

Interconnected Network of Cameras

Mahdad Hosseini Kamal, Hossein Afshari, Yusuf Leblebici, Alexandre Schmid, Pierre Vandergheynst

Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ABSTRACT

The real-time development of multi-camera systems is a great challenge. Synchronization and large data rates of the cameras adds to the complexity of these systems as well. The complexity of such system also increases as the number of their incorporating cameras increases. The customary approach to implementation of such system is a central type, where all the raw stream from the camera are first stored then processed for their target application. An alternative approach is to embed smart cameras to these systems instead of ordinary cameras with limited or no processing capability. Smart cameras with intra and inter camera processing capability and programmability at the software and hardware level will offer the right platform for distributed and parallel processing for multi-camera systems real-time application development. Inter camera processing requires the interconnection of smart cameras in a network arrangement. A novel hardware emulating platform is introduced for demonstrating the concept of the interconnected network of cameras. A methodology is demonstrated for the interconnection network of camera construction and analysis. A sample application is developed and demonstrated.

Keywords: Multi-camera system, Omnidirectional vision, Light fields, Real-time, FPGA.

1. INTRODUCTION

High-end computing systems are built by integrating large numbers of commodity processors in a network formation. Multiple independent tasks can run at the same time on such systems. A single task if distributed properly among many processors will run faster. Similar concept is practised in the Graphical Processing Units (GPUs) at the VLSI chip level, where many processing cores are realized for intensive parallel processing of graphical applications. In digital imaging a similar trend is practised at the application level. It has been demonstrated that multiple images of a scene can be used to enhance the performance envelope of digital cameras. Increased image resolution,¹ compressive acquisition,² and dynamic range^{3,4} are examples of such enhancements. Real-time implementation of multi-camera systems' applications which demand high processing, favor a distributed and parallel approach for their realization. Due to the constraints posed by technology, the parallel and distributed approach might be the only feasible solution for the real-time realization of such systems. The widespread use of multi-camera systems benefits from a modular and distributed approach for their development at all levels ranging from mechanical construction, hardware, firmware and software. It is required to enhance the features and capabilities of customary cameras to enable modularity at all development levels of a multi-camera system. This enhancement can be achieved at the board level for customary imagers by adding external modules to them or at the VLSI chip level by improving the imager architectures to support modularity features for the purpose of integration in multi-camera systems. In light of this achievement a camera module is not solely an imaging device. It also includes processing and communication capabilities. The processing capability enables the camera module of local processing down to pixel level, while communicating features permits information exchange and distributed and or collaborative processing among the camera modules. Integrating all functionalities at the VLSI scale for a camera module enables the modular construction of multi-camera systems. Hence multi-camera systems can be constructed through interconnection of many camera modules embedded on a surface structure. It is envisioned that multi-camera systems are constructable by plugging camera modules on to their spots on a surface or back-plane structure. The surface or the back-plane structure provides the interconnectivity among the camera modules. The back-plane structure can be a PCB or any other alternative technology. Similar trend can be practiced at different scales in terms of size and dimension for camera modules and in general for multi-camera systems.

Further author information: (Send correspondence to Mahdad Hosseini Kamal)

E-mail: {mahdad.hosseinikamal, hossein.afshari, yusuf.leblebici, alexandre.schmid, pierre.vandergheynst}@epfl.ch

2. OMNIDIRECTIONAL VISION RECONSTRUCTION ALGORITHM

The Panoptic system is a polydioptric camera comprised of several cameras each having its own vision of the world. The aggregate field of view of all cameras construct the omnivision of the Panoptic system. Constructing an omnidirectional view on a point inside the hemisphere of the Panoptic camera corresponds to estimating the intensity value of all light rays in the 3D scene that would cross the observer location point for all directions. This processing combines the available data recorded by the Panoptic imagers, and takes advantage of all the light rays crossing all the imager focal points in all the directions present in the imager angle-of-view (AOV).

