Real-Time Omnidirectional Imaging System with Interconnected Network of Cameras

Kerem Seyid[(✉)], Ömer Çogal, Vladan Popovic,
Hossein Afshari, Alexandre Schmid,
and Yusuf Leblebici

Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
`kerem.seyid@epfl.ch`

Abstract. The Panoptic camera is an omnidirectional multi-aperture visual system which is realized by mounting multiple imaging sensors on a hemispherical frame. In this chapter, we will present a novel distributed and parallel implementation of the real-time omnidirectional vision reconstruction algorithm of the Panoptic Camera, for camera modules with processing, memory and interconnectivity features. A methodology for the arrangement of camera modules with interconnectivity feature into a target interconnection network topology will be introduced. A unique custom-made multiple-FPGA hardware platform designed for the implementation of an interconnected network of a 49 camera prototype Panoptic system will be explained. A novel way to represent the omnidirectional data obtained from the Panoptic Camera and real-time high dynamic range (HDR) video application which is realized by the system will be presented.

Keywords: Field programmable gate arrays · Smart cameras · Real-time systems · Omnidirectional imaging · Telepresence · High dynamic range

1 Introduction

A trend in constructing high-end computing systems consists of parallelizing large numbers of processing units. A similar trend is observed in digital photography, where multiple images of a scene are used to enhance the performance of the capture process. The technique is called a multi-view imaging (MVI) and has attracted increasing attention due to the dropping cost of digital cameras [1]. Novel research themes and applications such as increasing image resolution [2], obtaining high dynamic range images [3,4], object tracking/recognition, environmental surveillance, industrial inspection, 3DTV, and free viewpoint TV (FTV) [5] are receiving increasing attention.

Most developed camera array systems are bulky and not easily portable platforms. Their control and operation depend on multi-computer setups. In addition, image sensors on camera arrays are usually mounted on planar surfaces

which prohibits them from covering the full view of their environment. Full view or panoramic imaging finds application in various areas such as autonomous navigation, robotics, telepresence, remote monitoring and object tracking. Several solutions for acquiring omnidirectional images and their application have been presented in [6].
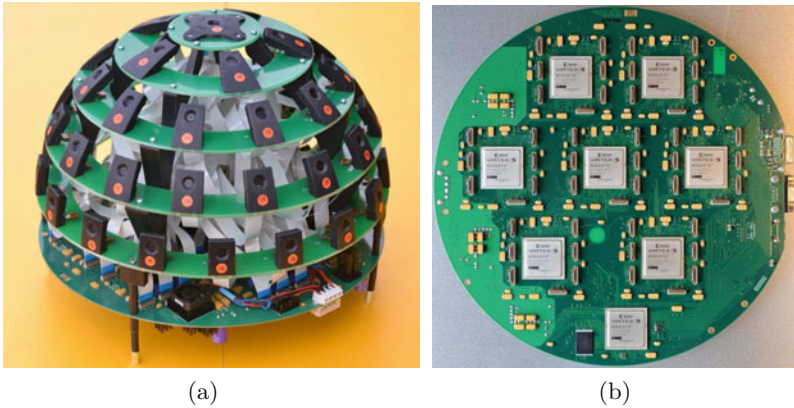

(a)   (b)

Fig. 1. (a) Built Panoptic prototype with 5 floors and 49 cameras. The sphere diameter of the prototype is $2r_\odot = 30$ cm. (b) Top view of the Panoptic Media FPGA-based development platform.

Early systems for capturing multiple views were based on a translating [7] or rotating [8] high-resolution camera for capturing, while rendering was carried out in post processing. The latter concept requires a long acquisition time. These ideas were later extended to a dynamic scene by using a linear array of still cameras [9]. For capturing large data sets, researchers focused on arrays of video cameras. In addition to the synchronization of the cameras, very large data rates present new challenges for the implementation of these systems. The first camera array systems were only built for recording and later post processing on Personal Computers [10]. Other such systems [11,12] were built with real-time processing capability for low resolution and low frame rates. A general-purpose camera array system was built at Stanford University [13] with limited local processing at the camera level. This system was developed to support recording of large amounts of data and subsequent intensive offline processing, but not for real-time operations.

In [14,15], real-time systems with six cubically arranged cameras are presented. These systems utilise high resolution imagers with a low number of cameras. Another six-camera panorama system with high resolution output is presented in [16]. Google Street View is one example of high resolution and increased number of cameras. The system in [17] is a 360° imaging system comprising 15 5 MP cameras, which covers 80 % of its surroundings. Lately, a novel system consisting of 44 5 MP cameras has been presented in [18], which offers

an output resolution over 82 MP with offline processing. Another camera system which is able to acquire an image frame with more than 1 Gigapixel resolution was developed [19]. The system uses a very complex lens system comprising a parallel array of micro cameras to acquire the image. Due to the extremely high resolution of the image, it suffers from a very low frame rate, even at low output resolution. Recently, a method for implementing bio-inspired cameras with hemispherical view is presented in [20]. However, it is limited to only 180 pixels.

An original approach for creating a multi-camera system distributed over a spherical surface is presented in [21]. This multi-camera system is referred to as the Panoptic camera. The Panoptic camera is an omnidirectional imager capable of recording light information from any direction around its center. It is also a polydioptric system [22] where each CMOS camera sensor has a distinct focal plane. The previously built Panoptic system is explained detailed in [21]. The system is implemented with a centralized approach, where data acquisition and data processing reside on the same unit. Figure 1 depicts the new Panoptic Media Platform of 5 floors and 49 cameras. The new prototype and architecture presented in this chapter aim to implement the reconstruction algorithm in a parallel and distributed fashion, where image processing applications reside at the camera level.

First, omnidirectional vision reconstruction algorithm will be presented in Sect. 2. Detailed explanations of the distributed and parallel implementation of vision reconstruction are given in Sect. 3. A definition of an interconnected network of cameras and a methodology to solve the camera assignment problem is given in Sect. 4. The details of a custom-made FPGA platform designed for the practice of the concept of an interconnected network of cameras are given in Sect. 5 with implementation and imaging results. An immersive way for visualizing the omnidirectional data is presented in Sect. 6. A real-time 360° high dynamic range (HDR) video application with Panoptic Camera is presented in Sect. 7. Future work is presented in Sect. 8.

2   Omnidirectional Vision Reconstruction Algorithm

The omnidirectional vision of a virtual observer located anywhere inside the hemisphere of the Panoptic structure can be reconstructed by combining the information collected by each camera in the light ray space domain (or light field [23]).

In this process, the omnidirectional view is estimated on a discretized spherical surface S_d of directions. The surface of this sphere is discretized into an equiangular grid with N_θ latitudes and N_ϕ longitudes samples, where each sample represents one pixel. Figure 2(a) shows a pixelized sphere with sixteen pixels for N_θ and N_ϕ each. A unit vector $\boldsymbol{\omega} \in S_d$, represented in the spherical coordinate system $\omega = (\theta_\omega, \phi_\omega)$, is assigned to the position of each pixel. Comparison of different pixel distributions over the sphere are discussed in Sect. 3.2.

The construction of the virtual omnidirectional view $\mathcal{L}(\boldsymbol{q}, \boldsymbol{\omega}) \in R$, where $\boldsymbol{q}$ determines the location of the observer, is performed in two steps. The first
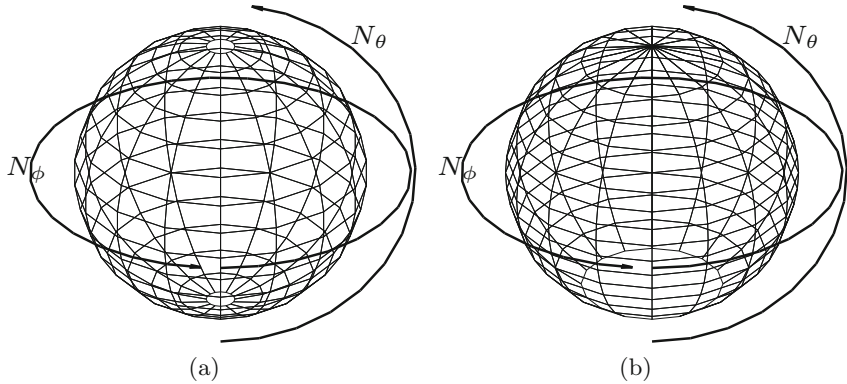
Fig. 2. Discretized sphere surface with $N_\theta = 16$ latitudes and $N_\phi = 16$ longitudes (256 pixels) (a) equi angular and (b) equal density pixelation.

step consists of finding a pixel in each camera image frame that corresponds to the direction defined by $\boldsymbol{\omega}$. The second step consists of blending all pixel values corresponding to the same $\boldsymbol{\omega}$ into one. The result is the reconstructed light ray $\mathcal{L}(\boldsymbol{q}, \boldsymbol{\omega})$.

