Journal of Digital Imaging

iScout: An Intelligent Scout for Accessing and Navigating Large Image Sets in a PACS

Adam Weathermon, BASc,[1] Jonathan Tsui, BSc,[1] and Robert Rohling, PhD[1,2]

A new software tool for PACS, called iScout (intelligent scout), has been developed and optimized for a radiology workstation. The purpose of iScout is to display an overview of a large image series, allowing the user to select images for priority downloading from a PACS server to a PACS workstation. This allows radiologists to reduce the delays that are associated with downloading hundreds or even thousands of images. Several schemes that semiautomatically manage the download process are presented along with tests to measure performance. The results of the tests confirm that priority downloading provides faster access to images in large image series and that the time savings increase in proportion to the study size.

KEY WORDS: Reconstructed scout, iScout, priority downloading, PACS

THE GROWING SIZE of image studies used in diagnostic imaging presents several challenges for PACS (Picture Archiving and Communication Systems).[1-4,21] One of the main challenges is to provide timely access to the images when they are being transferred over network connections in a hospital. The increased size of studies is due to a combination of factors, including the growing capabilities of imaging modalities such as multislice spiral CT (computed tomography) scanners and a trend in scanning techniques toward scans that are more data intensive, such as whole-body scanning. The emergence of 3-dimensional imaging techniques is also driving these changes.[5] In the data presented in ref. 1, the average number of images per CT (computed tomography) examination at a large medical facility doubled between 1994 and 1999, and the similar average in MR (magnetic resonance) studies increased by approximately 60% during the same time period. Perhaps more significantly, the same study showed the average amount of data produced per CT scanner per year increased from approximately 650 Gigabytes (Gb) to 1500 Gb per scanner per year between 1994 and 1998, and the authors projected this amount to triple to 4500 Gb per scanner per year by the year 2003. To illustrate the challenges, consider that a single scan from a modern CT scanner can generate over a thousand images with an uncompressed storage size of 524 Megabytes (assuming 1024 images with 512 × 512 pixels at 2 bytes/pixel and no compression). In a typical PACS, the set of images is packaged together with patient information into a database element called a study. Inside the study are one or more series—a set of images usually acquired during a single scan by the imaging device. In the CT example, the entire set of images can be represented as a single series in a study. A series is normally downloaded in its entirety to a workstation on an image-by-image basis. This represents a significant amount of data that must be transferred over a network from a PACS server to a PACS workstation when the series is reviewed by a radiologist. For example, the previously mentioned series would require

[1] From the Medical Imaging Group, McKesson Information Solutions, Richmond, British Columbia, Canada.

[2] From the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, British Columbia, V6T 1Z4 Canada.

Correspondence to: Robert Rohling, PhD, tel: 604-822-2045; fax: 604-822-5949; e-mail: robert_rohling@yahoo.ca

Copyright © 2004 by SCAR (Society for Computer Applications in Radiology)

Online publication 19 April 2004
doi: 10.1007/s10278-004-1005-7

Table 1. Measurement of iScout Overhead Averaged over 30 Trials

Series size (images)	Download time with iScout generation (s)	Download time without iScout generation (s)	Time overhead (s)	Percent overhead
118	10.94	9.58	1.37	14.3
500	28.3	21.5	6.81	31.7
1000	71.1	49.6	21.6	43.6

over seven minutes to transport assuming full utilization of a 10-Mbps network connection. Since full network bandwidth is unrealistic, actual delay times are longer for similar series and will continue to increase as the scanning technology progresses. Use of faster network hardware can improve the situation, but the costs of PACS will increase and the downloading delays will still be considerable. Moreover, download times for image series are also crucial in other applications such as telemedicine or web-based access that utilizes the reduced bandwidth of Internet connections.

An additional problem is that the size of some studies exceeds the physical memory of the workstation. Specifically, current 32-bit PCs can access 4 Gb of memory of which only 2 Gb can be reliably used to store the study. Therefore studies greater than 2 Gb cannot be loaded in their entirety and must be loaded one portion at a time. For these reasons, more intelligent strategies for image downloading and display are needed to allow timely and easy access to large data sets.

Previous research has focused mainly on compression techniques to simply improve the speed of transfer.[6] Another approach is to construct a multiresolution hierarchy of images and transmit low-resolution images quickly, followed by higher resolution.[7] We take a different approach and focus on methods that require little preprocessing overhead and focus on the *order* of downloading. The order of downloading is important because the workstation user is typically interested in a subset of images in a large study and will want to download these first. For example, a neuroradiologist will look only at a small number of the images in a whole-body spiral CT scan. For these reasons, a method of downloading images according to anatomical region with minimal overhead would be more efficient. To address these issues, the intelligent scout (iScout) tool has been developed. It shows an overview of a large series in a format similar to a traditional scout image, but instead of a scout image from the scanner, the iScout consists of two multiplanar reslice (MPR) views reconstructed directly from the series. By interaction with the iScout, the user can select and download a subset of images for review prior to downloading the rest of the images.

The reconstructed scout also provides a navigational tool to the user. On a typical radiology workstation, images can be displayed in stack mode where a single image from the series is displayed, with the user navigating through the image series by scrolling with the mouse wheel. By contrast, with the iScout, the user is able to navigate using the overview image by selecting the relevant anatomy with the mouse. The tool is integrated with the workstation so that the workstation will immediately display the image corresponding to the location selected in the iScout overview image. Using the iScout tool in this way, the user can quickly navigate from one anatomical region to another by observing a global overview of the set of images and their spatial locations.

