A Stable Vision System for Moving Vehicles

Jesse S. Jin, Member, IEEE, Zhigang Zhu, Member, IEEE, and Guangyou Xu, Senior Member, IEEE

Abstract—This paper presents a novel approach to stabilize the output of video camera installed on a moving vehicle in a rugged environment. A 2.5-D interframe motion model is proposed so that the stabilization system can perform in situations where significant depth changes are present and the camera has both rotation and translation. Inertial motion filtering is proposed in order to eliminate the vibration of the video sequences with enhanced perceptual properties. The implementation of this new approach integrates four modules: pyramid-based motion detection, motion identification and 2.5-D motion parameter estimation, inertial motion filtering, and affine-based motion compensation. The stabilization system can smooth unwanted vibrations or shakes of video sequences and achieve real-time speed. We test the system on IBM PC compatible machines and the experimental results show that our algorithm outperforms many algorithms which require parallel pipeline image processing machines.

Index Terms—Inertial model, motion estimation, motion filtering, tele-operation, video stabilization.

I. INTRODUCTION

The goal of video stabilization is to remove unwanted motion from dynamic video sequences. It plays an important role in many vision tasks such as tele-operation, robot navigation, ego-motion recovery, scene modeling, video compression, and detection of independent moving objects [1]–[6], [9], [10], [13]. There are two critical issues which decide the performance of a digital video-stabilization system. One is interframe motion detection. The other is motion smoothing.

The first issue depends on what kind of motion model is selected and how image motion is detected. It has been pointed out [1], [5] that human observers are more sensitive to rotational vibrations and only camera rotation can be compensated without the necessity of recovering depth information. Hansen et al. [2] developed an affine model to estimate the motion of an image sequence. The model results in large errors for rotational estimations if there are large depth variations within images and the translation of the camera is not very small. Duric et al. [5] and Yao et al. [6] dealt with this problem by detecting far-away-horizon lines in images since the motion of horizon lines is not affected by small translation of video camera. They assumed that long straight horizon line would exist in common outdoor images and it will have large gradients in gray-scale images. This assumption works only if there is a horizon line and its distance is further away than the average distance in the scene. It will fail in many cases where the horizon line is not very clear or just cannot be seen, or the points along the horizon line are not so far away.

The second issue is critical for eliminating unwanted vibrations while preserving smooth motion. Hansen et al. [2] first choose a reference frame, then align successive frames to the reference by warping them according to the estimated interframe motion parameters. The reconstructed mosaic image is the result of stabilization. This approach works fine if there are no significant depth changes in the field of view or the moving distance of the camera is not long. This approach fails in cases involving panning or tracking motion, where accumulated errors override the real interframe motion parameters. Duric and Rosenfeld [5] assume camera carriers move steadily between deliberate turns, thus they stabilize the video sequence by fitting the curves of motion parameters with linear segments. The resulting sequences are steady during each line segment but the moving speed changes suddenly between two segments. Another problem with these approaches is that the image sequence is delayed for several frames in order to get a better line fitting. Yao et al. [6] proposed a kinetic vehicle model for separating residual oscillation and the smooth motion of a vehicle (and camera). This model works well when the kinetic parameters of the camera’s carrier are known and the camera is firmly bound to the vehicle. Kinetic parameters include the vehicle’s mass, the characteristics of the suspension system, the distance between the front and rear ends, and the center of gravity. In most cases, either we do not know kinetic parameters, the camera carrier is not a grounded vehicle, or the camera is not firmly bound to the carrier.

Real-time performance is the basic requirement for video stabilizers. There is much research on video stabilization for military and tele-operation utilities, as well as application in commercial palmcorders. Hansen et al. [2] implemented an image stabilization system based on a parallel pyramidal hardware (VFE-100) which is capable of estimating motion flow in a coarse-to-fine multiresolution way. The motion parameters are computed by fitting an affine motion model to the motion flow. The stabilized output is given in the form of a registration-based mosaic of input images. Their system stabilized $128 \times 120$ images at 10 frames per second (fps) with displacement of up to $\pm 32$ pixels. Morimoto et al. [3] gave an implementation of a similar method on Datacube Max Video 200, a parallel pipeline image processing machine. Their prototype system achieves 10.5 fps with an image size of $128 \times 120$ and maximum displacement of up to $\pm 35$ pixels. Zhu et al. [1] implemented a fast stabilization algorithm on a parallel pipeline image processing machine called PIPE. The whole image is divided into four regions and several feature points are
selected within these regions. Each selected point is searched for in the successive frame in order to find a best match using a correlation method. The detected motion vectors are combined in each region to obtain a corresponding translation vector. A median filter is used to fuse the motion vectors into a plausible motion vector for the entire image. A low-pass filter was then used to remove the high-frequency vibrations of the sequence of motion parameters. The system runs at 30 fps with image size $256 \times 256$ pixels and can handle a maximum displacement of $-8$ to $+7$ pixels.