To reconstruct the omnidirectional view, all the cameras having an $\boldsymbol{\omega}$ in their angle of view are first determined. To extract the light intensity in that direction for each contributing camera, a pixel in the camera image frame has to be found. Due to the rectangular sampling grid of the cameras, the $\boldsymbol{\omega}$ does not coincide with the exact pixel grid locations on the camera image frames. The pixel location is chosen using the nearest neighbour method, where the pixel closest to the desired direction is chosen as an estimate of the light ray intensity. The process is then repeated for all $\boldsymbol{\omega}$ and results in the estimated values $\mathcal{L}(c_i, \boldsymbol{\omega})$, where c_i is the radial vector directing to the center position of the ith contributing camera's circular face. Figure 3(b) shows an example of the contributing cameras for a random pixel direction $\boldsymbol{\omega}$ depicted in Fig. 3(a). The contributing position A_ω of the camera A, providing $\mathcal{L}(c_A, \boldsymbol{\omega})$ is also indicated in Fig. 3(b).

The second reconstruction step is performed in the space of light rays given by direction $\boldsymbol{\omega}$ and passing through the camera center positions. Under the assumption of Constant Light Flux (CLF), the light intensity remains constant on the trajectory of any light ray. Following the CLF assumption, the light ray intensity for a given direction $\boldsymbol{\omega}$ only varies in its respective orthographic plane. The orthographic plane is a plane normal to $\boldsymbol{\omega}$. Such plane is indicated as the "$\boldsymbol{\omega}$-plane" in Fig. 3(c), and represented as a gray-shaded circle (the boundary of the circle is drawn for clarity purposes). The light ray in direction $\boldsymbol{\omega}$ recorded by each contributing camera intersects the $\boldsymbol{\omega}$-plane in points that are the projections of the cameras focal points on this plane. The projected focal points of the contributing cameras in $\boldsymbol{\omega}$ direction onto the $\boldsymbol{\omega}$-plane are highlighted by hollow

points in Fig. 3(c). Each projected camera point P_{c_i} on the planar surface is assigned the intensity value $\mathcal{L}(c_i, \boldsymbol{\omega})$, that is calculated in the first step.
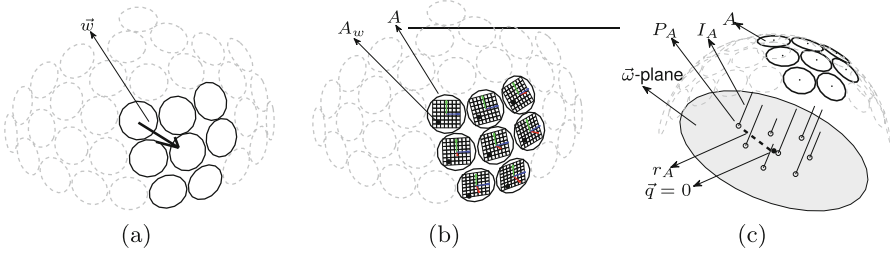

Fig. 3. (a) Cameras contributing to the direction $\boldsymbol{\omega}$ direction, (b) contributing pixel positions on the image frame of the contributing cameras for direction $\boldsymbol{\omega}$.

As an example, the projected focal point of camera A onto the $\boldsymbol{\omega}$-plane (*i.e.*, P_A) in Fig. 3(c) is assigned the intensity value I_A. The virtual observer point inside the hemisphere (*i.e.*, $\boldsymbol{q}$) is also projected onto the $\boldsymbol{\omega}$-plane. The light intensity value at the projected observer point (*i.e.*, $\mathcal{L}(\boldsymbol{q}, \boldsymbol{\omega})$) is estimated by one of the blending algorithms, taking into account all $\mathcal{L}(\boldsymbol{q}, \boldsymbol{\omega})$ values or only a subset of them. In the given example, each of the eight contributing camera positions shown with bold perimeter in Fig. 3(c) provides an intensity value which is observed into direction $\boldsymbol{\omega}$ for observer position $\boldsymbol{q} = 0$. The observer is located in the center of the sphere and indicated by a bold dot. A single intensity value is resolved among the contributing intensities through a blending procedure on its respective $\boldsymbol{\omega}$-plane.

When applying the nearest neighbour (NN) technique in the second reconstruction step, the light intensity at the virtual observer point for each $\boldsymbol{\omega}$ direction is set to the light intensity value of the best observing camera for that direction. The nearest neighbour technique is expressed in (1) in mathematical terms:

$$j = \operatorname{argmin}_{i \in I}(r_i)$$
$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{\omega}) = \mathcal{L}(c_j, \boldsymbol{\omega}) \tag{1}$$

where $I = \{i | \boldsymbol{\omega} \cdot \boldsymbol{t_i} \geq \cos(\frac{\alpha_i}{2})\}$ is the index of the subset of contributing cameras for the pixel direction $\boldsymbol{\omega}$. A pixel direction $\boldsymbol{\omega}$ is assumed observable by the camera c_i if the angle between its focal vector $\boldsymbol{t_i}$ and the pixel direction $\boldsymbol{\omega}$ is smaller than half of the minimum angle of view α_i of camera c_i. The length r_i identifies the distance between the projected focal point of camera c_i and the projected virtual observer point on the $\boldsymbol{\omega}$-plane. The camera with the smallest r distance to the virtual observer projected point on the $\boldsymbol{\omega}$-plane is considered the best observing camera. As an illustration, such distance is identified with r_A and depicted by a dashed line for the contributing camera A in Fig. 3(c).

Different brightness levels between cameras and misalignment causes sharp transition among the cameras. In order to resolve this transition problems,

several other blending techniques were proposed. For example, the linear blending scheme incorporates all the cameras contributing into a selected $\boldsymbol{\omega}$ direction through a linear combination [24]. This is conducted by aggregating the weighted intensities of the contributing cameras. The weight of a contributing camera is the reciprocal of the distance between its projected focal point and the projected virtual observer point on the $\boldsymbol{\omega}$-plane, *i.e.*, r_A in Fig. 3(c). The weights are also normalized to the sum of the inverse of all the contributing cameras distances.

The linear blending is expressed in (2) in mathematical terms.

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{\omega}) = \frac{\sum\limits_{i \in I} w_i \cdot \mathcal{L}(c_i, \boldsymbol{\omega})}{\sum\limits_{i \in I} w_i}$$

$$w_i = \frac{1}{r_i}$$
(2)

Apart from nearest neighbour interpolation and linear interpolation techniques, other novel algorithms such as Gaussion Blending [25], Restricted Gaussian Blending [26] and a probabilistic approach for omnidirectional image reconstruction is presented in [27].

For detailed discussion camera arrangement on spherical surface, different blending approaches, camera orientation and multiple camera calibration, the reader is referred to [21].

3   Distributed and Parallel Implementation

The system presented in [28] is implemented using a centralized approach where a single unit is responsible for data acquisition and data processing from multiple image sensors. The real-time implementation of multi-camera systems applications with a high number of cameras, high image sensor resolutions and the current image sensor architectures demands a high amount of hardware resources, and depending on the target application it might also demand high computing performances. This can create bottlenecks in such multi-sensor systems and limits the scalability. The number of cameras that can be connected to a single node is limited by the I/O constraints. For instance, interfacing 49 standard CMOS imagers with a single unit will not be feasible in terms of pin count. Furthermore, for high number of cameras and high camera resolutions, the memory bandwidth requirement increases significantly where a single unit will not be able to overcome the total bandwidth demand. Parallel processing approaches aim to overcome these limitations by distributing signal processing tasks and memory bandwidth usage among several signal processing blocks. This technique creates possibility of constructing higher resolution images beyond centralized approach. Moreover, parallel approaches are faster implementations compared to centralized approaches, which creates the possibility of creating higher resolution images beyond centralized approach. Due to the constraints posed by technology, the distributed and parallel approach can be a feasible solution for the real-time realization of such systems.

If tasks are distributed properly among many processors, the computation time will decrease significantly. In order to distribute the tasks among the nodes, it is required to enhance the features of customary cameras to include processing and communication capabilities. The processing capability enables the camera module to perform local processing down to pixel level, while communicating features permit information exchange among the camera modules. In contrast to previous centralized approaches pertaining to omnidirectional light-field reconstruction algorithms, a novel distributed and parallel algorithm for image reconstruction is implemented. Assuming that all cameras have signal processing capability and communication media that permits a communication with other cameras and a central unit, omnidirectional vision reconstruction algorithms can be realised in a distributed manner among camera nodes.

In the distributed and parallel implementation of the Panoptic Camera, each camera constructs a portion of the omnidirectional vision with the help of neighbouring cameras. For a distributed implementation of the omnidirectional algorithm, each ith camera must possess the knowledge of its covering directions and the information of the other contributing cameras for all of these directions. This information can be extracted by the internal and external calibration processes of the Panoptic system. After extracting the camera parameters, such as camera direction vectors and coordinates on spherical surface, angle of views (AOV) of each camera, etc., each camera can construct its responsible portion of omnidirectional view independently.

For instance, in the nearest neighbour technique, the best viewing camera for each ω is selected. Hence in this technique, each camera constructs a unique set of observation directions. The set of observation directions of each camera has no intersection with the other cameras of the Panoptic system in the nearest neighbour method. Therefore, camera modules can be limited to observe solely their own set of directions and construct their portions of omnidirectional vision, independently from each other.

