

ShareCam Part II: Approximate and Distributed Algorithms for a Collaboratively Controlled Robotic Webcam *

Dezhen Song, Ken Goldberg

IEOR and EECS Departments, UC Berkeley, United States

Anatoly Pashkevich

State University of Informatics and Radioelectronics, Belarus

ShareCam is a robotic pan, tilt, and zoom web-based camera controlled by simultaneous frame requests from on-line users. A companion paper¹ describes the system. This paper, Part II, focuses on algorithms. The ShareCam problem is to find a camera frame that optimizes a measure of total user satisfaction. We present a grid-based approximation algorithm: given camera frame requests from n users, and approximation bound ϵ , we analyze the trade-off between solution quality and processing speed and prove that the algorithm runs in $O(n/\epsilon^3)$ time. The algorithm can be distributed to run in $O(1/\epsilon^3)$ time at each client and in $O(n + 1/\epsilon^3)$ time at the server. Experiments suggest that performance of the distributed algorithm degrades gracefully as clients fail to complete their part of the computation. ShareCam can be found online at: <http://www.tele-actor.net/sharecam/>.

1 Introduction

Consider a robotic webcam set up at a panoramic site such as the San Francisco Bay, Sydney harbor, etc. The camera frame (based on pan, tilt, and zoom) can be remotely adjusted by viewers via the Internet to observe details in the scene. Current control methods rely on queueing, where users have to wait patiently for their turn to operate the camera. “ShareCam” is a new system that eliminates the queue, allowing many users to share simultaneous control of the camera. The Sharecam problem is to find a camera frame that maximizes a measure of user satisfaction.

The “Sharecam” system is an example of Collaborative Telerobotics, in this case the telerobot is a camera with 3 degrees of freedom. In the taxonomy proposed by Tanie et al. [3], ShareCam is a Multiple Operator Single

*Submitted IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2003. This work was supported in part by the National Science Foundation under IIS-0113147, by Intel Corporation, by Microsoft Corporation, and in part by UC Berkeley’s Center for Information Technology Research in the Interest of Society (CITRIS). For more information please contact dzsong@ieor.berkeley.edu or goldberg@ieor.berkeley.edu.

¹The paper *ShareCam Part I: Interface, Architecture, and Implementation* has been submitted separately to this conference, and can be found at <http://teleactor.net/sharecam/pubs>

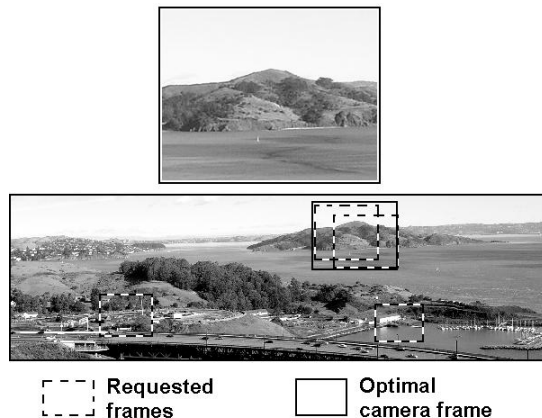


Figure 1: Sharecam’s Java-based interface on the Internet. The user interface includes two image windows. The lower window displays a fixed “panoramic” image based on the camera’s full workspace (reachable field of view). Each user requests a camera frame by positioning a dashed rectangle in the lower window. Based on these requests, the algorithm computes an optimal camera frame (shown with solid rectangle), moves the camera accordingly, and displays the resulting live streaming video image in the upper window.

Robot (MOSR) system. Collaborative Telerobotics is motivated by applications such as education and journalism, where groups of users desire simultaneous access to a single robotic resource. Inputs from each user are combined to generate a single control stream for the robot.

As illustrated in Figure 1, the Sharecam java-based interface includes two image windows. Problem input is the set of requested camera frames from n users. Problem output is a camera frame that maximizes user satisfaction as defined in Section 3. For approximation bound ϵ , our algorithm runs in $O(n/\epsilon^3)$ time.

2 Related Work

The Internet provides a low-cost and widely-available interface that can make physical resources accessible to a broad range of participants. Online robots, controllable over the

Internet, are an active research area. In addition to the challenges associated with time delay, supervisory control, and stability, online robots must be designed to be operated by non-specialists through intuitive user interfaces and to be accessible 24 hours a day; see [11, 12, 13, 15, 18, 21, 23, 24] for examples of recent projects.

Chong et al. [3] proposed the following taxonomy for teleoperation systems: Single Operator Single Robot (SOSR), Single Operator Multiple Robot (SOMR), and Multiple Operator Multiple Robot (MOMR). and Multiple Operator Single Robot (MOSR). Most online robots are SOSR, where control is limited to one operator at a time. One precedent of an online MOSR system is described by McDonald, Cannon, and colleagues [2, 19]. In their work, several users assist in waste cleanup using Point-and-Direct (PAD) commands. Users point to cleanup locations in a shared image and a robot excavates each location in turn.

