Adaptive Resolution System for Distributed Surveillance

Dorin Comaniciu[1] Fabio Berton[2] Visvanathan Ramesh[1]

[1]Real-Time Vision and Modeling Department
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540, USA

[2]Aitek SRL
Via della Crocetta 15
Genova, Italy

Abstract

We present a real-time foveation system for remote and distributed surveillance. The system performs face detection, tracking, selective encoding of the face and background, and efficient data transmission. A Java-based client-server architecture connects a radial network of camera servers to their central processing unit. Each camera server includes a detection and tracking module that signals the human presence within an observation area and provides the 2-D face coordinates and its estimated scale to the video transmission module. The captured video data is then efficiently represented in log-polar coordinates, with the foveation point centered on the face, and sent to the connecting client modules for further processing. The current setup of the system employs active cameras that track the detected person, by switching between smooth pursuit and saccadic movements, as a function of the target presence in the fovea region. High reconstruction resolution in the fovea region enables the successive application of recognition/verification modules on the transmitted video without sacrificing their performance. The system modules are well suited for implementation on the next-generation of Java-based intelligent cameras.

1 Introduction

Autonomous processing of visual information for the efficient description and transmission of events and data of interest represents the new challenge for next generation video surveillance systems [1, 2]. The advances in the new generation of intelligent cameras having local processing capabilities, either supporting Java applications or based on DSP chips, will make possible the customization of the devices for a specific video understanding and summarization task. As a result, important bandwidth reduction can be achieved, combined with a decrease in the operational cost [3]. The low bit rate allows wireless transmission of the data of interest to a central processing unit, for further processing, and/or retransmission. This scenario is not limited to video surveillance and is appropriate for industrial monitoring or videoconference applications [4, 5].

This paper presents a system for real-time distributed surveillance that follows the guidelines from above. Starting with our previous work on object tracking [6, 7], we have built a modular client-server architecture that supports the intelligent transmission of the visual information from a radial network of (active) cameras to the central unit (see Figure 1).

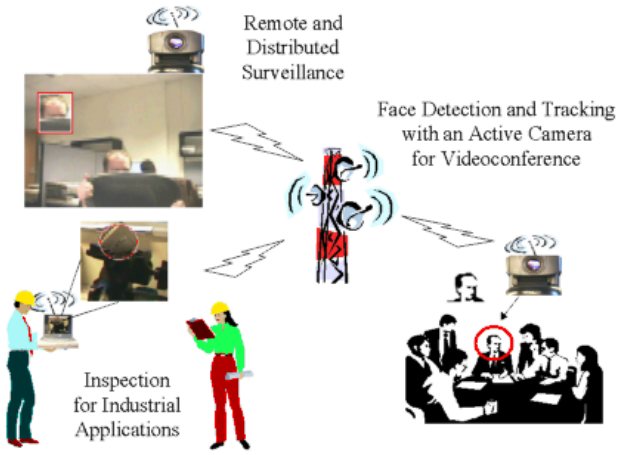


Figure 1: Intelligent processing and summarization of video in a radial network of cameras. Wireless connections can be employed for the transmission of data of interest to the central processing unit.

Each camera module functions as an image server that reports to the client (the central processing unit) whenever a human is present in its field of view. For the transmission of the visual information associated with this event, the data is filtered and sampled in log-polar coordinates, with the foveation point centered on the subject's face. The client performs the inverse mapping to derive an approximated replica of the original data.

A flexible log-polar representation of the background has been designed to deal in real-time with scale and location changes of the face of interest, while maintaining the required bit rate. The main idea is to preserve the high-resolution of the face and trade-off the quality of the background for the efficiency of the representation. Our system performs lossless transmission of the fovea region, hence, it allows the application of recognition/verification processes at the central unit. As a novelty, the sampling grid employed to compute the log-polar representation has locally a hexagonal lattice structure, known to be 13.4% more efficient than rectangular lattices in the case of circularly bandlimited signals [8].

The paper is organized as follows. Section 2 presents a short review of color-based face detection and tracking techniques and discusses foveation processing. Section 3 summarizes the mean shift-guided module performing the detection and tracking of faces. The flexible log-polar mapping is presented in Section 4. Section 5 describes the dual-mode active camera controller. The system performance is analyzed in Section 6.

2 Background

This section provides first a brief survey on face detection and tracking using color. For a more detailed presentation see [9, 10]. Color based methods are classified as feature invariant

approaches, since they aim at locating faces even when the illumination, viewpoint, or pose vary.

In [11] the face color is represented in the normalized RG space [12] and face color samples are obtained by analyzing faces of different human races. A face color model that is built with Gaussian assumption is back-projected in the current frame for face localization. Additional geometric and motion constraints are imposed. A different approach is taken in [13], where robustness to occlusions is achieved via an explicit hypothesis-tree model of the occlusion process. Histograms are employed in [14] for a better characterization of the face colors. The search for the face is based on histogram intersection. In addition to the color model, a gradient-based module enforces the elliptical shape of the detected face. The technique in [15] derives a skin-color reference map in the $YCrCb$ color space and binarize the input frame according to the model. Subsequent stages perform morphological operations and contour extraction for the localization of faces. The perceptually uniform *CIE Lab* color space is employed in [16] to build chroma charts that are used afterwards to transform the input into a skin-likelihood image. The peaks in the skin-likelihood image represent probable face locations. Adaptive Gaussian mixtures are used in [17] to model the face colors based on the changing in appearance. The HSI (hue, saturation, intensity) representation was employed and color distributions were modeled in the hue-saturation space.

