2

Incremental Engineering of Lung Segmentation Systems

Avishkar Misra, Arcot Sowmya, and Paul Compton

CONTENTS

In recent decades, medical imaging has become an increasingly invaluable resource for physicians by providing them with a noninvasive view inside the patient, allowing them to diagnose diseases (Doi 2005), plan for surgical interventions (Archip et al. 2006), and evaluate the effectiveness of treatments (Pien et al. 2005). More than 95 million high-tech scans, including computed tomography (CT), magnetic resonance imaging, and positron emission tomography, are conducted in the United States each year (Kolata 2009). Automatic interpretation of medical images by computer-aided detection or diagnosis (CAD) systems have the potential not only to significantly reduce the burden on physicians to interpret a large number of medical scans in a timely manner, but also to improve their consistency in diagnosis across the patients (Doi 2005). A fundamental component of any CAD system is medical image segmentation (Reinhardt et al. 2000; Sluimer et al. 2006).

OK to change to "across patient populations" instead?

The design and development of medical image segmentation systems is an extremely challenging task. A reasonably complex segmentation system has a number of components or processes intelligently combined to achieve the system goals. Computer vision experts, who develop these systems, must selectively apply various vision techniques and articulate the knowledge necessary to guide the image segmentation and interpretation tasks. In doing this, the expert articulates domain and control knowledge (Draper et al. 1996; Crevier and Lepage 1997; Clouard et al. 1999; Wangehheim et al. 2000). Image segmentation systems that segment lung regions in high-resolution CT (HRCT) scans require control knowledge about how to combine and apply the various computer vision algorithms appropriately, as well as knowledge about the medical domain itself, such as the anatomy or the appearance of anatomical structures under x-rays. The knowledge that experts attempt to encode in systems must address the complexities and intricacies of the domain.

Experts usually develop lung segmentation systems guided by their understanding of the segmentation techniques, the domain, as well as their intuitions. This naturally means that the quality of expertise available to develop these systems governs the quality of the resulting system. Considering that "the mental operations that we perform to interpret images lie almost totally beyond the threshold of consciousness" (Crevier and Lepage 1997), it is not surprising that experts find it so difficult to articulate the knowledge required. Vision experts therefore tend to be quasi experts, in the sense that they have a general understanding of how to interpret an image, but lack the specific knowledge of how the selection of algorithms and their parameters affect the interpretation of the image. Although pattern recognition and machine learning methods mitigate the influences of quasi-expertise by automatically deriving knowledge from the domain data (Duda et al. 2000), they do require a significant amount of labeled data. This would require medical experts to mark accurate ground truth for a large number of training instances, which can be difficult because of the demands on the expert's time. Even though some methods such as graph cut (Boykov and Kolmogorov 2004; Massoptier et al. 2009) eliminate the need for labeled data, they do require experts to design cost functions appropriate to the task and to tune their parameters. One can argue that defining a cost function is a skilled task, requiring significant domain and vision expertise, and this leaves systems yet again vulnerable to the influence of quasi-expertise. As Draper (2003) argues, vision researchers are "once again applying informal control policies under the guise of a theoretically sound system. Feature quality determines the quality of the image interpretation, but the features are selected heuristically."

In practice, experts discover the features appropriate for a specific segmentation task by trial-and-error, revising the features and tuning the parameters of the algorithms in a relatively ad hoc manner. The ad hoc incremental approach to engineering systems naturally introduces the risk in which changes made to correct errors for certain cases may unwittingly lead to errors for other cases that were correctly handled earlier. Furthermore, changes to one part of the system may adversely affect the performance of other parts of the system. The risk of system degradation because of well-intended changes grows in line with the size and complexity of the system. Yet, it is common practice for experts to tweak algorithms and tune the parameters of algorithms incrementally.

This chapter presents a framework called ProcessNet that helps experts to incrementally engineer a lung HRCT image segmentation system. ProcessNet allows experts to systematically revise existing algorithms, tune their parameters, or add newly developed algorithms in an image segmentation system. The framework enables experts to incrementally develop the system in a structured manner despite evolving expertise and incrementally available data. The incremental revisions also provide an opportunity for continual system improvement and adaptation for specific segmentation tasks. An expert developing a system using ProcessNet may continue to improve its components over time, even as it goes into production use.

A lung anatomy segmentation system built using ProcessNet demonstrates continual improvement in the segmentation of lung, spine, sternum, and shoulder blades, visible within a sparse HRCT study for a patient. Although the primary goal is the accurate segmentation of lung regions, the segmentation of other anatomical structures provides useful anatomical cues and partial registration to facilitate lung segmentation. Each of these anatomical structures must be segmented to sufficiently accurate levels and face their own segmentation challenges. ProcessNet mitigates the risks of incremental ad hoc revisions by the expert within the complex anatomy segmentation system.

This chapter is organized as follows. A background to some of the methods used in medical image segmentation systems along with the challenges they face appears in Section 2.1. A formulation of medical image systems as a network of processes is presented in Section 2.2. This forms the basis for the ProcessNet framework introduced in Section 2.3. A lung anatomy segmentation system that is incrementally developed using ProcessNet is introduced in Section 2.4. A discussion in Section 2.5 highlights the benefits and limitations of the ProcessNet system, followed by a summation in Section 2.6.

2.1 Background

Medical image segmentation systems attempt to partition an image into regions representing specific tissue or anatomical structures (Bankman 2009). To achieve these goals, several computer vision techniques have been applied to a range of medical imaging modalities. Regular reviews of the emerging challenges and techniques used in medical image analysis for various modalities have been conducted (Deklerck et al. 1993; Duncan and Ayache 2000; Frangi et al. 2001; Ginneken et al. 2001; Noble and Boukerroui 2006; Sluimer et al. 2006; Bankman 2009). The following sections present some of the approaches used to segment medical images and the challenges to accurately segment lungs in HRCT studies.

2.1.1 Approaches to Medical Image Segmentation

Over the years, experts have developed a plethora of methods to automatically segment medical images. Some of the common approaches to medical image segmentation include (Sluimer et al. 2006; Bankman 2009) classical image analysis, knowledge-based or syntactic techniques, deformable model-fitting, classification-based techniques, and atlas-based segmentation via registration.

Each of the approaches is described briefly and the challenges that experts face in using these techniques are highlighted. Although we focus on the segmentation of lungs in HRCT scans, relevant work highlighting novel techniques and applications in ultrasound and magnetic resonance imaging segmentation are also discussed. None of the applications, however, have thus far addressed the challenges of evolving quasi-expertise and labeled data, which are both encountered in developing medical imaging systems.

2.1.1.1 Classical Image Analysis

Classical image analysis techniques use standard image processing algorithms to segment an image into separate regions. Rogowska (2009) provides a good introduction to some of the fundamental imaging techniques used by medical image segmentation systems. These techniques use a variety of image processing algorithms such as thresholding, outlining, edge detection, morphological operators, and filters, with the objective of detecting regions or edges that define the boundaries between regions of anatomical structures (Ballard and Brown 1982). In almost all cases, a number of image processing algorithms are combined intelligently to develop a system that is capable of automatically segmenting the desired anatomy. Applications of classical image analysis techniques to the segmentation of CT and HRCT images have been reported (Ballard and Sklansky 1976; Giger et al. 1990; Duryea and Boone 1995; Armato et al. 1998; Hu et al. 2001; Zheng et al. 2003; Ukil and Reinhardt 2004; Zhou et al. 2006).

The line "Almost all adopted" was changed to "Almost all authors adopted" ""- please verify if this is correct.

Almost all authors adapted similar approaches to segmenting anatomy within their specific imaging modalities and segmentation objectives. In the segmentation of the lung in HRCT images, techniques take advantage of the contrast between the air-filled lungs and the surrounding tissues. In normal circumstances, a fixed threshold value of –1000 Hounsfield units would provide a segmentation of regions filled with air. A single global threshold value, however, often does not lead to good separation between tissues and, therefore, locally adaptive thresholding methods are often used. Adaptive thresholding and region-growing have been used (Zheng et al. 2003) and a three-dimensional (3D) watershed algorithm to segment lung regions in CT images has been used (Kuhnigk et al. 2003). Despite best efforts, thresholding may result in noise or undesirable connection between tissues, especially because of the presence of disease or imaging artifacts. Therefore, morphological operators such as erosion and dilation are used to eliminate noise and improve segmentation. Spatial and frequency domain filtering is also used to eliminate noise and segment anatomical regions (Malik and Choi 2006).

Possibly change to "the techniques used" instead?

Although most approaches attempt to segment a specific part of the anatomy, Zhou et al. (2006) segment multiple anatomical structures simultaneously. Their segmentation algorithm takes advantage of dense multidetector CT (MDCT) studies, in which more than 400 images provide better continuity between the slices, as compared with sparse HRCT studies with only 20 to 30 images. The proximity between the slices within MDCT studies makes it easier for algorithms to take advantage of continuity between images.

These techniques, however, are sensitive to noise, and because of the brittle nature of the algorithms, are seldom used in isolation. Instead, classical image analysis techniques are used in conjunction with other techniques, usually in preprocessing and segmentation stages. Classical image analysis techniques encode the required domain and control knowledge implicitly within the algorithms and the parameter values selected to control those algorithms. Experts develop classical image analysis–based systems by selecting intelligent combinations of the algorithms and their parameters.

2.1.1.2 Knowledge-Based or Syntactic Techniques

Knowledge-based or syntactic techniques represent knowledge in an explicit form independent of the image processing algorithms. The representations include rules (Matsuyama

1988; Levine and Nazif 1985), frames (McKeown et al. 1985), or semantic networks representing relationships (Freuder 1977; Paulus et al. 2000) and domain ontology (Maillot et al. 2004) between parts of the anatomy. By expressing characteristics about the anatomical structures and relationships between the structures, the vision system can gather support and reason about thus far unidentified objects within the scene. Rules also allow the expression of knowledge applicable under specific conditions, making it easier for experts to articulate the context-based applicability of a particular algorithm, parameter, or classification. Because these systems are closely related to expert systems for vision, they are commonly referred to as expert- or knowledge-based techniques.

Some examples of such systems include ANGY, a rule-based system to interpret angiograms (Stansfield 1986); IBIS, a rule and semantic network–based system to interpret magnetic resonance imaging (Vernazza et al. 1987); a system for 3D reconstruction of angiograms (Declaere et al. 1991); VISIPLAN, a hierarchical planning framework (Gong and Kulikowski 1994); a chest x-ray (radiography) segmentation system (Brown et al. 1998); a rule-based system to detect spinal cord in CT (Archip et al. 2002); and a multiagent system to segment intravascular ultrasound images (Bovenkamp et al. 2004).

The symbolic representations of objects within a semantic network offer a concise representation. However, as the complexity of the domain grows, describing and maintaining the relationships between the growing number of components becomes a challenge. The authors of the schema system (Draper et al. 1989) note that it took them over a month to develop and revise a relatively small semantic network. There may also be multiple relationships between the objects, according to different contexts. Bovenkamp et al. (2004) state that more than 450 rules were added to guide the image segmentation, yet no strategy was identified to ensure that the consistency and accuracy of the rules was maintained.

2.1.1.3 Deformable Model Fitting

Deformable model-fitting techniques represent expectations about an object being sought in a model, that is, fitted directly to the image data (McInerney and Terzopoulos 2009). In contrast with rigid pattern or template-matching techniques (Ballard and Brown 1982; Grimson 1990), deformable models allow the model to be deformed to fit the evidence within the images, albeit within certain limits. These techniques fit the model to the image data via optimization methods using energy or cost functions that evaluate the quality of the fit. The energy or cost function is often a composite of multiple terms, evaluating different aspects of the fit such as structural integrity of the boundary, the likely shape of the desired object, and the edges or regions in the image most likely to represent the object.

A variety of deformable models have been developed and applied to medical image segmentation, each of which attempts to find boundaries of regions or regions of homogeneity. These include active contours (Kass et al. 1988; McInerney and Terzopoulos 1996), active shape models (Cootes et al. 1995), active appearance models (Cootes et al. 2001), and level sets (Osher and Sethian 1988; Lin et al. 2004; Cremers et al. 2007).

