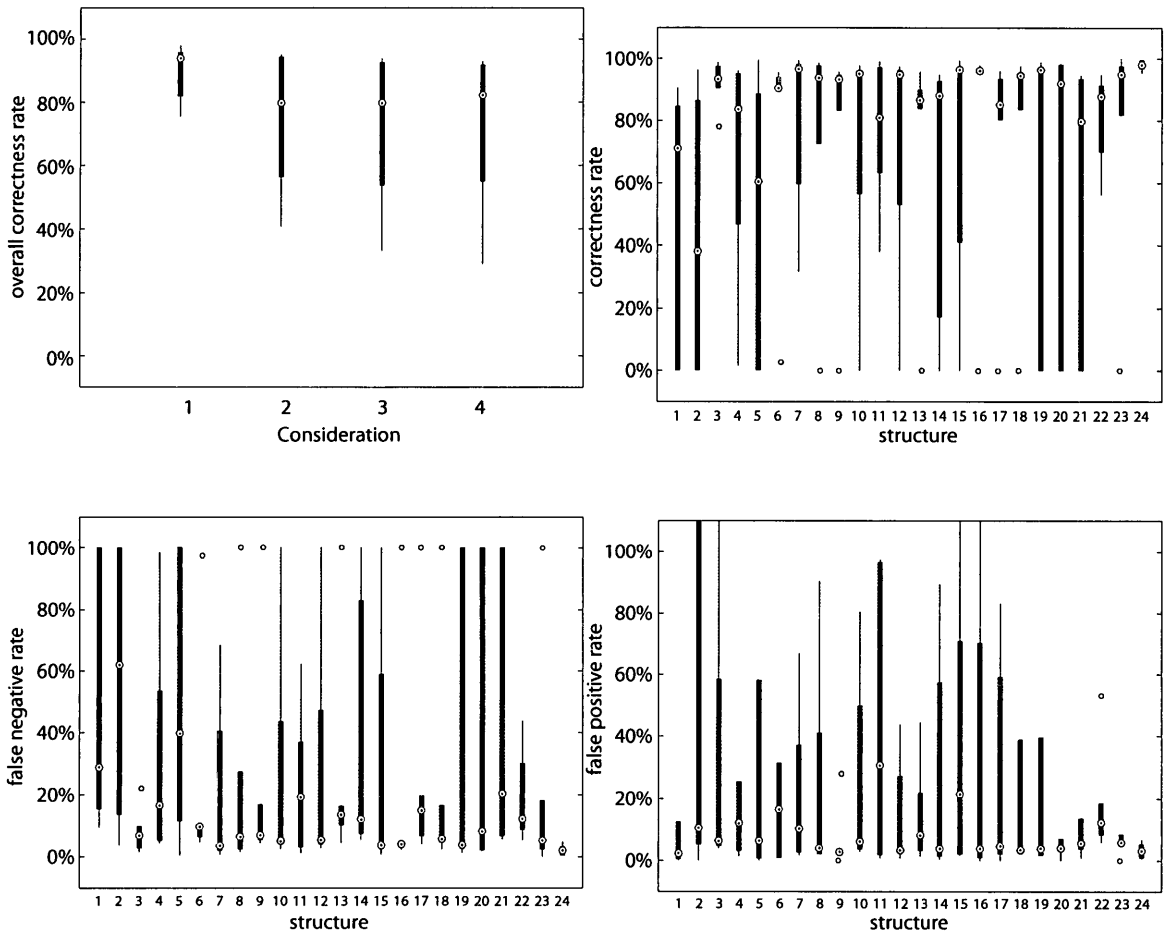


Fig. 4.6 Result boxplots of hand image set under F1-O1-B1-P1

The top left diagram shows the overall recognition correctness rates: (1) all structures are considered; (2) structures with labels, Ulna[1], Radius[2], Hand[22], Carpals[23], Picture[24], are excluded; (3) structures with labels, Hand[22], Carpals[23], Picture[24], are excluded; (4) structures with labels, Hand[22], Picture[24], are excluded.

The top right diagram shows the structure specific recognition correctness rates: names for structures can be found in Fig. 4.1.

The bottom left diagram shows the structure specific false negative error rates.

The bottom right diagram shows the structure specific false positive error rates, some structures may have error rates higher than 100%, such as structure Radius[2].

Note: (1) these boxplots are generated by excluding image 6 due to its extremely bad result. The red and blue color coding is used to group the finger bones into five individual fingers.

4.2.2 Replacing Feature Flow for One Hand Image

The misplaced labels produced in the hand image (for example, in hand image 2) may be due to the discontinuity preserving characteristics of the feature flow algorithm. To investigate this issue, we replaced the feature flow algorithm with a thin plate spline warp of the template hand to the target hand. The thin plate spline warp was computed by manually selecting corresponding points on the outlines of the entire hand of the template and target hand images. The thin plate spline warp was applied to the entire image to obtain the flow field from the template to the target hand; we called this flow field TPS-flow. The TPS-flow was given to the label transfer module to obtain the final segmentation. The results were slightly improved compared to the other variations, but it was still not satisfactory. Fig. 4.7 shows the warping result using TPS-flow. Table 4.4 shows the correctness rate comparison between using TPS-flow and using SIFT-flow.

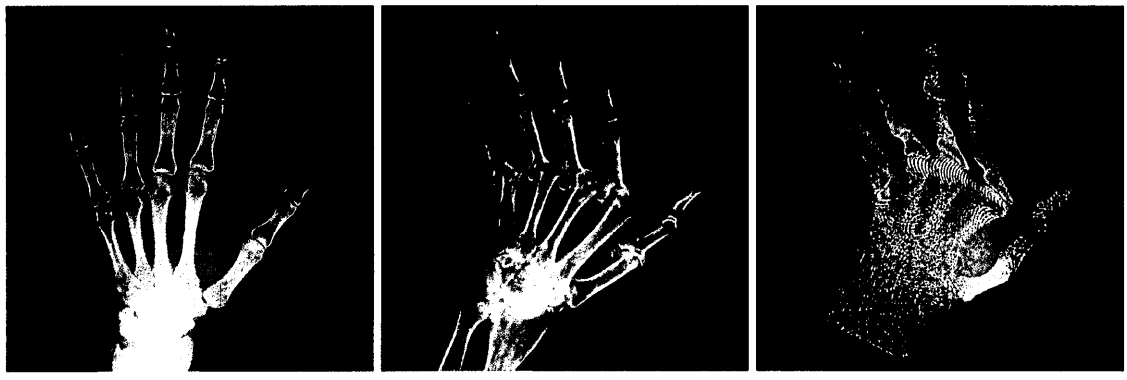


Fig. 4.7 Thin plate spline warping of the template to target hand

(Left) 20 landmarks were manually chosen on the outline of the template hand image [2]. (Middle) the 20 corresponding landmarks were manually chosen on the outline of the target hand image [5]. (Right) A thin plate spline warp that matched the landmarks from the template hand to the target hand was computed and applied to the template hand.

	Correctness Rate		False negative rate		False positive rate	
	TPS-flow	SIFT-flow	TPS-flow	SIFT-flow	TPS-flow	SIFT-flow
Overall 1	**78.07**	75.43	N/A	N/A	N/A	N/A
Overall 2	**43.71**	28.90	N/A	N/A	N/A	N/A
Mean of structures	**46.79**	44.90	**53.21**	55.10	**23.02**	43.63

Table 4.4 Performance comparison between TPS-flow and SIFT-flow

Overall correctness rate 2 is calculated by excluding labels "Picture" and "Hand". Numbers are in percentage. Bold numbers indicate better performance. False negative rates and false positive rates are structure specific; overall correctness rates are not applicable to these rates.

4.2.3 Results on hip images

For the hip images, the resulted segmentations under condition F1-O1-B1-P1and their corresponding ground truths are shown in Fig. 4.8.

Qualitatively, considering the accuracy of the essential structures (particularly, the pelvic sockets and femurs), we see that the results on the 1st, 2nd, 3rd, 5th, and 11th images are the best. The left sockets in 1st, 3rd images, and the right sockets in 3rd, 11th image are under-labeled. There is a common over-labeling problem of the right sockets, in which the system tries to resemble the appearance of the corresponding area on the template image.

There is also a common problem that the system labels the soft tissues or organs as a part of the pelvis as in image 4, 5, 6, 7, 9 and 13. Or conversely, the system labels the overlapping area of bony structure and soft tissues as a part of "Body", which appears in image 10, 12 and 13.

The worst results are achieved from the 8^{th} and 12^{th} images. In image 8, the left pelvis and tail bones are mixed up; the right femur, right socket and right pelvis are also incorrectly labeled. In image 12, both sides of the pelvis and socket areas are largely mislabeled.

The overall correctness rates and structure specific correctness rates of all 13 images are shown in Table 4.5.

Excluding labels 'Body' and 'Picture', we find that 8 images (1-5, 7, 9, and 11) have approximately 90% correctness rates, which compares favorably to the results in hand image set. But if we consider only the socket areas, the results are not acceptable.

Among the bony structures, both pelvic sockets are poorly recognized (correctness rates, left: 44.5%, right: 63.1%). Other bony structures are recognized with correctness rates over 80%.

The boxplots of correctness rates, false negative rates and false positive rates across images are shown in Fig. 4.9. Median false negative rates and false positive rates also reflect the poor recognition performance on the socket areas. General median error rates among structures are below 0.2, but for the two sockets, the rates go up, especially, the median false positive rate of right pelvic socket is even more than 1.