In this construction process, the omnidirectional view on a discretized sphere S_d of directions is estimated. The surface of this sphere is pixelized into an equiangular grid with N_θ latitudes and N_ϕ longitudes. The direction of each pixel in the omnidirectional view is identified by the unit vector $\vec{\omega} \in S_d$. The construction of the virtual omnidirectional view $\mathcal{L}(\vec{q}, \vec{\omega}) \in \mathbb{R}$, where $\vec{q} \in \mathbb{R}^3$ is the omnidirectional view point, for each omnidirectional pixel in direction $\vec{\omega}$ is performed in two algorithmic steps. First, all cameras participating to the construction, i.e. having $\vec{\omega}$ in their AOV, are determined. Since $\vec{\omega}$ does not necessarily belong to each camera pixel grid, a first level of interpolation is required to obtain the light intensity in that direction for each contributing camera. Second, an additional interpolation is performed in the space of light rays given by the direction $\vec{\omega}$ and passing through the camera origins. The aim is to estimate the intensity of a parallel light ray crossing the virtual camera center from these light rays. For the sake of simplicity, it is assumed that this center is localized in the hemisphere center, i.e. $\vec{q} = 0$, but the same developments can be generalized to any other observation point. Following shorthand is used $\mathcal{L}(\vec{\omega}) = \mathcal{L}(0, \vec{\omega})$.

2.1 First Algorithmic Step

The intensity $\mathcal{L}(\vec{\omega})$ of the virtual view is desired for estimation in the direction $\vec{\omega} \in S_d$. The cameras having $\vec{\omega}$ in their AOV amounts to determining all the camera index $0 \leq i < N_{\text{cam}}$ such that

$$\omega_{t_i} = \vec{\omega} \cdot \vec{t}_i > \cos \frac{\alpha}{2}, \quad (1)$$

where α is the camera AOV. The angle between $\vec{\omega}$ and \vec{t}_i is controlled to be smaller than $\frac{\alpha}{2}$. Having found the contributing cameras, the next step consists in translating the direction $\vec{\omega}$ in pixel coordinates of these cameras. Using the pinhole camera model,⁵ the contributing two dimensional position (x_{u_i}, x_{v_i}) on the i^{th} camera image plane (which is identified by coordinate unit vectors \vec{u}_i and \vec{v}_i) is expressed as:

$$(x_{u_i}, x_{v_i}) = -(\vec{\omega} \cdot \vec{u}_i, \vec{\omega} \cdot \vec{v}_i) \frac{f_L}{\omega_{t_i}}, \quad (2)$$

where f_L represents the camera focal length in (2). Fig. 1(a) illustrates an example of selecting the contributing cameras for a typical pixel direction $\vec{\omega}$ in a Panoptic device. The image frame and the planar arrays shown in Fig. 1(a) are fictitious and are only drawn for illustrative purposes resembling the image sensor array. Considering the selected direction of observation $\vec{\omega}$, indicated as an arrow in Fig. 1(a), the contribution of each imager into this direction is found at different (x_u, x_v) coordinate locations, since the imagers are not aligned. For example, the contribution of imager A in Fig. 1(a) into the direction of observation $\vec{\omega}$ consists of pixel A_ω .

The contributing position on the image frame of each contributing camera is likely not to coincide with an exact pixel location of a camera image frame. The light intensity of the contributing position is the desired quantity. The light intensity of the contributing position can be estimated by the light intensity of the nearest actual pixel location to the contributing position. An alternate method consists of using interpolation among light intensities of the actual neighboring pixels of the contributing position. As a final result, the first algorithmic step estimates the values $\mathcal{L}(\vec{c}_i, \vec{\omega})$ for each contributing camera i satisfying (1), which is the intensity of all light rays parallel to $\vec{\omega}$ as recorded by these cameras.

2.2 Second Algorithmic Step

Having obtained the intensity $\mathcal{L}(\vec{c}_i, \vec{\omega})$ of all the contributing cameras i in direction $\vec{\omega}$, a second algorithmic step is required in order to acquire the final intensity $\mathcal{L}(\vec{\omega})$ for the virtual omnidirectional vision on the hemisphere