One additional advantage of using a reconstructed scout occurs in situations where a traditional scout image may not spatially match a series of images because of patient motion between the scout scan and the original scan. For example, in MR imaging, a scout image may be taken several minutes after the initial scan. When patient motion occurs between acquisitions, a given physical location in the scout may not show the same anatomical feature as the same location in the series of images. This could have serious consequences in examinations where matching anatomical features (such as spinal elements) among images is important. Since the iScout is constructed from the series data directly, there are no repositioning errors associated with the relative location of anatomical features in the scout and series images.

A prototype iScout, briefly described in ref. 8, was shown to radiologists as part of a demonstration PACS. Based upon feedback from these demonstrations, a new version of iScout (internally called iScoutPro) was developed. The new version has an improved user interface, new preconfigurable download logic, and a number of user-defined parameters. In this article, we describe the new iScout in detail and present the results of performance tests.

MATERIALS AND METHODS

Description of iScout

High-Level Description. The iScout is a self-contained component that has been integrated with a multimodality radiology workstation from the McKesson Medical Imaging Group (Richmond, BC, Canada). iScout is written as a COM (Component Object Model) object in Visual C++ on a Windows platform. OpenGL is used for texture mapping and rendering the iScout. From a user's perspective, the iScout has an interface showing the download process according to anatomy, allowing users to select particular images for priority downloading and to adjust parameters to customize the download logic.

Requirements for iScout. The requirements for creating iScout images are as follows:

- The iScout must be created with a minimum of computational resources.
- iScouts must have low storage sizes relative to the storage size of a study.
- iScouts must conform to the DICOM standard.
- An iScout must depict an accurate overview of a given series.
- The iScout must contain patient orientation information.
- iScouts can be created on either on a PACS server or on a workstation. If the iScout is created on the workstation, the user can view the iScout while it is being built incrementally and still use the iScout for priority downloading during creation.

The requirements for viewing an iScout are as follows:

- iScouts must efficiently display all information in the iScout to the user.
- iScouts must occupy a small amount of screen space on the monitor.
- The user can select quickly one or more images in the vicinity of the desired part of the anatomy.
- iScouts must be shown according to a standard patient orientation.

iScout Description. An iScout is composed of two MPR images orthogonal to the set of images in the series. The two MPR images are created by extracting the middle row and middle column from each image in the series. The middle rows and columns of the images are then organized according to the image position and orientation information found in the DICOM header creating two orthogonal reslices of the series. Any gaps between images in the series are filled by interpolation so that all portions of the iScout are shown in the correct position, orientation, and scale. By using the middle row and column of pixel data, it is easy to create the iScout images with minimal computational cost. These two orthogonal reslices of the study represent a visual summary of the entire series that can be used for navigating to different locations in the study.

The iScout tool is composed of four components: the iScoutCreator, iScoutDownloader, iScoutViewer, and iScoutStorage. These components reside in a single COM object that can be run on either a PACS server or workstation and can be controlled using a simple COM interface. The iScoutCreator is responsible for creating the MPR images and adding them to the study. iScoutDownloader monitors and controls the image transfer from the server to the workstation. iScoutViewer displays the iScout images in the standard viewing window, and iScoutStorage stores the information needed to perform the semiautomatic priority downloading feature. Each of the software components is described in detail below.

iScoutCreator. iScoutCreator prepares the study for viewing by gathering and organizing data from the PACS server. The main features of iScoutCreator are

- reconstruction of the iScout's MPR images
- storage of the each original image's spatial location in a private tag of the iScout DICOM object
- storage of the iScout as a Secondary Capture DICOM object

iScoutDownloader. iScoutDownloader monitors and controls the image transfer from the server to the workstation. Through a series of configurable parameters, the user can control the logic of the download operation. Aside from several preset download configurations, iScoutDownloader also allows on-the-fly requests to download specific images. The main idea behind the download process is to automatically assign an order to the images to be downloaded. Without user interaction, these images are then downloaded in that order. But user interaction with the series *during* downloading can affect the designated order so that images of current interest are given higher priority. Deciding the ultimate order during downloading is thus a question of blending the preconfigured logic and user actions. To facilitate a number of different blending methods, iScoutDownloader has several preconfigured download logic schemes and user-defined download parameters, as described below.

Preconfigured download logic. It is easy to imagine several schemes for guessing in advance which images will be viewed first. A few such schemes are implemented and the user is given the choice of scheme.

- ***sequential***
    - The sequence in which the PACS server sends images to the workstation is the same as the order sent from the scanner.

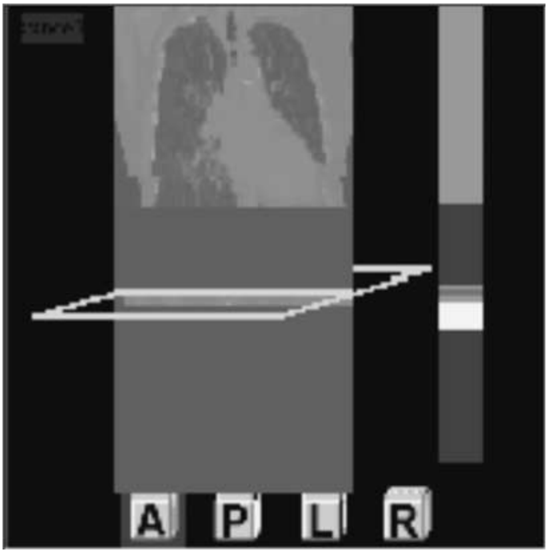


Fig 1. Snapshot of iScout while downloading. On the right hand side, a color bar (shown here in gray scale) represents the download information of each image of the study. Green (light colored) areas indicate images that are loaded, red areas (dark areas) are not loaded, and yellow (brightest) regions have been requested for priority downloading. A wireframe indicates the currently viewed image location in the corresponding stack. A button in the upper-left corner allows the user to cancel priority-downloading requests.