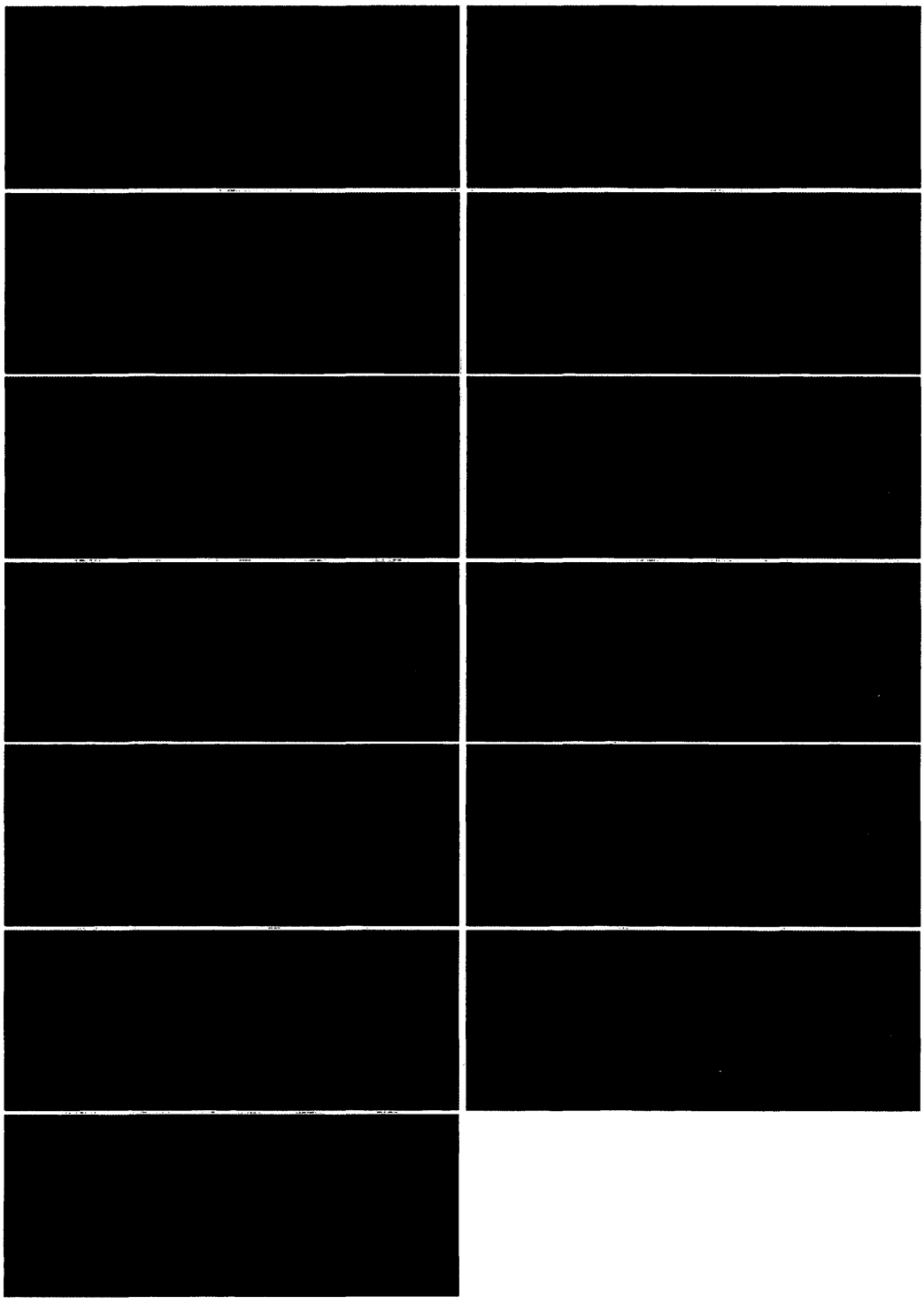


Fig. 4.8 Segmentation under F1-O1-B1-P1and ground truth on hip images

There are 13 pairs of resulted segmentation and corresponding ground truths are shown in this figure. In each pair, the left image is the resulted label map and the right is its ground truth.

	Img 1	Img 2	Img 3	Img 4	Img 5	Img 6	Img 7	Img 8	Img 9	Img 10	Img 11	Img 12	Img 13	Mean
Overall 1	90.9	90.0	87.8	79.0	83.6	79.1	87.0	74.9	86.2	83.1	88.2	72.7	79.9	83.3
Overall 2	90.3	91.9	93.1	90.2	91.4	85.9	89.4	76.6	89.5	81.3	92.4	73.0	79.3	86.5
Struct 1	94.2	86.5	81.9	72.5	73.5	64.7	81.7	70.6	82.0	85.3	85.7	73.5	84.8	79.8
Struct 2	95.8	84.8	95.1	93.3	75.4	86.5	90.6	82.5	86.6	81.9	97.5	90.8	49.2	85.4
Struct 3	97.4	80.7	97.2	79.1	94.0	86.5	94.0	85.1	89.0	89.5	96.6	81.5	79.6	88.5
Struct 4	90.6	94.4	93.7	96.2	96.2	86.0	96.6	82.6	88.5	87.5	93.4	77.5	82.4	89.7
Struct 5	87.9	93.6	68.0	93.5	84.4	85.0	94.4	84.0	97.9	97.3	84.7	74.9	71.9	86.0
Struct 6	98.9	97.8	96.5	98.1	99.7	98.7	97.8	96.1	99.8	59.7	99.3	17.1	96.7	88.9
Struct 7	43.0	60.4	19.7	46.3	52.5	58.6	57.8	48.2	7.3	54.0	60.7	54.8	15.9	44.5
Struct 8	90.5	95.0	93.3	95.5	96.5	83.1	90.3	76.7	90.1	81.4	97.1	69.8	92.4	88.6
Struct 9	92.1	97.0	93.3	94.7	93.2	85.5	95.6	36.8	95.0	97.2	92.1	88.9	97.3	89.1
Struct 10	92.6	100.0	96.6	97.5	97.6	99.9	97.6	98.8	96.7	65.6	98.5	11.9	95.0	88.3
Struct 11	77.9	80.7	63.2	75.6	68.1	67.7	82.3	26.3	59.1	57.5	70.2	36.1	55.4	63.1
Struct 12	78.5	23.1	97.4	37.7	0.0	75.1	76.6	0.0	48.5	90.7	69.2	7.6	33.8	49.1
Struct 13	86.3	90.1	92.6	84.8	89.0	89.5	78.7	69.6	92.4	69.8	85.0	69.4	76.1	82.6
Mean of struct	86.6	83.4	83.7	81.9	78.5	82.1	87.2	66.0	79.4	78.3	86.9	58.0	71.6	

Table 4.5 Recognition correctness rates on hip images under F1-O1-B1-P1

The overall rate 1 represents correctness rate considering all existing structures in an image. The overall rate 2 only considers structures excluding those labeled "Picture" and "Body" (colored in darker gray).
Note: Figures in this table are in unit percentage.

The variations of correctness rates and false negative rates are generally smaller than those of hand image set. However, the variations of false positive rates, especially, in the two socket structures, are large, which indicates the system cannot reliably segment such structures.

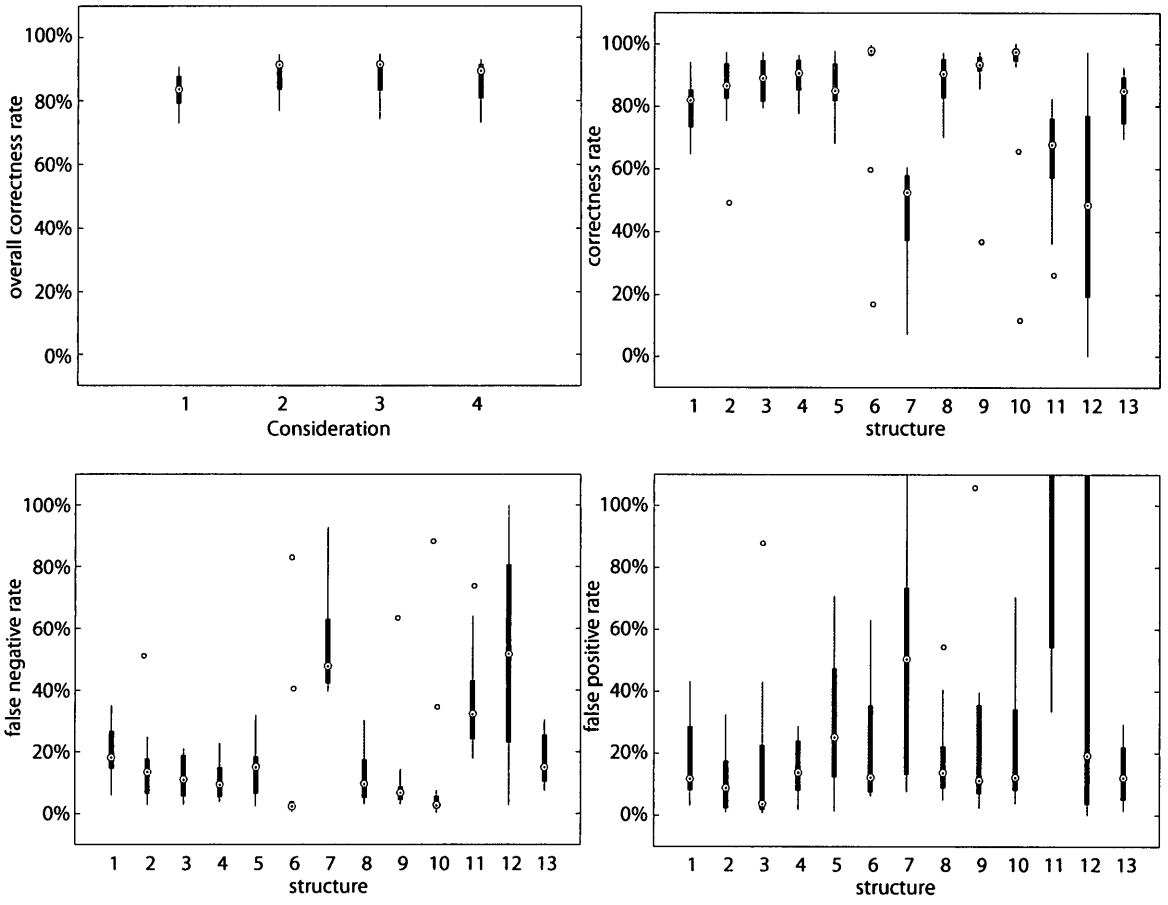


Fig. 4.9 Result boxplots of hip image set under F1-O1-B1-P1

The top left diagram shows the overall recognition correctness rates: (1) all structures are considered; (2) structures with labels, Body[1], PelvicHole(L)[6], PelvicHole(R)[10], Picture[12], TailBones[13] are excluded; (3) structures with labels, Body[1], Picture[12], TailBones[13], are excluded; (4) structures with labels, Body[1], Picture[12], are excluded.

