Optimal Scheduling of Tracing Computations for Real-time Vascular Landmark Extraction from Retinal Fundus Images

Hong Shen[1], Badrinath Roysam[1],Charles V. Stewart[1], James N. Turner[1,2], and Howard L. Tanenbaum[3]

[1]*Rensselaer Polytechnic Institute, Troy, NY 12180-3590.*

[2]*The Wadsworth Center for Laboratories and Research, NY State Dept. of Health, Albany, NY 12201-0509.*

[3]*The Center for Sight, Albany, NY 12204.*

Abstract

Recently, this group published fast algorithms for automatic tracing (vectorization) of the vasculature in live retinal angiograms, and for the extraction of visual landmarks formed by vascular bifurcations and crossings. These landmarks are used for feature-based image matching for controlling a computer-assisted laser retinal surgery instrument currently under development. This paper describes methods to schedule the vascular tracing computations to maximize the rate of growth of quality of the partial tracing results within a frame cycle. There are two main advantages. First, progressive image matching from partially extracted landmark sets can be faster, and provide an earlier indication of matching failure. Second, the likelihood of successful image matching is greatly improved since the extracted landmarks are of the highest quality for the given computational budget.

The scheduling method is based on quantitative measures for the computational work and the quality of landmarks. A coarse grid-based analysis of the image is used to generate seed points for the tracing computations, along with estimates of local edge strengths, orientations, and vessel thickness. These estimates are used to define criteria for real-time preemptive scheduling of the tracing computations. It is shown that the optimal schedule can only be achieved in perfect hindsight, and is thus unrealizable. This leads to scheduling heuristics that approximate the behavior of the optimal algorithm. One such approximation produced about a 400% improvement in the quality of the partial results, as compared to random scheduling. The resulting algorithm can be readily implemented on conventional and multiple-processor (MIMD) systems, and is being applied to computer-assisted laser retinal surgery.

Correspondence: Dr. Badrinath Roysam, ECSE Department, Rensselaer Polytechnic Institute, Troy, New York 12180-3590, USA. Phone: 518-276-8067, Fax: 518-276-6261, Email: roysam@ecse.rpi.edu

Printed: 11/4/99 11:33 AM

1. Introduction

This work addresses problems arising in the real-time extraction of landmarks (crossing and branching locations of the vasculature) from image sequences of the human retina [1]. This is done in the context of computer-assisted instrumentation for laser retinal surgery [2,3,4,5]. Fig. 1a shows a sample image frame, and Fig. 1e shows the vascular landmarks (crossovers and bifurcations) for this sample frame. These landmarks are matched for image registration in the context of spatial mapping and real-time spatial reckoning [6,3,7] relative to a previously constructed mosaic map of the entire retina [8]. This is an instance of a "hard" real-time system [9] in which the computations must be completed prior to a deadline, else the system is considered to have failed. A failure represents a loss of tracking, requiring the surgical laser to be switched off. System performance degrades with an excessive number of failures. The computational deadlines are dictated by the frame rate of the imaging camera (usually about 33ms/frame). Within this deadline, two computations must be completed: landmark extraction, and landmark-based image matching. For a given computer system, the computational budget is fixed within this deadline. It is desired to maximize the probability of a successful image match within this budget.

The process of landmark extraction is illustrated in Fig. 1 and the associated computational budgeting issues are shown in Fig. 2. One-dimensional image analysis on a coarse grid (Fig. 1b) is used to detect seed points for a procedure that recursively traces the vasculature. The landmarks are detected from the traces. The left and right columns of Fig 2 show the partial results of two different tracing procedures, captured at 8%, 28%, and 53% of the total computational effort [1]. The only difference between the two procedures is the order in which the vessel segments are traced; they use the same algorithm, the same total amount of computation, and produce the same final result (Fig. 1e). Although for each row the two partial results are obtained at the same amount of computational effort, the partial results in the right column are far more valuable, because they have more numerous and prominent landmarks. *The goal of this work is to discover and elucidate the design principles for achieving an early yield of high-quality landmarks, i.e., a sequence of partial results more like the right column of Fig. 2.*

Increasing the number and quality of landmarks can greatly improve the likelihood of a successful image match [10] for a given computational work. Given a high-quality partial result, such as the one in Fig. 2b, image matching can be attempted from the partially extracted landmark sets. If this is successful (i.e., sufficient confidence exists in the result), then the overall system can be much faster. Even if this is unsuccessful, it can form the basis for a better subsequent attempt. The frame cycle time can be sub-divided into a series of milestones, and the performance of the matching algorithms monitored at each milestone. Failure to reach set milestones can provide early indication of conditions such as poor image quality (images that are dim, saturated, out of focus, affected by glare, etc.).

The idea of utilizing partial results relates to the concept of imprecise computations as proposed by Lin et al. [11] within the real-time scheduling literature. An imprecise computation is a task with two parts: a mandatory

part that must be executed fully, and an optional part, that can be computed to the extent possible given the computational resources. A "correctness function" is defined to measure the quality (e.g. precision) of the system output as a function of time (processor cycles). Computation beyond the mandatory part results in monotonic increase of the correctness. Therefore the precision of the final output is related to the amount of computation. Such a formulation is attractive for designing predictable real-time systems that must handle variable amounts of computation [12,13,14,15,16,17]. This is exactly the situation faced in the tracing procedure where the frame cycle and application demands impose a deadline, but the amount of computation varies with the content and quality of the image data. The initial grid analysis and the extraction of a minimal landmark set (3 for affine indexing) represent the mandatory part of the computation, while extracting the remaining landmarks is the optional part. Our correctness measure is the cumulative quality of the landmarks, and our goal is to increase the correctness measure as quickly as possible. This work represents the first step towards realizing predictable real-time computer vision systems for spatial referencing, and it is the first time that the principles of imprecise computation have been used for medical image sequence analysis.

2. Background to the Present Study

The prior literature describes two approaches for extracting vascular landmarks in angiograms. The first approach requires extensive pixel processing, and generally relies on adaptive segmentation, followed by skeletonization and branch point analysis, or interest operators [e.g., 18,19,20,21]. Typically, this requires specialized hardware [22], scales poorly with image size, and does not provide useful partial results. The second approach [1,23], exemplified by this paper, is called vectorization, or exploratory tracing. This is much faster (e.g. video frame rates), more adaptive, and more practical for implementation on conventional and parallel MIMD computers [24]. The speed of this method can be appreciated by noting that computation of Sobel kernels at each image pixel, which is a computation representative of part of the processing in the first approach, requires 440ms; by contrast, the entire tracing computation (Fig. 1e) requires 200ms on the same image (Fig. 1a). Both results were obtained on 100MHz Silicon Graphics Indy using the same compiler settings. Vectorization also requires the fewest number of parameter settings, scales well with image size, and provides useful partial results.

The exploratory tracing algorithm, detailed in [1], is briefly summarized here. It proceeds in three stages. The first stage explores the image along a grid of pixel-wide lines (Fig. 1b), estimating the frame contrast and brightness levels, and seeking edges. A 1-D edge operator is applied, followed by local non-maximum suppression. The resulting points, illustrated in Fig. 1c, are termed edge pixels, or "edgels". In the figure, the small circles and squares mark the positions of the left and right edgels, respectively. The midpoint (marked white) of the strongest left and right edgels (E_L and E_R) is extracted as a seed point for tracing. These points are overlaid on the grid in Fig. 1b.

In the second stage, false seed points are filtered out by testing for the existence of a pair of sufficiently strong parallel edges. For this, a set of directional kernels [1] is applied to the seed's neighboring points along the

grid line, and the two strongest responses are found. The initial point is filtered out if the two strongest responses do not both exceed a sensitivity threshold, or if the directions of the two strongest edges are not sufficiently similar (within ±22.5°). On average, about 40% of the initial points are filtered out by this procedure.

The third stage is a sequence of recursive tracing steps illustrated in Fig. 1d. These are initiated at each of the filtered seed points, and proceed along vessel centerlines using an update equation of the following form:

$$\vec{p}^{\,k+1} = \vec{p}^{\,k} + \alpha \left[\cos\left(\frac{2\pi s^k}{N} \right), \quad \sin\left(\frac{2\pi s^k}{N} \right) \right]^T + \vec{\beta}_k , \tag{1}$$

where $\vec{p}^{\,k}$ and $\vec{p}^{\,k+1}$ denote the current and new (x,y) locations of the trace, α is a step size, s^k is an integer index specifying one of N angular directions (e.g., $N = 16$), and $\vec{\beta}_k$ is a lateral displacement that centers the new point $\vec{p}_k$ on the vessel. The angular direction is estimated using a set of N directional kernels that are applied separately to the left and right boundaries of each vessel. In Fig. 1d, this is illustrated for a pair of intersecting vessels. The left and right directional kernels at 0° and 45° are shown. In equation (1), s^k is estimated as the angle at which the correlation kernels produce the highest response. These maximum responses are computed by performing a local search along a line perpendicular to the current trace. The reader is referred to [1] for details, including a description of algorithms for pooling the traced segments, avoiding repeated searches, detecting branch points, crossovers, and end points (Fig. 1e), and correction of traces near branch points. Also described in this reference are methods for automatic estimation of the grid size used for initial exploration, the threshold used to terminate tracing, and the step size α.