He et al. (2008) provide a comparative study of various forms of deformable contour methods for medical image segmentation. Heimann and Meinzer (2009) provide a review of statistical shape models for 3D medical image segmentation. Lin et al. (2004) have incorporated region and boundary conditions in level sets.

The challenge in developing a deformable model-based solution is the high level of vision expertise required in defining the energy function and selecting its parameters. In addition, some techniques such as active contours must be initialized close to the target anatomical region; therefore, this requires either manual intervention or classical image analysis techniques to gain an initial estimate to automatically initialize the contours. Meanwhile, region growing or wave propagation techniques such as level sets (Osher and Sethian 1988) or T-snakes (McInerney and Terzopoulos 2000) may spill over the intended boundary in the absence of a sufficiently strong boundary condition, which therefore must be balanced by other terms within the energy function, such as shape information. Successful segmentation via deformable models rely on the weights for the terms within the energy function, which is a complex control problem understood poorly by the experts (Ozertem and Erdogmus 2007; He et al. 2008). Therefore, the definition of the energy function and the selection of its weights requires significant development effort via trial and error.

2.1.1.4 Classification-Based Techniques

Classification-based techniques attempt to label individual pixels or regions segmented using classical image analysis techniques with the labels of the anatomical structure of interest. These techniques derive numeric (continuous) or nominal (categorical) features from the pixel, a region, or the entire image, and use a classifier to predict the membership of specific classes. The classifier is developed using a variety of pattern recognition and machine learning techniques (Mitchell 1997; Duda et al. 2000; Jain et al. 2000). Factors such as similarity measures (e.g., clustering; Sutton 2010) or *k*-nearest neighbor (Jain et al. 1999), probabilistic support (e.g., Bayesian inference (MacKay 2003, p. 457), decision boundary (e.g., decision trees; Murthy 1998), SVM (Drucker et al. 1996), or neural networks (Wismueller 2010) are used by the classifier to perform class predictions.

Please spell out SVM here.

McNitt-Gray et al. (1995) classified pixels into regions of the heart, lung, and axilla in chest radiographs. Zhang and Valentino (2001) used artificial neural networks to classify pixels within CT scans. Wei (2002) extracted frequency domain features from the image and used *k*-nearest neighbor clustering for segmentation. Ghosh and Mitchell (2006) have used genetic algorithms to evolve the level set function to segment the prostrate in CT images. Sahba et al. (2007) have used reinforcement learning in the segmentation of ultrasound images. As mentioned previously, pattern recognition and machine learning techniques require experts to devise useful features and gather sufficient labeled data for training the classifier.

OK to change to "gather a sufficient amount of labeled data" instead?

Recently, graph cuts (Boykov and Jolly 2001) were used in the segmentation of lung in HRCT (Massoptier et al. 2009). These techniques classify each pixel within the image as either belonging to the foreground representing the desired object or the background. These techniques have demonstrated good segmentation results without requiring large amounts of labeled data to induce the classifier. The expert constructs cost functions that consider the membership of a pixel to the foreground or background, whereas also taking into account the membership of its neighboring pixels. This involves considerable involvement from vision experts in defining and tuning a complex cost function, similar to the definition of the energy function used by deformable model-fitting techniques.

2.1.1.5 Atlas-Based Segmentation via Registration

Atlas-based segmentation techniques, also known as segmentation via registration, attempt to segment a new image by drawing a correspondence between the new image and a previously segmented atlas (Maintz and Viergever 1998). The atlas is often the representative or average of the variations in the anatomy. Once a correspondence between the pixels of the image and the atlas has been established, the labels from each pixel within the atlas can be applied to pixels in the image. The matching process is framed as an optimization process, looking to improve the level of similarity between corresponding pixels, whereas maintaining relational consistency between the aligned pixels.

Atlas-based segmentation has been used in segmenting magnetic resonance images of the brain and heart (Kikinis et al. 1996; Cuadra et al. 2003; Lorenzo-Valdes et al. 2002) as well as CT of the abdomen (Park et al. 2000) and lungs (Zhang and Reinhardt 2000; Li et al. 2002; Zhang et al. 2003; Sluimer et al. 2004). Sluimer et al. (2004) used classifier techniques to improve the quality of segmentation at the boundaries between regions, despite the presence of diseases.

Atlas-based segmentation techniques have the advantage of being robust in the presence of disease (Sluimer et al. 2004). However, to construct a statistically significant atlas, considerable effort is required in labeling pixels from a large number of training images, as well as aligning and deforming these in the process of generating the atlas.

2.1.2 Challenges to Segmenting Lungs in HRCT

In addition to the developmental challenges of quasi-expertise and incrementally available data, image segmentation systems face challenges specific to their domain. These challenges vary according to the nature of the anatomy and the imaging modality used. Interpretation of HRCT images, particularly in the presence of diffuse lung diseases (Webb 1990), faces a number of challenges that medical image segmentation systems must also contend with (Sluimer et al. 2006). These can be categorized as (1) interpatient and intrapatient variations, (2) real and artificial artifacts, and (3) competing definitions of ground truth. Each of these challenges is briefly discussed.

2.1.2.1 Interpatient and Intrapatient Variations

There is a significant amount of natural variation in the expected anatomy, both within a patient and across patients. For example, the shape of the lung within the apical region (roughly top third) is circular, whereas the shape of the lung within the basal region (roughly bottom third) has a sickle-like shape. Therefore, a shape model or template to identify a lung must allow for a significant degree of variation across the images. Similarly, the lungs of a 68-year-old man would vary significantly in size, when compared with those of a 16-year-old girl.

Possibly change to “across patient groups” instead?

2.1.2.2 Real and Artificial Artifacts

A range of real and artificial artifacts can affect the appearance of objects in the images. Artificial artifacts are those that are introduced in the process of acquiring the scans, and may not appear if scanned differently or at another time. Meanwhile, real artifacts are those that exist within the body and influence the appearance of the anatomy, and cannot be eliminated by changing the scanning protocol.

An example of an artificial artifact is the movement of a beating heart, which may appear as blurring of the boundaries, as shown in Figure 2.1a. CAD systems could mistake the high opacity of the motion artifact as abnormal lung tissue indicative of consolidation. Another example of an artificial artifact is the change in the density of the lung tissue because of gravity. This would show up in different parts of the lung, depending

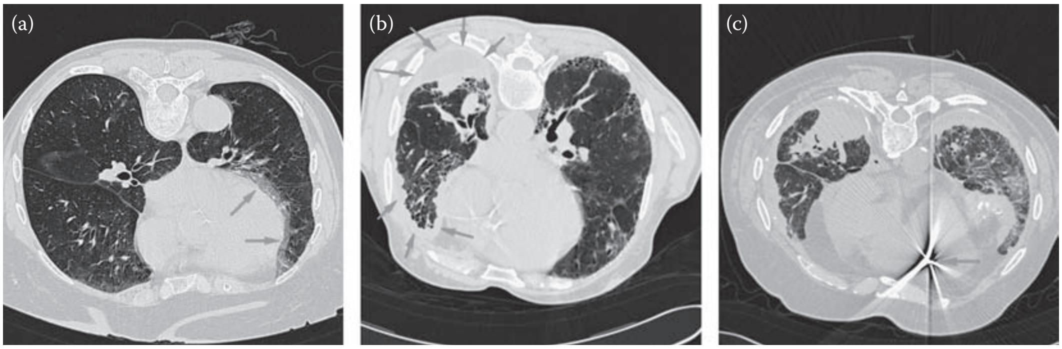

FIGURE 2.1
Three examples of real and artificial artifacts: (a) motion artifact because of a beating heart, (b) pleural plaques deforming lung tissue, and (c) metallic staples in the heart.

on whether the patient is lying on their stomach (i.e., prone) or back (i.e., supine) during the scan.

Similarly, real artifacts also complicate the task of automatic segmentation. Because medical scans are generally taken to facilitate diagnosis for a patient presenting with medical symptoms, the patient is more likely to have a disease of some kind. In many cases, the presence of disease may influence not just the anatomy or tissue that it affects, but also the adjacent structures, by deforming a structure's shape and position. A patient with severe pleural plaques in the right side of the lung is shown in Figure 2.1b. The degradation of the pleural tissue of the lung gives it the appearance of muscle tissue outside the lung. Lung boundary segmentation may incorrectly segment the boundary at the edges of the darker air-filled regions, instead of the correct lung boundary marked by the blue arrows.

Another form of real artifact occasionally observed in scans is caused by the presence of implants, such as that shown in Figure 2.1c. The bright streaking effects, because of the metal staples embedded within the patient's heart, makes it difficult to segment the sternum, which may serve as an anatomical landmark for subsequent segmentation of other parts. The streaking effect also influences the appearance of the lung tissue.

2.1.2.3 Conflicting Ground Truth Definitions

Although ideally, one expects that all radiologists have the same notion of truth in defining anatomy, there are often differing perspectives among them, with disagreement on the delineation of an anatomical structure's boundary. For example, a radiologist may delineate the lung boundary by considering the region with the lung tissue as indicated by the green line in Figure 2.2a. Another radiologist may consider the bronchial tree at the hilum to be a part of the lung as shown by the blue line in Figure 2.2b.

Similarly, radiologists may disagree on the categorization of certain disease-affected tissue, and the ensuing differential diagnosis. This is particularly common when a patient has multiple pathologies or diseases and requires arbitration from more experienced radiologists. HRCT for diffuse lung diseases also tends to have a high level of variations in interobserver and intraobserver interpretations (Aziz et al. 2004). This is mainly because of the lack of standard criteria.

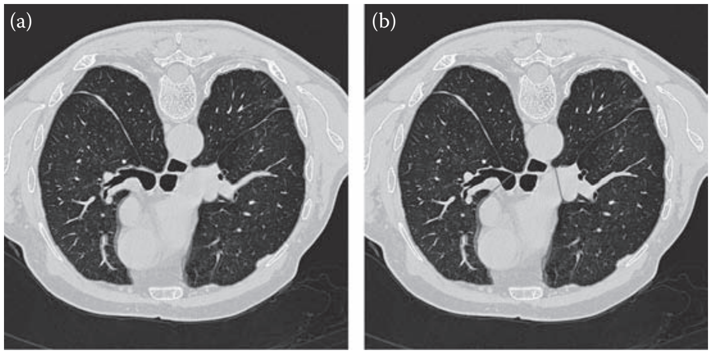

FIGURE 2.2
Differences in lung boundary delineation by experts: (a) marking by expert A and (b) marking by expert B.

2.2 ProcessNet as a Network of Processes

A lung HRCT image segmentation system can be formulated as a network of processes, wherein each process is responsible for a specific task. Each process produces an output and consumes the outputs from other processes as its input. All of the processes, in undertaking their specific task, collectively solve the overall system goals. The system can be represented as a directed graph in which each process is a node and a connection between them depicts the information flow between the processes. Note that only directed acyclic graphs are considered, so the system will have no circular dependencies between processes. A sorted topology of the system dependency graph provides the order in which the processes must be executed, such that data requirements for each process are satisfied. This sequence is called the process run order, which imposes no restriction on the parallel execution of processes and merely requires that if process B depends on the output of process A, then process B must only be executed after process A has produced the required data.

The algorithms within each process express domain and control knowledge to guide the various tasks. Over time, the expert might make changes to individual processes to improve the accuracy with which a process undertakes its task. Alternatively, the expert might introduce new processes to add new functionality to the system. The nature of a process and the types of changes that may affect the processes within the system are now discussed.

2.2.1 Internals of a Process

A process represents an arbitrarily complex algorithmic step. It may implement a simple image thresholding algorithm or apply the more sophisticated deformable models. Internally, a process may call on a machine learning algorithm or a decision tree to guide the selection of an algorithm, parameter values for the algorithm, or to classify a case. Such knowledge-based processes have the capacity to intelligently guide the processing appropriate to the circumstances of the input. Alternatively, a process may be a single processing algorithm, with no dynamic context-specific control.

