A Novel Side Information Generation Algorithm in Distributed Multi-View Video Coding

Mohammad Haqqani, Kaamran Raahemifar, and Mahmood Fathy

***Abstract*—Inter-camera registration in multi-view systems with overlapped views has an especially long and sophisticated research history within the computer vision community. It represents a genuine challenge as a result of necessary data at decoder for generating the side information without the a priori knowledge of every instant camera position. This paper proposes a remedy to the problem centered on successive multi-view registration and motion compensated extrapolation for on-the-fly re-correlation of two views at decoder. This novel technique for side information generation is camera position independent, robust and flexible with regard to any free localization of the cameras. Furthermore, it doesn't require any additional information from encoders nor communication between cameras or offline training stage. Link between simulation demonstrate significant data compression in Wireless Sensor Networks, during keeping data quality.**

***Index Terms*—Distributed video coding, image mosaicing, side information generation.**

I. Introduction

The practical deployment of wireless sensor networks [1] and the option of small CMOS camera chips has held out the likelihood of populating the entire world with networked wireless video sensors. This type of setup may be used for a wide selection of applications, which range from surveillance to entertainment. As an example, a method endowed with multiple views can improve tracking performance by to be able to disambiguate the results of occlusion [2]. Free viewpoint TV and 3-D TV [3], [4] and tele-immersive applications may also take advantage of the easy deployment of dense networks of wireless cameras.

The applications described above, need certainly to depend on a strong infrastructure which can be effective at delivering accurate video streams from the wireless cameras. Unfortunately, this can be a rather challenging task. The wireless environment poses bandwidth constraints and channel loss, whilst the sensor mote platform has limited processing capability and limited battery life [1]. In applications such as for instance real-time surveillance, you will find very stringent end-to-end delay requirements, which impose tight latency constraints on the system.

Traditional hybrid video encoders such as for instance MPEGx and H.26x, while achieving high compression, have high encoder complexity due partly to the usage of motion compensation, and are prone to prediction mismatch, or "drift", in the clear presence of data loss. Such drift causes visually disturbing artifacts and is manufactured particularly worse in wireless channels where packet losses are bursty and more frequent than in wired networks. On another hand, Motion JPEG1 (MJPEG) is computationally light-weight and robust to channel loss, but has poor compression performance. Recent work with low-complexity video codecs using joint source-channel coding ideas centered on distributed source coding (DSC) principles supply a promising middle-ground involving the robustness and low encoding complexity of MJPEG and the compression efficiency of full-search motion-compensated MPEG/H.26x [5], [6].

We shall discuss our proposed method of robust and distributed multi-view video compression third chapter. In the next chapter, we review the relevant background topics that are essential in presenting our approach: distributed source coding (DSC), epipolar geometry and disparity estimation and compensation.

Manuscript received August 26, 2013; revised March 31, 2014.

M. Haqqani is with the Computer Engineering Department, Shiraz University, Shiraz, Iran (e-mail: mohammad.haqqani@gmail.com).

K. Raahemifar is with the Electrical and Computer Eng. Department, Ryerson University, Toronto, Canada (e-mail: kraahemi@ee.ryerson.ca).

M. Fathy is with the Computer Engineering Department, Iran University of Science and Technology, Tehran, Iran (e-mail: mahfathy@iust.ac.ir).

II. Literature Overview

A. Distributed Source Coding

Allow distributed coding of physically separated sources, we count on and are inspired by both information-theoretic and practical results in a specific setup of distributed source coding: lossy source coding with side-information, illustrated in Figure1. In a video coding context, X_n is the present video block to be encoded, and Y_n is the greatest predictor for X_n from reconstructions of reference frames such as for example temporally neighboring frames or spatially neighboring camera views. $\{ X_i , Y_i \}$ i=1 are *i.i.d*. with known joint probability distribution $p(x, y)$, and $X^{\wedge}_n$ is the decoder reconstruction of X_n. The objective is to recover $X^{\wedge}_n$ to within distortion D for some per-letter distortion $d(x, x^{\wedge})$. Note that in the setup, Y_n is only available at the decoder.

