Maximizing heterogeneous coverage in over and under provisioned visual sensor networks

Introduction

- A visual sensor network (VSN) consists of a large number of visual sensors
- Visual sensors can be either omni-directional or directional

Introduction: Deployment of visual sensor networks

A visual sensor network can be deployed in two ways:

(i) **Deterministic placement**
The visual sensors can be suitably positioned to meet the coverage requirements. However, this is only possible in a small or medium-scale network.

(ii) **Random scattering**
Most convenient and (perhaps) the only option for large scale network

Introduction: Necessity of fault tolerance in visual sensor network

- In real environment, a target may lose its coverage due to various reasons.
- Besides coverage we also need to deliberately introduce fault tolerance by covering each target with more than one sensor.

Introduction: Practicality of under-provisioned network

- We face the problem of scarcity of sensors in real environment.
- Focus on under-provisioned networks where there are insufficient number of sensors to ensure the heterogeneous coverage requirements.
    - Assign coverage priorities: targets with higher coverage requirement should get higher coverage.
    - Maintain a balanced coverage targets with the same coverage requirements would get more or less similar coverage

Introduction: Contribution

Major contributions of the paper:

(i) Study the heterogeneous coverage as a variant of Maximum Coverage with Minimum Sensors (MCMS). Propose ILP and IQP formulations for heterogeneous coverage, especially for under-provisioned networks.

(ii) Derive an upper bound on the penalty coefficient of ILP formulation.

(iii) Provide prioritized Integer Quadratic Programming (pIQP) and reduced-variance IQP formulations for under-provisioned.

(iv) Devise a greedy algorithm SOGA that gives near optimal solutions for the above formulations.

(v) Compare methodologies and algorithm with two state-of-the-art algorithms available for target coverage.

Preliminaries: Network model

- A visual sensor network (VSN) consists of smart sensors, these cameras are directional in their field of view (FoV). FoV is the span of observable area at any given direction by the camera.
- Assume a camera can move only in the horizontal direction. Present the sensing area or pan of a camera as a sector in a circle lying on a 2-dimensional plane.

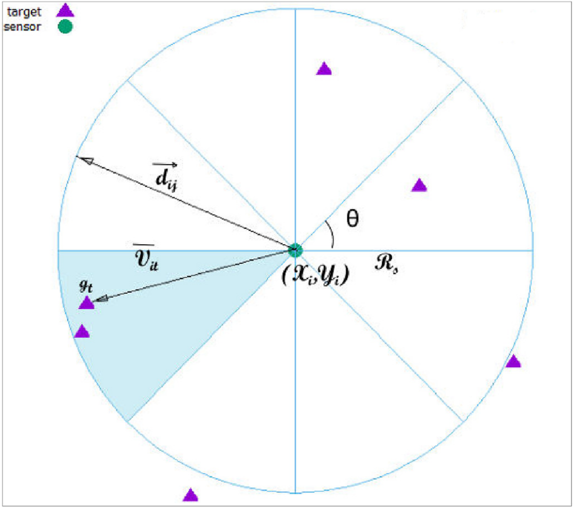
Preliminaries: Network model

Parameter:

- (x_i, y_i): the location of camera s_i.
- θ: Angle of View (AoV) / maximum coverage angle of a camera in any direction.
- R_s: maximum coverage range of a camera.
- $\overrightarrow{d_{ij}}$: a unit vector which passes through the middle of a pan, representint the orientation of camera s_i towards pan p_j.
- $\overrightarrow{v_{it}}$: a vector in the direction from camera s_i to target g_t.

Assume that cameras are homogeneous. Also, each camera has finite orientations or pans and the pans are disjoints.

Preliminaries: Network model



Preliminaries: Target in which pan (TIWP) test

- Check whether the target vector falls within the FoV of the camera s_i by checking the constraint:

$$\overrightarrow{v_{it}} \cdot \overrightarrow{d_{ij}} \geq \left|\overrightarrow{v_{it}}\right| \cos\left(\frac{\theta}{2}\right)$$

- Verify whether the target gt is within the sensing range of the camera by checking, $\left|\overrightarrow{v_{it}}\right| \leq R_s$.

If a target passes these steps $\Rightarrow$ the camera s_i covers the target g_t in pan p_j
$\mathcal{T}_{ij}$: subset of targets coverable by sensor s_i in pan p_j.