In the linear interpolation technique, similar to the nearest neighbour technique, each camera can still be assigned to the task of vision reconstruction for its particular partition. For this purpose, each camera would need the information about which other cameras contribute to the particular ω and the intensity values obtained by the contributing cameras. For a constant set of ω directions, these parameters are only required to be calculated once and are stored in a local memory for real-time access. The distributed implementation of the algorithm is summarized in Algorithm 1. The required information can be calculated once by the central unit and updated to the local memory of the camera modules. Alternatively, each camera module can calculate its own required information using its own processing features.

In the initialization process, the set of best observing directions for each camera is extracted. Furthermore, other contributing cameras for each coverage direction and their weights used in the second interpolation step are extracted. After the initialization process, each camera has the knowledge of which ω to construct, which other cameras are contributing to the same ω and, depending

Algorithm 1. Distributed Reconstruction Algorithm for Camera Nodes

1: calculate calibration data
2: calculate weights
3: **for all** best observing directions **do**
4:     $P_m := read\_pixel\_from\_memory$
5:     $p_{contr,2..n} := request\_pixels\_from\_contributing\_cameras$
6:     $C := W_m \cdot P_m + \sum_2^n P_{s,n}$
7:     send C to central unit
8: **end for**
9: **for all** other observing directions **do**
10:     wait for request from principal camera
11:     $P_s := read\_pixel\_from\_memory$
12:     $P_{s,out} := W_s \cdot P_s$
13:     send $P_{s,out}$ to principal camera
14: **end for**

on the interpolation type, what are the camera weights contributing to the final level of interpolation. Assuming cameras have processing capabilities, the missing variables to construct the light field are the light intensity values obtained by the other cameras. This creates the necessity of a communication scheme among the camera modules.

The distributed and parallel implementation of omnidirectional reconstruction algorithm is explained in detail in Algorithm 1. Firstly, the initialization phase is conducted. For each camera, all observing directions ($\boldsymbol{\omega}$) and weights for the chosen interpolation technique are extracted. Then, for each new frame, camera creates its responsible portion of the final omnidirectional image. For all best observing directions, cameras read from the memory the corresponding pixel light intensity value (P_m) and weight (W_m). In the meantime, the camera module requests contributing light intensity values from the other cameras which observes the same direction. Each camera sends the light intensity value multiplied by the weight. After obtaining all values, the camera sends the sum of all intensity values to the central unit for display.

For directions other than the best observing ones, cameras still possess the weight and light intensity values. When a new light intensity request comes from the best observing camera, the camera reads the light intensity value (P_s) and weight (W_s). Afterwards it reconstructs the light intensity value $P_{s,out}$ for given direction and sends the value to the best observing camera

3.1   Processing Demands

The proposed architecture in [28] performs the omnidirectional vision reconstruction in a pipeline flow for both the nearest neighbour and the linear interpolation techniques. Assuming that the memory used in the system can sustain consecutive access cycles, F_{clk} for the presented real-time omnidirectional vision reconstruction architecture is derived from (3) as follows:

$$N_{acs} \times F_{ps} + T_{lat} \leq F_{clk} \tag{3}$$

For approximations, the latency term T_{lat} in (3) can be neglected. The maximum number of access time is $N_{acs} = N_{cam} \times N_\theta \times N_\phi$ which occurs in case when the cameras contribute in all directions for the linear interpolation technique. For high number of cameras and high camera resolutions, the F_{clk} is the dominant demand. The aggregate of the two demands is translatable to the memory bandwidth requirement of the system using the multiplying factor for number of pixels (N_{pix}). As the output resolution increases, memory bandwidth and clock frequency increases accordingly. Processing demand becomes major bottleneck, and distributing the algorithm becomes inevitable.

3.2   Effects of Pixelation Schemes

The pixel gridding scheme for the omnivision application has an effect on the load imposed on each camera module of the Panoptic system when implemented distributively.

The pixel directions $\boldsymbol{\omega}$ shown in Fig. 2(a) derive from an equi angular segmentation of longitude and latitude coordinates of a unit sphere into N_ϕ and N_θ segments, respectively. This pixelation enables the rectangular presentation of the reconstructed image suitable for ordinary displays, but results in a non-equal contribution of the Panoptic cameras. The density of the pixel directions close to the poles of the sphere is higher compared to the equator of the sphere in the equi angular pixelation scheme. Hence, the cameras positioned closer to the poles of the sphere contribute to more pixels in comparison to the other cameras of the system. The equi angular pixelation derives mathematically from (4):

$$\begin{aligned}
\phi_\omega(i) &= \frac{2\pi}{N_\phi} \times i, \quad 0 \leq i < N_\phi \\
\theta_\omega(j) &= \frac{\pi}{2N_\theta} \times (j + \frac{1}{2}), \quad 0 \leq j < N_\theta
\end{aligned} \tag{4}$$

The equi angular pixel gridding scheme depicted in Fig. 2(a) does not yield an equal number of $\boldsymbol{\omega}$ pixel direct camera to construct. For the nearest neighbour interpolation of the distributed and parallel approach, computational load is not equally distributed among the camera modules. For example, the camera which is placed in the north pole of the system, is responsible for more than 10 % of $\boldsymbol{\omega}$ pixel directions. The workload among the cameras are not distributed evenly, which is not suitable for implementation of the omnidirectional vision reconstruction algorithm in parallel.

An equal density pixelation scheme depicted in Fig. 2(b) resulting in an approximately even contribution of the cameras is devised for the Panoptic system. The scheme is based on enforcing a constant number of pixels per area, as expressed in (5) and (6). Compared to the equi angular pixelation, the change is observed in latitude angles.

$$\phi(i) = \frac{2\pi}{N_\phi} \times i, \qquad\qquad 0 \le i < N_\phi \qquad\qquad (5)$$

$$\theta(j) = \arccos(1 - \frac{j}{N_\theta}), \qquad\qquad 0 \le j \le N_\theta \qquad\qquad (6)$$

The equi angular pixelation leads to an evenly distributed workload among the camera modules, that makes distributing and parallelizing the algorithm more feasible. Detailed discussion on how the different pixelation scheme effects the distributed algorithm and corresponding memory demand for increased resolution and number of the cameras can be seen in [29].

4   Interconnected Network of Cameras

An interconnection network is a programmable system capable of transporting data between terminals. The system illustrated in Fig. 4 shows N terminals, $C_1 \ldots C_N$ connected to a network. For example, when terminal C_2 wishes to exchange data with terminal C_5, C_2 sends a message containing the data to the network and the network delivers the message to C_5. The terminals C_i resemble the camera nodes with processing and networking features in addition to basic imaging.
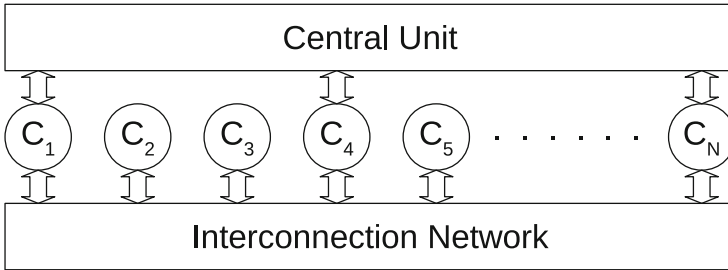

Fig. 4. High level model of an interconnected network of cameras. All cameras C_i are connected via interconnection network and some cameras have direct access to central unit.

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 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 may not have enough ports to interface with all the cameras of the system. In a 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 cameras in Fig. 4.

In multi-camera applications, information exchange mostly takes part among the neighbouring cameras. Thus, during the creation of an interconnection network, neighbourhood relations of camera modules should be preserved as much as possible. The neighbourhood relation for the cameras in Panoptic System can be seen in Fig. 5(b). It is an irregular graph-based topology. However, in most of the systems, this irregular graph-based topology is hard to implement and control on hardware level. A regular graph-based topology can be used to simplify the implementation of the interconnection network. Instead of creating an irregular graph-based network shown in Fig. 5(b), a regular graph-based, 7×7 mesh topology is chosen in order to realise interconnected network of cameras.

A regular network topology is relatively simple to implement and control. It is scalable and easy to extend, add or subtract nodes. Flow control mechanisms and packet structures are easier to construct at the hardware level. Furthermore, it generalizes the problem regardless of the source network topology and the camera arrangement in the physical hemisphere dome. However, mapping cameras into the network nodes creates new problems.

4.1 Camera Assignment Problem

In order to obtain the neighbourhood relation graph of the Panoptic system, the surface of the Panoptic device hemisphere is partitioned into a set of cells centered on the camera locations. Each cell is defined as the set of all points on the hemisphere which are closer to the camera location contained in the cell than to any other camera positions. The boundaries of the cells are determined by the points equidistant to two nearest sites, and the cell corners (or nodes) to at least three nearest sites. This particular partitioning falls into the category of a well-established geometry concept known as the Voronoi diagram (or Voronoi tessellation [30]). The Voronoi diagram of a 5 floors and 49 cameras Panoptic system can be seen in Fig. 5(a). The geometrical neighbourhood relation of 5 floors and 49 cameras extracted from the Voronoi diagram is shown in Fig. 5(b).