3. A Quantitative Basis for Scheduling

The vascular tracing algorithm [1] generates a sequence of partial results, each consisting of traced vascular fragments and detected landmarks. The goal of the present work is to schedule the tracing computations to produce an early harvest of numerous high-quality landmarks (again, compare Fig. 2a and Fig. 2b). Designing the scheduling strategy requires quantifying both the quality of each partial result, and the computational work required to produce it.

Quality of a Partial Result. A first-order evaluation of the quality of a partial result is based on the *number* and *individual quality* of vascular landmarks. The quality of an individual landmark can be quantified based on the prominence of the intersecting vessels and their contrast, as follows. Suppose a single landmark is formed by the intersection of m vessel segments. Let t_p be the estimated thickness of segment p, and s_p a measure of its edge strength. Then, the quality of the landmark can be measured as

$$q = \sum_{p=1}^{m} s_p \cdot t_p . \tag{2}$$

The edge strength s_p is estimated using the directional correlation kernels described in our earlier work [1] and summarized in the previous section. The quality of a partial result is measured simply by summing the qualities of the individual landmarks constituting it. In future work, it may be possible to incorporate more discriminating criteria, such as the spatial distribution and uniqueness of landmarks [10].

Quantifying the Computational Work. One objective method to quantify computational work is to count the processor cycles. However, such a measure is necessarily dependent upon a specific combination of processor, compiler, and other implementation details. An extensive profiling analysis [24] revealed that the vast majority of the processor cycles were expended on computing the response of local correlation kernels [1]. This suggests a simple system-independent measure of computational work: simply count the number of correlation kernel computations. We use the variable w to denote this count.

The quality and work measures described above can be used to monitor the progress of the tracing computations, as follows. The tracing of the vasculature is carried out through a sequence of computations, with the work (as measured by counting correlation kernels) denoted $w_1, w_2, \ldots, w_N$, that produce a sequence of non-negative incremental results $q_1, q_2, \ldots, q_N$. The quality of the partial result at the completion of w_k is given by:

$$Q_k = \sum_{n=1}^{k} q_n \,, \tag{3}$$

where q_n is the quality measure defined in equation 2 for the n^{th} landmark. The cumulative computational effort used to produce the above partial result is given by:

$$W_k = \sum_{n=1}^{k} w_n \,. \tag{4}$$

At this point some observations may be made.

- The sequence of partial results is always of monotonically improving quality, i.e., $Q_{k+1} \geq Q_k$. In the terminology of imprecise computation, the tracing algorithm is an "anytime algorithm" [25].
- The amount of vasculature in the (finite-size) image is finite. Thus, a definite end point exists for the tracing, and the last partial result also represents the complete result. For a given image, and algorithm settings such as the grid size and sensitivity, the quality of the complete result, denoted Q_{total} , and the total computational work W_{total} , are both finite and fixed.
- Finally, the tracing of a vascular segment is independent of segments that were traced in previous steps, so the order of tracing can be changed without affecting the final result.

The above properties suggest that, in principle, one could divide the vasculature arbitrarily into vessel segments and decide the order in which the vessel segments are traced. Equivalently, a scheduling algorithm makes specific decisions regarding the starting, stopping, and restarting of the tracing, and choice of seed points.

An optimal scheduling algorithm makes decisions that maximize the rate at which the quality of partial results improves.

4. The Optimal Schedule

The monotonically improving quality of a partial result can be described by a growth function *Q(W)*. This function does not have a closed-form expression, and is represented by an empirically derived *Q vs. W* curve. Now, the quality and total work (Q_{total} and W_{total}) are different for different images. To enable meaningful comparison and averaging across images, it is advantageous to normalize Q and W by Q_{total} and W_{total} , respectively. With this in mind, the question of interest is: given a partial computation $W_k / W_{total} \leq 1$, how can the tracing computations be scheduled so that the normalized quality of the corresponding partial result, Q_k / Q_{total} , is maximized for every *k*. Another way to view this is maximizing the area under the normalized *Q vs. W* curve. The fully optimal scheduling algorithm must perform this maximization globally, for every partial result – a daunting task.

In order to maximize the optimality criterion just described, the scheduling algorithm must somehow perform just-sufficient tracing for detecting each of the landmarks, thereby minimizing the amount of work done. Simultaneously, it should trace around the highest-quality landmarks first, thereby maximizing the quality measure. Unfortunately, achieving this objective requires prior knowledge of the very traces that are sought - an inherently impossible task. *This realization leads to the inevitable conclusion that an optimal schedule can only be computed in hindsight, or equivalently, with perfect foresight.* In other words, the optimal schedule is a hypothetical one. Even with perfect foresight, computation of the optimal schedule is a difficult global combinatoric optimization problem; for each value of $W_k \leq W_{total}$ one must select a set of landmarks so that the sum of their computational work does not exceed W_k , and the sum of their qualities is the maximum. The following section describes a useful approximation of the optimal schedule based on perfect foresight.

4.1 An Approximation of the Optimal Schedule Based on Perfect Foresight

The quality growth curve *Q vs. W* can be estimated for a hypothetical optimal algorithm. First, the entire vasculature in the image is traced, and the quality of each landmark is computed. During the tracing, the computational work for each vascular segment is recorded. From these recordings, the computational work expended in computing each landmark is estimated using the following rules (illustrated in Fig. 3a).

1. If there are no landmarks between a landmark X and the end of the vessel, the work of tracing between X and the end of the vessel is attributed completely to this landmark.
2. The work of tracing a segment connecting two landmarks X and Y is divided into two equal halves and attributed equally to these two landmarks.

Fig.3a illustrates a sample vessel structure on which the above rules are applied. Landmark X is formed by vessel segments a, b, & c, while landmark Y is formed by vessel segments a, d, & e. If the computational work for segments *a, b, c, d, & e* is denoted w_a, w_b, w_c, w_d, & w_e, *respectively,* then the computational work for landmarks X and Y are estimated as $w_X = w_b + w_c + 0.5w_a$, and $w_Y = w_d + w_e + 0.5w_a$, respectively.

The assignment of computational work to the landmarks is independent of the schedule - it depends only on the spatial structure of the vessel network. Note, however, that item 2 above represents a simplification: during the tracing, the work of tracing the segment is not necessarily evenly divided between the two landmarks.

The above data provide the basis for approximating the optimal schedule in hindsight. The approximation is based on the heuristic observation that the ratio of quality, *Q,* and work, *W,* is high when the quality is high and the work is low. Of course, the quality to work ratio *Q/W* is not unique in this respect and other functions may be used.

The incremental tracing with the largest value of q_1/w_1 is scheduled first. The quality of the partial result at this point is $Q_1=q_1$, and the computational work is $W_1=w_1$. Thenceforth, subsequent partial results are constructed according to the following recursion. Given a partial result with quality Q_k and work W_k, the next item q_{k+1} is selected so that the new ratio Q_{k+1}/W_{k+1} is the highest, where Q_{k+1} is given by:

$$Q_{k+1} = Q_k + q_{k+1}. \quad (5)$$

The corresponding computational work is similarly given by:

$$W_{k+1} = W_k + w_{k+1}. \quad (6)$$

In Fig 3b, a plot of *Q vs. W* for the image in Fig 1a is shown. Also shown is the average of such curves for a diverse collection of 84 retinal images. These images were drawn from two clinics, and a published image collection on CD-ROM [26]. This set represents a benchmark for the present study. The optimal curves exhibit sharp increases at the beginning and a gradual flattening out to the maximum possible value.

5. Realizable Sub-Optimal Methods for Scheduling Tracing Computations

As noted above, the optimal scheduling algorithm is not achievable in practice since it requires prior knowledge of the very traces that are sought. This section describes methods that are realizable, and seek to approximate the optimal scheduling algorithm. The common basis for these methods is: (i) the exploitation of quantitative hints that are available during the initial 1-D grid search of the image for seed points (Fig. 1b), and (ii) methods to limit the computational effort.

5.1 Exploiting Hints based on Edge Strength and Vessel Width Estimates

The quality of a landmark as defined above (equation 2) depends on the edge strengths and widths of the intersecting vessel segments, suggesting that vessels with higher edge strength (denoted s), and larger thickness (denoted t) should perhaps be traced first. Furthermore, since both these quantities are involved as a product in equation 2, their product, $p = s \times t$ is also interesting as a prioritization criterion. These observations are

consistent with visual assessments of retinal images, in which the major vessels not only have higher contrast and thickness, but also are longer, and hence likely to have more numerous intersections/bifurcations. The initial grid search for seed points is an attractive opportunity to estimate the edge strengths, thickness, and products thereof, of vessels (Figs. 1b& 1e, [1]). The methods are summarized below.