Preliminaries: Problem formulation

Given:

- a set of targets, $\mathcal{G} = \{g_1, g_2, g_3, \ldots, g_m\}$.
- a tuple of positive integer, $\mathcal{K} = (k_1, k_2, k_3, \ldots, k_m)$ where k_t is the required coverage of target g_t; targets g_i and g_j are in the same coverage group if $k_i = k_j$.
- a set of homogeneous cameras, $\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_n\}$, each of which can be oriented in one of q possible disjoint pans.
- a set of disjoint (or non-overlapping) pans, $\mathcal{P} = \{p_1, p_2, p_3, \ldots, p_q\}$
- a set of all possible (sensor, orientation) pairs is defined as, $\mathcal{F} = \{(s_i, p_j) | \textit{sensor } s_i \textit{ is activated on pan } p_j\}$.

Problem: Find a subset $\mathcal{C}$ of $\mathcal{F}$ such that, there is at most one pair for a sensor in $\mathcal{C}$ and achieved coverage of each target g_t gets maximized under the condition that target g_t is covered by at least k_t sensors and the total number of active sensors gets minimized.

ILP approach for HCT problem: ILP formulation

Parameters for ILP formulation:

- n: number of sensors
- m: number of targets
- q: number of non-overlapping pans of a sensor
- α_t: *actual coverage* of target g_t by the sensors
- ψ_t: *achieved coverage* of target g_t by the sensors; the maximum value of this variable is bounded by k_t
- $\chi_{(i,j)}$: a binary variable with value one when the sensor s_i is in orientation p_j and zero otherwise
- *Iversion bracket*: let P be a proposition, either true or false; then *Iversion bracket* is defined as: $[P] = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{otherwise} \end{cases}$

α_t can be expressed as: $\alpha_t = \sum_{i=1}^{n} \sum_{j=1}^{q} [g_t \in \mathcal{T}_{ij}] \, \chi_{(i,j)}$

ILP approach for HCT problem: ILP formulation

The ILP formulation for HCT problem:

$$\textbf{maximize:} \quad \sum_{t=1}^{m} \psi_t - \rho \sum_{i=1}^{n} \sum_{j=1}^{q} \chi_{(i,j)}$$

subject to constraints:

- $\dfrac{\alpha_t}{n} \leq \psi_t \leq \alpha_t, \quad \forall t = 1, 2, \ldots, m$
- $\psi_t = 0$ or 1 or 2 or $\ldots$ or $k_t, \quad \forall t = 1, 2, \ldots, m$
- $\sum_{j=1}^{q} \chi_{(i,j)} \leq 1, \quad \forall i = 1, 2, \ldots, n$
- $\chi_{(i,j)} = 0$ or $1, \quad \forall i = 1, 2, \ldots, n, \forall j = 1, 2, \ldots, q$

ILP approach for HCT problem: Impact of penalty coefficient

The penalty coefficient ρ is a positive real number, used to impose a penalty on the number of activated sensors. If there are two solutions, one of them has higher total coverage than the other, this penalty coefficient ensures to choose the former one, irrespective of the number of activated sensors.

Lemma

Let n be the number of sensors. Then for ILP formulation, the penalty coefficient, ρ should be smaller than $\frac{1}{n}$ to pick a solution with higher coverage count, irrespective of activated sensor count.

If $\rho > 1$, then more priority will be given to reduce the number of activated sensors than maximizing the total coverage.

IQP approach

- The motivation behind using quadratic programming method to solve HCT problem is to minimize the distance between required and achieved coverage vectors by activating minimum number of sensors.
- Minimize the vector distance $d = \| \overrightarrow{k} - \overrightarrow{\psi} \|_2$ and also the total number of active sensors.
- New objective function:

$$\textbf{minimize:} \quad \sum_{t=1}^{m} (k_t - \psi_t)^2 + \rho \sum_{i=1}^{n} \sum_{j=1}^{q} \chi_{(i,j)}$$

All the other constraints are same as ILP formulation.

Special constraints for under provisioned network: Prioritized coverage

- When the problem is not solvable we may assign priorities–targets with higher coverage requirements will be assigned with higher priorities.
- Modify the objective function of the basic IQP to capture the priority criteria as follows:

$$\textbf{minimize:} \quad \sum_{t=1}^{m} k_t (k_t - \psi_t)^2 + \rho \sum_{i=1}^{n} \sum_{j=1}^{q} \chi_{(i,j)}$$

All the other constraints remain same as for the ILP.