In this paper, we proposed a new digital video stabilization approach based on a 2.5-D motion model with inertial filtering. A prototype has been developed, which includes block-based motion estimation, 2.5-D affine parameter estimation, the linear Householder transformation, and recursive vibration filtering. Our system achieves 26.3 fps on an IBM PC compatible using an Intel Pentium 133-MHz microprocessor without any image processing acceleration hardware. The image size is $128 \times 120$ with a maximum displacement of $\pm 32$ pixels. The frame rate is 7.4 fps with a maximum displacement of up to $\pm 64$ pixels and an image size of $256 \times 256$.

II. INTERFRAME MOTION MODEL

Assume that the camera is viewing a static scene and all motions in the image are caused by the movement of the camera. A reference coordinate system $O-XYZ$ is defined for the moving camera and the optical center of the camera is $O$ (see Fig. 1). $W-UV$ is the image coordinate system, which is the projection of $O-XYZ$ onto the image plane. The camera motion has six degrees of freedom: three translation components and three rotation components. In other words, it can be interpreted that the viewed scene has six motion parameters, since the camera is the reference. Considering only the interframe case, represent the three rotational angles (roll, pitch, and yaw) by $\alpha, \beta, \gamma$ and the three translations as $(T_x, T_y, T_z)$. A point in space $(x, y, z)$ with an image coordinate $(u, v)$ will move to $(x', y', z')$ in the next frame. Suppose the camera focal length is $f'$ after motion. Under a pinhole camera model, the relations between these coordinates are [15]

$$
\begin{bmatrix}
  x' \\
  y' \\
  z'
\end{bmatrix}
= \begin{bmatrix}
  a & b & c \\
  i & j & k \\
  l & m & n
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix}
- \begin{bmatrix}
  T_x \\
  T_y \\
  T_z
\end{bmatrix}
$$

If the rotation angle is less than $5^\circ$, (1) can be approximated as

$$
\begin{cases}
  u' \approx f' \frac{u + \alpha v - \gamma f - f T_z/z}{\gamma u - \beta v + f - f T_z/z} \\
  v' \approx f' \frac{-\alpha u + v + \beta f - f T_y/z}{\gamma u - \beta v + f - f T_z/z}
\end{cases}
$$

Let

$$
\begin{align}
  s &= (\gamma u - \beta v + f - f T_z/z)/f' \quad (3a) \\
  s \cdot u' &= u + \alpha v - \gamma f - f T_z/z \\
  s \cdot v' &= -\alpha u + v + \beta f - f T_y/z. \quad (3b)
\end{align}
$$

A. Affine Motion Model

From (2) and (3b) we have the following observations.

1) **Pure rotation.** If the translations of the camera are zero, the images will only be affected by rotation and the effect is independent of the depths of scene points. When the rotation angle is very small (e.g., less than $5^\circ$), camera roll is the main factor of image rotation, and the effect of the pitch and yaw is reflected by 2-D translations of the image.

2) **Small translation and constrained scene.** If image points are of the same depth, or the depth difference among image points is much less than the average depth, small camera translation will mainly cause homogeneous scaling and translation of 2-D images.

3) **Zooming.** Changing camera focal length will only affect the scale of images.

4) **General motion.** If depths of image points vary significantly, even small translations of the camera will make different image points move in quite different ways.

For cases 1), 2), and 3), a 2-D affine (rigid) interframe motion model can be used

$$
\begin{align}
  s \cdot u' &= u + \alpha v - T_u \\
  s \cdot v' &= v - \alpha u - T_v
\end{align}
$$

where $s$ is scale factor, $(T_u, T_v)$ is the translation vector, and $\alpha$ is the rotation angle. Given more than three pairs of corresponding points between two frames, we can obtain the least-squares solution of motion parameters in (4). Since this model is simple and works in various situations, it has been widely adapted [1]–[6].