1. **Edge strength estimates** $\{\hat{s}_{1D}\}$: The 1-D edge strength $\hat{s}_{1D}$ is defined as $s_{1D}(i, j) = L[i] + R[j]$, where $L[i]$ and $R[j]$ are the correlation responses at the left edgel, denoted i , and the right edgel, denoted j, to one-dimensional kernels of the form $[1, 2, 0, -2, -1]^T$ and $[-1, -2, 0, 2, 1]^T$, respectively.

2. **Vessel thickness estimates** $\{\hat{t}_{1D}\}$: The 1-D estimate of vessel thickness, denoted $\hat{t}_{1D}$, is defined as $t_{1D} = |j - i|$, where i and j are the indices of the left and right edgels, as defined above.

3. **Strength-thickness product estimates** $\{\hat{p}_{1D}\}$: The strength thickness product $\hat{p}_{1D}$ is computed straightforwardly as follows:

$$\hat{p}_{1D} = \hat{s}_{1D} \cdot \hat{t}_{1D} . \tag{7}$$

Interestingly, this product is less affected by the angle that a vessel makes with a grid line than the strength and thickness estimates individually, because the strength is underestimated and the width is overestimated. Suppose a vessel makes an angle θ with a horizontal grid line and has a true edge strength s. Then $\hat{s}_{1D}$ is the following projection:

$$\hat{s}_{1D} = s \sin\theta . \tag{8}$$

Similarly, if the true thickness of a vessel is t, then the one-dimensional estimate is a projection given by

$$\hat{t}_{1D} = t / \sin\theta . \tag{9}$$

From the above equations, we have

$$\hat{p}_{1D} = \hat{s}_{1D}\hat{t}_{1D} = st . \tag{10}$$

Prioritizing the seed points based on some combination of $\{\hat{s}_{1D}, \hat{t}_{1D}, \hat{p}_{1D}\}$ is a reasonable strategy if choosing the strongest seed points will lead to tracing more prominent vessels early in the computation, leading to earlier detection of high-quality landmarks.

Figs. 4a & 4b compare the impact of prioritizing the selection of seed points based on estimates of $\hat{s}_{1D}$, $\hat{t}_{1D}$, and $\hat{p}_{1D}$. These strategies are compared based on their quality growth curves (Q *vs.* W), averaged over the benchmark collection of 84 retinal images. The graphs on the left column of Fig. 4 are based on an initial search grid of 25 lines each along the x and y axes on each image, whereas the graphs on the right-hand column are based on 40 lines per axis. All the values of Q's and W's are normalized by the maximums among all the strategies. For comparison, we also studied methods that (i) select seed points randomly; and (ii) represent a reverse

prioritization of seed points based on the strength-thickness product. The last case is indicative of the worst-case scenario, and is therefore of considerable engineering value.

The graphs in Fig. 4 indicate the following trends among prioritization schemes. The scheme based on prioritizing by $\hat{p}_{1D}$ has one of the fastest growth rates on the quality of partial results. For the same partial computational work *W*, it has the largest quality of partial result among all curves. The random scheduling scheme is much worse, and the reverse scheduling scheme is the worst among all schemes. The prioritization scheme based on edge strength $\hat{s}_{1D}$ is almost the same as the one based on product of strength and width, $\hat{p}_{1D}$. The prioritization based on $\hat{t}_{1D}$ was mediocre, probably because the vessel thickness estimates are not as reliable as edge strength estimates. *Overall, these results indicate some of the benefits of scheduling, even when it is based on rudimentary 1-D measurements performed at seed points.*

Comparison of Fig. 4a and Fig. 4b illustrates the impact of the grid density. Fig. 4b, with the finer grid, has more initial points detected. The majority of the (more numerous) initial points are on shorter and weaker vessels that are missed by the sparser grid in Fig. 4a. Hence, the quality values of the initial points are more diverse, and the effect of prioritization is more pronounced. This leads to a larger spread among the curves. In Fig. 4a, the difference is less apparent, because the quality of the initial points is less varied and different scheduling schemes lead to similar results. *The practical impact of this study is that as computing speeds improve, and denser grids become computationally affordable, proper prioritization will become even more important.*

5.2 Impact of Improved Estimation of Strength and Thickness

The estimation of edge strength and vessel thickness can be improved by using a 2-D procedure. These estimates can be derived as byproducts of the seed point verification procedure (Stage 2) that filters out false initial points [1] using the correlation kernels, as described in Section 2 above. The 2-D edge strength $\hat{s}_{2D}$ is simply the sum of the highest left and right correlation kernel responses. The 2-D estimate of vessel thickness, denoted $\hat{t}_{2D}$, is the distance between the two points that have the strongest 2-D kernel responses. The strength thickness product $\hat{p}_{2D}$ is computed by multiplying $\hat{s}_{2D}$ and $\hat{t}_{2D}$.

Figs. 4c & 4d show the performance of scheduling based on $\hat{s}_{2D}$, $\hat{t}_{2D}$, and $\hat{p}_{2D}$. Notwithstanding the higher accuracy of 2-D estimation, there was no significant improvement over prioritization using 1-D quantities. This can be explained as follows. Whereas the prioritization of initial points by the above measures does lead to the tracing of more prominent vessels, it does not necessarily lead to prioritized tracing of the most important segments – ones that form the landmarks. *We therefore conclude that the strategy of filtering all the initial points beforehand does not significantly improve the growth rate of the partial result quality.*

5.3 Impact of Preempting Long Traces to Limit Computational Work

From the similar quality growth curves resulting from the 4 prioritization schemes based on $\hat{p}_{1D}$, $\hat{s}_{1D}$, $\hat{p}_{2D}$, and $\hat{s}_{2D}$, we may conclude that the sub-linear behavior of these curves is about the best that one can achieve using any of the prioritization algorithms discussed above. To improve upon these strategies, we note that they consider the quality of the initial points, which is related to the quality of the landmarks, but do not consider the computation needed to detect the landmark. For instance, it is quite possible, indeed common, for the algorithm to get caught up in a long trace over a strong and thick vessel that yields very few intersections, as illustrated in Fig. 2(a). To prevent such a situation, it makes sense to preempt the tracing. This is loosely analogous to preemptive scheduling of processes in operating systems [17,27], where preemption serves to improve the average response time, and limits the damaging effects of processes that are caught in infinite loops.

The simplest preemptive scheme is to stop a tracing thread after a certain fixed number of steps, and returning the stopping point to the priority queue. The priority of the returned point is computed the same way as that for initial points. Figs. 5a & 5b show the impact of preempting traces after 40 steps. The number of steps was estimated empirically. Even for this simple-minded scheme, the *Q vs. W* curve is promising - rising above the best non-preemptive result (Fig. 4).

Some modifications of the simple preemptive algorithm were also studied. As an example, we note that often the tracing passes through a branch point and continues (e.g., see Fig 2(a)). If during the tracing, such a case could be detected [1], the next tracing thread could be initiated from the strongest seed point near the location of detection, hoping that an immediate intersection will result. Although this strategy outperforms the simple-minded preemptive schedule, the improvement is modest. This is because our detection strategy for these cases is unsophisticated – based on comparing the edge response on the left and right boundaries of the vessel being traced, and declaring a detection when the ratio is large enough [1]. A false indication, or a poor choice of nearby seed point disrupts the order of prioritization without any payoff. If numerous such cases occur, this algorithm will be out-performed by a simple non-preemptive prioritization scheme. Unfortunately, the detection method has to be a simple scheme that does not add too much to the overall computational cost.

In conclusion, preemptive algorithms are generally better than the non-preemptive ones, since they have the ability to end long and non-productive tracing threads.

5.4 Heuristic Algorithm Based on Spatial Prioritization of Tracing

A detailed step-by-step dissection of the optimal schedule for several images reveals that it primarily derives its high performance from advance knowledge of the spatial locations and quality values of the landmarks. It concentrates just-sufficient levels of tracing effort around the most promising landmarks. This observation suggests predicting landmark locations and qualities, and using these to schedule the tracing.

Landmark locations and qualities can be predicted from the initial processing (Stages 1 and 2). As noted earlier, a filtering procedure can be used to eliminate all seed points that do not correspond to a pair of nearly parallel edges, and as a byproduct, produce local orientation, strength, and width estimates (Fig. 6a). Shown in Figs 6b - 6g are cropped and enlarged views of selected boxes formed by the grid lines. In these images, the short lines crossing the grid lines mark the estimated orientations of the initial points. The black dots on these lines indicate the location of the initial points after the filtering. From these images, the presence/absence of landmarks inside the grid box, can be guessed even before tracing, by analyzing the distributions and orientations of the initial points around the grid box. For example, in Figs. 6b-6d, the orientations of the initial points clearly suggest the existence of at least one landmark inside the grid box. Conversely, in Figs. 6e & 6f, although there are initial points associated with that grid box, their parallel orientations suggest that there is probably no landmark in the grid box. However, not all grid boxes exhibit the simplicity of Figs. 6b-6f. For instance, the right side box in Fig. 6g shows a more complex case. Although there is no landmark inside this box, a landmark is next to its left border. This suggests that one should not only consider the possibility of a landmark inside a box, but also its neighborhood. On the other hand, errors in seed point detection or filtering can produce misleading conclusions about grid boxes.