- iScout does not change the order list in this method, so this is the same order the PACS workstation receives the images when no iScout is installed.

- ***spatial***
  - This sequence downloads images according to their spatial location with respect to the Frame of Reference in the DICOM object. The choice of ‘‘top-down’’ or ‘‘bottom-up’’ is specified in the user-defined download parameter: *direction.*

- ***historical–statistical***
  - This sequence is meaningful only when the images of the study have been previously downloaded either by the same user at an earlier time or by another user.
  - The order of this sequence is calculated from both the previous download order and the record of which images were viewed on the stack.
  - The images that were most frequently viewed would have highest priority.

- ***current-position***
  - The sequence is based on the user’s actions, so that the current image being displayed on the stack and its neighbors are downloaded first.

With any of these schemes, the user can still exert direct control over the download order. For example, if sequential downloading requires the user to wait for a desired image to appear, the user can simply click (or click-and-drag for multiple images) on the corresponding anatomical region in the iScout. The desired image at that region will then be given higher priority and downloaded immediately.

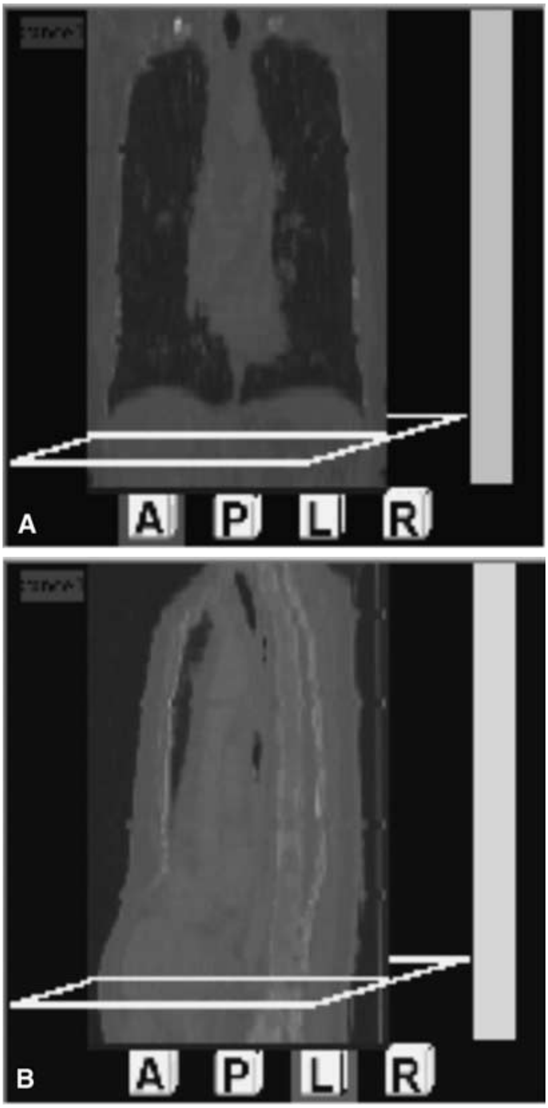


Fig 2. The user can choose between the two orthogonal views stored in iScout. Anterior and left views are reconstructed from this particular series. The two additional views (posterior and right) are simple reversals of these views. The bottom buttons are shown as cubes whose orientation and labeling describe the patient orientation.

User-defined download parameters. These parameters control some of the pre-configured download behaviors described above. The parameters include:

- **direction**—This setting specifies the direction that the download logic will take place in *sequential* or *spatial* logic. The choices are ‘‘top-to-bottom’’ or ‘‘bottom-to-top.’’ The explicit download request will also proceed in the direction chosen.