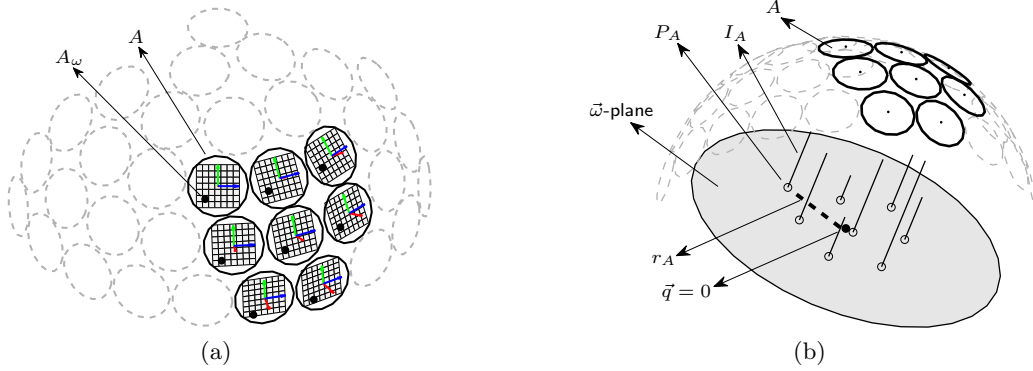


Figure 1: (a) Cameras contributing to $\vec{\omega}$ direction with their corresponding pixels. (b) Projection of the camera centres contributing in direction $\vec{\omega}$ onto the planar surface normal to $\vec{\omega}$.

center. Assuming that light intensity remains constant on the trajectory of any light ray (constant light flux (CLF) assumption), i.e. $\mathcal{L}(\vec{q}, \vec{\omega}) = \mathcal{L}(\vec{q} + \lambda\vec{\omega}, \vec{\omega}) \quad \forall \lambda \in \mathbb{R}$. An orthographic plane is defined as the perpendicular plane to a given direction $\vec{\omega}$. The light ray intensity only varies in the orthographic plane.

The orthographic plane is indicated as the “ $\vec{\omega}$ -plane” for the direction $\vec{\omega}$ in Fig. 1(b). Following the CLF assumption, the $\vec{\omega}$ -plane can be considered crossing the virtual camera center, which is the center of the sphere in our simplified analysis. The sphere center is marked by a bold point in 1(b). The light rays of direction $\vec{\omega}$ recorded by each contributing camera intersect the $\vec{\omega}$ -plane in points that are the projection of the camera focal points on this plane.

The projected points of the contributing camera positions in $\vec{\omega}$ onto the $\vec{\omega}$ -plane are highlighted by hollow points 1(b). Following the CLF assumption, each projected camera position P_{c_i} on the planar surface is assigned an intensity value $\mathcal{L}(\vec{c}_i, \vec{\omega})$.

As an example, the camera indicated as A of position c_A in Fig. 1(b) contributes in direction $\vec{\omega}$. The contributing pixel position on the image frame of the camera A is denoted as A_ω . The projection of the camera center c_A onto the $\vec{\omega}$ -plane is indicated as P_A . The position $P_A = P_{c_A}$ on the $\vec{\omega}$ -plane is assigned the intensity value $I_A = \mathcal{L}(\vec{c}_A, \vec{\omega})$. The intensity value I_A is the light intensity observed by the pixel position A_ω of the image frame of camera A . The same process is applied to the seven other participating cameras. For the eight intensity values in our illustration, the intensity value which is observed into direction $\vec{\omega}$ can be estimated by the aggregate participating cameras through a two-dimensional interpolation, i.e. using an algorithmic aggregate of the eight intensity values such as the linear interpolation estimation method.

2.3 Linear Interpolation

The linear interpolation scheme incorporates all or subset of the contributing camera intensity values through a linear combination. This procedure is conducted by aggregating the weighted intensities of the contributing cameras. The weight of a contributing camera is an inverse function of its distance (i.e. r_i) between its projected focal point and projected virtual observer point on the $\vec{\omega}$ plane. The weights are also normalized to the sum of the inverse of all the contributing cameras distances. The linear interpolation is expressed in (3).

$$\mathcal{L}(\vec{q}, \vec{\omega}) = \frac{\sum_{i \in \mathcal{I}} \frac{1}{r_i} \mathcal{L}(c_i, \vec{\omega})}{\sum_{i \in \mathcal{I}} \frac{1}{r_i}}. \quad (3)$$

The selection of the number of incorporating cameras in the linear interpolation per $\vec{\omega}$ direction estimation primarily depends on the actual size of the Panoptic device sphere and hence how dense the cameras are arranged on its surface. The distance of objects in the scene from the Panoptic system would also effect the correct selection of the number of incorporating cameras in the linear interpolation scheme. Assuming the objects far enough from the Panoptic sphere radius (i.e. $R \gg r_\odot$) all the contributing cameras for the infinite case can be incorporated in the linear interpolation scheme.