The top right diagram shows the structure specific recognition correctness rates: names for structures can be found in Fig. 4.3.

The bottom left diagram shows the structure specific false negative error rates.

The bottom right diagram shows the structure specific false positive error rates, some structures may have error rates higher than 100%, such as structure Picture[12].

Note: The structures colored in red belong to the left side of the body. The structures colored in blue belong to the right side of the body.

4.3 Experiment analysis

The segmentation results under condition F1-O1-B1-P1 are not satisfying, even though the correctness rates of the hip image set are over 85%, which is above 76.67% achieved by Liu and colleagues [1]. This is because in medical image analysis, we generally require more accurate recognition. In searching for substantial improvement of the Label Transfer System, we incorporated alternations in the major components (factors) as stated in Section 3.2. We compared the recognition performance under 144 conditions and could not find significant improvements than that of F1-O1-B1-P1.

In this section, we describe the statistical performance differences controlled by the four factors separately. Particularly, we fix three factors and vary the other factor to determine if any alternation of this factor can improve the recognition performance. For example, we choose feature as 'SIFT', bilateral treatment as 'None' and neighborhood/prior as 'one template and one prior', then compare the results using the 6 different optimizers. We denote this configuration as 'F1-O*-B1-P1'. In the following subsections, we show the comparison and analysis of configurations, 'F*-O1-B1-P1', 'F1-O*-B1-P1', 'F1-O1-B*-P1', and 'F1-O1-B1-P*'. For each configuration, we give the statistic tables of structure-specific correctness rates and overall (image-specific) correctness rates; and a group of boxplots representing structure-specific recognition correctness rates, false negative rates, and false positive rates.

We also use ANOVA to determine the significance of the differences achieved by the different choices of factor in each configuration. To justify the ANOVA normal distribution assumption, we run Jarque-Bera tests under all factor combinations. The results show that the correctness rates across 7 hand images have the possibility of being normally distributed under all factor combinations. For hip images, under 88.2% of factor combinations, the correctness rates may be normally distributed. Hence, we assume the samples (images) represent the normally distributed populations, and ANOVAs on these samples are approximately valid. Then we give the n-way ANOVA result of all configurations 'F*-O*-B*-P*' excluding those options that produced decreased accuracy. In the end, we also report the effects on flow estimation time and recognition time caused by different selections of factors.

4.3.1 Analysis on feature descriptor selection

Under condition F*-O1-B1-P1, we compare the recognition accuracy of cases using different feature descriptor for flow estimation. Results show that the mean structure-specific correctness rate using STSF on the hand is 0.59438, which is significantly lower than the other descriptors (Table 4.6) with $F_{544,3} = 5.6, p < 0.0009$ in ANOVA. A similar result occurred for the hip images with mean structure-specific correctness rate of 0.6925, $F_{648,3} = 13.24, p < 0.0001$.

In terms of overall correctness rates (Table 4.7), the results are similar. STSF accuracy is significantly worse than the others. ($F_{15,3} = 3.4, p < 0.0454$ for hand images, $F_{36,3} = 7.0, p < 0.0008$ for hip images).

Boxplots shown in Fig. 4.10 also indicate that the variation of correctness rates and false negative rates are much larger by using STSF than by others. Also, the false negative rates of using STSF are obviously larger. We conclude that, STSF is not reliable as a feature descriptor.

Because we are looking for improvement in the system, inferior descriptors such as STSF should be removed from feature descriptor choices. ANOVA of structure-specific correctness rates without STSF shows that the difference among the remaining three is not significant, with $F_{544,3} = 5.6, p < 0.8805$ for hand images and $F_{544,3} = 0.3, p < 0.7375$ for hip images. However, in terms of overall correctness rates, with STSF removed, ANOVA shows that: there is no significant difference among the remaining features in hand image set with $F_{10,2} = 0.08, p < 0.9228$; but there is significant difference among the remaining features in hip image set with $F_{24,2} = 8.5, p < 0.0016$. From Table 4.7, we can see SIFT's accuracy is statistically higher than SURF and SSLH for hip images. The differences do not lead to the conclusion that SIFT is superior to SURF and SSLH such that SURF and SSLH should be excluded as capable feature descriptors.

Thus, we conclude that, only the STSF should be ruled out as a feature descriptor and change of descriptors cannot significantly improve recognition accuracy under conditions F*-O1-B1-P1.

Hand image set	SIFT	SURF	STSF	SSLH
mean	0.7183	0.7073	0.5948	0.7010
median	0.9064	0.8833	0.7173	0.8904
standard deviation	0.3657	0.3416	0.3833	0.3485
Hip image set	SIFT	SURF	STSF	SSLH
mean	0.7873	0.7769	0.6925	0.7866
median	0.8572	0.8594	0.7921	0.8665
standard deviation	0.2248	0.2181	0.2818	0.2112

Table 4.6 Structure specific statistics of correctness rate under F*-O1-B1-P1

In hand image set, statistical measures are calculated from 144 samples (24 structures in 6 test images, excluding image 6). In hip set, we have 169 samples (13 structures in 13 test images)

Hand image set	SIFT	SURF	STSF	SSLH
mean	0.7215	0.7420	0.5884	0.7351
median	0.8220	0.8182	0.6156	0.8228
standard deviation	0.2593	0.1972	0.3099	0.2076
Hip image set	SIFT	SURF	STSF	SSLH
mean	0.8649	0.8311	0.8014	0.8369
median	0.8951	0.8474	0.8019	0.8525
standard deviation	0.0670	0.0755	0.0785	0.0724

Table 4.7 Image specific statistics of correctness rate under F*-O1-B1-P1

In hand image set, statistical measures are calculated from overall correctness rates of 6 images. In hip set, we have 13 images.

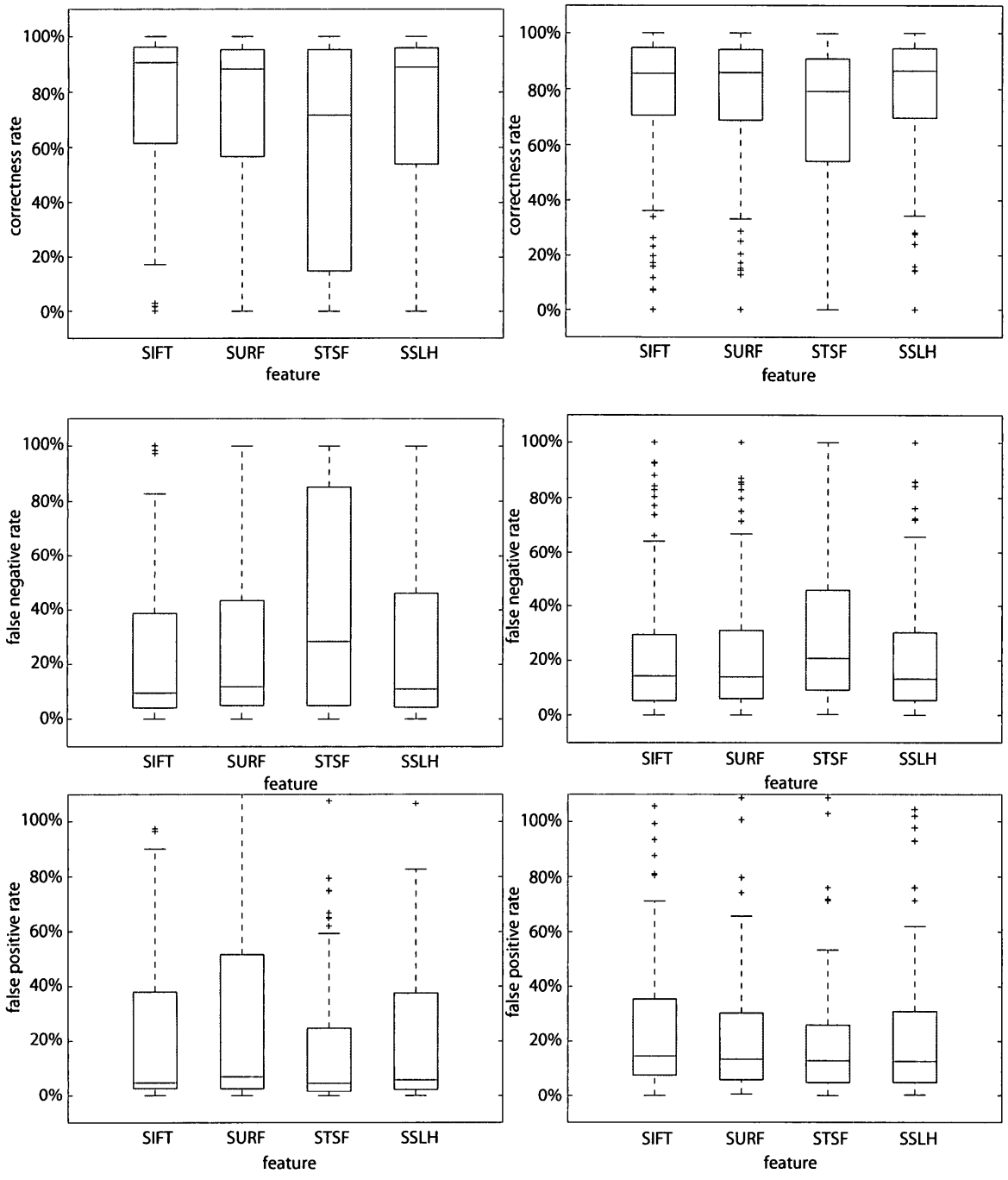


Fig. 4.10 Results on features (F*-O1-B1-P1)

Left columns are boxplots for hand image set and the right columns are for the hip image set. Plots on 1st row are correctness rates. Plots on 2nd row are for false negative rates and plots on 3rd row are for false positive rates.

