This paper has appeared in *Physics in Medicine and Biology*, 58(21):7857-73, 2013

DQS Advisor: A visual interface and knowledge-based system to balance dose, quality, and reconstruction speed in iterative CT reconstruction with application to NLM-regularization

Z Zheng, E Papenhausen and K Mueller

Visual Analytic and Imaging Lab, Computer Science Department,
Stony Brook University, NY 11790, USA

E-mail: {zizhen, epapenhausen, mueller}@cs.stonybrook.edu

Abstract. Motivated by growing concerns with regards to the X-ray dose delivered to the patient, low-dose Computed Tomography (CT) has gained substantial interest in recent years. However, achieving high-quality CT reconstructions from the limited projection data collected at reduced X-ray radiation is challenging, and iterative algorithms have been shown to perform much better than conventional analytical schemes in these cases. A problem with iterative methods in general is that they require users to set many parameters, and if set incorrectly high reconstruction time and/or low image quality are likely consequences. Since the interactions among parameters can be complex and thus effective settings can be difficult to identify for a given scanning scenario, these choices are often left to a highly-experienced human expert. To help alleviate this problem, we devise a computer-based assistant for this purpose, called DQS-Advisor. It allows users to balance the three most important CT metrics – dose, quality and speed (DQS) – by ways of an intuitive visual interface. Using a known gold-standard, the system uses the ant-colony optimization algorithm to generate the most effective parameter settings for a comprehensive set of DQS configurations. A visual interface then presents the numerical outcome of this optimization, while a matrix display allows users to compare the corresponding images. The interface allows users to intuitively trade-off GPU-enabled reconstruction speed with quality and dose, while the system picks the associated parameter settings automatically. Further, once the knowledge has been generated, it can be used to correctly set the parameters for any new CT scan taken at similar scenarios.

1. Introduction

Flat panel cone-beam CT has become a major imaging technique due to its simplicity, uniform 3D resolution, and low scanning time. The traditional cone-beam CT reconstruction method is the FDK algorithm (Feldkamp et al 1984), which can provide excellent results but requires several hundreds of X-ray projections. With growing concerns about the potential risk of X-ray radiation exposure to the human body, low-dose CT has become a significant research topic (Evans et al 2011, Nuyts et al 2013, Thibault et al 2007, Sidky and Pan 2008, Jia et al 2011, Xu and Mueller 2009). Dose reduction usually involves lowering the X-ray energy per projection (i.e. low mAs) and/or reducing the total number of projections. Both methods suffer from low signal-to-noise ratio (SNR) in the reconstructed images. In the low mAs case, the images are typically noisy, while for low projection numbers the images can suffer from severe streak artifacts. Iterative reconstruction algorithms, especially when

matched with suitable regularization techniques have been shown to cope well in the adverse settings of low-dose CT. Since they are based on numerical optimization, both typically offer a diverse set of parameters that allow users to control quality and computation time. Therefore these schemes are significantly more complex to use than FDK. Given a specific imaging task, choosing the best settings for these parameters can be tedious and is often a matter of domain expertise and intuition. Since the interactions among the parameters can be complex, the knowledge on which settings to pick for a given task scenario must be acquired through many experiments which are often conducted in brute force fashion. This cumbersome process makes it difficult to recognize and then balance trade-offs that may exist among the parameters. As a result, unleashing the true potential of iterative methods for everyday clinical use is still an on-going effort.

What is currently lacking is an efficient automated procedure that can determine good parameter settings at high confidence – and furthermore, enable the use of these settings for any new reconstruction within a similar scenario. With prolonged training such a system would then store a good deal of scanning expertise and could serve as an advisor to the human operator. An early attempt in this direction was a framework which used the computational power of GPUs to quickly compute the outcome of all possible parameter combinations, at some level of discretization, and then present the optimal combination to the user (Xu and Mueller 2009, Xu et al 2010). While this work revealed interesting relationships among the parameters tested – there were two – such an exhaustive scheme is clearly not scalable in the number of parameters. In our paper we derive such a scalable framework. It uses a specific form of genetic algorithm – ant colony optimization (Dorigo et al 1999) – to navigate the high-dimensional parameter space efficiently.

1.1. General Design Goals

Image quality and radiation dose are important factors, but when it comes to clinical practice, the time it takes to arrive at a suitable reconstructed image also plays a critical role. The latter is an objective often disregarded in the literature, but it can be of great importance when CT scanning is part of a surgical procedure, in emergency departments, but also when the patient is anxiously waiting for a diagnosis. Thus altogether we are facing three co-dependent objectives – radiation dose, reconstruction quality, and computational speed (DQS). This three-tier multi-objective optimization problem has several well-known non-linear trade-offs governed by its native domain. For example, a lower radiation dose will require a higher computational effort to reach a certain reconstruction quality – but it may never reach the quality of a regular-dose reconstruction no matter how involved the computations are. This is true even for an optimal parameter configuration, and it essentially means that there are many parameter configurations that are non-competitive. This enables us to efficiently cull the search space in the optimization procedure. Essentially, we can reject any solution which, at the same quality and dose, has a lower computational speed. Furthermore, assuming similar growth patterns in quality, we can reject, for computational reasons, all solutions that at any stage of the iterative process have fallen significantly behind the others in terms of quality.

Our optimization is geared towards quality (Q) and speed (S) – dose (D) is an input factor. We design the QS optimization system especially for an algebraic reconstruction framework with edge-preserving regularization. In this research area, the non-local means (NLM) filtering (Buades et al. 2005) has been used for regularization in the past with good success (Huang et al. 2011, Zheng et al. 2011, Xu and Mueller 2012). Further encouraging is the recent work by Li et al. (2012) which makes a strong case for NLM as a possibly superior alternative to Total Variation Minimization (TVM). We have further augmented the NLM-based filtering with unsharp masking for edge enhancement. Our research advances the early work of Xu and Mueller (2011) which used genetic algorithm in conjunction with bilateral filtering (Tomasi and Manduchi 1998) for regularization.

Our QS optimization component is quite general and could also be used to find good parameter settings for other iterative reconstruction algorithms, such as EM, and other regularization techniques, such as TVM (Sidky et al 2006, Sidky and Pan 2008, Song et al 2007), soft-threshold filtering (Yu and Wang 2010), or Tight Frame (TF) regularization (Jia et al 2011). In fact, we show that our scheme

(NLM with unsharp masking) compares quite favorably with ASD-POCS (Sidky and Pan 2008). It differs from this scheme in that it does not require a special – and time consuming – control unit to guide and discover regularization parameters during the convergence. Rather, our method is fully informed by prior knowledge and so can complete its procedure along a given path deterministically without any further probing of the current state.

As mentioned, DQS advisor adds the reconstruction speed component to the more standard DQ comparison. Tang et al. (2009) performed such a DQ study on a single mAs setting and found that it is preferable to distribute total imaging dose into many view angles to reduce the streak artifacts caused by angular under-sampling. Yan et al. (2012) investigated this subject with multi-mAs settings. They plotted both quality and dose as functions of number of projections and mAs per projection. Based on this insight, protocols can then be developed to maximize the dose reduction while minimizing the loss of image quality for various imaging tasks. Our DQS optimizer enables similar insight but in the context of the time needed to produce the reconstruction.