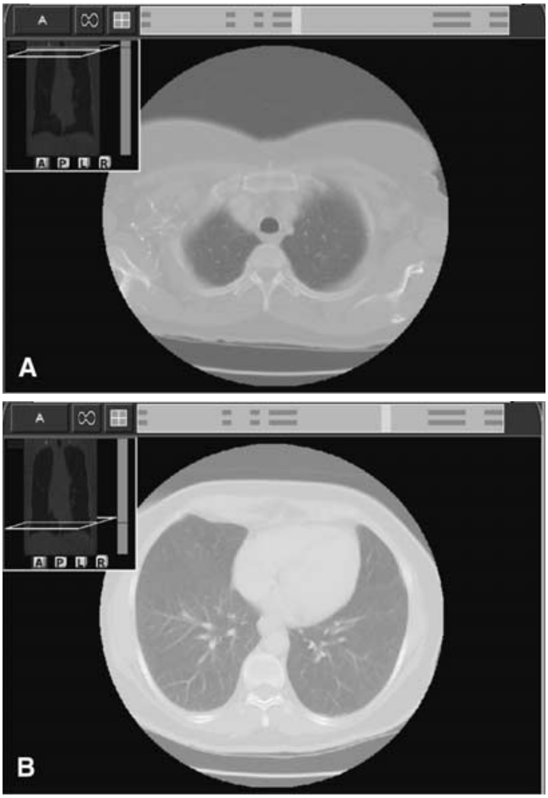
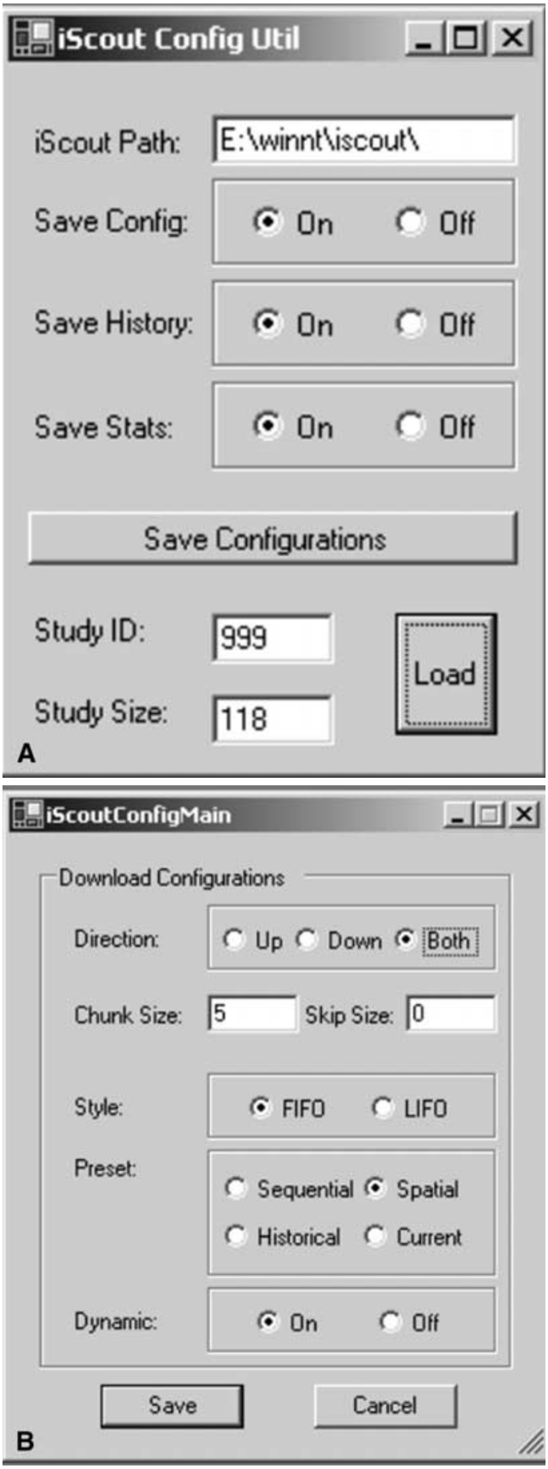


Fig 3. iScout is inserted as a picture-in-picture within a traditional stack view of a series. The iScout and the stack are linked, so that scrolling to a new image on the stack will update the frame in iScout to show the corresponding plane and vice versa.

- **image chunk size**—This setting specifies the size of a chunk of neighboring images to be downloaded around the image requested explicitly by the user or skipped to when using image skip (in *current-position*,*sequential*, or *spatial* logic).
- **image skip**—This setting specifies the number of images to skip when downloading. This option allows the user to sample the study quicker, (in *sequential*,*spatial*, or *current-position* logic).
- **request style**—This setting specifies the way iScout responds to explicit user requests for images, either first-in-first-out (FIFO) or last-in-first-out (LIFO). This refers to how the download order is changed when more than one request is made to change this order (for all logic methods).

iScoutViewer. iScoutViewer is responsible for displaying the iScout and interfacing with the user. The display is available during and after image downloading. Interaction with iScout is consistent with other imaging tools on the workstation and includes window-level control and patient orientation labels. The integration of the iScoutViewer in the workstation software synchronizes the operations to the rest of the PACS. The functionality is described in Figures 1–3.



iScout Config Util

iScout Path: E:\winnt\iscout\

Save Config: ● On ○ Off

Save History: ● On ○ Off

Save Stats: ● On ○ Off

Save Configurations

Study ID: 999

Study Size: 118

Load

A

iScoutConfigMain

Download Configurations

Direction: ○ Up ○ Down ● Both

Chunk Size: 5 Skip Size: 0

Style: ● FIFO ○ LIFO

Preset: ○ Sequential ● Spatial ○ Historical ○ Current

Dynamic: ● On ○ Off

Save Cancel

B

Fig 4. User-selected parameters. The Study ID and Study Size load function are only used for testing purposes.