In second part of this section we discuss the problem of foveation processing. The foveation processing of video is mainly inspired by the primate visual system that reduces the enormous amount of available data through a non-uniformly sampled retina [18]. The density of photoreceptors and ganglion cells varies in the retina of the human eye, being larger at the point of the fovea and dropping away from that point as a function of eccentricity. There has been a growing interest in recent years in video processing techniques that resemble the primate visual system [19, 20]. These techniques divide the field of view into a region of maximal resolution, (fovea), and a region whose resolution decreases towards the extremities (periphery) [21]. The motivation is that many applications only require high resolution representation for some parts of the video data. Generally, the fovea covers the object of interest and the background is represented by the periphery. This framework extends naturally to multiple objects of interest. In a surveillance scenario, for example, one is interested to represent with high resolution the human faces for subsequent face recognition/verification procedures.

A very common implementation of the foveation is through log-polar sampling [22]. The fovea is typically uniformly sampled, while the periphery is sampled according to a log-polar grid. Space-variant sensors that implement in CCD technology the mapping between the Cartesian and log-polar plane are presented in [23, 24]. The use of a specific lens system to obtain non-uniform resolution is discussed in [25]. However, a major drawback of these approaches

is that the parameters of the data reduction are fixed, depending on the physical design. A more flexible solution, to which we subscribe, is the software implementation of the log-polar mapping [26]. The performance through the use of additional hardware is discussed in [27].

The definition of anti-aliasing and interpolation filters is a difficult problem for non-uniform sampling. In the simpler case of polar sampling, it can be shown [28, 29, 30] that a real function $f(\rho, \theta)$ whose Fourier transform is of compact support can be reconstructed from its equally spaced samples in the θ direction and at the normalized zeros of Bessel functions of the first kind, in the ρ direction. Efficient approximations for reconstruction are presented in [31]. However, the Fourier transform formulation is not valid when the mapping is space-variant, such as the log-polar mapping. Although there exist fast methods for reconstruction from non-uniform samples [32], their efficiency is still not appropriate for real-time applications. The anti-aliasing filtering in the case of log-polar sampling is typically based on position-dependent Gaussian filters with a disk-shaped support region with exponentially growing radius size [27]. Note that a frequency domain representation can be constructed by applying the log-polar mapping directly to the Fourier integral. The result is the Exponential Chirp Transform, shown to preserve the shift-invariant properties of the usual Fourier transform [33].

3 Face Detection and Tracking

This section presents our module performing the detection and tracking of faces. The main idea is to combine viewpoint invariant models of the face with efficient gradient based optimization. For more details please see [6].

3.1 Modeling and Optimization Framework

The color model of the face is obtained by computing the mean histogram of face instances recorded in the morning, afternoon, and at night. The mean histogram is represented in the intensity normalized RG space with 128×128 bins. As a dissimilarity measure between the face model and the face candidates, a metric based on the Bhattacharyya coefficient [34] is employed. Hence, the problem of face localization is reduced to a metric minimization, or equivalently to the maximization of the Bhattacharyya coefficient between two color distributions. By including spatial information into the color histograms we showed that the maximization of the Bhattacharyya coefficient is equivalent to maximizing a density estimate. As a consequence, the gradient ascent mean shift procedure [35] can be employed to guide a fast search for the best face candidate in the neighborhood of a given image location.

The resulting optimization achieves convergence in only a few iterations, being thus well suited for the task of real-time detection and tracking. To adapt to the scale changes of the

target the scale invariance property of the Bhattacharyya coefficient is exploited as well as the gradient information on the border of the hypothesized face region. Mean shift processes are run at different scales and the scale corresponding to the best response is chosen.

3.2 Detection

The detection process involves the mean shift optimization with multiple initializations, each one in a different location of the current image frame. For the current settings of the system (320×240 pixel images with subjects at a distance between 30cm to 3m from the camera) we use five initial regions of elliptical shape with semi-axes equal to $(37, 51)$, as shown in Figure 2. This arrangement guarantees that at least one initial ellipse is in the basin of attraction of a face of typical size.

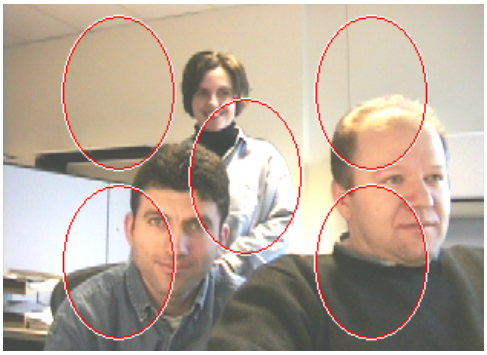


Figure 2: *Three people* image and initialization ellipses used for face detection.

Figure 3 presents the surface obtained by computing the Bhattacharyya coefficient for the entire image from Figure 2. One can easily identify the three peaks of the surface, one for each face in the image. The advantage of our method should be obvious from Figure 3. While most of the tracking approaches based on regions must perform an exhaustive search of the image (or a given neighborhood) to find the maximum value of the similarity measure, our algorithm exploits the gradient of the surface to climb to the closest peak. With proper multiple initializations, the highest peak is found very quickly.

3.3 Tracking

The tracking process involves only optimizations in the neighborhood of the previous face location estimate, and is therefore sufficiently fast to run at the frame rate (30fps). The module that implements the log-polar mapping receives from tracking module two vectors for each frame, representing the estimated position and scale of the currently observed face.