In [6, 5], Goldberg and Chen described an Internet-based MOSR system that averaged multiple human inputs to simultaneously control a single industrial robot arm. In [7] Goldberg, Song, et al. propose the “Spatial Dynamic Voting” (SDV) interface. The SDV collects, displays, and analyzes a sequence of spatial votes from multiple online operators at their Internet browsers. The votes drive the motion of a single mobile robot or human “Tele-Actor”.

Our problem is a nonlinear optimization problem with a non-differentiable objective function. The structure of the problem is closely related to the planar p -center Facility Location problem, which was proved to be NP-complete by Megiddo and Supowit [20]. Using a geometric approach, Eppstein [4] gave algorithm for the the planar 2-Center problem in $O(n \log^2 n)$. Halperin et al. [10] gave an algorithm for the 2-center problem with m obstacles that runs in randomized expected time $O(m \log^2(mn) + mn \log^2 n \log(mn))$.

In almost all nonlinear mathematical programming approaches, a constrained optimization problem is converted to a series of unconstrained problems using barrier or penalty methods. Line search is then used to solve the unconstrained optimization problems. Although there are many different ways of guiding search direction and step size, most of these methods are based on derivatives [22].

Rectangle-related problems in computational geometry include range searching and rectangle intersection. Agarwal and Erickson [1] provide a review of geometric range searching and its related topics. Grossi and Italiano [8, 9] proposed the cross-tree data structure, a generalized version of balanced tree, to speed up range searching in high-dimensional space. Kankanhalli and Franklin [14] developed an algorithm that can compute the area and perimeter of the union of a set of iso-oriented rectangles in parallel.

In independent work, Kimber, Liu, Foote et al describe a multi-user robot camera in [16, 17]. Similar to Sharecam, they formulate the frame selection for multiple simultaneous requests as an optimization problem based on position and area of overlap. To solve their version, they propose an approximation based on the bounding boxes of all combinations of user frames. This algorithm requires exponential time and does not provide formal bounds on approximation error.

In [25], Song, van der Stappen, and Goldberg gave the first algorithms for the ShareCam problem. We defined the Generalized Intersection Over Maximum (GIOM) metric for user “satisfaction” based on how the user’s requested frame compares with a candidate camera frame and reported an $O(n^2 m)$ exact algorithm for a continuous pan and tilt with discrete m levels of zoom. In the present paper, we relax the assumption that zoom has to be discrete and report approximation algorithms for continuous pan, tilt, and zoom, which runs in $O(n/\epsilon^3)$ time.

ShareCam Part I, A companion paper submitted separately to this conference, describes system interface, architecture, and implementation.

3 Problem Definition

In this section, we formulate the collaborative camera control problem as an optimization problem: finding the camera frame that maximizes total user satisfaction. Note that in contrast to [25], where zoom is a discrete variable, we formulate the more general version with continuous pan, tilt, and zoom parameters.

Let ϕ be a vector of parameters that users can control. In the Sharecam system, frame $\phi = [x, y, z]$, where x, y specify the center point of the input rectangle, which is corresponding to pan and tilt, and z specifies size of the rectangle, which can be used to control zoom. ϕ defines a rectangular camera frame (the camera has a fixed aspect ratio of 4:3). For frame $\phi = [x, y, z]$, the width of the frame is $4z$, the height of the frame is $3z$, and the area of the frame is $12z^2$. User i requests a desired frame ϕ_i . Given requests from n users, the system computes a single global frame ϕ^* that will best satisfy the set of requests.

Define w and h to be the width and height of the panoramic image, let $\Theta = \{(x, y) : x \in [0, w], y \in [0, h]\}$ be the set of all reachable x, y pairs. Let $Z = [z_l, z_u]$ be the range of zoom. Set $\Phi = \Theta \times Z = \{(x, y, z) | (x, y) \in \Theta, z \in Z\}$ as the feasible region of the problem.

As described in [25], user “satisfaction” is a Generalized Intersection Over Maximum (GIOM) function. It is based on how a user’s requested frame compares with a candidate camera frame. Recall that ϕ_i is the frame requested by user i , and let $\phi = [x, y, z]$ be a candidate camera frame. The

metric is a scalar $s_i \in [0, 1]$, the level of ‘‘satisfaction’’ that user i receives. User i gets no satisfaction if the candidate frame does not intersect ϕ_i : $s_i = 0$ when $\phi \cap \phi_i = \emptyset$. User i is perfectly satisfied when the candidate frame is identical to ϕ_i : $s_i = 1$ when $\phi = \phi_i$. When there is partial overlap, let a_i be the area of frame ϕ_i , p_i be the area of overlap between ϕ_i and ϕ ,

$$s_i(\phi_i, \phi) = (p_i/a_i) \min(z_i/z, 1) \quad (1)$$

If z is bigger, the candidate frame will be bigger. A sufficiently large z can define a candidate frame that covers all requested frames: $p_i/a_i = 1$ for $i = 1, \dots, n$. However, user satisfaction is not necessarily high because a user wants to see the requested frame at a desired zoom level. The term $\min(z_i/z, 1)$ characterizes this desire: it reaches its maximum of 1 if the candidate frame is the same or smaller than the requested frame. The $s_i \in [0, 1]$ is normalized.