3. DISTRIBUTED HARDWARE IMPLEMENTATION

For distributed implementation of the omnidirectional algorithm each i^{th} camera is required the knowledge of its covering directions (i.e. \mathcal{L}_{S_i}) and the information of the other contributing cameras for all of these directions (i.e. $\mathcal{C}_{i,\vec{\omega} \in \mathcal{L}_{S_i}}$). The contributing weight factors per each supporting observation direction must be known by each i^{th} . These weight factors are denoted as $\mathcal{A}_{i,\vec{\omega} \in \mathcal{L}_{S_i}}$ for each i^{th} camera and $\vec{\omega}$ direction. The $\mathcal{A}_{i,\vec{\omega}}$ is derivable from (3). As mentioned the exact pixel position on the camera image frame per $\vec{\omega}$ direction is also required to be calculated. This information is required for correct access of the pixels on the camera image frame for each $\vec{\omega}$ direction. These pixel positions information are denoted as $\mathcal{P}_{i,\vec{\omega} \in \mathcal{L}_{S_i}}$. These pixel positions information are extractable through the calibration procedure of the Panoptic system.

For each i^{th} camera in the system, the $\vec{\omega} \in \mathcal{L}_{S_i}$, $\mathcal{C}_{i,\vec{\omega} \in \mathcal{L}_{S_i}}$, $\mathcal{A}_{i,\vec{\omega} \in \mathcal{L}_{S_i}}$ and the $\mathcal{P}_{i,\vec{\omega} \in \mathcal{L}_{S_i}}$ needs to be calculated. In effect for a constant set of $\vec{\omega}$ directions for each camera these parameters are only required to be calculated once and stored in a local memory for real-time access. The required information can be calculated once by the central unit and updated to the local memory of the camera modules. It is also feasible that each camera module calculates its own required information using its own processing features. The manipulated information can be stored on the cameras local memory for use in real-time processing of a target application.

3.1 Omnivision Reconstruction Procedure

For the purpose of distributed vision reconstruction all the camera modules first go through an initialization phase. In the initialization phase the cameras are synchronized and the required information for each of them is calculated and written to their local memory. After the initialization phase the camera modules are triggered to start operating at the same time in the omnivision reconstruction mode. Upon entering the omnivision construction mode each camera modules waits in an idle state. The camera modules transit from the idle state to a processing state when a new image frame is ready. In the processing state each camera module scans through all of its predefined directions. The predefined directions are within the circumscribing zonal segment of the cameras field of view. Each observing direction information is retrieved, gathered and combined from other cameras in the linear interpolation case and send to the central unit. After all directions are scanned the camera modules transits to the idle state and wait for the arrival of the next image frame. Information exchange is achieved through packet based communication with the central unit and other cameras.

In the interpolation state, all the $\vec{\omega} \in \mathcal{L}_{\mathcal{Z}}$ directions are scanned. Each $\vec{\omega} \in \mathcal{L}_{\mathcal{Z}}$ is checked for being best observed by the camera module. For each best observing $\vec{\omega}$ its corresponding light information from other contributing cameras are requested. The request mechanism is conducted through a request packet sending mechanism via the router port of the camera module to other camera modules. The contributing camera modules respond to this request packet via sending and acknowledge packet back. The camera module waits for all the acknowledge packets to be fully received. The light information embedded within each acknowledge packet is extracted and added to the camera modules own light information. The final light information value is send via a packet to the central unit. Since each camera in the linear interpolation receives information requests from other cameras, a separate process is envisioned for acknowledging these request.

This request response process has two states. The idle state and the acknowledge state. In the idle state of this process waits for the arrival of a new light information request. When an information request is received the acknowledge state is entered. In acknowledge state the request is serviced. Several request are acknowledge one at a time. Hence information requests are queued in order for all of them being acknowledged by the camera module. The acknowledge state is only left when there is no more information request for servicing by the camera module.

4. INTERCONNECTION NETWORK

The majority of digital systems are composed of three building blocks, logic, memory and communication.⁶ Logic is for generating the necessary data, like arithmetical results. Memory is used for data storage. And communication is the movement of data from one place to another. Performance of a hardware is often limited by the communication speed of digital systems. Speed of communication is not as fast as speed of memories and

processors. And wire delays are also considerably higher compared to gate delays. Hence, the communication speed is the bottleneck of modern digital systems.

