ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

A Rain Pixel Recovery Algorithm for Videos

Bhuvaneshwari C.Melinamath Amruta S. Bandarwad

Assistant Prof. Dept. of Computer Science and Engineering, BLDEA,s CET, Vijaypur, India

Dept. of Computer Science and Engineering, BLDEA,s CET , Vijaypur, India

ABSTRACT: Rain removal is a highly useful and important technique in many applications such as surveillance and movie editing. In the recent years several rain removal algorithms have been proposed, where photometric, chromatic, and probabilistic properties of the rain pixels have been utilized to detect and remove the rainy effect. The existing methods generally work well with light rain and relatively static scenes, but when dealing with heavier rainfall in dynamic scenes in motion, these existing methods give very poor results. The proposed algorithm is based on the concept of motion segmentation of dynamic scene. After applying photometric and chromatic constraints for rain detection, several rain removal filters are applied on the rainy image such that their dynamic property as well as motion occlusion clues are taken into consideration; both spatial and temporal information are then adaptively exploited during rain pixel recovery. Extensive simulation results show that the proposed algorithm shows a much better performance for rainy scenes with large motion than the existing algorithms.

KEYWORDS: Motion segmentation, motion occlusion, dynamic scene, motion buffering, adaptive filters, rain removal, noise suppression.

I. INTRODUCTION

RAIN removal is a complex job. In rainy videos pixels show small but frequent strength ups and downs, and this unsteadiness could be caused by (more than two, and not a lot of) other reasons besides rain fall, namely, worldwide lighting up/education change, camera move, and object motion etc. In need to remove the rainy effect, it is necessary to detect the ups and downs and variations that are caused by rain, and then replace those variations with their original value. Some good sets of computer instructions have been proposed for this purpose.

II. RELATED WORK

Garg and Nayar first carefully studied the physical and photometric properties of the rain [1], [2], and they used their (instances such as watching, noticing, or making a statement) data to apply both strength and time-related restrictions to detect and then remove the rain. However, their ideas such as the uniform speeds and directions of the rain drops limited its performance. Zhang [13] proposed a method based on chromatic restriction [3], where they assume that strength changes in the R, G, B channels caused by rain are about the same, while those caused by object motion are (the left side different from the right side). This method is only related with static background, and it gives out faulty results for particular foreground colors.

Tripathi et al. [14] proposed a probabilistic spatial-temporal model [14], [15], in which they propose to use statistical features, such as intensity fluctuation range, and spread asymmetry, which are extracted from a spatial-temporal neighborhood to classify the rain affected pixels [3]. This method is more robust when dealing with dynamic scenes [6], however some statistical features it proposes [8] (i.e. spread asymmetry) works very poorly in many occasions, and it gives a lot of false detections [6]. Except from rain detection [5], one shared drawback of the existing methods [6] is that, in the prediction of the rain covered pixel's original value [3], where they simply compute its temporal mean. This causes serious corruptions in areas where obvious motion [5] is present; important information are erased [5]; and a totally undesired ghost effect is often seen. The proposed algorithm is based on motion segmentation [13] of dynamic scene [12]. After applying photometric and chromatic constraints [3] for rain detection, rain removal filters are applied [6] on pixels such that their dynamic property as well as motion occlusion [3] clue are considered [12]; both spatial and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

temporal information are then adaptively [8] exploited during rain pixel recovery [9]. Experimental results and simulations [8] show that our algorithm outperforms existing ones in highly dynamic scenarios [6].

III. MOTION SEGMENTATION

The pixel intensity fluctuation [5] of a rainy scene is caused by rain and object motion. The fluctuations [4] caused by rain need to be removed, and the ones caused by object motion [5] need to be retained. Thus motion field segmentation [2] naturally becomes a fundamental procedure [5] of our algorithm.