Ant Colony Optimization (ACO), along with Genetic algorithms (GA) and Simulated Annealing (SA), belongs to the class of metaheuristic approaches. By iteratively modifying heuristics to efficiently produce high-quality solutions, metaheuristic approaches provide approximate solutions when the global optimum is infeasible to compute due to incomplete or imperfect information or limits on computational bandwidth. Blum and Roli (2003) reviewed the above methods and found that they are conceptually identical and better solutions can be found by integrating them together. We choose ACO due to its simplicity and amenability to parallelization.

2.2 The Importance of Visualization

When it comes to DQS, there are always trade-offs. And if only for educational purposes, visualizing these tradeoffs in a global context can be immensely helpful to appreciate their extent. Since we have a tertiary relationship a bivariate linear graph is insufficient to capture these relationships. In (Xu and Mueller, 2010) an interactive parameter space visualization framework for iterative CT was introduced that used icons to visualize additional variables in a 2D plot. We have chosen a different approach that requires less interaction. It plots the third variable as a backdrop layer of a 2D plot and it allows users to click on this plot to visualize an actual reconstructed image. Upon clicking, a marker is placed into the plot and the image is inserted into a comparative matrix of images. We believe that comparing actual images is ultimately the best way to appreciate quality since any error metric, RMS, CC, SSIM, CNR, and various others, never capture the full impact an image has on the human visual system. Finally, the knowledgebase through its visual front-end is exposed as a web service and can be accessed via standard web browsers. Future generations of this service will allow users to upload their own data, simulate low-dose effects, run the optimizer, and visualize the DQS.

2. Approach

Figure 1 illustrates the parameter optimization component during the training stage. The gold-standard is generated by the regular dose scan, and we perform iterative reconstruction with regularization in the low-dose acquisitions. The optimization is done per iteration in the iterative reconstruction.

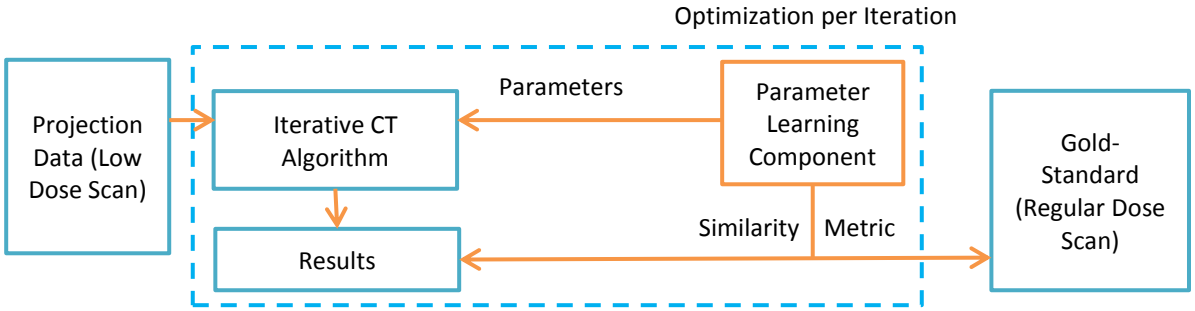


Figure 1. Parameter optimization for the training dataset.

2.1. Iterative Reconstruction with Regularization

We use the OS-SIRT algorithm with GPU-accelerated forward projection and back-projection (Xu and Mueller 2011). The forward projection operator simulates X-ray images at a certain viewing angle φ. The result of this projection is then compared to the acquired image obtained at the same viewing angle. The forward projection is computed as:

$$r_i = \sum_{l=1}^{N} w_{il} \cdot v_l \qquad i = 1,2,...,M \tag{1}$$

where M and N are the number of rays and voxels, respectively. Here, the weight factor w_{il} determines the contribution of v_l to r_i and is given by the interpolation kernel used for sampling the volume. The back-projection is computed as:

$$v_j^{(k+1)} = v_j^{(k)} + \lambda \frac{\sum_{p_i \in OS_s} \frac{p_i - r_i}{\sum_{l=1}^{N} w_{il}} w_{ij}}{\sum_{i=1}^{N} w_{ij}} \tag{2}$$

$$r_i = \sum_{l=1}^{N} w_{il} \cdot v_l^{(k)} \tag{3}$$

Here, p_i represents the pixels in the M/S acquired images that form a specific (ordered) subset OS_s where $1 \leq s \leq S$ and S is the number of subsets. The factor λ is the relaxation factor that scales the corrective update to each voxel. This factor is important in balancing quality and speed and will be optimized by our framework. The factor k is the iteration counter. It will be incremented when all M projections have been processed. In the GPU implementation, the forward projection uses a ray-driven approach, where each ray is a parallel thread and interpolates the voxels on its path. Conversely, the back-projection uses a voxel-driven approach, where each voxel is a parallel thread and interpolates the 2D correction projections. This offers the highest degree of parallelism and so the greatest GPU-speedup (Xu and Mueller 2009, Xu and Mueller 2011). While in this unmatched projector/backprojector approach the weights used in the two projectors are slightly different, it was shown that the actual effect on reconstruction quality is minimal (Zeng and Gullberg 2000).

We use a 2D NLM filter (Buades et al. 2005) for regularization. The NLM filter is a non-linear filter that replaces the pixel $P(x, y)$ with the mean of the pixels $P(x', y')$ whose Gaussian neighborhoods look similar to the neighborhood of $P(x, y)$. There are two windows: (1) the search window V, in which the algorithm looks for similar neighborhoods, and (2) the similarity window U which is essentially a Gaussian mask.

$$P_{NLM}(x, y) = \frac{\sum_{P(x',y') \in V(x,y)} w(x'-x, y'-y) P(x', y')}{\sum_{P(x',y') \in V(x,y)} w(x'-x, y'-y)} \tag{4}$$

$$w(x, y) = \exp\left(-\frac{\sum_{P(x',y') \in U(x,y)} G_a(x'-x, y'-y) \,|\, P(x', y') - P(x, y)|^2}{h^2}\right)$$

$$G_a(x, y) = \exp\left(-\frac{x^2 + y^2}{2a^2}\right)$$

$V(x, y)$ is the 2D search window centered at $P(x, y)$ and $U(x, y)$ is the 2D similarity window centered at $P(x, y)$ and $P(x', y')$, respectively. G_a is a 2D Gaussian kernel with standard deviation a, and the variable h acts as a parameter to control the smoothing. Thus, the NLM filter contains several parameters that can be tuned to achieve the best quality. With regards to performance, with NLM filtering being non-iterative, the computational performance is stable, unlike TVM regularization. We have used the scheme described in (Zheng et al. 2011) to make NLM filtering efficient on the GPU.

The last stage of our regularization is an unsharp masking step (Suetens, 2009) which corrects for the edge blurring of the iterative reconstruction. It also has a parameter, α that determines the added sharpness, i.e., the amount of high frequency detail that is added to a blurred version of the image:

$$I_s = g_\sigma * I + (1+\alpha)(I - g_\sigma * I) \quad (5)$$

Here, I_s is the sharpened image, I is the original image and g_σ is a Gaussian blurring kernel with σ=3.0. The symbol $*$ is the convolution operator. Figure 2 summarizes our iterative reconstruction pipeline and the parameters we optimize.