In case when X and Y are jointly Gaussian and the distortion measure could be the mean square error (MSE), it could be shown utilizing the Wyner-Ziv theorem [7] that the rate- distortion performance of coding X_n is the exact same whether Y_n can be acquired at the encoder. That is also true when $X_n \equiv Y_n + N_n$, with N_n being *i.i.d*. Gaussian and the distortion measure being the MSE [8]. However, generally, there's a tiny loss in rate-distortion performance, termed the

DOI: 10.7763/IJIEE.2014.V4.448

Wyner-Ziv rate loss, when correlated side-information isn't offered at the encoder [9].

While the above mentioned answers are non-constructive and asymptotic in nature, a functional approach was proposed by Pradhan and Ramchandran [10] and subsequently placed on video coding [5], [6].

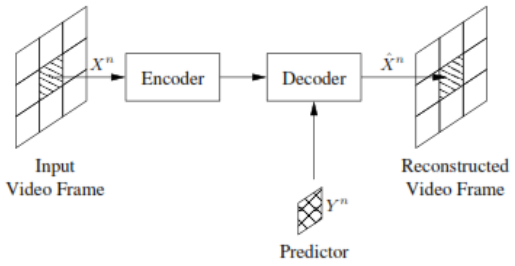


Fig. 1. Source coding models. DSC model, where side-information Y^n is available only at the decoder.

B. Epipolar Geometry

The geometric constraint of just one point imaged in two views, utilizing the projective camera (pinhole camera) model, is governed by epipolar geometry [11]. As shown in Fig. 2, given a graphic point in the initial view, the corresponding point in the 2nd view may be located on the epipolar line if it's not occluded in the 2nd view. Furthermore, the epipolar line may be computed from the career of the purpose in the initial view and the projection matrices of the cameras, and is independent of the scene geometry. Therefore, if the cameras are assumed to be stationary, then it's only essential to calibrate the cameras once at the start to acquire the fundamental matrix required for computation of the epipolar line between both views [11].

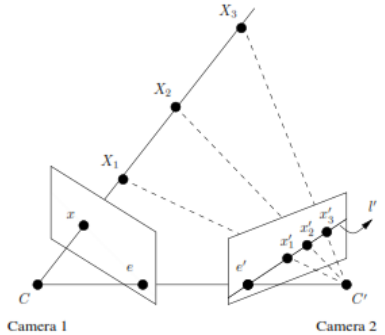


Fig. 2. Epipolar geometry [11].

Cameras 1 and 2 have camera centers at *C* and *C*' respectively. An epipole is the projected image of a camera center in the other view; c and c' are the epipoles in this diagram. A point x seen in the image plane of camera 1 (assuming a projective camera) could be the image of any point along the ray connecting C and x, such as X_1, X_2 or X_3. This ray projects to the epipolar line *l'* in camera 2. *L*' represents the set of all possible point correspondences for x. If *x* was the image of X_2, then the corresponding image point in Camera 2 would be X_2.

C. Disparity Estimation and Compensation

Disparity identifies the shift in horizontal locations of a corresponding point imaged in two rectified views; in Fig. 3, the disparity of point U imaged in camera 1 regarding camera 2 is merely $u_1 - u_2$. The depth of a place is inversely proportional to its disparity; small the disparity, the farther away the point. Disparity estimation, or stereo correspondence, is just a problem in computer vision that is worried with computing a thick disparity map from two rectified stereo images under known camera geometry. From the disparity map, a member of family depth map representing scene geometry could be computed. Among other activities, depth maps can be utilized for disparity compensation as discussed later. Further discussion of disparity estimation is away from scope with this dissertation, but a great survey and taxonomy of disparity estimation algorithms has been presented by Scharstein and Szeliski [12].

In computer graphics, both view synthesis and image based rendering involve solving the situation of using a couple of captured images from calibrated cameras to generate a picture that could have been captured by way of a camera at an ideal viewpoint. For the synthesized image to be reasonably accurate, the required viewpoint ought to be close to the capturing viewpoints. If the camera views are rectified, the other also can use disparity compensation as a view synthesis method of predict an ideal view.

