An Approximation Algorithm for the Least Overlapping p-Frame Problem with Non-Partial Coverage for Networked Robotic Cameras

Yiliang Xu, Dezhen Song, Jingang Yi, and A. Frank van der Stappen

***Abstract*— We report our algorithmic development of the p-frame problem that addresses the need of coordinating a set of p networked robotic pan-tilt-zoom cameras for n, $(n > p)$, competing polygonal requests. We assume that the p frames have almost no overlap on the coverage between frames and a request is satisfied only if it is fully covered. We then propose a Resolution Ratio with Non-Partial Coverage (RRNPC) metric to quantify the satisfaction level for a given request with respect to a set of p candidate frames. We propose a lattice-based approximation algorithm to search for the solution that maximizes the overall satisfaction. The algorithm builds on an induction-like approach that finds the relationship between the solution to the $(p-1)$-frame problem and the solution to the p-frame problem. For a given approximation bound ϵ, the algorithm runs in $O(n/\epsilon^3 + p^2/\epsilon^6)$ time. We have implemented the algorithm and experimental results are consistent with our complexity analysis.**

I. Introduction

Networked robotic pan-tilt-zoom cameras have found many applications such as natural environment observation, surveillance, and distance learning. Consider that a group of p networked robotic pan-tilt-zoom cameras have been installed for public surveillance in a popular location such as Time Square in New York city. There are n different concurrent requests initiated by a variety of sources such as networked chemical sensors, online user requests, and scheduled events. Fig. 1 illustrates the p-frame problem: how to identify optimal p frames that best satisfy the n different polygonal requests.

We assume that the p frames have the least overlap (will be formally defined later) on the coverage between the frames and a request is satisfied only if it is fully covered by one of the p frames. Under the assumptions, we propose a Resolution Ratio with Non-Partial Coverage (RRNPC) metric to quantify the satisfaction level for a given request with respect to a set of p candidate frames. Hence the p-frame problem is to find the optimal set of (up to p) frames that maximizes the overall satisfaction. Building on the results in [1], we propose a lattice-based approximation algorithm. The algorithm builds on an induction-like approach that finds the relationship between the solution to the $p-1$ frame problem and the solution to the p-frame problem. For a given approximation bound ϵ, the algorithm runs in $O(n/\epsilon^3 + p^2/\epsilon^6)$ time. We have implemented the algorithm and experiment results are consistent with our complexity analysis. We will begin with the related work.

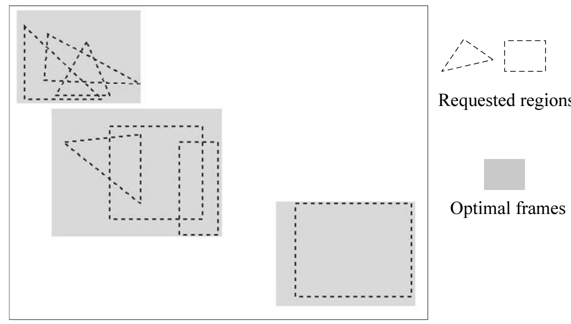


Fig. 1. An illustration of the least overlapping 3-frame problem.

II. Related Work

The p-frame problem relates to networked robotics, the facility location problem in operations research, and the single frame selection problem.

The development of the Internet allows more users to access online resources. The p frames taken by p networked pan-tilt-zoom cameras can be viewed as a special case of networked tele-operation, where each robotic camera has 3 Degrees of Freedom (DOF). According to the taxonomy proposed by Chong et al. [2], this system belongs to Multiple Operator Multiple Robot (MOMR) systems. The low cost robot and sensor network makes the MOMR system a very popular research domain [3]–[5]. In [6], [7], Liu and his colleagues developed a competitive MOMR system under a game setting. Our work emphasizes on the geometric coverage attributes of the robotic camera and addresses the MOMR problem in an optimization framework.

The p-frame problem is structurally similar to the p-center facility location problem, which has been proven to be NP-complete [8]. Given n request points on a plane, the task is to optimally allocate p points as service centers to minimize the maximum distance (called min-max version) between any request point and its corresponding service center. In [9], an $O(n \log^2 n)$ algorithm for a 2-center problem is proposed. As an extension, replacing service points by orthogonal boxes, Arkin et al. [10] propose a $(1+\epsilon)$-approximation algorithm that runs in $O(n \min(\lg n, 1/\epsilon) + (\lg n)/\epsilon^2)$ for the 2-box covering problem. Alt et al. [11] proposed a $(1+\epsilon)$-approximation algorithm that runs in $O(n^{O(m)})$,

This work was supported in part by the National Science Foundation under IIS-0534848 and 0643298.

Y. Xu and D. Song are with Computer Science Department, Texas A&M University, College Station, TX 77843. Email: ylxu@cs.tamu.edu and dzsong@cs.tamu.edu.