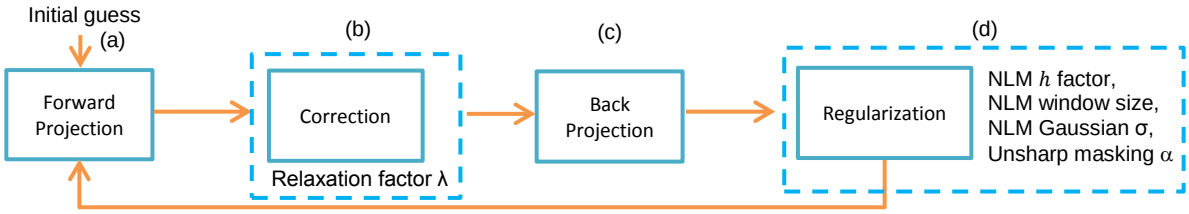


Figure 2. Iterative reconstruction and parameters. (a) simulate X-ray projections from present reocnstruction, (b) compute difference between real projection and estimated projections multiplied by λ, (c) add differences to present reconstruction, (d) regularize updated reconstruction.

2.2. Ant Colony Optimization (ACO)

We model iterative CT reconstruction as a path searching problem. In this graph of nodes and edges, each node represents a unique state of the volume that is being reconstructed, while each edge represents the computation of a correction pass and a regularization pass at a given setting of parameters. The edge weight represents the time cost, which in our case can be uniformly set to 1 since all passes have a fixed constant cost. Each node is tagged with a score that encodes a quality metric. The overall goal is to find the node that has the highest score at a given cost. Since all edges have the same weight, the problem is reduced to find the best score after a fixed number of steps.

The ant colony optimization (ACO) algorithm (Dorigo et al 1999) is a swarm intelligence method to search for good paths in discrete graphs. Intuitively, it launches a large number of artificial ants searching for the best score. In our case, each artificial ant independently moves through the regularized iterative CT reconstruction graph and receives a score based on image quality, reconstruction time. Similar to the natural world, ants with good scores have their paths reinforced with *pheromones* to attract other ants. The probability for an ant to choose an edge connecting two nodes is increased when the amount of pheromone deposited there is high, that is, when a large number of high scoring ants have traversed it.

We optimize six parameters including the relaxation factor λ in Equation (2), the *h* factor, Gaussian blur factor, window size and mask size for the NLM filter in equation (4) and the un-sharp masking factor α. These parameters can be different per iteration, resulting in an astronomical search space. For example, assuming we allow 100 discretized values for each parameter and run the pipeline for 10 iterations, the search space will be 10^{120}. This search space is so huge that simple exhaustive search algorithms will fail to find the optimal solution in a reasonable amount of time, hence the need for a metaheuristic algorithm such as ACO.

We adopt a greedy heuristic to prune the search tree. This heuristic guides ants with a "best guess" for the path on which the solution lies. This greedy ant system only searches the best parameter setting for a single iteration, then adopts this best setting and moves on to the next iteration. In this way, the solution space can be reduced from 10^{120} to 10^{13}. Figure 3 shows the pseudo-code for this scheme. For each iteration of iterative reconstruction (line 2-19), our system creates generations of artificial ants to optimize the parameter values used at that iteration. The input for each generation (line 6-9) is the previous CT iterations result, which is initialized to zero (line 0) and updated per CT iteration (line 14). Line 6-7 is the most time consuming part. In that stage, each ant first obtains its parameter settings probabilistically using the current pheromone deposits, and then uses these settings to perform one pass of forward projection, correction, back-projection and regularization. Then the system stores a quality score for each ant. Using these scores we then update the pheromone values for the

associated parameter settings (line 11). Line 12 is the convergence conditions, augmented by line 13 that makes sure that the current iteration has a better score than the previous iteration – else we launch another generation of ants until both conditions being met.

```
0   Initialize prev-iteration-best-result, current-generation-best-result and
pheromone to zero
1       FOR each reconstruction step in iterative reconstruction
2            SET GENERATION_COUNT = 1
3            WHILE GENERATION_COUNT <= MAX_GENERATION
4               FOR each ant in the current generation with size ANT_GN
5                      Use prev-iteration-best-result as input
6                           Obtain a set of randomized parameters using pheromone
7                      Perform one iteration of iterative reconstruction
8                      Record the quality scores using quality metric and gold
standard
9                      Update current-generation-best-result
10              ENDFOR
11              Update pheromone
12               IF ABS(current-generation-best-result - prev-generation-best-result)
< ε
13                 AND current-generation-best-result > prev-iteration-best-result
14                     SET prev-iteration-best-result = current-generation-best-
result
15                     GOTO LINE 19
16              ENDIF
17              SET GENERATION_COUNT = GENERATION_COUNT +1
18              SET prev-generation-best-result = current-generation-best-result
19           ENDWHILE
20      ENDFOR
```

Figure 3. The pseudo code for ant colony optimization.

Figure 4 uses a parallel coordinate visualization (Inselberg 1985) to show a few sets of parameter value settings. An ant's choice is a 6-tuple shown as a piecewise linear spline – a *polyline*. The ant colony algorithm launches many polylines (value tuples) which eventually converge into a single polyline through pheromone control. The pheromone for the current generation is updated for the next generation's use, making ants more likely to choose parameter settings similar to the previous best settings. In the following, we describe the pheromone's role in detail. The probability for an ant to choose a discretized value (setting) j for the ith parameter is:

$$P_{i,j} = \frac{\tau_{i,j}}{\sum_{q=0}^{R_i} \tau_{i,q}} \qquad (6)$$

where $\tau_{i,j}$ is the pheromone on value j for the ith parameter and R_i is the discrete resolution of the ith parameter. The value of $\tau_{i,j}$ is initially set to one; this will allow ants to make purely random decisions. After a small group of ants (in our case it is 50) finishes their moves for all 6 parameters, the pheromone for each parameter value is updated as:

$$\tau_{i,j}^{(m+1)} = (1-\rho)\tau_{i,j}^{(m)} + \bar{s}_{i,j} \qquad (7)$$

where $s_{i,j}$ is the normalized score (with $0< s_k <1$) of an ant to choose the value j for the ith parameter; $\bar{s}_{i,j}$ is the average score of all ants that choose the value j for the ith parameter; $\tau_{i,j}^{(m)}$ is the pheromone

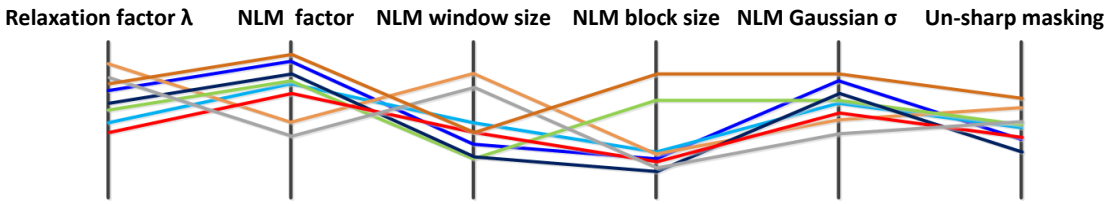
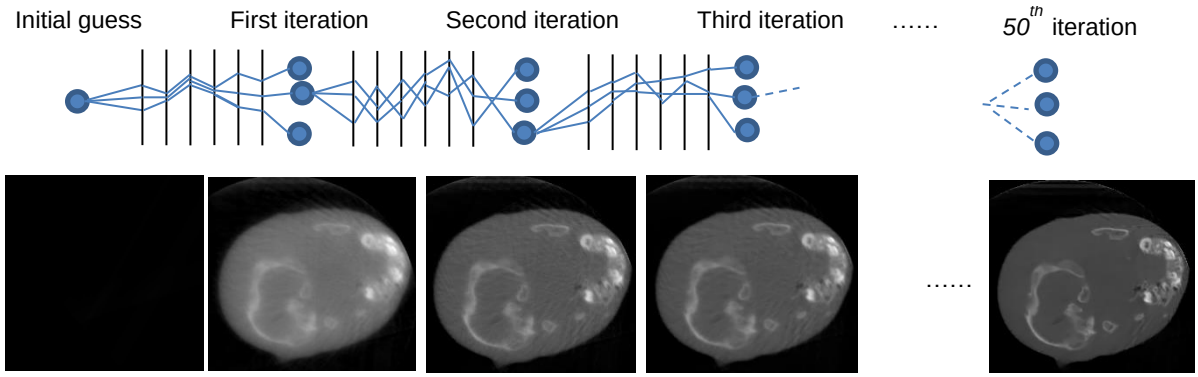