B. The 2.5-D Motion Model

Although the affine motion model is simple in calculation, it is not appropriate for the general case (see case 4), where depths of scene points change significantly and the camera movement is not very small. It is impossible to estimate motion without depth information. However, recovering surface depth in a 3-D model is too complicated and depth perception itself is a difficult research issue. We aim at a compromise solution involving depth at estimation points only.

Careful analysis of (1)–(3b) reveals that

1) **Dolly movement:** If the camera moves mainly in the $Z$ direction, the scale parameters of different image points depends on their depths [see (3a)].
2) **Tracking movement**: If the camera moves mainly in the X direction, different image points will have different horizontal translation parameters relating their depths. Similarly, if the camera moves mainly in the Y direction, different image points will have different vertical translation parameters corresponding to their depths [see (3b)].

3) **Rotational/zooming movement**: Panning, tilting, rolling, and zooming, and their combinations generate homogeneous image motion [see (3b)].

4) **Translational movement**: If the camera does not rotate, there exists a direct solution of the relative camera translation parameters and scene depths [see (1) or (2)].

5) **General movement**: If camera moves in arbitrary directions, there is no direct linear solution of these motion parameters. This is the classic “structure from motion (SfM)” problem [7]–[9] and will not be addressed here.

Cases 1), 2), and 3) cover the most common camera motions: dolly (facing front), tracking (facing side), panning/tilting/rolling (rotation), and zooming movement. A 2.5-D interframe motion model is proposed by introducing a depth-related parameter for each point. It is an alternative between 2-D affine motion and the real 3-D model.

The model consists of three parts. For dolly movement (involving rotation and zooming, i.e., case 1) with case 3)), we have $T_x \approx 0, T_y \approx 0$, hence the 2.5-D dolly model is

$$\begin{align*}
\begin{cases}
 s_i \cdot u'_i = u_i + \alpha v_i - T_u \\
 s_i \cdot v'_i = v_i - \alpha u_i - T_v
\end{cases} \\
(i = 1, \ldots, N),
\end{align*}$$  
(5)

For horizontal tracking movement (involving panning, rolling, and zooming), we have $T'_x \approx 0, T'_y \approx 0$. The 2.5-D H-tracking model is

$$\begin{align*}
\begin{cases}
 s \cdot u'_i = u_i + \alpha v_i - T_{ui} \\
 s \cdot v'_i = v_i - \alpha u_i - T_{vi}
\end{cases} \\
(i = 1, \ldots, N),
\end{align*}$$  
(6a)

Similarly, for vertical tracking movement (involving tilting, rolling, and zooming), we have $T'_x \approx 0, T'_y \approx 0$. The 2.5-D V-tracking model is

$$\begin{align*}
\begin{cases}
 s \cdot u'_i = u_i + \alpha v_i - T_{ui} \\
 s \cdot v'_i = v_i - \alpha u_i - T_{vi}
\end{cases} \\
(i = 1, \ldots, N),
\end{align*}$$  
(6b)

In most situations, the dominant motion of the camera satisfies one of the 2.5-D motion models. It should be noted that in cases (5), (6a), and (6b) there are $N + 3$ variables in $2N$ equations. Different $N$ variables contribute to the solution in each case, i.e., scale $s_i$ for depth-related parameters, $T_{ui}$ for horizontal translation, and $T_{vi}$ for vertical translation, respectively.

III. M O T I O N D E T E C T I O N A N D C L A S S I F I C A T I O N

A. Numerical Method for Parameter Estimation

If there are more than three ($N \geq 3$) correspondences between two frames, the least-squares solution can be obtained for any of the three cases mentioned above. Since the forms of the equations are similar in all three cases, the 2.5-D dolly motion model is used as an example. Rearranging the order of the $2N$ equations, (5) may be expressed in the following augmented matrix form:

$$Ax = b$$  
(7a)

where

$$A = \begin{bmatrix}
u_1 & 0 & \cdots & 0 & -u_1 & 1 & 0 \\
0 & u_2 & \cdots & 0 & -v_2 & 1 & 0 \\
0 & 0 & \cdots & u_N & -v_N & 1 & 0 \\
v_1 & 0 & \cdots & 0 & u_1 & 0 & 1 \\
0 & 0 & \cdots & 0 & u_N & 0 & 1
\end{bmatrix}$$

We can solve (7a) by a QR decomposition using the Householder transformation. Due to the special form of matrix $A$, the Householder transformation can be realized efficiently: The first $N$ columns only require $(N + 1) \times 8$ multiplications and $N$ square root operations; then the Householder transformation is carried out on an $N \times 3$ dense submatrix. The interim result of $A$ can be stored in a $2N \times 4$ array. This approach requires less computation than that of the 2-D affine model, with a $2N \times 4$ dense matrix. This method is also more numerically stable than the affine method since in most cases it has a condition number around 300, while that of the 2-D affine model is between 150 and 300. A detailed discussion and numerical testing of this model can be found in [12].

