Technical report

CCA Functional Structure & Architecture

Item, NTNU

Version: 1.06

Project: Collaboration Surfaces

Author: Leif Arne Rønningen

Date: 29Aug11

File: CCAFuStruct&Arch1

The Camera Cluster Array

A Camera Cluster Array consists of an array of Clusters. A Cluster consists of a number of densely placed cameras. A number of Camera Cluster Arrays are used in Collaboration Surfaces, as described below.

Collaboration Surfaces – DMP, the Hems Lab

Collaboration Surfaces are essential parts of the DMP three-layer systems architecture [RON11a]. DMP provides near-natural virtual networked autostereoscopic multiview video and multi-channel sound collaboration between users, and users and servers. Users use Collaboration Surfaces to interact with the DMP system.

The end-to-end time delay is guaranteed by allowing the quality of audio-visual content and the scene composition vary with time and traffic. The adaptation scheme is called Quality Shaping. The scheme uses traffic classes, controlled dropping of packets, traffic measurements, feedback control, and forecasting of traffic. Adaptive parameters are the end-to-end delay, the number of 3D scene sub-objects and their temporal and spatial resolution, adaptive and scalable compression of sub-objects, and the number of spatial views. The scheme also includes scene object behaviour analysis. The architecture supports pertinent security and graceful degradation of quality.

The Hems Lab is a downscaled DMP realization [RON11a]. While the quality of the scenes intends to approach ‘near-natural quality’ in the long term, the present stereoscopic implementation (version 1.0) is of good quality. The environment combines virtual (from digital library and remote collaborators) and live musical theatre drama elements and sequences, including actors/singers playing/singing roles, and scenography, selected from musical theatres/operas. Other applications of Hems Lab are TV productions, games, education, and virtual meetings.

The Camera Cluster

The Camera Cluster version v1.0 consists of nine LUPA-300 Image Sensors [LUP07]. By means a sophisticated mechanical arrangement the Cluster enables each camera to be oriented to cover a certain part of the scene in front. The cameras can synchronously shoot nine non-overlapping images, partly overlapping images, or completely overlapping images. The shots are synchronized at a maximum rate of 250 images per second. A Camera Cluster provides a stitched image of maximum spatial resolution 9 x 640 x 480 pixels = 2.764.800 pixels and a temporal resolution of 250 images per second. In the other extreme, the Cluster can shoot images of 640 x 480 pixels interlaced at a rate of 9 x 250 = 2250 images per second.

The next version of the Camera Cluster, v2.0, should use image sensors with the same shutter and features as the LUPA-300, but hopefully with 150 fps temporal and 2k x 1k spatial resolution. The sensor should use the same package and pinning (48 pin LCC). Several configurations are possible, 4 sensors: 4k x 2k, 6 sensors: 6k x 2k or 9 sensor: 6k x 3k pixels at 150 fps. One to three sensors should be capable of filtering out defined spectral bands from the infrared range. This can be used to support video segmentation.

The Camera Cluster mechanics

The Camera Cluster consists of up to nine Camera Modules. The Camera Modules are mounted on a Common Stand, as shown in Figure 1. Several Camera Clusters can be grouped to constitute a Camera Cluster Array.

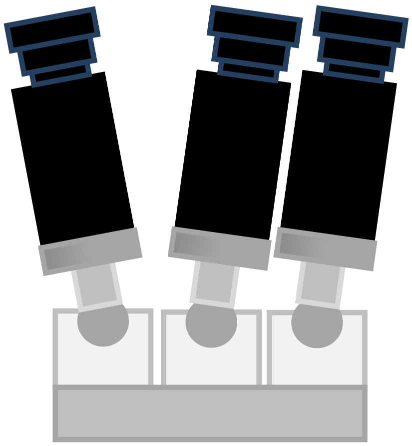

Figure 1. Three Camera Modules placed on the Common Stand.

The upper part is the Lens System, detailed in the next section. The lens system is screwed into the Sensor House from top, to adjust focus. The Sensor House is fastened to a Pose Mechanism. The Sensor is soldered to the Sensor PCB, which has 48 leads through vias to the other component side of the FCB. Flexible Flat Cables interconnects the PCB to an FPGA card (see other sections for more details). The Pose Mechanism can pose the Camera Module to point in any direction into the scene in front (within limits). 3-axis adjustment is needed, and can be accomplished by a manual ball-joint arrangement, a micro-robot, or other.

In documents [RON09a] and [RON09b] the overall mechanical construction and the micropositioning system are shortly described.

The LUPA-300 CMOS Image Sensor

The main features of the LUPA-300 are shown in Table 1. Figure 2 shows the imaging surface, the bonding, the pin layout and the size of the LUPA-300 integrated circuit [LUP07].