In both scenarios, the process captures and applies the knowledge necessary to guide its specific task and contribute to the overall segmentation goals of the system. Considering

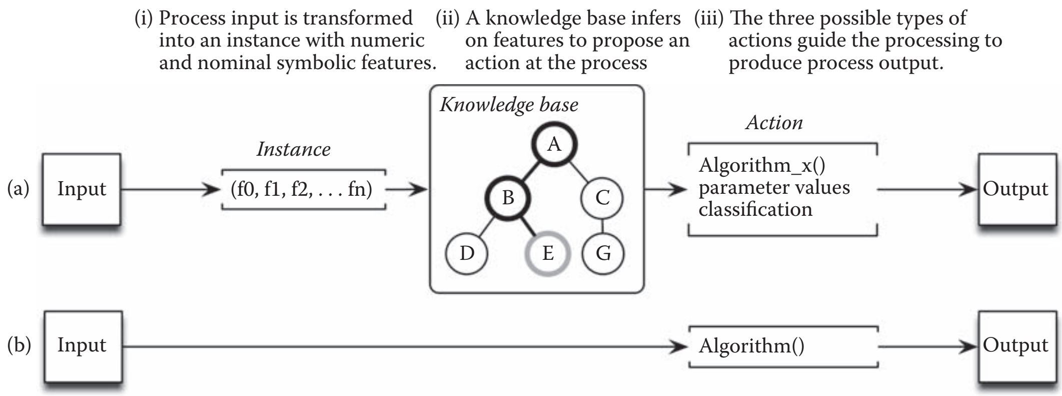

FIGURE 2.3
Internal processing steps for (a) a knowledge-based process and (b) a fixed algorithm.

the evolving nature of knowledge when developing such systems, it is reasonable to expect that even the single processing algorithm may undergo revisions, as problems or limits with the algorithm are detected.

The core functions of (a) a typical knowledge-based process and (b) a fixed algorithm process are shown in Figure 2.3. The knowledge-based process carries out three basic tasks:

i. Input data is transformed into features and represented as an inferable instance or case.
ii. The knowledge base infers on the instance or case to predict the action. The action may be the selection of an algorithm sequence, the parameters for an algorithm, or the class label to assign to the case.
iii. The actual processing as dictated by the action, generating the process output.

The internal functions of a machine learning–based process are similar to that of a knowledge-based process, with the inference system using a decision-making function defined by the machine learning algorithm used. The machine learning algorithm would also attempt to predict the action of the process, based on the features derived from the input data.

A process with a handcrafted vision algorithm may operate directly on the input data without deriving any features or inferring the correct algorithm, parameter, or label to assign. In such a process, the algorithm explicitly encodes the knowledge required to consume the process input and produce the output.

2.2.2 Changes Affecting a Process

The correct behavior of a process may be influenced by changes both external and internal to the process because of incremental revisions to the system. The four types of changes that may adversely affect process capacity to correctly carry out its task are changes to

i. Raw input. Input data for a process may change if the output of other processes that serves as input to this process, changes. Alternatively, an expert may introduce new data sources, or remove redundant ones, that are no longer deemed to be useful for the process.

ii. Feature extraction. The derivation and expression of features that are used by an inference system may be revised, as the expressive limits of existing features are discovered. Experts often invent new features or project current features into new feature spaces, leading to changes in the derivation and expression of features.

iii. Control or inference knowledge. The decision-making mechanism that determines the actions of the process may change. These actions might be the selection of an algorithm, parameters to use, or the classification label. The inference system may be constructed using multiple machine learning techniques that guide the processing using different features. Improvements to the inference system may occur in the light of more labeled data becoming available for training, or revisions to the learning algorithms and its parameters. Examples include the induction of a decision tree from available data, and tuning of its pruning parameters.

iv. Algorithm, parameter mappings, and class labels. The actual processing algorithms, parameter mappings, and labels, which are used as actions of the process, may change. These algorithms may be improved independently of the control knowledge within the inference system.

A process representing a fixed algorithm may face changes to its input (i) and the algorithm itself (iv), whereas a knowledge-based vision process may encounter changes at any of the four levels. A method to help experts to manage these changes is presented in the following section.

2.3 ProcessNet Framework

The ProcessNet framework applies the concepts of incremental validated change management that have been successfully used by ripple-down rules (RDR) in addressing incremental ad hoc revision to knowledge in a variety of domains (Compton and Jansen 1988). Over the past two decades, RDRs have been successfully applied to a range of complex classification, control, and search problems. Richards (2009) presents an excellent review of RDR research during the last two decades. Although details about RDR are presented in Section 2.3.1, the key inspiration drawn from RDR is to ensure that any change made to knowledge is allowed only if the knowledge is maintained consistent to the evidence that has led to the knowledge thus far. Grounding of knowledge on evidence is fundamental to the process of scientific discovery (Popper 1963). In RDR, the knowledge is maintained within rules and the evidence is maintained in the form of cornerstone cases that act as support for the knowledge within the rule. Any changes to the rules are evaluated against the cornerstone cases of the rules potentially affected by the revision.

The ProcessNet framework generalizes the incremental case-based revision of RDR knowledge bases to system engineering in general. In ProcessNet, a segmentation system is represented as an acyclic network of processes, each of which may represent its internal knowledge in a form independent of other processes. The processes apply their knowledge and communicate the results to other processes. The knowledge within a ProcessNet processes may be represented within any arbitrary form. The ProcessNet framework offers strategies for

i. Detecting changes and evaluating their effect on the processes within the ProcessNet, discussed in Section 2.3.2.
ii. Managing the impact of changes on processes within the ProcessNet, discussed in Section 2.3.3.

These two central tasks are critical for systematic incremental engineering of medical image segmentation systems, and form the change management mechanisms of ProcessNet. An example of validation of a ProcessNet system is demonstrated in Section 2.3.3. Details about an implementation of the ProcessNet framework used to build medical image segmentation system are presented in Section 2.3.4.

2.3.1 Ripple-Down Rules

RDRs are an incremental knowledge acquisition technique that recognizes the inability of experts to correctly and completely articulate the knowledge necessary when constructing knowledge- or expert-based systems (Compton and Jansen 1989). These are the same challenges encountered by quasi experts in vision domains, as they attempt to engineer medical image segmentation systems in light of evolving expertise and incrementally available data. RDRs provide decision-making support by classifying a given instance (or case) according to its characteristics or features. A case is defined in terms of the features that serve as input to the RDR and the predicted conclusion.

RDRs maintain knowledge in a nested hierarchy of rules and their exception rules. Each rule within the RDR knowledge base defines the context or conditions, which must be met in order for the rule to be applicable for a given case. A rule may have exception rules that override its conclusion, provided the context of the exception rules is also met by the case. Therefore, the inference process begins at a root rule and progresses down the rules and their exceptions until the last firing rule is reached. The conclusion of the last firing rule defines the conclusion for the given case. This conclusion represents a classification for a task given the context of the case interpreted by the knowledge base.

Recognizing the fact that knowledge within a knowledge base may not be complete, RDRs provide an easy way for experts to correct and update the incorrect knowledge. If the concluding RDR rule is incorrect and misclassifies a given case, the expert can correct the RDR knowledge base by adding a new exception rule to the misclassifying rule. In constructing the new rule, the expert merely has to justify his or her conclusion by defining the context of the new rule, such that it correctly applies to or "covers" the new case. A training event is initiated whenever a case or instance is incorrectly classified, prompting the expert to provide an exception rule to correct this behavior.

To ensure that a change to the knowledge base does not degrade existing knowledge, RDRs use validated change criteria that must be satisfied before the addition of a new exception rule. Each rule in an RDR knowledge base retains cornerstone cases, which are exemplars used to construct or modify the rule. A new exception rule must cover the training case, whereas not changing the conclusion of cornerstones of the parent and sibling rules within the neighborhood. The nested hierarchy of rules and their exceptions means that the expert has to consider only a small subset of cornerstone cases when attempting a revision to an existing knowledge base.

RDR knowledge bases are built incrementally, with each new rule added to resolve an error as it is discovered with new cases. Unlike batch mode learning systems, there is no distinction between the knowledge construction phase and knowledge application phase. They are equally part of the continuum of knowledge maintenance, where knowledge is used and corrected as and when issues are discovered. This makes RDR particularly effective in successfully operating in domains where data becomes available incrementally.

RDR are able to elicit even tacit knowledge that may not be consciously recognized by the domain experts, but used subconsciously in the application of their expertise to address complex tasks within the domain. The ease with which RDR facilitates changes to the knowledge base means that experts can quickly build large knowledge bases that eventually converge to describe the knowledge required to undertake the task. RDR have demonstratively addressed the knowledge acquisition bottleneck faced by expert-engineered knowledge-based systems in a number of applications including the interpretation of pathology results (Compton and Jansen 1988; Compton et al. 2006), VLSI design (Bekmann and Hoffmann 2005), network security (Prayote 2007), and image analysis (Amin et al. 1996; Kerr and Compton 2003; Misra et al. 2004, 2006; Singh and Compton 2005; Park et al. 2008).

Please spell out VLSI here.

The cornerstone case-based validation of changes does not eliminate all risks of degradation of the knowledge base, and the consistency of the knowledge is contingent on the quality of the cornerstone cases used to cover the possible variations within the domain. In practice, however, it offers experts a reasonable means to evaluate the quality of their revisions. Over time, as sufficient cases are evaluated by RDR, limitations in knowledge are highlighted and provide experts with specific exemplars to assist in their efforts to revise the knowledge base.

2.3.2 Detecting Process Change via Cornerstone Shift

In keeping with RDR, a case is defined for each process in terms of its raw input data and the resulting output. This ensures that a process only considers the minimal set of data sources necessary to undertake its task. The input and output data at a process capture the process-specific behavioral requirements. Like RDR, any changes made to a process should be motivated by a case, and once the change is made, the case is captured as a cornerstone case for the process. Each process is considered a knowledge source, the consistency of which is evaluated against the set of cornerstone cases that influenced its construction.

The effect of the four types of changes on process consistency can be detected by evaluating the cornerstone cases for the process. Any difference to the output and input data for a cornerstone case highlights a potential affect as a consequence of changes at either the process itself or at other processes that this process depends on for its input data, respectively. These differences are called cornerstone shifts.

A cornerstone case for a process captures an example describing the requirements for that given process, but may not capture the implicit contract of expectation that other dependent processes have with the process. Therefore, although the correctness of a process can be validated against its own cornerstone cases, the responsibility of validating its effect on dependent processes lies with the latter. Each of the dependent processes should also be evaluated for changes to their input and subsequent output, by checking against their own set of cornerstone cases. This is because the original process may not have insight into how its dependent processes use its output.

This approach frees the expert from having to consider the whole system when revising a single process. An expert improves the correctness of a specific process with respect to its cornerstone cases, with the expectation that other dependent processes take responsibility for detecting and handling the cornerstone shifts relevant to them, which is discussed in the following section. The cornerstone validation process starts at the first process that is modified and continues in the process run order. The requirement of ProcessNet to allow only acyclic dependencies ensures that each process needs to be evaluated only once during a single validation check.

2.3.3 Managing Cornerstone Shifts

As discussed previously, cornerstone shifts are any deviation in the input and output data of a cornerstone case because of changes within the system. The key difference between RDR and ProcessNet is the handling of cornerstone shifts. In RDR, only changes to the conclusions of a cornerstone case are allowed. This means that only process outputs may change, whereas the inputs to the process, and the features used to infer against the knowledge base, do not change. In vision domains, however, even the inputs to a process may change. Therefore, ProcessNet allows changes to any aspect of the cornerstone case, as long as the resulting output of the cornerstone case agrees with ground truth for that case.

Furthermore, the notion of truth and the nature of data within the vision domain may change over time with evolving domain insight. This means that even the ground truth for a cornerstone case may change. To accommodate for this, ProcessNet allows for shifts in cornerstones as long as they are consistent with the present notion of truth and are deemed to be acceptable by the expert. If cornerstone shifts do not agree with the available ground truth or the expert's notion of truth, then the expert must either revert the changes that lead to the cornerstone shift or make further changes to ensure consistent operation of the system.