Each of n users submits a request. Let the total user satisfaction be

$$s(\phi) = \sum_{i=1}^n s_i(\phi_i, \phi) \quad (2)$$

We want to find ϕ^* , the value of ϕ that maximizes $s(\phi)$. Since $\phi = [x, y, z]$, we now have a maximization problem: $\max_{\phi} s(\phi)$. We next present two grid-based approximation algorithms to solve it.

4 Algorithms

We begin with uniform grid-based approximation algorithm and derive formal approximation bounds that characterize the tradeoff between speed and accuracy. We then describe a distributed version of the algorithm.

4.1 Approximation Algorithm

Since requested frames are drawn by hand by each user, an approximate solution may be acceptable. We propose an algorithm that searches a regular lattice for an approximate solution $\tilde{\phi}$.

Define the lattice as the set of points with coordinates,

$$L = \{(pd, qd, rd_z) | pd \in [0, w], qd \in [0, h], rd_z \in [z_l, z_u + 2d_z], p, q, r \in \mathcal{N}\} \quad (3)$$

where d is the spacing of the pan and tilt samples, d_z is the spacing of the zoom, and p, q, r are positive integers.

To find $\tilde{\phi}$, we evaluate all $(wh/d^2)(g/d_z)$ candidate points, where $g = z_u - z_l$. According to Equation 2, it takes $O(n)$ computing time to determine the satisfaction for a single candidate frame ϕ . The total amount of computation of the algorithm is

$$O((wh/d^2)(g/d_z)n). \quad (4)$$

How good is the approximate solution in comparison to the optimal solution? Specifically, what is the tradeoff between solution quality and computation speed?

Let ϕ^* be an optimal solution. Let ϵ characterize the comparative ratio the of objective values for the two solutions:

$$s(\tilde{\phi})/s(\phi^*) = 1 - \epsilon \quad (5)$$

Since equation 2 defines a maximization problem, $s(\phi^*)$ is always greater than or equal to $s(\tilde{\phi})$ so the $0 \leq \epsilon < 1$. As $\epsilon \rightarrow 0$, $s(\tilde{\phi}) \rightarrow s(\phi^*)$.

We will establish theorems that give an upper bound c such that $\epsilon \leq c$ for given d and d_z . This characterizes the tradeoff between solution quality and computation speed. We first prove lemmas based on 2 observations:

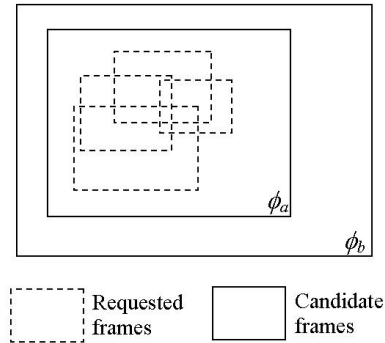


Figure 2: Example illustrating the lower bound on solution quality.

- As illustrated in Figure 2, consider a set of user requested frames ϕ_i , each with zoom level z_i . Now consider two candidate frames for the camera, ϕ_a, ϕ_b with z_a, z_b such that for all i , $\phi_i \subset \phi_a \subset \phi_b$ and $z_i < z_a < z_b$. This case provides a lower bound where $s(\phi_b)/s(\phi_a) = z_a/z_b$. In general cases, some user frames ϕ_i will be included in ϕ_b but outside ϕ_a , which will only increase the ratio.
- Now consider the smallest frame on the lattice that contains an optimal frame. Its size is a function of the size of the optimal frame z^* , d , and d_z , as derived in lemma 2.

We now prove these formally in the general case to obtain a bound on solution quality.

Lemma 1. For two candidate frames $\phi_a = [x_a, y_a, z_a]$ and $\phi_b = [x_b, y_b, z_b]$, if ϕ_a is within ϕ_b , then $\frac{s(\phi_b)}{s(\phi_a)} \geq \frac{z_a}{z_b}$.