Table 1. Main parameters of LUPA-300 Image Sensor (source, LUPA-300 data sheet)

Parameter	Typical View
Optical Format	½ inch
Active Pixels	640 (H) x 480 (V)
Pixel Size	9.9 µm x 9.9 µm
Shutter Type	Electronic Snapshot Shutter
Maximum Data Rate/	
Master Clock	80 MPS/80 MHz
Frame Rate	250 fps (640 x 480)
ADC Resolution	10-bit, on-chip
Responsivity	3200 V.m2/W.s 17 V/lux.s
Dynamic Range	61 dB
Supply Voltage	
Analog	2.5V-3.3V
Digital	2.5V
I/O:	2.5V
Power Consumption	190 mWatt
Operating Temperature	–40C to 70C
Color Filter Array	RGB Bayer Pattern
Packaging	48-pins LCC

The snapshot shutter is a must to avoid distortion of fast moving objects of a scene. Rolling shutters do not meet our requirements.

(Adapted from the LUPA-300 data sheet):
The VGA-resolution CMOS active pixel sensor features synchronous shutter and a maximal frame-rate of 250 fps in full resolution. The readout speed can be boosted by means of sub-sampling and windowed Region of Interest (ROI) readout. High dynamic range scenes can be captured using the double and multiple slope functionality. User programmable row and column start/stop positions allow windowing and sub sampling reduces resolution while maintaining the constant field of view and an increased frame rate. The programmable gain and offset amplifier maps the signal swing

to the ADC input range. A 10-bit ADC converts the analog data to a 10-bit digital word stream. The sensor uses a 3-wire Serial-Parallel (SPI) interface. It operates with a 3.3V and 2.5V power supply and requires only one master clock for operation up to 80 MHz pixel rate. It is housed in a 48-pin ceramic LCC package.

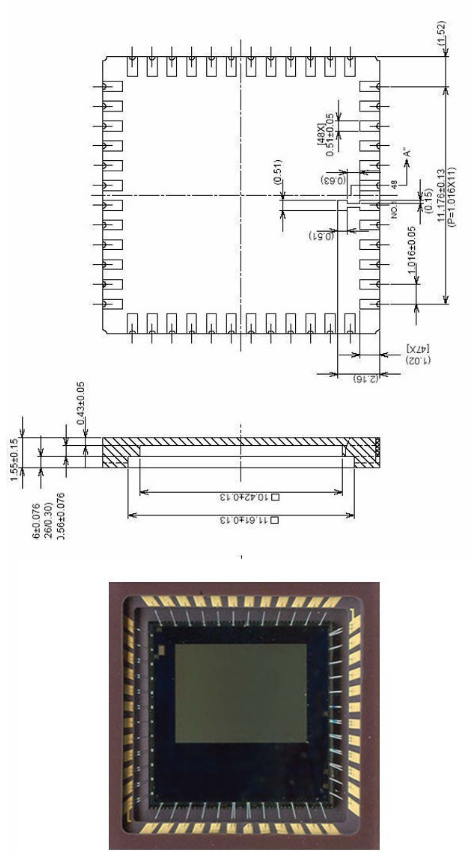

Figure 2. The LUPA-300 CMOS Image Sensor (source, LUPA-300 data sheet)

The sensor used here is the Bayer (RGB) patterned color filter array version.

Note that the image rate for a single sensor can be increased to 1076 images per second, but with 256 x 256 pixels per image.

The LUPA evaluation kit

The LUPA Demo Kit consists of two main parts, a box with an image sensor, lens and hardware, and image software running on a PC under Windows XP. The box contains a RAM buffer which can receive and store short videos at maximum shooting rate. When the buffer is full, the video can be transferred to the PC at a much lower data rate for further processing and inspection. The video can be stored in different file formats. The PC can configure the sensor on-the-fly (see LUPA-300 data sheet for details) sensor to evaluate the specification and quality [LUP09].

The Printed Circuit Board - PCB

The PCB is a four-layer, surface mount board. The highest frequencies are 80 MHz. Some analog parts are sensitive to noise, and are separated from digital signals. Separate earth planes are used for the analog and digital parts. To minimize interference through power supply leads, all analog power supplies are implemented by separate regulators. The image sensor is placed on one side of the PCB while all regulators, connectors, R and C are placed on the other side. Two shielded FFCs connect the PCB and a ‘trans-connector’ on the Common Stand. The power supply is +3.3 V regulated DC, one lead for digital supplies and one for analog supplies. An image of the circuit board is shown in Figure 3.