Cornerstone shifts such that they no longer support the knowledge defined using the cornerstone cases might be handled in one of two ways guided by the two perspectives on the nature of knowledge. One perspective is that because the knowledge no longer has any support of the cornerstone cases, it should be removed and new knowledge defined using the "corrected" cornerstone case be added. An alternative perspective is that the knowledge is still valid, albeit with no current support for it. There may be cases that the process has correctly handled using this knowledge and it may continue to be relevant. Such a view suggests that knowledge previously defined is still valid, and may be retained, with new evidence recorded for that knowledge as soon as it becomes available.

Although there are valid arguments in support of both approaches, the implementation of ProcessNet removes knowledge that no longer has the support of any cornerstone cases. This ensures that the knowledge within a process can shift and evolve in line with the concept drift in the underlying domain (Agnew et al. 1997; Clancey 1997).

Not all cornerstone shifts will lead to revisions. For example, minor deviations within the image pixels may not influence the high-level interpretations for the case, namely, the process output. Each of the shifted cornerstones that continue to be handled correctly by the process can be safely ignored. Cornerstone cases no longer handled correctly, however, require the expert to revise the process to handle these cases correctly.

The specific changes made to a process to handle the new versions of cornerstone cases are contingent on the internal nature of the processes and the expert's own assessment. For example, a process using an RDR knowledge base may require new rules added to the knowledge base, whereas a process implementing a handcrafted algorithm in source code would require the expert to make changes to the algorithm's source code. Once all of the cornerstones at a process are handled correctly, the validation process can continue onto validating other processes that are dependent on any revised processes.

Detecting cornerstone shifts and revising processes to handle these shifts may continue sequentially down the dependency graph of processes. The order in which the processes are validated and revised is determined by the system run order, which is the order in which processes are executed to satisfy the data requirements. All changes to cornerstone cases and processes must be considered tentative until the entire system of processes has been validated. Once all of the processes of the system have been validated and revised to handle any cornerstone shifts, the entire system can then be considered valid and the changes committed. Otherwise, the expert can wind back changes to any process and cornerstone case. This flexibility allows the expert to treat the system as an experimental sandbox to try different solutions, where the merits of each solution are evaluated by the cornerstone cases.

2.3.4 Change Validation in a ProcessNet

Consider an example of a ProcessNet system with processes A to F as shown in Figure 2.4. The arrows indicate the data flow between the processes. One process run order for the system is A, B, C, D, E, F. The run order of A, E, C, B, D, F is just as valid. Because there is no dependency between processes B, C, and E, they can be executed in parallel and interleaved in any order.

Consider a scenario in which process B is modified. The cornerstone cases at B would be used to validate B's consistent behavior against its existing cornerstone cases. After that, its direct dependents—in this case, process D—will also be validated against their own cornerstone cases. If process D handles any shifts in its cornerstone cases correctly, the validation process may terminate here. However, if process D requires further revisions leading to changes in its output, then its dependent process F must also be validated.

Consider an alternate scenario in which processes A and D are modified at the same time. The validation will start at process A, followed by its dependents (B, C, and E). Once these processes have been validated and/or modified to handle any cornerstone shifts, process D will be validated to check its own changes and those resulting from changes to outputs of processes B and C. The sequential validation in the process run order for the acyclic network of processes means that each process needs to be verified only once.

The incremental revision and validation of individual processes and, collectively, the entire system ensures that the system continues to adapt in the light of more labeled data and evolving expertise.

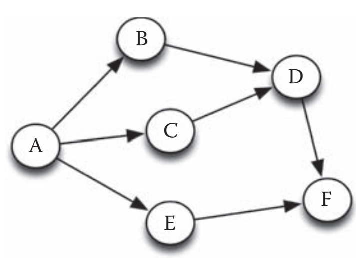

FIGURE 2.4
A typical ProcessNet system.

2.4 Lung Anatomy Segmentation Using ProcessNet

The ProcessNet framework has been used to incrementally develop an anatomy segmentation system that automatically segments multiple anatomical regions visible within sparse HRCT studies from real patients. Each HRCT study has an average of 15 images of 1 mm thickness, spaced every 15 mm along the axial plane of the body. The sparse HRCT studies lack the spatial continuity between images that was available within the MDCT studies used by Zhou et al. (2006). The sparse HRCT studies are however useful to detect a variety of diffuse diseases affecting the lung parenchyma. This is because they provide high spatial resolution while exposing patients to a lower dose of radiation as compared with MDCT studies. The studies were from real patients experiencing a variety of diffuse lung diseases that affected the parenchyma and pleura of the lung. A few patients had metallic implants within the body that introduced noisy artifacts in the scans, further complicating the segmentation task.

The anatomy segmentation system developed using ProcessNet segments the spine, sternum, shoulder blades, trachea, bronchi, esophagus, and lungs. Although the primary goal of the anatomy segmentation system is to correctly segment lung regions, the segmentation of the other anatomical structures provides useful spatial and anatomical cues. The expert incrementally introduces processes to undertake smaller tasks, which collectively operate to segment the anatomy. Each anatomical structure segmented relies on multiple processes. A process may implement an algorithm or call on a library of image processing functions available within the ImageJ library. A number of processes use RDR decision trees internally to guide classification.

The application of the ProcessNet framework to medical image segmentation evaluates three aspects of the framework and the developed system. First is an evaluation of ProcessNet in the development of the anatomy segmentation system in Section 2.4.1, which presents details about ProcessNet in operation. The capacity to continually revise and develop the anatomy segmentation system using ProcessNet means that it is never deemed finished. However, a snapshot of the anatomy segmentation system at the end of nine patient studies, denoted as PN9, is presented in Section 2.4.2. Finally, quantitative evaluation of the evolving anatomy segmentation system during various development stages is presented in Section 2.4.3.

2.4.1 Evaluation of ProcessNet in Operation

The anatomy segmentation system is developed incrementally in three training phases—A, B, and C. During each training phase, a vision expert introduces three patient studies, with an average of 15 images each, to the system and adapts the system to deal with them. Each image is first processed by the anatomy segmentation system and the results reviewed by the expert. If the expert notices any errors, the corresponding processes are determined and revised. Alternatively, the expert might introduce new processes to handle new segmentation tasks. Although system processing and revision occurs on a per image basis, a revision may involve multiple cases because of shifts in cornerstone cases requiring revisions to address more than one issue.

In the training phases, a process is added or modified as the need arises for better features or improved techniques to handle a specific task. Alternatively, processes that perform poorly or become redundant are removed from the system. In total, 61 revisions were made to the lung anatomy segmentation system across the three training phases. A

TABLE 2.1

Summary of Revisions during Training Phases A, B, and C

	A	B	C	Total
Studies	3	3	3	9
Images	59	47	58	164
System revisions	47	6	8	61
• Process additions	22	4	3	30
• Process modifications	147	51	23	221
• Process removal	0	1	3	4
Number of processes	22	25	25	25
Cornerstones added	52	45	38	135
Cornerstones shifted	21	28	55	104

summary of revisions undertaken during the training phases A, B, and C is presented in Table 2.1. Training phase B begins after revision number 47, and training phase C after revision number 53.

During training phase A, there were 47 system revisions/training events, in which a vision expert revised part of the system; these diminished to six and eight revisions by phases B and C, respectively. Each training event involved changes to one or more processes, as new processes were added and existing ones modified or removed.

Please verify if the changes made here are correct.

In phase A, 22 new processes were added and later modified 147 times during the course of training, as issues were discovered in the light of new training data. The vision algorithms necessary for the system were introduced and later refined during this period. The processes implemented basic image-processing algorithms such as image filtering, thresholding, morphology, image subtraction, and region growing. By training phases B and C, the number of new processes added has decreased to 4 and 3, respectively, and the modifications made to processes also decreased to 51 and 23, respectively.

In phases B and C, one and three processes, respectively, were removed. This is because some processes were found not to perform well. The new processes added in lieu of these processes offered better algorithms to carry out specific tasks and achieve overall system goals. For example, the lungs detection processes evolved from a simple thresholding algorithm in phase A, to applying locally parameterized active contours in phase B, and was finally superseded by a statistical model and an RDR classifier using relational features defined with respect to spine, sternum, and left and right shoulders in phase C. This was driven by the expert's choice, but ProcessNet supported the decisions by providing for consistency checks on cornerstone cases.

At the end of phase C, the 164 images from nine patient scans available for training had been used to add 135 cornerstones to 25 processes within the system. Further analysis of the cornerstones shows that these 135 cornerstones came from 97 unique training images. Because each process maintains its own set of cornerstone cases, a new training image that results in revisions to multiple processes may lead to multiple cornerstone copies of the same case, stored against the process revised.

Figure 2.5 shows the number of processes added, modified, and removed during each of the 61 revisions across training phases A, B, and C. The *x*-axis represents the revision number and the start of each training phase is delineated by dashed vertical lines. In a similar format, the number of cornerstones added and the number of shifts in existing cornerstones are shown in Figure 2.6. The large number of cornerstone shifts tends to coincide with significant changes to the system, such as the removal of processes during training

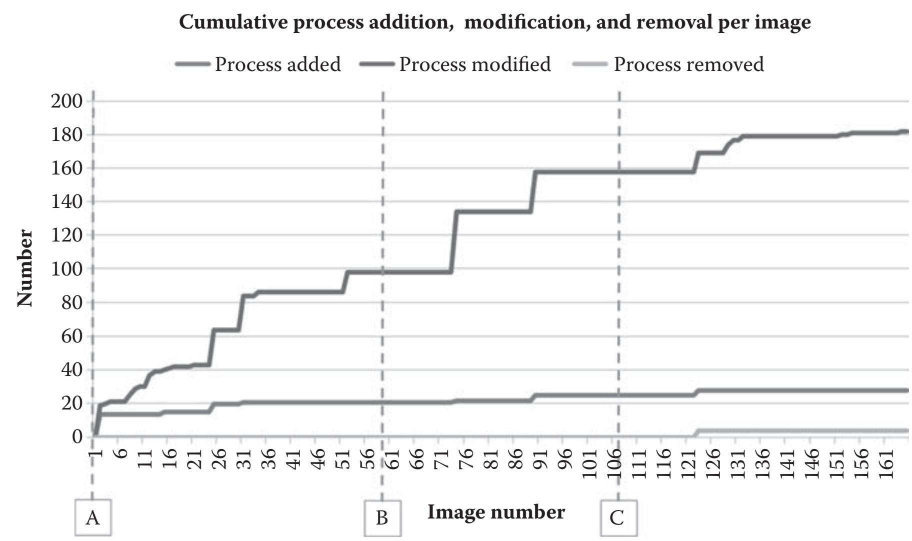

FIGURE 2.5
Number of processes removed, modified, or added for each system revision.

phases B and C. In total, there were 104 cornerstone shifts over the course of the training. A single cornerstone case at a process may shift with each revision made to the process.

In total, the vision expert made 221 revisions to the processes, in the form of either source code edits or the addition of new rules to process knowledge. Further analysis reveals that more than 70% of the modifications were source code edits, whereas the remaining 30% were revisions to the RDR knowledge bases used for classification within processes.