Proof. Recall the a_i is the area of user i 's requested frame. Let

- p_{ai} be the intersected area between ϕ_a and user i 's requested frame ϕ_i
- p_{bi} be the intersected area between ϕ_b and user i 's requested frame ϕ_i . $p_{bi} \geq p_{ai}$
- S_a be the set of users whose requested frames intersect with frame ϕ_a .
- S_b be the set of users whose requested frames intersect with frame ϕ_b . $S_a \subseteq S_b$
- S' be the set of users whose requested frames intersect with optimal frame ϕ_a and are bigger than the ϕ_a , $S' = \{[x, y, z] | [x, y, z] \in S \text{ and } z \geq z_a\}$.
- S'' be the set of users whose requested frames intersect with optimal frame ϕ_a and are bigger than the ϕ_b , $S'' = \{[x, y, z] | [x, y, z] \in S \text{ and } z \geq z_b\}$. $S'' \subseteq S' \subseteq S_a$ because $z_b \geq z_a$.

we have,

$$s(\phi_a) = \sum_{i \in S} (p_{ai}/a_i) \min(z_i/z_a, 1)$$

and because $S_a \subseteq S_b$,

$$\begin{aligned} s(\phi_b) &= \sum_{i \in S_b} (p_{bi}/a_i) \min(z_i/z_b, 1) \\ &\geq \sum_{i \in S_a} (p_{bi}/a_i) \min(z_i/z_b, 1) \end{aligned}$$

Therefore, $s(\phi_b)/s(\phi_a)$

$$\begin{aligned} &\geq \frac{\sum_{i \in S_a} (p_{bi}/a_i) \min(z_i/z_b, 1)}{\sum_{i \in S_a} (p_{ai}/a_i) \min(z_i/z_a, 1)} \\ &\geq \frac{\sum_{i \in S_a} (p_{ai}/a_i) \min(z_i/z_b, 1)}{\sum_{i \in S_a} (p_{ai}/a_i) \min(z_i/z_a, 1)} \\ &= \frac{\sum_{i \in S''} (p_{ai}/a_i) + \sum_{i \in S_a - S''} (p_{ai}/a_i)(z_i/z_b)}{\sum_{i \in S'} (p_{ai}/a_i) + \sum_{i \in S_a - S'} (p_{ai}/a_i)(z_i/z_a)} \\ &\geq \frac{\sum_{i \in S_a - S''} (p_{ai}/a_i)(z_i/z_b)}{\sum_{i \in S' - S''} (p_{ai}/a_i) + \sum_{i \in S_a - S'} (p_{ai}/a_i)(z_i/z_a)} \end{aligned}$$

Define $S_l = \sum_{i \in S' - S''} (p_{ai}/a_i)(z_i/z_a)$. We know that

$$z_i/z_a \geq 1, \forall i \in S' - S''.$$

Then

$$\sum_{i \in S' - S''} (p_{ai}/a_i) \leq S_l$$

So,

$$\begin{aligned} s(\phi_b)/s(\phi_a) &\geq \frac{\sum_{i \in S_a - S''} (p_{ai}/a_i)(z_i/z_b)}{S_l + \sum_{i \in S_a - S'} (p_{ai}/a_i)(z_i/z_a)} \\ &= \frac{\sum_{i \in S_a - S''} (p_{ai}/a_i)(z_i/z_b)}{\sum_{i \in S_a - S''} (p_{ai}/a_i)(z_i/z_a)} \\ &= \frac{(1/z_b) \sum_{i \in S_a - S''} (p_{ai}/a_i) z_i}{(1/z_a) \sum_{i \in S_a - S''} (p_{ai}/a_i) z_i} \\ &= \frac{1/z_b}{1/z_a} = \frac{z_a}{z_b} \end{aligned}$$

□

Now, we are ready to find the smallest frame on the lattice that contains the optimal frame.

Lemma 2. Recall that d is the spacing of the lattice and d_z is the spacing for zoom levels. For any frame $\phi = [x, y, z] \in \Phi$, there exists $\phi^c = [x^c, y^c, z^c] \in L$ such that ϕ^c is the smallest frame on the lattice that ensures ϕ is within ϕ^c , which implies,

$$|x - x^c| \leq d/2, \quad (6)$$

$$|y - y^c| \leq d/2, \quad (7)$$

$$z^c \leq \lceil \frac{3z + d}{3d_z} \rceil d_z. \quad (8)$$

If we choose $d = 3d_z$, then

$$z^c \leq z + 2d_z. \quad (9)$$

Proof. The center (point B in figure 3) of the given frame ϕ must have four neighboring lattice points. Without loss of generality, let's assume the nearest lattice point of the center is the top right lattice point, which is point O in figure 3. Other cases can be proven by symmetric.

Since frame ϕ^c is the smallest frame on the lattice that ensure ϕ is within ϕ^c , (x^c, y^c) has to be the closest neighboring lattice point of the (x, y) on the Θ plane, which implies that equation 6 and 7 have to be true.