Figure 4. Different settings of parameters shown in a parallel coordinate visualization.

on value j for the ith parameter for the mth generation of ants; ρ is the pheromone evaporation factor (with $0< \rho <1$). The range of pheromone is clamped within [0,1]. Equation (7) updates the pheromone such that later ants will be more likely to follow the path of previous high-scoring ants.

Figure 5 illustrates the progression of our scheme. The top row shows the parameter settings of the various ants in parallel coordinate representation, the middle row shows the best ant's reconstruction, while the bottom row shows the parameter settings of the best ant's solution -- all at a given iteration.



	1st iteration	2nd iteration	3rd iteration		50th iteration
NLM Gaussian	2.1355	1.1005	2.1655	…	2.7115
NLM h factor	4.241935	4.319805	4.319805	…	0.677885
NLM window size	4	4	2	…	3
Lambda	1.979001	1.919004	1.819009	…	0.237053
Unsharp masking α	0.173913	0.131334	0.16092	…	0.138731
Quality (CC)	0.98446	0.990733	0.992593	…	0.9952

Figure 5. Ant colony optimization, searching for the best parameter setting per iteration.

2.3. Quality Metric

In the absence of a human observer, s_k in equation (7) is generated by a computer based on a quality metric. The regular dose FDK image is used as a gold-standard. We tested four different quality metrics: RMS error, Correlation Coefficient (CC), Edge Correlation Coefficient (ECC) (Xu and Mueller, 2009) and Structural Similarity (SSIM) (Wang et al. 2004). We find that CC and ECC can effectively guide ACO to converge, while RMS and SSIM do not perform well to their sensitivity to small DC offsets. We decided to use CC to guide the parameter optimization since it is more commonly used in the literature.

2.4. Visualization Design

There are four dimensions that represent one reconstruction: X-ray current (mAs), number of projections, quality and time. For ease of comprehension we would like our visualization system to present these 4D data in a single picture. For this reason we created an auxiliary variable, dose = X-ray current × number of projections, to reduce this 4D space to 3D. See figure 6 which visualizes a multi-mAs experiment. Figure 6(a) uses a scatter plot – each point is due to one experiment – where dose, quality and computation time are mapped into the three spatial axes, respectively. Conversely, figure 6(b) visualizes the data via an interpolated surface – a continuous scatterplot – where trends are easier to see. On this surface, a rainbow color map is applied to code computation time.

In these plots we observe that the data points cluster around several vertical lines. This occurs for two reasons. First, since our iterative reconstruction optimization engine runs on an iteration-basis, the data points show the optimized quality and speed of the iterative reconstruction for each separate iteration. Thus, as the iteration number grows, both quality and time change but the radiation dose does not since the reconstructions are still based on the same acquisition data. Thus we see a series of points that change in the y-coordinate (and color) but not in the x-coordinate. A second reason is that

since the X-ray tube current and projection number only have discrete values available on the CT

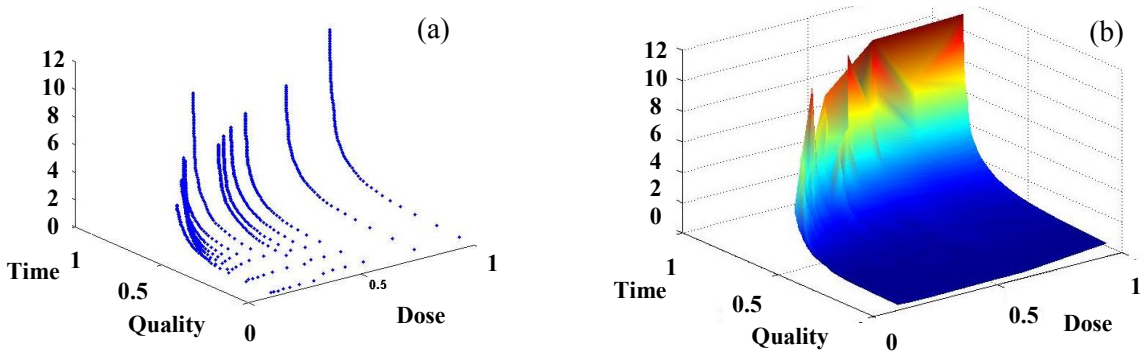


Figure 6. The input data visualized in 3D (a) scatter plot, (b) interpolated "continous" scatterplot with time mapped to rainbow color.

scanner, the data points will group into several discrete value clusters along the dose axis.

Since 3D plots are difficult to navigate in practice, we derive two 2D plots from Figure 6(b) via orthogonal projection: (1) a quality vs. dose plot with time mapped to color, and (2) a time vs. dose plot with quality mapped to color. A conflict occurs when two data points are projected into the same 2D location. In this case, with the remaining fields being equal, the preference will go to the point with better speed (or higher quality). During the graphics rendering this can be easily achieved using OPENGL depth-culling, where we can flip the depth testing direction by setting GL_GREATER or GL_LESS. This yields a set of two scatterplots which we then triangulate and visualize using smooth color shading. Next, we use the quality vs. dose mode as an example to describe our user interface.

2.5. User Interface

The interface of our DQS Advisor framework is composed of the main plot, a few control sliders and a set of reconstructed images. Figure 7 shows an example for this interface. We will mainly discuss its components in this current section – the reader is referred to Section 3.2 to learn about the