A set of experiments are presented in Figure 4, demonstrating face detection and tracking on a sequence of 1059 frames. The subject's face has been detected within a few frames after

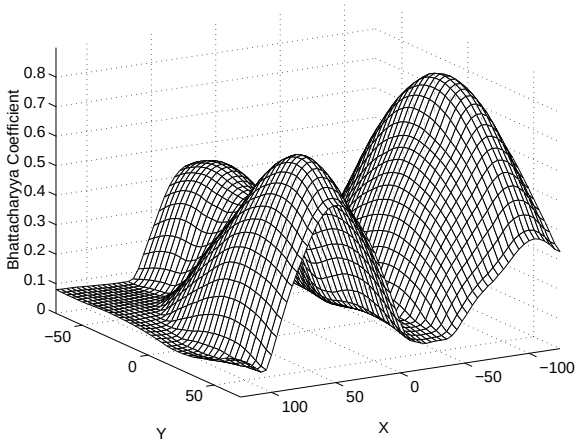


Figure 3: Values of the Bhattacharyya coefficient corresponding to the image shown in Figure 2. One can identify the three peaks of the surface, one for each face in the image.

entering the field of view of the camera (frame 42). The detected face is shown in the small upper-left window. The camera is then tracking the face during walking (frames 69 and 147) and turnings (frames 99 and 177). The subject tries to escape the tracker by performing fast lateral movements (frame 267) or hand waving (frame 309). Observe the blurring that accompanies these movements, without affecting the tracker. Next, the subject tries to hide behind a chair (frames 552 and 582), but only when the head is completely occluded the tracker fails (frame 654). However, once the occlusion is terminated, the face is immediately recovered. Finally, one can see the scale adaptation working when the subject approaches the camera.

4 A Flexible Log-Polar Mapping

The design of a flexible log-polar mapping that satisfies certain geometric constraints is presented in this section. Let us denote by (x, y) the Cartesian coordinates, with the origin assumed to be in the center of the image, and by (ρ, θ) the polar coordinates. The transformation between Cartesian and polar grid is then defined by the pair

$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \theta = \arctan y/x \end{cases} \tag{1}$$

and

$$\begin{cases} x = \rho \cos\theta \\ y = \rho \sin\theta \end{cases} \tag{2}$$

The log-polar mapping is defined by the transformation

$$\begin{cases} \rho_\xi = A\lambda^\xi - B \\ \theta = \theta_0 \phi \end{cases} \tag{3}$$

where (ξ, ϕ) are positive integers representing the log-polar coordinates, $\lambda > 1$ is the base of the transformation, and A, B, and θ_0 are constants that will be derived from geometric constraints.

Figure 4: *Dorin* sequence.

The coordinate ξ represents the index of rings whose radius increases exponentially, while ϕ denotes equidistant radii starting from origin.

4.1 Geometric Constraint for Rings

Let the index of the smallest ring be $\xi = 0$ and the index of the largest ring be $\xi = M$, and ρ_0 and ρ_M their radii, respectively. There are M rings that cover the periphery (see Figure 5). The ring of index 0 is the border between the fovea and periphery.

We assume in the sequel that M, ρ_0, and ρ_M are know and will show how to compute the parameters A, B, and λ of transformation (3). Since the fovea is represented with full resolution, a natural geometric constraint is that the radius difference between the rings of index 1 and 0 is equal to one. Using the above constraint it can be shown (see Appendix) that λ is the real root of the polynomial of order $M - 1$

$$g(\lambda) = \lambda^{M-1} + \lambda^{M-2} + \ldots + 1 - (\rho_M - \rho_0) \; . \tag{4}$$

When $M < \rho_M - \rho_0$, the polynomial $g(\lambda)$ has a root larger than 1. Using the same constraint

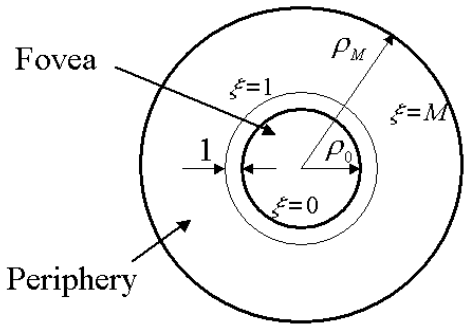


Figure 5: The first geometric constraint imposes that the radius difference between the rings of index 1 and 0 is equal to one.

we obtain

$$A = \frac{1}{\lambda - 1}. \tag{5}$$

and

$$B = \frac{1}{\lambda - 1} - \rho_0 \tag{6}$$

Introducing now (5) and (6) in (3), we have

$$\rho_\xi = \rho_0 + \frac{\lambda^\xi - 1}{\lambda - 1}\ . \tag{7}$$

4.2 Geometric Constraint for Radii

The parameter θ_0 is derived such that the resulting sampling grid (defined by the centers of the log-polar pixels) has locally a hexagonal lattice structure. This modified log-polar mapping has been suggested in [30], but (to our knowledge) has not yet been exploited. The processing of hexagonal sequences is discussed in [8].