This assignment strategy is known in the context of a facility allocation problem called the Quadratic Assignment Problem (QAP). The QAP models the following real-life problem: In a graph-based topology, for each pair of locations a distance is specified and for each pair of facilities a weight or flow (e.g. the amount of supplies transported between two facilities) 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. A planar graph representing the neighbouring of the cameras is extracted in Fig. 5(b) where the nodes of the extracted graph represent the cameras and its edges resembles the neighbouring of the cameras. Hence, in the latter graph two nodes are connected
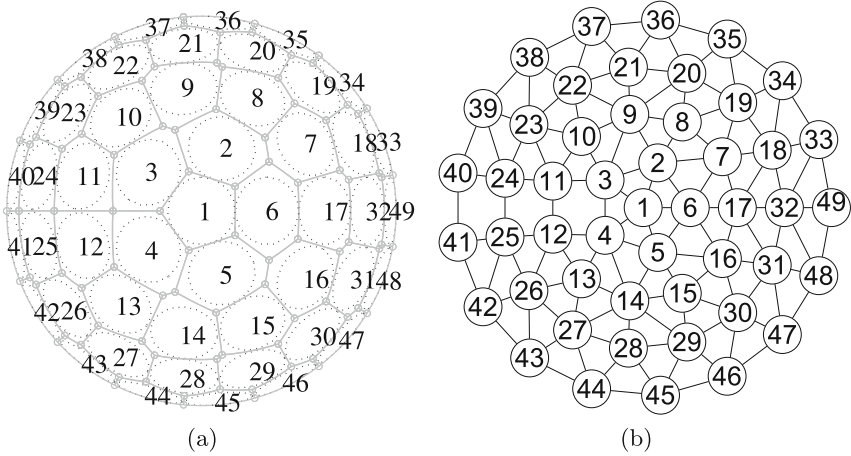
Fig. 5. (a) Top view of the Voronoi diagram of a five-floor Panoptic system containing 49 camera locations (b) The planar graph extracted from the Voronoi diagram.

if their respective cameras are geometrical neighbours. The adjacency matrix of this graph can be used as the flow matrix of the QAP.

The QAP is an NP-hard problem; which means there is no known algorithm for solving this problem in polynomial time, and even small instances may require long computation time. Among different proposed solutions, sparse version of the GRASP algorithm [31] has given the best result solving the QAP. The assigned camera numbers of Fig. 5(b) is represented on the mesh graph shown in Fig. 6(a). The assignment allocates the cameras such that all geometrical neighbouring cameras are not more than three hops away from each other in the new 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 larger than the number of cameras of the Panoptic system, by 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 Central Unit Access

As stated previously, having a distributed camera system does not imply the omission of a central unit. However, we have explained that connecting all units to the central unit is also problematic. We need to find candidate cameras that will have direct access to central unit. In order to decide which cameras will have direct access, the problem to solve is which p candidate cameras to select to have 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 reducing access time between the central unit and any camera of the interconnected
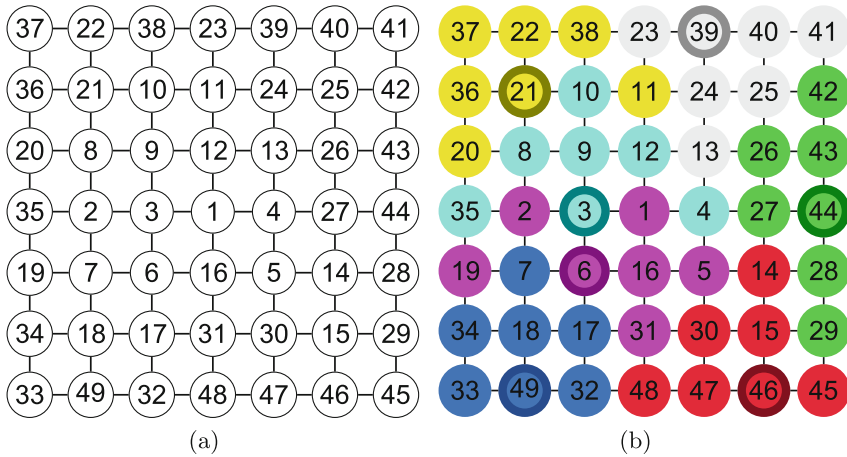
Fig. 6. (a) The assigned 7×7 mesh topology interconnected network (b) The 7×7 Mesh topology with 7 vertex p-centers.

network, assuming sufficient channel bandwidth is available. The latter problem can also be mapped into a 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 [32]. In order to distribute forty-nine cameras' load equally, the p value is chosen as seven. As an example, a vertex 7-center problem has been solved for the mesh graph topology depicted in Fig. 6(b) assuming that each camera with access to central unit can support up to 7 clients. The problem is solved using an exact algorithm for the capacitated vertex p-center problem [33]. The solution is depicted in Fig. 6(b). All the cameras acting as p-center (*i.e.*, with access to central unit) are shown with a bold edge. The cameras belonging to the same p-center are also filled with similar colors. All cameras are at most two hops away from their supporting facility camera. This strategy aims to minimize the network load caused by the transmission of central unit access packets.

4.3   Verification

The designed interconnection network is simulated under real or close-to-real conditions. The "BookSim" simulator [34] is used for the purpose of performance analysis of the interconnection network of cameras. The BookSim simulator is a C++ based cycle-accurate interconnection network simulator. The simulator is extended to support custom-defined traffic patterns which are configured by a custom text file. This development was accomplished to support any traffic pattern for target networks under test. A MATLAB-based routine is developed in order to simulate different injection rates with several different test patterns. Optimal parameters for router unit such as number of virtual channels, buffer
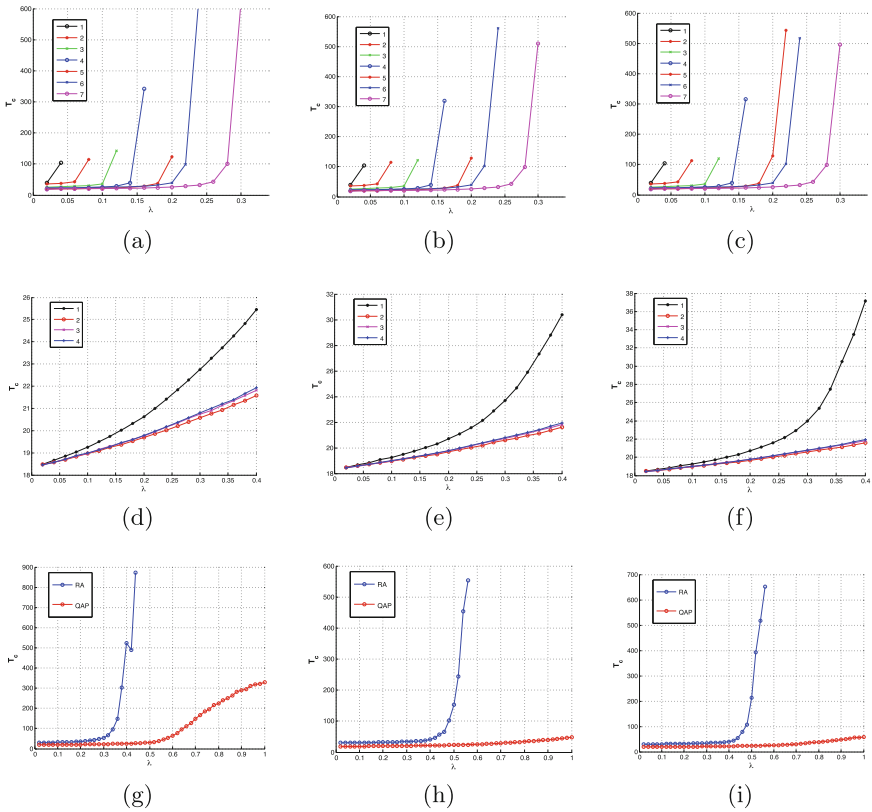
Fig. 7. Average packet latency (T_c) vs average throughput (λ) (i.e., packet injection rate) graphs. The (a), (b), (c) and graphs demonstrate the latency vs throughput for routers with flit buffer size equal to 8, 32 and 64, respectively. The (d), (e) and (f) graphs demonstrate the latency vs throughput for routers for a 7×7 mesh network with QAP assigned camera locations comparing number of virtual channels, flit buffer sizes equal to 8, 32 and 64, respectively. The (g), (h) and (i) graphs demonstrate the comparison in between QAP camera assignment versus random camera assignments, flit buffer sizes equal to 8, 32 and 64, respectively. Several different random assignments have been conducted and the average latency values obtained through BookSim simulations

size etc. are extracted in terms of latency (T_c) versus throughput (λ) with custom created Panoptic traffic pattern. Injection rate is indicating how frequently a new packet is injected into network while latency indicates how many clock cycles it takes for a network packet to traverse to the destination node. All the injection rates are normalized to channel bandwidth and latency is expressed in number of cycles.