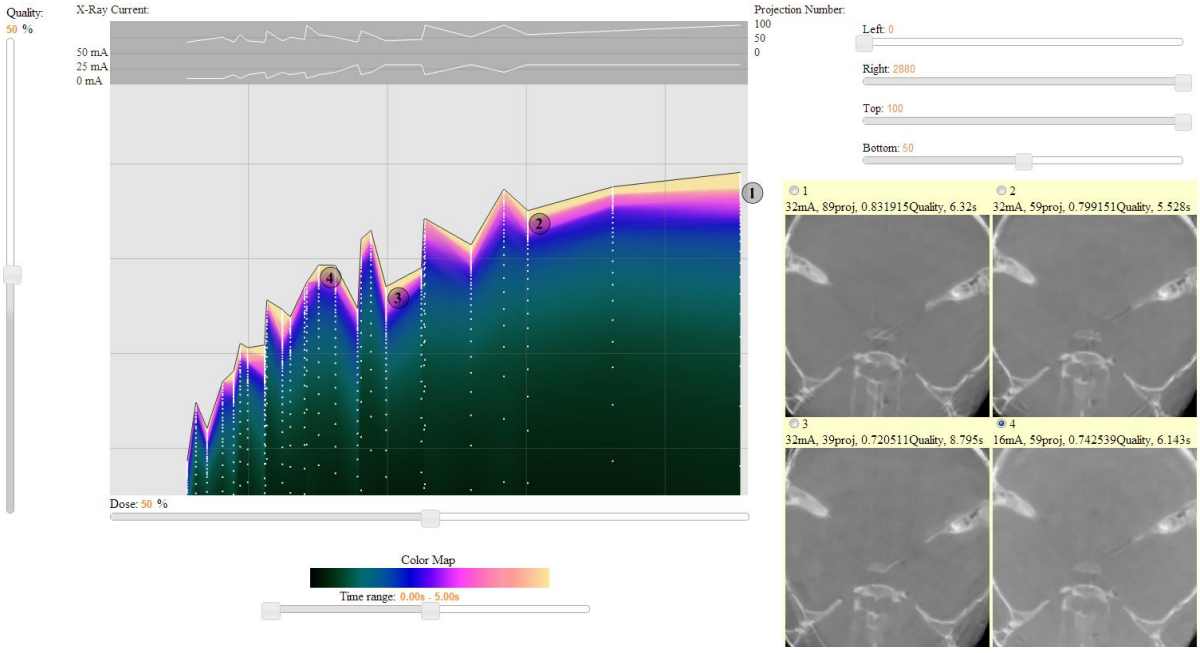


Figure 7. The quality vs. dose plot. The numbers inserted into the plot refer to the image matrix (the *Light Box*) on the right. Image 1 is the image of the highest quality but requires a high dose. Images 3 and 4 require much less dose but have streak artifacts or blurred features, respectively. Image 2 seems to be a good compromise – lower dose than image 1 but still offering good quality. It could serve as a starting point for further exploration of the plot, where the user would mouse-click at desirable plot locations and insert the corresponding images into the Light Box. (see Section 3.2 for more detail).

particular imaging scenario studied and the findings that were derived. The main window of the interface shows the 2D scatter plot of dose vs. quality with color mapped to computation time (or dose vs. time with color mapped to quality). The background color of a pixel is derived by interpolation from the nearby tuples' time (or quality) field and the global colormap can be adjusted by means of a range slider at the bottom. There are two sliders along the X and Y axis, respectively, which can be jointly used to display a highlighted view of a certain region. The two graphs on the top of the main window visualize dose composition – mAs and number of projections. The four sliders on the right can be used to specify a certain region of the plot to be visualized. Finally, at the bottom-right there is what we call the 'light box' – four image slots by which users can place noteworthy reconstruction results for comparative studies. The locations/origins of these images in the main plot are indicated by circled numbers. We now discuss the various features and supported user interactions in closer detail.

Main plot: The X axis shows the dose as a product of mA and number of projections while the Y axis shows the measured quality on an exponential scale. The actual data points are displayed as white points on the interpolated map. The user has the option to enable or disable the data points.

Adjustable colormap: The colorrnap and associated range slider are placed right below the main plot (see figure 7). The color spectrum goes from black on the left to yellow on the right, and the range slider can be used to dynamically adjust the upper/lower values to be mapped to black/yellow. For example, figure 8(a) shows the default color-coded plot and figure 8(b) shows the updated plot after adjusting the slider's upper limit. This functionality enables users to zoom into regions of interest.

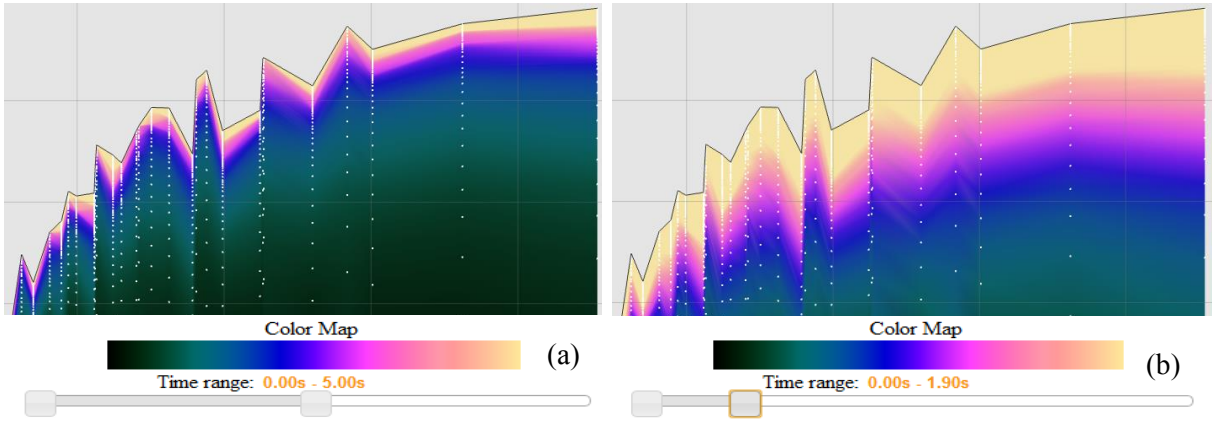


Figure 8. Changing the colormap.

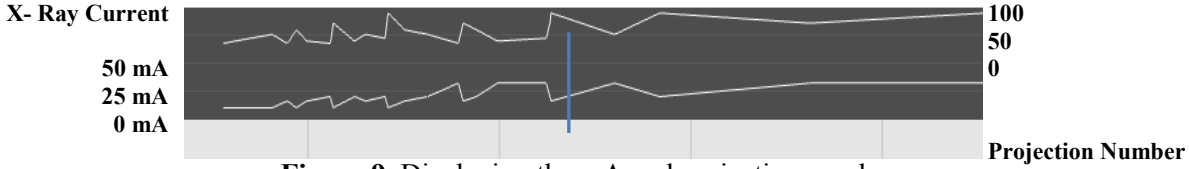


Figure 9. Displaying the mA and projection number.

Dose composition plot: Abstracting mA × number of projections into dose enabled us to compress two dimensions into one scalar value, mapped to the X axis. To convey insight into the original two variables, we provide two graphs on top of the main window – one for mA and one for number of projections. This allows users to appreciate the reason for the dose. For example, in figure 9 the dose level indicated by the blue vertical line is imposed from acquiring 89 projections at 16 mA each.

Region of interest (ROI): By modifying the left/right/top/bottom boundary of the main plot, users can pan or zoom in/out to inspect the plot, helping them to explore the sparse space. The zoom in/out is a linear function that preserves the neighborhood without distortion. Figure 10 shows two detailed plots after adjusting the ROI. In panel (a) a small region containing 80+ points is displayed in a magnified manner – the detail can be seen well. Panel (b) shows another ROI but now with the data points disabled – the background color can be better visualized without the data points.

Image Light Box: Users can mark up to four points in the main window and the corresponding images will be displayed in the light box on the right for side-by-side comparison, along with the data information. Whenever the user left-clicks the mouse on a plot location, the system will find the nearest data point around the cursor. This nearest neighbor search is accelerated by ways of an R-tree structure (discussed in Section 2.6). A number icon will be placed on the clicked location to provide a location reference. In figure 11, four images were put side by side for comparison, noting various settings on their tops. For example, the first image was reconstructed from 32mA and 89 projections with quality score 0.8319 and a computation time of 6.32 seconds. The last image was reconstructed from 16mA and 59 projections with quality score 0.7425 and speed 6.14 seconds. In this way, users can understand image quality much better and more comprehensively than with plain numbers.