J. Yi is with Mechanical Engineering Department, San Diego State University, San Diego, CA 92182. Email: jgyi@mail.sdsu.edu.

A.F. van der Stappen is with Department of Information and Computing Sciences, Utrecht University, PO Box 80089, 3508 TB Utrecht, The Netherlands. Email: frankst@cs.uu.nl.

where $\epsilon = O(1/m)$, for the multiple disk covering problem. The requests in these problems are all points instead of polygonal regions as those in our p-frame problem and the objective of the p-frame problem is to maximize the satisfaction, which is not a distance metric.

The p-frame problem also relates to the art gallery problem [12]. The art gallery problem is to minimize the number of security guards to guard an art gallery, which is usually represented by a polygon with n vertices. Each guard has a certain range of vision. The location of the guard can be represented by a point while the reachable region of the guard can be represented by any geometrical shapes. Agarwal et al. [13] consider a variation of the art gallery problem where the terrain is not planar and there are only two guards with minimal heights. They propose an exact algorithm that runs in $O(n^2 \log^4 n)$ time. In [14], Eppstein et al. propose the sculpture garden problem where each guard has only a limited angle of visibility. They prove that the upper bound is $n-2$ and the lower bound is $n/2$ for the number of the guards needed. More results on the art gallery problem can be found in [15]. Unlike the art gallery problem, the p-frame problem does not need to cover all requests. However, the selection has to be made based on maximizing the level of satisfaction of covered requests.

Our group has worked on camera frame selection problems since 2002. We have addressed the Single Frame Selection (SFS) problem and its variations such as approximate solution with continuous zoom [16], approximate solution with fixed zoom [17], and exact solution with continuous zoom and rectangular requests with fixed aspect ratio [18] or variable aspect ratio [19]. Extending the results for SFS to the p-frame problem is non-trivial. This paper is the first attempt to tackle the problem.

III. Problem Definition

In this section, we formulate the p-frame problem. We begin with the definition of the inputs and outputs. Assumptions are then presented. We establish the request satisfaction metric so that we can formulate the problem as a geometric optimization problem.

A. Input and Output

The input of the problem is a set of n requests $R = \{r_i | i = 1, 2, ..., n\}$. Each request is defined as $r_i = [T_i, z_i]$, where T_i denotes the polygonal requested region and $z_i \in Z$ specifies the desired resolution level, which is in the range of $Z = [\underline{z}, \overline{z}]$. The only requirement for T_i is that its coverage area can be computed in constant time.

A solution to the p-frame problem is a set of p camera frames. Given a fixed aspect ratio (e.g. 4:3), a camera frame can be defined as $c = [x, y, z]$, where pair (x, y) denotes the center point of the rectangular frame and $z \in Z$ specifies the resolution level of the camera frame. Here we consider the coverage of the camera as rectangular according to the camera configuration space. Therefore, the width and height of the camera frame can be represented as $4z$ and $3z$ respectively. The coverage area of the frame is $12z^2$. The four corners of the frame are located at $(x \pm \frac{4z}{2}, y \pm \frac{3z}{2})$.

Given w and h are the camera pan-tilt ranges respectively, then $\mathbb{C} = [0, w] \times [0, h] \times Z$ defines the set of all candidate frames. Therefore, $\mathbb{C}^p$ indicates the solution space for the p-frame problem. We define any candidate solution to the p-frame problem as $C^p = (c_1, c_2, ..., c_p) \in \mathbb{C}^p$, where $c_i, i = 1, 2, ..., p$, indicates the i-th camera frame in the solution. In the rest of the paper, we use superscription * to indicate the optimal solution. The objective of the p-frame problem is to find the optimal solution $C^{p*} = (c_1^*, c_2^*, ..., c_p^*) \in \mathbb{C}^p$ that best satisfies the requests.

B. Set Operators

We clarify the use of set operators such as "$\cap$", "$\subseteq$ " and "$\notin$" to represent the relationship between frames, frame sets, and requests in the rest of the paper.

- When two operands are frames or requests (e.g., $r_i \in R, c_u, c_v \in \mathbb{C}$), the set operators represent the 2-D regional relationship between them. For example, $r_i \subseteq c_u$ represents that the region of r_i is fully contained in that of frame c_u while $c_u \cap c_v$ represents the overlapping region of frames c_u and c_v .
- When the operands are one frame (e.g., $c_i \in \mathbb{C}$) and one frame set (e.g., $C^k \in \mathbb{C}^k, k < p$), we treat the frame as an element of a frame set. For example, $c_i \notin C^k$ represents that c_i is not an element frame in the frame set C^k.
- When the operands are two frame sets, we use set operators. For example, $\{c_1\} \subset C^p$ means frame set $\{c_1\}$ is a subset of C^p. Frame set $\{c_1, c_2\} = \{c_1\} \cup \{c_2\}$ is different from $c_1 \cup c_2$. The former is the frame set that consists of two element frames and the later is the union area of the two frames.