A computationally simple yet effective measure is needed to capture these intuitions. Primarily, it must rate the likelihood of a landmark occurring inside a grid box, and estimate its quality. We choose a weighted angular diversity measure adapted from angular statistics, [28,4]. The angular variance of a set of unit vectors $\{\vec{u}^{(1)}, \vec{u}^{(2)} \cdots \vec{u}^{(K)}\}$ is defined as $1 - \frac{1}{K}\left\| \sum_{k=1}^{K} \vec{u}^{(k)} \right\|$, where ||.|| is the standard Euclidean norm. Intuitively, the length of the resultant of the unit vectors is as large as possible K when all the vectors are aligned with each other, and the least possible 0 when they are pointing in opposite directions, canceling each other out. For the present work, a strength-weighted version of the angular variance is used, and is further weighted by the total number of vectors. Let the number of initial points on a grid box be K. Let the 2-D estimates of edge strength be $\hat{s}_{2D}^{(k)}$, k = *1,2...K,* and let the unit vectors $\vec{u}^{(k)} = [\cos\theta^{(k)}, \sin\theta^{(k)}]^T$ indicate the local vessel orientation estimates. Then the strength-weighted angular diversity measure for a grid box G is:

$$D(G) = \sum_{k=1}^{K} \hat{s}_{2D}^{(k)} - \left\| \sum_{k=1}^{K} \hat{s}_{2D}^{(k)} [\cos 2\theta^{(k)}, \quad \sin 2\theta^{(k)}] \right\|. \tag{11}$$

The angles must be doubled because the angular diversity between a pair of vectors is maximum when they differ by $\pi/2$, and minimum when they differ by π. In computing the above measure, an initial point is assigned to a grid box if it is located on or sufficiently close to any of the gridlines that form the grid box. Therefore, an initial point may belong to more than one grid box.

The grid boxes are prioritized by their values of $D(G)$. Boxes with more numerous and stronger initial points and with a higher angular diversity will have greater values of $D(G)$. To illustrate the effectiveness of this approach, the top 10% of the grid boxes in the priority queue for the image shown in Fig 6a are indicated with dots in the center. Note that most of these marked boxes have at least one landmark within or nearby.

Within a grid box, the initial points are prioritized equally. The tracing is preempted after a number of steps that is proportional to the size of a grid box. Upon preemption, the stopping point is inserted back into the priority queue at the same level of priority as the grid box that the tracing was stopped in. An exception occurs when the tracing is preempted in a grid box whose priority is higher than the grid box from which the tracing was initiated. In this case, the tracing is continued until the next preemption. Therefore this is a preemptive scheduling algorithm with both spatial and edge strength based prioritization. We term this "spatial prioritization scheduling".

Fig.7 shows the results of applying the above scheduling algorithm on 4 selected images. The curves in the top row represent cases when the scheduling algorithm was successful in approximating the optimal schedule.

The lower row of Fig. 7 shows two less-successful cases. Although these results are still superior to random scheduling, they exhibit a characteristic “plateau”. These occur most often due to errors in the seed points. When many initial points are missed, or wrongly filtered, especially initial points on the strong vessels, the *Q vs. W* curve exhibits a sharp increase in the beginning, a plateau in the middle, and another sharp increase at the end. At the beginning, boxes with the most numerous and strongest initial points are processed, leading to the detection of numerous and strong landmarks, and a corresponding sharp increase in the quality of partial results. However, due to missing initial points, some of the boxes containing landmarks are assigned low priority. In quite a few cases, the missed initial points are located on strong vessels. They are often missed due to imaging artifacts such as glare, and because they sometimes appear “hollow” [1], i.e., the central portion is lighter than the boundaries. In the late stages, an initial point for this vessel that belongs to another box with very low priority may finally trace to the landmark located in that patch and there will be a sharp increase at the end.

The problem of a plateau in the growth function is addressed by dynamically updating the priorities of the grid boxes as the tracing progresses. Specifically, seed points that were missed during the grid search are detected whenever a trace crosses a grid line, and the priority queue is updated accordingly. The panels in the second row of Fig. 7 illustrate the visible improvement resulting from such dynamic updating. Illustrated in the panels of the bottom row are the two cases where the dynamic updating scheme results in improvements on the growth rate of the quality. In both cases, the missing or false filtering of an initial point greatly reduces the priority of the related grid box. Fortunately, a nearby grid box has a high priority and the trace initiated from that box crosses the grid line where the initial point was missed. Because of dynamic updating, the detection of this missing initial point would greatly increase the priority of the related grid boxes. Therefore, the landmark in this box will be obtained much earlier, resulting in an earlier increase of the quality measure.

The curve shown in Fig.8 is the average of a variety of cases representing varying degrees of success. Overall, this curve is much closer to the optimal curve and clearly outperforms all the others described above. Indeed, this is the only realizable algorithm (among those studied) that delivers a growth rate that is better than linear. Visual inspection of this algorithm's behavior (illustrated in Fig. 2b) indicates that it has the qualitative behavior of the optimal schedule - performing limited and localized tracing just around the most promising regions.

Summary: Table 1 summarizes the relative performance of the scheduling strategies discussed above. It shows the quality of the partial result at the 33% effort level; i.e., $W/W_{total} = 0.33$. The most effective and systematic approach for the design of real-time vectorization algorithms was suggested by the spatial prioritization behavior of the optimal schedule. This algorithm produces a partial result whose quality is roughly 400% better than that produced with purely random scheduling.

6. Discussion and Conclusions

This paper has described several methods for scheduling the computations needed to trace the vasculature in retinal images. This is needed for reliable real-time extraction of vascular landmarks as part of image-based spatial referencing techniques for laser retinal surgery. The optimal schedule is defined as the one maximizing the area under a normalized quality (Q) vs. computational cost (W) curve.

An optimal scheduling algorithm is impossible to attain, and the optimal schedule is difficult to determine even after the traces and landmarks are known. A heuristic estimate of the optimal schedule was developed and used to guide the design of realizable scheduling algorithms. The most important idea in the design was the use of the initial grid analysis to predict locations of landmarks. Seed points near predicted locations received high priority for tracing. Dynamically recomputing priorities based on the tracing results, and preempting tracing also contributed to the final scheduling algorithm.

The Q *vs.* W curve is somewhat analogous to the "correctness function" discussed within the real-time computation literature (see for example, the work of Lin et al., [11], who introduced the concept of "imprecise computations"). Starting from zero, the correctness function is formulated to grow monotonically with computation until it reaches unity. In this type of computation, it is of interest to maximize the growth in correctness to ensure that the best-possible partial result is available when the computational deadline occurs. The present work appears to be the first to apply these ideas to real-time image analysis.

Within the broader context of the intended application, much work remains. For the landmark tracing itself, as indicated by the difference between the approximately optimal schedule and our best algorithm, opportunities for future improvements exist. For example, the proposed methods can be applied more locally, or modified based on regions of surgical interest. The position of the aiming laser, often present in instruments, can serve as another

cue for prioritization. The grid computations may themselves be prioritized in a multi-resolution manner. For the present work, this was not pursued since this step takes little effort, so the potential payoff is modest. This could change if more sophisticated grid analysis algorithms are adopted. The measure used to prioritize the grid boxes could be improved using geometric models of vasculature [29], so that the prediction of locations of landmarks can be more accurate. The preemption can also be made more adaptive than the current simple strategy. On the other hand, more sophisticated strategies may not necessarily be better overall. For instance, as noted in Section 5, prioritizations based on the more accurate 2-D estimates of $\hat{p}_{1D}$, $\hat{s}_{1D}$, $\hat{p}_{2D}$, and $\hat{s}_{2D}$ do not significantly outperform the algorithms based on 1-D measures. Further research is needed to discover strategies that approximate the optimal schedule even better, once the overheads are accounted for. Most importantly, these ideas must be evaluated in terms of their effect on the ultimate goal of real-time spatial referencing.

7. Acknowledgements

The authors would like to thank the staff at the Center for Sight, especially photographers Gary Howe and Mark Fish, for assisting with image acquisition. Thanks also to Matthew Freshman for a part of the retinal photography. The authors appreciate the insightful inputs of Dr. George Nagy at Rensselaer, especially regarding vectorization methods from the document image analysis literature, and those of Dr. Robert Kelley. Finally, many thanks to Ali Can and Khalid Al-Kofahi for various valuable inputs, and to Edward Balduf for the dual-processor frame-rate video based implementation.