Interconnection network is a programmable system that transports data between terminals. When a terminal needs to communicate with another terminal, the network is configured to establish a communication link between them. Message passing occurs between terminals. Each node generates a message which includes the information of the target node address inside the network and data. Interconnected network is responsible for delivering this message to the destination node. Communication speed is the limiting performance factor of many digital systems today. For example, two key performance factors in a computer system are memory latency and memory bandwidth of the interconnection network between processor and memory. Fast interconnection networks are necessary and its absence is a critical bottleneck in most digital systems.

4.1 Camera Assignment Problem

The selection of a regular graph based topology is based on its cost and performance. The cost is determined by the number and complexity of the chips required to realize the network, and the density and length of the interconnections on board or over cables, between these chips. Performance of interconnection networks is identified with two parameters: bandwidth and latency, which both of these factors are determined by factors other than topology, for example flow control, routing strategy and traffic pattern.⁶ A topology is evaluated through its bisection bandwidth,^{7,8} channel load, and path delay.⁶

After selection of a regular topology for the implementation of an interconnected network of cameras, a strategy is required to assign the cameras to the nodes of the target topology. For the purpose of application development an assignment is desirable which conserves the geometrical neighbouring of the cameras in the target topology as much as possible. The majority of information exchanges occur among the geometrically neighbouring cameras. Therefore such an assignment improves the mutual access times among these cameras, provided that sufficient bandwidth is available, on the target topology and hence improve the over all performance of the interconnected network.

4.1.1 Quadratic Assignment Problem

The assignment of cameras to a target interconnected network nodes can be defined in the context of a facility allocation problem known as the Quadratic Assignment Problem. The QAP models the following real-life problem: There are a set of n locations. For each pair of locations, a distance is specified and for each pair of facilities a weight or flow is specified. The problem is to assign all facilities to different locations with the goal of minimizing the sum of the distances multiplied by the corresponding flows. The formal definition of the QAP is as follows: Given two sets P (facilities) and L (locations), of equal size, together with a weight function w and distance function d , the bijection f is desired that minimizes the following cost function

$$\sum_{a,b \in P} w(a,b) \cdot d(f(a), f(b)). \quad (4)$$

The facilities in our QAP case are the cameras and its locations are the nodes in a target network topology. The distance function is equivalent to the distance matrix of the target network graph topology. In graph theory the distance matrix is a matrix containing the pairwise minimum distances of the nodes. The distance matrix has a size of $N \times N$ where N is the number of nodes. The nodes of the extracted graph represent the cameras and its edges resemble the neighbouring of the cameras. Hence in the latter graph two nodes are connected if their respective cameras are geometrical neighbours diagram. The adjacency matrix of this graph can be used as the flow matrix in the QAP. The adjacency matrix indicates which nodes of a graph are adjacent to which other nodes. The adjacency of a simple graph with N nodes is an $N \times N$ matrix where its non-diagonal entry a_{ij} is the number of edges from node i to node j , and its diagonal entry a_{ii} is all zero. Further more a weighted adjacency matrix can be utilized which can provide a better approximation of the information exchange flow among the cameras.

The QAP is a NP-hard problem so there is no known algorithm for solving this problem in polynomial time, and even small instances may require long computation time. Nevertheless several heuristic approaches to solve QAP have been proposed in the literature.⁹ A QAP problem has been solved for assigning the cameras

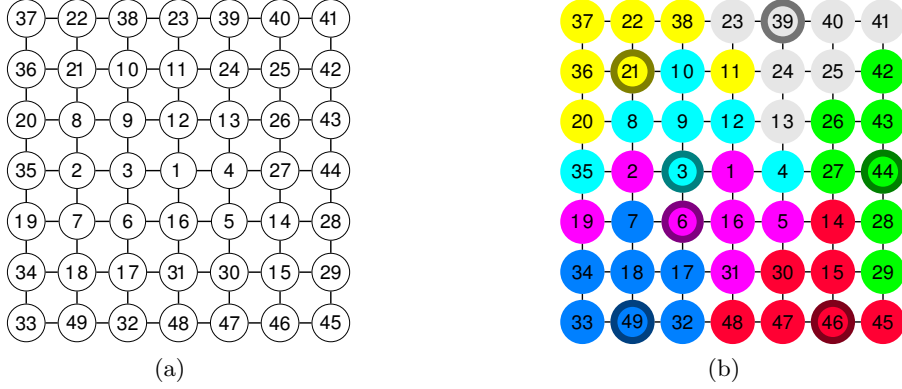


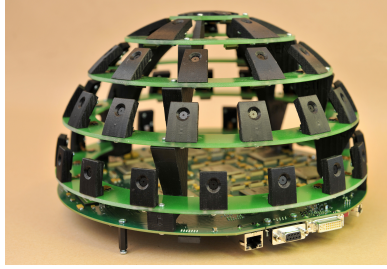
Figure 2: (a) The assigned 7×7 mesh topology interconnected network. (b) The 7×7 mesh topology with 7 vertex p-centers.