Recall that d is the spacing of the lattice. To ensure the point O is the nearest lattice point, equation 6 and 7 mean the point B must satisfy following constraints,

$$|\overline{OB}| \sin \alpha \leq d/2, \text{ and } |\overline{OB}| \cos \alpha \leq d/2. \quad (10)$$

Let's define frame $\hat{\phi} = [x^c, y^c, \hat{z}]$ be the smallest frame containing frame ϕ such that $(x^c, y^c) \in \Theta$ and $\hat{z} \in \mathcal{R}^+$. In other words, the frame $\hat{\phi}$ is located at (x, y) lattice but with continuous zoom \hat{z} . It is not difficult to find the relationship between ϕ^c and $\hat{\phi}$:

$$z^c = \lceil \hat{z}/d_z \rceil d_z \quad (11)$$

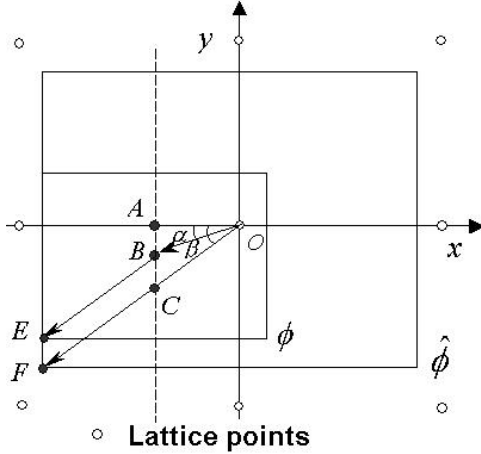


Figure 3: Relationship between frame ϕ and the smallest frame $\hat{\phi}$ on the lattice that encloses it. In the figure, $\alpha = \angle AOB$, $\beta = \angle AOC$, and the frame ϕ is centered at point B .

Since point F at (x_F, y_F) is the bottom-left corner of the frame $\hat{\phi}$ and point E at (x_E, y_E) is the bottom-left corner of the frame ϕ , the condition that the frame ϕ is located inside the frame $\hat{\phi}$ is equivalent to following conditions,

$$x_F \leq x_E, \text{ and } y_F \leq y_E. \quad (12)$$

Since the frames are iso-oriented rectangles and with same aspect ratio, their diagonal lines have to be parallel to each other:

$$\overline{BE} \parallel \overline{OF}$$

Therefore, when $0 \leq \alpha \leq \beta$, \overline{BE} is always on top of \overline{OF} , if $x_F = x_E$, then $y_F \leq y_E$. The boundary conditions for the $\hat{\phi}$ can be simplified:

- case 1: $x_F = x_E$ if $0 \leq \alpha \leq \beta$, and
- case 2: $y_F = y_E$ if $\beta \leq \alpha \leq \pi/2$.

Figure 3 describes case 1. We draw a vertical line at point B , which intersects x axis at point A and \overline{OF} at point C . Since $x_F = x_E$, we know $\overline{EF} \parallel \overline{AC}$. Therefore, we have $|\overline{CF}| = |\overline{BE}|$ and

$$|\overline{OC}| = |\overline{OF}| - |\overline{CF}| = |\overline{OF}| - |\overline{BE}|. \quad (13)$$

Also, since $\overline{AC} \perp \overline{OA}$, we have,

$$\begin{aligned} |\overline{OC}| \cos \beta &= |\overline{OB}| \cos \alpha \\ \Rightarrow (|\overline{OF}| - |\overline{BE}|) \cos \beta &= |\overline{OB}| \cos \alpha \end{aligned}$$

According to equation 10,

$$\Rightarrow (|\overline{OF}| - |\overline{BE}|) \cos \beta \leq d/2$$

The aspect ration of the frame is $4 : 3 \Rightarrow \cos \beta = 4/5$.

$$\Rightarrow |\overline{OF}| \leq |\overline{BE}| + 5d/8$$

Similarly, we can get $|\overline{OF}| \leq |\overline{BE}| + 5d/6$ from case 2. Combine two cases, we know, $|\overline{OF}| \leq |\overline{BE}| + 5d/6$. Since $|\overline{OF}| = 5\hat{z}/2$ and $|\overline{BE}| = 5z/2$,

$$\hat{z} \leq z + d/3.$$

Plug it into equation 11,

$$z^c \leq \lceil \frac{3z + d}{3d_z} \rceil d_z.$$

If we choose $d = 3d_z$, equation 8 can be simplified as,

$$z^c \leq \lceil \frac{z}{d_z} \rceil d_z + d_z \leq z + 2d_z$$

□

Theorem 1. Let z_{min} be the smallest z among all requested frames and $d = 3d_z$. The approximation factor of the deterministic lattice based algorithm ϵ is bounded below by some constant c , $0 \leq \epsilon \leq c$, where

$$c = \frac{2d_z}{z_{min} + 2d_z}$$

Proof. Recall that,

- $\phi^* = [x^*, y^*, z^*]$ be the optimal frame,
- $\phi^c = [x^c, y^c, z^c]$ be the closest lattice point with the smallest zoom level that ensure ϕ^* is within ϕ^c .
- $\tilde{\phi} = [\tilde{x}, \tilde{y}, \tilde{z}]$ be the lattice point found by the approximation algorithm,