8. References

1. Can, A., Shen, H., Turner, J. N., Tanenbaum, H. L., and Roysam, B., “Rapid Automated Tracing And Feature Extraction From Retinal Fundus Images Using Direct Exploratory Algorithms”, IEEE Transactions on Biomedical Information Technology, vol.3 no.2, June 1999.
2. Barrett, S. F., Wright C. H. G., Jerath, M. R., “Computer-aided Retinal Photocoagulation System, Journal of Biomedical Optics, vol. 1, pp.83-91, 1996.
3. Berger, J. W., and Shin D. S., “Computer Vision Enabled Augmented Reality Fundus Biomicroscopy,” Ophthalmology, vol. 106, no. 10, Oct. 1999.
4. Becker, D. E., "Real-Time Algorithms For Automated Laser Eye Surgery," PhD thesis, Rensselaer Polytechnic Institute, 1995.
5. Shen, H., “Optical Instrumentation and Real-Time Image Processing Algorithms for Simultaneous ICG and Red-Free Video Angiography of the Retina,” M.S. Thesis, Rensselaer Polytechnic Institute, Troy, NY, 1996.

6. Becker, D. E., Can, A., Tanenbaum, H. L., Turner, J. N., Roysam, B., "Image Processing Algorithms For Retinal Montage Synthesis, Mapping, And Real-Time Location Determination," *IEEE Trans. on Biomed. Eng.*, vol. 45, no. 1, Jan. 1998.
7. Mahurkar, A. A., M. A. Vivino, B. L. Trus, E. M. Kuehl, M. B. Datiles III, M. I. Kaiser-Kupfer, "Constructing Retinal Fundus Photomontages," Investigative Ophthalmology & Visual Science, vol. 37, no. 8, pp. 384-401, pp. 1675-1683, July 1996.
8. Can, A., Stewart, C. V., and Roysam, B., "Robust Hierarchical Algorithm for Constructing a Mosaic from Images of the Curved Human Retina," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. II, 286-292, Fort Collins, Colorado, June 1999.
9. Laplante, P. A. *Real-Time Systems Design And Analysis, An Engineer's Handbook*, second edition, IEEE press, 1997.
10. Haralick , R. M., and Shapiro, L. G., *Computer and Robot Vision*, vol.1, Addison Wesley Publishing Company, Inc., 1992.
11. Lin, K.J., Natarajan, S., and Liu, J.W.-S., "Imprecise Results: Utilizing Partial Computations In Real-Time Systems," NASA Report CR-180561, 1987.
12. Natarajan, S., *Imprecise and Approximate Computation,* Kluwer Academic Publishers, Boston, 1995.
13. Baruah, S. K., and Hickey, M.E., "Competitive On-Line Scheduling Of Imprecise Computations," IEEE Transactions on Computers, vol. 47, no.9, Sept., 1998.
14. Chong, E. K. P., and Zhao, W., "User-Controlled Optimization Of Task Scheduling For Imprecise Computer Systems," Information And Software Technology, vol. 34, no. 4, April, 1992.
15. Chen, I.R., "On Applying Imprecise Computation To Real-Time AI Systems," The Computer Journal, vol. 38, no.6, 1995.
16. Shih, W.K., and Liu, J.W.S., "Algorithms For Scheduling Imprecise Computations With Timing Constraints To Minimize Error," IEEE Transactions On Computers, vol. 44, no.3, March, 1995.
17. Liu, C.L., and Layland, J.W., "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," Journal of the Association for Computer Machinery, vol.20, no.1, January 1973.
18. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M., "Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters," *IEEE Transactions on Medical Imaging*, vol.8, 263-269, Sept. 1989.
19. Goldbaum, M. H., Kouznetsova, V., Coté, B. L., Hart, W. E., and Nelson, M., "Automated Registration Of Digital Ocular Fundus Images For Comparison Of Lesions," in *SPIE: Ophthalmic Technologies III*, vol. 1877, 94-99, 1993.

20. Goldbaum, M. H., Katz, N., Chaudhuri, S., Nelson, M., and Kube, P., “Digital Image Processing For Ocular Fundus Images,” *Ophthalmol. Clin. N. Amer.,* vol. 3, no. 3, 447-466, Sept. 1990.
21. Cohen, A. R., Roysam, B., and Turner, J. N., "Automated Tracing and Volume Measurements of Neurons from 3-D Confocal Fluorescence Microscopy Data," Journal of Microscopy, Vol. 173, Pt. 2, February 1994.
22. Polli, R., and Valli, G., “An Algorithm For Real-Time Vessel Enhancement And Detection,” *Computer Methods and Programs in Biomedicine,* vol. 52, 1-22, 1997.
23. Coatrieux, J. L., Garreau, M., Collorec, R., and Roux, C., “Computer Vision Approaches For The Three-Dimensional Reconstruction: Review And Prospects,” *Critical Reviews in Biomedical Engineering,* vol. 22, no. 1, 1-38, 1994.
24. Balduf, E., “Real-Time Algorithms for a Laser Retinal Surgery System - Implementation on the Silicon Graphics Octane Dual-Processor System", M.S. Thesis, Rensselaer Polytechnic Institute, 1998.
25. Zilberstein, S., and Russel, S., “Approximate Reasoning Using Anytime Algorithms,” pp. 43-62, in *Imprecise and Approximate Computation, (ed. Swaminathan Natarajan), Kluwer Academic Publishers, 1995.*
26. Yannuzzi, L. A., Guyer, D. R., and Green, R. W., “The Retina Atlas (on CD-ROM),” Mosby Year Book, Inc., St. Louis, MO, 1998.
27. Silberschatz, A., and Galvin, P.B., *Operation system concepts*, Addison Wesley Longman Inc., 1998.
28. Watson, G. S., *Statistics on Spheres*, John Wiley & Sons, New York, 1983.
29. Schreiner, W., and Buxbaum, P. F., “Computer-Optimization of Vascular Trees”, *IEEE Transactions on Biomedical Engineering,* vol.40, no.5, May 1993.

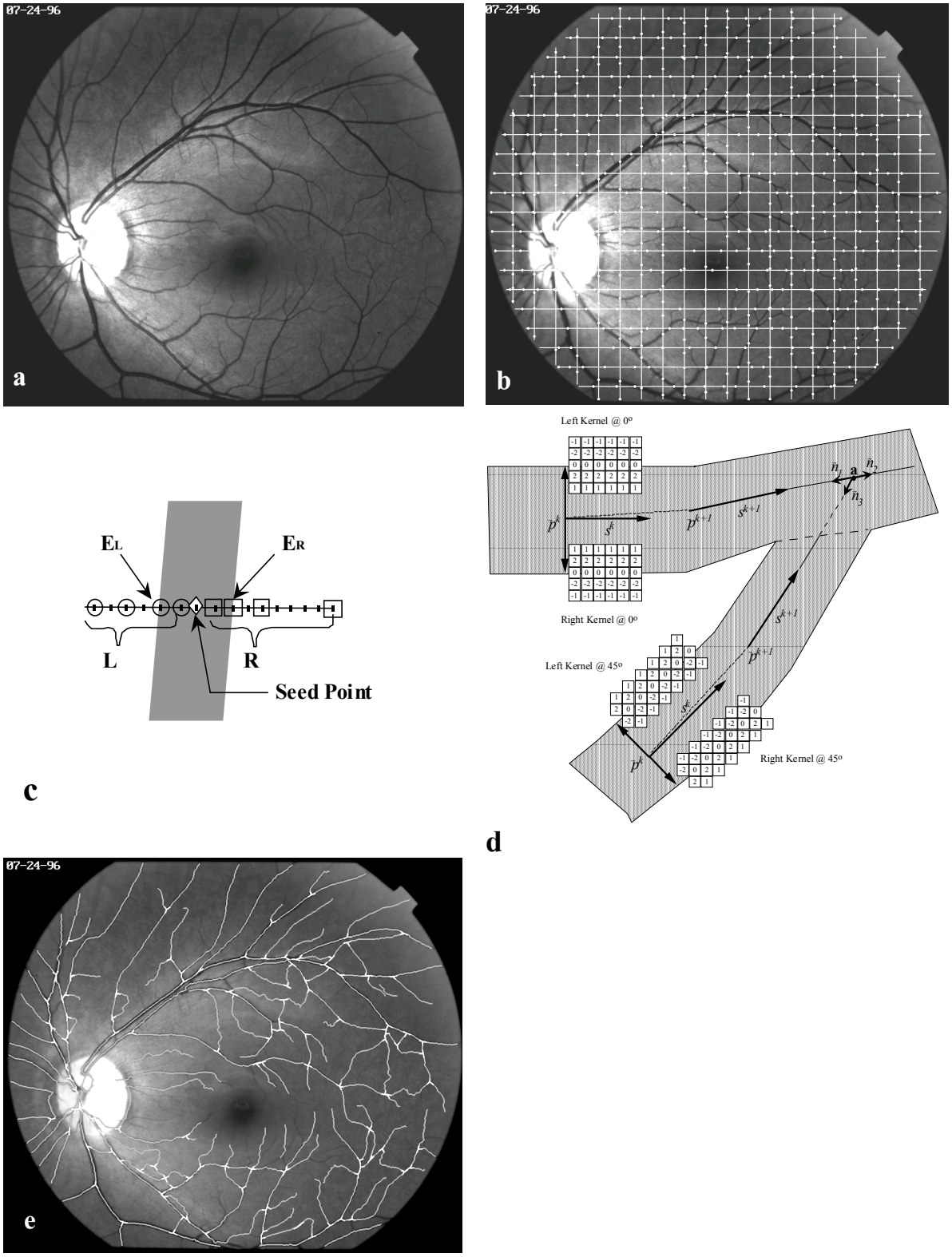


Figure 1: Illustrating the vectorization approach for tracing vasculature: (a) a sample red-free retinal angiogram (1024×1024×8 bits); (b) initial sparse grid search for seed points; (c) illustrating the method for deriving the seed points indicated in panel b; (d) illustrating the recursive tracing algorithm; (e) completed tracing result, with the vascular landmarks (crossing and branch points) highlighted.