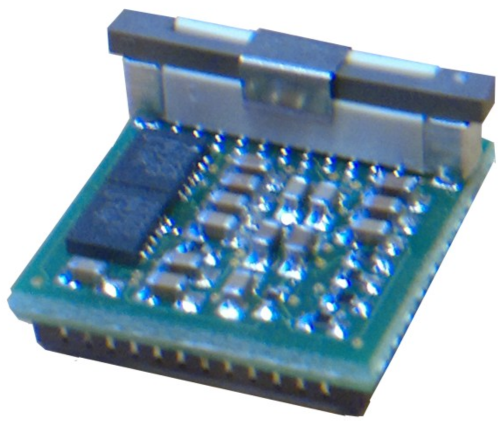

Figure 3. The camera PCB with the image sensor mounted on bottom side. Size 15.5 x 15.5 mm

The Camera Cluster optics

The Lens System is provided by Lensagon, the model used is BSM16016S12 [BSM09]. The main technical features and specifications are given in Table 2 below. Figure 4a shows a perspective image of the lens system. In Figure 4b the mechanical drawing with measures is shown.

Figure 4a. Lensagon BSM16016S12 lens

Table 2. Technical Specifications

Format		1/2"
Mount		M12, 0.5
Focal length		16mm
Aperture		1:1.6
Focus Extent		0.2m – infinite
Back Focal Length		6.58mm
F.O.V.(DxHxV)		24° x 19.7° x 16.2°
Lens Construction		4 Components 4 Group
Weight		6g
Note		Megapixel, IR corrected

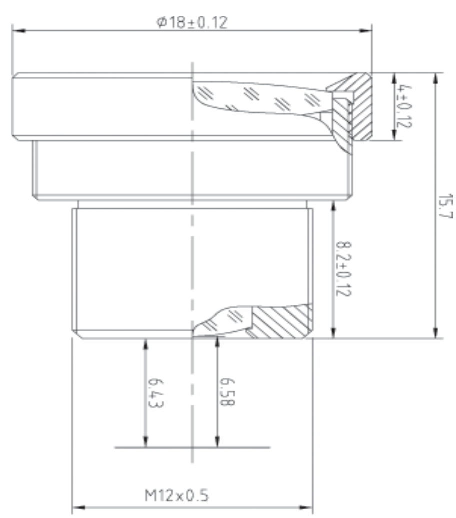

Figure 4b. BSM16016S12 lens, mechanical drawing (source: Lensagon).

Balancing optics and sensor resolution

As described by Fiete in [FIE10], the resolution of the optics and the sensor should be balanced. In theory, the ratio of the detector sampling frequency to the optics diffraction cutoff should be one, giving the so-called Q-value equal to two. In practice, 0.5 < Q < 1.5 is often chosen, to obtain a sharper and brighter image of large scenes. In this design, the imaging chip has been chosen to meet the frame rate of 250 Hz. The reason is that the perceived quality of high frame rates (50-250 Hz) is to be tested. The lens is then selected to match the imaging chip. In the final design it is desirable to have a lens diameter of 1 mm, and then the optical system will be diffraction limited. When and if imaging chips with say 150 Hz frame rate and 2k x 1 k resolution will be available, is uncertain.

The Camera Cluster

The Common Stand supports normally nine Camera Modules per Cluster. The Camera Modules are placed as dense as practical for adjustments, see Figure 5. The distance between the centers of Clusters are normally 6.5 cm, the normal distance between human eyes. The Common Stand can be mounted on a standard camcorder tripod (two standard holes), or other.

One Cluster includes three CCAB-01 boards with a Virtex 6 FPGA onboard. That is, one CCAB-01 can serve three Camera Modules. Flexible flat cables, each 22 lead, connect the Camera PCBs with the CCAB-01. The CCAB-01 provides power, 3.3 V DC for the Camera Modules. The PCIe standard is used for external communication.

One Xilinx ML605 board with a three-input PCIe Mezzanine Card can serve several Camera Clusters. A high-end PC can include three ML605 boards. Backplanes with PCIe switches can take 10-20 PCIe x4 and some x8 boards.

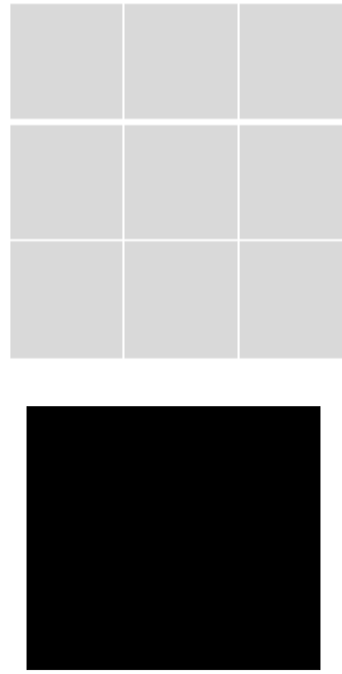

Figure 5. A Cluster on the Common Stand

Micropositioning Electro-mechanics

This section describes a micropositioning stage with rotation (motion) around the x, y and z-axis, and translation in the z direction. It is based on the principle of 'hinges', as shown in Figure 5 in [RON07a], but has got ultra-compact stepper motors with gears to provide accurately controlled motion.