Note that $\tilde{\phi}$ is the solution to:

$$\max_{\phi \in L} s(\phi).$$

Since $\phi^c \in L \subset \Phi$, we know that $s(\phi^c) \leq s(\tilde{\phi})$. Therefore

$$s(\phi^c) \leq s(\tilde{\phi}) \leq s(\phi^*)$$

Therefore, applying Lemma 1,

$$1 - \epsilon = s(\tilde{\phi})/s(\phi^*) \geq s(\phi^c)/s(\phi^*) \geq \frac{z^*}{z^c}.$$

Apply equation 9 of lemma 2, we have

$$z^c \leq z^* + 2d_z.$$

Using this result, we have,

$$1 - \epsilon \geq \frac{z^*}{z^* + 2d_z}$$

On the other hand, we know $z^* \geq z_{min}$, so

$$1 - \epsilon \geq \frac{z_{min}}{z_{min} + 2d_z} \leftrightarrow \epsilon \leq \frac{2d_z}{z_{min} + 2d_z}$$

□

Theorem 1 says that the approximation bound $\frac{2d_z}{z_{min} + 2d_z}$ is a monotonic increasing function of d_z . It characterizes the tradeoff between accuracy and computation speed for uniform grid-based approximation algorithm:

- i) For a given approximation bound ϵ , compute the appropriate lattice spacing: choose $d = 3d_z$, according to theorem 1, we set

$$\epsilon = \frac{2d_z}{z_{min} + 2d_z} \Rightarrow d_z = \frac{1}{2} \left(\frac{\epsilon}{1 - \epsilon} \right) z_{min}$$

This is the maximum d_z that ensures the objective function value is bounded above $(1 - \epsilon)s(\phi^*)$.

- ii) Compute the objective function value for each lattice point, output the the lattice point with the largest objective function value as the approximated solution.

The relationship between solution quality and computation speed is summarized by theorem 2.

Theorem 2. *We can solve the Sharecam problem in $O(n/\epsilon^3)$ for a given approximation bound ϵ .*

Proof. Since $d = 3d_z$ and $d_z = \frac{1}{2} \left(\frac{\epsilon}{1 - \epsilon} \right) z_{min}$, we need to evaluate all $(wh/d^2)(g/d_z) = \frac{whg}{(9/4) \left(\frac{\epsilon}{1 - \epsilon} \right)^3 z_{min}^3}$ points. According to equation 2, each point will take $O(n)$ time. Removing constants, ϵ approaches zero, so computation time approaches $O(n/\epsilon^3)$ □

4.2 Distributed algorithm

In the ShareCam system, n is the number of users online, which is also the number of computers connected to our server. The larger the value of n , the more computation power in the system. This suggests that a distributed algorithm, where each client shares in the computation, can reduce overall computation time. In particular, we propose an algorithm where each client searches a coarse lattice with appropriate offsets. The more clients that participate, the more lattice points that are searched. The algorithm described in the previous section can be divided into client and server components by dividing the lattice L into n sub lattices L_i , $i = 1, \dots, n$, where sub lattice L_i will be searched by user i .

Recall that $[z_l, z_u]$ is the zoom range, define

$$o_i = (i - 1)d_z + z_l$$

be initial zoom offset for user i . Then the sub lattice L_i is:

$$L_i = \{(pd, qd, rnd_z + o_i) | pd \in [0, w], qd \in [0, h], rnd_z + o_i \in [z_l, z_u + 2d_z], p, q, r \in \mathcal{N}\}.$$

Therefore, $\bigcup_{i=1}^n L_i = L$.

4.2.1 Server algorithm.

Let s_i^* be the optimal solution given by i^{th} user, the server should do the following.

- i). Compute the d_z , set $d = 3d_z$,
- ii). Send all d_z , d , and all requested frames to clients,
- iii). Wait until all clients send their solutions $\{s_1^*, \dots, s_n^*\}$ back and pick the largest.

4.2.2 Client algorithm.

The i^{th} user should do following after receives d_z , d , and requested frames from server,

- i). Compute zoom offset o_i .
- ii). Evaluate objective function with respect to sub lattice L_i .
- iii) Send the s_i^* , the optimal on lattice L_i to server.

Since each sub lattice only contains $1/n$ points of L , the computation complexity is,

Theorem 3. *Each client runs in $O(1/\epsilon^3)$ time and the server runs in $O(n + 1/\epsilon^3)$ time.*

4.2.3 Performance with network client failures.

We distribute computation to client computers under the risk that some clients may drop out the system in the middle of the computation. One can also see from the algorithm that the speed of computation is limited by the slowest client. We may not have the luxury to wait for the slowest client to send his/her result back to server. Therefore, we are interested in the algorithm performance if one or more clients fail to submit their computation results before a time-out.