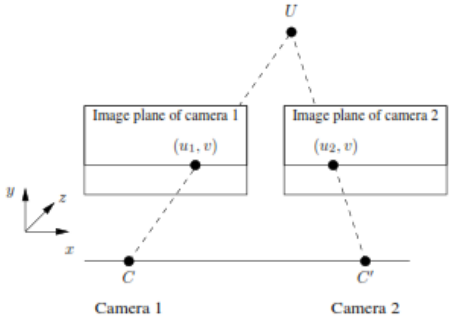


Fig. 3. Parallel cameras setup.

Cameras 1 and 2 have camera centers at *C* and *C*' respectively, whose displacement is parallel to the x-axis. The image planes are parallel to the *x*-*y* plane, and the camera axis is parallel to the z-axis. In this case, the epipoles lie at infinity. A point at (u_1; *v*) in the image plane of camera 1 would have a corresponding image point at (u_2; *v*) in camera 2.

III. Proposed Method

Since in the sensor networks, the majority of the covered spaces of the video sensors are typical, we choose a few of the cameras that may provide the complete scene of the network, altogether. We know these cameras as key cameras and we obtain them at first step of the algorithm. This way only a few cameras (the key cameras) send their data completely to the decoder and the residual (non-key cameras) are exempted from sending images and produce parity bites and send them. This selection will undoubtedly be done in ways that the non-key cameras' images overlap completely with total spaces of the main cameras. Then, a mechanism will be applied that even the key cameras send just areas of the images and will be exempted from sending some areas of

their frame which sent by other key cameras to the decoder. We construct the total space covered by the key cameras by creation of a mosaic image. For this matter, after sending the very first frame by each camera, a mosaic image will be created that determine common dispatched areas of the key cameras and decide centered on a unique mechanism, what part of frame should be send by the key cameras to be able to compute whole reference frame together. Therefore, utilizing the suggested algorithm, the dispatched data by the existing cameras in the network will undoubtedly be decreased and longevity of the network will increase. In follow, the steps of the algorithm are given in detail.

A. Selecting Key Cameras

In the initial step, all of the cameras code the initial frames by H.264 method and send it to decoder. A few of the cameras are selected as key cameras to make mosaic image, on the basis of the algorithm in [13]. The algorithm functions in conclusion: first camera is appointed as a key camera and its common image is likely to be calculated by other cameras. This is completed by corners estimation of frame. If the calculated overlap is higher than specified threshold, we refer to another camera; otherwise, it is likely to be selected as a key camera. In this defined threshold, the key cameras and corners of the mosaic image are likely to be up dated. This method is repeated for most of the cameras.

B. Generating Reference Frame using Mosaicking Algorithm

After sending the first frame to decoder by all of the video sensors, and determination of the key cameras, now total scene space, covered by all of the key cameras, is created by mosaicking algorithm (Fig. 4 (left)); mosaicking process fulfill this in five steps. At first, it extracts image's feature points by SIFT algorithm [14]; then, matches the points of each pair of images by Cross Correlation or Nearest Neighbor techniques [15]. While, some of the conformities are incorrect, in the next step, the incorrect conformities will be omitted by a conformity estimation model. Applying correct conformities, homograph matrix will be created for the images. The homograph matrix is one that maps the most points of the first image to the corresponding points of the second image. In the last step of mosaicking process, using homograph matrix, every image will be transmitted and placed in the mosaic image to build the final mosaic image. Meanwhile, we keep quadrangle coordinates of the images that are matching to the mosaic image, towards total scales. In this stage, before placing each image in the mosaic, we reach a filter image for it that include the space with image in order to omit effects of rotation and changing scale of the image by the filter (Fig. 5 (left)).