C. Assumptions

We assume that the p-frames are either taken from p cameras that share the same workspace or taken from the same camera. Therefore, if a location can be covered by a frame, other frames can cover that location, too.

We assume that the solution C^{p*} to the p-frame problem satisfies the following condition.

Definition 1 (Least Overlapping Condition (LOC)): $\forall r_i$, $i = 1, ...n$, $\forall c_u \in C^{p*}$, $\forall c_v \in C^{p*}$, and $c_u \neq c_v$,

$$r_i \nsubseteq c_u \cap c_v. \tag{1}$$

The LOC means that the overlap between frames is so small that no request can be fully covered by more than one frame simultaneously. The LOC forces the overall coverage of a p-frame set $\cup_{j=1}^{p} c_j$ to be close to the maximum. This is meaningful in applications when the cameras need to search for unexpected events while best satisfying the n existing requests because the ability to search is usually proportional to the union of overall coverage. Therefore, the LOC can increase the capability of searching for unexpected events. The extreme case of the LOC is that there is no overlap between camera frames.

Definition 2 (Non-Overlapping Condition (NOC)): Given a p-frame set $C^p = (c_1, c_2, ..., c_p) \in \mathbb{C}^p$ ($p \geq 2$), C^p satisfies the NOC, if

$$\forall u = 1, 2, ..., p, \forall v = 1, 2, ..., p, u \neq v, c_u \cap c_v = \phi \ .$$

It is not difficult to find that the NOC is a sufficient condition to the LOC. The NOC yields the maximum union coverage and is a favorable solution to applications where searching ability is important.

D. Satisfaction Metric

To measure how well a p-frame set satisfies the requests, we need to define a satisfaction metric. We extend the Coverage-Resolution Ratio (CRR) metric in [19] and propose a new Resolution Ratio with Non-Partial Coverage (RRNPC).

Definition 3 (RRNPC metric): Given a request $r_i = [T_i, z_i]$ and a camera frame $c = [x, y, z]$, the satisfaction of request r_i with respect to c is computed as

$$s(c, r_i) = I(c, r_i) \cdot \min(\frac{z_i}{z}, 1), \tag{2}$$

where $I(c, r_i)$ is an indicator function that describes the non-partial coverage condition,

$$I(c, r_i) = \begin{cases} 1 & \text{if} \quad r_i \subseteq c, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Eq. (3) indicates that we do not accept partial coverage over the request. Only the requests completely contained in a camera frame contribute to the overall satisfaction. From (2) and (3), the satisfaction of the ith request is a scalar $s_i \in [0, 1]$.

Based on (2), the satisfaction of r_i with respect to a candidate least overlapping p-frame set $C^p = (c_1, c_2, ..., c_p) \in \mathbb{C}^p$ is,

$$s(C^p, r_i) = \sum_{u=1}^{p} I(c_u, r_i) \cdot \min(\frac{z_i}{z_u}, 1), \tag{4}$$

where z_i, z_u indicate the resolution values of r_i and the u-th camera frame in C^p respectively. The LOC implies that although (4) is in the form of summation, at most one frame contains the region of request r_i and thus non-negative $s(C^p, r_i)$ has a maximum value of 1. Therefore, RRNPC is a standardized metric that takes both the region coverage and the resolution level into account.

To simplify the notation, we use $s(c) = \sum_{i=1}^{n} s(c, r_i)$ to represent the overall satisfaction of a single frame c. We also use $s(C^k) = \sum_{j=1, c_j \in C^k}^{k} s(c_j)$, to represent the overall satisfaction of a partial candidate k-frame set C^k, $k < p$.

E. Problem Formulation

Based on the assumption and the RRNPC metric definition above, the overall satisfaction of a p-frame set $C^p = \{c_1, c_2, ..., c_p\} \in \mathbb{C}^p$ over n requests is the sum of the satisfaction of each individual request $r_i, i = 1, 2, ..., n$,

$$s(C^p) = \sum_{i=1}^{n} \sum_{u=1}^{p} I(c_u, r_i) \cdot \min(\frac{z_i}{z_u}, 1). \tag{5}$$

Eq. (5) shows that the satisfaction of any candidate C^p can be computed in $O(pn)$ time. Now we can formulate the least overlapping p-frame problem as a maximization problem,

$$C^{p*} = \arg \max_{C^p \in \mathbb{C}^p} s(C^p). \tag{6}$$

IV. Algorithm

Solving the optimization problem in (6) is nontrivial. To enumerate all possible combinations of candidate solutions by brute force can easily take up to $O(n^p)$ time. In this section, we present a lattice-based approximation algorithm beginning with the construction of the lattice. To maintain the LOC in the lattice framework, we introduce the Virtual Non-Overlapping Condition(VNOC). Based on the VNOC, we analyze the structure of the approximate solution and derive the approximation bound with respect to the optimal solution that satisfies the NOC . To summarize this, a lattice-based induction-like algorithm is presented at the end of the section.