Theorem 4. *If one client fails to submit his/her result, the approximation bound ϵ will gracefully increase from $\frac{2d_z}{z_{min} + 2d_z}$ to $\frac{3d_z}{z_{min} + 3d_z}$.*

Proof. Without loss of generality, let's assume user i drops out. Let $\phi^c = [x^c, y^c, z^c] \in L - L_i$ be the smallest frame that ensures ϕ^* is within ϕ^c and $\tilde{\phi}$ be the optimal found on lattice $L - L_i$. We consider two scenarios:

- i). $\phi^c \notin L_i$. Then $1 - \epsilon = \frac{s(\tilde{\phi})}{s(\phi^*)} \geq \frac{s(\phi^c)}{s(\phi^*)} \geq \frac{z_{min}}{z_{min} + 2d_z} > \frac{z_{min}}{z_{min} + 3d_z}$ still holds. Therefore $\epsilon < \frac{3d_z}{z_{min} + 3d_z}$.

ii). $\phi^c \in L_i$. $\frac{s(\tilde{\phi})}{s(\phi^*)} \geq \frac{s(\phi^c)}{s(\phi^*)}$ is invariant. From the result of Lemma 1, we know $\frac{s(\phi^c)}{s(\phi^*)} \geq \frac{z^*}{z^c}$. Since $\phi^c \in L_i$, equation 11 will not hold. Instead, for any frame $\phi^c \in L_i$,

$$z^c = \lceil \hat{z}/d_z \rceil d_z + d_z.$$

Then

$$z^c \leq z + 3d_z.$$

Similar to the proof of Theorem 1, we have

$$1 - \epsilon = \frac{s(\tilde{\phi})}{s(\phi^*)} \geq \frac{z_{min}}{z_{min} + 3d_z}.$$

Hence,

$$\epsilon \leq \frac{3d_z}{z_{min} + 3d_z}$$

□

It is good to see that the approximation bound does not change dramatically after one client drops out of the system. In fact, the proof of theorem 4 shows that the approximation bound change is very small. We can also extend the proof to cases where more than one user drop out. Numerical results are shown in section 4.3.

4.3 Algorithm Results

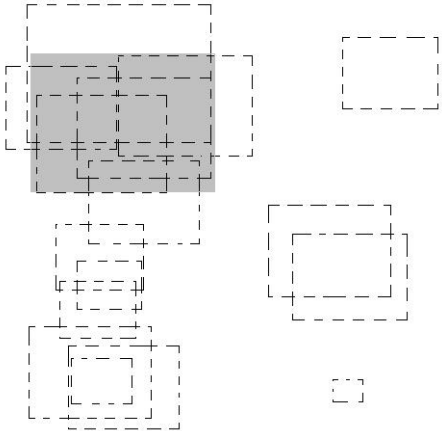


Figure 4: Sample output of algorithm with 16 requested frames and $d_z = 2$.

Figure 4 shows the algorithm's result for a testing case containing 16 requested frames. It is interesting to see that this algorithm functions like a combination of "clustering" algorithm for rectangles with different location and size and an algorithm that identifies the region of panoramic view with the highest requested frame density.

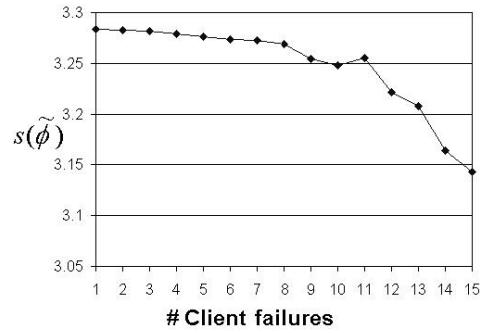


Figure 5: Performance of the Distributed Algorithm: solution quality as network clients fail. Each data point is an average of 10 runs with random dropouts. Initially there are 16 clients and 16 requested frames as shown in the previous figure. Here we plot solution quality as more and more clients fail to complete their lattice searches. Note that solution quality decreases, but not dramatically: even when only one client remains (after 15 have failed) searching only one coarse lattice is enough to find a reasonable solution.

Figure 5 shows how solution quality decreases as network clients fail, based on the example in figure 4. This supports the analysis in the previous section: the approximation algorithm degrades gracefully with client failures.

5 Conclusions and Future Work

We present new algorithms for the ShareCam Problem: controlling a single robotic pan, tilt, zoom camera based on simultaneous frame requests from n online users. We formalize the problem with continuous pan, tilt, and zoom. With approximation bound ϵ , the algorithm runs in $O(n/\epsilon^3)$ time.

We also show that the algorithm can be distributed to run in $O(1/\epsilon^3)$ time at each client and in $O(n+1/\epsilon^3)$ time at the server. Unlike computing with multiple processors in a single supercomputer, distributed computing over the Internet requires input from a variety of heterogenous processors, each with different and varying communication delays. We show that the grid-based algorithm handles client failures gracefully.

In future work, we will explore how these algorithms can be further improved with Brand and Bound techniques and advanced data structures.

A preliminary version of the Sharecam system went online in Sep. 2002 at <http://tele-actor.net/sharecam>. System interface, architecture, and implementation are reported in Part I. The system may be offline for a short period in April as we move the camera to an outside campus location for testing.