B. Motion Detection and Classification

Detection of the image velocities is the first step in motion estimation. Instead of building the registration between two frames for the distinguished features as in [6], the image velocities of the representative points are estimated by a pyramid-based matching algorithm applied to the original gray-scale images [16]. The algorithm consists of the following steps.

1) Construct the pyramids for the two consecutive images.
2) Evaluate the image velocity for each small block (e.g., $8 \times 8$ or $16 \times 16$) in the first frame using a coarse-to-fine correlation matching strategy.
3) Calculate the belief value of each match by combining the texture and the correlation measurements of the block.

This step is important because the value is used as a weight for motion parameter estimation.

Fig. 2 shows the result of rotation angle estimation for one of the testing sequences, where the 2-D affine model and the proposed 2.5-D dolly and 2.5-D H-tracking models are used for comparisons. The original sequence is a side view taken by a camera mounted on a forward-moving vehicle. The vibration is mainly camera rolling. The accumulated rolling angle (global rotation) is approximately zero. The motion satisfies the horizontal tracking model. The motion vectors in Fig. 2(a) show that the image velocities are quite different due to the different scene depths. The curves in Fig. 2(b) are the accumulated interframe rotation angles estimated by the three models. False global rotation appears in the affine model.
An important issue of the 2.5-D motion model is the correct selection of the motion model (dolly, H-tracking, or V-tracking). False global rotation is also derived if improper motion models are applied, as shown in Fig. 2(b), where the dolly model is an incorrect representation for the type of the movement. The classification of the image motion is vital. Simply, the motion model can be selected by an operator (person) since the qualitative model class can be easily identified by a human observer, and the type of motion within a camera snap usually lasts for a certain time period. The selection of motion models can also be made automatically by a computer. Careful investigations of the different cases of camera movements indicate that the motion can be classified by the patterns of image flows (or the accumulated flows). Dolly (and/or zooming), H-tracking (and/or panning), V-tracking (and/or tilting), and rolling movements result in four distinctive flow patterns [12]. It is possible to classify them automatically.

### IV. Motion Filtering

The second vital stage of video image stabilization is the elimination of unwanted high-frequency vibrations from the detected motion parameters. The differences between smoothed motion parameters and original estimated motion parameters can be used to compensate the images in order to obtain the stabilized sequence. Removing (high-frequency) vibrations is virtually a temporal signal filtering (motion filtering) process with the following special requirements.

1. The smoothed parameters should not be biased from the real gentle changes. Bias like phase shift and amplification attenuation may cause undesired results.
2. The smoothed parameters should comply with the physical regularities, otherwise the result will not satisfy human observers.

Based on the above considerations, a generic inertial motion model is proposed as the motion filtering model. This filter fits physical regularities well and has little phase shift over a large frequency range. The only requirement is to set some assumptions of the sequence’s dynamics. In the context of image stabilization, it should be made clear which kind of smooth motion needs to be preserved and what vibration (fluctuation) needs to be eliminated. There are some differences for each of the three kinds of motion in our model.

For the dolly movement, the smooth motion is mainly translation in the optical direction, possibly with deliberate zooming and rotation of one to three degrees of freedom. The fluctuations are mainly included in three motion parameters: $\alpha$, $T_w$, and $T_v$ [see (5)], which may be caused by the bumping and rolling of the carrier (vehicle). The depth-related scale parameter $s_i$ for each image block is related to the zooming and dolly movement and the depths of the scene. In order to separate the depth information from the motion parameter, we calculate the fourth motion parameter by

$$s = \sum_{i=1}^{N} s_i.$$  \hspace{1cm} (8)

The fluctuation caused by dolly motion and zooming will be extracted from this parameter.