4.3.2 Analysis on alternate MRF optimizers

Under condition F1-O*-B1-P1, the results show that the mean structure-specific correctness rate using BP-S on the hand is 0.2894, which is significantly lower than the other optimizers (Table 4.8) with $F_{830,5} = 48.82, p < 0.0001$ in ANOVA. A similar results occurs for the hip images with a mean correctness rate of 0.5422, $F_{984,5} = 41.93, p < 0.0001$. In terms of overall correctness rates (Table 4.9), the results are similar ($F_{25,5} = 89.88, p < 0.0001$ for hand images and $F_{60,5} = 60.85, p < 0.0001$ for hip images).

Boxplots in Fig. 4.11 also indicate that the variation of correctness rates and false negative rates are much larger using BP-S compared to the other optimizers. Also, the false negative rates of using BP-S stand out from others.

We believe that BP-S is not reliable as a MRF optimizer. ANOVA of structure-specific correctness rates without BP-S shows that the difference among the rest five is not significant, with $F_{687,4} = 0.02, p < 0.9994$ for hand images, and $F_{816,4} = 0.81, p < 0.5188$ for hip images. Similarly, ANOVA of overall correctness rates without BP-S indicates no significant accuracy difference with $F_{20,4} = 1.75, p < 0.1777$ for hand images and $F_{48,4} = 0.76, p < 0.5541$ for hip images.

Thus, we conclude that, BP-S should be ruled out as a MRF optimizer, and changing of optimizers cannot significantly improve recognition accuracy under conditions F1-O*-B1-P1.

Hand image set	BP-Liu	BP-M	BP-S	Expansion	Swap	TRW-S
mean	0.7183	0.7176	0.2894	0.7123	0.7116	0.7127
median	0.9064	0.9117	0.0000	0.9143	0.9143	0.9152
standard deviation	0.3657	0.3657	0.3621	0.3665	0.3664	0.3669
Hip image set	BP-Liu	BP-M	BP-S	Expansion	Swap	TRW-S
mean	0.7873	0.7840	0.5422	0.7699	0.7642	0.7675
median	0.8572	0.8576	0.6983	0.8624	0.8645	0.8559
standard deviation	0.2248	0.2271	0.3732	0.2525	0.2608	0.2553

Table 4.8 Structure specific statistics of correctness rates under F1-O*-B1-P1

In hand image set, statistical measures are calculated from 144 samples (24 structures in 6 test images, excluding image 6). In hip set, we have 169 samples (13 structures in 13 test images)

Hand image set	BP-Liu	BP-M	BP-S	Expansion	Swap	TRW-S
mean	0.7215	0.7216	0.3080	0.7155	0.7149	0.7145
median	0.8220	0.8225	0.3633	0.8194	0.8192	0.8184
standard deviation	0.2593	0.2611	0.2093	0.2644	0.2643	0.2641
Hip image set	BP-Liu	BP-M	BP-S	Expansion	Swap	TRW-S
mean	0.8649	0.8651	0.7643	0.8656	0.8654	0.8643
median	0.8951	0.8986	0.7769	0.8980	0.8981	0.8953
standard deviation	0.0670	0.0666	0.0774	0.0664	0.0662	0.0660

Table 4.9 Image specific statistics of correctness rates under F1-O*-B1-P1

In hand image set, statistical measures are calculated from overall correctness rates of 6 images. In hip set, we have 13 images.

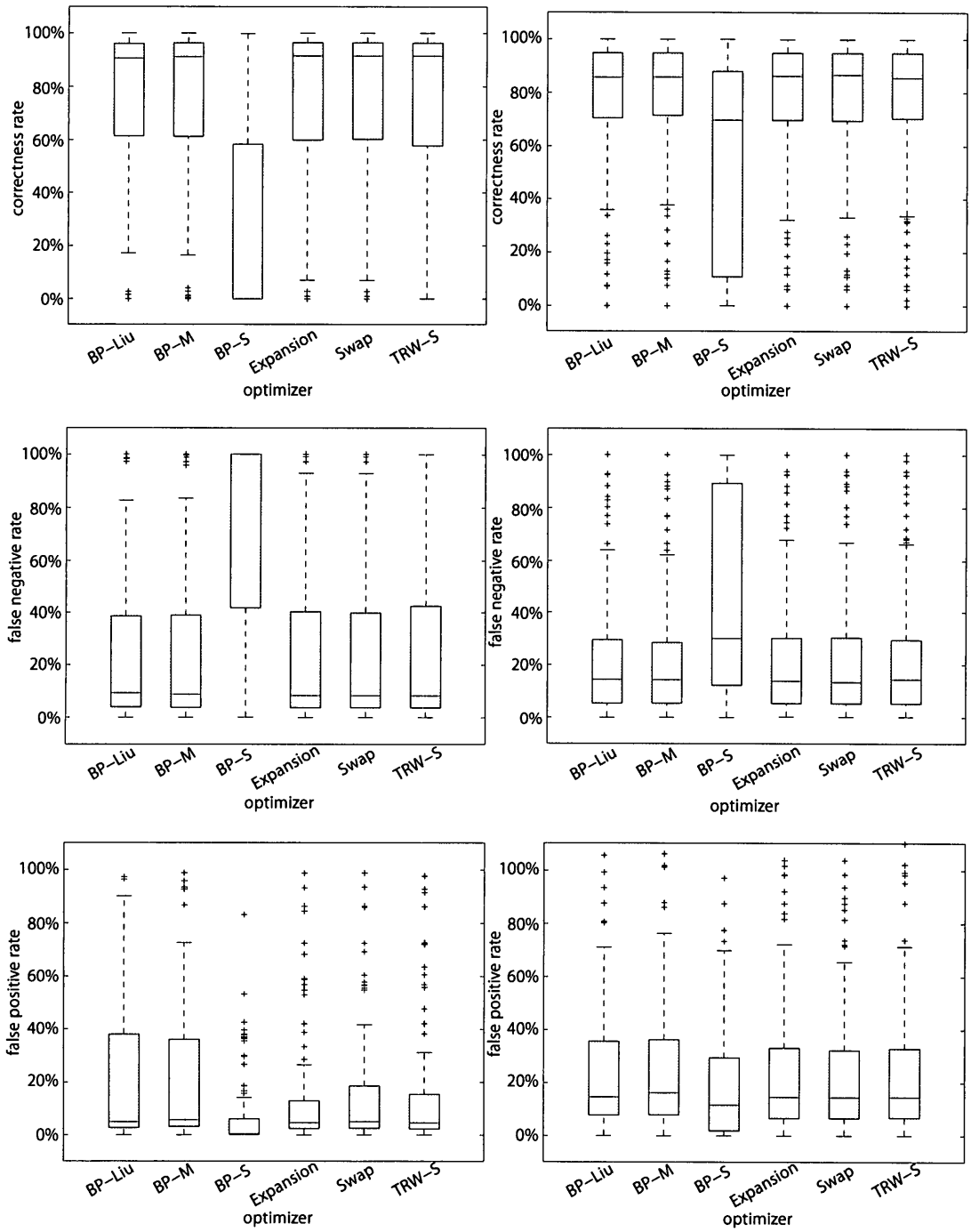


Fig. 4.11 Results on optimizers (F1-O*-B1-P1)

Left columns are boxplots for hand image set and the right columns are for the hip image set. Plots on 1st row are correctness rates. Plots on 2nd row are for false negative rates and plots on 3rd row are for false positive rates.

4.3.3 Analysis on bilateral treatments

Under conditions F1-O1-B*-P1, results of mean correctness rate are shown in Table 4.10. For both hand and hip image sets, the mean structure-specific correctness rates with and without using bilateral filter are very similar. ANOVA shows the differences are insignificant, with $F_{258,1} = 0.02, p < 0.8935$ for hand images, and $F_{312,1} = 2.54, p < 0.1120$. In terms of overall correctness rates (Table 4.11), the results are similar ($F_{5,1} = 0.42, p < 0.5470$ for hand images and $F_{12,1} = 2.75, p < 0.1233$ for hip images). Boxplots in Fig. 4.12 also indicate that the results with and without using bilateral filter are very similar, either for hand images or for hip images. We conclude that, change of bilateral treatments cannot significantly improve recognition accuracy under conditions F1-O1-B*-P1.