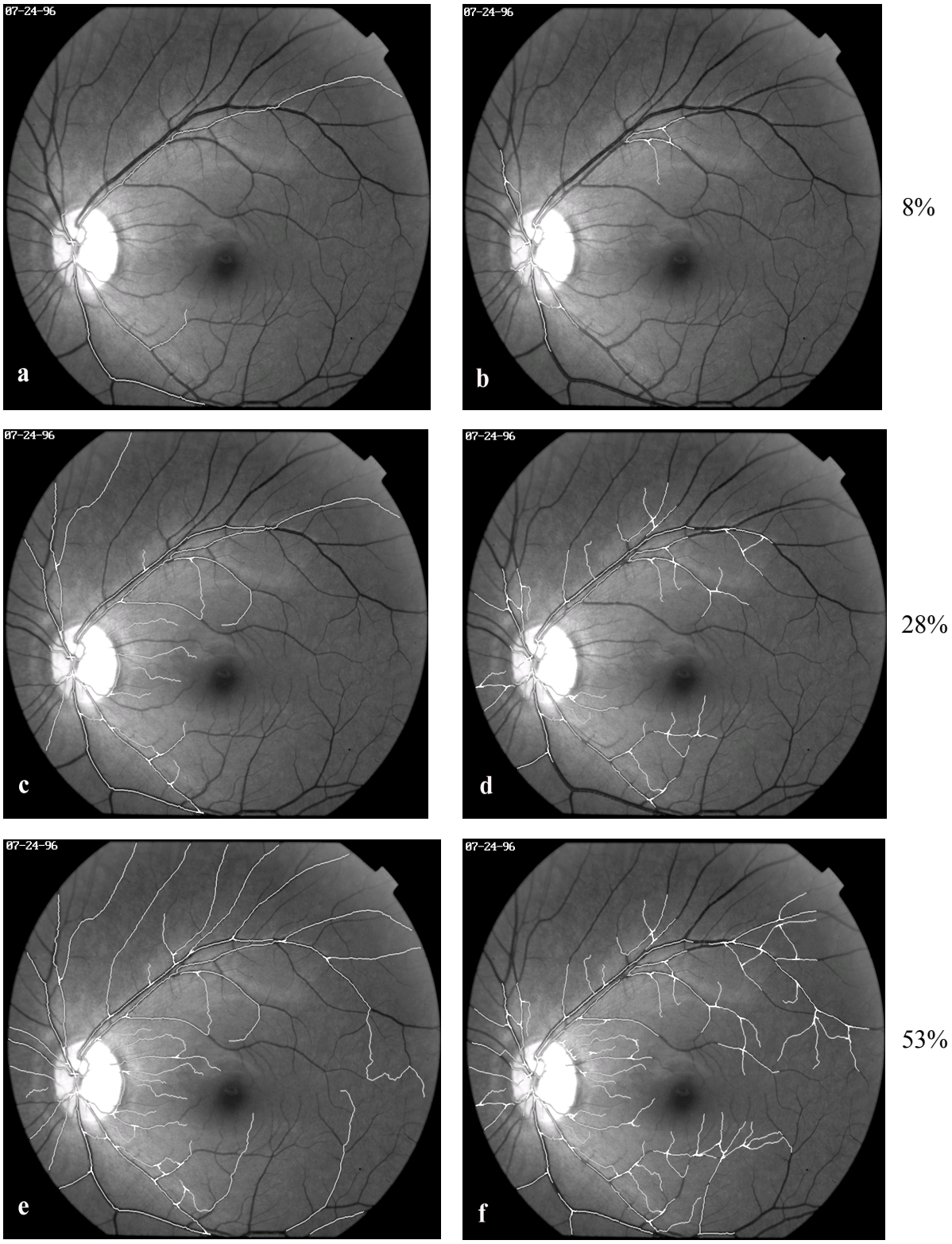
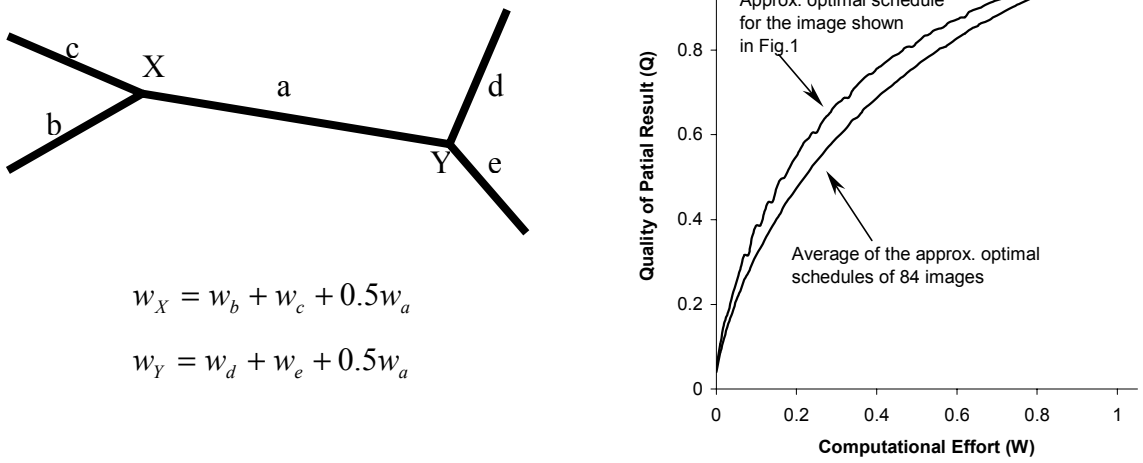


Figure 2: Illustrating the impact of scheduling on landmark detection performance for the same level of computational effort (W/W_{total}): (a) a partial result from tracing Fig. 1(a) using a poorly scheduled algorithm shows the work wasted on long and unproductive tracing; (b) a partial result from a better scheduled algorithm for the same computational work. The latter is much more productive, having prioritized its efforts in a manner that yields numerous prominent landmarks early; (c-f) Partial results for the two algorithms at 28% and 53% effort.



(a) (b)

Figure 3: (a) Illustrating the approximate method for calculating the computational work for extracting landmarks X and Y for the purpose of estimating the optimal schedule. The work of tracing the segment connecting X and Y is attributed equally to X and Y. (b) Approximation of the optimal schedule for the image in Fig. 1, and the average approximation for a set of 84 images.