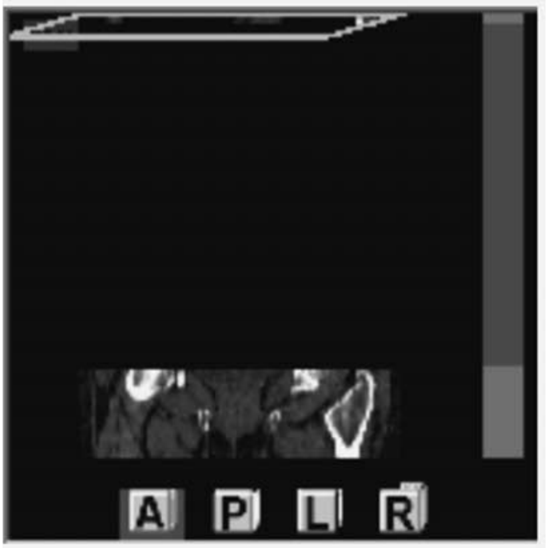


Fig 5. ***Spatial*** **logic with bottom-to-top downloading.**

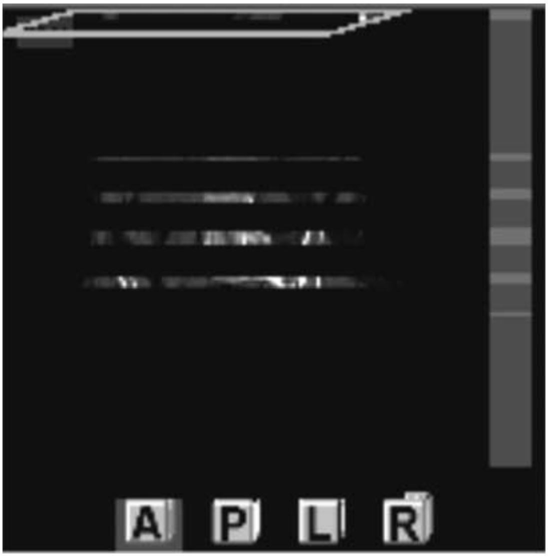


Fig 6. ***Spatial*** **logic with middle-out and image-skip downloading.**

iScoutStorage. iScoutStorage preserves the information that iScout collects about the users' activities during study review. This information is used to optimize downloading for both a particular study and user. The information includes:

- **usage statistics**—iScout records the number of times a specific study is viewed, monitors the average download speed, and keeps track of each image's view frequency. This information is used in the *historical–statistical* preconfigurable download logic.
- **history of previous usage**—The parameters from the last usage are kept and are restored for the next session. Parameters such as window/level and patient orientation are restored when opening the study again. Requested images, viewed images, and download order are also preserved.
- **user configuration data**—The preconfigured download logic and the user-defined download parameters are kept so that the next time the study is opened *by that user*, the appropriate configuration will be used.

Probablity Downloading Model. To consider the potential time savings of explicitly selecting subsets of images for priority downloading using the iScout tool, a new model of the downloading process has been developed. Previous results in ref. 8 show that priority downloading significantly reduces the time to access an image of interest while downloading a large image series. The premise in ref. 8 is that the user is interested in a single image in the study and workflow depends on the time for that image to download. A more reasonable model considers the situation where multiple images are selected for priority downloading rather than a single image. The purpose of the model is to predict the expected time needed to download a subset of images (normally for very large image series) without any priority downloading capability. We use this model to validate our test data and extend the results to larger image series.

Some simplifying assumptions are needed for the new model. First, the bandwidth of the system is assumed to be constant at all times during the download process, and the images are of equal size. If the order of images to be downloaded is considered random, a random variable could be used to predict the average time needed for a specific series of images to be downloaded. However, normally the ordering of images is not random and sometimes is well ordered with respect to position of the images in space. Under these conditions, a simpler model of the download process can be used. Specifically, we assume the images are approximately ordered and in this case the downloading time for a subset of images is, on average, half of the time to download the entire study, plus the time required to download the images that compose the subset. This model additionally assumes the number of images in the subset is much less than the total number of images in the study. Under these conditions, the time to download a subset can be approximately expressed as

$$T_C = T_{\mathrm{all}}/2 + (C/N)T_{\mathrm{all}} = (1/2 + C/N)T_{\mathrm{all}},$$

where T_{all} is the time needed to download the entire study, integer C is the subset size, and integer N is the size of the study. The time needed to download a subset of images using iScout through explicit requests is ideally

$$T'_C = (C/N)T_{\mathrm{all}},$$

which is simply the time needed to download this portion of the study relative to the total download time. This model is not meant to estimate the relative efficiencies of the various logic schemes and user-defined parameters, only the aspect of priority downloading shared by all schemes.

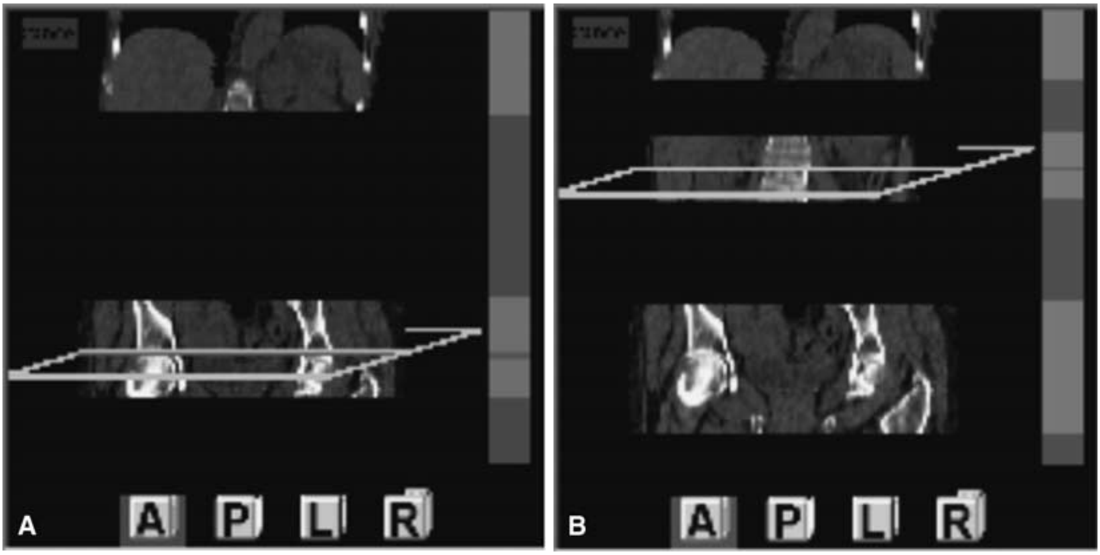


Fig 7. *Current-position* logic with middle-out downloading. When the user moves to a new location in the stack where images are not yet downloaded (switching from left to right figure), the download order is updated so that images nearby will be downloaded first.