C. Determine the Area of Non-Key Cameras in Reference Frame

While, in determination of the key cameras, selection is completed in ways that the residual cameras (non-key cameras) have complete overlap with the produced reference frame. So, we give a mechanism to obtain the space that the non-key cameras are covered in the reference frame. The objective is using these windows as side information for non-key cameras. The non-key cameras just encode their information by producing and sending parity bits to the decoder. We apply these steps for extraction of the related window to each non-key camera:

- First, we find the 2 extreme SIFT points in the non-key frame and its equal in the reference frame (points with the greatest distance from each other's, horizontally and vertically)
- Then, we obtain the rectangle of the 2 points in a non-key frame and reference frame.
- We calculated distance of every edge of the rectangle to the nearest two sides of the non-key frame and ratio of distances alongside them.
- Now, applying the ratios across the rectangle, in the reference frame, we reach a screen across the rectangle that does add up to the related non-key frame.
- We repeat the prior two stages for all edges.
- We perform it for every one of the non-key cameras towards reference frame to be able to obtain related window of the non-key cameras (Fig. 4).

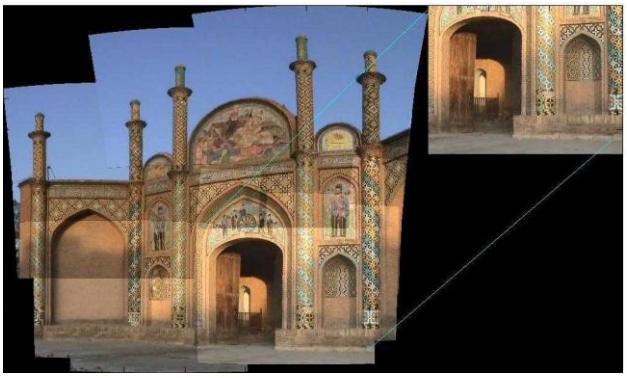


Fig. 4. Mosaicked frame (left) and non-key camera frame (right).

D. Dispatch and Decode Key Cameras Information

Here, we provide a solution, by it in sending the second frame, each key camera only sends parts of the image that is designated to it; meaning, the parts that will not be send by the other cameras in future. For this purpose, we assign the images, prepared by each camera, to it that have no similarity with other key cameras by using images' coordinates in the reference frame. About common spaces that are covered by two or more cameras, we will divide the spaces between the cameras in a way that the assigned measuring to the key cameras towards whole of the frame, and were adjusted in the network (Fig. 6). Thus, in the next steps, sending is done for the second subsequent frames; each key camera will be responsible for sending its frame that is assigned to it (Fig. 7). Now, to inform each key camera about the assigned space, we have created a mask for each camera and multiply it to the reverse of homograph matrix and mosaic image in order to obtain a mask equal to the scale of the main frame, as it was before placing the image in the mosaic image (Fig. 5 (right)). Then, we divide the mask to 8×8 blocks and sent towards camera to do sending based on the blocks. Before sending each frame, the camera multiplies it to the mask and sends. Thus, the assigned blocks to the camera will be dent to the decoder. Each key camera, moreover to the assigned blocks, produces several parity bites for the blocks that is exempted from sending, and provides it to the decoder.

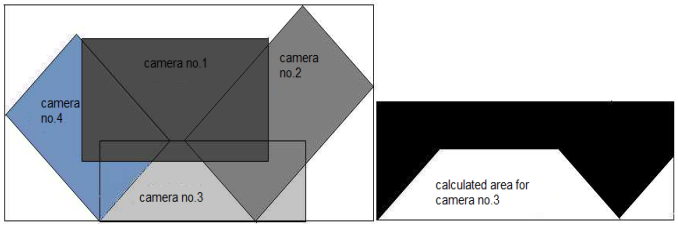


Fig. 5. Presumed Reference frame(left) and subpart frame to dispatch to a key-camera(right).

For the second frame and the subsequent frames, which the key cameras send only the assigned parts and the non-key cameras send parity bites, based on the mentioned part, the only action of the decoder for mosaic image, is only multiplying the frame to the related homograph matrix and put it in the reference frame, so there is no need to do all of the stages of mosaicking process. Up to now, we consider data transmission and creation of mosaic image. Now, we will discuss about receiving and decoding of information by the decoder. In a decoder, every stage's obtained mosaic is used as a tool for compression and data images of the key-cameras, particularly, data that are not sent by the camera, are extracted from the reference frame. It means that not sent data of each camera will be calculated by sending data of the other cameras that are common. To do this, we consider the following steps.