A. Motion Field Estimation

Motion field is a 2-D vector field [4] projected from the 3-D velocity field [2] of a dynamic scene [2]. Independent moving objects could be detected using motion segmentation [4], [1]. Optical flow is used to evaluate the existence of motion [2]. With the constraint of intensity conservation, apply the chain rule for differentiation, we have the below equation [8]:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{dI}{dt} = 0$$

Where I (x, y, t) is the image brightness at pixel P(x, y) at time t. Let u = dx/dt , v = dy/dt , Ex = ∂ I/∂x , Ey = ∂ I/∂y , Et = ∂ I/∂t , this equation can be rewritten as

$$Exu + Eyv + Et = 0$$

Horn's method [5] is used for the estimation of u, v, which are the optical flow velocities.

Optical flow is accurate for estimating the relative displacement between two adjacent frames for most objects. However for rain, which is an ensemble of rain drops falling at high velocities [6], its contribution to the motion field can be eliminated by setting a preset threshold value [2] (ρm) to f (x, y). Here m is the mean of the motion field, ρ is a parameter set according to how bad the rainy effect [4] is. In typical applications, ρ = 0.01 ∼ 0.1. By applying thresholding [2] it can also eliminate the bad influences caused by slow global camera motion [5]. A binary motion pixel map Im(x, y) is retrieved based on the thresholded optical flow field [4].

$$I_m(x, y) = \begin{cases} 0 & f(x, y) \geq \rho m \\ 1 & f(x, y) < \rho m, \end{cases}$$

$$f(x, y) = \sqrt{u(x, y)^2 + v(x, y)^2},$$

$$m = \frac{1}{MN}\sum_x \sum_y f(x, y).$$

Using optical flow for motion target segmentation has its intrinsic drawbacks [3], as it can only detect obvious intensity changes, which are usually at the boundary [4] of the target. Areas that belong to the target [5] without obvious intensity change cannot be effectively recognized [4], which causes the "Aperture Problem" [2]. In an effort to tackle this problem, a parametric Gaussian Mixture Model is used (GMM) [1] to simulate the distribution of Im(x, y). Initially, K GMM components are presumed to exist [8] in the optical flow field. Then, Expectation Maximization (EM) algorithm [8] is applied to calculate the optimal mean and variance [7] for each component [2]. The maximum likelihood function for the motion target can thus be expressed as below:

$$p(F_1(i, j)|C_1) = \sum_{k=1}^{K} \pi_k N((i, j)|\mu_k, \Sigma_k)$$

Where πk is the mixing coefficient for each Gaussian component, its values are determined in the EM iteration, with the constraint of sum (K) for k=1 to πk = 1. Here F1 represents the motion cue [12], which is combined later with locality cue [5] F2 for foreground/background classification [6]. C1 represents the pixel class for foreground object [4], and C2 represents the pixel class for background [5], the simulation result of motion target estimation on two video sequences [5]. Regarding the value of parameter K [6], the proposed algorithm does not rely on the assumption that each GMM [7] component represents [8] one single motion object, and it works well in situations [5] where the number of GMM components K [3] is larger than the number of motion object in the frame [5]. The GMM models the motion field of the whole frame, each pixel's motion cue [6] is determined by the combination of all Gaussian components [7]. Therefore, each motion object can be represented by one or many GMM components [3]; the algorithm works well when 'K' is chosen such that it is large enough [5]. For most applications, K-the maximum number of motion object

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

[6], can be estimated beforehand [9]. For example, for a street traffic surveillance camera with resolution 540×620, 20 motion object number is already sufficient [6], considering the possible number of vehicles that can appear in such limited space [5].
Considering the similarity between adjacent video frames, GMM parameters [1] from the current frame can be used as initial parameters for EM iteration [5] of the next frame, which will make the algorithm converge [9] very fast and highly efficient [9].