A. Construction of Lattice

We construct a regular 3-D lattice, which is inherited from [1] to discretize the solution space $\mathbb{C}^p$. Let 2-D point set $V = \{(\alpha d, \beta d) | \alpha d \in [0, w], \beta d \in [0, h], \alpha, \beta \in \mathcal{N})$ discretize the 2-D reachable region and represent all candidate center points of rectangular frames, where d is the spacing of the pan and tilt samples. Let 1-D point set $\mathcal{Z} = \{\gamma d_z | \gamma d_z \in [\underline{z}, \overline{z} + 2d_z], \gamma \in \mathcal{N}\}$ discretize the feasible resolution range and represent all candidate resolution values for the camera, where d_z is the spacing of the zoom. Therefore, we can construct the lattice as a set of 3-D points, $L = V \times \mathcal{Z}$.

Each point $c = (\alpha d, \beta d, \gamma d_z) \in L$ represents the setting of a candidate camera frame. There are totally $(wh/d^2)(g/d_z) = |L|$ candidate points/frames in L, where $g = \overline{z} - \underline{z}$. We set $d_z = d/3$ for cameras with an aspect ration of $4:3$ according to [1].

What is new is that the spacing of the lattice d and d_z also depends on the size of the requested regions. For any request $r_i \in R$, there exists an Iso-oriented Bounding Box (IBB) for each r_i. Let us define λ and μ as the smallest width and height across all IBBs, respectively. We choose d such that

$$d < \min(3\lambda/10, \mu/3). \tag{7}$$

This input-sensitive lattice setting can help us to establish the LOC on the lattice and will be discussed later in Section IV-B. From here on, we use symbol $\tilde{}$ to denote the lattice-based notations. For example, $\tilde{C}^p$ denotes a p-frame set on lattice L.

Definition 4: For any camera frame $c \in \mathbb{C}$,

$$\tilde{c}' = \min \tilde{c}, \text{ s.t. } \tilde{c} \in L \text{ and } c \subseteq \tilde{c}.$$

Hence $\tilde{c}'$ is the smallest frame on the lattice that fully encloses c.

In the rest of the paper, we use symbol $'$ to denote the corresponding smallest frame(s) on the lattice. For any camera frame $c = [x, y, z]$ and its corresponding $\tilde{c}' = [\tilde{x}', \tilde{y}', \tilde{z}']$, we define their bottom-left corners as (x^l, y^b) and

$(\tilde{x}'^l, \tilde{y}'^b)$, and their top-right corners as (x^r, y^t) and $(\tilde{x}'^r, \tilde{y}'^t)$, respectively.

From the results of [1], we have

$$\begin{aligned} &x^l - \tilde{x}'^l \leq 5d/3, \quad \tilde{x}'^r - x^r \leq 5d/3, \\ &y^b - \tilde{y}'^b \leq 3d/2, \quad \tilde{y}'^t - y^t \leq 3d/2. \end{aligned} \tag{8}$$

B. Virtual Non-Overlapping Condition

The NOC defined in Definition 2 guarantees the LOC. However, due to the limitation of lattice spacing, it is very difficult for candidate frames on the lattice to follow the NOC. Actually, it is unnecessary (though sufficient) to follow the NOC to satisfy the LOC. It is possible to allow a minimum overlap that is controlled by the lattice spacing and meanwhile guarantee that the LOC is still satisfied, which yields the Virtual Non-Overlapping Condition (VNOC).

Definition 5 (Virtual Non-Overlapping Condition(VNOC)): Given any j-frame set $C^j = (c_1, c_2, ..., c_j) \in \mathbb{C}^j, j = 2, 3, ..., p$ and any two frames $c_u, c_v \in C^j$, then C^j satisfies the VNOC, if $\min(x_u^r - x_v^l, x_v^r - x_u^l) \leq 10d/3$ or $\min(y_u^t - y_v^b, y_v^t - y_u^b) \leq 3d$.

Corollary 1: Given any two frames $c_1, c_2 \in \mathbb{C}$, if $\{c_1, c_2\}$ satisfies the VNOC, then $\{c_1, c_2\}$ also satisfies the LOC.

Proof: From the definition of VNOC and the settings of λ and μ, we see that the size of the overlapping region $c_1 \cap c_2$, on either the x-axis or y-axis, is less than the size of the smallest request. This guarantees that no requested region is fully contained in the overlapping region. Therefore, the LOC is satisfied. ■

Lemma 1: Given any two frames $c_1, c_2 \in \mathbb{C}$ such that $\{c_1, c_2\}$ satisfies the VNOC, then

$$s(\{c_1, c_2\}) = s(c_1) + s(c_2). \tag{9}$$

Proof: From Corollary 1, $\{c_1, c_2\}$ satisfies the LOC. From the definition of the LOC and the RRNPC satisfaction metric defined in (2), the conclusion follows. ■

C. Approximation Solution Bound

The construction of the lattice allows us to search for the best p frames on the lattice, which yields an approximation solution. Furthermore, the VNOC and Lemma 1 assist us in deriving the approximation bound.