RESULTS

iScout Implementation

One of the main features of iScout is the ability to select single images or groups of images from a study for priority downloading. Using iScout, radiologists can semiautomate the download prioritization by using one of the logic schemes. The menu used to modify the user preferences for iScout is shown in Figure 4.

Figures 5 and 6 show the effect of these parameters for *spatial* download logic. Along with a large skip size and subset size combination, the user can obtain a quick overview of a region of interest with little manual interaction.

With the *current-position* logic scheme, the download order is continually modified so that images nearby the current stack position are given high priority. The user can specify the direction to load from the current image, and specify a nonzero skip size to give a quick overview nearby the current stack position. Figures 7 and 8 demonstrate the *current-position* method. It is worth restating the point that the preceding figures have generally shown iScout implemented on the workstation with a study being loaded for the first time. If the iScout is implemented on the server, or the study had been previously downloaded, then a fully reconstructed iScout is displayed immediately. The user then interacts with iScout, as before, to determine priority downloading. In the *sequential* logic scheme, no change is made to the ordering of the images that are transferred to the workstation, while in the *historical* method, any previous usage statistics from

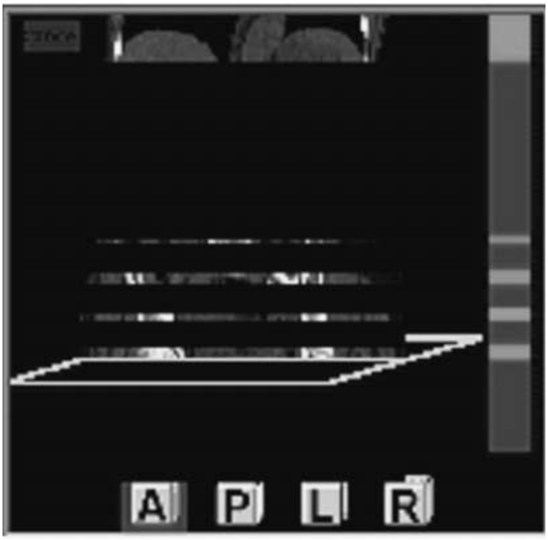


Fig 8. *Current-position* logic with bottom-to-top and image-skip downloading.

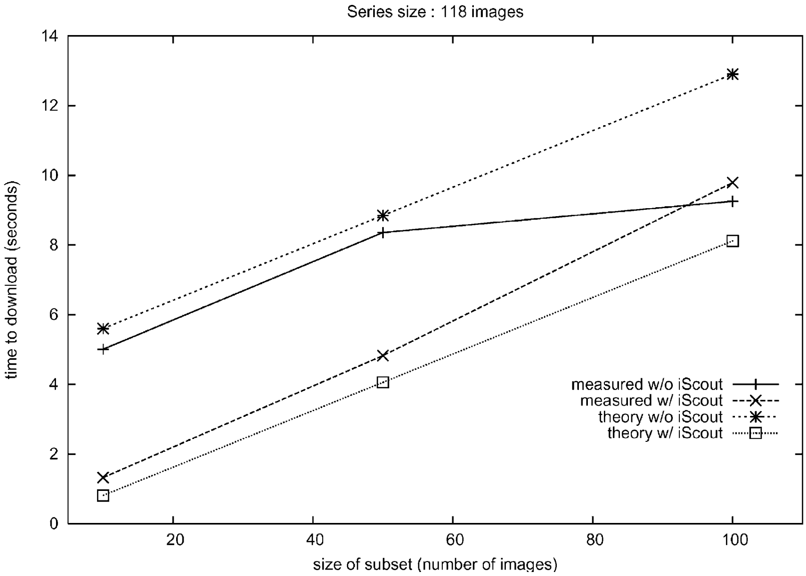


Fig 9. Statistics and predictions of the downloading model for a series of 118 images.

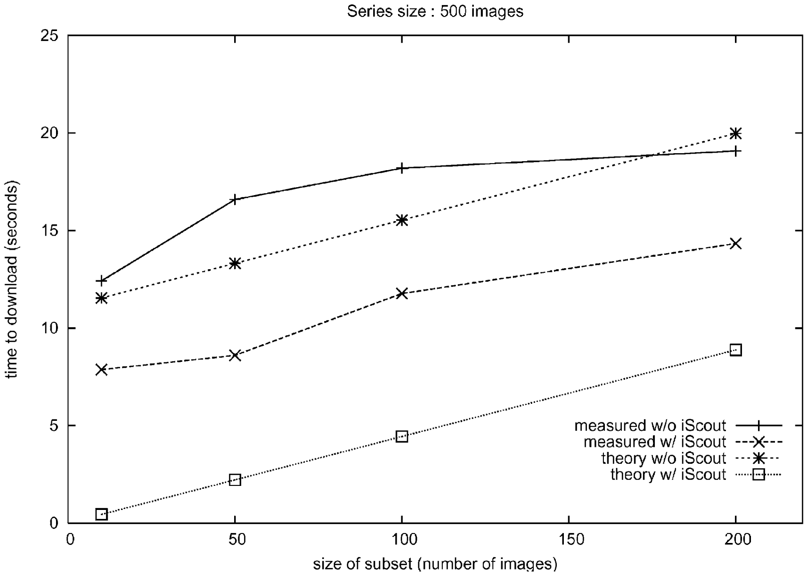


Fig 10. Statistics and predictions of the downloading model for a series of 500 images.

previous reviews are used to download the most frequently viewed images first. If the series is being viewed for the first time, then the *historical* logic has no effect on the download order.

