Optimizing Load Distribution in Camera Networks with a Hypergraph Model of Coverage Topology

Aaron Mavrinac and Xiang Chen
Department of Electrical and Computer Engineering, University of Windsor
Email: {mavrin1,xchen}@uwindsor.ca

***Abstract*—A new topological model of camera network coverage, based on a weighted hypergraph representation, is introduced. The model's theoretical basis is the coverage strength model, presented in previous work and summarized here. Optimal distribution of task processing is approximated by adapting a local search heuristic for parallel machine scheduling to this hypergraph model. Simulation results are presented to demonstrate its effectiveness.**

I. Introduction

Multi-camera systems have been studied extensively for a wide variety of applications. Although centralized architectures for fusing and processing data from these multiple sources are a natural extension of traditional computer vision methods, such configurations are limited in scalability and robustness. The increasingly popular distributed smart camera network [1] paradigm is the answer to this challenge. In such a system, each camera node possesses local processing capabilities, and data is increasingly abstracted (and thus increasingly compact) as it is communicated and processed farther from its original source. Zivkovic and Kleihorst [2] give an overview and analysis of smart camera node architecture illuminating the benefits of this design.

Naturally, any initial image or video processing tasks which require data only from a single node are assigned to that node. However, if the nodes themselves are also responsible for fusing and processing data from multiple sources – as they must be, in a true distributed smart camera network – it is less obvious where to assign such tasks.

Scheduling has been an active area of research for decades, and algorithms solving a variety of different problems have been used in such diverse applications as manufacturing and distributed computing [3]. Formulating an appropriate scheduling problem requires domain-specific knowledge; in our case, an understanding of the underlying nature of a multi-camera task.

The scale and performance of most tasks in multi-camera networks (indeed, in sensor networks generally) are directly related to the volume of coverage of the sensor(s) in question. In previous work [4], we developed a real-valued coverage model for multi-camera systems, inspired by task-oriented sensor planning models from the computer vision literature [5] and by coverage models used for various purposes in wireless sensor networks [6], [7]. We demonstrate that, given a set of a priori parameters of the multi-camera system and some task requirements, this model accurately describes the true coverage of a scene in the context of the task. In order that this work be self-contained, we provide in Section II a reduced but functionally complete description of the model. This provides us with a basis for a priori quantitative characterization of multi-camera tasks.

The next step is to abstract this understanding into a topological structure suitable for optimization over the network. Our first contribution is a novel topological model for camera network coverage using a hypergraph representation, described in Section III. Devarajan and Radke [8] propose the vision graph as a theoretical topological model for pairwise tasks in camera networks; it has since been constructed and employed in several such applications [9], [10]. Lobaton et al. [11], [12] recognize the inadequacy of a graph for accurately capturing topology, and generalize to a simplicial complex representation. In the context of camera networks which may be processing a coverage-bound task with data from arbitrary combinations of sensors, we contend that only the hypergraph representation is sufficiently general. Additionally, we define a hyperedge weighting function which incorporates the salient coverage information for optimization.

Our second contribution, detailed in Section IV, is the characterization of the optimal task processing distribution problem in the hypergraph framework, and the adaptation of a local search heuristic from the scheduling literature [13] which has been shown to exhibit good performance for this class of problem.

We present simulated experimental results demonstrating the method on a virtual network of 23 cameras in Section V. Finally, we give some concluding remarks in Section VI.

II. Coverage Model

A. Stimulus Space

The sensor coverage model requires the definition of a *stimulus space* to describe individual observable data. A visual stimulus is localized to a point in three-dimensional space, and also has a direction (normal to the surface on which the point lies, i.e. view angle). We therefore define a *directional space* as the stimulus space.

Definition 1: The directional space $\mathbb{D}^3 = \mathbb{R}^3 \times [0, \pi] \times [0, 2\pi)$ consists of three-dimensional Euclidean space plus direction, with elements of the form (x, y, z, ρ, η).

We term $\mathbf{p} \in \mathbb{D}^3$ a *directional point*. For convenience, we denote its spatial component $\mathbf{p}_s = (\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z)$ and its directional component $\mathbf{p}_d = (\mathbf{p}_\rho, \mathbf{p}_\eta)$.

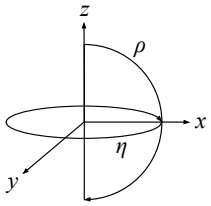

Fig. 1. Axes and Angles of $\mathbb{D}^3$

A standard 3D pose $P : \mathbb{R}^3 \to \mathbb{R}^3$, consisting of rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$, may be applied to $\mathbf{p} \in \mathbb{D}^3$. The spatial component is transformed as usual, i.e. $P(\mathbf{p}_s) = \mathbf{R}\mathbf{p}_s + \mathbf{T}$. The direction component is transformed as follows. If $\mathbf{d}$ is the unit vector in the direction of $\mathbf{p}_d$, then $P(\mathbf{p}_d) = (\arccos(\mathbf{R}\mathbf{d}_z), \arctan 2(\mathbf{R}\mathbf{d}_y, \mathbf{R}\mathbf{d}_x))$.