Our construction assumes the rotation of each other ring by $\theta_0/2$. To obtain a structure that is locally hexagonal, the aspect ratio of the log-polar pixels should be $\sqrt{3}/2$, that is, the centers of three adjacent pixels should form an equilateral triangle. Although the aspect ratio changes with the index of the ring, for typical parameter values the changes are negligible. By enforcing the hexagonal constraint for the pixels at the half of the periphery region ($\xi_h = M/2$), it can be shown after some trigonometric manipulations that

$$\theta_0 = 2 \arctan \frac{2}{\sqrt{3}} \frac{\lambda^{\xi_h} - \lambda^{\xi_h - 1}}{\lambda^{\xi_h} + \lambda^{\xi_h - 1} - 2 + 2\rho_0(\lambda - 1)}\ . \tag{8}$$

4.3 Mapping Design

According to the formulas derived in Sections 4.1 and 4.2, the log-polar grid is completely specified if the number of rings M is given, together with the radius of fovea region ρ_0, and the maximum radius of the periphery ρ_M. As an example, Figure 6 presents the log-polar pixels and the sampling grid corresponding to an image of 160×120 pixels. We imposed a number of $M = 30$ rings, a fovea radius $\rho_0 = 9$, and a maximum periphery radius $\rho_M = 100$, equal to half of the image diagonal. Solving for the real root of polynomial (4) it results that $\lambda = 1.068$ and using (8) we have $\theta_0 = 0.0885$ radians.

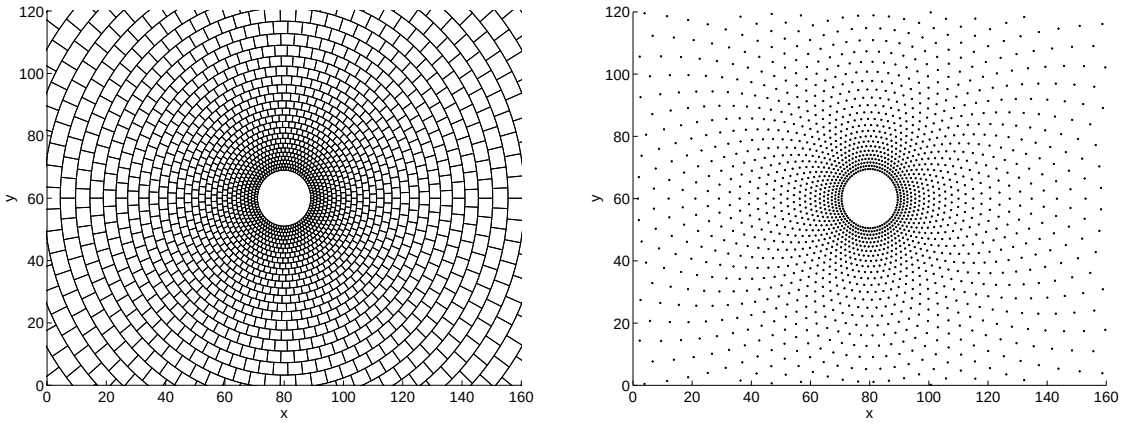


Figure 6: (a) Log polar pixels for an image of 160×120 pixels. (b) The corresponding sampling grid exhibits a local hexagonal structure. See text for the numerical value of the parameters.

The number of log-polar pixels is $b = M * \lceil \frac{2\pi}{\theta_0} \rceil = 30 \times 71 = 2130$ pixels, which is 9 times smaller than the number of Cartesian pixels (the notation $\lceil . \rceil$ denotes the ceiling function). When the foveation point does not coincide with the image center, the number of log-polar pixels decreases.

4.4 Imposing the Transmission Bandwidth

Assume in the sequel a communication scenario with fixed bandwidth. The main idea of our system is to transmit with full resolution the detected face, while adapting the log-polar mapping for the remaining bandwidth of the communication channel. The error free transmission of the face is required in the case when a recognition module is employed at the receiver site.

Since the dependence between the number of log-polar pixels and the input parameters of the mapping is complex, we employ a LUT of size 100KB to determine the mapping parameters for a given bandwidth and fovea size.

4.5 Implementation Details

The input image of size 320×240 is first converted to the $YCbCr$ color format, the color planes being subsampled to 160×120 pixels. Based on the location and scale information received from the tracking module, the fovea regions of both the luminance and color planes are transmitted separately.

If the bandwidth of the communication channel is known in advance, the log-polar mapping has only one variable parameter, the radius of fovea region (the number of rings being constrained by the bandwidth). By coarsely quantizing the range of values of the fovea radius into ten values, for each radius value we built a set of LUTs for anti-aliasing filtering and sampling. The pre-sampling filtering employs Gaussian filters whose kernel width is position dependent. More specifically, the width of each kernel is proportional to the local distance between two consecutive rings at the location of the kernel. The Gaussian kernels are precomputed in small LUTs. The relative position of the sampling points for the ten fovea radii is memorized in a LUT of 500KB for the luminance data and a LUT of 125KB for the color data. Bilinear interpolation is used for the recovery of uniformly sampled data. The interpolation for each Cartesian location is based on its bounding triangle (Figure 6b).

The resulting system maintains an approximately constant transmission bandwidth, while allowing the full resolution transmission of the detected face, independent of the scale. The penalty is paid by the periphery (background) whose quality decreases when the face scale increases.

5 Camera Control

In the case when active cameras are used, the adequate control of the pan, tilt, and zoom is an important phase of the tracking process. The camera should execute fast saccades in response to sudden and large movements of the target while providing a smooth pursuit when the target is quasi-stationary [36, 37]. We implemented this type of control which resembles that of the human visual system. The fovea subimage occupies laterally about 6 degrees of the camera's 50 degrees field of view, at zero zoom.