The basic building block is the combination of a small cylindrical stepper motor house with a mounting plate rigidly attached, and a cylindrical gear house with another mounting plate rigidly attached. The two cylinders are mounted 'end-to-end' and rotate relative to each other, by means of the stepper motor and the gear, and with a roller bearing in between. This is shown in Figure 6. When two building blocks are joined together by fastening the mounting plates 90° to each other, rotation around the x and y-axis are obtained.

A second building block, with slightly different mounting plates as shown in Figure 4 in [RON07b], is introduced to obtain rotation around the z-axis, and translation in the z direction. Four lead-screws take care of the z-translation, while the same stepper motor with gear as above can be used for the rotation.

When two first and one second building blocks are joined together, rotation around x, y and z-axis are obtained by electronic control, and in addition, z-axis translation can be done manually.

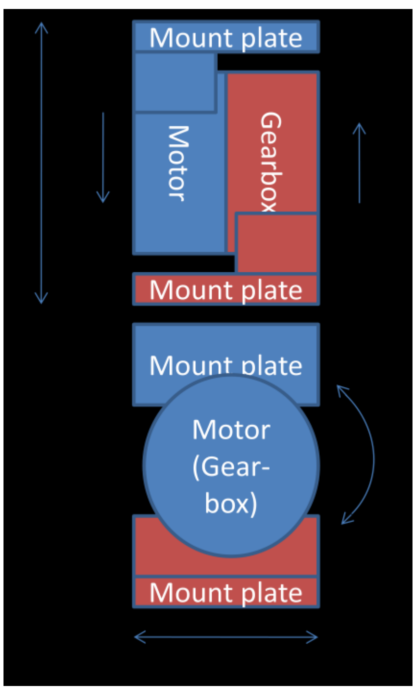

Figure 6. The building block

Table 1. SM4.3 stepper motor specification

Product name	Stepper Motor "SM4.3 series"
Dimensions	4.3mm(dia.) x 5.15mm(long)
Shaft diameter	0.7mm
Weight	0.4g
No. of steps (step angle)	20 (1Step angle=18)
Coil resistance	23 ohm/ Phase
Voltage between terminals	3V
Pull-in torque	0.07mN.m(2 Phase.500pps hour)

Stepper motors provide motion in discrete angular steps. Positioning sensors or servo-loops are not needed. A position is always predictable, and it takes a time proportional to the number of steps to go to the actual position. The controllers (control electronics) needed are then quite simple. Drawbacks with stepper motors compared to the alternatives, are high power consumption when holding a position, lower dynamic performance and vibration, if not handled adequately.

Microstepping, for example sine cosine microstepping, can be used to obtain higher position resolution [WIK09a]. Assume that the step of a motor can be reduced from 18° to 18/256 = 0.070° . If a gear of rate 10 is added, the resolution could theoretically be 0.007°. Another advantage with microstepping is that smoother motion is obtained.

In Figure 7 the principle behind a planetary or epicyclic gear is illustrated [WIK09a].

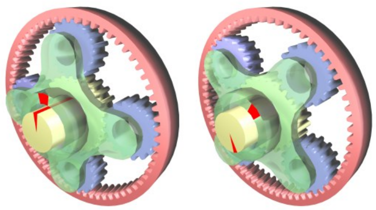

Figure 7. Planetary gear principle.

It is used here to decrease output speed. The planet gear carrier (green) is fixed to the motor house. The sun gear (yellow) is the motor shaft, while the ring gear (red) is rotating output (adapted from [WIK09a]).

Pose Control System

The pose of the Camera Module attached to the positioning combination as explained above, can be controlled by four screws manually for the z-translation, and the z, y and z-rotation by three parallel stepper motor controllers. It is desirable to use the pose control system to calibrate the cameras (and maybe to track and follow moving objects in space), and the controllers should therefore be integrated with the image processing system (FPGA-implementation).

As pointed out, the controllers should be of the sine cosine microstepping type.

In our application, the micropositioning systems will be part of the closed control loop of the Camera Cluster Array, CCA. The cameras will shoot images of the scenes with known reference points and images. This can first of all be used to calibrate each camera, the Camera Clusters, CC, and CCA. The reference information can also be used to avoid too large position deflections of the moving parts of the positioning system. Furthermore, the sine cosine microstepping can be made adaptive.