B. Inclusion of Local Properties

In an effort to include the local properties (pixel location, and chromatic values) into the segmentation, a feature vector is formed consisting of each pixel's spatial and color information [12], [13]:

$$F_{ij} = (R_{ij}, G_{ij}, B_{ij}, w \cdot i, w \cdot j)$$

In above Eqn. Ri j, Gi j, Bi j are the intensity values of the R, G, B channel at P (i, j), w is the weighting factor [4] between the color and position space [7]. For different frame size and scene complexity [5], w should be adjusted as different values. After generating the feature vectors [4], k-means clustering [3] is applied on the feature vector space. The total number of clusters is predefined [6] according to the frame size and scene complexity [6] as well. After the clustering [6], we count the number of motion pixels (according to the binary map Im(x, y)) that fall into each cluster [2]. The percentage of motion pixel number against the number of cluster's total pixel number will be used as that cluster's likelihood of motion is given by:

$$p(F_2(i,j)|C_1) = \frac{no.\ of\ motion\ pixels\ within\ cluster}{total\ no.\ of\ pixels\ of\ the\ cluster}$$

Here the symbol F2 represents the locality cue [7]. Since adjacent video frames [6] are closely related to each other, clustering results from the current frame [5] can be used as the initial parameters [7] for the iteration [12] on the next frame, this will help the algorithm converge [4] within several iterations, and makes the computation highly efficient [5].

C. Combining locality and motion cues

Let us assume that the likelihoods of two cues [4], i.e. motion cue and locality cue F1, F2 are totally independent, the combined conditional probability [5] is given by

$$p(A|C_i) = \prod_{k=1}^{2} p(F_k|C_i)$$

here p(A|Ci) is the combined conditional probability [9] for the pixel class Ci [12] . The posterior probability [5] for the foreground motion target can be written (defined as class C1) as shown below:

$$p(C_1|A) = \frac{p(A|C_1)p(C_1)}{p(C_1)p(A|C_1) + p(C_2)p(A|C_2)}$$

Here, p(C1) and p(C2) are the prior possibilities [6] for each class. p(C1) can be predicted as the ratio between the motion target area against the frame size, such that p(C2) = 1−p(C1). The decision boundary can be obtained by setting the likelihood in both classes to be equal, i.e.

$$p(C1|A) = p(C2|A) = 0.5$$

and a binary motion decision map is generated [5] to update Im. There are several other existing methods [11] for optical flow field segmentation [13], one famous algorithm is [15], where brightness constancy, gradient constancy, and smoothness assumptions are combined into a variation model [5]. A coarse to- fine warping strategy [5] is later applied to minimize the total energy consumed. This method provides a better optical flow estimation accuracy [5], however with an increased computational complexity [5]. As will be explained in the following sections, for rain removal [3], we need a segmentation [4] that can tell the motion intensive area roughly from the relatively static scene image background. The segmentation boundary is not at all essential, since it is of very low possibility that boundary pixels

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

[5] are covered by rain; accurate optical flow magnitude estimation [5] is also totally unnecessary, as long as it's well above the threshold [4]. Under such considerations [12], the proposed efficient method is preferred over others [4, 5] for the motion field segmentation [3, 5].

RAIN DETECTION

The proposed rain detection scheme is discussed below:

D. Differencing and thresholding

Initially, the grey scale intensity differences [2] between two successive [3] frames are calculated and thresholded [6]. The threshold value are set such that all the intensity fluctuations [3] caused by rain can be detected [1], the typical value is Dth = 3 (out of intensity scale 255) [2].

$$I_{diff} = \begin{cases} 1 & I_N - I_{N-1} \geq D_{th} \\ 0 & I_N - I_{N-1} < D_{th} \end{cases}$$

A binary difference map Idiff is calculated using above equation.

E. Applying photometric and chromatic constraints

Photometric constraints (according to [2], [12] and [10]) are applied on the candidate rain mask Idiff [3] calculated in the previous step. In the first step, constraint of intensity fluctuation range is applied on the pixels [4]; and then by considering the speed of rain streak [1], as well as the comparative camera sensor dimension [3], the connected area of the rain is limited to a certain number of pixels θ. Based on the experiment, epsilon = 3 ~ 20/255, θ = 30 ~ 50/ pixels, differing on different rain volume [3], camera frame size [4] and focal length settings [17]. The chromatic constraint [3] is also applied here, the final absolute sum of color channel differences caused by rain should be within a bound φ = 15/255 [1]. After applying the photometric and chromatic constraints [3], the pixels in Idiff that fail the constraints [3] are excluded from the final rain mask IRain as given below [1]

I_Rain = I-Diff – I_Fail

Quantitative comparison between different methods [4], [5] are carried out for rain detection [8] over static scene background. The video we use is 'car' (frame size 340 × 520), the rain is rendered using methods proposed in [8], and the rain ground truth is calculated as difference between the two videos before and after rain rendering [9]. The measuring metric is the rain mis-detection rate (MD/pels) [5], and false detection rate (FD/pels) [6].One can see that our hybrid method produces lower total error as compared to other competing methods [4], [7].