of the five-floor Panoptic system containing $N_{\text{cam}} = 49$ cameras to a 7×7 mesh topology graph as the target interconnection network. The sparse version of the GRASP algorithm¹⁰ is used for solving the QAP. The adjacency matrix of the graph is chosen as the flow matrix. The assigned camera numbers is represented on the mesh graph shown in Fig. 2(a). The assignment allocates the cameras such that all geometrical neighbouring cameras are not more than three hops distant away from each other in the target topology. The number of nodes in the target topology and the cameras of the Panoptic system are the same in the demonstrated example. The same method is applicable if the number of the nodes in the target topology is greater than the number of cameras of the Panoptic system assuming to have cameras with no flow exchanges with other cameras. This solution is considered when no regular based graph topology is selectable to support the exact number of cameras of the Panoptic system.

4.2 Vertex-p center problem

Having a distributed camera system does not imply the omission of a central unit. For example a central unit is required for the cameras to send their processed information for the purpose of display. Also a hybrid approach for the application deployment can be considered where some of the processing is done distributed at the camera level and the rest of the processing is conducted in the central unit. For this purpose it is preferred that all the distributed cameras also have a direct access to a central unit. This feature is not feasible or optimal in most cases. A central unit might not have enough ports to interface with all the cameras of the system. In case where all the cameras are connected to the central unit with distinct interfaces, and the respective bandwidth of these connections are not fully utilized, an inefficient usage of resources is taking place. Hence it is more efficient to provide some of the cameras with direct accessing capability to the central unit and share these connections with the cameras that do not have a direct interface to the central unit. The availability of an interconnection network permits the utilization of this strategy. The latter concept is depicted for the Panoptic system with $N_{\text{cam}} = 49$ cameras in Fig. 2.

The problem to solve is which p cameras to select to have direct access to the central unit so that the rest of the cameras can access the central unit with minimum number of hops. This feature is desired for reduction of access times between the central unit and any camera on the interconnected network, assuming sufficient channel bandwidth is available. The latter problem can also be mapped into another known facility allocation problem known as the vertex p -center problem. The basic p -center problem consists of locating p facilities and assigning clients to them so as to minimize the maximum distance between a client and the facility it is assigned to. This problem is also known to be NP-hard.¹¹ In the capacitated version of the p -center problem, each client is labelled with some quantitative demand, and the assignment of clients to facilities is constrained with capacity restrictions of facilities (i.e. the total demands of clients assigned to a certain facility cannot exceed the facility's capacity). The capacitated p -center problem is articulated as locating p capacitated facilities on a network and assigning clients to them within capacity restrictions to minimize the maximum distance between a client and the facility that is assigned to it.



(a)



(b)

Figure 3: (a) View of a Panoptic system with 49 cameras. (b) Omnidirectional reconstruction of a scene with the Panoptic system composed by 15 cameras.

The cameras that are provided with direct access to the central unit are considered as facilities and the others as clients. The number of facility cameras depends (i.e. the number of p) on the capacity of a single facility. The total demand of all the clients must be handled by all the facilities in general. As an example a vertex 7-center problem has been solved for the mesh graph topology depicted in Fig. 2(b) assuming that each facility can support up to 7 clients. The problem is solved using an exact algorithm for the capacitated vertex p -center problem 12. The solution is depicted in Fig. 2. All the cameras acting as facility (i.e. with access to central unit) are shown with a bolder edge. The clients belonging to the same supporting facility are also filled with similar colors. It is observed that all clients are at most two hops away from their supporting facility.