Results of Testing and Model Comparison

Tests were performed to determine the relative efficiency of the priority downloading feature of iScout compared with conventional downloading methods in terms of the time needed to access images and groups of images that form segments of the study assumed to represent regions of interest to a radiologist. The tests were performed by downloading studies from a PACS server to a PACS workstation on an in-house 100-Mbps local area network. The results are presented for three studies of different sizes, including 118, 500, and 1000 images. Each image in the 118-image study is 512 × 512 pixels, while each image in the 500- and 1000-image studies is 256 × 256 pixels. All images were compressed with JPEG lossless compression. Figures 9–11 show the time required to download a variable-sized subset of

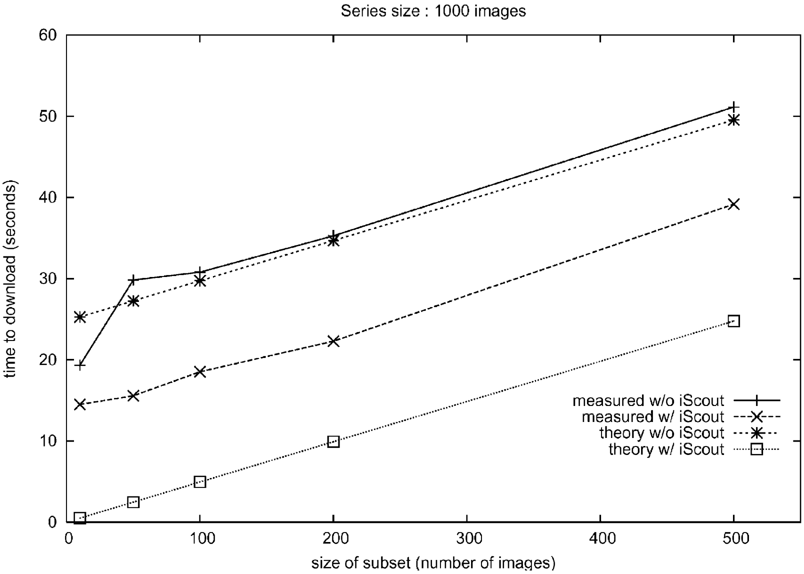


Fig 11. Statistics and predictions of the downloading model for a series of 1000 images.

images using the priority downloading feature of iScout, averaged over 30 trials, compared with the time needed for the subset of images to be received without priority downloading.

The results show that the expected download times without iScout predicted by the model are accurate for the largest study, and that good agreement is obtained when the subset size is much less than the study size. However, as expected, the model is less accurate when the subset size is close to the study size (eg, in Fig. 9, where $C = 100$ and $N = 118$, the prediction is significantly larger than the average measured time). This is expected because the model assumes that the fraction of images that are part of the subset will be small compared with the fraction of images that would need to be downloaded on average to find the first image in the subset (approximately one half of the total number of images). When C approaches N, this assumption is not valid and the predictions of the model will exceed the real download time. The results also show that the predicted time to download a subset of images using the iScout priority downloading is in error by a constant offset that increases with the size of the study. This offset is due to the initialization and initial display of the iScout and is, in general, hard to model since it depends in part on the capabilities of the workstation and the processing power available when downloading is initiated. Our simplified model ignores this offset; however, this is not a significant problem since there will be some time lag between the time the user opens the study to initiate downloading and the time the user selects a region of interest for priority downloading.

Results of Overhead Testing

Tests were also performed to determine the relative computational overhead of generating an iScout. In a normal installation, the PACS server would run the iScoutCreator tool to automatically reconstruct the iScout at the time the study is transferred from the scanner to the PACS server. The completed iScout is then added to the study. When the study is later requested for transfer to a PACS radiology workstation, the iScout is loaded first and the iScout tool can be used to manage the download process. In our current implementation, iScoutCreator runs on the workstation and as a result iScout reconstruction is performed on the workstation incrementally during the download process. Once the download to the workstation is complete, the iScout is then added to the study in the same manner as the server configuration. The downside of this approach is that while the iScout can still be used to manage the first download, the incremental reconstruction of the iScout increases the amount of time needed to download the study. This overhead is an important factor (also for server implementations) in the previous results and is measured in a number of performance tests.

Tests were performed by downloading the test studies from the PACS server to the PACS workstation with and without iScout running. When iScout is running, the iScout images are generated automatically and incrementally during the download. The results of these tests are shown in Table 1. The results show that the overhead is small for moderate series (100–500), but increases with number of images. For large series, the overhead is large enough to prefer iScout creation on the server instead of the workstation in order to hide the performance cost from the user. It is expected that the server configuration of iScout would be significantly more efficient because the server need not display the series or the iScout during construction. In either configuration, almost no significant performance penalty is paid to download an already-created iScout, since the size of an iScout is approximately equal to the size of a single image in the study.