F. Motion Exclusion

Rain pixels within the motion object and the background must be treated separately, hence here IRain is divided into two sets namely: 1. rain candidate pixels in the motion target area will comprise the set Sm; 2. Rain candidate pixels in the background area will comprise the subset Sb. Finally, the pixels that are not included in Sm or Sb form the set Sp, which are the pixels that are not covered by rain. The definition of Sm, Sb and Sp are expressed in the equation below. SC is the complete set of frame pixels [5], BM is the motion buffer [9].

$$\begin{aligned} S_m &= \{I(x,y) | I_{Rain}(x,y) = 1 \ \& \ B_M(x,y,n) = 1\} \\ S_b &= \{I(x,y) | I_{Rain}(x,y) = 1 \ \& \ B_M(x,y,n) = 0\}, \\ S_p &= S_C - S_m - S_b. \end{aligned}$$

IV. FRAME, RAIN AND MOTION BUFFER

A total of three buffers are created for the rain removal [4]: video frame buffer BI (len,wid, stk) [5], rain buffer BR(len,wid, stk) [1] and motion buffer BM(len,wid, stk) [9]. Here len × wid is the video frame size [6], stk is the depth of the buffer, and it is set as stk = 9 in the experiment [9] for a better recover performance [12]. Each layer of BI

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

comprises one video frame [4]. The newest frame is pushed on top of the buffer (buffer layer 1), and the oldest frame [3] is moved out from the bottom (buffer layer stk) [1], and the rest of the layers update accordingly [7]. The rain buffer BR(len,wid, stk) records the binary rain map IRain [7] for each corresponding video frame in BI [9], and motion buffer BM(len,wid, stk) records [9] the corresponding binary motion decision map Im. Both BR, BM update in sync with the video frame rate buffer BI [5]. The scene recovery algorithm works on the central frame (BI (x, y, n), n = stk+12), such that both historical [5] (BI, R, M(x, y, 1 ~ stk−1 2)) and future (BI, R, M(x, y, stk+32~stk)) information in the video, rain and motion buffer could be retrieved for a better scene recover performance [5].

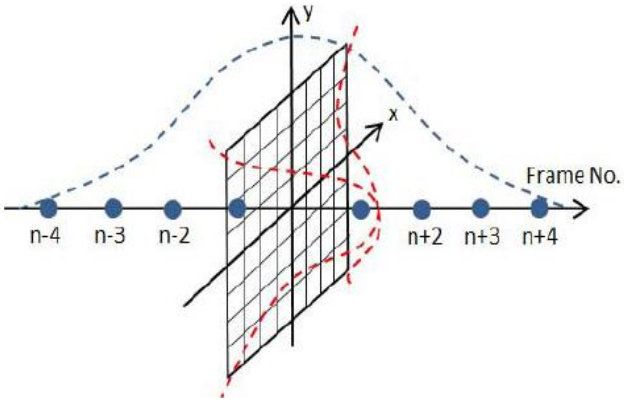

Fig. 1 Illustration of the 88-spatial-temporal neighborhood pixels used for rain removal