B. Coverage Strength Model

The coverage strength model of a given sensor system (which may be a single physical sensor or multiple sensors) assigns to every point in the stimulus space a measure of coverage strength.

Definition 2: A coverage strength model is a mapping $C : \mathbb{D}^3 \to [0, 1]$, for which $C(\mathbf{p})$, for any $\mathbf{p} \in \mathbb{D}^3$, is the strength of coverage at $\mathbf{p}$.

Definition 3: The set $\langle C \rangle = \{\mathbf{p} \in \mathbb{D}^3 | C(\mathbf{p}) > 0\}$ is the coverage hull of a coverage strength model C.

In order for the coverage strength model to offer a useful gauge of sensor system performance, it requires the context of a task.

Definition 4: A relevance model is a mapping $R : \mathbb{D}^3 \to [0, 1]$, for which $R(\mathbf{p})$, for any $\mathbf{p} \in \mathbb{D}^3$, is the minimum desired coverage strength or coverage priority at $\mathbf{p}$.

The coverage strength model is defined in part by task requirements, defined by a set of *task parameters* which encapsualate various properties of the a posteriori quality of sensed data. These parameters and a relevance model together fully describe a task.

Given coverage strength and/or relevance models C_i and C_j, we define their union and intersection, respectively, as

$$C_i \cup C_j(\mathbf{p}) = \max(C_i(\mathbf{p}), C_j(\mathbf{p})) \quad (1)$$

$$C_i \cap C_j(\mathbf{p}) = \min(C_i(\mathbf{p}), C_j(\mathbf{p})) \quad (2)$$

for all $\mathbf{p} \in \mathbb{D}^3$. This, together with Definition 3, implies that $\langle C_i \cup C_j \rangle = \langle C_i \rangle \cup \langle C_j \rangle$ and $\langle C_i \cap C_j \rangle = \langle C_i \rangle \cap \langle C_j \rangle$.

The k-coverage strength model for a subset of sensor systems $M \subset N$, where $|M| = k$, is

$$C_M = \bigcap_{m \in M} C_m \quad (3)$$

The k-coverage strength model for the network is

$$C_N^k = \bigcup_{M \in \binom{N}{k}} C_M \quad (4)$$

where each M is a k-combination of N. Note that in the common case where $k = 1$, (3) and (4) reduce to $C_N^1 = \bigcup_{m \in N} C_m$.

C. Single-Camera Model

First, we present a single-camera parameterization of the coverage strength model, for which the full theoretical derivation can be found in [4].

Given a task parameter γ indicating a margin in the image (in pixels) for full coverage, the horizontal and vertical cross-sections of the visibility component, C_V, are given by

$$C_{Vh}(\mathbf{p}) = B_{[0,1]}\left(\frac{\min\left(\frac{\mathbf{p}_x}{\mathbf{p}_z} + \sin(\alpha_{hl}), \sin(\alpha_{hr}) - \frac{\mathbf{p}_x}{\mathbf{p}_z}\right)}{\gamma_h}\right) \quad (5)$$

$$C_{Vv}(\mathbf{p}) = B_{[0,1]}\left(\frac{\min\left(\frac{\mathbf{p}_y}{\mathbf{p}_z} + \sin(\alpha_{vt}), \sin(\alpha_{vb}) - \frac{\mathbf{p}_y}{\mathbf{p}_z}\right)}{\gamma_v}\right) \quad (6)$$

for $\gamma > 0$, where α_{hl} and α_{hr} are the horizontal field-of-view angles, and α_{vt} and α_{vb} are the vertical field-of-view angles. The complete C_V is then given by

$$C_V(\mathbf{p}) = \begin{cases} \min(C_{Vh}(\mathbf{p}), C_{Vv}(\mathbf{p})) & \text{if } \mathbf{p}_z > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The resolution component, C_R, is given by

$$C_R(\mathbf{p}) = B_{[0,1]}\left(\frac{z_2 - \mathbf{p}_z}{z_2 - z_1}\right) \quad (8)$$

for $R_1 > R_2$, where the values of z_1 and z_2 are given by (9), substituting task parameters R_1 (ideal resolution) and R_2 (minimum resolution), respectively, for R.

$$z_R = \frac{1}{R} \min\left[\frac{w}{2\sin(\alpha_h/2)}, \frac{h}{2\sin(\alpha_v/2)}\right] \quad (9)$$

In the preceding equation, $\alpha_h = \alpha_{hl} + \alpha_{hr}$ and $\alpha_v = \alpha_{vt} + \alpha_{vb}$.