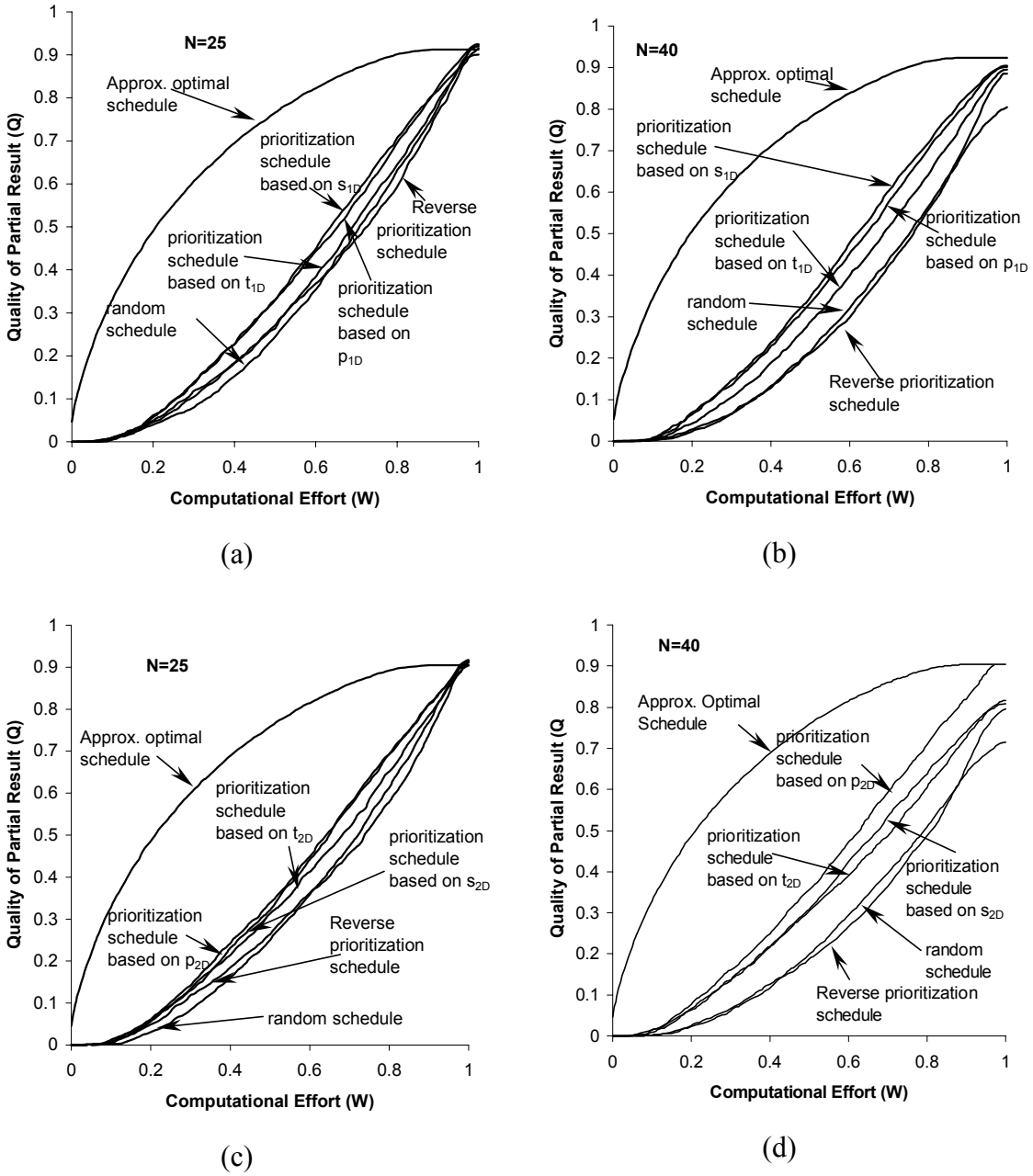
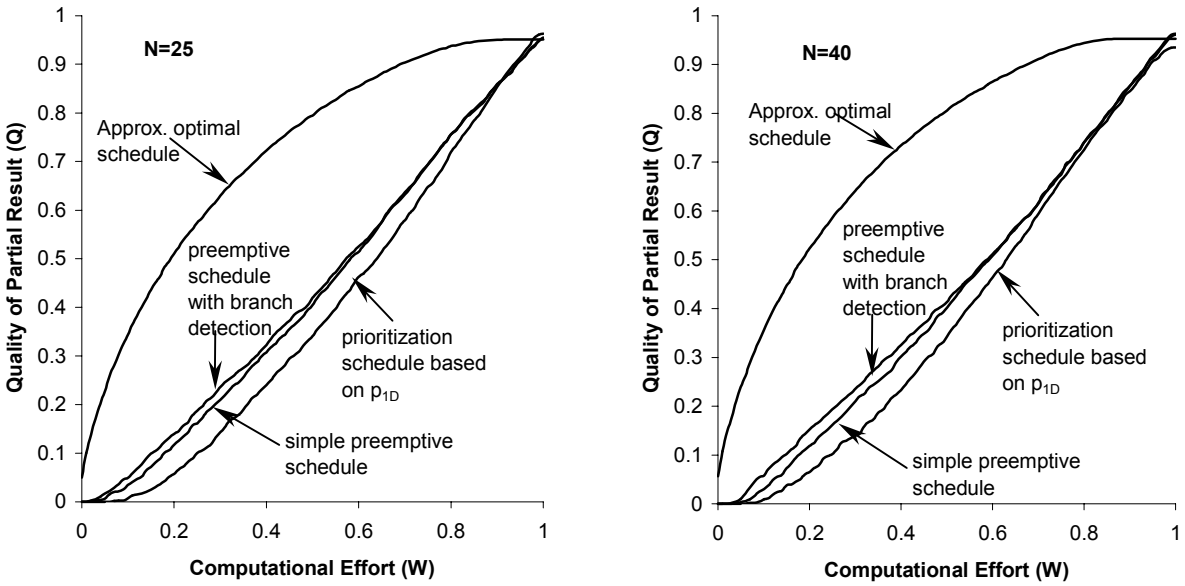


Figure 4: *Q vs. W* plots showing the results of scheduling based on prioritizing seed points using 1-D (upper row) and 2-D (lower row) estimates of $\hat{s}_{1D}$, $\hat{t}_{1D}$, $\hat{p}_{1D}$, $\hat{s}_{2D}$, $\hat{t}_{2D}$, $\hat{p}_{2D}$, at grid sizes of 25×25 lines (left column), and 40×40 lines (right column), respectively. For the 1-D case, the quality growth curve is seen to be worst for reverse prioritization by $\hat{p}_{1D}$, and best for prioritization based on $\hat{s}_{1D}$ or $\hat{p}_{1D}$. Random scheduling yields an intermediate level of performance. As expected, the denser grid search better reveals the differences between the prioritization schemes. The 2-D quality growth curve shows the same trends as the 1-D case.



(a) (b)

Figure 5: Illustrating the impact of preempting the tracing after 40 steps, at grid sizes of 25×25 lines (a), and 40×40 lines (b), respectively. The *Q vs. W* curve for the preemptive schedule outperforms the best of the non-preemptive schemes (seed points prioritized based on $\hat{p}_{1D}$), reproduced here for comparison.

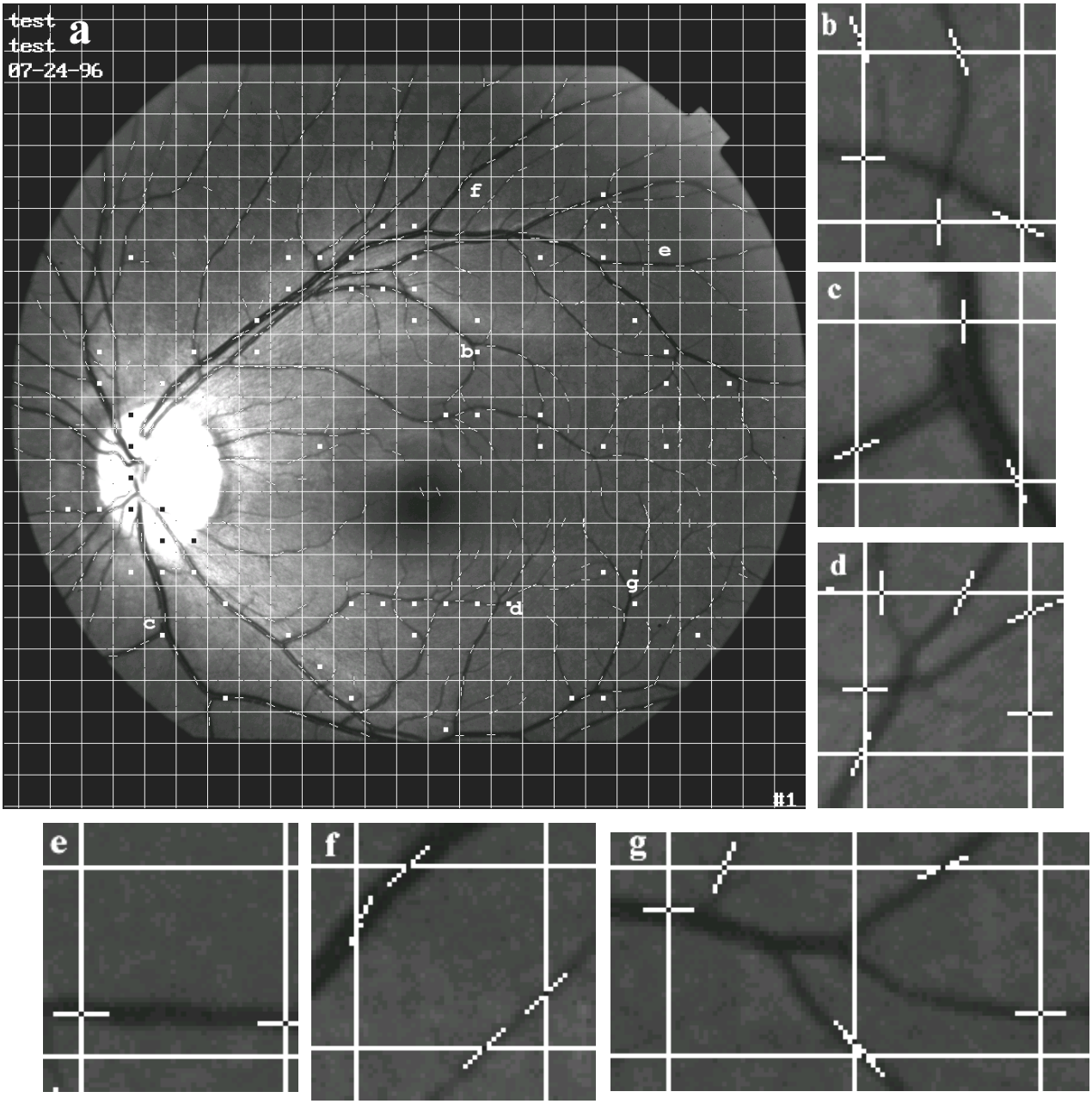


Figure 6: Illustrating the basis for spatial prioritization. (a) The small line segments crossing the gridlines indicate the filtered seed points from Fig.1b. Their orientations indicate the estimated local vascular orientations. Panels b-g highlight selected regions from Panel a. Boxes with the highest 10% of $D(G)$ values are shown marked with a dot in the middle. Panels (b-d) illustrate cases when the filtered initial points provide strong clues about the presence of landmarks within the grid box. Panels e & f illustrate cases when the initial points suggest the absence of landmarks. Panel g illustrates, a more complex case. Although there is no landmark inside this box, a landmark is right beside its left border. This example suggests that one should not only consider the possibility of a landmark inside a box, but also in the box's neighborhood. Notwithstanding such cases, this approach is very successful.