Highlighting slider: This facility enables users to highlight a certain 2D neighbourhood utilizing a vertical slider and a horizontal slider. Other non-highlighted regions will be gradually color-blended.

Right click menu: Right-clicking the mouse pops-up a menu by which users can set various preferences on some of the display modes. Users can also display/hide the data points, horizontal lines, vertical lines and the highlighting slider, and they can switch the plot between color and grayscale.

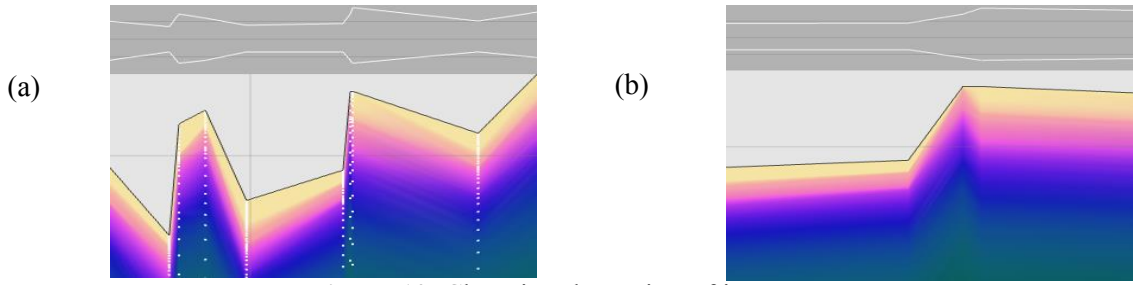


Figure 10. Changing the region of interest.

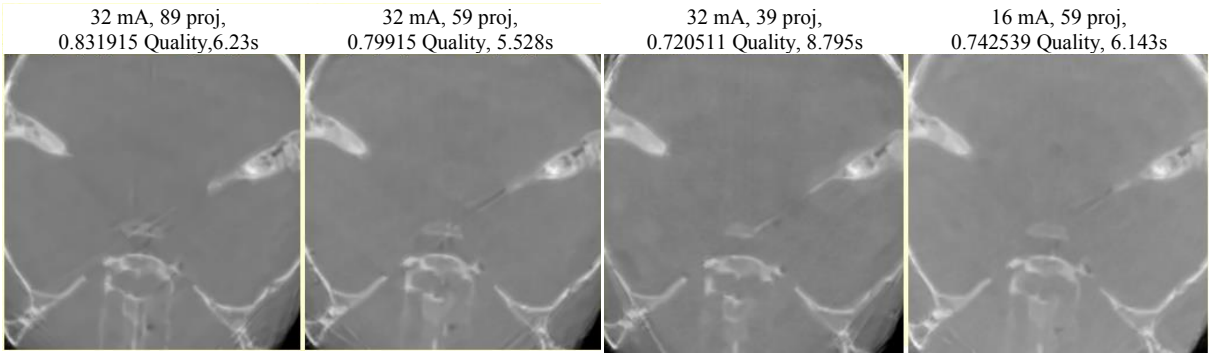


Figure 11. A side-by-side images comparison.

2.6. Public Access and Scalability

Our visualization tool is hosted on a web server which can be accessed publicly at http://vail.cewit.stonybrook.edu/Projects/CTPlot/. Written in WebGL and JavaScript, it requires a WebGL-enabled browser, such as Chrome and Firefox. We designed the system to be capable of interactively displaying relatively large data sets. A typical desktop browser can interactively manage 2.2k data points and 2.2k 512×512 images. The input data consists of 4-tuples. The four fields of the tuples are ordered as mA, projection number, quality and time. The tuples are stored in a CSV file, and the images associated with the data points are stored in a folder. The name format of the images is the same as the 4-tuple. We use the JPEG format to store the compressed images to minimize storage and data transfer latency. A naïve search algorithm will take linear search time (complexity $O(n)$ for n data points) which is impractical for large datasets. In order to facilitate interactive 2D search we use a non-recursive R-tree. The R-tree's nearest neighbour search time has complexity $O(log\ n)$.

3. Experiments

3.1. Optimization validation

In our first set of experiments we collected two scans of a head phantom using a cone-beam CT scanner (Medtronic O-arm system). We performed the reconstruction algorithms on the central slice only. The detector 1D resolution was 1,024 pixels with pixel-size 0.388 mm. The low-dose case was set to 72 evenly-distributed projections and the gold-standard was generated by the FDK algorithm with 360 projections. To set up our OS-SIRT scheme, we formed 20 subsets with each subset containing 3-4 projections. The reconstruction resolution was 512×512. Figure 12 presents some of the results we obtained. The reconstruction after 50 iterations of OS-SIRT with a constant λ=1 is shown in panel (a). It has severe streak artifacts. The gold-standard is shown in panel (b). Panel (c) presents the reconstruction obtained after 50 iterations of ASD-POCS – we used OS-SIRT instead of ART in this ASD-POCS implementation to make a fair comparison. We observe that the ASD-POCS result is smooth but has some typical cartoon-like structures due to the TVM scheme that it uses. The reconstruction we obtained with our optimization framework is shown in panel (d). It appears more similar to the gold standard in panel (b) and seems to preserve sharper details better than ASD-POCS.

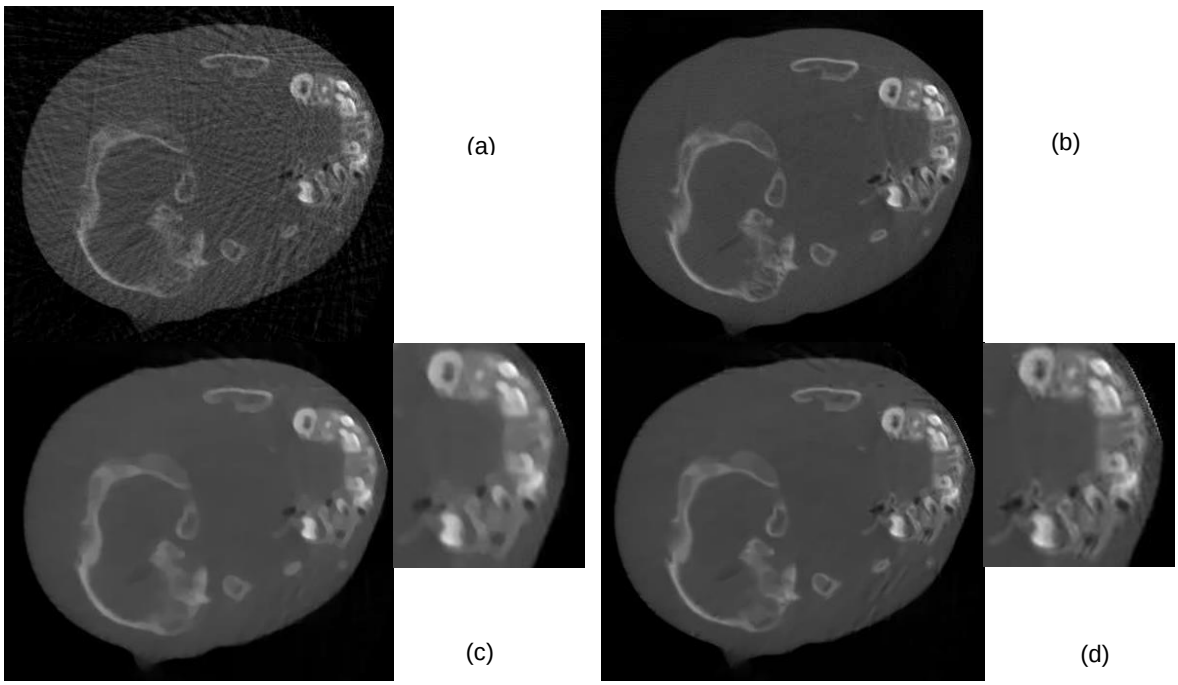


Figure 12. A central slice of a reconstruction for a training dataset. (a) low-dose OS-SIRT. (b) gold-standard FDK by 360 projections. (c) low-dose ASD-POCS and (d) optimized low-dose. (a), (c) and (d) use 60 projections and 60 iterations.