The number of cornerstone cases added does decrease with each training phase. This is not surprising because the cornerstone cases that are added during each training phase

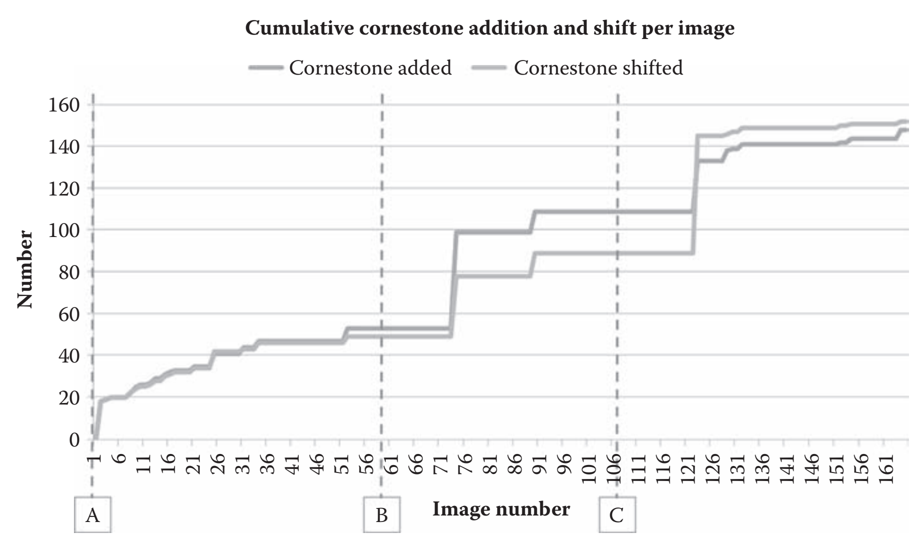

FIGURE 2.6
Number of cornerstone shifts and cornerstones added for each system revision.

represent noteworthy cases not previously encountered. As the cornerstone cases increasingly cover the variations in training data, the number of additions naturally decreases. Over the three training phases, an average of two cornerstone cases is added at each revision. This indicates that, on average, each revision required updates or changes to two processes.

The number of cornerstone cases that shift during revisions to the system increased during each training phase. As more cornerstone cases are added, changes to underlying processes would naturally affect an increasing number of other cornerstone cases. A change at a process that changes the output would naturally shift the cornerstone cases for the process, and the cornerstone cases of its dependents. It should also be noted that the majority of cornerstone cases shifted during the removal of existing processes. This is inevitable, as all cornerstone cases that previously covered the output of the deleted process must now be updated.

2.4.2 Analysis of the PN9 Anatomy Segmentation System

The incremental development using the ProcessNet framework means that the anatomy segmentation system can continue to evolve over time. The state of the system at the end of training phase C represents a snapshot of the knowledge within the system in the continuum of incremental development. This snapshot represents the knowledge acquired after nine patient studies and therefore is called PN9. The network of processes within the PN9 system is shown in Figure 2.7 and details about the processes and their dependencies are shown in Table 2.2. A short description of each process is also included in the table, identifying its role within the anatomy segmentation system.

An example of the segmentation results of the anatomy segmentation system is shown in Figure 2.8. The example shows the system correctly segmenting spine, sternum, shoulders, lung, and the left bronchus. It incorrectly identified the right bronchus as the esophagus. In ProcessNet, this represents an opportunity to revise the system and correct the knowledge responsible for identifying the bronchus, with the case serving as a cornerstone case for the relevant processes. The most likely candidates for revision would be *Bronchus Estimate*, *Bronchus Detection*, and *Sanity Check Bronchial Tree* processes.

The number of lines of Java source code for each process is listed in Table 2.3. The number of rules within the RDR knowledge bases used by processes is shown in Table 2.4. The ProcessNet framework automatically generates the source code in the *Main.java* class and the expert does not modify it. The expert introduces all the remaining classes, each representing a process within the system. Each process maintains an RDR knowledge base to capture knowledge within the form of rules. The expert modifies the source code within the class or adds rules to the knowledge base.

The complexity of the algorithm within a process may be roughly estimated by the number of lines of source code because it is indicative of the number of steps or method/function calls required to undertake the respective task. It is a commonly used metric in software engineering to define a module's complexity (Kaner and Bond 2004; Sommerville 1996, pp. 592–594). The smallest process is the *PrevKnownSternum* process, with only 113 lines of code, and it is responsible for simply loading the previously known sternum detected for the current HRCT study from the *results* directory.

The two largest processes are *ResultSaver* (802 lines of code) and *LungEstimate* (733 lines of code). The *ResultSaver* process captures the outputs of all other processes within the system and saves them to a results directory. Validating *ResultSaver* involves evaluating all the cases in the results directory to detect any shift in these cases. In essence, all the cases

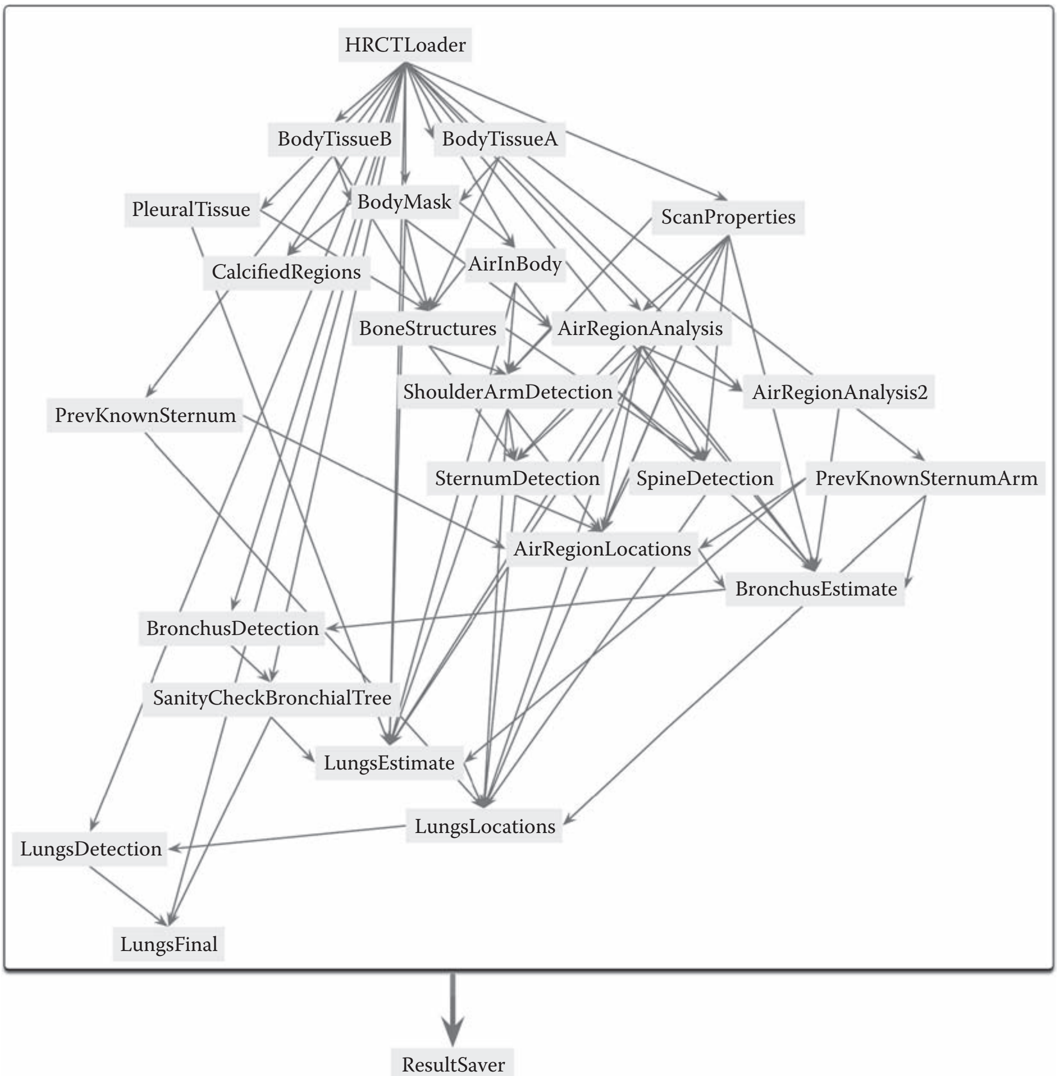

FIGURE 2.7
Data flow between processes of the anatomy segmentation system. ResultSaver gets data from all processes.

in the results directory form *ResultSaver's* cornerstone cases and record all cases processed by the system. This allows the expert to identify adverse effects on any case previously processed by the system, even if the case was not captured as a cornerstone case.

The *LungEstimate* process generates regions that are candidates for classification as lung or nonlung regions by the *LungsDetection* process. Because patients may experience diseases affecting the lung tissue, the volume of air present within the lungs may be significantly reduced. Therefore, the air-filled regions may not serve as a good estimate for lung regions. To handle these difficult cases, *LungEstimate* combines the air-filled regions with pleural regions detected by *PleuralTissue* process to generate candidate regions for lung regions. Of these, the regions that have already been labeled as trachea, bronchus, or esophagus by *SanityCheckBronchialTree* (and its predecessors—*BonchusEstimate* and

TABLE 2.2

Processes of the Anatomy Segmentation System, Their Purposes, and Dependencies

ID	Process	Purpose	Requires
1	HRCTLoader	Loads and enhances DICOM images	–
2	PrevKnownShoulderArm	Loads previously segmented shoulder regions	1
3	PleuralTissue	Segments pleural tissue	1
4	BodyTissueB	Segments fatty body tissue	1
5	BodyTissueA	Segments muscular body tissue	1
6	BodyMask	Generates a mask of the body	1, 4, 5
7	AirInBody	Detects air-filled regions within the body	1, 6
8	BoneStructures	Detects bone structures	3–7
9	CalcifiedRegions	Detects calcified regions	1, 6
10	PrevKnownSternum	Loads previously segmented sternum regions	1
11	ScanProperties	Extract scan information from DICOM header	1
12	AirRegionAnalysis	Measures structural features for air regions	1, 6, 7, 11
13	ShoulderArmDetection	Detects shoulder, arms, and shoulder blades	7, 8, 11, 12
14	SpineDetection	Detects the bone structures of the spine	8, 11–13
15	SternumDetection	Detects the bone structures of the sternum	8, 11, 12, 13
16	AirRegionLocations	Measures location-based features for air regions	1, 12
17	AirRegionAnalysis2	Measures textural features for air regions	2, 10–15
18	BronchusEstimate	Estimates bronchus, trachea, and esophagus	1, 2, 11–13, 16, 17
19	BronchusDetection	Labels bronchus, trachea, and esophagus	1, 18
20	SanityCheckBronchialTree	Resolve conflicts within bronchus regions	1, 19
21	LungsEstimate	Estimates regions likely to be lung	1–3, 6, 7, 11–13, 20
22	LungsLocation	Measures location-based features for lungs	2, 10–15, 21
23	LungsDetection	Labels lungs as left, right, or merged	1, 22
24	LungsFinal	Resolves conflicts between lung and bronchi	1, 20, 23
25	ResultSaver	Saves results of processes as XML and images	1–24

BronchusDetection) are eliminated. A 3D probability map of regions most likely to be lung is then used to score each candidate region's likelihood to be lung. The 3D map is generated using all previous lung segmentation results that have been verified by the expert. Each image that is being processed is aligned to the 3D map according to the position of the patient, the slice relative to the study and the pixel resolution. The 3D probability map

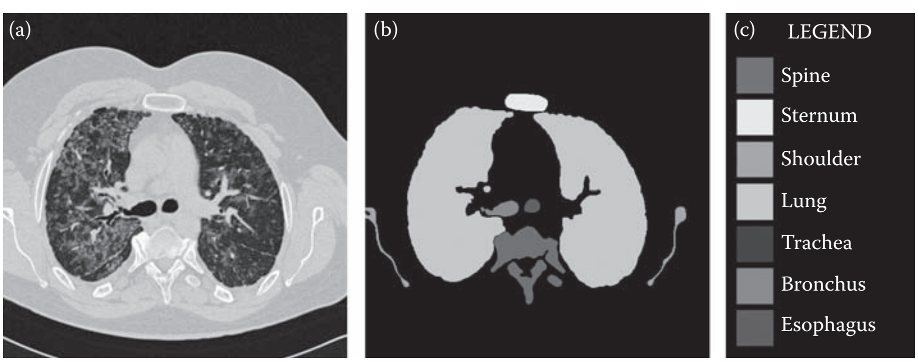

FIGURE 2.8
Anatomy segmentation results: (a) original HRCT image, (b) segmentation results, and (c) legend.