For the H-tracking movement, the smooth translational component is included in the parameter $T_{hi}$ which is depth-related. Apart from the three affine parameters $s$, $\alpha$, and $T_v$ [see (6a)], we separate the depth information by averaging the horizontal components of all image blocks and obtain the fourth motion parameters by

$$T_{hi} = \sum_{i=1}^{N} T_{hi}.$$  \hspace{1cm} (9)

A more detailed discussion can be found in [13]. Similarly, the four motion parameters $s$, $\alpha$, $T_{hi}$, and $T_v$ can be obtained in the case of V-tracking movement [see (6b)].

Since the four motion parameters are independent of each other, we can treat them separately. For example, Fig. 3 is a schematic diagram of an inertial filter, where $\alpha(t)$ is carrier’s
displacement, $x(t)$ is camera’s displacement, $m$ is the mass of the camera, $k$ is elasticity of a spring, $p$ is the damping parameter, and $f(t)$ is a controllable feedback compensator. Assuming $f(t) = 0$, the kinematic equation can be written as

$$m \frac{d^2x}{dt^2} + p \left( \frac{dx}{dt} - \frac{dc}{dt} \right) + k(x - c) = 0,$$  \hspace{1cm} (10)

This second-order equation represents a device under forced oscillation. The intrinsic oscillation frequency is $f_0 = \frac{1}{2\pi} \sqrt{\frac{k}{m}}$, the damping ratio is $\xi = (\gamma T) / \sqrt{m k}$, and the relative frequency is $\lambda = (f/f_n)$. The frequency response can be expressed as

$$H(\lambda) = \sqrt{\frac{1 + 4\lambda^2}{(1 - \lambda^2)^2 + 4\xi^2\lambda^2}},$$  \hspace{1cm} (11)

When the sampling rate (i.e., frame rate) is much higher than $f_n$, we can directly write the discrete form of (10) as a differential equation

$$x(n) = a_1x(n-1) + a_2x(n-2) + b_1c(n) + b_2c(n-1) + f(n)$$  \hspace{1cm} (12)

with

$$a_1 = \frac{2m + pT}{m + pT + KT^2}, \quad a_2 = \frac{-m}{m + pT + KT^2},$$
$$b_1 = \frac{pT + KT^2}{m + pT + KT^2}, \quad b_2 = \frac{-pT}{m + pT + KT^2},$$

and where $f(n)$ is the feedback. This is a second-order digital filter. By choosing suitable values for these parameters, high-frequency vibrations can be eliminated. For example, if the frame frequency is 25 Hz and high-frequency vibrations have a frequency $f_r$ larger than 0.5 Hz, we can choose the termination frequency $f_T = 0.3$ Hz. Assuming that $k = 1$ and $\xi = 0.9$, we have $f_n = 0.1288$, $m = 1.5267$, and $p = 2.2241$. Compared to the second-order Butterworth low-pass filter (LPF) that has $f_T = 0.3$ Hz, the inertial filter has a similar terminating attribute but with much less phase delay. Note that $c(t)$ is the measurement and $x(t)$ is the filtering output. Fig. 4 shows the comparison of results.

For better tuning of the filter response and a reduction of the phase shift, proper feedback compensation is added. Negative feedback can reduce phase shift but will increase the bandwidth of the filter. An effective way is to use threshold-controlled feedback: if $|x(n) - c(n)|$ is greater than a predefined value, the feedback is turned on till the error value is less than a threshold. Fig. 5 is the experimental results on real video sequence images. The feedback function is

$$f(n) = -0.005(x(n) - c(n))$$

and the threshold is 8% of $|c(n)|$. The feedback weight and the threshold are not crucial and can be obtained from spectrum analysis of $c(n)$. The weight is usually between $-0.001$ and $-0.01$. The threshold is less than 10%.

V. EXPERIMENTAL RESULTS

We have implemented a system based on the 2.5-D motion model and inertial motion filtering. Our system consists of four modules as shown in Fig. 6. Feature matching is performed by multiresolution matching of small image blocks. Reliability of each matched block is evaluated by accumulating gray-scale gradient values and the sharpness of correlation peak. This is an indicator as areas without strong texture (such as sky, wall, and
road surface) may have large matching errors. The evaluation value of each image block is used as a weight of corresponding equation pairs (e.g., (5)) when they are used to obtain interframe motion parameters. Interframe motion parameters are calculated and accumulated as global motion parameters of the reference frame and are passed through the inertial filter. The differences between the original and filtered motion parameters are used to warp the current input frame to obtain the stabilized output.