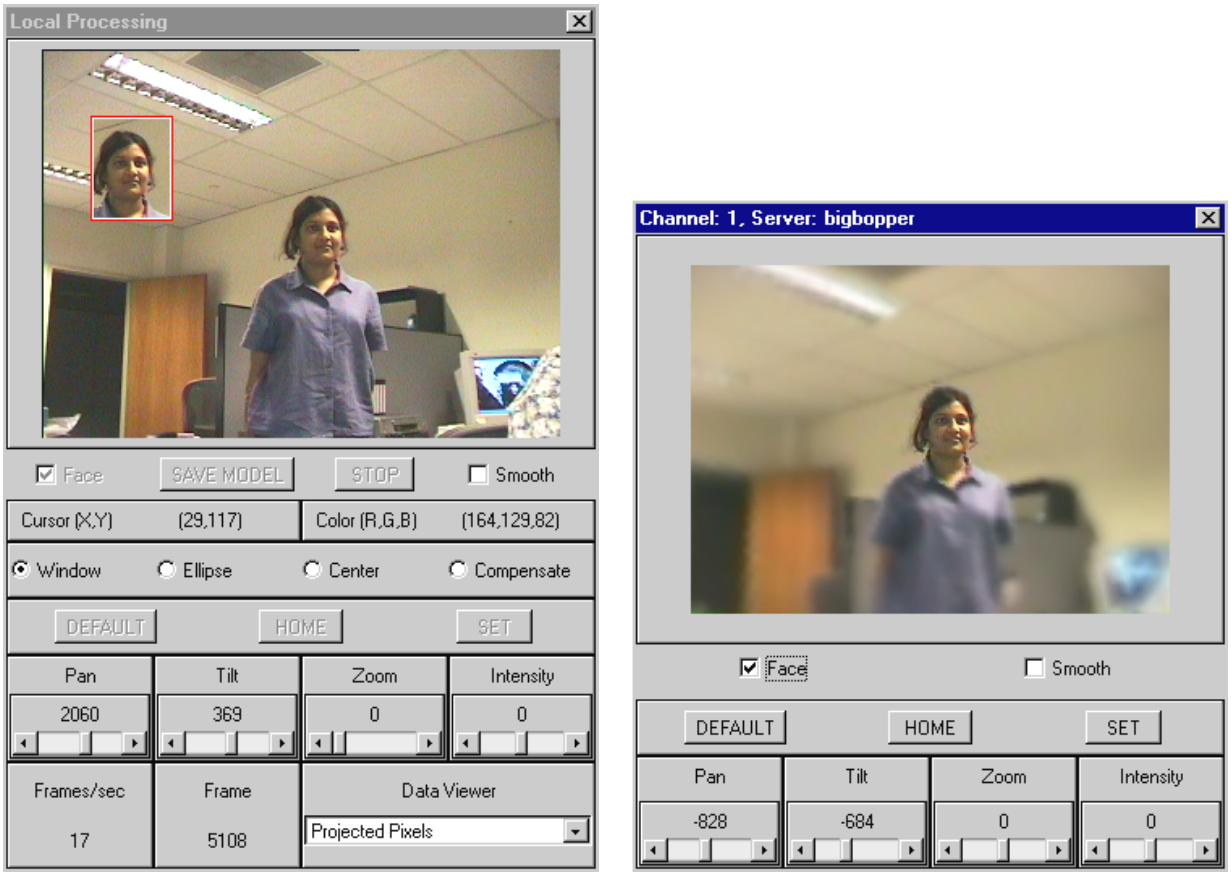
However, contrary to other tracking systems that suspend the processing of visual information during the saccades movements [38], our visual face tracker is sufficiently robust to deal with the large amount of blurring resulting from camera motion. As a result, the visual tracking is a continuous process that is not interrupted by the servo commands. A standard $RS-232C$ interface is used to communicate with the $SonyEVI-D30$ camera.

6 System Performance

We discuss in this section a set of results obtained by running the system in an indoor environment. For all considered experiments we impose an overall compression ratio of 16.

6.1 Reconstruction Quality

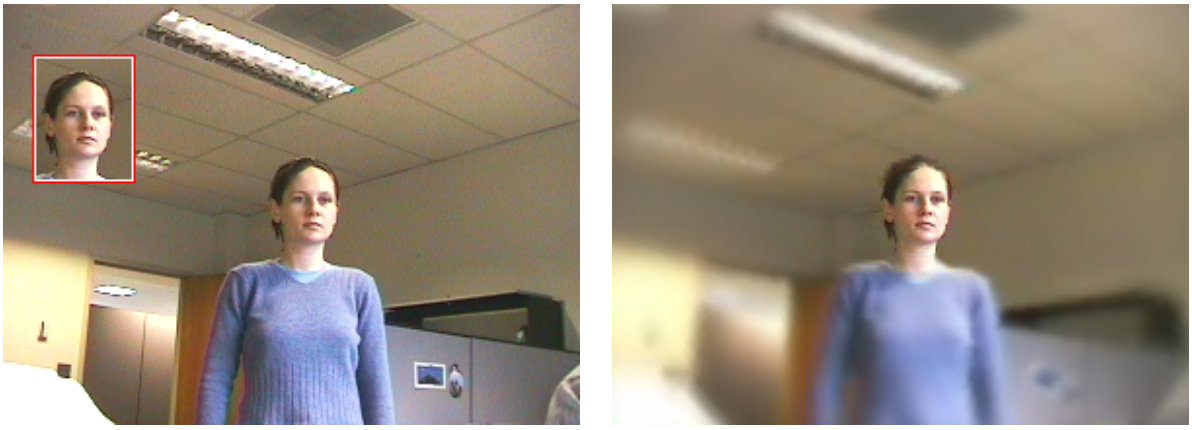
Figure 7(a) presents the user interface of a camera server, showing the the detected face in the upper-left side of the image. The entire server package, performing, detection, tracking, camera control, and adaptive log-polar representation is well suited for implementation on the next generation of intelligent cameras. Figure 7(b) shows the reconstructed image and user interface of the primary client, which is able to control the camera from the distance. Note the preservation of the details on the face region.