Special constraints for under provisioned network: Groupwise balanced coverage

- Another approach for under-provisioned case, is to minimize coverage variance within each individual coverage group. It is desirable to have the same coverage for the targets which have the same requirements.
- To incorporate the balance we modify the objective function of the basic IQP as:

$$\textbf{minimize:} \quad \sum_{t=1}^{m}(k_t-\psi_t)^2+\sum_{t=1}^{m}\frac{(\psi_t-\mu_{k_t})^2}{m_{k_t}}+\rho\sum_{i=1}^{n}\sum_{j=1}^{q}\chi_{(i,j)}$$

Special constraints for under provisioned network: Groupwise balanced coverage

- m_{k_t} is the number of target that require k_t-coverage.
- μ_{k_t} represents the mean coverage of k_t-coverage group. More precisely, μ_{k_t} is defined as:

$$\mu_{k_t} = \frac{\sum_{i=1}^{m} \psi_i \left[k_i = k_t\right]}{m_{k_t}}$$

  All the other constraints remain same as for the ILP.
- This IQP reduces the total variance of different coverage groups as well as minimizes the vector distance.

Algorithms

- The above-mentioned approaches are time-consuming and are non-scalable for large networks.
- The authors present a sensor oriented greedy algorithm, a greedy polynomial time heuristic to solve HCT problem.
- This is an iterative algorithm. The basic idea of this algorithm is to activate (at each iteration) a sensor in a pan that maximizes the benefit value.
- The benefit of a pan for a sensor depends on the nature of the algorithm.

Algorithms

Algorithm 1 SOGA (Sensor Oriented Greedy Algorithm) to solve HCT problem.

$\mathcal{F}$ {a set of all possible (sensor, orientation) pairs}
$\mathcal{C} \leftarrow \emptyset$ { a subset of $\mathcal{F}$, final output of this algorithm}
$\mathcal{N} \leftarrow \mathcal{G}$ {a set of unachieved targets}
$a_t \leftarrow 0$ {achieved coverage for each target g_t upto previous iteration}
While $\mathcal{F} \neq \emptyset$ **and** $\mathcal{N} \neq \emptyset$ **do**
    $(s_{i^*}, p_{j^*}) \leftarrow \arg \max_{(s_i, p_j) \in \mathcal{F}} \textit{benefit}(s_i, p_j)$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(s_{i^*}, p_{j^*})\}$
    for all $g_t \in \mathcal{T}_{i^*j^*}$ **and** $g_t \in \mathcal{N}$ **do**
        $a_t \leftarrow a_t + 1$
        if $a_t = k_t$ **then**
            $\mathcal{N} \leftarrow \mathcal{N} - \{g_t\}$
    remove all pairs for sensor s_{i^*} from $\mathcal{F}$
return $\mathcal{C}$

SOGA

Frame Title

Algorithm 2 *benefit*(s_i, p_j).

1: *value* $\leftarrow 0$

2: **for all** $g_t \in \mathcal{T}_{ij}$ and $a_t < k_t$ **do**

3: **if** *linear* **then**

4: *value* $\leftarrow$ *value* $+ 1$ (i.e. modified from Fusco and Gupta (Fusco and Gupta, 2009))

5: **if** *quadratic* **then**

6: *value* $\leftarrow$ *value* $+ [(k_t - a_t)^2 - (k_t - a_t - 1)^2]$

7: **return** *value*

Benefit function for IQP

Algorithms: Time complexity

The complexity of the Algorithm 1 in worst case is as follows:

- The outer while loop iterates $O(n)$ times; for each iteration the line 6 takes $O(nq)$ checking.
- For each checking the benefit function costs $O(m)$.
- The inner loops cost $O(m)$.
- So each iteration costs $O(nqm + m)$.

$\Longrightarrow$ the time complexity is $O(n^2qm)$.

Algorithms: Prioritize coverage

Algorithm 3 *benefit*(s_i, p_j) (for prioritized-IQP).