DISCUSSION

The tests have focused on the time associated with downloading variable-sized subsets of series. This recognizes the fact that the radiologist is interested in viewing an anatomical region of interest immediately. The region of interest may vary in size, and the improved efficiency is related to the time needed to retrieve all the images of interest. The download time was measured as the time from the request of the study for download to the time when the last image of the region of interest is downloaded. In reality, there will be an additional operator-dependent lag between the display of the iScout and the selection of a region of interest for priority downloading. This lag will vary with the complexity of the anatomy being studied and the type of study being reviewed. Other factors that affect the download time include the use of compression schemes such as JPEG or fluctuations in the available bandwidth for transferring data. The use of compression should decrease the total download time, so the time savings would decrease in the same proportion. However, the relative time savings of using iScout do not decrease since iScout improves the download order, which is independent of the size of each image.

The results presented for the time savings of downloading indicate that the approximate downloading model can be used to predict potential time savings for larger studies. Specifically, the difference between the download times for a subset of images with and without iScout can be compared according to the approximate model,

$$T_{\text{savings}} = T_C - T_{C'} = T_{\text{all}}/2.$$

According to this model, the time savings should be approximately 50% of the total download time when the size of the subset is much less that the size of the entire study. For example, a 4000-image CT study with 256 × 256 pixels and 2 bytes/pixel will take 3.93 to load a 150-image region of interest using iScout compared with 56.4 s without iScout. This is assuming a network speed of 40 Megabits/s, or 40% utilization of a 100-base T network link.

If the iScout is created on the workstation, then our test results indicate that the actual time savings would be slightly less than this prediction for the first time it is downloaded. The difference is due to the overhead of downloading and displaying the iScout, as well as the lag between displaying the iScout and selection of the subset of images. Nevertheless, the use of iScout still represents a potentially large increase in efficiency. The use of the semiautomatic download logic also has the potential to mitigate the lag detected by the user. The advantage of the logic schemes is that there is less need for explicit user intervention. For example, if the iScout is configured to download in *historical–statistical* order with the most-often-viewed images first, it will result in faster access to images of interest if the same portion of the study is being reviewed. Of course, in all logic schemes, the user can override the download order by simply clicking the images to be given higher priority.

In many cases, the user wishes to review all images in a study without missing any of them. With large studies and a customized downloading scheme, it can be difficult to remember which were reviewed. The iScout tool has the potential to remind the user about images that were missed. For example, in addition to the color-coding scheme for download status, an additional color scheme could easily indicate which images were reviewed. This is another avenue for future research that may improve workflow for the user.

CONCLUSIONS

Anecdotal evidence as well as published reports indicates that the amount of image data generated by imaging modalities is growing. The newly developed iScout tool addresses the problem of large image sets by providing a flexible interface that can facilitate easy navigation of a study as well as intelligent management of the downloading of studies from PACS servers to PACS workstations. Simulations of downloading of portions of image sets on a PACS network were compared with an approximate model of the download process and good agreement was found for large studies. The model predicts that the time savings with the use of iScout for priority downloading scales up with the number of images, so it is especially useful for very large series. In general, the work aims to improve radiology workflow by providing anatomy-based image access and navigation instead of storage order or acquisition order. iScout provides these capabilities with only a small overhead. In addition, it is small and easy to use and can operate with little user interaction through the use of configurable download logic. Future work will include tests of the efficiencies of the various logic schemes and the use of on-the-fly prioritizing for a variety of PACS and users. Another pre-configured logic scheme—called *dynamic* mode—is also under development. This logic scheme uses a combination of the other basic schemes and constantly updates the logic for a given user. The eventual goal is to provide a way of predicting which images the user is likely to request next without the need to set the various options manually.

ACKNOWLEDGMENTS

The authors would like to thank the McKesson staff—including Len Grenier, Warren Edwards, Allan Noordvyk, David Branson, David Heaney, Mahmoud Ramze–Rezaee, and the Emerald team—for valuable advice on this research. Financial support from the British Columbia Advanced Systems Institute is also gratefully acknowledged.

REFERENCES

1. Erickson, BJ, Persons, KR, Hangiandreou, NJ, et al: Requirements for an Enterprise Digital Image Archive. J Digit Imaging 14:72-82, 2001

2. Minglin, L, Wilson, D, Wong, M, et al: The evolution of display technologies in PACS applications. Comput Med Imaging Graph 27:175-184, 2003

3. Dumery, B: Digital image archiving: challenges and choices. Radiol Manage 24:30-38, 2002

4. Horii, S, Nisenbaum, H, Farn, J, et al: Does use of a PACS increase the number of images per study? A case study in ultrasound. J Digit Imaging 15(1):27-33, 2002

5. Nichols S, Mehta A, Dreyer K, et al: Integrating advanced 3-D imaging into a PACS: pitfalls and solutions. PACSweb http://www.diagnosticimaging.com/pacsweb/archive/0999featl.shtml, 1999

6. Erickson, BJ, Manduca, A, Palisson, P, et al: Wavelet compression of medical images. Radiology 206:599-607, 1998

7. Hovanes, ME, Grizz–Deal, JR, Rowberg, AH: Seamless multiresolution display of portable wavelet-compressed images. J Digit Imaging 12:109-111, 1999

8. Weathermon, AC, Rohling, RN: iScout: an intelligent scout for navigating large image series. Proc SPIE 5033:319-329, 2003