Acknowledgments

We are grateful to Frank van der Stappen, Sarel Har-Peled, Vladlen Koltun, Dana Plautz, Cory Cox, and Eric Paulos for insightful discussions and feedback. Thanks to M. Faldu and W. Guan for their contributions to the ShareCam system, and I.Y. Song, K. “Gopal” Gopalakrishnan, Ron Alterovitz, A. Levandowski, M. Mckelvin, A. Ho, and J. Vidales for helpful suggestions.

References

- [1] P. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry, volume 23 of Contemporary Mathematics*.
- [2] D. J. Cannon. *Point-And-Direct Telerobotics: Object Level Strategic Supervisory Control in Unstructured Interactive Human-Machine System Environments*. PhD thesis, Stanford Mechanical Engineering, June 1992.
- [3] N. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, and K. Tanie. Remote coordinated controls in multiple telerobot cooperation. In *IEEE International Conference on Robotics and Automation*, April 2000.
- [4] D. Eppstein. Fast construction of planar two-centers. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, 1997.
- [5] K. Goldberg and B. Chen. Collaborative control of robot motion: Robustness to error. In *International Conference on Intelligent Robots and Systems (IROS)*, 2001.
- [6] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, and G. Smith. Collaborative teleoperation via the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, April 2000.
- [7] K. Goldberg, D. Song, Y. Khor, D. Pescovitz, A. Levandowski, J. Himmelstein, J. Shih, A. Ho, E. Paulos, and J. Donath. Collaborative online teleoperation with spatial dynamic voting and a human “tele-actor”. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2002.
- [8] R. Grossi and G. F. Italiano. Efficient cross-trees for external memory. In James Abello and Jeffrey Scott Vitter, editors, *External Memory Algorithms and Visualization*, pages 87–106. American Mathematical Society Press, Providence, RI, 1999.
- [9] R. Grossi and G. F. Italiano. Revised version of “efficient cross-trees for external memory”. Technical Report TR-00-16, 10 2000.
- [10] D. Halperin, M. Shairir, and K. Goldberg. The 2-center problem with obstacles. *Journal of Algorithms*, 32, 2002.
- [11] K. Han, Y. Kim, J. Kim, and S.Hsia. Internet control of personal robot between kaist and uc davis. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
- [12] H. Hu, L. Yu, P. Tsui, and Q. Zhou. Internet-based robotic systems for teleoperation. *Assembly Automation Journal*, 21(2), 2001.
- [13] S. Jia, Y. Hada, G. Ye, and K. Takase. Distributed telecare robotic systems using corba as a communication architecture. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
- [14] M.S. Kankanhalli and W. R. Franklin. Area and perimeter computation of the union of a set of iso-rectangles in parallel. *Journal of Parallel and Distributed Computing*, 27, 1995.
- [15] J. Kim, B. Choi, S. Park, K. Kim, and S. Ko. Remote control system using real-time mpeg-4 streaming technology for mobile robot. In *IEEE International Conference on Consumer Electronics*, 2002.
- [16] D. Kimber, Q. Liu, J. Foote, and L. Wilcox. Capturing and presenting shared multi-resolution video. In *SPIE ITCOM 2002. Proceeding of SPIE, Boston*, volume 4862, pages 261–271, Jul. 2002.
- [17] Q. Liu, D. Kimber, J. Foote, L. Wilcox, and J. Boreczky. Flyspec: A multi-user video camera system with hybrid human and automatic control. In *ACM Multimedia 2002, France*, Dec. 2002.
- [18] R.C. Luo and T. M. Chen. Development of a multibehavior-based mobile robot for remote supervisory control through the internet. *IEEE/ASME Transactions on Mechatronics*, 5(4), 2000.
- [19] M. McDonald, D. Small, C. Graves, and D. Cannon. Virtual collaborative control to improve intelligent robotic system efficiency and quality. In *IEEE International Conference on Robotics and Automation*, April 1997.
- [20] N. Megiddo and K.J. Supowit. on the complexity of some common geometric location problems. *SIAM J. Comput.*, 13, 1984.
- [21] T. Mirfakhrai and S. Payandeh. A delay prediction approach for teleoperation over the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
- [22] S.G. Nash and A. Sofer, editors. *Linear and Nonlinear Programming*. The McGraw-Hill Companies, Inc, 1996.
- [23] L. Ngai, W.S. Newman, and V. Liberatore. An experiment in internet-based, human-assisted robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.
- [24] R. Safaric, M. Debevc, R. Parkin, and S. Uran. Telerobotics experiments via internet. *IEEE Transactions on Industrial Electronics*, 48(2):424–31, April 2001.
- [25] D. Song, A. F. van der Stappen, and K. Goldberg. Exact and distributed algorithms for collaborative camera control. In *The Workshop on Algorithmic Foundations of Robotics, December*, Dec. 2002.