Lemma 2: For any two frames $c_1, c_2 \in \mathbb{C}$, if $\{c_1, c_2\}$ satisfies the NOC, then $\{\tilde{c}_1', \tilde{c}_2'\}$ satisfies the VNOC.

The proof of the lemma is trivial based on the definition of VNOC and the settings of λ and μ.

Given the optimal solution $C^{p*} = (c_1^*, c_2^*, ..., c_p^*)$ for the optimization problem defined in (6) that satisfies the NOC, there is a solution on the lattice $\tilde{C}'^{p*} = (\tilde{c}_1'^*, \tilde{c}_2'^*, ..., \tilde{c}_p'^*)$ whose element frames are the corresponding smallest frames on the lattice that contain those of C^{p*}. Lemma 2 implies that $\tilde{C}'^{p*}$ exists and satisfies the VNOC. However, how good is this solution in comparison to the optimal solution? We define the approximation bound ϵ which characterizes the comparative ratio of the approximation solution to the optimal solution

$$s(\tilde{C}'^{p*})/s(C^{p*}) \geq 1 - \epsilon. \tag{10}$$

Based on Lemma 1 and Theorem 1 in [1], we have

$$s(\tilde{C}'^{p*})/s(C^{p*}) \geq 1 - \frac{2d_z}{\underline{z} + 2d_z}. \tag{11}$$

Let $\tilde{C}^{p*}$ denote the optimal p-frame set on the lattice. Since $\tilde{C}'^{p*}$ is one of the p-frame sets on the lattice, then we have

$$\frac{s(\tilde{C}^{p*})}{s(C^{p*})} \geq \frac{s(\tilde{C}'^{p*})}{s(C^{p*})} \geq 1 - \frac{2d_z}{\underline{z} + 2d_z}. \tag{12}$$

Eq. (12) implies that we can use the solution $\tilde{C}^{p*}$ as the approximate solution to the optimal solution. Let the approximation bound be

$$\epsilon = \frac{2d_z}{\underline{z} + 2d_z}. \tag{13}$$

Solving (13) and combining the upper bound value of d as in (7), we have

$$d = 3d_z = \min(\frac{3}{2}(\frac{\epsilon}{1-\epsilon})\underline{z}, \min(3\lambda/10, \mu/3)). \tag{14}$$

Eq. (14) indicates that when $\epsilon \to 0$,

$$d = 3d_z = \frac{3}{2}(\frac{\epsilon}{1-\epsilon})\underline{z}. \tag{15}$$

Eqs. (13) and (15) imply that we can control the quality of the approximate solution by tuning the lattice spacing d. On the other hand, based on the lattice structure and the definition of the approximation bound, we know that the number of all candidate ponints/frames on the lattice is,

$$|L| = O(1/\epsilon^3). \tag{16}$$

D. Lattice-based Algorithm

With the approximation bound established, the remaining task is to search $\tilde{C}^{p*}$ on L. We design an induction-like approach that builds on the relationship between the solution to the $(p-1)$-frame problem and the solution to the p-frame problem. The key elements that establish the connection are Conditional Optimal Solution (COS) and Conditional Optimal Residual Solution (CORS).

Definition 6 (Conditional Optimal Solution): $\forall \tilde{c} \in L$, the COS, $\tilde{U}_j(\tilde{c}) = \{\tilde{C}^{j*} | \tilde{c} \in \tilde{C}^{j*}\}$, is defined as the optimal j-frame set, $j = 1, 2, ..., p$, for the j-frame problem that must include $\tilde{c}$ in the solution set. Also, $\tilde{U}_j(\tilde{c})$ satisfies the VNOC.

Therefore, we can obtain the optimal solution, $\tilde{C}^{p*}$, on the lattice by searching $\tilde{c}$ over L and its corresponding COS,

$$\tilde{C}^{p*} = \tilde{U}_p(\tilde{c}^*), \tag{17}$$

where $\tilde{c}^* = \arg\max_{\tilde{c} \in L} s(\tilde{U}_p(\tilde{c}))$.

Definition 7 (Conditional Optimal Residual Solution): Given any COS, $\tilde{U}_{j+1}(\tilde{c}), j = 0, 1, ..., p-1$, we define the j-frame CORS with respect to $\tilde{c}$ as: $\tilde{Q}_j(\tilde{c}) = \tilde{U}_{j+1}(\tilde{c}) - \{\tilde{c}\}$.