TABLE 2.3

Lines of Code for Processes of Anatomy Segmentation System

Source Code	Lines of Code (Includes Blank Lines for Readability)
HRCTLoader.java	272
PrevKnownShoulderArm.java	237
PleuralTissue.java	140
BodyTissueB.java	137
BodyTissueA.java	133
BodyMask.java	173
AirInBody.java	138
BoneStructures.java	169
CalcifiedRegions.java	144
PrevKnownSternum.java	**113**
ScanProperties.java	334
AirRegionAnalysis.java	353
ShoulderArmDetection.java	586
SpineDetection.java	547
SternumDetection.java	541
AirRegionLocations.java	428
AirRegionAnalysis2.java	187
BronchusEstimate.java	483
BronchusDetection.java	589
SanityCheckBronchialTree.java	348
LungsEstimate.java	**733**
LungsLocations.java	461
LungsDetection.java	588
LungsFinal.java	354
ResultSaver.java	**802**
Main.java	1012
Sum	**10,002**
Average	**389**

Some values, aside from the totals, are in boldface font; please indicate reason for emphasis, otherwise remove it.

is scaled, rotated, and translated to match the image being interpreted. The body's rotation is determined by the angle between the line connecting left and right shoulder regions and the image horizontal x-axis. The 3D map can be represented as a sequence of 2D slices, as shown in Figure 2.9. Each image represents the probability density map for 10% of the lung region along the axial plane.

The RDR knowledge bases have only been used at two processes—*BronchusDetection* and *LungsDetection*. These processes infer on a range of features generated by other processes

TABLE 2.4

Number of Rules Within RDR Knowledge Bases for Processes of Anatomy Segmentation System

Process Using RDR	Rules
BronchusDetection.java	47
LungsDetection.java	25

FIGURE 2.9
Lung probability map as slices, at 10% increments along the axial plane.

for extracted regions to classify them as trachea, bronchus, esophagus, lung, or unknown. The features include measurements on pixel intensity, image texture, structural features, as well as relational features that define a region's position relative to a number of anatomical landmarks detected by other processes. The anatomy landmarks used are the body, spine, sternum, and left and right shoulder regions. There are also a number of features that extract scan and patient information from the DICOM image header. Despite the range of features at the expert's disposal, the following features dominated the rule context within the resulting knowledge base of *LungsDetection*.

i. RELATIVE_DISTANCE_FROM_L2R_SHOULDERS: defines a value between 0.0 and 1.0, indicating the region's closest position on a line between left and right shoulders.
ii. RELATIVE_DISTANCE_ON_SPINE_STERNUM: defines a value between 0.0 and 1.0, indicating the region's closest position on a line between spine and sternum.
iii. DISTANCE_TO_BODY_CENTRE: defines the distance of the region to the body's centre in millimeters.
iv. LABEL_PROB_MODE: defines the mode score assigned by the 3D probability map to the pixels in the region within the *LungsEstimate* process.
v. LABEL_PROB_AVG: defines the average (mean) score assigned by the 3D probability map to the pixels in the region within the *LungsEstimate* process.
vi. AREA_REL_BODY: defines the ratio between the area of the region and the area of the body as measured for the given slice.
vii. Slice Location wrt Study: defines the relative position in terms of a percentage of the current slice within the study. Please verify if this data was formatted correctly.
viii. PIXEL_VALUE_MODE: defines the mode of the pixel intensities on the original HRCT image bounded by the region.

A number of rules at *LungsDetection* were added with only slight changes in the boundary conditions for the same features. This indicates that the specific boundary cutoff values for a feature were not clear to the expert, leading to some duplication in the rules as the correct value is incrementally discovered with new cases. There is potential for inductive algorithms, such as Induct-RDR (Gaines and Compton 1995), to further revise the knowledge base once a sufficient number of cases have been gathered and the ground truth labeled.

2.4.3 Quantitative Evaluation of Anatomy Segmentation

ProcessNet's incremental validated change strategy should help the expert in ensuring that the performance of the anatomy segmentation system is either improved or maintained consistently, despite the frequent revisions by the expert as he or she attempts to expand the system's ability to segment new anatomical regions or improve its existing capability to segment an anatomical region. This means that the system performance is expected to ideally improve or conservatively remain consistent, despite system revisions.

To evaluate ProcessNet's ability to facilitate incremental engineering of vision systems, the anatomy segmentation system built using ProcessNet was quantitatively evaluated against ground truth data hand-marked by domain experts, at the end of each of the three training phases—A, B, and C. The accuracy of the system in segmenting anatomy was quantitatively evaluated on two independent test sets of hand-marked ground truth. The first test set, called Anatomy-20, contains ground truth image masks for lungs, spine, sternum, and shoulder regions for each of 342 images from 20 patient studies. The second test set, called Lungs-40, contains ground truth masks for only lung regions in 583 images from 40 patient studies. The 20 patient studies in the Anatomy-20 test set is a subset of the 40 patient studies in the Lungs-40 test set. These test sets were not used for training the system.

The segmentation results were compared against the hand-marked ground truth masks for spine, sternum, shoulder, and lung regions using the metrics of sensitivity and specificity defined below

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP})$$

where TP is true positive, the number of pixels correctly labeled as lungs; TN is true negative, the number of the number of pixels correctly labeled as nonlung; FP is false positive, the number of pixels incorrectly labeled as lungs; FN is false negative, the number of pixels incorrectly labeled as nonlung.

Please verify if the changes made here are OK.

A good segmentation system should have high sensitivity and specificity values. A sensitivity of 100% for the lung would indicate that all pixels within the image considered to be lung by the ground truth were detected as lung by the system. A specificity of 100% would indicate that all pixels identified as not belonging to lung by the ground truth were successfully excluded.

The mean sensitivity and specificity for lungs, spine, sternum, and shoulders across the three training phases are shown in Table 2.5. The high specificity values indicate that the

TABLE 2.5

Mean Sensitivity and Specificity as Percentages at the End of Phases A, B, and C

	A		B		C	
Test Set	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**
Anatomy-20						
• Lungs	72.85	92.57	79.43	99.75	82.34	99.52
• Spine	40.90	99.81	88.28	99.59	88.28	99.59
• Sternum	7.58	99.95	40.72	99.85	46.98	99.79
• Shoulder	6.21	99.87	44.52	99.97	40.15	99.98
Lungs-40	88.40	99.80	93.90	99.70	95.90	99.40

processes are conservative in their detection and labeling of regions, hence, less likely to include regions that are not lung, spine, sternum, or shoulder. This is true for both Anatomy-20 and Lungs-40 data sets, with the mean specificity values around 99% across the three training phases. Only the lungs (in Anatomy-20) incorrectly include nonlung regions during phase A, as indicated by a specificity of 92.57%, which was improved in the subsequent training phases.

The sensitivity in segmenting lung regions saw an improvement across the three training phases for both the Anatomy-20 and the Lungs-40 data sets. The sensitivity for spine, sternum, and shoulder regions improves significantly during phase B and remains relatively consistent during phase C. The sensitivities for shoulders (at 40.15%) and sternum (at 46.98%) remain quite low even during phase C, but this must be evaluated in the context of their role within the system.

There are two reasons why this does not have an effect on lung segmentation. First, the intended purpose of shoulder and sternum segmentation is to acquire an approximation of landmarks to correctly align the 3D probability map used to segment lung regions. Even if all the pixels for the shoulder or sternum are not fully captured, even a partial segmentation is sufficient for their intended task. Second, *PrevKnownShoulderArm* and *PrevKnownSternum* help to compensate for regions missed by *ShoulderArmDetection* and *SternumDetection*, respectively, by retrieving the segmentation results for the preceding images for the same patient. Engineering a system as a network of processes allows limitations of an algorithm within one process to be compensated by algorithms in other processes. This means that the expert needed to ensure only a certain level of segmentation accuracy, around 40% sensitivity, to facilitate the remaining system.

Three examples of segmentation results from the Anatomy-40 test set are shown in Figure 2.10. The three columns represent segmentation performance for an image from three different patients—P34, P40, and P44—across the three training phases. The top row represents the original HRCT images and subsequent rows represent the segmentation results after training phases A, B, and C, respectively. As mentioned previously, the training and test sets contain patients with a variety of diffuse lung diseases. The lungs of patient P34 are affected by emphysema and honeycombing diseases patterns. The lungs of patient P40 are affected by ground-glass opacity. Patient P44 has normal lung tissue, albeit with pleural plaques affecting small parts of the lung wall.

At the end of training phase A, the anatomy segmentation system could correctly segment the healthy lung tissue in patient P44 but had difficulty segmenting lung regions in studies where the patient was severely affected by a variety of diffuse lung diseases, as in patients P34 and P40. In training phase B, revisions to the system led to an improvement in its ability to segment lung regions in patients affected by emphysema, as observable by the improvements in results for patient P34. It was also successful in segmenting shoulder regions missed previously by adjusting for the body's rotation as observed for patients P40 and P44. By the end of training phase C, the system correctly segments lungs severely affected by honeycombing (patient P34) and ground-glass opacity (patient P40). The segmentation for the sternum regions for patient P44 incorrectly includes other calcified regions and ribs. In patient P40, the sternum is fused with the ribs, thus making it difficult to separate them. Note that none of these cases from the test sets were used to train or revise the system. The improvements in these test cases were a result of improvements achieved on other training cases.

The erroneous cases in the test set denote the system's current limitation, and if used to revise the system, will become cornerstones for the processes of the anatomy segmentation system. Once captured as cornerstones, the validated change strategy of ProcessNet will

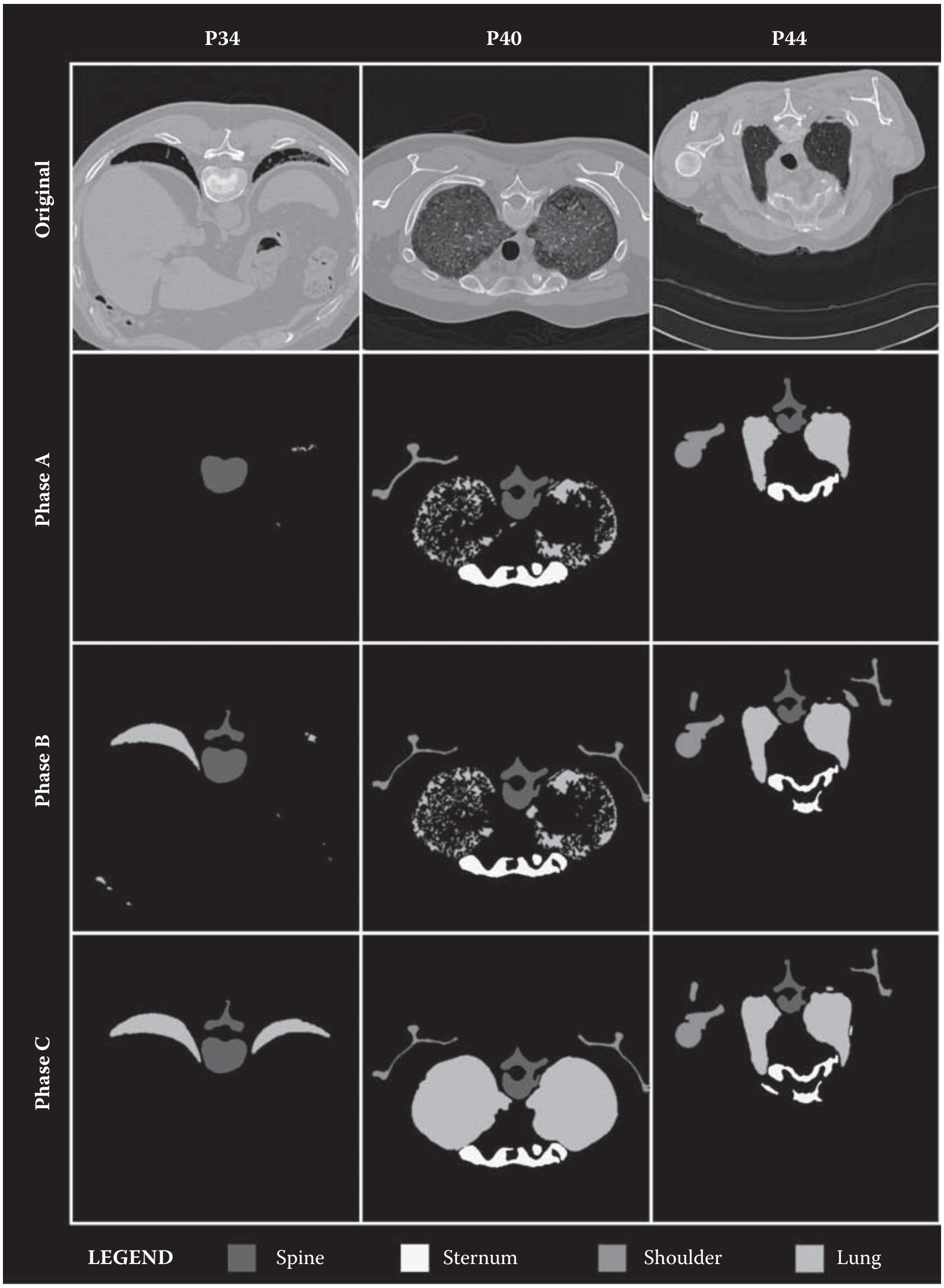

FIGURE 2.10
(See color insert.) Anatomy segmentation results: (left to right) patients P34, P40, and P44; (top to bottom) original HRCT scan, segmentation result after training phases A, B, and C.

ensure that any future changes to the system do not degrade the performance of the processes and system for these cases. Naturally, the strength of such a system lies in the quality of its cornerstone cases. The relatively poor segmentation results for shoulder regions between phases B and C indicate that the current set of cornerstones did not cover the variations present within the test set. Meanwhile, improvements in the segmentation for lung, spine, and sternum regions over the three training phases indicate that the cornerstone cases captured by the relevant processes were indeed representative of the variations in the test set.