Hand image set	no filter	bilateral	**Hip image set**	no filter	bilateral
mean	0.7183	0.7136	mean	0.7873	0.7620
median	0.9064	0.9113	median	0.8572	0.8292
standard deviation	0.3657	0.3604	standard deviation	0.2248	0.2223

Table 4.10 Structure specific statistics of correctness rates under F1-O1-B*-P1

In hand image set, statistical measures are calculated from 144 samples (24 structures in 6 test images, excluding image 6). In hip set, we have 169 samples (13 structures in 13 test images)

Hand image set	no filter	bilateral	**Hip image set**	no filter	bilateral
mean	0.7215	0.7415	mean	0.8649	0.8497
median	0.8220	0.7844	median	0.8951	0.8611
standard deviation	0.2593	0.1970	standard deviation	0.0670	0.0595

Table 4.11 Image specific statistics of correctness rates under F1-O1-B*-P1

In hand image set, statistical measures are calculated from overall correctness rates of 6 images. In hip set, we have 13 images.

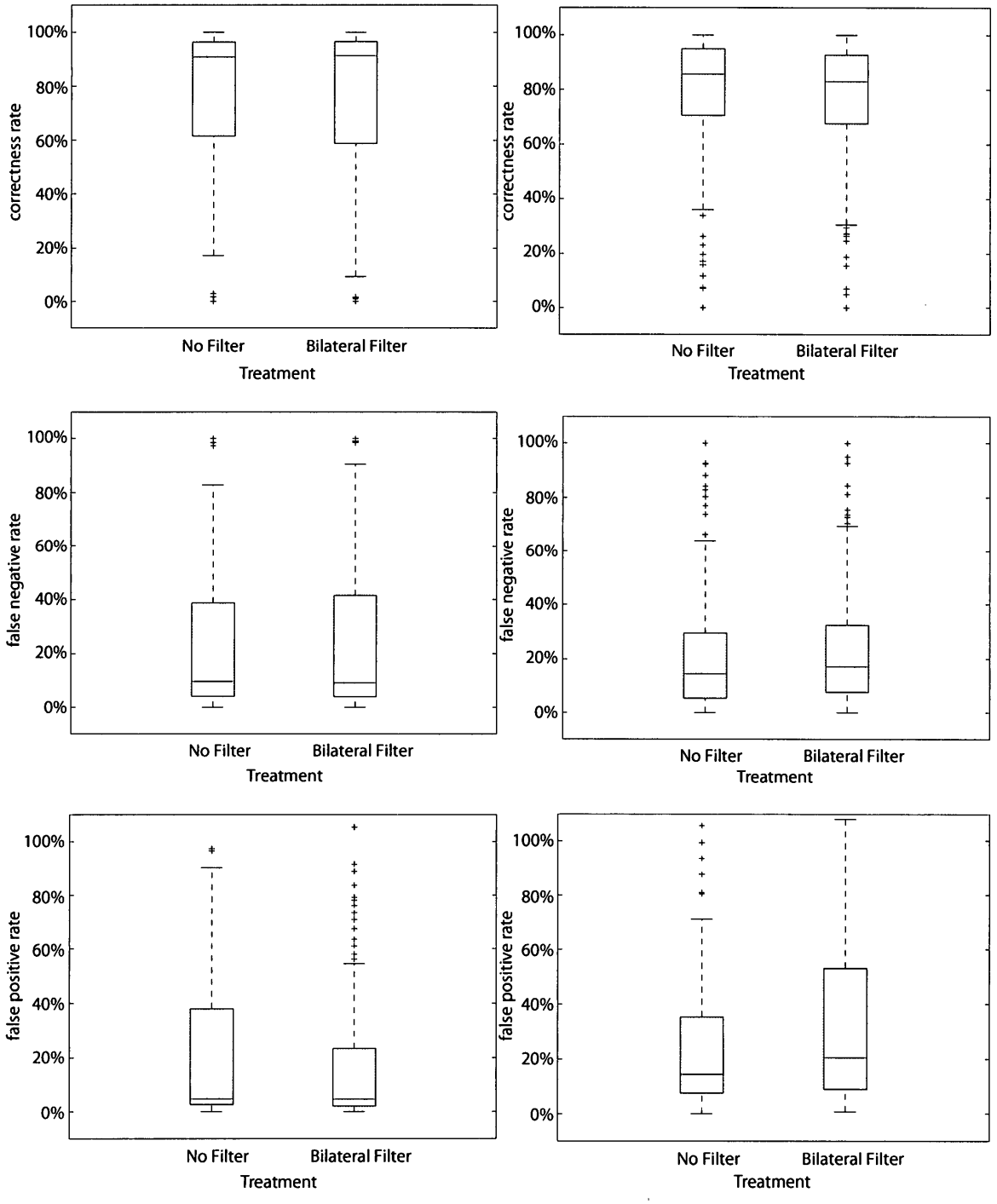


Fig. 4.12 Results on bilateral treatments (F1-O1-B*-P1)

Left columns are boxplots for hand image set and the right columns are for the hip image set. Plots on 1st row are correctness rates. Plots on 2nd row are for false negative rates and plots on 3rd row are for false positive rates.

4.3.4 Analysis on neighborhood and prior configurations

Under conditions F1-O1-B1-P*, the results of mean correctness rate are shown in Table 4.12. For both hand and hip image sets, the mean structure-specific correctness rates among using three different neighborhood/prior configurations are very similar. ANOVA shows the differences are insignificant, with $F_{401,2} = 1.39, p < 0.2499$ for hand images, and $F_{480,2} = 0.35, p < 0.7063$. In terms of overall correctness rates (Table 4.13), there is no significant accuracy difference among the three configurations for the hand images with $F_{10,2} = 1.33, p < 0.3085$; but there is statistically significant difference for the hip images with $F_{24,2} = 10.6, p < 0.0005$. From Table 4.13, we see that the overall correctness rates achieved by TkMP in hip images are larger than those obtained by T1SP and T1MP. But the differences do not lead to the conclusion that T1SP and T1MP is inferior and they should be excluded as capable configuration.

Boxplots in Fig. 4.13 indicate that the results among using three different neighborhood/prior configurations are very close, both for hand images and for hip images.

We conclude that, changing of neighborhood/prior configurations cannot significantly improve recognition accuracy under conditions F1-O1-B1-P*.

Hand image set	T1SP	TkMP	T1MP
mean	0.7183	0.7678	0.7215
median	0.9064	0.9255	0.9179
standard deviation	0.3657	0.3068	0.3659
Hip image set	T1SP	TkMP	T1MP
mean	0.7873	0.8008	0.7917
median	0.8572	0.9089	0.8555
standard deviation	0.2248	0.2491	0.2198

Table 4.12 Structure specific statistics of correctness rates under F1-O1-B1-P*

In hand image set, statistical measures are calculated from 144 samples (24 structures in 6 test images, excluding image 6). In hip set, we have 169 samples (13 structures in 13 test images)

Hand image set	T1SP	TkMP	T1MP
mean	0.7215	0.7663	0.7253
median	0.8220	0.8829	0.8246
standard deviation	0.2593	0.2109	0.2583
Hip image set	T1SP	TkMP	T1MP
mean	0.8649	0.9105	0.8655
median	0.8951	0.9086	0.8970
standard deviation	0.0670	0.0348	0.0665

Table 4.13 Image specific statistics of correctness rates under F1-O1-B1-P*

In hand image set, statistical measures are calculated from overall correctness rates of 6 images. In hip set, we have 13 images.