Given a task parameter $c_{\max}$ indicating the maximum acceptable blur circle diameter, the focus component, C_F, is given by

$$C_F(\mathbf{p}) = B_{[0,1]}\left(\min\left(\frac{\mathbf{p}_z - z_n}{z_\triangleleft - z_n}, \frac{z_f - \mathbf{p}_z}{z_f - z_\triangleright}\right)\right) \quad (10)$$

for $c_{\max} > c_{\min}$, where $(z_\triangleleft, z_\triangleright)$ and (z_n, z_f) are the near and far limits of depth of field as given by (11), substituting blur circle diameters $c_{\min}$ and $c_{\max}$, respectively, for c.

$$z = \frac{Afz_S}{Af \pm c(z_S - f)} \quad (11)$$

In the preceding equation, A is the effective aperture diameter, f is the focal length, and z_S is the subject distance. Generally, $c_{\min}$ is equal to the physical pixel size, yielding the depth of field for effectively perfect focus.

The direction (angle of view) component, C_D, is given by

$$C_D(\mathbf{p}) = B_{[0,1]}\left(\frac{\Theta(\mathbf{p}) - \pi + \zeta_2}{\zeta_2 - \zeta_1}\right) \quad (12)$$

where $\zeta_1, \zeta_2 \in [0, \pi/2]$ are task parameters indicating the ideal and maximum view angles, respectively, and $\Theta(\mathbf{p})$ is defined as

$$\Theta(\mathbf{p}) \equiv \mathbf{p}_\rho - \left(\frac{\mathbf{p}_y}{r}\sin\mathbf{p}_\eta + \frac{\mathbf{p}_x}{r}\cos\mathbf{p}_\eta\right)\arctan\left(\frac{r}{\mathbf{p}_z}\right) \quad (13)$$

where $r = \sqrt{\mathbf{p}_x^2 + \mathbf{p}_y^2}$.

The full coverage strength model is simply the product of these components:

$$C(\mathbf{p}) = C_V(\mathbf{p})C_R(\mathbf{p})C_F(\mathbf{p})C_D(\mathbf{p}) \quad (14)$$

D. Multi-Camera System Model

A set of single-camera models may be placed in the context of a world coordinate frame and a scene, and then combined into multi-camera coverage models. Again, theoretical details may be found in [4].

The six degrees of freedom of a camera's world frame pose $P : \mathbb{R}^3 \to \mathbb{R}^3$ are called the *extrinsic parameters* of the camera [14]. As discussed in Section II-A, P can be extended to $P_D : \mathbb{D}^3 \to \mathbb{D}^3$. The in-scene model for a single camera, then, is the single-camera model C with its domain transformed to the world frame, defined by

$$C^s(\mathbf{p}) = C(P_D^{-1}(\mathbf{p})) \quad (15)$$

for any world frame point $\mathbf{p} \in \mathbb{D}^3$.

Given a *scene model* S consisting of a set of plane segments (which represent opaque surfaces in the scene), the point $\mathbf{p}_s$ is occluded iff the point of intersection between the line from $\mathbf{p}_s$ to the camera's principal point and any plane segment in S exists, is unique, and is not $\mathbf{p}_s$.

If $V : \mathbb{R}^3 \to \{0, 1\}$ is a bivalent indicator function such that $V(\mathbf{p}_s) = 1$ iff $\mathbf{p}_s$ is not occluded from a given camera's viewpoint, then the in-scene model with static occlusion is defined by

$$C^o(\mathbf{p}) = C^s(\mathbf{p})V(\mathbf{p}_s) \quad (16)$$

for any $\mathbf{p} \in \mathbb{D}^3$, where $\mathbf{p}_s$ is the spatial component of $\mathbf{p}$, and where C^s is given by (15).

Finally, the k-ocular multi-camera system model is computed via (3) and (4).

E. Discrete Model

While it is feasible to compute the vertices of the coverage hull $\langle C^o \rangle$ of an in-scene camera coverage strength model with occlusion directly from the parameterizations in Sections II-C and II-D, the only obvious way to obtain $\langle C^o_M \rangle$, where $|M| > 1$, is to find $\bigcap_{m \in M} \langle C^o_m \rangle$. This involves finding the intersection of arbitrary, generally non-convex polytopes given by vertices, which has been shown to be NP-hard by Tiwary [15].

An arbitrarily close approximation can be achieved in the discrete domain.[1] A coverage strength model C has a discrete counterpart denoted $\dot{C}$ such that $\dot{C}(\mathbf{p}) = C(\mathbf{p})$ for all $\mathbf{p} \in \dot{\mathbb{D}}^3$, where $\dot{\mathbb{D}}^3$ is a discrete subset of $\mathbb{D}^3$ (once this subset has been defined, it should be used consistently). We denote the summation $\sum_{\mathbf{p} \in \dot{\mathbb{D}}^3} \dot{C}(\mathbf{p})$ as $|\dot{C}|$. Then, given $\dot{C}_i$ and $\dot{C}_j$ sampled over the same discrete subset of $\mathbb{D}^3$, $\dot{C}_i \cap \dot{C}_j$ can be computed exhaustively.

[1] Incidentally, this also greatly simplifies the computation of occlusion in C^o as per (16).

III. Coverage Topology

A. Mathematical Background

A *hypergraph* $\mathcal{H}$ is a pair $\mathcal{H} = (V, E)$, where V is a set of vertices, and E is a set of non-empty subsets of V called *hyperedges*. If $\mathcal{P}(V)$ is the power set of V, then $E \subseteq \mathcal{P}(V) \backslash \emptyset$.