1: $value \leftarrow 0$

2: **for all** $g_t \in \mathcal{T}_{ij}$ and $a_t < k_t$ **do**

3: $\quad value \leftarrow value + k_t[(k_t - a_t)^2 - (k_t - a_t - 1)^2]$

4: **return** $value$

Benefit function for pIQP

Algorithms: Groupwise balanced coverage

Algorithm 4 *benefit*(s_i, p_j) (for reduced-variance IQP).

1: $value \leftarrow 0$

2: **for all** $g_t \in \mathcal{T}_{ij}$ and $a_t < k_t$ **do**

3: $\quad \mu \leftarrow \mu_{k_t}$ {average of k_t-coverage group}

4: $\quad g \leftarrow m_{k_t}$ {number of targets in k_t-coverage group}

5: $\quad old \leftarrow (k_t - a_t)^2 + \frac{(a_t - \mu)^2}{g}$

6: $\quad new \leftarrow (k_t - a_t - 1)^2 + \frac{(a_t - \mu + 1 - \frac{1}{g})^2}{g}$

7: $\quad value \leftarrow value + (old - new)$

8: **return** *value*

Benefit function for rvIQP

Simulations

- Using a 2D grid as ployment area:
  - Size 200×200 sq. units for "Small" scale network.
  - Size 1000×1000 sq. units for "Large" scale network.
- Targets and sensors are considered as points and are randomly deployed.
- For the deployment of sensors, consider two types of distributions:
  1. *Uniform:* sensors are distributed uniformly all over the deployment are.
  2. *Zipf:* 80% sensors are distributed within 20% of the are and remaining 20% sensors are distributed within 80% of the deployment area.

Simulations: Performance metrics

- **Distance Index** $(\mathcal{DI})$
  $\mathcal{DI} = \frac{\sum_{t=1}^{m} k_t^2 - \sum_{t=1}^{m}(k_t - \psi_t)^2}{\sum_{t=1}^{m} k_t^2}$
  Higher value of this metric indicates better coverage achieved with respect to requirements.
- **Activated Sensors** If an algorithm provides same coverage as another algorithm, but activates less number of sensors, we say that first algorithm is better than the second.

Simulations: Performance metrics

- **Variance** To evaluate the group-wise balance approach, variance is measured which is the sum of group-wise variances. Measure the following terms:

$$\sum_{t=1}^{m} \frac{(\psi_t - \mu_{k_t})^2}{m_{k_t}}$$

  A lower value on this metric indicates better coverage balancing.

- **Power(W)** The total consumed power (P_c) is calculated as:
  $P_c = n_a \times P_a + n_i \times P_i + n_s \times P_s$
  A lower value of this metric indicates better energy efficiency.

Simulations: Performance metrics

- **Coverage Quality**

$$U(i,j,t) = \begin{cases} 1 - \left(\frac{\left| \overrightarrow{v_{it}} \right|}{R_s} \right)^2, & \text{if } g_t \in \mathcal{T}_{ij} \text{ and } \overrightarrow{v_{it}} < R_s \\ 0, & \text{otherwise.} \end{cases}$$

- The total coverage quality is the total achieved qualities of all the targets for a given solution:

$$\mathcal{CQ} = \sum_{i,j,t} U(i,j,t) \left[s_i \textit{ is activated in } p_j \right]$$

The higher value of this index denotes better quality of coverage.

Simulations

Summary of experimental results.

Approach	Result
IQP	Outperforms all the other approaches in terms of distance index
Greedy quadratic	Approximates the IQP but requires less time-complexity
ILP	Provides maximum total coverage
Modified greedy algorithm (Fusco and Gupta)	Approximates the ILP
Greedy-Razali	Uses lower number of sensors and consumes lower power than IQP. But it performs worse than IQP in terms of distance index.
pIQP	Provides higher distance index (i.e., higher coverage) in higher coverage group
Priority-based Greedy quadratic	Approximates the pIQP but requires less time-complexity
Reduced-variance IQP	Provides balanced coverage within each coverage group
Reduced-variance Greedy quadratic	Approximates the reduced-variance IQP but requires less time-complexity

Conclusion

- IQP outperforms all the approaches in terms of fulfilling coverage requirements although its sensor usage and power consumption are higher than other approaches
- In large networks, IQP is not suitable. Greedy IQP has shows almost similar performance behavior of IQP.
- Prioritized IQP is good at covering targets with higher coverage requirements and reduced variance IQP is suitable for coverage balancing within each coverage requirement group