If wanted, the control loop can be used for moving the cameras when tracking objects. This also gives the possibility to increase temporarily the resolution in time and/or space for important objects which are tracked.

Figure 8. The Camera Module assembled, using a mechanical positioning system. The size of house with lens is approx. 17 x 17 x 38 mm.

The Camera Cluster Array Functional Structure and Behavior

The CCA Functional Structure is shown in Figure 9. The Camera Module.

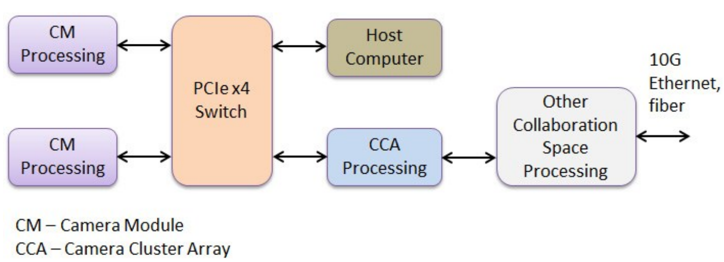

Figure 9. The Camera Cluster Array Functional Structure

Figure 10 shows the functional structure of the CM processing block.

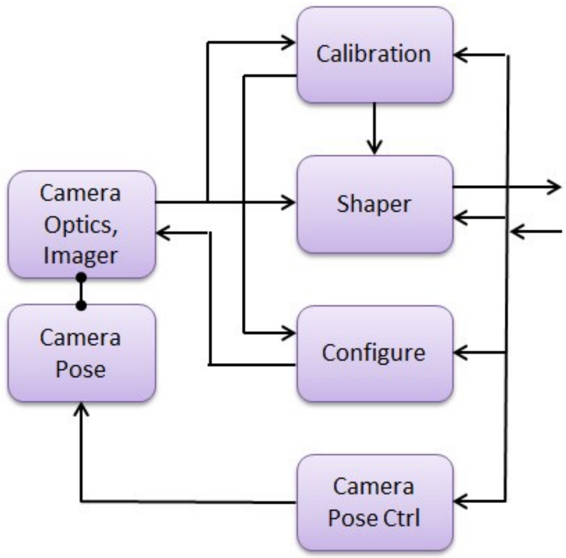

Figure 10. Camera Module Processing

In the first version, the Camera Pose sub-system will be static and mechanical, and manually adjusted.

Initially, calibration of the camera is started by a signal from the Host Computer (operator) with a reset of the Imager. When the camera generates images it sends 10 bits in parallel to the Calibration and Shaper blocks, synchronized by frame valid, line valid and clock signals. The Calibration block starts the FPN and PRNU imager correction procedures, see below, sends the correction and configuration data to the Configure block, which in turn configures the Imager. Then, the Calibration block runs the camera calibration procedures (estimating five distortion parameters and four intrinsic parameters) using chess board images in known pose in front of the camera. The calibration data is used by the Shaper block when real-time generation of images takes place.

FPN and PRNU

FPN (,,,) noise stems from individual variations of the transistors from pixel to pixel, and PRNU (,,) noise stems from variations of column amplifier of the image chip. Corrections can be performed by using known optical inputs, inspecting the resulting image, and writing correction data into chip registers, as described in [LUP09]. Software for this correction is available. A linear regression model built from measured intensities for given calibration scenes can also be used, see [FIE10].

Camera calibration

The camera calibration is normally based on a frontal pinhole camera model which describes idealized projection of points in 3D space onto an image surface, and is extended with corrections parameters for non-square and skewed pixels, changed image origin, and lens distortions. Mathematically, this can be described by projective geometry. Given a known calibration surface (chessboard) in 3D space and the shot 2D image of it, the calibration parameters can be calculated. See [MA06], [SON10] for detailed mathematical descriptions, and [BRA08], [MAT11] for software calibration tools.

Camera pose

The camera pose can be (second version) be realized using microposition techniques, as described in an earlier section.

Shaper

The Shaper in addition performs several functions as shown in Figure 11. Possible parallel and sequential processing is suggested.