V. SCENE RECOVERY

A 88-spatial-temporal neighborhood V [5] is designed for rain removal, which is illustrated in Fig. 1. Lie in the center of the neighborhood system [2] is the pixel BI (x, y, n). V consists 8 pixels along the time axis in BI with location [8] BI (x, y, (n−4) ~ (n+4)), and 80 pixels in the current frame with area BI ((x −4) ~ (x +4), (y−4) ~ (y +4), n). Similar neighborhood set-up could be found in [1], where they use a 26-connected 3×3 neighborhood [5]. The temporal neighborhood is set to be 8 in our method for a better recovering visual result [6], and for a better resilience against motion occlusion [2]. The spacial neighborhood is set as 80, for the fact that rain [8] streak breadth is usually around 3 to 8 pixels. The pixel values [9] in the neighborhood V will later be used to predict the center rain covered pixel by a weighted sum [8]. The definition of the weight is given in Eqn. below, they will later be assigned to each pixels according to their adjacency [7] to the center BI (x, y, n) as shown below:

$$w_{x,y,n}(i,j,t) = \bar{B}_R(x,y,n) \times exp\left(-\alpha||v(x,y,n) - v(x,y,t)||^2 - \beta||v(x,y,n) - v(i,j,n)||^2\right),$$

Here ¯BR is the binary complement of BR, it is used to filter out the pixels in the neighborhood [6] that are also covered by rain. V (i, j, t) is the vector representing [9] the spatial-temporal location [3] of each pixel in the 88-neighborhood V. Within the exponential term [7], (−α||v(x, y, n) − v(x, y, t) ||2) regulates the filter weights [5] along the time (frame no.) axis, and (−β||v(x, y, n) − v (i, j, n) ||2) regulates the filter weights [2] in the pixel's spatial neighborhood [5].

In the previous section, for each frame the pixels are divided into three sets: Sb, Sm, and Sp. Different filter coefficient α, β in Eqn. below are used for pixels of different sets, based on their different dynamic properties [6].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

G. Rain pixels in static background

For the rain covered pixels in static background [7] (pixels in set Sb), filter coefficients [6] are set as α = 2 stk, β = 0, this makes the filter shaped as Gaussian [9] with variance stk2 along the time axis, but no spacial neighbor [8] values are used. With the Gaussian pattern [7], temporally adjacent records (BI (x, y, n−1) and BI (x, y, n+1)) are assigned with highest weights [9], and this is out of two considerations: First, according to [11], when a certain pixel is covered by rain, its temporally adjacent pixels [9] will have a higher possibility that they will not also be covered by rain [9]; Second, if considering camera motion, or background lighting change [9], adjacent pixels are certainly more likely to be close to the original value [3]. The rain removal filter kernel [7] for this set of pixels is therefore defined as:

$$B_I(x,y,n) = \frac{\sum_{v(i,j,t)\in V} w_{x,y,n}(i,j,t)\bar{B}_M(i,j,t)B_I(i,j,t)}{\sum_{v(i,j,t)\in V} w_{x,y,n}(i,j,t)\bar{B}_M(i,j,t)},$$

Where ¯BM is the binary complement [6] of the motion buffer[3], which will filter out the pixels in the time axis that have been classified as motion object [3]. The motion buffer BM provides the information [5] which layers of BI are usable for the background recovery [3].

It is possible that a pixel can be judged to belong to background [8], while in the motion buffer, all the historical and future records are judged to be motion object [6], and vice versa. This happens when motion segmentation results fail, it can also happen when object motion [4] is very fast. In both situations, no temporal references could be used for pixel recovery. For this specific case, β is set to be $14\sqrt{2}$ to utilize the spacial neighborhood information [6], similar to the recovery for motion objects [2].

H. Rain Covered Pixels in Motion Objects

For the rain covered pixels in motion objects [5] (pixels in set Sm), filter coefficients are set as α = 1, $\beta = 14\sqrt{2}$. In the time axis, this gives only considerable weights [3] to the two nearest neighbor in the time axis BI (x, y, n − 1), BI (x, y, n + 1), giving consideration to the fact that the pixel value changes [8] fast when it belongs to a motion object [4], so when it is covered by rain [6], buffer layers that are far away from the buffer center provides little relevant information [10] for the motion object, thus their weights are set to be very small [5]. On the contrary, as $\beta = 14\sqrt{2}$, more weights are given to the motion pixel's special neighbors [6], and it has a 2-D Gaussian shaped distribution [5] that corresponds to the red curves . The recovered pixels value is calculated as:

$$B_I(x,y,n) = \frac{\sum_{v(i,j,t)\in V} w_{x,y,n}(i,j,t)B_M(i,j,t)B_I(i,j,t)}{\sum_{v(i,j,t)\in V} w_{x,y,n}(i,j,t)B_M(i,j,t)}$$

C. Pixels Uncovered by Rain

Finally, for the pixels that are not covered by rain, we will simply keep their values.