A *weighted hypergraph* $\mathcal{H} = (V, E, w)$ also includes a weight function over its hyperedges $w : E \to \mathbb{R}^+$. An unweighted hypergraph may be interpreted as a weighted hypergraph for which $w(e) = 1$ for all $e \in E$.

The *degree* of a vertex in $\mathcal{H}$, denoted $\delta_{\mathcal{H}}(v)$ for some $v \in V$, is the total weight of hyperedges incident to the vertex.

$$\delta_{\mathcal{H}}(v) = \sum_{e \in E} \begin{cases} w(e) & \text{if } v \in e \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

Following the definition of Frank et al. [16], a *directed hypergraph* is a pair $\mathcal{D} = (V, \vec{\mathcal{E}})$, where $\vec{\mathcal{E}}$ is a set of *hyperarcs*; a hyperarc is a hyperedge $e \subseteq V$ with a designated head vertex $v \in V$, denoted e^v. The remaining vertices $e \backslash v$ are called *tail vertices*. Two additional notions of vertex degree are defined: the indegree, $\delta^i_{\mathcal{H}}(v)$, is the total weight of hyperarcs of which v is the head vertex, and the outdegree, $\delta^o_{\mathcal{H}}(v)$, is the total weight of hyperarcs of which v is a tail vertex.

An *orientation* Λ of an undirected hypergraph $\mathcal{H}$ has the same vertex and hyperedge sets (and the same weight function, if applicable), but assigns a direction (head vertex) to each hyperedge. In an orientation of a simple hypergraph, if $e^v \in \vec{\mathcal{E}}$, then $e^u \in \vec{\mathcal{E}}$ implies $u = v$ (that is, e is unique). Therefore, we omit the head vertex superscript in certain circumstances; for example, the weight of e^v is denoted simply $w(e)$.

B. Coverage Hypergraph

The *coverage hypergraph* of a camera network N is the hypergraph $\mathcal{H}_C = (N, E_C, w_C)$. Its hyperedge set is defined as

$$E_C = \{M \in \mathcal{P}(N) | \langle C_M \cap R \rangle \neq \emptyset\} \quad (18)$$

where C_M is computed by (3) for a given task, R is a relevance model for the task, and $\mathcal{P}(N)$ denotes the power set of N. Intuitively, $M \in E_C$ indicates that nodes M have mutual coverage of some region of $\mathbb{D}^3$ with respect to R.

Theorem 1: E_C is an abstract simplicial complex; that is, for every $M \in E_C$, and every $L \subseteq M$, $L \in E_C$.

Proof: If $n \in M$, then by (3), $C_M = C_{M \backslash n} \cap n$. From (2), for all $\mathbf{p} \in \mathbb{D}^3$, $C_M(\mathbf{p}) \leq C_{M \backslash n}(\mathbf{p})$. Then, from Definition 3, clearly $\langle C_M \rangle \subseteq \langle C_{M \backslash n} \rangle$, and $\langle C_M \cap R \rangle \subseteq \langle C_{M \backslash n} \cap R \rangle$. Thus, for every $M \in E_C$, and every $M \backslash n \subset M$, $M \backslash n \in E_C$. ∎

The hyperedge weight function of $\mathcal{H}_C$, $w_C : E_C \to \mathbb{R}^+$, is defined as

$$w_C(M) = |\dot{C}_M \cap \dot{R}| \quad (19)$$

for some discrete subset $\dot{\mathbb{D}}^3$ of the stimulus space.

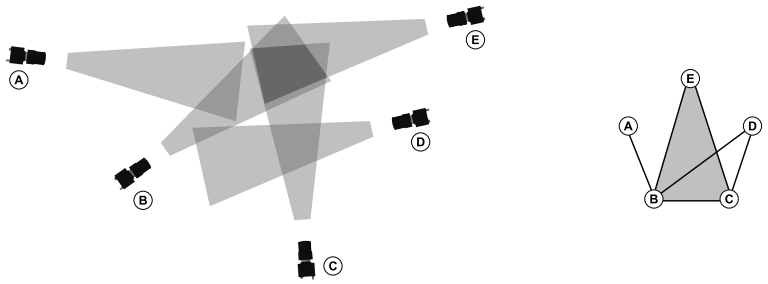

Fig. 2. Example Camera Network Layout with Coverage Hypergraph

Theorem 2: For any $L \subseteq M \in E_C$, $w_C(L) \geq w_C(M)$.

Proof: From the proof of Theorem 1, for all $\mathbf{p} \in \mathbb{D}^3$, $C_M(\mathbf{p}) \leq C_{M\backslash n}(\mathbf{p})$, so $|\dot{C}_M| \leq |\dot{C}_{M\backslash n}|$. Thus, for every $M \in E_C$, and every $M\backslash n \subset M$, $w_C(M\backslash n) \geq w_C(M)$. ■

Consider a partial hypergraph $\mathcal{H}_C^K = (N, E_C^K, w_C)$ of $\mathcal{H}_C$ with hyperedge subset

$$E_C^K = \{M \in E_C || M| \in K\} \tag{20}$$

where $K \subset \mathbb{Z}^+$. When $K = \{k\}$, we term this the k-coverage hypergraph of N. When $K = \{k, l\}$, we term this the k, l-coverage hypergraph of N, and so on.