5. SIMULATION RESULTS

An interconnected camera network is designed for a Panoptic system with $N_{\text{flo}} + 1 = 5$ floors and $N_{\text{cam}} = 49$, shown in Fig. 3(a). The target application of the Panoptic system is an omnidirectional vision reconstruction with a resolution of $N_{\phi} \times N_{\theta} = 1024 \times 768$ at 25 frame per seconds. As the Panoptic system with 49 cameras is not yet completely functional, Fig. 3(b) demonstrates the omnidirectional view of a scene reconstructed by the Panoptic system with 15 cameras based on linear interpolation and a central interpolation algorithm.¹³ We expect a better reconstruction quality with 49 cameras. The AOV of the all camera imagers is assumed 60° . The algorithm is distributed and conducted in parallel among the cameras. The target network topology for the camera interconnections is a 7×7 mesh graph. The maximum flit size of the packets for information exchange in the system is chosen as four. For information exchange between the cameras request and acknowledge packets are used. The number of flits for the request packets is chosen as three and four for the acknowledge packets. The number of flits for the packet exchange occurring between the camera modules and the central unit is chosen as four.

For the purpose of performance analysis of the interconnection network of cameras the “BookSim” simulator¹⁴ is used. The BookSim simulator is a C++ based cycle accurate interconnection network simulator developed by Concurrent VLSI Architecture Laboratory of the Stanford University. The BookSim is a parametric configurable simulator, permitting the selection of different network topologies and parameters for the routers of the interconnection network. The BookSim simulator can be used to survey the effect of router parameters such as number of virtual channels and buffer size on the over all interconnection network performance.

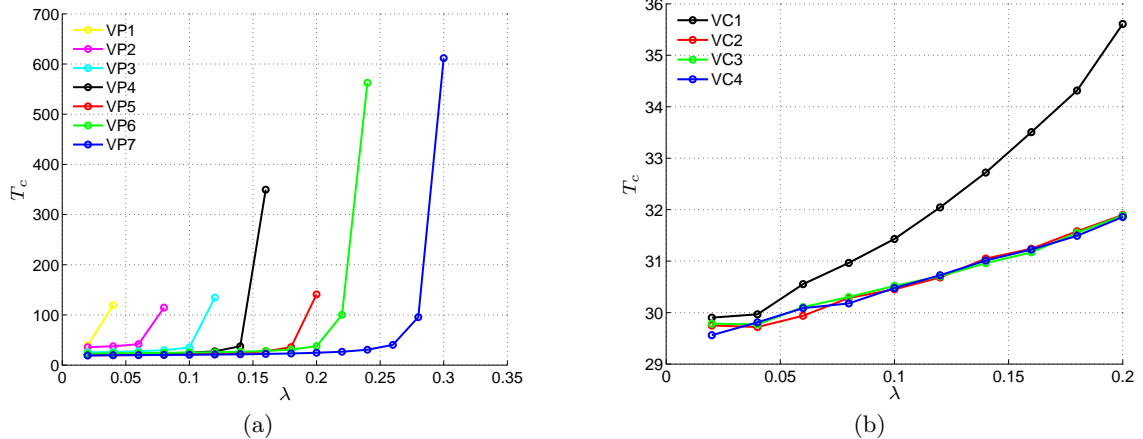


Figure 4: The average packet latency for different throughputs. (a) The effect of considering different number of vertex-p centres. (b) The average packet latency for different number of routers virtual channels.

As mentioned there is no need for connecting all the camera modules directly to the central unit. A few can be selected for direct access to the central unit and their connection can be shared with other camera modules through the interconnection network. The number and the positions of these camera modules also have an impact on the interconnection network performance demand. The position for the p cameras to be connected to the central unit is obtained from the capacitative version of the vertex- p center problem of a graph. In the first experiment, the number of vertex- p centres is varied from one to seven. Since random assignment is not unique, multiple random assignments have been conducted and the overall average result of all random assigned cases is sketched for the latency (T_c) against throughput λ . Fig. 4(a) demonstrate the latency T_c against throughput λ for routers with flit buffer size 16. The number of virtual channels for the routers are selected as one in the simulation. The latency respect to throughput graphs are drawn up to the point which stability was achieved for their simulation. It is inferred from the graphs shown in Fig. 4(a) that as the number of vertex- p center increases the network sustain higher packet injection rates. Whereas the number of vertex- p centres decrease the network can sustain less packet injection rate. The increase in the router buffer size does not improve the packet injection rate or latency. The number of vertex- p centres should be chosen high enough for the network sustain the requested demand.