VI. RESULTS

The algorithms were run on on two videos of highly dynamic rainy scenes. The frames are shown in Fig. 2. And Fig.4 the results are shown in Fig. 3 and 5.

As it can been seen from the results, rain pixels are very well removed. The cars are also not blurred by the rain removal algorithm in spite of its large motion occlusion, and no leaving trails (ghost effect) are observable in the Fig.3.and 5.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Fig.2: Rainy image

In general, the results show that the proposed rain removal algorithm has a much better performance over others for rain removal in highly dynamic scenes [4] than any other existing algorithms [8].

Fig.3. Rain Pixel Removal Image

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Fig.4 Another rainy image

Fig.5 Rain pixel removed image

VII.CONCLUSIONS

From the simulation results, we can conclude that the existing rain removal algorithms perform poorly in highly dynamic scenes, some serious pixel corruptions occur in motion intensive areas, which is caused by ignoring motion occlusions during pixel recovery. Based on our proposed motion segmentation scheme, our method recovers most of the rain pixels such that each pixel's dynamic property as well as motion occlusion clues are considered; both spatial and temporal information are adaptively and successfully exploited during rain pixel recovery. Experiment results obtained using Matlab show that our algorithm outperforms existing ones in highly dynamic scenarios. In other words the proposed systems works extremely well in all the conditions.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

REFERENCES

[1] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 1, Jul. 2004, pp. 528–535.
[2] K. Garg and S. K. Nayar, "Vision and rain," Int. J. Comput Vis., vol. 75, no. 1, pp. 3–27, 2007.
[3] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in Proc. IEEE Int. Conf. Multimedia Expo, Jul. 2006, pp. 461–464.
[4] A. K. Tripathi and S. Mukhopadhyay, "A probabilistic approach for detection and removal of rain from videos," IETE J. Res., vol. 57, no. 1, pp. 82–91, Mar. 2011.
[5] A. Tripathi and S. Mukhopadhyay, "Video post processing: Low-latency spatiotemporal approach for detection and removal of rain," IET Image Process., vol. 6, no. 2, pp. 181–196, Mar. 2012.
[6] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," IEEE Trans. Pattern Anal. Mach. Intell., vol. 11, no. 5, pp. 490–498, May 1989.
[7] A. Ogale, C. Fermuller, and Y. Aloimonos, "Motion segmentation using occlusions," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 988–992, Jun. 2005.
[8] J. P. Koh, "Automatic segmentation using multiple cues classification," M.S. dissertation, School Electr. Electron. Eng., Nanyang Technol. University, Singapore, 2003.
[9] B. K. Horn and B. G. Schunck, "Determining optical flow," Artif. Intell., vol. 17, pp. 185–203, Jan. 1981.
[10] M. Shen and P. Xue, "A fast algorithm for rain detection and removal from videos," in Proc. IEEE Int. Conf. Multimedia Expo, Jul. 2011, pp. 1–6.
[11] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer-Verlag, 2006, pp. 423–455, Ch. 9.
[12] B. Heisele, U. Kressel, and W. Ritter, "Tracking non-rigid, moving objects based on color cluster flow," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 1997, pp. 257–260.
[13] J. Fan, D. Yau, A. Elmagarmid, and W. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," IEEE Trans. Image Process., vol. 10, no. 10, pp. 1454–1466, Oct. 2001.
[14] H. Stark and J. W. Woods, Probability and Random Process with Applications to Signal Processing, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002, pp. 589–601.
[15] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in Proc. ECCV, May 2004, pp. 25–36.
[16] R. J. Kubesh and K. V. Beard, "Laboratory measurements of spontaneous oscillations for moderate-size raindrops," J. Atmos. Sci., vol. 50, no. 1, pp. 1089–1098, 1993.