The graphs in Fig. 7(a), (b) and (c) depicts the latency vs. injection rate for different number of vertex p-centers selected for direct access to the central unit. It is observed that for the nearest neighbour technique traffic pattern, the

demands on the interconnection network tend to reduce as the number of vertex p-centers grows. As the number of vertex p-centers grows, the traffic becomes more balanced and localized.

The 7×7 mesh network is also simulated under linear interpolation traffic pattern. The number of vertex-p centers is chosen as seven. The assignment provided by the QAP approach and shown in Fig. 6(b) is used. The graphs in Fig. 7(d), (e) and (f) demonstrate the latency versus throughput for routers with flit buffer size equal to 8, 32 and 64, respectively. The results are given for throughput values of $\lambda < 0.4$, as it is expected that the injection rate will not be higher than 0.4.

For the purpose of comparison, a set of average packet latency versus average throughput graphs under linear interpolation traffic pattern for a 7×7 mesh network with random and QAP assigned camera locations. Figure 7(g), (h), and (i), demonstrate the latency versus throughput for routers with one virtual channel and flit buffer size equal to 8, 32 and 64, respectively.

Simulations prove that Panoptic traffic pattern can be implemented with expected injection rate and latency. Extracted parameters utilized during the implementation of the router mechanism in an FPGA platform. For the FPGA implementation, an open source Network-on-Chip Router in RTL provided by [34] is utilized.

5   Panoptic Media Platform

A custom-made FPGA platform is designed for the practice of the concept of an interconnected network of cameras. The developed platform is referred to as the Panoptic Media. A Panoptic system comprising 49 cameras is interfaced to this platform. The design and implementation of the parallel and distributed approach of the omnidirectional vision reconstruction algorithm of the Panoptic camera is elaborated for the Panoptic Media platform. The Panoptic Media Board (PMB) is an FPGA-based development board. The PMB includes eight Xilinx XC5VLX110 Virtex5 FPGAs. One FPGA is targeted for the implementation of the central unit and the other seven are slaves and used for emulating an interconnected network of cameras. The FPGA hosting the central unit is referred to as the central/master FPGA and FPGAs hosting cameras is referred to as the slave FPGAs. The top view of the designed platform is shown in Fig. 1(b).

5.1   Central FPGA

The central FPGA hosts the central unit of the system. It is designed to be in charge of initialization, synchronization among the FPGAs and camera nodes, camera router nodes configuration and control, display and external host communications of the system. For external communications, the central unit has access to a USB-2.0 device and $1\,\mathrm{Gb/s}$ Ethernet physical controller device.

At system power up the central unit enters an initialization phase. In this phase, the external physical channel ports of the central FPGA which are connected to that of the slave FPGAs are synchronized. This synchronization is conducted on all FPGAs to achieve a fully synchronous interconnected network. The synchronization is a phase alignment process in which the data bus connections are adjusted at the receiver side for optimum clock sampling. The phase alignment is adjusted using the dynamic time delay adjustment feature of the Virtex-5 FPGA IO buffers. For this purpose, a synchronization pattern is first transmitted on all transmitting bus connections (*i.e.*, outward bus connections) while the receiver bus connections IO buffer time delays are adjusted for optimum clock sampling by their host MicroBlaze processor, on all FPGAs.

The central unit can communicate with all camera router nodes of the interconnection network through packet transmission and reception. Two types of packet exist in the system, named control and data packets. Control packets are used for configuring camera router modules or monitoring and status check purposes. The central FPGA's MicroBlaze processor can access all the register banks of the SmartCam IPs via the interconnection network using packet based messages. The data packets contain image information data which are used for display or for transfer to an external host. Each packet type and subtype is identified using a specific packet ID.

Each data packet contains a pixel information of an image frame. The data packets can be sent by all the cameras simultaneously. Therefore the pixels of an image are receivable in a shuffled order by the central unit. Hence all the data packets pertaining to an image frame are temporarily stored by the RCTRL IP in the ZBT-SRAM first. The shuffled order of the receiving data packets implies a random write access nature to a memory. To this aim, the ZBT-SRAM is chosen for the temporary storage of the data packets pixel information part. When a full frame is received the RCTRL IP transfers the received frame to the SDRAM. The SDRAM is used as the video memory for external display interfaces like monitors or projectors.

The Central unit has access to a USB-2.0 device through a Xilinx external peripheral controller (EPC) IP. The Central unit identifies the USB-2.0 device as an asynchronous FIFO memory. The EPC IP is configured for correct access times with the USB-2.0 device. The USB-2.0 device is used as the primary path for external host communication. The ordered image data in the ZBT SRAM can be transferred to an external host.

5.2 Slave FPGAs

The role of a slave FPGA is to emulate a portion of a 7×7 mesh interconnected network of cameras. Each slave FPGA is responsible for seven imagers, hosts seven camera modules and seven ASRAM memories, application control unit (ACU) and channel synchronisation (CHSYNC) modules for inter fpga communication and synchronization. The SoC architecture for Slave unit can be seen in Fig. 8. Each imager is interfaced to a custom-designed smart camera IP (Smart-Cam). The SmartCam IP is a camera module with router connectivity, memory
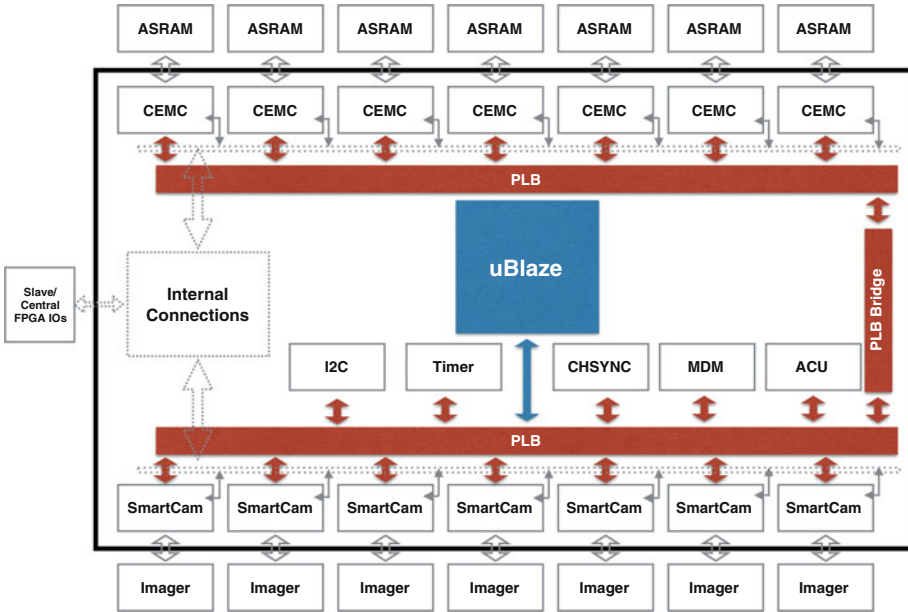
Fig. 8. SoC architecture of the slave FPGA.

and application processing units. The internal blocks of the custom-designed SmartCam IP is shown in Fig. 9. Each SmartCam IP interfaces with a custom external memory controller (CEMC). SmartCam IPs are provided access to an ASRAM via its interfacing CEMC IP.

The SmartCam IP comprises five sub-blocks. The applications intended for the SmartCam IP are implemented into the Image Processing Unit sub-block.The Image Processing (IP) sub-block is designed to perform image processing applications. There are three modes of operation, named as video stream, nearest neighbour and linear interpolation. In the video stream, the SmartCam IP transfers the video stream generated by the camera to the central unit for visual display or external host transfer. This mode is necessary for calibration purposes. In the nearest neighbour and linear interpolation modes, it is responsible for creating network demand packets for pixel values obtained by the other contributing cameras and performing the first and second steps of interpolations of the reconstruction algorithm. Each SmartCam IP provides its portion of omnivision of the panoptic system to the central unit.

The Imager Interface sub-block is responsible for image acquisition and transfers the video stream generated by the imager to the ASRAM memory. The IP sub-block communicates with the central unit and other SmartCam IPs in the panoptic system through the Router sub-block. The Router sub-block comprises five-ports (*i.e.*, north, south, east, west and an input/output port in order to
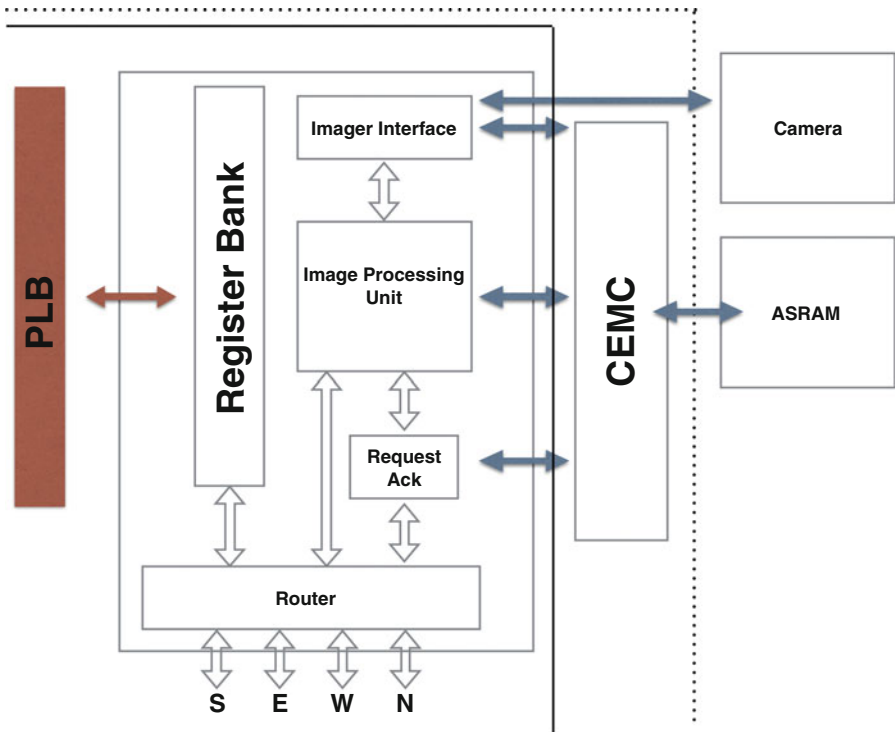
Fig. 9. Internal blocks of the SmartCam IP used in the slave FPGAs

enter or flush out of the network ports). The router sub-blocks' main aim is to create the communication medium among the SmartCams.