Since E_C is an abstract simplicial complex, the 2-coverage hypergraph $\mathcal{H}_C^2$ is the (weighted) primal graph of $\mathcal{H}_C$, qualitatively equivalent to the vision graph as described in most other sources. We formally define the vision graph as $\mathcal{H}_C^2$.

IV. Task Processing Distribution

A. Problem Statement

Consider the portion of a k-ocular task in camera network N which involves processing data from all of $M \subseteq N$, where $|M| = k$; we shall term this an M-subtask. Only stimuli within $\langle C_M \rangle$ are relevant to an M-subtask. Given a relevance model R for the task, the expected processing load for a given M-subtask is proportional to $|\dot{C}_M \cap \dot{R}|$. Although this conjecture is tautological given that R is arbitrary, since R ideally represents the distribution of the stimuli necessary to perform the task, it is reasonable to assume in general that it also reflects the distribution of the processing load incurred by said stimuli. This is supported by empirical evidence [4], [17].

Assuming that N consists of smart camera nodes with equal local computational resources, the problem is to distribute the processing of all M-subtasks over the nodes such that the maximum load on any one node is minimized.

The set of eligible nodes to which M-subtasks may be assigned is restricted to M, for the following reasons:

1) *Robustness:* If a node $n \in M$ fails, the M-subtask can no longer be processed. Thus, assigning it to any $n \in M$ carries no risk of disrupting service for valid models.
2) *Locality:* In a large network, because the sensing range is finite, if $\langle C_M \rangle \neq \emptyset$, it is likely that nodes M are physically proximate. Since we assume nothing about the network structure, it is sensible to keep the M-subtask processing node physically local for communication efficiency.

The usefulness of this restriction is especially apparent in the special case $k = 1$, allowing camera-local subtasks (image preprocessing, etc.) to be included in the accounting.

Given a K-ocular task, where $K \subset \mathbb{Z}^+$, this problem can be solved by finding an orientation of H_C^K which minimizes the maximum weighted indegree.

B. Minimum Indegree Orientation

The *minimum maximum indegree orientation* problem for hypergraphs can be stated as follows. Given a simple, undirected, weighted hypergraph $\mathcal{H} = (V, E, w)$, find an orientation Λ of $\mathcal{H}$ which minimizes $\max_{u \in V}[\delta_\Lambda^i(u)]$.

This is equivalent to the scheduling problem of offline makespan minimization over identical parallel machines with eligibility constraints [18]; according to the three-field notation by Graham et al. [19], $P|M_j, M_j \neq M_k$ if $i \neq k|C_{\max}$. This is a special case of $P|M_j|C_{\max}$, which in turn is a special case of $R||C_{\max}$ [20]. The problem is NP-hard [21], but a number of approximation algorithms and search heuristics have been proposed.

We present here a local search heuristic based on the GR/EFF descent of Piersma and Van Dijk [13]. The main differences are the use of hypergraph notation and some simplifications made possible by constraints particular to our problem.

Initialization

Suppose the given hypergraph is $\mathcal{H} = (V, E, w)$. Let $\Lambda = (V, \vec{\mathcal{E}}, w)$, with $\vec{\mathcal{E}} = \emptyset$ initially.

Starting Point

Consider E in any order. For each $e \in E$, add e^u to $\vec{\mathcal{E}}$ such that $\delta_\Lambda^i[u] = \min_{v \in e} \delta_\Lambda^i[v]$.

Neighbourhood Search

1) Choose $v_{\max} \in V$ such that $\delta_\Lambda^i[v_{\max}] = \max_{v \in V} \delta_\Lambda^i[v]$. Let $\mathcal{R} = \{(v, e^{v_{\max}}) | v \in V\backslash v_{\max}, v \in e, e^{v_{\max}} \in \vec{\mathcal{E}}\}$.
2) If $\mathcal{R} = \emptyset$, go to Step 4. Otherwise, consider any $(v, e^{v_{\max}}) \in \mathcal{R}$; remove $(v, e^{v_{\max}})$ from $\mathcal{R}$.
3) If $\delta_\Lambda^i[v] < \delta_\Lambda^i[v_{\max}] - w(e)$, replace $e^{v_{\max}}$ with e^v in $\vec{\mathcal{E}}$ and go to Step 1. Otherwise, go to Step 2.
4) Sort V in nonincreasing order of indegree. Let v_1 and v_2 be its last and first elements, respectively.
5) Let $\vec{\mathcal{E}}_1 = \{e^{v_1} | v_2 \in e, e^{v_1} \in \vec{\mathcal{E}}\}$ and $\vec{\mathcal{E}}_2 = \{e^{v_2} | v_1 \in e, e^{v_2} \in \vec{\mathcal{E}}\}$. Let $\mathcal{I} = \vec{\mathcal{E}}_1 \times \vec{\mathcal{E}}_2$.