(a) (b)

Figure 7: (a) User interface of the image server. (b) User interface of the primary client.

Another pair of original and reconstructed image is presented in Figure 8(a) and (b), respectively. Again, the face details in the reconstructed image are identical to those in the original.



(a) (b)

Figure 8: (a) Original. (b) Received.

The left part of Figure 9 shows 4 frames of a sequence of about 450 frames containing the detection and tracking with an active camera of a subject that meets another person. The same frames, reconstructed by the client application, are presented in the right part of Figure 9. For visualization purposes, both sequences were captured in real-time, simultaneously, on the same computer. Since the entire data transmission chain was operating, one can observe a slight time delay between the original and received frames. The transmission mechanism was implemented through serializable Java classes. An image server and a client performing all processing stages (detection, tracking, efficient representation, transmission, reconstruction, and visualization) operate at frame rate on a single computer.

6.2 Computational Load

We measured the computational load for running the system on a 900MHz PC. The face tracking at 30 fps involves only $10-20\%$ of the CPU. The filtering and log-polar sampling needs additional 40%. The reconstruction filters at the receiver require about 20% of computational power. The Java software can be further optimized by creating C++ DLLs for its most critical parts.

7 Conclusions

This paper presented a flexible framework towards intelligent processing of visual data for low bandwidth communication, with direct application in distributed surveillance using a radial network of cameras. We showed that the region of interest (target) from the current image frame can be detected and transmitted in real-time with high resolution. To compensate for bit rate variations due to the changes in the target scale, the representation quality of the

background is modified through an adaptive log-polar mapping.

We envision further developments of the current system. The extension of the detection module for multiple targets, for example, will also allow the transmission of the face of the second person from Figure 9, frame 240. At the same time, the two streams of information (faces and log-polar pixels corresponding to the background) can be further compressed in the MPEG-4 [39] framework, resulting in additional bandwidth reduction. The overall complexity would be decreased in comparison to direct MPEG-4 compression since the time-consuming motion estimation part of MPEG-4 will be applied only for the reduced log-polar frames. Ongoing research focuses on the implementation of this scenario on smart cameras.

Acknowledgment

We would like to thank our colleagues who helped us with the experiments. Some parts of the material were presented in [40].

APPENDIX

We derive below the base λ and the constants A and B of the transformation

$$\rho_\xi = A\lambda^\xi - B \ . \tag{A.1}$$

By imposing a unit difference between the radii of the rings $\xi = 1$ and $\xi = 0$ we obtain

$$A = \frac{1}{\lambda - 1} \ . \tag{A.2}$$

Using (A.2) in the smallest ring equation

$$\rho_0 = A\lambda^{\xi_m} - B \tag{A.3}$$

it results that

$$B = \frac{1}{\lambda - 1} - \rho_0 \ . \tag{A.4}$$

To obtain λ we introduce (A.2) and (A.4) in (A.1) and write the expression of the largest periphery ring

$$\rho_M = \frac{\lambda^M - 1}{\lambda - 1} + \rho_0. \tag{A.5}$$

which is equivalent to

$$\lambda^{M-1} + \lambda^{M-2} + \ldots + 1 - (\rho_M - \rho_0) = 0. \tag{A.6}$$

This shows that λ is the real root of polynomial of order $M - 1$.

References

[1] Aggarwal, J.K. & Cai, Q. (1999) Human Motion Analysis: A Review. *Computer Vision and Image Understanding.* **73**:428-440.

[2] Collins, R.T., Lipton, A.J. & Kanade, T. (1999) A System for Video Surveillance and Monitoring. *American Nuclear Society Eight Intern. Meeting on Robotics and Remote Systems* Pittsburgh, PA.

[3] Geisler, W.S. & Perry, J.S. (1998) A Real-Time Foveated Multiresolution System for Low-Bandwidth Video Communication. *Human Vision and Electronic Imaging.* Rogowitz, B. & Pappas, T. (Eds), SPIE **3299**:294-305.

[4] Crowley, J.L. & Berard, F. (1997) Multi-Modal Tracking of Faces for Video Communications. *IEEE Conf. on Comp. Vis. and Pat. Rec.* Puerto Rico, 640-645.

[5] Eleftheriadis, A. & Jacquin, A. (1995) "Automatic Face Location Detection and Tracking for Model-Assisted Coding of Video Teleconference Sequences at Low Bit Rates. *Signal Processing - Image Communication.* **7**(3): 231-248.

[6] Comaniciu, D., Ramesh, V. & Meer, P. (2000) Real-Time Tracking of Non-Rigid Objects using Mean Shift. *IEEE Conf. on Comp. Vis. and Pat. Rec.* Hilton Head Island, South Carolina, **2**:142-149.