Even though the expert is responsible for revising the processes, their efforts were directed and supported by the cornerstone-based validation of ProcessNet. It allows experts the opportunity to identify poorly performing parts of the system and use evidence in the form of cornerstone cases to guide the revisions to improve them. The automatic evaluation of system consistency via the cornerstone cases allows the expert to focus on solving specific vision tasks at each process and make judicious decisions on the quality of any changes. The ProcessNet framework facilitated incremental convergence in the knowledge of the anatomy segmentation system within its source code and rules.

2.4.4 Discussion

The ProcessNet framework helps vision experts in managing the incremental ad hoc revisions that are inevitable because of evolving quasi-expertise and incrementally available data when developing medical image segmentation systems. Experts revise systems, with the intent of addressing the incomplete and incorrect knowledge within the system. However, each revision carries a risk of adversely affecting the system performance for cases that the system handled correctly before the revision.

ProcessNet provides a systematic means to validate the changes at a process and its dependents, allowing the expert to revise each affected process using the cornerstone cases as concrete evidence for the desired handling of data. The management of changes across processes is critical as the complexity of the system grows. In the anatomy segmentation system developed using ProcessNet, the 25 processes were incrementally developed to address their limitations and adapt their content with changes at other processes. The complexity of managing these revisions was significantly reduced for the expert by the identification of specific issues via the cornerstone cases and allowing the expert to focus their effort on addressing each process in turn. The development effort required to engineer a network of processes was found to be manageable despite an increasing number of processes because the expert revises and validates a single process at a time. In addition, the acyclic nature of the dependencies means that the expert considers each process only once.

There are two limitations of the ProcessNet framework. The first is that the framework relies heavily on the expert in incrementally developing a vision system. This makes vision systems likely to be influenced by the level of the expertise at hand. A poor expert is likely to add incorrect knowledge or make imprecise revisions to the system which, in turn, may lead to further changes down the track. This is unavoidable as all expert-based systems, including RDR, fundamentally depend on a sufficient level of expertise (Cao et al. 2004). The role that cornerstone cases play within ProcessNet, however, helps to mitigate the influence of quasi-expertise. Any revisions proposed by the expert must satisfy the consistent handling of the cornerstone cases. Over time with an increasing number of cornerstone cases, the expert can use pattern recognition and machine learning methods that require sufficiently labeled data.

The second limitation is that the quality of the knowledge within the system is also dependent on the nature of cases that the system encountered during development. These cases, which prompted revisions by experts and formed cornerstone cases of processes, justify the knowledge within the process. Therefore, the resulting system is only as good as the quality of its cornerstone cases. This is a challenge faced by all data-driven systems. Over time with sufficient cases, one would expect that the accumulated cornerstone cases would provide better cover over the instance space of the domain. However, no guarantee can be provided on the uniform coverage of the domain by the training data available to a system.

2.5 Conclusions

Incremental and ad hoc development of medical image segmentation systems is an inevitable consequence of engineering systems, as the available data, expertise, and techniques evolve. The risk of a change degrading the system's performance restricts the expert's ability to build complex systems and confidently assimilate new data and techniques. In this chapter, an incremental validated change strategy called ProcessNet that mitigates these risks has been presented. Although ProcessNet may not eliminate all risks of degradation, the experiments support the idea that the accumulated set of cornerstone cases offers a data-driven means to assess the quality of a change to the system. This is the same argument and experience that underlies standard RDR.

ProcessNet captures and validates knowledge represented explicitly (such as in the form of rules) or implicitly in the libraries and algorithms defined in the source code of a process. The use of cornerstones at each process means that the knowledge behind heuristically defined algorithms can be supported by evidence grounded in data. The system developed using ProcessNet demonstrates that despite a large number of ad hoc revisions, a good solution can be discovered and improved on over time in a systematic manner. Although the ProcessNet framework has been applied specifically for the task of segmenting lungs and other anatomical structures in HRCT images, the framework is generic enough to address similar incremental ad hoc engineering for other types of medical image segmentation systems or computer vision systems in general.

References

Agnew, N.M., K.M. Ford, and P.J. Hayes. 1997. Expertise in context: Personally constructed, socially situated, and reality-relevant? In *Expertise in Context: Human and Machine*, Feltovich, P.J., Ford, K.M., and Hoffman, R.R. (eds.), 219–244. AAAI Press/MIT Press.

Amin, A., M. Bamford, A. Hoffmann, A. Mahidadia, and P. Compton. 1996. Recognition of hand-printed Chinese characters using ripple down rules. *Advances in Structural and Syntactical Pattern Recognition*, 371–380.

Archip, N., P.J. Erard, M. Egmont-Petersen, J.M. Haefliger, and J.F. Germond. 2002. A knowledge-based approach to automatic detection of the spinal cord in CT images. *IEEE Transactions on Medical Imaging*, 21:1504–16.

Archip, N., A. Fedorov, B. Lloyd, N. Chrisochoides, A.J. Golby, P.M. Black, and S.K. Warfield. 2006. Integration of patient specific modeling and advanced image processing techniques for image guided neurosurgery. *Proceedings of SPIE on Medical Imaging*.

Armato, S.G., M.L. Giger, and H. MacMahon. 1998. Automated lung segmentation in digitized posteroanterior chest radiographs. *Academic Radiology*, 5:245–255.

Aziz, Z., A. Wells, D. Hansell, G. Bain, S. Copley et al. 2004. HRCT diagnosis of diffuse parenchymal lung disease: Inter-observer variation. *Thorax*, 59(6):506–511.

Ballard, D.H., and J. Sklansky. 1976. A ladder-structured decision tree for recognizing tumors in chest radiographs. *IEEE Transactions on Computers*, C-25(5):503–513.

Ballard, D.H., and C.M. Brown. 1982. *Computer Vision*, Englewood Cliffs, NJ: Prentice-Hall.

Bankman, I.N. 2009. *Handbook of Medical Image Processing and Analysis*. 2nd ed. Elsevier Inc., ISBN: 978-0-12-373904-9 2008.

Bekmann, J., and A. Hoffmann. 2005. Improved knowledge acquisition for high-performance heuristic search. *International Joint Conference on Artificial Intelligence*.

Bovenkamp E.G.P., J. Dijkstra, J.G. Bosch, and J.H.C. Reiber. 2009. User–agent cooperation in multiagent IVUS image segmentation. *IEEE Transactions on Medical Imaging*, 28(1):94–105.

Boykov, Y., and M.P. Jolly. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in N–D images. *Proceedings of the International Conference on Computer Vision*, 1:105–112.

Boykov, Y., and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

Brown, M.S., L.S. Wilson, B.D. Doust, R.W. Gill, and C. Sun. 1998. Knowledge-based method for segmentation and analysis of lung boundaries in chest X-ray images. *Computerized Medical Imaging and Graphics*, 22(6):463–477.

Cao, T.M., E. Martin, and P. Compton. 2004. On the convergence of incremental knowledge base construction. *Discovery Science*, 207–218.

Clancey, W.J. 1997. *Situated Cognition: On Human Knowledge and Computer Representation*. Cambridge University Press.

Clouard, R., A. Elmoataz, C. Porquet, and M. Revenu. 1999. Borg: A knowledge-based system for automatic generation of image processing programs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):128–144.

Compton, P., and R. Jansen. 1988. Knowledge in context: A strategy for expert system maintenance. *Proceedings of the Australian Joint Artificial Intelligence Conference (AI'88) Adelaide*, 283–297.

Compton, P., and R. Jansen. 1989. A philosophical basis for knowledge acquisition. *European Knowledge Acquisition Workshop*.

Compton, P., L. Peters, G. Edwards, and T. Lavers. 2006. Experience with ripple-down rules. *Knowledge Based Systems*.

Cootes T., C. Taylor, D. Cooper, and J. Graham. 1995. Active shape models—Their training and application. *Computer Vision and Image Understanding*, 38–59.

Cootes, T.F., G.J. Edwards, and C.J. Taylor. 2001. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685.

Cremers D., M. Rousson, and R. Deriche. 2007. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215.

Crevier, D., and R. Lepage. 1997. Knowledge-based image understanding systems: A survey. *Computer Vision and Image Understanding*, 67(2):161–185.

Cuadra, M.B., C. Pollo, A. Bardera, O. Cuisenaire, J.G. Villemure and J.P. Thiran. 2004. Atlas-based segmentation of pathological MR brain images using a model of lesion growth. *IEEE Transactions on Medical Imaging*, 23(10):1301–1314.

Declaere, D., C. Smets, P. Suetens, and G. Marchal. 1991. Knowledge-based system for the three-dimensional reconstruction of blood vessels from two angiographic projections. *MBEC North Sea Special Feature*.

Publisher location for Elsevier?

Please provide publisher details (volume number, page range used).

Deklerck, R., J. Cornelis, and M. Bister. 1993. Segmentation of medical images. *Image and Vision Computing*, 486–503.

If possible, provide volume number and page range used here.

Doi, K. 2005. Current status and future potential of computer-aided diagnosis in medical imaging. *British Journal of Radiology*.

Draper, B., J. Brolio, R. Collins, A. Hanson, and E. Riseman. 1989. The schema system. *International Journal of Computer Vision*, 2(3):209–250.

If possible, provide volume number and page range used here.

Draper, B., A. Hanson, and E. Riseman. 1996. Knowledge-directed vision: Control, learning, and integration. *Proceedings of the IEEE*.

Draper, B. 2003. From knowledge bases to Markov models to PCA. *Proceedings of Workshop on Computer Vision System Control*.

Duda, R.O., P.E. Hart, and D.G. Stork. 2000. *Pattern Classification and Scene Analysis*. 2nd ed., New York: John Wiley & Sons.

Duncan, J.S., and N. Ayache. 2000. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):85–106.

Duryea, J., and J.M. Boone. 1995. A fully automated algorithm for the segmentation of lung fields on digital chest radiographics images. *Medical Physics*, 22(2):183–191.

Frangi, A.F., W.J. Niessen, and M.A. Viergever. 2001. Three-dimensional modeling for functional analysis of cardiac images: A review. *IEEE Transactions on Medical Imaging*, 20(1):2–5.

Provide volume and page range used, along with publisher name and location.

Freuder, E.C. 1977. A Computer System for Visual Recognition Using Active Knowledge Sources. In *International Joint Conference on Artificial Intelligence*. Cambridge, MA.