The Request Acknowledge sub-block responds to the incoming demand packets from other SmartCam IPs. It creates respond packets that contains the necessary intensity values and coefficients that are used in the second step of interpolation. Register Bank sub-block is for the IP's mode configuration, monitoring and status checks. It can be reached by the central unit via the interconnection network to perform overall control of the system.

Forty-nine SmartCams distributed over seven FPGAs are operate in parallel for omnidirectional vision reconstruction. Throughout the interconnection network, pixel intensity values are interchanged among the modules and each camera constructs its assigned portion of the omnidirectional vision. The central unit is responsible for obtaining all reconstructed pixels and displaying them.

5.3   Inter FPGA Communication

Each FPGA has twelve sets of 24-bit bus connections. Each two set of 24-bit bus connections are bundled to form a physical channel port for an FPGA. Each FPGA contains six physical channel ports. The direction of one bus connec-

Table 1. FPGA Device Utilization for Central FPGA and an example of the Slave FPGAs, for both nearest neighbour technique and linear interpolation techniques

Resources	Central FPGA		Slave NN		Slave LIN		Avail.
	Used	Util.	Used	Util.	Used	Util.	
Slice LUTs	19495	28 %	54360	78 %	61416	88 %	69120
Slice registers	20617	29 %	32326	47 %	40038	57 %	69120
BlockRAM/FIFO	93	72 %	89	69 %	89	69 %	128
DSP48Es	3	4 %	26	40 %	61	95 %	64

tion is chosen as outward while the other one is selected as inward. However, the physical channel ports of the FPGAs can contain multiple logical channels. For the presented partitioning scheme of a 7×7 mesh interconnected network among the slave FPGAs of the PMB, it is sufficient to have a maximum of four logical channels within a physical channel port. Logical channels are realizable through time multiplexing while operating at higher frequency rates within a single physical channel. Four logical channels are realized by doubling the slave FPGA clock frequency and sending the packets in dual data rate (DDR) mode.

5.4 Implementation Results

A Panoptic multi-camera sphere of diameter $2r_\odot = 30$ cm is built by stacking circular PCB rings on top of each other as shown in Fig. 1(a). VGA camera modules are operated at 25 fps. The camera module requires to be programmed for activation through a two-wire I2C serial interface. At start-up, the central FPGA resets all the slave FPGAs and hence the interconnection network. At first, a calibration phase of the physical interconnect channels that exist among the FPGAs of the system is conducted. After the synchronization among the FPGAs is accomplished, the slave FPGAs' MicroBlaze processors start initializing the ASRAM memories. The central FPGA is designed as a control unit and its MicroBlaze processor can access all the register banks of the camera IPs of any slave FPGA via the interconnection network using packet-based messages. The system was found to support the real-time operation of a 7×7 interconnected network of VGA cameras and providing an omnidirectional vision with a 25 frame per second rate an XGA (1024×768 pixel) resolution with the linear interpolation method. The resource utilization percentage of the central FPGA and one of the slave FPGAs can be seen in Table 1.

6 Visualization of Omnidirectional Data

The Panoptic Camera can be used as a perfect example of a telepresence system. Unlike the virtual reality systems, where users are transported to a virtual scene, telepresence allows users to be in another location in real world. Videoconferencing is one example of telepresence. Among the benefits of videoconferencing, we

can say it lowers the travel requirements, improves dialog efficiency and allows mobility impaired people to visit distant places. Instead of using narrow angle field of view cameras, we can achieve a better telepresence with the Panoptic Camera.

Early omnidirectional imagers were mainly using extreme fish eye lenses or hyperboloidal mirrors, such as described in [35]. These imaging systems are limited by the resolution capabilities of a single sensor and feature strong distortions. The resolution can be increased by using more modern image sensors, as described in [36], however the distortions remain. Additionally a big portion of the image is covered by the reflection of the camera lens. In [37] a multi-camera approach is proposed for omnidirectional video generation for telepresence. However this particular solution cannot be used for real-time video streaming, because the video generation is achieved in post processing.

In this section, we will present a novel telepresence system, which allows users to naturally observe the remote location. Omnidirectional data will be created with the Panoptic Media Platform, and remapping of omnidirectional data through observed direction will be created by using the wide field of view head mounted display (HMD) Oculus Rift.

As explained in Sect. 5.4, Panoptic camera has two different operation modes. For the telepresence system, full XGA resolution (1024×768) of the [29] will be used. The system can be divided into two parts, one is the server application and the second one is the client application.

6.1   Server Application

The omnidirectional XGA output generated by the Panoptic Camera is transmitted via the DVI output. A capture card connected to the server PC is utilized to transfer omnidirectional data into the server PC. The main task of the server PC is to distribute the whole omnidirectional image via TCP to clients. The application automatically adapts to input resolution changes and can therefore also be used with other camera systems and future versions of the Panoptic camera. The server application is able to stream video to multiple clients at the same time via TCP.

6.2   Client Application

The client application receives the TCP stream originating from the server application and generates the views for the head-mounted display. Optionally it can also directly receive the images from the DVI capture card, when the camera system is close to the user.

In order to display the hemispherical image on the client side of the telepresence system, a virtual environment is created. This virtual environment is created using the OpenGL API and consists of a user controlled camera and a large overhead hemisphere, onto which the image is mapped. The camera rotates according to the sensor data received from the head-mounted display. The omnidirectional image is used as a texture for the virtual hemisphere. To retrieve the

correct dimensions of the captured objects, the equal density mapping scheme expressed in (5) and (6) needs to be reversed.

Using the inverse mapping functions (7) and (8) the original angular directions are restored. In these equations, $\frac{i}{N_\phi}$ and $\frac{j}{N_\theta}$ correspond to the OpenGL texture coordinates s and t respectively.

$$s(\phi) = \frac{i(\phi)}{N_\phi} = \frac{\phi}{2\pi} \qquad\qquad 0 \leq \phi < 2\pi \qquad\qquad (7)$$

$$t(\theta) = \frac{j(\theta)}{N_\theta} = 1 - \cos(\theta) \qquad\qquad 0 \leq \theta \leq \frac{\pi}{2} \qquad\qquad (8)$$

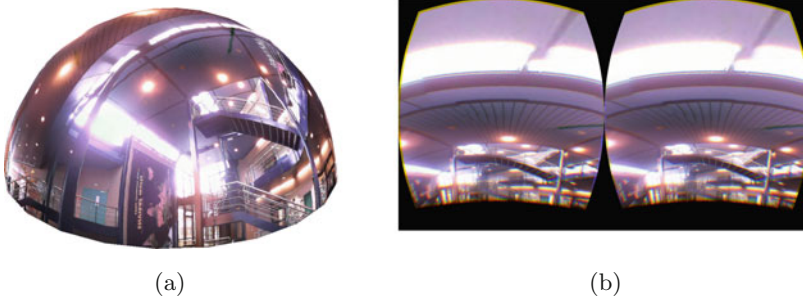(a)   (b)

Fig. 10. (a) The textured OpenGL hemisphere showing a captured image, viewed from the side. (b) The client application generating the left and right eye view for the head-mounted display.

Figure 10(a) shows the textured virtual hemisphere from the side. When using the application with the head mounted display, the user Viewpoint is in the middle of the sphere. Figure 10(b) shows the application in normal use with the HMD.

In order to ensure a high frame rate at all times, the application receives new omnidirectional images in a secondary thread. Thanks to the multi-threaded implementation the rendering frame rate is independent from the USB or network connection speed, as well as the camera frame rate. This is important for the network streaming functionality, in which the frame rate can vary.