- We extract images' coordinators of each key camera from the mosaic image, which is reserved in the fifth stage of mosaicking algorithm and multiply the extracted space for each key camera that is created in a mask in the last stage of mosaicking, in order to return to the first state, in case that it was rotated in the mosaic and/ or its scale was changed.
- We multiply the result of previous stage to the reverse of related homograph matrix of this frame and the reference frame to obtain the main frame.
- While, some parts of the frames are sent by the this camera and the other parts by the other key cameras, we separate the part that the camera was exempted from sending it and provide it to the decoder as side information. After decoding, this part will be added to the main part and frame of the camera is decoded, with precision near to the key frame. We will do this for all of the key cameras.
- About the non key cameras, as it was mentioned previously, we multiply the window to reverse related homograph matrix, after extraction of the window coordinators; then, provide the result together with parity bites of that non key camera to the decoder in order to after decoding, related frame of that non key camera will be obtained.

Therefore, in this algorithm not only energy consumption reduced by exempting a lot of cameras from sending their frames, but additionally the residual cameras will lead to sending parts of their frames. Based on this sort of data compression, that will be followed closely by significant decrease of energy consumption, this algorithm could increase longevity of sensor network significantly, and it does not need complex calculations, and adjust sensors' energy limitation. Outcomes of the algorithm simulation will be provided within the next section, which will be taking into consideration the progression clearly.

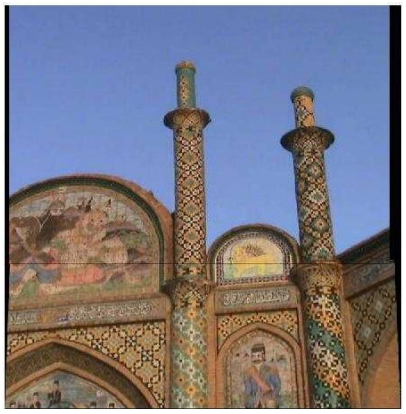


Fig. 6. Mosaiked frame of first and second key cameras.

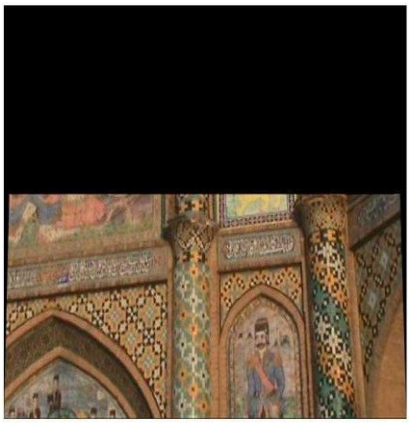


Fig. 7. Allocated part to dispatch for the first key camera.

IV. Simulation Results

For evaluating the proposed method we've used a data set contains of 112 cameras that covered a mosque. From these cameras the key camera selection algorithm selected 12 cameras from them and another hundred is recognized as non-key cameras. Once we discussed earlier, the key cameras only send the part of frame that's essential for generating the reference frame, Fig. 8 shows the source rate of key cameras for the very first frame. As you will see the cameras sent only 40% of their frame in average.

As we talked about before, the non-key cameras are exempted from dispatching their frame and only generate and send parity bits to the decoder. Fig. 9 shows the PSNR ratio for generated side information for the other 100 non-key cameras. As you can see the average PSNR for generated side information is above 30 that is a very significant PSNR of side information, and you have to consider that we gain this PSNR only using the 12 key cameras information.

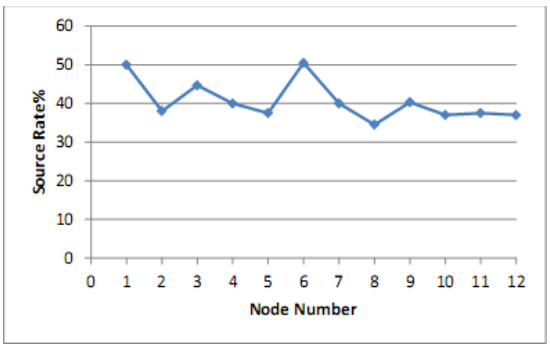


Fig. 8. Dispatch source rate of key – cameras.