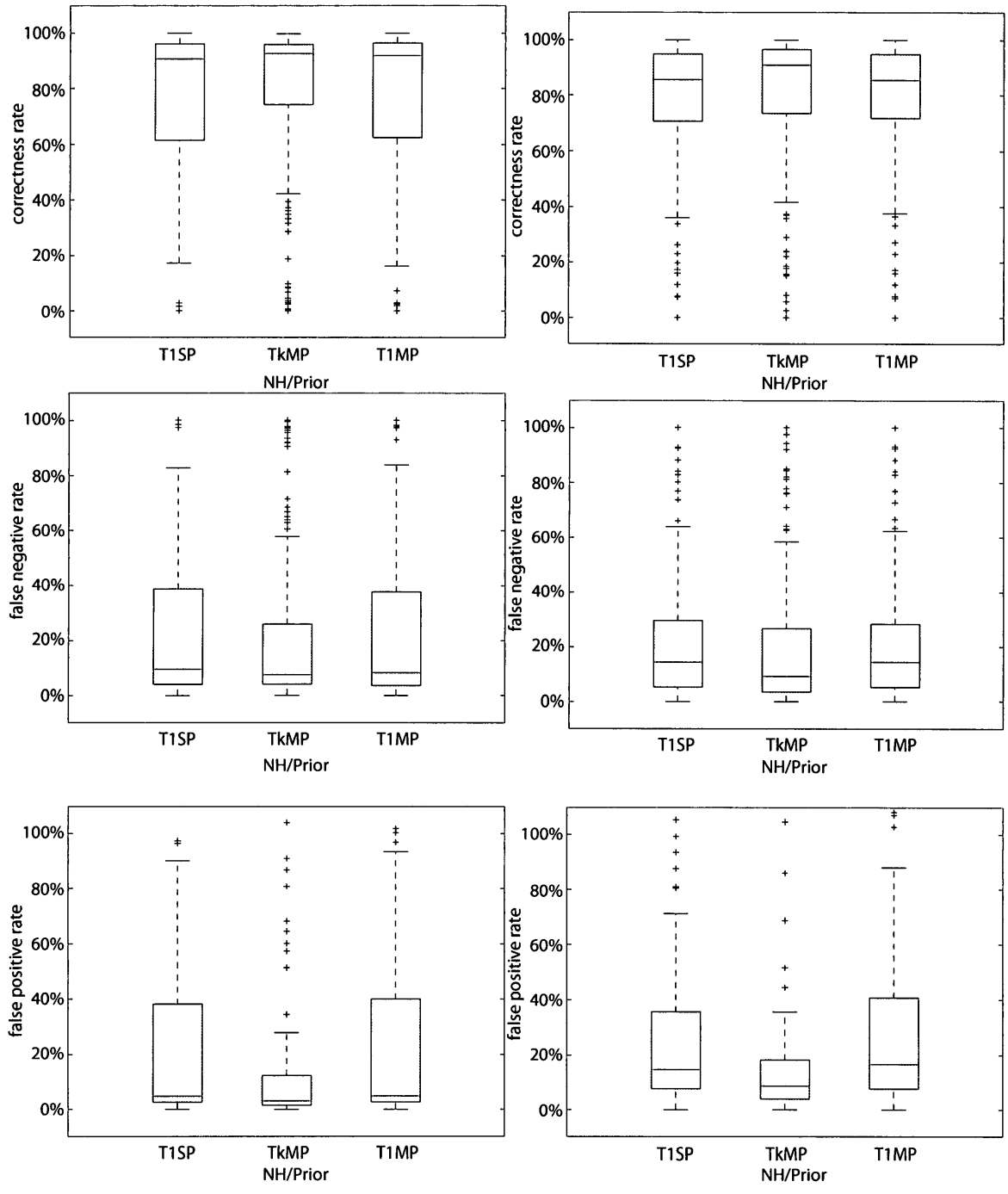


Fig. 4.13 Results on neighborhood system and prior selections (F1-O1-B1-P*)

Left columns are boxplots for hand image set and the right columns are for the hip image set. Plots on 1st row are correctness rates. Plots on 2nd row are for false negative rates and plots on 3rd row are for false positive rates.

4.3.6 N-way ANOVA on four factors

In the previous subsections, we analyze the configurations 'F*-O1-B1-P1', 'F1-O*-B1-P1', 'F1-O1-B*-P1', and 'F1-O1-B1-P*'. We found that STSF descriptor and BP-S optimizer were inferior in achieving more accurate recognitions. We also found that after removing STSF, the remaining descriptors do not produce significant differences in correctness rates. Similarly, the remaining 5 optimizers also do not produce significant differences in correctness rates. The bilateral treatments or neighborhood/prior configurations also do not produce significant differences. Now, we want to verify whether this situation stand for all configurations, i.e. F*-O*-B*-P*.

We conduct 5-way ANOVA, which involves factors: image, feature descriptor, optimizer, bilateral treatment, and neighborhood/prior configuration, with STSF and BP-S results being stripped out from the dataset.

The result shows that, the choice of optimizer cannot significantly change the recognition accuracy, with $F_{525,4} = 0.03, p < 0.9983$ for hand image set and $F_{1148,4} = 0.09, p < 0.9861$ for hip image set. However, neighborhood/prior configuration, feature and bilateral treatment still play important roles on accuracy. This indicates, for some specific configurations, recognition performance may benefit from some particular choice of bilateral treatment, feature or neighborhood/prior configuration. But this beneficial choice is not consistent across all configurations. Another finding is that, in all ANOVA mentioned above, factor 'image' always gives significant impact on the performance.

This may indicate that the Label Transfer System performs differently across different test images, which violates our intention to find a generic automatic segmentation system for medical images.

4.3.7 Analysis on label transfer recognition time and feature flow estimation time

From ANOVA, we find that the major factor on label transfer recognition time is the MRF optimizer, with $F_{847,5} = 341.93, p < 0.0001$ for hand images and $F_{1848,5} = 1416.52, p < 0.0001$ for hip ones. For hand images, factor neighborhood/prior, bilateral, and feature do not produce significant changes in label transfer time. For the hip images, all factors have statistically significant effects on label transfer time. The mean label transfer time among different optimizers can be found in Table 4.14. Boxplots on optimizers of label transfer recognition times across all images are shown in Fig. 4.14. Among the six optimizers, BP-S produced the fastest recognition time, which is 34% faster than BP-Liu in the hand image set, and 29% in the hip image set. TRW-S is slightly slower than BP-S in the hip image set, but 33% faster than BP-Liu. In the hand image set, TRW-S is slightly faster than BP-S, and 30% faster than BP-Liu. The graph cut based variations, Expansion and Swap run slowest, with almost twice longer than BP-Liu in hand image set, but close to BP-Liu in hip image set. The number of pixels in a hand image is 2.38 times of that in a hip image; the running time in hand image is 4.35 times of that in hip image using BP-Liu. But using Expansion or Swap, the time ratios become 10.42 and 22.58.

Hand image set	BP-Liu	BP-M	BP-S	Expansion	Swap	TRW-S
mean	9.22	8.54	6.57	29.49	27.09	6.47
median	9.16	8.54	6.56	28.52	26.66	6.46
standard deviation	0.20	0.10	0.03	12.16	12.30	0.03
Hip image set	BP-Liu	BP-M	BP-S	Expansion	Swap	TRW-S
mean	2.12	2.34	1.40	2.83	2.34	1.43
median	2.11	2.34	1.40	2.78	2.32	1.42
standard deviation	0.05	0.03	0.01	0.60	0.32	0.01

Table 4.14 Statistics of label transfer time using different MRF optimizers

In hand image set, statistical measures are calculated from 168 samples (4 features, 2 bilateral treatments, 3 neighborhood/prior configurations in 7 test images). Similarly, in hip set, we have 312 samples for the 13 test images. The unit is second.

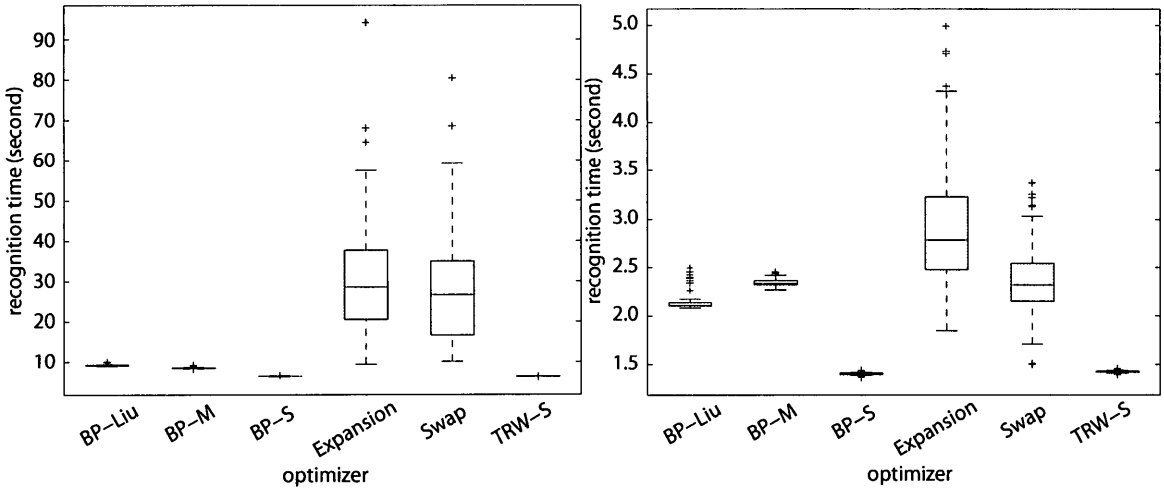


Fig. 4.14 Recognition time on optimizers

Left boxplots are for hand image set and the right ones are for the hip image set. The unit is second.

Considering the results in Section 4.3.1 that BP-S is not a suitable MRF optimizer due to the poor accuracy, and that the other 5 optimizers produce insignificant accuracy differences, we believe TRW-S is the better optimizer for this system.

Feature flow running time is not dependent on the optimizers, neighborhood/prior configuration or bilateral treatments because feature flow is the first step in the system. Here, we compare the flow estimation times using 4 different feature descriptors. Table 4.15 shows the mean and median of flow estimation times using the four different feature descriptors. Boxplots of label transfer recognition times on feature descriptors across all images are shown in Fig. 4.15.

Hand image set	SIFT	SURF	STSF	SSLH
mean	67.23	58.13	76.86	125.36
median	67.37	58.18	76.85	125.43
standard deviation	0.86	0.57	0.63	1.22
Hip image set	SIFT	SURF	STSF	SSLH
mean	28.19	24.25	31.80	52.44
median	28.17	24.25	31.79	52.40
standard deviation	0.25	0.21	0.31	0.44

Table 4.15 Statistics of flow estimation time using different feature descriptors

In hand image set, statistical measures are calculated from 42 samples (2 bilateral treatments, 3 neighborhood/prior configurations in 7 test images). Similarly, in hip set, we have 78 samples for the 13 test images. The unit is second.

SSLH-flow relies on the results of SIFT-flow and SURF-flow, hence its estimation time is just the sum of the other two. Because the STSF descriptor is the concatenation of SIFT and SURF descriptor, containing 192 components, it is reasonable that STSF-flow has a longer computation time than SIFT-flow. The SURF descriptor contains only 64 components, and the SIFT descriptor contains 128. Not surprisingly, SURF-flow was faster than SIFT-flow as well. Variations are very small; this indicates flow estimations

across different configurations are very consistent. Hence, SURF may be the best feature descriptor based on the fact there was insignificant accuracy differences among SIFT, SURF, and SSLH.