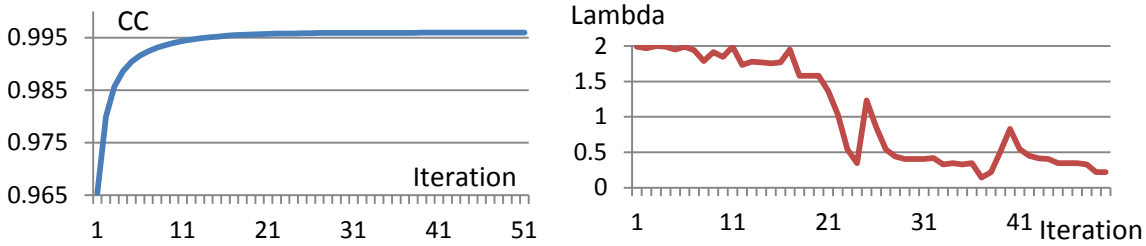


Figure 13. The CC-based quality score (left) and λ in Equation (2) (right) through 50 iterations

We applied the CC metric to the central 256×256 size region to emphasize the region of interest. Figure 13 (left) shows the CC score over 50 iterations – the CC only improved very little after 40 iterations and stopped improving after 50 iterations. Figure 13 (right) plots the relaxation factor λ. We observe the optimized parameter schedule starts with a rather large value of around 2.0 and then decreases gradually to 0.1 and the overall trend is not perfectly linear but contains some perturbations.

We then tested if the learned parameter settings can be used to reconstruct data from a different scan and anatomy. For this purpose the central slice was shifted to another autonomy region in the head phantom. Figure 14 shows the image quality of this new dataset with (a) 50 iterations of OS-SIRT with a constant λ=1, (b) 360 projection FDK, (c) 50 iterations of ASD-POCS and (d) 50 iterations with optimized parameters. We observe that even in this new scan, the parameters can guide the reconstruction to achieve better detail reproduction than ASD-POCS. Some streaks remain for both ASD-POCS and our approach and the next section (section 3.2) offers some further insight on this.

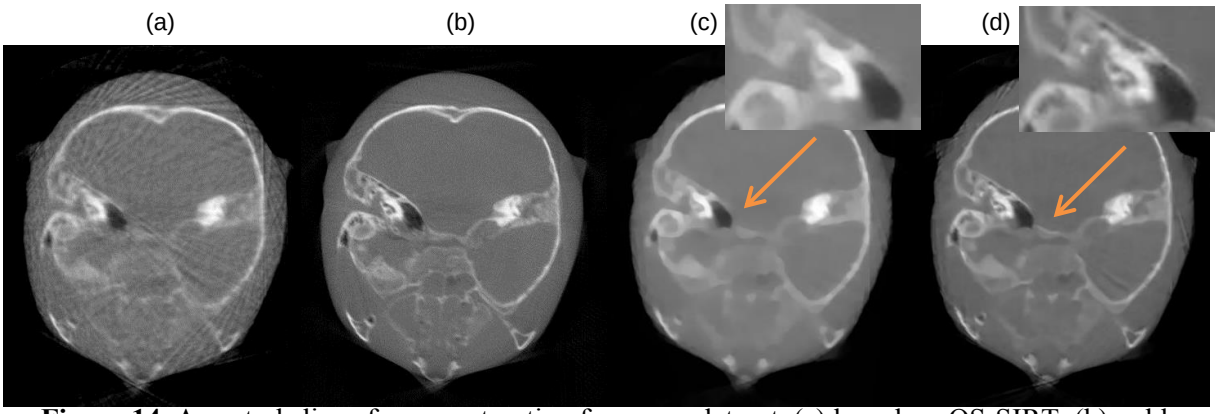


Figure 14. A central slice of a reconstruction for a new dataset. (a) low-dose OS-SIRT. (b) gold-standard FDK by 360 projections. (c) low-dose ASD-POCS and (d) optimized low-dose. (a), (c) and (d) use 72 projections and 50 iterations.

3.2. Case study: a head scan

To demonstrate how one would identify a desirable parameter configuration with the help of DQS Advisor, we have chosen a head scan as an example. The first step in this process is the knowledge acquisition phase – of course this only needs to be done once per scenario. To collect the domain knowledge for this example scenario we acquired four scans of a head phantom using a Medtronic O-arm CT scanner. The scans were conducted at four representative mA levels – 10, 16, 20 and 32 mA – each at 120 kV. The exposure time for each projection was 10ms, and hence the mAs value for a single projection is the mA setting × 0.01. With each mA setting, we selected 7 evenly distributed projection sets, each with a different angular spacing – 4, 5, 6, 7, 8, 9 and 10 degrees – which produced sets of 90 to 36 projections. The subset number for OS-SIRT was held constant at 20 for all reconstructions. We optimized for the reconstruction of the central slice without loss of generality.

Figure 7 shows the interactive visual interface that exposes the acquired knowledge to the users – its components have been discussed in Section 2.5. Just looking at the main plot we can make a few quick observations. We observe the well-known fact that very low doses (x-axis) cannot yield reconstructions of good quality (y-axis). This is visualized by the plot's declining upper fringe with decreasing dose. But we also observe that a certain amount of computational effort (for example, the blue color band) yields less and less quality as the dose decreases. In other words, computations do not get cheaper with decreased dose – there will just be more iterations and filtering to obtain the best possible result. Also, the margin of benefit that can be attained by increased computation narrows with decreasing dose – the blue band transitions quickly to yellow towards the fringe. So essentially, as the dose falls off, even heavy computation will not increase reconstruction quality significantly.

Our next focus is the dose composition plot above the main plot. We can easily identify the positive correlation that exists between the number of projections and maximum quality. We recognize that upward spikes in the main plot's upper fringe are typically related to larger numbers of acquired projections (and lower mA). Conversely, downward drops are related to smaller numbers of projections (but higher mA). This implies that reducing the number of projections harms image quality more than turning down the mA, which is an observation conformant with the recent findings of Yan et al (2012). Therefore, the user of DQS Advisor would be informed that in order to lower the dose it will be better to reduce the tube current than reducing the number of projections. And, in addition, the user would

also know how many projections and mA are needed to produce a desired quality score and how long he/she would have to wait for it.