6) If $\mathcal{I} = \emptyset$, go to Step 8. Otherwise, consider any $(e_1^{v_1}, e_2^{v_2}) \in \mathcal{I}$; remove $(e_1^{v_1}, e_2^{v_2})$ from $\mathcal{I}$.
7) If $\max(\delta_\Lambda^i[v_1] - w(e_1) + w(e_2), \delta_\Lambda^i[v_2] - w(e_2) + w(e_1)) < \max(\delta_\Lambda^i[v_1], \delta_\Lambda^i[v_2])$, replace $e_1^{v_1}$ and $e_2^{v_2}$, respectively, with $e_1^{v_2}$ and $e_2^{v_1}$ in $\vec{\mathcal{E}}$ and go to Step 4. Otherwise, go to Step 6.
8) Let v_2 be the next element in V. If $v_2 = v_1$, let v_1 be the previous element in V and let v_2 be the first element in V. If v_1 is the first element of V, return Λ. Otherwise, go to Step 5.

V. Experimental Results

A. Description of Simulation

We test task distribution on a simulated network N of 23 camera nodes arranged in a virtual environment with walls and other occlusions. Our tasks are independent of the directional dimensions ρ and η; accordingly, we simplify the discussion by working exclusively in $\mathbb{R}^3$. A top view of the environment is shown in Figure 3, along with the relevance model R, which is uniform in z from 1.5m to 2.0m (with the floor at 0m, and all cameras at 2.5m), and the locations of the cameras.

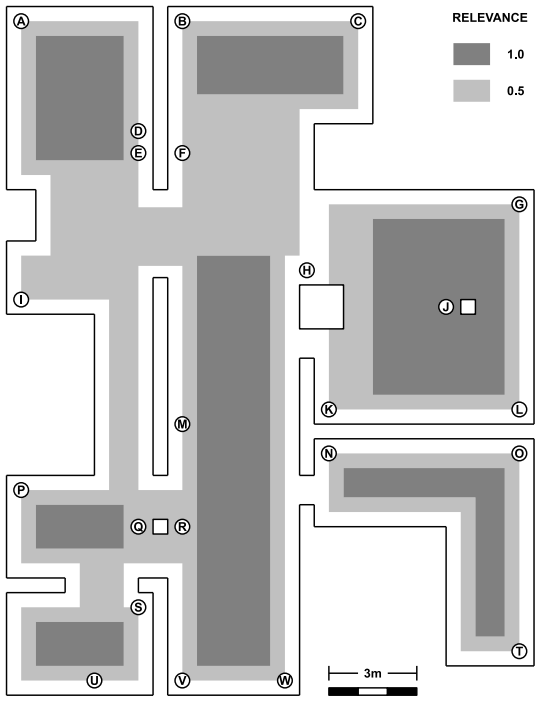

Fig. 3. Floor Plan and Relevance Model

The camera coverage strength models are derived from real parameters of a calibrated Prosilica EC-1350 1.3MP grayscale CCD camera with a Computar M3Z1228C-MP lens. The specific task parameters used are $\gamma = 20$, $R_1 = 0.3$, $R_2 = 0.01$, and $c_{\max} = 0.008$ (ζ_1 and ζ_2 are unused). Extrinsic parameters are defined manually to deploy the cameras in a reasonable arrangement covering the environment (82.42% coverage performance with respect to R).

The camera network and environment are simulated using our Adolphus[2] simulation software (Figure 4).

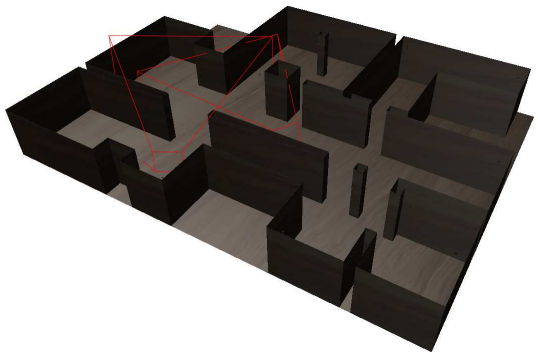

Fig. 4. Adolphus Showing $\langle C_I^i \rangle$ and $\langle C_M^i \rangle$

The coverage hypergraph $\mathcal{H}_C$ for N and R is computed over the discrete space $\dot{\mathbb{R}}^3 = \{(250x, 250y, 250z) | x, y, z \in \mathbb{Z}\}$, with coordinates in millimeters. Although it is too large to represent here graphically, Table I shows some statistics of the hyperedges in the complete $\mathcal{H}_C$.

TABLE I
Hyperedges in $\mathcal{H}_C$

Edge Size	Count	Mean Weight
1	23	750.51
2	78	155.66
3	130	50.13
4	152	23.49
5	122	14.09
6	61	9.37
7	17	6.40
Total	583	71.85

For each task, events of interest are points $\mathbf{p} \in \mathbb{R}^3$ generated randomly using $\lambda^{-1} R$ as a probability density function, where $\lambda = \iiint_{\mathbb{R}^3} R \, dx \, dy \, dz$. The detection probability for event $\mathbf{p}$ by camera node n is $C_n(\mathbf{p})$. Camera nodes individually detect events and are assumed to propagate their data to the appropriate nodes for processing.