Corollary 2: $\tilde{Q}_j(\tilde{c})$ is the optimal j-frame set that satisfies,

- $\tilde{c} \notin \tilde{Q}_j(\tilde{c})$,
- $\{\tilde{c}\} \cup \tilde{Q}_j(\tilde{c})$ satisfies the VNOC.

What is interesting is that CORS allows us to establish the relationship between $\tilde{Q}_j$ and $\tilde{Q}_{j-1}$.

Lemma 3:

$$\tilde{Q}_j(\tilde{c}_u) = \tilde{Q}_{j-1}(\tilde{c^*}) \cup \{\tilde{c^*}\}, \tag{18}$$

where $\tilde{c^*} = \arg\max_{\tilde{c} \in L} s(\tilde{Q}_{j-1}(\tilde{c}) \cup \{\tilde{c}\})$, subject to the constraint that $\{\tilde{c}_u, \tilde{c}\} \cup \tilde{Q}_{j-1}(\tilde{c})$ satisfies the VNOC.

Proof: We prove the lemma by contradiction. Notice that the right hand side of (18) returns one of the j-frame sets that satisfy the two conditions in Corollary 2, while the left hand side is defined to be the optimal j-frame set that satisfies the same two conditions. Therefore, if we assume (18) does not hold, the only possibility is,

$$s(\tilde{Q}_j(\tilde{c}_u)) > s(\tilde{Q}_{j-1}(\tilde{c^*}) \cup \{\tilde{c^*}\}). \tag{19}$$

Take an arbitrary frame $\tilde{c}_v \in \tilde{Q}_j(\tilde{c}_u)$ out of $\tilde{Q}_j(\tilde{c}_u)$, the result is $\tilde{Q}_j(\tilde{c}_u) - \{\tilde{c}_v\}$ and according to Lemma 1, we have,

$$s(\tilde{Q}_j(\tilde{c}_u) - \{\tilde{c}_v\}) = s(\tilde{Q}_j(\tilde{c}_u)) - s(\tilde{c}_v). \tag{20}$$

Take $\tilde{c}_v$ out of $\tilde{Q}_{j-1}(\tilde{c}_v) \cup \{\tilde{c}_v\}$, the result is $\tilde{Q}_{j-1}(\tilde{c}_v)$ and

$$s(\tilde{Q}_{j-1}(\tilde{c}_v)) = s(\tilde{Q}_{j-1}(\tilde{c}_v) \cup \{\tilde{c}_v\}) - s(\tilde{c}_v). \tag{21}$$

Based on (19) and the fact that

$$s(\tilde{Q}_{j-1}(\tilde{c^*}) \cup \{\tilde{c^*}\}) \geq s(\tilde{Q}_{j-1}(\tilde{c}_v) \cup \{\tilde{c}_v\}),$$

we have,

$$s(\tilde{Q}_j(\tilde{c}_u)) > s(\tilde{Q}_{j-1}(\tilde{c}_v) \cup \{\tilde{c}_v\}). \tag{22}$$

Take $\tilde{c}_v$ out of both sides and combine with (20) and (21) respectively, we have,

$$s(\tilde{Q}_j(\tilde{c}_u) - \{\tilde{c}_v\}) > s(\tilde{Q}_{j-1}(\tilde{c}_v)). \tag{23}$$

The frame set on the right hand side of (23), $\tilde{Q}_{j-1}(\tilde{c}_v)$, is defined to be the optimal $(j-1)$-frame set that satisfies the two conditions in Corollary 2 while the frame set on left hand side, $\tilde{Q}_j(\tilde{c}_u) - \{\tilde{c}_v\}$, is only one of the $(j-1)$-frame sets that satisfy the two conditions. Contradiction occurs. ■

It is worth mentioning that it takes $O(p)$ time to check if $(\{\tilde{c}_u, \tilde{c}\} \cup \tilde{Q}_j(\tilde{c}))$ satisfies the VNOC. Because $\{\tilde{c}\} \cup \tilde{Q}_j(\tilde{c}) = \tilde{U}_{j+1}(\tilde{c})$ satisfies the VNOC as defined in Definition 6 and thus we only need to check if $\{\tilde{c}_u\} \cup \tilde{U}_{j+1}(\tilde{c})$ satisfies the VNOC, which takes $O(p)$ time.

Eq. (17) implies that we can obtain the approximation solution $\tilde{C}^{p*}$ from $\tilde{U}_p$. Definition 7 indicates that we can obtain $\tilde{U}_p$ from $\tilde{Q}_{p-1}$. Now Lemma 3 implies that we can construct $\tilde{Q}_j$ from $\tilde{Q}_{j-1}$, $j = 1, 2, ..., p-1$. Considering the fact that $\tilde{Q}_0 = \phi$, this allows us to establish the algorithm using an induction-like approach. Algorithm 1 shows the complete lattice-based algorithm. Considering any candidate frame $\tilde{c} \in L$, we pre-calculate the satisfaction values for all the $|L|$ candidate frames and store the values in a lookup table to avoid redundant calculation. Given any candidate frame $\tilde{c}_u \in L$ as the input, the lookup function l returns the satisfaction value of $\tilde{c}_u$, $l(\tilde{c}_u) = s(\tilde{c}_u)$. We implement the lookup function using the array, $l[u] = s(\tilde{c}_u)$. From the pseudo code in Algorithm 1, it is not difficult to know that,

Theorem 1: Algorithm 1 runs in $O(n/\epsilon^3 + p^2/\epsilon^6)$ time.