[7] Greiffenhagen, M., Ramesh, V., Comaniciu, D. & Niemann, H. (2000) Statistical Modeling and Performance Characterization of a Real-Time Dual Camera Surveillance System. *IEEE Conf. on Comp. Vis. and Pat. Rec.* Hilton Head Island, South Carolina, **2**:335-342.

[8] Woodward, M. & Muir, F. (1984) Hexagonal Sampling. *Stanford Exploration Project.* **SEP-38**:183-194.

[9] Hjelmas, E. & Low, B.K. (2001) Face Detection: A Survey. *Computer Vision and Image Understanding.* **83**(3):236-274.

[10] Yang, M.H., Kriegman, D.J. & Ahuja, N. (2002) Detecting Faces in Images: A Survey. *IEEE Trans. Pattern Analysis and Machine Intell.* **24**(1):34-58.

[11] Yang, J. & Waibel, A. (1996) A Real-Time Face Tracker. *IEEE Work. on Applic. Comp. Vis.* Sarasota, 142-147.

[12] Wyszechi, G. & Stiles, W.S. (1982) *Color Science: Concepts and Methods, Quantitative Data and Formulae.* Second Ed. New York: Wiley.

[13] Fieguth, P. & Terzopoulos, D. (1997) Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates. *IEEE Conf. on Comp. Vis. and Pat. Rec.* Puerto Rico, 21-27.

[14] Birchfield, S. (1998) Elliptical Head Tracking using intensity Gradients and Color Histograms. *IEEE Conf. on Comp. Vis. and Pat. Rec.* Santa Barbara, 232-237.

[15] Chai, D. & Ngan, K.N. (1999) Face Segmentation Using Skin Color Map in Videophone Applications. *IEEE Trans. Circuits Sys. Video Tech.* **9**(4): 551-564.

[16] Cai, J. & Goshtasby, A. (1999) Detecting Human Faces in Color Images. *Image and Vis. Computing.* **18**:63-75.

[17] McKenna, S.J., Raja, Y. & Gong, S. (1999) Tracking Colour Objects using Adaptive Mixture Models. *Image and Vision Computing.* **17**:223-229.

[18] Wandell, B.A. (1995) Foundations of Vision. *Sinauer Associates, Inc.* Sunderland, MA.

[19] Wallace, R.S., Ong. P.W, Bederson, B.B. & Schwartz, E.L. (1994) Space Variant Image Processing. *Intern. J. Comp. Vis.* **13**(1):71-90.

[20] Sela, G. & Levine, M.D. (1997) Real-Time Attention for Robotic Vision. *Real-Time Imaging.* **3**:223-194.

[21] Wang, Z. & Bovik, A.C. (2001) Embedded Foveation Image Coding. *IEEE Trans. Image Processing.* **2**(10):1397-1410.

[22] Jurie, F. (1999) A New Log-Polar Mapping for Space Variant Imaging. Application to Face Detection and Tracking. *Pattern Recognition.* **32**:865-875.

[23] Ferrari, F, Nielsen, J., Questa, P. & Sandini, G. (1995) Space Variant Sensing for Personal Communication and Remote Monitoring. *EU-HCM Smart Workshop*. Lisbon, Portugal.

[24] Sandini, G. et al. (1996) Image-based Personal Communication Using an Innovative Space-variant CMOS Sensor. *ROMAN'96*. Tsukuba, Japan.

[25] Shin, C.W. & Inokushi, S. (1994) A New Retina-Like Visual Sensor Performing the Polar Transform. *IAPR Workshop on Machine Vision Applications*. Kawasaki, Japan, 52-56.

[26] Rojer, A.S. & Schwartz, E.L. (1990) Design Considerations for a Space-Variant Visual Sensor with Complex-Logarithmic Geometry. *Int'l Conf. Pattern Recognition*. Atlantic City, New Jersey, 278-285.

[27] Bolduc, M. & Levine, M.D. (1997) A Real-Time Foveated Sensor with Overlapping Receptive Fields. *Real-Time Imaging*. **3**:195-212.

[28] Jerri, A.J. (1977) The Shannon Sampling Theorem - Its Various Extensions and Applications: A tutorial Review. *Proceedings of the IEEE*. **65**(11):1565-1596.

[29] Stark, H. (1979) Sampling Theorems in Polar Coordinates. *J. Opt. Soc. Am.*. **69**(11): 1519-1525.

[30] Lewitt, R.M. (1983) Reconstruction Algorithms: Transform Methods. *Proceedings of the IEEE*. **71**(3):390–408.

[31] Derrode, S. & Ghorbel, F. (2001) Robust and Efficient Fourier-Mellin Transform Approximations for Gray-Level Image Reconstruction and Complete Invariant Description. *Computer Vision and Image Understanding*. **83**:57-78.

[32] Feichtinger, H.G., Grochenig, K. & Strohmer, T. (1995) Efficient Numerical Methods in Non-Uniform Sampling Theory. *Numerische Mathematik*. **69**:423-440.

[33] Bonmassar, G. & Schwartz, E.L. (1997) Space-Variant Fourier Analysis: The Exponential Chirp Transform. *IEEE Trans. Pattern Analysis Machine Intell.* **19**(10):1080-1089.

[34] Kailath, T. (1967) The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Commun. Tech.* **COM-15**:52-60.

[35] Comaniciu, D. & Meer, P. (1999) Mean Shift Analysis and Applications. *IEEE Int'l Conf. Comp. Vis.* Kerkyra, Greece, 1197-1203.