B. Task 1: Generic Multi-View Processing

The first simulation experiment models a generic task in which each event is processed by every combination of camera nodes which detects it. Processing an event charges one unit of processing load to the node to which the combination is assigned (i.e., the vertex in $\mathcal{H}_C$ which is the head of the edge comprising the combination).

We generated 10,000 random events and assigned their processing to nodes according to Λ, the minimum maximum weighted indegree orientation of $\mathcal{H}_C$ approximated per the algorithm in Section IV-B. For comparison, we also assigned the same event detections using four other orientations of $\mathcal{H}_C$: the optimal unweighted minimum maximum indegree

[2]Adolphus is free software licensed under the GNU General Public License. Python source code and documentation are available at `http://github.com/ezod/adolphus`.

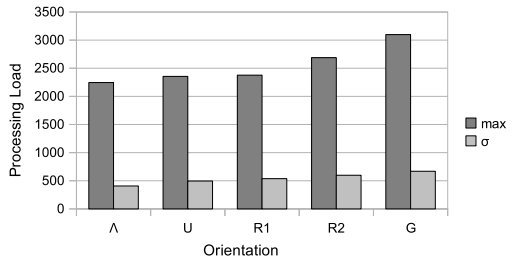

Fig. 5. Load Statistics for Task 1

orientation U, two random orientations R_1 and R_2, and a greedy orientation G (edges oriented in arbitrary order to the vertex with least indegree). Figure 5 shows the maximum and standard deviation of processing loads (with a mean of 1378.39) for each strategy.

The Λ distribution yields both the least maximum load and the most consistent distribution of load over the network, with improvements of 5% and 22%, respectively, over the next best strategy tested.

C. Task 2: Best-Pair Stereo Reconstruction

The second simulation experiment models a best-pair stereo reconstruction task. Hypothetically, upon detection of an event, camera nodes estimate their pairwise coverage of the event, then reach network-wide consensus on the pair with best coverage; the best pair then proceeds to perform a dense 3D reconstruction of the event. In our model, each estimation of pairwise coverage charges one unit of processing load to the assigned node, and each reconstruction charges five units of processing load to the assigned node for the best pair.

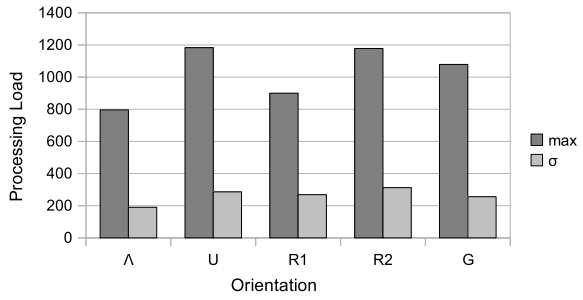

Fig. 6. Load Statistics for Task 2

We generated 2,000 random events and assigned their processing to nodes according to Λ, the minimum maximum weighted indegree orientation of $\mathcal{H}_C^2$. Again, we compare this to the unweighted solution U, two random orientations R_1 and R_2, and a greedy orientation G. Figure 6 shows the maximum and standard deviation of processing loads (with a mean of 373.87) for each strategy.

Again, the Λ distribution yields both the least maximum load and the most consistent distribution of load over the network, with improvements of 13% and 35%, respectively, over the next best strategy tested.

VI. Conclusions

The coverage hypergraph is a generalization of previous models of camera network coverage topology which fully captures node-level coverage relationships. As such, it is a useful combinatorial structure for optimization in distributed smart camera applications. We have demonstrated with simulated experiments its application to optimizing the distribution of task processing load, by adapting and applying an algorithm for a related scheduling problem.

This model is conceptually simple, but shows much promise as a powerful tool given that it has a strong, reliable theoretical foundation and tractability with a large volume of well-studied optimization techniques.

A. Future Work

Although the coverage strength model provides an excellent theoretical basis for defining the coverage hypergraph, in practice the necessary calibration parameters are often unavailable. We propose to construct $\mathcal{H}_C$ probabilistically from sensor data, following approaches used to construct other topological models. Cheng et al. [9] build the vision graph by pairwise matching of digests of local features. The exclusion approach of Detmold et al. [22], [23] also builds the vision graph, starting with a complete graph and eliminating (or reducing the likelihood of) edges when an occupancy mismatch is detected. Lobaton et al. [12] construct their *CN*-complex by matching detection and occlusion events.

Developing such a method would also allow us to attempt an experimental application using a real camera network without calibration, with one or more tasks of a less contrived nature than those in Section V. Our results currently depend on our assumptions about computational cost and detection probability holding in practice, since in simulation we have no means by which to generate events besides the relevance and coverage strength models, which are also used to construct the hypergraph itself. Our previous work [4], [17] provides some evidence that these assumptions are generally valid, but a complete real-world application would present a more convincing case.