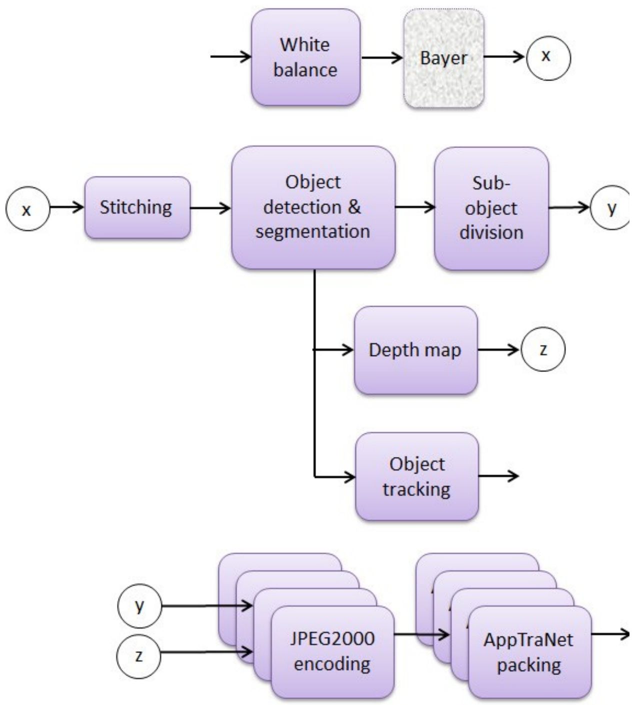

Figure 11. Functional blocks of the Shaper

White balance

The white balance can be corrected by changing the contribution of R, G, and B components of the image pixels, by shooting a known white surface. Most cameras have white balance correction [CAM11], automatic or manual.

Bayer interpolation

The Bayer interpolation is normally performed immediately after the whole image is grabbed from the sensor and stored. In our case, Bayer interpolation is performed just before images are shown, stored for future use, or used in a scene composition. That means that all interpolation is done at the receiver. From the camera, the R, G and B components are low-pass filtered. From the original G component, edges, transitions, features and objects are segmented. At the receiver the Bayer interpolation is performed, and un-sharp edges in the objects are sharped using the edges from the original G component [FIE10].

Stitching

Stitching is to align slightly overlapping images together covering a scene, shot from various positions and viewing angles. Typically images from one cluster will be stitched together. See [WIK11]

Object detection and segmentation

A major feature of DMP is to detect and recognize objects as important, and then segment them from other objects and the background. The start of the process is to find features like corners and edges that can be boundaries of the objects. The theory behind can be found in [SON10], [MA06], and software tools in [MAT11], [BRA08].

Object tracking

When an object has been segmented, it can be of interest to track it for some time over the visual field, to let cameras focus and follow the object. The intention can be to increase the spatial resolution of the images of the object, or send data of the object only, leaving the background out. Theory, techniques and tools can be found in [MAT11], [BRA08], [MA06], [SON10].

Depth map generation

It is intractable to shoot and send all possible views of a scene, and a good compromise is to use depth maps for reconstruction of lacking views. Depth map generation are described in [FAN11],[MAT11], [BRA08], [MA06], [SON10], [LI08].

Sub-object division

The DMP sub-objects are described in [RON11a]

JPEG2000 encoding

See [RON11a]

AppTraNet packing

See [RON11a]

The Camera Cluster Array Processing Architecture

In Figure 12 a possible CCA processing architecture is shown. As described earlier it consists of Camera Modules, processing boards CCAB-01, ML605 boards and PCs.

When the Pose System is mechanical, manual and static, the processing requirements to a number of functions are small. All imager and camera calibration procedures can then be run in software on the Host Computer [LUP10].

Although the functions in Figure 11 should be performed by one ml605 on a complete stitched image from a cluster of cameras, it should be investigated if some functionality could be shared by CCAB-01s and ml605. In the following, it is assumed that Bayer interpolation is not performed by the sender.

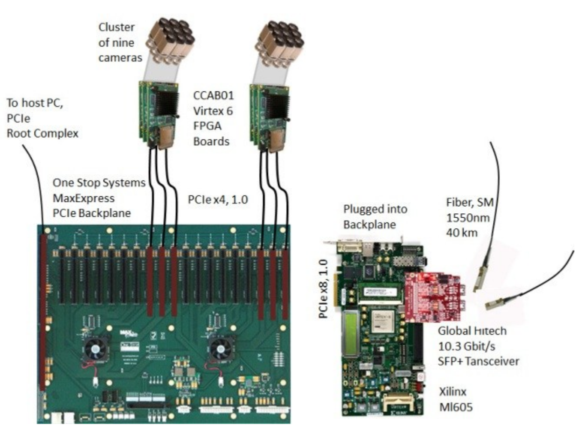

Figure 12. CCA Processing Architecture

See [RON11b] for description of hardware architecture for CCA processing. Note that CCAB-01 boards can be connected in series via a backplane switch. The same can be done with ml605 boards, and combinations of various boards.

Note: Camera modules of a CC or several CCA can be synchronized. This requires a common 80 MHz clock generator. After power on (VDDD on), minimum 500 nano seconds are needed to upload LUPA-300 internal registers, before the input signal RESET_N determines the timing and sequence of operation.