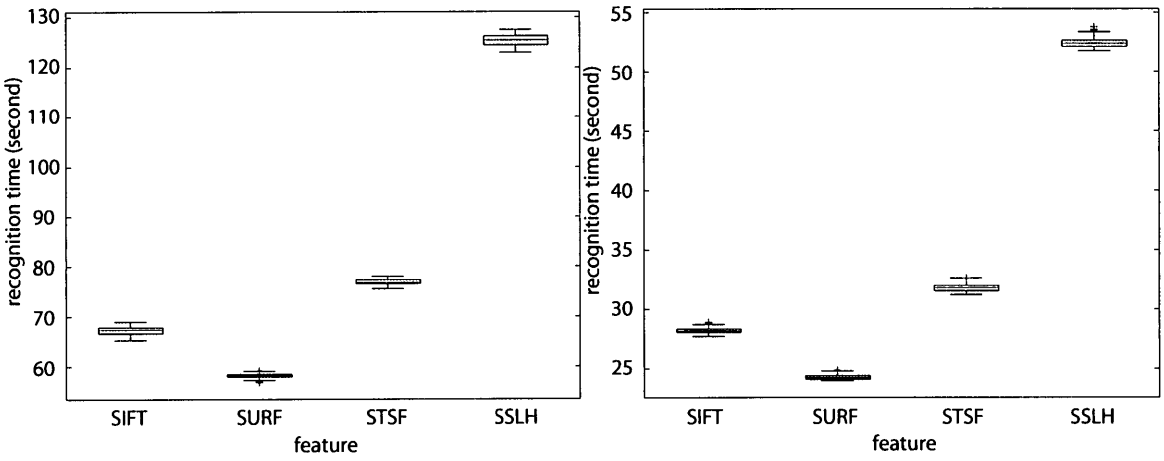


Fig. 4.15 Recognition time on feature

Left boxplots are for hand image set and the right ones are for the hip image set. The unit is second.

4.4 Discussion

The mean correctness rates across all 144 conditions are 89.3% for the hand image set (image 6 excluded) and 82.58% for the hip image set. The instances with best overall recognition rates are shown in Table 4.16. The correctness rates are higher than 95% among these best instances. The best recognized images are image 4 and image 7 in the hand image set, and image 9 and image 11 in the hip set. Although these results are quite high compared to those achieved in natural scene parsing by Liu and colleagues [1], we still cannot find solid evidence to prove the Label Transfer System is reliable in segmenting medical images.

First, despite the claim that utilizing SIFT descriptor can achieve good performance in rotation-invariant cases, our results show otherwise. For example, the recognition results of, the slightly rotated thumb in hand image 4, the rotated carpals in hand image 5 hand, and the extreme case of image 6, are very disappointing. Also, statistics of Table 4.3 shows that, among bony structures in hand, the 1^{st} distal phalange and the 5^{th} distal phalange are the worst recognized. This may be due to their outmost positions that cause rotation to produce the greatest displacement among all hand bones. The reason that SIFT descriptor does not give expected rotation invariant ability may lie on the fact that dense SIFT in Label Transfer System does not consider multiple orientations at single position and sacrifices detection and localization precisions as stated in Section 3.2.1. Also, the problem may be rooted in the mathematical model of flow estimation, which

articulates displacement penalties without addressing that existence of rotations may cause large displacements.

Hand image set										
Image #	4	4	4	4	4	4	7	7	7	7
Feature	4	2	2	3	3	3	3	3	3	3
Optimizer	3	1	1	6	4	2	1	2	6	5
Bilateral	2	2	2	2	2	2	2	2	2	2
NH/Prior	3	3	1	2	2	2	3	3	3	3
Correctness Rate 1	0.9722	0.9819	0.9820	0.9796	0.9827	0.9823	0.9539	0.9528	0.9590	0.9602
Correctness Rate 2	0.9653	0.9636	0.9634	0.9585	0.9584	0.9550	0.9543	0.9542	0.9537	0.9532
Hip image set										
Image #	11	9	11	9	11	11	9	11	11	11
Feature	4	1	4	1	4	2	4	2	2	4
Optimizer	1	4	2	5	6	1	6	2	2	5
Bilateral	1	1	1	1	1	1	1	1	2	1
NH/Prior	2	2	2	2	2	2	2	2	2	2
Correctness Rate 1	0.8632	0.9356	0.8606	0.9353	0.8598	0.8676	0.9642	0.8600	0.8504	0.8605
Correctness Rate 2	0.9611	0.9603	0.9601	0.9597	0.9592	0.9591	0.9589	0.9588	0.9586	0.9582

Table 4.16 Instances with highest overall correctness rates

Instances are sorted by correctness rate 2 in descending order. In hand image set, correctness rate 2 is calculated by excluding labels "Picture" and "Hand". In hip set, correctness rate 2 is obtained by excluding labels "Picture" and "Body".

Another finding is, for structures displaced by a large amount, as long as the features are highly distinguishable, the results are good. For example, in hip image 3, even though the image is shifted upwards compared to the other hip images, the structures are still well recognized. But for structures with less distinguishable features, the results become much

worse. For example, the middle finger and index finger of hand image 2 are mixed up because they are close to each other yet have similar structures. Small structures such as the hip sockets are not well distinguished from surrounding structures. Soft tissues in hip image set also cause ambiguity to the system. This indicates the Label Transfer System may require more distinguishable feature descriptor.

Another problem is that certain structures may not be recognized because the templates lack these structures or contain smaller portions of these structures; such structures include the Ulna, Radius in the hand images. This indicates, to a certain degree, the system depends on prior information on the training templates.

Although the system does not work well with extreme low quality images such as hip image 8, in general, the system performs well in images with signal inhomogeneity and low contrast contents. This may benefit from the luminance invariant of SIFT or SURF features.

In searching for better combinations of the four major factors (feature descriptor, MRF optimizer, preprocessing filter, neighborhood/prior configuration) in the system, we have the following findings. STSF is not considered a good feature descriptor. Among the remaining three descriptors, there is no significant difference in recognition accuracy. SURF gives fastest performance and SURF may be considered the best descriptor. Similarly, BP-S is not considered a good MRF optimizer. TRW-S is the fastest and is

considered the best choice because there were no significant accuracy differences among the remaining 5 optimizers. Bilateral filtering did not produce consistent differences for both hand images and hip images; however, the best recognized hand instances were treated by bilateral filter. Neighborhood/prior configurations did not produce significant differences. But since TkMP involves multiple training templates (K nearest neighbors), For each neighbor, the system needs to find the flow between this neighbor and the test image. It is likely ($K \geq 2$) that multiple flow estimations have to be calculated for one test image. Because the major bottle neck in this system is the flow estimation, TkMP is substantially slower than T1SP and T1MP. Also, T1MP traverses all training set to calculate prior but T1SP only needs one Gaussian filter operation. Hence, T1SP is considered a better approach based on the insignificant accuracy among the three configurations. From N-way ANOVA, we do not see changes of the 4 factors can give consistent or substantial improvements in terms of recognition accuracy.

Finally, as stated in Section 4.2.1, variations of recognition rates are generally large in the hand set. As stated in Section 4.3.6, n-way ANOVA showed that the 'image' factor had significant effects on performance. The 'image' factor seems to be the most influential factors of Label Transfer System; thus, the variations for the Label Transfer System we studied do not fulfill our goal to find a generic automatic segmentation system across variant medical images.

Chapter Five: Conclusions and Outlook

We explored and assessed the Label Transfer System for segmentation of medical x-ray images. We also attempted to modify certain factors in this system to search for performance improvements. The results showed recognition correctness rates higher than those in natural scene parsing applications, but the results were judged to be not reliable enough for clinical use. Variations of this system did provide performance improvements in certain images, but the improvements were not consistent across all images. In this chapter, we conclude the feasibility of using feature-flow and label transfer system to segment medical images with deformed anatomy in orthopedic surgery. Furthermore, we describe the future work can be done that may make use of this method in medical image segmentation.

5.1 Future work

Even though, we created and assessed many variations (144 in total) of the Label Transfer System by altering some major factors, there are still some possible options we did not implement in our assessment platform.

1) We did test different MRF optimizers in the label transfer module, but we did not implement these optimizers in the feature flow estimation procedure. Liu's belief propagation algorithm (BP-Liu) was designed to incorporate decoupling of the

smoothness term in flow estimation. A similar decoupling would need to be integrated with the other optimizers we studied before they could be used for feature flow estimation.

2) The feature flow estimation model may put too much constraint on displacements and thus limit the possibility of rotations. We may consider a better model that can take both into account.

3) The robust feature descriptors did not give the expected rotation-invariant performance. This may due to the fact that dense features omit the existence of multiple orientations at a single location. The lack of rotation invariance is problematic for highly articulated structures such as the hand where the angle between the individual fingers can vary, and the relative rotation between individual bones can vary dramatically (such as in the rheumatoid arthritis hand). A parts-based segmentation approach, where the individual parts that make up the object are detected, might be a better alternative for highly articulated anatomy.

5.2 Conclusions

While atlas-based methods and model-based methods are considered the most advanced state-of-art medical segmentation methods, they are built on specific or isolated structures and relatively consistent knowledge base. They have not been shown to work well in cases with highly variable anatomies, such as when unusual distortions and deformations are present. Liu and colleagues' Label Transfer System utilizes robust local feature flow and is able to preserve discontinuities between objects. This may give us the opportunity

to incorporate such a system in inventing reliable medical image segmentation methods that can handle unpredictable situations with least prior information. However, this goal is extremely difficult to achieve.

In this thesis, evaluation results of Label Transfer System and its variations showed that this system achieved better recognition rates than those in natural scene parsing applications. The high recognition rates were not consistent across all images or structures. And even in the best instances, the correctness rates were not accurate enough to be considered clinically practical.