The light box display on the right of the main plot adds a qualitative perspective to this selection. In our example, the user marked four probe points in the main plot, labeled 1 through 4. The system retrieves the corresponding images and places them into the light box. By comparing image 3 with image 4 the user quickly sees the reasons for the lower quality of image 3 – slight streak artifacts which for example distort the anatomical feature just below the image center. The user also learns that image 4, while acquired at low dose, is too blurry to be useful for his imaging goals, using (high-dose) image 1 as a reference. So he begins with image 2 and marches leftward on the plot's fringe (in the color band that has the acceptable computation time) to locate the just acceptable quality. He then looks up the associated dose. Then, once this DQS triple is determined, the scan is executed and the system looks up the associated system parameter settings to perform the reconstruction in the appropriate protocol.

4. Conclusions

We have devised an interactive framework we call DQS (Dose Quality Speed) Advisor that can guide users in the often daunting task of creating desirable iterative reconstruction protocols. It uses an ant colony optimization algorithm to learn the best settings for a set of standard iterative CT parameters, for a wide range of DQS configurations. Our preliminary results indicate that these learned parameter settings can then be readily applied to similar scanning scenarios, which provides the pay-off for the more involved optimization phase. Further, by importing the learned knowledge into DQS Advisor's visual interactive interface, users can easily recognize and balance trade-offs in dose, quality and computation time and so are able to make well-informed choices for these parameters. We believe that DQS Advisor, apart from practical clinical applications, could also find use in education and training.

In future work, we would like to also include the number of subsets into the optimization framework, akin to Xu and Mueller (2009). Work is also underway to extend the current framework to additional vital parameters such as patient size, anatomy, X-ray tube voltage, and so on. With regards to optimization, according to Blum and Roli (2003) additional benefits may result from using a hybrid version of GA, SA, and ACO, and instead of just using the best ant to seed the next iteration we could use the best k ants. Furthermore, our parameter search is based on an automated quality observer in place of a human observer. We could improve the automated quality observer by studying the link between quality metric and clinical need. Perceptual quality metrics are good candidates for this purpose. Ultimately, the goal is to develop an advanced quality metric which would pass a doctor's validation. Here, the Channelized Hotelling Observer (CHO) is only one type of automated observer, mainly dedicated to detection tasks.

Finally, our current parameter visualization interface is designed to be available after parameter search, but we might also invite user interaction during the optimization stage. This more sophisticated interface would then be an online monitoring tool. It could enable one to visualize the search process while the computation is proceeding in the background, accelerated by GPUs. It might also open the door for dynamic optimization functionalities. In fact, when embedded into a cloud computing infrastructure, medical imaging researchers could then use our DQS Advisor engine and visual interface to determine the parameter landscape for their own imaging scenario.

Acknowledgements

This work was supported by NSF grant 1117132, the Korean Ministry of Science, ICT and Future Planning under the ITCC Program by NIPA. We also thank Medtronic for research funding and data.

References

Blum C, and Roli A 2003 Metaheuristics in combinatorial optimization: overview and conceptual comparison *ACM Computing Surveys* **35** 268-308

Buades A, Coll B and Morel J M 2005 A non-local algorithm for image denoising *Proc. Computer Vision and Pattern Recognition* 60-5

Dorigo M, Caro G D and Gambardella L M 1999 Ant algorithms for discrete optimization *Artificial Life* **5** 137-72

Evans J D, Politte D G, Whiting B R, O'Sullivan J A and Williamson J F 2011 Noise-resolution tradeoffs in x-ray CT imaging: a comparison of penalized alternating minimization and filtered backprojection algorithms *Med. Phys.* **38** 1444-58

Feldkamp L, Davis L and Kress J 1984 Practical cone-beam algorithm *J. Opt. Soc. Am. A* **1** 612-9

Huang J, et al. 2011 Sparse angular CT reconstruction using non-local means based iterative-correction POCS *Comput. Biol. Med.* **41** 195-205

Inselberg A 1985 The plane with parallel coordinates *The Visual Computer* 1(2): 69-91

Jia X, Dong B, Lou Y F and Jiang S B 2011 GPU-based iterative cone-beam CT reconstruction using tightframe regularization *Phys. Med. Biol.* **56** 3787-807

Li Z, Yu L, Trzasko J, Fletcher J, McCollough C and Manduca A 2012 Adaptive nonlocal means filtering based on local noise level for CT denoising *Proc. SPIE* **8313** 83131H

Nuyts J , De Man B, Fessler J, Zbijewski W and Beekman F J 2013 Modelling the physics in the iterative reconstruction for transmission computed tomography *Phys. Med. Biol.* **58** R63-96

Sidky E Y, Kao C M and Pan X 2006 Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT *J. X-ray Sci. Technol.* **14** 119-39

Sidky E Y and Pan X 2008 Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization *Phys. Med. Biol.* **53** 4777-807

Song J, Liu Q H, Johnson G A and Badea C T 2007 Sparseness prior based iterative image reconstruction for retrospectively gated cardiac micro-CT *Med. Phys.* **34** 4476-83

Suetens P 2009 *Fundamentals of Medical Imaging*, Cambridge University Press

Tang J, Nett B E and Chen G H 2009 Performance comparison between total variation (TV)-based compressed sensing and statistical iterative reconstruction algorithms *Phys. Med. Biol.* **54** 5781-804

Thibault J B, Sauer K D, Bouman C A and Hsieh J A 2007 A three dimensional statistical approach to improved image quality for multi-slice helical CT *Med. Phys.* **34** 4526–44

Tomasi C and Manduchi R 1998 Bilateral filtering for gray and color images *Proc. Int. Conf. on Computer Vision* 839-46

Wang Z, Bovik A, Sheikh H and Simoncelli E 2004 Image quality assessment: From error visibility to structural similarity *IEEE Transactions on Image Processing*, 13(4):600-612.

Xu F, Xu W, Jones M, Keszthelyi B, Sedat J, Agard D and Mueller K 2010 On the efficiency of iterative ordered subset reconstruction algorithms for acceleration on GPUs *Computer Methods and Programs in Biomedicine*, **98** 261-270

Xu W and Mueller K 2010 Parameter space visualizer: an interactive parameter selection interface for iterative CT reconstruction algorithms *Proc. SPIE* **7625** 76251Q

Xu W and Mueller K 2012 Efficient Low-Dose CT artifact mitigation using an artifact-matched prior scan *Med. Phys.* **39** 4748-60

Xu W and Mueller K 2009 A performance-driven study of regularization methods for GPU-accelerated iterative CT *Proc. Fully 3D* 20-3

Xu W and Mueller K 2011 Using GPUs to learn effective parameter settings for GPU-accelerated iterative CT reconstruction algorithms *GPU Computing Gems Emerald Edition* 693-708

Xu W and Mueller K 2009 Learning Efficient Parameter Settings for Iterative CT Reconstruction Algorithms Fully3D 251-254

Yan H, Cervino L, Jia X and Jiang S 2012 A comprehensive study on the relationship between image quality and imaging dose in low-dose cone beam CT *Phys. Med. Biol.* **57** 2063-80

Yu H and Wang G 2010 A soft-threshold filtering approach for reconstruction from a limited number of projections *Phys. Med. Biol.* **55** 3905-16

Zeng G and Gullberg G 2000 Unmatched projector/backprojector pairs in an iterative reconstruction algorithm *IEEE Trans. Med. Imaging* **19** 548-55

Zheng Z, Xu W and Mueller K 2011 Performance tuning for CUDA-accelerated neighborhood denoising filters *Proc. Workshop on High Performance Image Recon.* 52-5