Gaines, B.R., and P. Compton. 1995. Induction of ripple-down rules applied to modeling large databases. In *Journal of Intelligent Information Systems*, 5(3):211–228.

Ghosh, P., and M. Mitchell. 2006. Segmentation of medical images using a genetic algorithm. *Genetic and Evolutionary Computation Conference*, 1171–1178.

Giger, M.L., K. Doi, H. MacMahon, C.E. Metz, and F.F. Yin. 1990. Pulmonary nodules: Computer-aided detection in digital chest images. *Radiographics*, 10(1):41–51.

Ginneken, B., H. Romeny, and M.A. Viergever. 2001. Computer-aided diagnosis in chest radiography: A survey. *IEEE Transactions on Medical Imaging*. 1228–1241.

If possible, provide volume number and page range used here.

Gong, L., and C. Kulikowski. 1994. VISIPLAN: A hierarchical planning framework for composing biomedical image analysis processes. *Computer Vision and Pattern Recognition*.

Grimson, W.E.L. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press.

He, L., Z. Peng, B. Everding, X. Wang, C. Han, K. Weiss, and W.G. Wee. 2008. A comparative study of deformable contour methods on medical image segmentation. *Image and Vision Computing*, 26:141–163.

Heimann, T., and H.P. Meinzer. 2009. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, 13(4):543–563.

Hu, S., E.A. Hoffman, and J.M. Reinhardt. 2001. Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images. *IEEE Transactions on Medical Imaging*, 20(6):490–498.

Jain, A.K., M.N. Murty, and P.J. Flynn. 1999. Data clustering: A review. *ACM Computing Surveys (CSUR)*, 31(3):264–323.

Jain, A., R. Duin, and J. Mao. 2000. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

Kaner, C., and W.P. Bond. 2004. Software engineering metrics: What do they measure and how do we know? *International Software Metrics Symposium*, 8:6.

Kass, M., A. Witkin, and D. Terzopoulos. 1988. Snakes: Active contour models. *International Journal of Computer Vision*, 321–331.

Kerr, J. and P. Compton. 2003. Toward generic model-based object recognition by knowledge acquisition and machine learning. In *Proceedings of the IJCAI-2003 Workshop on Mixed-Initiative Intelligent Systems, Acapulco*, 80–86.

Kikinis, R., M.E. Shenton, D.V. Iosifescu, R.W. McCarley, P. Saiviroonporn, H.H. Hokama, A. Robatino, D. Metcalf, C.G. Wible, C.M. Portas, R.M. Donnino, and F.A. Jolesz. 1996. A digital brain atlas for surgical planning, model-driven segmentation, and teaching. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):232–241.

Kolata, G. 2009. Good or Useless, Medical Scans Cost the Same. *New York Times*, March 1.

Kuhnigk, J.M., H.K. Hahn, M. Hindennach, V. Dicken, S. Krass and H.-O. Peitgen. 2003. Lung lobe segmentation by anatomy guided 3D watershed transform, *Proceedings of SPIE on Medical Imaging*, 5032:1482–1490.

If available, provide volume number and page range used here.

Levine, M., and A. Nazif. 1985. Rule-based image segmentation: A dynamic control strategy approach. *Computer Vision*.

Li, B., G. Christensen, J. Dill, E. Hoffman, and J. Reinhardt. 2002. 3-D inter-subject warping and registration of pulmonary CT images for a human lung model. *Proceedings of the SPIE Conference on Medical Imaging, San Diego, CA*, 4683:324–335.

Lin, P., C. Zheng, Y. Yang, and J. Gu. 2004. Medical image segmentation by level set method incorporating region and boundary statistical information. In *Computer Science*, 654–660.

Lorenzo-Valdes, M., G. I. Sanchez-Ortiz, R. Mohiaddin, and D. Rueckert. 2002. Atlas-based segmentation and tracking of 3D cardiac MR images using nonrigid registration. In *Lecture Notes in Computer Science*. 2488, Proc. Med. Image Comput. Comput.-Assisted Intervention, 642–650. Berlin: Springer-Verlag.

MacKay, D.J.C. 2003. *Information theory, inference, and learning algorithms*. Cambridge University Press.

If possible, provide volume number and page range used here.

Maillot, N., M. Thonnat, and A. Boucher. 2004. Towards ontology-based cognitive vision. *Machine Vision and Applications*.

Maintz, J., and M. Viergever. 1998. A survey of medical image registration, *Medical Image Analysis*, 2:1–36.

Malik, A., and T. Choi. 2006. A novel algorithm for segmentation of lung images. *Lecture Notes in Computer Science*, 346–357.

Massoptier, L., A. Misra, and A. Sowmya. 2009. Automatic lung segmentation in HRCT images with diffuse parenchymal lung disease using graph-cut. *International Conference Image and Vision Computing New Zealand*, 266–270.

If possible, provide volume number and page range used here.

Matsuyama, T. 1988. Expert systems for image processing-knowledge-based composition of image analysis processes. *Pattern Recognition*.

McInerney, T., and D. Terzopoulos. 1996. Deformable models in medical image analysis. *Proceedings of the Workshop on Mathematical Methods in Biomedical Image Analysis*, 171–180.

If possible, provide volume number and page range used here.

McInerney, T., and D. Terzopoulos. 2000. T-snakes: Topology adaptive snakes. *Medical Image Analysis*.

McInerney, T., and D. Terzopoulos. 2009. Deformable models. In *Handbook of Medical Image Processing and Analysis*, 145–166.

McKeown, D.M. Jr., W.A. Harvey, Jr., and J. McDermott. 1985. Rule-based interpretation of aerial imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(5):570–585.

McNitt-Gray, M.F., H.K. Huang, and J.W. Sayre. 1995. Feature selection in the pattern classification problem of digital chest radiograph segmentation. *IEEE Transactions on Medical Imaging*, 14(3):537–547.

Misra, A., A. Sowmya, and P. Compton. 2004. Incremental learning of control knowledge for lung boundary extraction. *Pacific Knowledge Acquisition Workshop as part of Pacific Rim International Conference on Artificial Intelligence (PRICAI)* 2004, 1–15.

Misra, A., A. Sowmya, and P. Compton. 2006. Incremental learning for segmentation in medical images. Biomedical Imaging: nano to macro, *3rd IEEE International Symposium*, 1360–1363.

Mitchell, T. 1997. *Machine Learning*. New York: WCB/McGraw-Hill.

Murthy, S.K. 1998. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2:345–389.

Noble, J.A., and D. Boukerroui. 2006. Ultrasound image segmentation: A survey. *IEEE Transactions on Medical Imaging*, 25(8):1–24.

Osher, S.J., and J.A. Sethian. 1988. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computer Physics*, 79:12–49.

Ozertem, U., and D. Erdogmus. 2007. A nonparametric approach for active contours. *International Joint Conference on Neural Networks*, 1–4.

Park, M., L.S. Wilson, and J.S. Jin. 2000. Automatic extraction of lung boundaries by a knowledge-based method. Selected papers from the Pan-Sydney workshop on Visualisation, 2:11–16.

Park, M., B. Kang, S. Jin, and S. Luo. 2008. Computer aided diagnosis system of medical images using incremental learning method. *Expert Systems With Applications*, 36:7242–7251.

Paulus, S., U. Ahlrichs, B. Heigl, J. Denzler, J. Hornegger, M. Zobel, and H. Niemann. 2000. Active knowledge-based scene analysis. *Videre: Journal of Computer Vision Research*, 1(4):25.

Pien, H.H., A.J. Fischman, J.H. Thrall, and A.G. Sorensen. 2005. Using imaging biomarkers to accelerate drug development and clinical trials. *Drug Discovery Today*, 10(4):259–266.

Popper, K.R. 1963. *Conjectures and Refutations*. London: Routledge and Kegan Paul.

Prayote, A. 2007. Knowledge based anomaly detection. PhD thesis. University of New South Wales.

Reinhardt, J.M., R. Uppaluri, W.E. Higgins, and E.A. Hoffman. 2000. Pulmonary imaging and analysis. In *Medical Image Processing and Analysis*, J.M. Fitzpatrick and M. Sonka (eds.), 1005–1060.

Richards, D. 2009. Two decades of ripple down rules research. The *Knowledge Engineering Review*, 1–26.

Rogowska, J. 2009. Overview and fundamentals of medical image segmentation. In *Handbook of Medical Image Processing and Analysis*, I. Bankman (ed.), 73–90.

Sahba, F.T., H.R. Salama, M.M.A. 2007. A reinforcement agent for object segmentation in ultrasound images. *Expert Systems with Applications*, 1–9.

Singh, P.K., and P. Compton. 2005. Combining machine learned and heuristic rules using GRDR for detection of honeycombing in HRCT lung images. *Knowledge-Based Intelligent Information and Engineering Systems*, 131–137.

Sluimer, I.C., M. Niemeijer, M., and B. van Ginneken. 2004. Lung field segmentation from thin-slice CT scans in presence of severe pathology. *Proceedings of SPIE Medical Imaging*, 5370:1447–1455.

Sluimer, I., A. Schilham, M. Prokop, B.van Ginneken. 2006. Computer analysis of computed tomography scans of the lung: A survey. *IEEE Transactions on Medical Imaging*, 25(4):385–405.

Please verify publisher location (city/state) provided here.

Sommerville, I. 1996. *Software Engineering*. 5th ed. Reading, MA: Addison-Wesley.

Stansfield, S.A. 1986. ANGY: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2):188–199.

Sutton, M.A. Image segmentation by fuzzy clustering: Methods and issues. *Handbook of Medical Image Processing and Analysis*, 91–111.

Ukil, S. and J. Reinhardt. 2004. Smoothing lung segmentation surfaces in 3D X-ray CT images using anatomic guidance. *Proceedings of SPIE*, 1066–1075.

Drucker, H., C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. 1996. Support vector regression machines. *Advances in Neural Information Processing Systems*, 155–161, MIT Press.

Vernazza, G.L., S.B. Serpico, and S.G. Dellepiane. 1987. A knowledge-based system for biomedical image processing and recognition. *IEEE Transactions on Circuits and Systems*, 1399–1416.

Wangehheim, A.V., H. Wagner, W. Comuello, D. Krechel, M.M. Richter, and P. Conrad. 2000. Cyclops—Expert System Shell for Development of Applications in Area of Medical Image Analysis Workshop 2000 of The German–Brazilian Cooperation Programme on Information Technology, 1:129–134.

Webb, S. 1990. *The Physics of Medical Imaging*. Bristol, PA: Adam Hilger.

Wei, J. 2002. Image segmentation based on situational DCT descriptors, *Pattern Recognition Letters*, 23:295–302.

Wismueller, A. 2009. Segmentation with neural networks. *Handbook of Medical Image Processing and Analysis*:113–143.

Zhang, L., and J. Reinhardt. 2000. 3D pulmonary CT image registration with a standard lung atlas. In *Proceedings of SPIE on Medical Imaging*, 4322:67–77.

Zhang, D., and D.J. Valentino. 2001. Segmentation of anatomical structures in X-ray computed tomography images using artifical neural networks, *Proceedings of SPIE*, 4684:1640–1652.

Zhang, L., E.A. Hoffman, and J.M. Reinhardt. 2003. Atlas-driven lung lobe segmentation in volumetric X-ray CT images. *Proceedings of SPIE*, 5031:306–315.

Zheng, B., J.K. Leader, G.S. Maitz, B.E. Chapman, C.R. Fuhrman, R.M. Rogers, F.C. Sciurba, A. Perez, P. Thompson, W.F. Good, and D. Gur. 2003. A simple method for automated lung segmentation in X-ray CT images. *Proceedings of SPIE on Medical Imaging*, 5032:1455–1463.

Zhou, X., T. Hayashi, T. Hara, H. Fujita, R. Yokoyama, T. Kiryu, and H. Hoshi. 2006. Automatic segmentation and recognition of anatomical lung structures from high-resolution chest CT images. *Computerized Medical Imaging and Graphics*, 299–313.