In another experiment, the number of vertex- p centres is set to seven and the cameras are randomly assigned to the networks nodes in the simulation. Fig. 4(b) demonstrates the latency T_c against throughput λ for different virtual channels of routers with buffer size 16. The latency is expressed in terms of clock cycles and the throughput is normalized to a single channel bandwidth. It is inferred that the buffer size does not have considerable impact on packet latency as long as the injection rates are below $\lambda < 0.2$. Increase in the number of virtual channels decreases the packet latency for the same packet injection rate. The latter decrease is greater when the injection rate increases. Increase in the number of virtual channels beyond two does not improve the packet latency for the same injection rates.

6. CONCLUSION

In this paper the concept of the interconnected network of camera is introduced. The implementation of the omnidirectional vision reconstruction algorithm of the Panoptic camera in the context of a distributed camera module system is elaborated. The method presented is also applicable to any other spatial camera arrangement in 3D space and interconnected network topology. The described method is applicable for problems when the central unit has limited number of direct access connections to the cameras. The simulation results demonstrate the effectiveness of our distributed reconstruction model for different network parameter.

The realization of multi-camera systems through interconnected network of camera modules with extended features offers more reliability in case of connection failures as compared to the centralized case where the failure

of a connection from a camera module to the central unit effectively removes the camera module from the system. Failures are not limited to connection type but also to devices itself. The utilization of several cameras for a visual system also offers a degree of assurance for a target application functionality in case of failure of some cameras. Fault detection and recovery strategy for such systems are open for investigation. In addition an interconnected network of camera also offers the feature of process load balancing for applications. In case the imaging device of a camera module is not utilized, its processing features can be exploited by other camera modules. In a networked formation, the camera modules of a system are not bound to run a single application. Several cameras might pursue an application different from others. Hence several applications or modes might be run using the system at the same time. Therefore partial dynamic reconfiguration at the application and system level is another feature of an interconnected network of camera. Dynamic load balancing and system reconfiguration are open for investigation for different applications.

REFERENCES

- [1] Szeliski, R., "Image mosaicing for tele-reality applications," *IEEE Workshop on Applications of Computer Vision* , 44–53 (dec 1994).
- [2] Hosseini Kamal, M. and Golbabaee, M. and Vanderghenst, P., "Light Field Compressive Sensing in Camera Arrays," *ICASSP* (2012).
- [3] Mann, Picard, Mann, S., and Picard, R. W., "On Being 'undigital' with Digital Cameras: Extending Dynamic Range By Combining Differently Exposed Pictures," *Proceedings of IS&T* , 442–448 (1995).
- [4] Debevec, P. E. and Malik, J., "Recovering High Dynamic Range Radiance Maps from Photographs," *ACM SIGGRAPH* , 1–10 (2008).
- [5] Hartley, R. I. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press (2004).
- [6] Dally, W. and Towles, B., [*Principles and Practices of Interconnection Networks*], Morgan Kaufmann Publishers Inc. (2003).
- [7] Thomborson, C., "The economics of large-memory computations," *Inf. Process. Lett.* , 263–268 (1998).
- [8] Thompson, C. D., *A Complexity Theory for VLSI*, PhD thesis (1980).
- [9] Burkard, R. E., Karisch, S. E., and Rendl, F., "QAPLIB - A Quadratic Assignment Problem Library," tech. rep. (1996).
- [10] Pardalos, P. M., Qian, T., Mauricio, and Resende, G. C., "A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem," *Quadratic Assignment and Related Problems* , 237–261 (1994).
- [11] Kariv, O. and Hakimi, S. L., "An algorithmic approach to network location problems. I: The p-centers," *SIAM Journal on Applied Mathematics* **37**, 513–538 (1979).
- [12] Pinar, M. C. and Ozsoy, F. A., "An Exact Algorithm for the Capacitated Vertex p-center Problem," *Computers and Operations Research* , 1420–36 (2006).
- [13] Afshari, H., Popovic, V., Tasci, T., Schmid, A., and Leblebici, Y., "A Spherical Multi-camera System with Real-time Omnidirectional Video Acquisition Capability," *Consumer Electronics , IEEE Trans. on* (2012).
- [14] Jiang, N., Micheliannakis, G., Daniel Becker, B. T., and Dally, W., "Booksim 2.0 Users Guide," (2010).