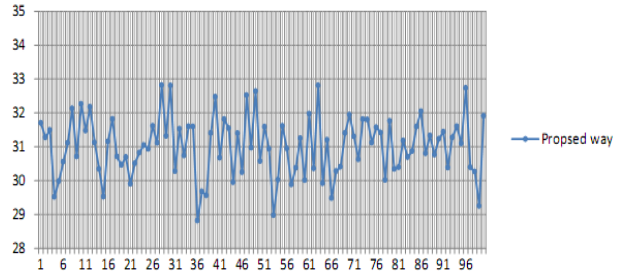


Fig. 9. PSNR of generated side information for the first frame of nonkey cameras.

V. Conclusion

We presented an algorithm which reduces transferring video information volume in multi-cameras networks by using Distributed Video Coding. Proposed algorithm divides the cameras in the network into two sets: key and non-key cameras. The algorithm not only exempted the non-key cameras from sending their frames to the decoder, but calculates the important region for key cameras too.

References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM International Workshop on Wireless sensor networks and applications*, 2002, pp. 88-97.

[2] S. L. Dockstader and A. M. Tekalp, "Multiple camera tracking of interacting and occluded human motion," in *Proc. the IEEE*, vol. 89, no. 10, Oct 2001, pp. 1441-1455.

[3] W. Matusik and H. Pester, "3D TV: a scalable system for real-time acquisition, transmission, and auto stereoscopic display of dynamic scenes," *ACM Transactions on Graphics*, vol. 23, no. 3, Aug 2004, pp. 814-824.

[4] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "Multiview imaging and 3DTV," *IEEE Signal Processing Magazine*, vol. 24, no. 6, Nov. 2007, pp. 10-21.

[5] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," presented at Allerton Conference on Communication, Control and Computing, 2002.

[6] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proc. Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 1, 2002.

[7] A. D. Wyner and J. Ziv, "The rate distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, Jan 1976, pp. 1-10.

[8] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Transactions on Information Theory*, vol. 49, no. 7, July 2003, pp. 1181-2003.

[9] R. Zamir, "The rate-loss in the Wyner-Ziv problem," *IEEE Transactions on Information Theory*, vol. 42, no. 11, Nov 1996, pp. 2073-2084.

[10] S. S. Pradhan, K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, Mar 2003, pp. 626-643.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

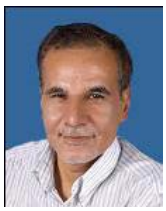
[12] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, 2002, pp. 7-42.

[13] M. J. Fadaeieslam, M. Soryani, and M. Fathy, "Efficient key frames selection for panorama generation from video," *Journal of Electronic Imaging*, vol. 20, no. 023015, 2011.

[14] D. G. Lowe. "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2-91, November 2004.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, 1981, pp. 381-395.

Mohammad Haqqani received the B.S. and M.S degrees from Buali Sina Uniersity (BASU), Hamedan, Iran, and Iran University of Science and Technology (IUST) Tehran, Iran, in 2009 and 2011, respectively, all in computer Engineering. He is currently a PhD student in Shiraz University, Shiraz, Iran. His research interests include video coding and streaming, image processing, and Multi-Objective Optimization.



Mahmood Fathy received his B.Sc. degree from Iran University of Science & Technology, Tehran, Iran, in 1984, his M.Sc. degree from Bradford University, West Yorkshire, U.K., in 1987, and his Ph.D. degree in image processing and computer architecture from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1991. Since 1991, he has been an associate Professor WITH the Department of Computer Engineering, Iran University of Science & Technology. His research interests include the quality of service in computer networks, the applications of vehicular ad hoc networks in intelligent transportation systems, and real-time image processing, with particular interest in traffic engineering, bio informatics, and bio computers.

Kaamran Raahemifar received his B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1988, his M.Sc. degree from Waterloo University, Waterloo, Ontario, Canada in 1993, and his Ph.D. degree from Windsor University, Ontario, Canada, in 1999. Since 2002, he has been an associate professor with the Department of Electrical and Computer Engineering, Ryerson University. His research interests include VLSI Circuit Simulation, Design, and Testing, Signal Processing and Hardware Implementation of Biomedical Signals.