Algorithm 1: Lattice-based Algorithm

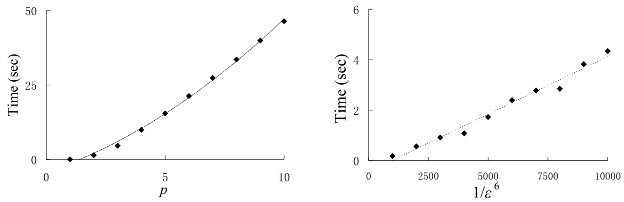
```
begin
    for j ← 1 to |L| do                                        O(1/ε^3)
        l[j] = s(c̃_j)                                          O(n)
        Q̃_0(c̃_j) = ∅;                                          O(1)
        s(Q̃_0(c̃_j)) = 0;                                       O(1)
    end
    for k ← 1 to p do                                          O(p)
        C̃^{k*} = ∅;                                            O(1)
        s(C̃^{k*}) = 0;                                         O(1)
        for u ← 1 to |L| do                  update C̃^{k*}, O(1/ε^3)
            if s(C̃^{k*}) < s(Q̃_{k-1}(c̃_u)) + l[u] then
                C̃^{k*} = Q̃_{k-1}(c̃_u) ∪ {c̃_u};                  O(1)
                s(C̃^{k*}) = s(Q̃_{k-1}(c̃_u)) + l[u];            O(1)
            end
        end
        for u ← 1 to |L| do                update Q̃_k(c̃_u), O(1/ε^3)
            Q̃_k(c̃_u) = Q̃_{k-1}(c̃_u) ∪ ∅;                       O(1)
            s(Q̃_k(c̃_u)) = s(Q̃_{k-1}(c̃_u));                   O(1)
            for v ← 1 to |L| do                                O(1/ε^3)
                if s(Q̃_k(c̃_u)) < s(Q̃_{k-1}(c̃_v)) + l[v] AND
                {c̃_u, c̃_v} ∪ Q̃_{k-1}(c̃_v)
                satisfies the VNOC                             O(p)
                then
                    Q̃_k(c̃_u) = Q̃_{k-1}(c̃_v) ∪ {c̃_v};           O(1)
                    s(Q̃_k(c̃_u)) = s(Q̃_{k-1}(c̃_v)) + l[v];     O(1)
                end
            end
        end
    end
    return C̃^{p*};
end
```

V. Experimental results

We have implemented the algorithm using Java. The computer used is a desktop computer with an Intel Core 2 Duo 2.13GHz CPU and 2GB RAM. The operating system is Windows XP. In experiments, we test the algorithm speed with different parameter settings including the number of request n, the number of camera frames p, and the approximation bound ϵ.

In the experiments, both triangular and rectangular inputs are randomly generated. First, s_d points in V are uniformly generated across the reachable field of view. These points indicate the locations of interest and are referred to as seeds. Each seed is associated with a random radius of interest. To generate a request, we randomly assign it to one seed. For a triangular request, three 2-D points are randomly generated within the radius of the corresponding seed as the vertices of the triangle. For a rectangular request, a 2-D point is randomly generated as the center of the rectangular region within the radius of corresponding seed and then two random numbers are generated as the width and height of the request. Finally, the resolution value of the request is uniformly randomly generated across the resolution range $[\underline{z}, \overline{z}]$.

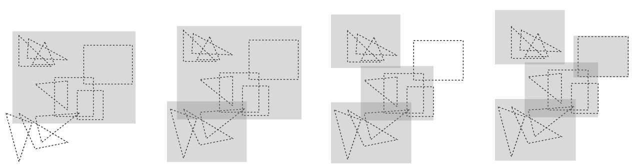
Across the experiment, we set w=80, h=60, $\underline{z}$=5, $\overline{z}$=15 and s_d=4. For each parameter setting, 50 trials have been carried out for averaged performance. The simulation results indicate



(a) The computation time vs. the number of frames p, (n=100, ϵ=0.25).

(b) The computation time vs. the approximation bound ϵ, (n=100, p=2). Note that the horizontal axis is $1/\epsilon^6$.

Fig. 2. Speed testing results.



(a) p=1, s=4.21

(b) p=2, s=6.32

(c) p=3, s=8.11

(d) p=4, s=9.07

Fig. 3. Sample outputs when p increases for a fixed input set $n = 10$.

the linear relationship between the computation time and n. Fig. 2 illustrates the relationship between the computation time and the parameters p and ϵ. The results are consistent with our analysis.