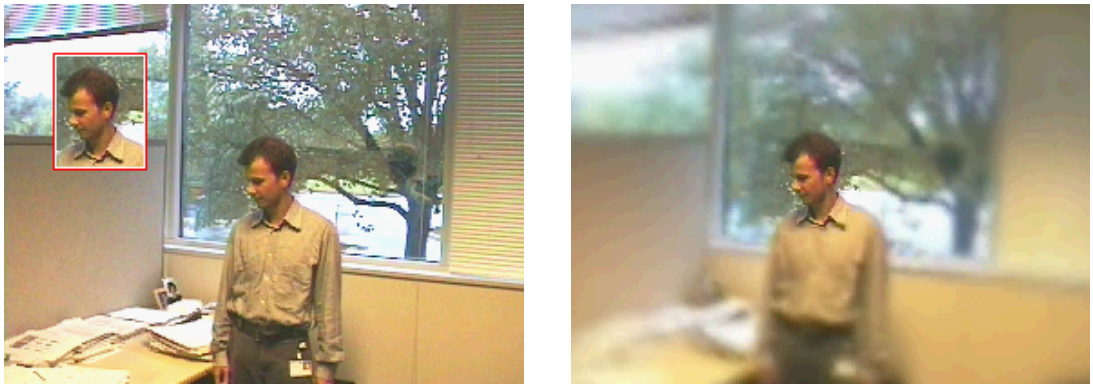
[36] Murray, D.W., Bradshaw, K.J., McLauchlan, P.F., Reid, I.D. & Sharkley, P.M. (1995) Driving Saccade to Pursuit using Image Motion. *Intern. J. Comp. Vis.* **16**(3):204-228.

[37] Rotstein, H.P. & Rivlin, E. (1996) Optimal Servoing for Active Foveated Vision. *IEEE Conf. on Comp. Vis. and Pat. Rec.* San Francisco, 227-182.

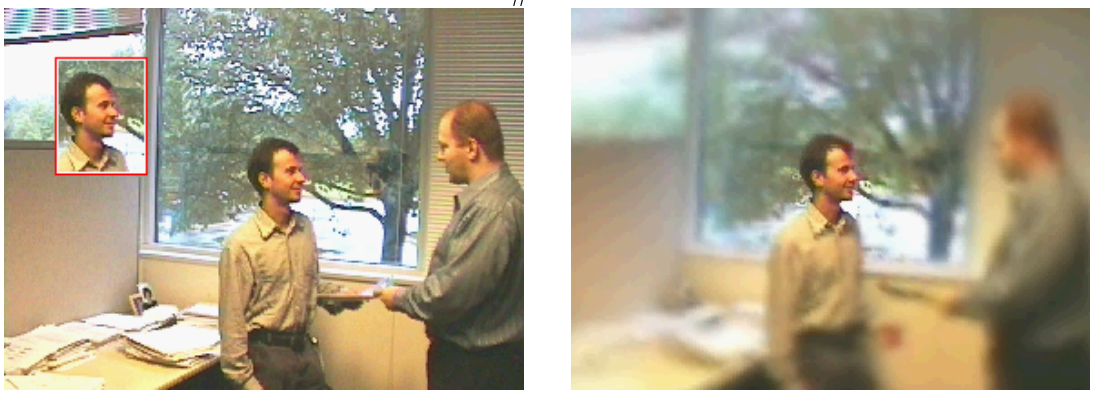
[38] Batista, J., Peixoto, P. & Araujo, H. (1998) Real-Time Active Visual Surveillance by Integrating Peripheral Motion Detection with Foveated Tracking. *IEEE Workshop on Visual Surveillance*. Bombay, India, 18–25.

[39] Moving Picture Experts Group (2000) Overview of the MPEG-4 Standard. *ISO/IEC JTC1/SC29/WG11*.

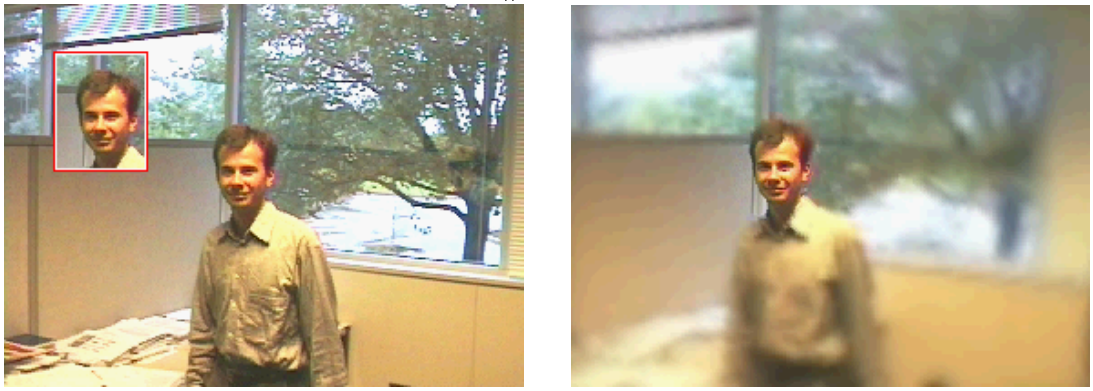
[40] Comaniciu, D. & Ramesh, V. (2000) Robust Detection and Tracking of Human Faces with An Active Camera. *IEEE Int'l Workshop on Visual Surveillance*. Dublin, Ireland, 11-18.



30



240



390

420

Figure 9: *Alessio* sequence. Left: original data. Right: received data. The frames 30, 240, 390, and 420 are shown.