Short review of multiple view, object oriented camera systems

In [MA06] Ma et. al. 'invite to 3D-vision', and cover the classical theory of two-view epipolar geometry, as well as extensions to rank-based multiple view geometry. The textbook also reviews facts from linear algebra, image formation, and photometrics.

Sonka et. al. present in their textbook [SON10] a thorough basis for video segmentation. 3D vision geometry and applications are also treated.

Video segmentation of scenes into background and foreground objects has been a major research area for 2-3 decades. When objects are identified in the static case, the next natural step is to track moving objects. Some papers addressing segmentation are reviewed below.

G Mohommadi et. al. [MOH09] classifies multi-camera tracking into three categories:

- Geometry-based
- Color-based

- Hybrids

Another dimension is whether the cameras are calibrated or not.

In [MOH09] a hybrid version is proposed, using two-views and the following steps:

- Recover the homography relation between camera views
- Find corresponding regions (of objects) between different views by use of regional descriptors (of spatial and temporal features)
- Track objects in space simultaneously across multiple viewpoints

Video segmentation with invisible illumination can be regarded as a hybrid category. Ifrared (IR) light is then added to the RGB light in the scene to support the segmentation. This can also be regarded as an extension of chroma keying, known from movie production for decades.

Support of IR light has been tested by several researchers, e. g., [DAV98], [WU08], [CON06]. In [WU08] Wu et. al. perform bilayer video segmentation by combining IR video with color video. Their contrast-preserving relaxation labeling algorithm is shown to perform better than graph-cut algorithms.

A problem with IR supported segmentation arises when scene objects are made of glass. To avoid this, Ben-Ezra has in a paper [BEN02] proposed the Catadiopric camera/image sensor design using beamsplitters and prisms to polarize light on backgrounds while the foreground objects are illuminated by non-polarized light.

There are today several cameras in the market (but expensive) that perform depth measurement and depth map generation, in addition to normal RGB image shooting [LI08].

Advanced Imaging

As pointed out in the introductory section, the camera module can be built using various imaging sensors if the 48 pin LCC package is used. If the pinning is different from LUPA PCBs and the control HW/SW also must be redesigned.

Great possibilities open up if the imaging sensors (say 150Hz, 2k x 1k resolution) basically are monochrome, but have different light filters in front. In addition to using IR for video segmentation, we here propose to use UV light as well, and maybe combine IR, UV and polarization. If a certain amount of cerium is added to phosphate glass, efficient blocking of UV light is obtained. This is certainly for the future, but also silicate glass (normal glass today) with cerium shows improved blocking effect of UV light. How various transparent plastic materials behave have to be studied.

Each of the cameras in a CCA (36 cameras) can in principle apply different optical filters. If we to start with, limit the different filter to within a cluster, we can have

A. nine spectral sub-bands in the visible frequency band, 150Hz, 2k x 1k, or
B. six spectral sub-bands in the visible band and three infrared or/and ultraviolet sub-bands, 150Hz, 2k x 1k, or
C. seven spectral sub-bands, i.e., seven Gaussian filters with 50 nm interval between 400 and 700, and a bandwidth of 70 nm (6), plus two infrared or/and ultraviolet sub-bands, 150Hz, 2k x 1k, or
D. several 150Hz, 4k x 2k configurations, or
E. other

In case A we can with resolution of 12-16 bits per sub-band pixel shoot images with nearly perfect colors [MAR02]. In case B, the infrared bands support video segmentation in 3D space. When three infrared sub-bands are available, the method described in the VASARI project [MAR02] can be extended to help finding the contour of objects seen from three different directions. Case C is an extension of the VASARI project.

It is of course possible to obtain higher spatial resolutions, e.g., 4k x 2k pixels as in Case D. Three spectral sub-bands (RGB) are realistic. One camera could be used for IR.

Other interesting problems we intend to study are related to integrating a camera array with a display (to avoid the Mona Lisa effect [NGU09]). Small lenses of 1-3 mm diameter can be applied, and used in a 'full-duplex' (two-way) manner: shooting scene images inwards, and showing remote scene image pixels outwards. In this way the cameras will be nearly invisible, but will be placed close to the areas of the display that the viewer focuses on.

Due to diffraction effects, small lenses with a diameter of less than 1 mm limit the spatial resolution. Using a number of such lenses to shoot one object, can improve the spatial resolution.

To start with, an up-scaled test bed can be built for experimentation: Install a projector and a back-projection screen. Use the pixels of the projector as sub-pixels for a multiple view (five or more) experiment. Place a 3 x 3 lens array with lenses of diameter 92 mm (cut down to 65 mm squares) in front of the screen. Adjust the projector so that 5 x 5 sub-pixels are shown behind each lens of the array. Watch at a distance of more than (tbd) meters, the multi-view effect. Study the possibility of using a convex lens array on the back side of the screen in order to smooth out the discrete nature of the sub-pixels.