Fig. 3 shows how the output of the algorithm for a fixed set of inputs (n=10) changes when p increases from 1 to 4. It shows that our algorithm reasonably allocates camera frames in each case.

VI. Conclusion

We formulated the least overlapping p-frame problem with non-partial coverage as an optimization problem. A lattice-based approximation algorithm was proposed for solving the problem. Given n requests and p camera frames, the algorithm runs in $O(n/\epsilon^3 + p^2/\epsilon^6)$ time with the approximation bound ϵ. We have implemented the algorithm and tested it on random inputs. The experimental results are consistent with our theoretical analysis.

In future work, we will explore the new geometric data structures to improve complexity results. We will also develop algorithms for different variations of the problem such as allowing camera frames to overlap with each others. We plan to apply the algorithm in an outdoor collaborative observation system in the natural environment for field experiments. We plan to make it available online for Internet users with various background and observation purposes.

Acknowledgement

We would like to thank R. Volz and K. Goldberg for their insightful input, George Lee and Junku Yuh for their consistent support, and N. Qin, H. Lee, B. Green, C. Kim, H. Wang, P. Ritchey, and J. Kelm for their contribution to the Networked Robot Lab at Texas A&M University.

References

[1] D. Song and K. Goldberg, "Approximate algorithms for a collaboratively controlled robotic camera," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1061–1070, Oct. 2007.

[2] 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 (ICRA), San Francisco, CA*, April 2000, pp. 3138–3343.

[3] X.-G. Wang, M. Moallem, and R. Patel, "An internet-based distributed multiple-telerobot system," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 33, no. 5, pp. 627–634, Sept. 2003.

[4] S. Sirouspour and P. Setoodeh, "Multi-operator/multi-robot teleoperation: an adaptive nonlinear control approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems,(IROS)*, vol. 1, August 2005, pp. 1576–1581.

[5] C.-H. Fua and S. S. Ge, "Cobos: Cooperative backoff adaptive scheme for multirobot task allocation," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1168–1178, Dec. 2005.

[6] J. Liu, L. Sun, T. Chen, X. Huang, and C. Zhao, "Competitive multi-robot teleoperation," in *IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain*, April 2005, pp. 75–80.

[7] C. Zhao, J. Liu, Y. Li, T. Chen, and L. Sun, "Virtual repulsive force in competitive multi-robot teleoperation," in *Third International Conference on Information Technology and Applications (ICITA), Sydney, Australia*, vol. 2, July 2005, pp. 15–20.

[8] N. Megiddo and K. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13, pp. 182–196, February 1984.

[9] D. Eppstein, "Fast construciton of planar two-centers," in *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, January 1997, pp. 131–138.

[10] E. M. Arkin, G. Barequet, and J. S. B. Mitchell, "Algorithms for two-box covering," in *Symposium on Computational Geometry*, June 2006, pp. 459–467.

[11] H. Alt, E. M. Arkin, H. Brönnimann, J. Erickson, S. P. Fekete, C. Knauer, J. Lenchner, J. S. B. Mitchell, and K. Whittlesey, "Minimum-cost coverage of point sets by disks," in *Symposium on Computational Geometry*, June 2006, pp. 449–458.

[12] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[13] P. K. Agarwal, S. Bereg, O. Daescu, H. Kaplan, S. C. Ntafos, and B. Zhu, "Guarding a terrain by two watchtowers," in *Symposium on Computational Geometry*, June 2005, pp. 346–355.

[14] D. Eppstein, M. T. Goodrich, and N. Sitchinava, "Guard placement for efficient point-in-polygon proofs," in *Symposium on Computational Geometry*, 2007, pp. 27–36.

[15] T. Shermer, "Recent results in art galleries," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1384–1399, Sep. 1992.

[16] D. Song, A. Pashkevich, and K. Goldberg, "Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam," in *IEEE/RSJ International Conference on Intelligent Robots (IROS), Las Vegas, NV*, vol. 2, Oct. 2003, pp. 1087 – 1093.

[17] S. Har-Peled, V. Koltun, D. Song, and K. Goldberg, "Efficient algorithms for shared camera control," in *19th ACM Symposium on Computational Geometry, San Diego, CA*, June 2003.

[18] D. Song, A. van der Stappen, and K. Goldberg, "An exact algorithm optimizing coverage-resolution for automated satellite frame selection," in *IEEE International International Conference on Robotics and Automation (ICRA 2004), New Orleans, US*, May 2004.

[19] D. Song, A. F. van der Stappen, and K. Goldberg, "Exact algorithms for single frame selection on multi-axis satellites," *IEEE Transactions on Automation Science and Engineering*, vol. 3, no. 1, pp. 16–28, January 2006.