The particular optimization over $\mathcal{H}_C$ presented in this work could be adapted to a variety of more complex task distribution scenarios. Multiple tasks with different computational costs could be combined into a single objective. Other problems aside from task processing distribution may require different interpretations of $\mathcal{H}_C$ (e.g. a redefinition of the weight function) and/or different optimization approaches.

Finally, it is ultimately desirable that any such optimization algorithms be decentralized, so that they may be computed on the camera network itself. This is a non-trivial problem and certainly warrants further investigation.

Acknowledgment

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] B. Rinner and W. Wolf, "An Introduction to Distributed Smart Cameras," *Proc. IEEE*, vol. 96, no. 10, pp. 1565–1575, 2008.

[2] Z. Zivkovic and R. Kleihorst, "Smart Cameras for Wireless Camera Networks: Architecture Overview," in *Multi-Camera Networks: Principles and Applications*, H. Aghajan and A. Cavallaro, Eds. Academic Press, 2009, ch. 21, pp. 497–510.

[3] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 2nd ed. Prentice-Hall, 2002.

[4] A. Mavrinac, J. L. Alarcon Herrera, and X. Chen, "A Fuzzy Model for Coverage Evaluation of Cameras and Multi-Camera Networks," in *Proc. 4th ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2010, pp. 95–102.

[5] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A Survey of Sensor Planning in Computer Vision," *IEEE Trans. Robotics and Automation*, vol. 11, no. 1, pp. 86–104, 1995.

[6] B. Wang, *Coverage Control in Sensor Networks*. Springer, 2010.

[7] H. Ma and Y. Liu, "Some Problems of Directional Sensor Networks," *Int. J. Sensor Networks*, vol. 2, no. 1-2, pp. 44–52, 2007.

[8] D. Devarajan and R. J. Radke, "Distributed Metric Calibration of Large Camera Networks," in *Proc. 1st Wkshp. on Broadband Advanced Sensor Networks*, 2004.

[9] Z. Cheng, D. Devarajan, and R. J. Radke, "Determining Vision Graphs for Distributed Camera Networks Using Feature Digests," *EURASIP J. Advances in Signal Processing*, 2007.

[10] G. Kurillo, Z. Li, and R. Bajcsy, "Wide-Area External Multi-Camera Calibration using Vision Graphs and Virtual Calibration Object," in *Proc. 2nd ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2008.

[11] E. J. Lobaton, S. S. Sastry, and P. Ahammad, "Building an Algebraic Topological Model of Wireless Camera Networks," in *Multi-Camera Networks: Principles and Applications*, H. Aghajan and A. Cavallero, Eds. Academic Press, 2009, ch. 4, pp. 95–115.

[12] E. J. Lobaton, R. Vasudevan, R. Bajcsy, and S. Sastry, "A Distributed Topological Camera Network Representation for Tracking Applications," *IEEE Trans. Image Processing*, vol. 19, no. 10, pp. 2516–29, 2010.

[13] N. Piersma and W. Van Dijk, "A Local Search Heuristic for Unrelated Parallel Machine Scheduling with Efficient Neighborhood Search," *Mathematical and Computer Modelling*, vol. 24, no. 9, pp. 11–19, 1996.

[14] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Computer Vision*. Springer, 2004.

[15] H. R. Tiwary, "On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes," *Discrete and Computational Geometry*, vol. 40, no. 3, pp. 469–479, 2008.

[16] A. Frank, T. Király, and Z. Király, "On the Orientation of Graphs and Hypergraphs," *Discrete Applied Mathematics*, vol. 131, no. 2, pp. 385–400, 2003.

[17] A. Mavrinac, J. L. Alarcon Herrera, and X. Chen, "Evaluating the Fuzzy Coverage Model for 3D Multi-Camera Network Applications," in *Proc. 3rd Int. Conf. Intelligent Robotics and Applications*, 2010, pp. 692–701.

[18] K. Lee, J. Y.-T. Leung, and M. L. Pinedo, "A Note on Graph Balancing Problems with Restrictions," *Information Processing Letters*, vol. 110, no. 1, pp. 24–29, 2009.

[19] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey," *Ann. of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[20] J. Y.-T. Leung and C.-L. Li, "Scheduling with Processing Set Restrictions: A Survey," *Int. J. Production Economics*, vol. 116, no. 2, pp. 251–262, 2008.

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[22] H. Detmold, A. R. Dick, A. Van Den Hengel, A. Cichowski, R. Hill, E. Kocadag, K. Falkner, and D. S. Munro, "Topology Estimation for Thousand-Camera Surveillance Networks," in *Proc. 1st ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2007, pp. 195–202.

[23] H. Detmold, A. R. Dick, A. Van Den Hengel, A. Cichowski, R. Hill, E. Kocadag, Y. Yarom, K. Falkner, and D. S. Munro, "Estimating Camera Overlap in Large and Growing Networks," in *Proc. 2nd ACM/IEEE Int. Conf. Distributed Smart Cameras*, 2008.