References

[BEN00] Ben-Ezra, M. "Segmentation with Invisible Keying Signal". Proc. IEEE Computer Vision and Pattern Recognition (CVPR), Hilton Head Island, South Carolina, I:32-37, June 2000.

[BRA08] Bradski, G, Kaehler, A. "OpenCV. Computer Vision with OpenCV Library". O'Reilly 2008.

[BSM09] Lens BSM16016S12, Lensagon, 2011

[CAM11] Cambridge in Colors – A Learning Community For Photographers, August 2011. http://www.cambridgeincolour.com/tutorials/white-balance.htm

[CON06] Conaire, C., O'Connor, N. E., Cooke, E., Smeaton, A. F. "Multispectral Object Segmentation and Retrieval in Surveillance Video". ICIP 2006 - 13th International Conference on Image Processing, Atlanta, GA, 8-11 October 2006

[DAV98] Davis, J., Bobick, A. "A Robust Human-Silhouette Extraction Technique for Interactive Virtual Environments". MIT Media Lab, Springer 1998

[FAN11] Yu-Cheng Fan, Wei-Lun Chien, Jan-Hung Shen. "Depth Map Measurement and Generation for Multi-view Video System. Instrumentation and Measurement Technology Conference, IEEE 2011.

[FDK05] "Ultra-compact Stepper Motor, SM4.3 series", FDK Coorporation, June 2005.

http://www.fdk.com/hatsnew-e/release050606-e.html

[FIE10] Fiete, R. "Modeling the Imaging Chain of Digital Cameras". Spie Press 2010.

[LI08] Li, F, Yu, J, Chai, J. "A hybrid camera for motion deblurring and depth map super-resolution". IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[LUP07] LUPA-300 CMOS Image Sensor, Cypress 2007

[LUP09] LUPA-300 Demo Kit User Manual, Cypress 2009.

[MA06] Ma, Y., Soatto, S., Kosecka, J., Sastry, S. "An Invitation to 3-D Vision". Springer 2006. ISBN: 10: 0-387-00893-4

[MAR02] Martinez, K., Cupitt, J., Sounders, D. and Pilly, R. "Ten years of Art Imaging Research". Proceedings of the IEEE 90(1) 2002, pp. 28-41

[MOH09] Mohammadi, G., Dufaux, F., Minh, T. H., Ebrahimi, T. "Multi-view video segmentation and tracking for video surveillance". SPIE Proc. Mobile Multimedia/Image Processing, Security and Applications, 2009

[NGU09] Nguyen, D T., Canny, J., "More than Face-to-Face: Empathy Effects of Video Framing". CHI2009 – Telepresence and Online Media. Boston April 6th 2009.

[NID09] "PPD6, 6mm Stepping Motor", Nidec Copal, November 2009. http://www.nidec-copal-usa.com/PDFs/LPD6%20data.pdf

[PI09] "Micropositioning Fundamentals", page 128 in "Micropositioning Precision Linear / Rotary / Multi-Axis Positioning Systems". PI October 2009.

[RON09a] Rønningen, L A. "The Camera Cluster Array. Electromechanical Design", Technical Report, Item, NTNU October 2009.

[RON09b] Rønningen, L A. "Micropositioning System for the Camera Cluster Array. Electromechanical Design",Technical Report Item, NTNU October 2009.

[RON11a] Rønningen, L A. "The Distributed Multimedia Plays Architecture". Technical Report, Item 2011. http://www.item.ntnu.no/people/personalpages/fac/leifarne/the_dmp_architecture

[RON11b] Ronningen, L A. "DMP Processing Architectures based on FPGA and PCIe". NTNU August 2011. http://www.item.ntnu.no/_media/people/personalpages/fac/leifarne/pciearch2.docx

[SON08] Sonka, M., Hlavac, V., Boyle, R. "Image Processing Analyses, and Machine Vision". Thomson 2008. ISBN: 10: 0-495-24438-4

[WIK09]Wikipedia, "Stepper Motor", November 2009. http://www.wikipedia. http://en.wikipedia.org/wiki/Stepper_motor

[WIK09] Wikipedia, "Epicyclic gearing", October 2009. http://en.wikipedia.org/wiki/Epicyclic_gearing

[WIK11] Image stitching, 2011 http://en.wikipedia.org/wiki/Image_stitching

[WO08] Wu, Q., Boulanger, P., Bischof, W. F."Automatic Bi-Layer Video Segmentation Based on Sensor Fusion".

Proceeding of the 16th ACM
international conference on
Multimedia, Vancouver, British
Columbia, Canada 2008