6.3   Future Work

Panoptic Camera can broadcast the omnidirectional image and clients with head mounted display can observe the surroundings of the system remotely. Thanks to the natural behaviour of the system, each user can observe different direction at the same time. The Oculus Rift tracks the direction and angle of the users head, thus enables to render the observerd location from the obtained real-time

omnidirectional data. An example video can be seen in [38]. We believe the system can be used to broadcast the concerts or sports events and allow people to visit distant places. Currently, it is planned to develop a new Panoptic Camera system, which will increase the output resolution of the system as the resolutions of the head mounted displays increase. Furthermore, video compression and decompression will be implemented, in order to allow higher frame rates via internet.

7   A Real-Time HDR Panorama with Panoptic Camera

High dynamic range (HDR) images are usually obtained by capturing several images of the scene at different exposures. Previous HDR video techniques adopted the same principle by stacking HDR frames in time domain. We have modified Panoptic Camera Platform in order to construct and render HDR panoramic video in real-time, with 1024×256 resolution and a frame rate of 25 fps. We exploit the overlapping fields-of-view between the cameras in the bottom row of the Panoptic Camera with different exposures to create an HDR radiance map. We have proposed a method for HDR frame reconstruction which merges the previous HDR imaging techniques with the algorithms for panorama reconstruction. The developed FPGA-based processing system is able to reconstruct the HDR frame using the proposed method and tone map the resulting image using a hardware-adapted global operator. Detailed explanation of the implementation of the HDR Panoptic System can be seen in [39] and [40].

Dynamic range in the digitally acquired images is defined as the ratio between the brightest and the darkest pixel in the image. Most modern cameras cannot capture sufficiently wide dynamic range to truthfully represent radiance of the natural scenes, which may contain several orders of magnitude from light to dark regions. This results in underexposed or overexposed regions in the taken image and the lack of local contrast. The underexposed and overexposed images show fine details in very bright and very dark areas, respectively. These details cannot be observed in the moderately exposed image.

High dynamic range (HDR) imaging technique was introduced to increase dynamic range of the captured images. HDR imaging is used in many applications, such as remote sensing [41], biomedical imaging [42] and photography [43], thanks to the improved visibility and accurate detail representation in both dark and bright areas.

Besides capturing the natural scenes, another problem occurs when displaying them. The modern displays are limited to the low dynamic range, which causes inadequate representation of even standard LDR images. In order to avoid such problems, a tone mapping operation is introduced to map the real pixel values to the ones adapted to the display device. The purpose of tone mapping is to compress the full dynamic range in the HDR image, while preserving natural features of the scene.

In this section we present a new imaging system for HDR video construction and rendering. The key idea is to use a multi-camera Panoptic Camera

setup to create a composite frame, where cameras with the overlapping field-of-view (FOV) are set to different exposure times. Such system reduces the motion blur, as there is no inter-frame gap time (which can be several hundreds milliseconds in the standard HDR cameras). Additionally, the frames are captured at the same moment by all cameras, which reduces the intra-frame motion of the scene objects to the difference interval of cameras' exposure times. We developed a hardware prototype customized for real-time video processing, utilizing the multi-camera setup. It is a high performance field programmable gate array (FPGA) based system which provides capability for real-time HDR frame construction and tone mapping.

The pixel streams coming from the cameras are processed in real-time; hence, HDR video is created as a stack of HDR frames in time domain. Construction of each frame can be divided into two independent processes: (1) construction of HDR composite frame, and (2) tone mapping the composite frame to achieve realistic rendering.

7.1   HDR Composite Frame

Thanks to the circular arrangement of the cameras on this prototype, we adopted the similar approach as in [25], simplified to a two-dimensional case. The installed cameras were calibrated for their intrinsic and extrinsic parameters: focal length, frame center position, lens distortion and angular position in space (yaw, pitch, roll) with geometric center of the prototype as the origin point. To be able to reproduce the HDR image, the cameras are also color calibrated. The camera's response curve is recovered using a set of shots of the same scene with different exposure settings. The response curve is recovered by applying the algorithm proposed by Debevec and Malik [4]. Only one camera is color calibrated, as we assumed that the response curve is identical for all installed cameras. Both calibrations are done only once, as the parameters do not change over time.

FOVs of the cameras overlap such that each point in space is observed by at least two cameras. We exploit this property and set the camera exposures to different values. During the camera initialization phase, all cameras are set to the auto-exposure mode. The camera with the longest exposure time, *i.e.*, the one observing a dark region, is taken as a reference. In the following step, half of the cameras are set to the reference exposure t_{ref}, while other half is set to $t_{ref}/4$, such that two cameras with overlapping FOVs have different exposure times.

Even though after the calibration process, the registration errors and visible seams are unavoidable. Hence, an additional blending process is required. The Gaussian blending method proposed in [25] is based on a weighted average among the cameras contributing to the observed direction. The result of applying Gaussian blending on the acquired data provides the composite HDR radiance map, which should be tone mapped for realistic display (Table 2).

Table 2. FPGA Device Utilization for Central FPGA and an example of the Slave FPGAs, for High Dynamic Range Image Reconstruction

Resources	Central FPGA		Slave FPGAs		Avail.
	Used	Util.	Used	Util.	
Slice LUTs	18376	26 %	63732	92 %	69120
Slice registers	17498	25 %	40509	58 %	69120
BlockRAM/FIFO	88	69 %	89	69 %	128
DSP48Es	58	90 %	61	95 %	64

7.2 Tone Mapping

Yoshida et al. [44] made an extensive comparison of the tone mapping operators. The comparison was realized by human subjects grading several aspects of the constructed image, such as contrast, brightness, naturalness and detail reproduction. One of the best graded techniques in this review was the local operator by. Therefore, this operator will be taken as a base for the development of an FPGA-suitable operator. Even though this mapping is created for interactive applications, its speed is very slow for video applications. The reported frame rate is below 10 fps, for 720×480 pixels image, without any approximations which decrease the image quality [45].

Drago et al. [45] proposed changing logarithm base and calculating only natural and base-10 logarithms. In this approach, fast calculation of generic power functions, is not possible. Hence, we adopted the parameters to relax the hardware implementation, without losing any image quality. In [40], the new tone mapping operator suitable for hardware implementation is described in detail. The set of required mathematical operations is reduced to only addition, multiplication and division, which are suitable for fast implementation.

7.3 FPGA Implementation

Smart Camera IP in Fig. 9 is now responsible for calculation of the final HDR pixel value. Using the calibration data, the block reads the appropriate pixel from memory, multiplies it with the weight, and requests the weighted pixel from the secondary camera. The secondary pixel has already been multiplied by the HDR blending weight in the Secondary pixel block, thus only final addition is required. The resulting HDR pixel is further provided to the central unit.

In Omnidirectional implementation of the Panoptic Camera, the Central Unit was solely responsible for reordering the pixel packets coming from Slave FPGAs and displaying the output data from DVI display. In HDR reconstruction, the Central Unit also responsible for tone mapping algorithm. The tone mapping implementation consists of two parts: finding the maximum pixel luminance L_{max} and tone mapping curve implementation. Finding L_{max} consists of finding the maximum value in a sequence of the read luminances. L_{max} value is needed

for the core tone mapping operation. When HDR video stream is processed, L_{max} is taken from the previous frame, under the assumption that the scene illumination does not vary faster than response time of the human visual system. The parameter is updated at the end of each frame.

7.4   Discussion and Future Work

Our HDR construction method does not provide as significant increase in dynamic range as some of the other methods, due to the use of only 2 f-stops. However, up to our knowledge, it is the only system which uses multiple cameras to create and render HDR radiance map simultaneously, and provides real-time HDR video signal at the output. The next step is to additionally improve the dynamic range by increasing the number of cameras, and using more than two different exposures per reconstructed pixel. Furthermore, image quality can be improved by using a more complex blending algorithm, such as [46]. However, real-time implementation of such algorithm requires a more powerful hardware setup.

8   Conclusion and Future Work

A novel parallel and distributed technique for the omnidirectional vision reconstruction of the Panoptic camera in an interconnected network of cameras arrangement is presented. A methodology for camera's assignment into regular network node and selecting candidate cameras for central unit communication is shown. A custom-made FPGA-based platform termed the Panoptic Media Board (PMB) was implemented and introduced for the emulation of a 7×7 mesh interconnected network of smart cameras. The system-level design of the PMB was elaborated. The SoC architecture of the FPGAs of the PMB was presented. The PMB prototype provides a real-time 25 frame per second omnidirectional vision at XGA resolution. For a second operation mode in addition to the XGA output resolution, Panopticmedia also be found to support a 256×1024 output resolution with nearest neighbour interpolation method. During the display of 256×1024 resolution, a chosen camera in VGA resolution can also be displayed below the 360 omnidirectional output. Reader is referred to [38] for examples of such videos taken by the PanopticMedia Platform. Furthermore, we have presented a visualization technique for Omnidirectional Camera in order to make the created 360° view suitable for telepresence applications. Finally, by utilizing the overlapped angle of views of the cameras in the bottom row of the Panoptic System, we have created and rendered a real-time multiple camera HDR video. The distributed and parallel implementation of multi-band blending technique [46] is considered for the next real-time application deployment of the Panoptic device. Moreover, a new version of the Panoptic System is under development, for providing higher output resolutions than XGA. Future work and application areas of the Panoptic device are not limited to omnidirectional reconstruction. Depth-map estimation, super-resolution and multi-view imaging are open research topics.