Programs are written in C and have been executed on IBM PC compatibles with an Intel 80486DX2/66 MHz or an Intel Pentium/133 MHz microprocessor. The execution time for each module is listed in Table I. Table II is the comparison of the proposed implementation with existing systems. Fig. 7 shows images from several test sequences. Some stabilization results are available on the Internet [14].

<table>
<thead>
<tr>
<th>CPU</th>
<th>Image size (pixels)</th>
<th>Block size (pixels)</th>
<th>Match (ms)</th>
<th>Motion (ms)</th>
<th>Warp (ms)</th>
<th>Frame rate (frames/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>486/66</td>
<td>128×120</td>
<td>16×16</td>
<td>95</td>
<td>4</td>
<td>54</td>
<td>6.54</td>
</tr>
<tr>
<td>486/66</td>
<td>256×256</td>
<td>25×25</td>
<td>341</td>
<td>7</td>
<td>229</td>
<td>1.73</td>
</tr>
<tr>
<td>486/66</td>
<td>360×270</td>
<td>20×20</td>
<td>468</td>
<td>17</td>
<td>333</td>
<td>1.22</td>
</tr>
<tr>
<td>P5/133</td>
<td>128×120</td>
<td>16×16</td>
<td>25</td>
<td>1</td>
<td>12</td>
<td>26.32</td>
</tr>
<tr>
<td>P5/133</td>
<td>256×256</td>
<td>25×25</td>
<td>83</td>
<td>1</td>
<td>51</td>
<td>7.4</td>
</tr>
<tr>
<td>P5/133</td>
<td>360×270</td>
<td>20×20</td>
<td>122</td>
<td>3</td>
<td>77</td>
<td>4.95</td>
</tr>
</tbody>
</table>

VI. Conclusion

A new video image stabilization scheme is proposed which is featured by a 2.5-D interframe motion model and an inertial model based motion filtering method. The experimental system running on an Intel Pentium 133 MHz PC performs better than other existing systems that require special image processing hardware. Real-time processing has been achieved due to the simple motion parameters in the model, the fast Householder transform, and the recursive vibration filtering process. The method has been applied in several vision tasks, such as 3-D natural scene modeling and surveillance [13].

Fig. 7. Key frame of testing sequences. (a) Dolly movement with small vibration. (b) Tracking movement with rolling fluctuation. (c) Dolly movement with rotation. (d) Dolly movement with severe vibration.
Automatic selection of optimal interframe motion models is crucial and the best results may be achieved by applying different models in a certain order. This issue requires further investigation in future research.

**REFERENCES**


JIN et al.: A STABLE VISION SYSTEM FOR MOVING VEHICLES 39

Jesse Jin (M’98) received the Ph. D. degree from University of Otago, New Zealand. Currently, he is the Director of the Visual Information Processing Laboratory and the Head of Information Engineering Group in the School of Computer Science and Engineering, University of New South Wales, Sydney, Australia. He has received many grants from both academic sources and industry sources including Motorola, SiliconGraphics, CyberView, etc. His research interests include multimedia, the Internet, and image processing.

Zhigang Zhu (S’94–A’97–M’99) received the B.S., M.S., and Ph.D. degrees in computer science from Tsinghua University, China, in 1988, 1991, and 1997, respectively.

In 1991, he joined the faculty of the Department of Computer Science and Technology, Tsinghua University, where he is an Associate Professor. From 1997 to 1999 he was the Director of Information Processing and Application Division. He is also a Visiting Professor in the Computer Science Department, the University of Massachusetts at Amherst. His research interests include computer vision, image and video analysis, intelligent mobile robot, and virtual reality. He has published more than 70 technical papers in related fields.

Guangyu Xu (SM’91) graduated with the B. Sc.(Hons) degree from the Department of Control and Automation, Tsinghua University, in 1963.

He was a Visiting Scholar at Purdue University, West Lafayette, IN, working with Professor K. S. Fu from 1982 to 1984, and a Visiting Professor at Beckman Institute, University of Illinois at Urbana-Champaign from December 1994 to June 1995. Currently, he is a Chair Professor in the Department of Computer Science, Tsinghua University. He has received many research grants and is authors of many papers and four books. His research interests include computer vision and multimedia information systems.