Variations achieved by altering factors such as, feature descriptor, MRF optimizers, preprocessing filters, and neighborhood/prior settings obtained performance improvement for specific images or structures. But again, these improvements did not appear to be consistent.

We conclude that the Label Transfer System and its variations presented in this paper cannot provide reliable segmentation for medical images. Although they cannot be used as a standalone segmentation tool in practice, it is still possible that we may improve this system by other means or incorporate this system with other medical segmentation tools.

REFERENCES

[1] C. Liu, J. Yuen and A. Torralba, "Nonparametric Scene Parsing via Label Transfer," *IEEE Transactions On Pattern Analysis And Machine Intelligence,* vol. 33, no. 12, 2011.

[2] V. Gilsanz and O. Ratib, Hand bone age: a digital atlas of skeletal maturity, Springer, 2005.

[3] [Online]. Available: http://www.diaxray.com/services/xray.html.

[4] [Online]. Available: http://www.flickr.com/photos/tracemeek/5327224133/.

[5] [Online]. Available: http://www.visualphotos.com/image/2x4139561/arthritic_hand_x-ray_of_the_hand_of_a_patient_with.

[6] [Online]. Available: http://drugline.org/ail/pathography/1814/.

[7] [Online]. Available: http://camilleherron.com/2012/02/05/my-experience-with-hernias-and-a-labral-tear/.

[8] [Online]. Available: http://www.wheelessonline.com/ortho/inlet_and_outlet_views.

[9] [Online]. Available: http://orthofuture.org/ooth.html.

[10] [Online]. Available: http://www.kneehipjointcare.com/hipjointarthritis.html.

[11] [Online]. Available: http://eorif.com/HipThigh/Xray%20Hip.html.

[12] [Online]. Available: http://www.zimbio.com/Fosamax+Recall/articles/ibEx4uJhEpH/Osteomalacia+Vitamin+Important+Bone+Health.

[13] [Online]. Available: https://picasaweb.google.com/lh/photo/YrtkVa_xRdGt7G6M6Dnzcw.

[14] [Online]. Available: http://www.endotext.org/parathyroid/parathyroid15/parathyroidframe15.htm.

[15] [Online]. Available: http://www.sciencephoto.com/media/251347/view.

[16] [Online]. Available: http://aluyachting.it/rectus-femoris-avulsion-fracture&page=6.

[17] [Online]. Available: http://www.hip-clinic.com/en/news/115-orthopaedics-this-week.

[18] [Online]. Available: http://samisarkis.photoshelter.com/image/I00007w37.5gwSKM.

[19] [Online]. Available: http://hipchicksunite.com/blog/?page_id=76.

[20] B. Ma, "Robust surface-based registration from sparse measurements," 1998.

[21] J. Maintz and M. Viergever, "A survey of medical image registration," *Medical Image Analysis,* vol. 2, pp. 1-36, 1998.

[22] A. Elnakib, G. Gimel'farb, J. S. Suri and A. El-Baz, "Medical Image Segmentation: A Brief Survey," in *Multi modality state-of-the-art medical image segmentation*

and registration methodologies. Volume II, New York, Springer, c2011, pp. 1-38.

[23] D. L. Pham, C. Xu and J. L. Prince, "Current Methods in Medical Image Segmentation," *Annu. Rev. Biomed. Eng.*, vol. 02, pp. 314-37, 2000.

[24] W. Niessen, "Model-Based Image Segmentation for Image-guided Interventions," in *Image-guided interventions: technology and applications*, New York, Springer, 2008, pp. 219-239.

[25] C. Harris, "Geometry from visual motion," in *Active Vision*, MIT Press, 1992, pp. 263-284.

[26] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *ICCV*, 2001.

[27] D. G. Lowe, "Object Recognition from Local Scale-invariant Features," in *ICCV*, Kerkyra, Greece, 1999.

[28] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, no. 27, pp. 1615-1630, 2005.

[29] Y. Ke and R. Sukthankar, "PCA-SIFT: "A More Distinctive Representation for Local Image Descriptors"," in *Proc. Conf. Computer Vision and Pattern*, 2004.

[30] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding,* vol. 110, pp. 346-359, 2008.

[31] C. Liu, J. Yuen and A. Torralba, "SIFT Flow: Dense Correspondence across Scenes and its Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, no. 5, 2011.

[32] T. S. Huang, G. J. Yang and G. Y. Tang, "A Fast Two-Dimensional Median Filtering Algorithm," *IEEE Transactions on Acoustics, Speech, And Signal Processing,* Vols. ASSP-27, no. 1, 1979.

[33] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *International Conference on Computer Vision*, Bombay, India, 1998.

[34] J. J. Koenderink, "The structure of images," *Biological Cybernetics,* vol. 50, pp. 363-396, 1984.

[35] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at differenct scales," *Journal of Applied Statistics,* vol. 21, no. 2, pp. 224-270, 1994.

[36] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, January 2004.

[37] M. Brown and D. G. Lowe, "Invariant features from interest point groups," in *British Machine Vision Conference*, Carfidd, Wales, 2002.

[38] T. Lindeberg, "Feature detection with automatic scale selection," *IJCV,* vol. 30, no. 2, pp. 79-116, 1998.

[39] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," 2006.

[40] Y. Boykov, O. Veksler and R. Zabih, "Fast Approximate Energy Minimization via

Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 11, pp. 1222-1239, 2001.

[41] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin and M. Cohen, "Interactive Digital Photomontage," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 294-302, 2004.

[42] C. Rother, V. Kolmogorov and A. Blake, "'GrabCut'-Interactive Forground Extraction Using Iterated Graph Cuts," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 309-314, 2004.

[43] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Transactions On Pattern Analysis And Machine Intelligence,* vol. 30, no. 6, June 2008.

[44] R. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision,* vol. 70, no. 1, pp. 41-54, 2006.

[45] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. Royal Statistical Soc., Series B,* vol. 48, no. 3, pp. 259-302, 1986.

[46] R. Szeliski, "Image Alignment and Stitching," in *Handbook of Mathemetical Models in Computer Vision*, New York, Springer Science+Business Media, 2006, pp. 273-292.

[47] A. Bruhn and J. Weickert, "Lucas/Kanade Meets Horn/Schnuck: Combining Local and Global Optic Flow Methods," *International Journal of Computer Vision,* vol. 61, no. 3, pp. 211-231, 2005.

[48] G. L. Barrows, J. S. Chahl and M. V. Srinivasan, "Biologically inspired visual sensing and flight control," *Aeronautical Journal,* vol. 107, no. 1069, pp. 159-168, 2003.

[49] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black and R. Szeliski, "A database and evaluation methodology for optical flow," in *IEEE International Conference on Computer Vision*, 2007.

[50] [Online]. Available: http://nokia-ph-gs.com/wherescritter/ap-pelvis-radiograph.

[51] "Motion perception - Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Motion_perception.

[52] T. Brox, A. Bruhn, N. Papenberg and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *ECCV*, Berlin, Germany, 2004.

[53] A. Shekhovtsov, I. Kovtun and V. Hlavac, "Efficient MRF deformation model for non-rigid image matching," *Computer Vision and Image Understanding,* vol. 112, no. 1, pp. 91-99, 2008.

[54] Y. Weiss, "Correctness of Local Probability Propagation in Graphical Models with Loops," *Neural Computation,* vol. 12, no. 1, pp. 1-41, 2000.

[55] J. M. Mooij and H. J. Kappen, "Sufficient Conditions for Convergence of the Sum–Product Algorithm," *IEEE Transactions on Information Theory* , vol. 53, no. 12,

pp. 4422-4437, 2007.

[56] B. C. Russell and A. Torralba, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision,* vol. 77, pp. 157-173, 2008.

[57] A. Oliva and A. Torralba, "Modeling the shape of scene: a holistic rerepsentation of the spatial evelope," *International Journal of Computer Vision,* vol. 42, no. 3, pp. 145-175, 2001.

[58] A. Torralba, K. P. Murphy, W. T. Freeman and M. A. Rubin, "Context-based vision system for place and object recognition," in *IEEE International Conference on Computer Vision*, 2003.

[59] S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition*, 2006, 2006.

[60] J. Xiao, J. Hays, K. Ehinger, A. Oliva and A. Torralba, "SUN database: large-scale scene recognition from abbey to zoo," in *Computer Vision and Pattern Recognition*, 2010.

[61] J. Shotton, J. Winn, C. Rother and A. Criminisi, "Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision,* vol. 81, no. 1, pp. 2-23, 2009.

[62] C. Liu, "Nonparametric Scene Parsing via Label Transfer Data & Matlab/C++ Code," 2011. [Online]. Available: http://people.csail.mit.edu/celiu/LabelTransfer/code.html.

[63] H. Prasantha, S. H.L., K. Murthy and M. Lata.G, "Medical Image Segmentation," *International Journal on Computer Science and Engineering,* vol. 2, no. 4, pp. 1209-1218, 2010.

[64] D. Holmes III, M. Rettmann and R. Robb, "Visualization in Image-Guided Interventions," in *Image-guided interventions: technology and applications*, New York, Springer, 2008, pp. 45-80.

[65] J. Inoue, "Computer-Aided Surgical Planning," 2008.

[66] S. T. Acton and N. Ray, Biomedical Image Analysis: Segmentation, Austin, TX: Morgan & Claypool, 2009.

[67] M. Brown and D. Lowe, "Recognizing panoramas," in *ICCV*, Nice, France, 2003.

[68] D. Gossow, P. Decker and D. Paulus, "An evaluation of open source SURF implementations," in *RoboCup 2010: robot soccer world cup XIV*, Berlin, Heidelberg, Springer-Verlag, 2011, pp. 169-179.