References

1. Kubota, A., Smolic, A., Magnor, M., Tanimoto, M., Chen, T., Zhang, C.: Multiview imaging and 3DTV. IEEE Sig. Process. Mag. **24**(6), 10–21 (2007)
2. Szeliski, R.: Image mosaicing for tele-reality applications. In: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, pp. 44–53, Dec 1994
3. Mann, S., Picard, R.W.: On being undigital with digital cameras: extending dynamic range by combining differently exposed pictures. In: Proceedings of IS&T, pp. 442–448 (1995)
4. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, pp. 369–378 (1997)
5. Tanimoto, M., Tehrani, M., Fujii, T., Yendo, T.: Free-viewpoint TV. IEEE Sig. Process. Mag. **28**(1), 67–76 (2011)
6. Yagi, Y.: Omni directional sensing and its applications. IEICE Trans. Inform. Syst. **E82–D**(3), 568–579 (1999)
7. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New York, NY, USA, pp. 31–42 (1996)
8. Schechner, Y., Nayar, S.: Generalized mosaicing. In: IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 17–24 (2001)
9. Taylor, D.: Virtual camera movement: the way of the future? Am. Cinematographer **77**(8), 93–100 (1996)
10. Rander, P., Narayanan, P.J., Kanade, T.: Virtualized reality: constructing time-varying virtual worlds from real world events. In: IEEE Visualization 1997, pp. 277–284 (1997)
11. Zhang, C., Chen, T.: A self-reconfigurable camera array. In: Eurographics Symposium on Rendering, pp. 243–254 (2004)
12. Yang, J.C., Everett, M., Buehler, C., McMillan, L.: A real-time distributed light field camera. In: Proceedings of the 13th Eurographics Workshop on Rendering, Aire-la-Ville, Switzerland, pp. 77–86 (2002)
13. Wilburn, B., Joshi, N., Vaish, V., et al.: High performance imaging using large camera arrays. ACM Trans. Graph. **24**, 765–776 (2005)
14. Yuan, X., Qinghai, Z., Liwei, G., Mingcheng, Z., Xiaohong, D., Teng, R.: High speed simultaneous image distortion correction transformations for multi-camera cylindrical panorama real-time video system using FPGA. IEEE Trans. Circuits Syst. Video Technol. **24**(6), 1061–1069 (2013)
15. Ladybug, Point Grey Research Inc. http://ww2.ptgrey.com
16. Schreer, O., Feldmann, I., Weissig, C., Kauff, P., Schafer, R.: Ultrahigh-resolution panoramic imaging for format-agnostic video production. Proc. IEEE **101**(1), 99–114 (2013)
17. Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., Weaver, J.: Google street view: capturing the world at street level. Computer **43**(6), 32–38 (2010)
18. Cogal, O., Akin, A., Seyid, K., Popovic, V., Schmid, A., Ott, B., Wellig, P., Leblebici, Y.: A new omni-directional multi-camera system for high resolution surveillance. In: SPIE Sensing Technology Applications. International Society for Optics and Photonics (2014)
19. Brady, D., Gehm, M., Stack, R., Marks, D., Kittle, D., Golish, D., Vera, E., Feller, S.: Multiscale gigapixel photography. Nature **486**(7403), 386–389 (2012)

20. Song, Y.M., Xie, Y., Malyarchuk, V., Xiao, J., Jung, I., Choi, K.-J., Liu, Z., Park, H., Lu, C., Kim, R.-H., et al.: Digital cameras with designs inspired by the arthropod eye. Nature **497**(7447), 95–99 (2013)
21. Afshari, H., Jacques, L., Bagnato, L., et al.: The PANOPTIC camera: a plenoptic sensor with real-time omnidirectional capability. J. Sig. Process. Syst., 1–24 (2013)
22. Neumann, J., Fermuller, C., Aloimonos, Y.: Polydioptric camera design and 3d motion estimation. In: 2003 Proceedings, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II-294-301, June 2003
23. Levoy, M., Hanrahan, P.: Light field rendering. In SIGGRAPH 1996, Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 31–42. ACM (1996)
24. Afshari, H., Akin, A., Popovic, V., et al.: Real time FPGA implementation of linear blending vision reconstruction algorithm using a spherical light field camera. In: IEEE Workshop on Signal Processing Systems (2012)
25. Popovic, V., Afshari, H., Schmid, A., Leblebici, Y.: Real-time implementation of Gaussian image blending in a spherical light field camera. In: 2013 IEEE International Conference on Industrial Technology (ICIT), pp. 1173–1178, Feb 2013
26. Popovic, V., Seyid, K., Akin, A., Cogal, Ö., Afshari, H., Schmid, A., Leblebici, Y.: Image blending in a high frame rate FPGA-based multi-camera system. J. Sig. Process. Syst., pp. 1–16
27. Cogal, O., Popovic, V., Leblebici, Y.: Spherical panorama construction using multi sensor registration priors and its real-time hardware. In: 2013 IEEE International Symposium on Multimedia (ISM), pp. 171–178, Dec 2013
28. Afshari, H., Popovic, V., Tasci, T., Schmid, A., Leblebici, Y.: A spherical multi-camera system with real-time omnidirectional video acquisition capability. IEEE Trans. Consum. Electron. **58**(4), 1110–1118 (2012)
29. Seyid, K., Popovic, V., Cogal, O., Akin, A., Afshari, H., Schmid, A., Leblebici, Y.: A real-time multiaperture omnidirectional visual sensor based on an interconnected network of smart cameras. IEEE Trans. Circuits Syst. Video Technol. **25**(2), 314–324 (2015)
30. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications, 2nd edn. Springer, Heidelberg (2000)
31. Pardalos, L., Resende, M.: A greedy randomized adaptive search procedure for the quadratic assignment problem. In: Quadratic Assignment and Related Problems. DIMACS Series on Discrete Mathematics and Theoretical Computer Science, vol. 16, pp. 237–261 (1994)
32. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems. I: the p-centers. SIAM J. Appl. Math. **37**, 513–538 (1979)
33. Pinar, M.C., Ozsoy, F.A.: An exact algorithm for the capacitated vertex p-center problem. Comput. Oper. Res. **33**(5), 1420–1436 (2006). http://dx.doi.org/10.1016/j.cor.2004.09.035
34. Jiang, N., Becker, D., Michelogiannakis, G., et al.: A detailed and flexible cycle-accurate network-on-chip simulator. In: Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software (2013)
35. Onoe, Y., Yamazawa, K., Takemura, H., Yokoya, N.: Telepresence by real-time view-dependent image generation from omnidirectional video streams. Comput. Vis. Image Underst. **71**(2), 154–165 (1998). http://www.sciencedirect.com/science/article/pii/S1077314298907056
36. Yamazawa, K., Takemura, H., Yokoya, N.: Telepresence system with an omnidirectional HD camera. In: Proceedings of the 5th Asian Conference on Computer Vision (ACCV 2002), vol. 2, pp. 533–538 (2002)

37. Ikeda, S., Sato, T., Yokoya, N.: Panoramic movie generation using an omnidirectional multi-camera system for telepresence. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 1074–1081. Springer, Heidelberg (2003)
38. LSM. Real-time panoptic video by EPFL-LSM. http://www.youtube.com/user/LSMPanoptic/videos
39. Popovic, V., Seyid, K., Pignat, E., Cogal, O., Leblebici, Y.: Multi-camera platform for panoramic real-time HDR video construction and rendering. J. Real-Time Image Process., 1–12 (2014). http://dx.doi.org/10.1007/s11554-014-0444-8
40. Popovic, V., Pignat, E., Leblebici, Y.: Performance optimization and FPGA implementation of real-time tone mapping. IEEE Trans. Circuits Syst. II Express Briefs **61**(10), 803–807 (2014)
41. Chander, G., Markham, B.L., Helder, D.L.: Summary of current radiometric calibration coefficients for Landsat MSS, TM, ETM+, and EO-1 ALI sensors. Remote Sens. Environ. **113**(5), 893–903 (2009)
42. Jungmann, J.H., MacAleese, L., Visser, J., Vrakking, M.J.J., Heeren, R.M.A.: High dynamic range bio-molecular ion microscopy with the timepix detector. Anal. Chem. **83**(20), 7888–7894 (2011)
43. Bloch, C.: The HDRI Handbook 2.0: High Dynamic Range Imaging for Photographers and CG Artists. Rocky Nook, Santa Barbara (2013)
44. Yoshida, A., Blanz, V., Myszkowski, K., Seidel, H.-P.: Perceptual evaluation of tone mapping operators with real-world scenes. In: SPIE Human Vision and Electronic Imaging X, pp. 192–203 (2005)
45. Drago, F., Myszkowski, K., Annen, T., Chiba, N.: Adaptive logarithmic mapping for displaying high contrast scenes. Comput. Graph. Forum **22**(3), 419–426 (2003)
46. Brown, M., Lowe, D.: Automatic panoramic image stitching using invariant features. Int. J. Comput. Vision **74**(1), 59–73 (2007)