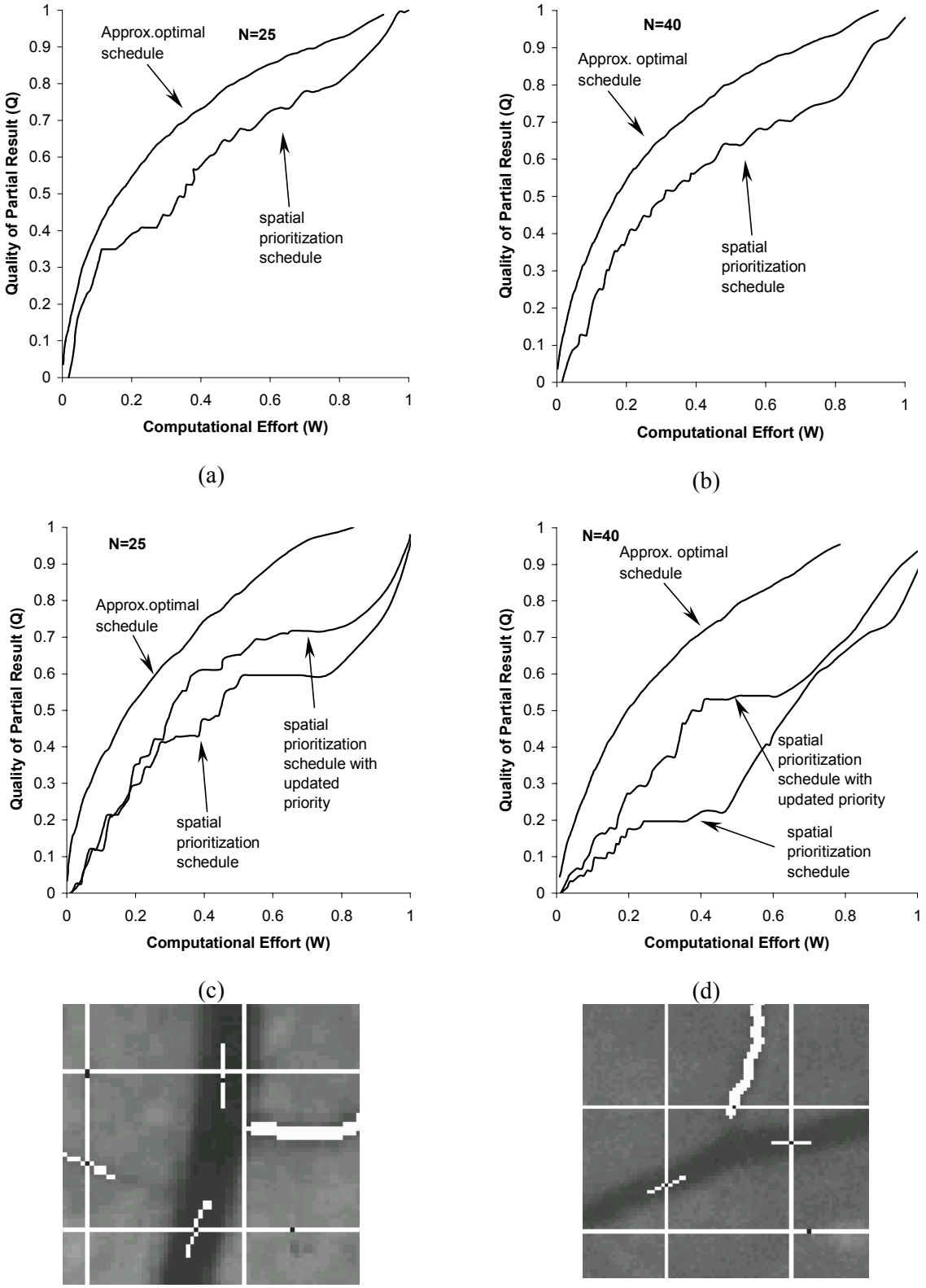
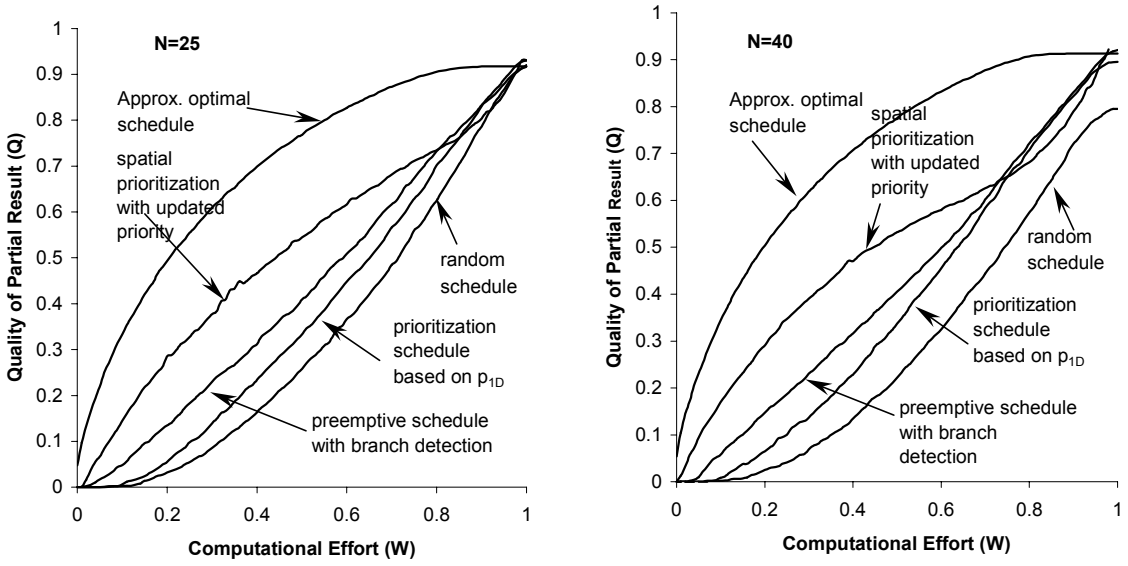


Figure 7: Examples of *Q vs. W* curves for selected images for spatial prioritization scheduling. The top row indicates successful cases with the curve approaching the optimal. The second row shows less successful cases, labeled "spatial prioritization schedule". Some improvement was observed for such cases by dynamically updating the grid box priorities (curves labeled "spatial prioritization with updated priority"). The bottom row shows two boxes whose priority increased significantly due to dynamic updating.



(a) (b)

Figure 8: Average *Q vs. W* plots showing the impact of preemption, spatial prioritization of seed points, and dynamic updating of priorities, ideas based on approximately mimicking the optimal schedule (labeled "spatial prioritization with updated priority") at grid densities of 25 (panel a), and 40 (panel b), respectively. The combination of these approaches is seen to outperform all other schedules that were studied. The relative performance is quantified in Table 1.

Table 1: Summarizing the average quality of the partial result attained after 33% of the total computational work is expended. Note that the best case (preemptive spatial prioritization scheduling) has a quality value that is roughly 400% better compared to random scheduling.

Scheduling Method	Q @ W = 33% (N=25)	Q @ W = 33% (N=40)
1. Random prioritization	0.10	0.09
2. Non-preemptive prioritization without seed point filtering		
• Reverse prioritization by p_{1D}	0.13	0.08
• Prioritization by t_{1D}	0.13	0.13
• Prioritization by s_{1D}	0.16	0.17
• Prioritization by p_{1D}	0.17	0.16
3. Non-preemptive Prioritization after filtering seed points		
• Reverse prioritization by p_{2D}	0.13	0.08
• Prioritization by t_{2D}	0.15	0.15
• Prioritization by s_{2D}	0.16	0.15
• Prioritization by p_{2D}	0.17	0.20
4. Preemptive Scheduling with:		
• Prioritization by p_{1D}	0.24	0.24
• Prioritization by p_{1D} with branch detection	0.26	0.27
• Spatial Prioritization Scheduling with dynamically updated priority.	0.41	0.41