https://doi.org/10.2352/EI.2022.34.15.COLOR-234
© 2022, Society for Imaging Science and Technology

Motion Detection In a Color Video Sequence with an Application to Monitoring a Baby*

Yang Yan[a], Qian(Rain)Guo[b], Wengui(White)He[b], Peng(Davi)He[b], GeorgeChiu[a], JanP.Allebach[a]
[a]PurdueUniversity,WestLafayette,IN47906,USA
[b]SunvalleytekInc.,Shenzhen,Guangdong,China

Abstract

Personal real-time video monitoring devices are popular in recent years especially for people living in bigger houses who are taking care of babies. People take advantage of a real-time video monitoring device to monitor the babies' activities while they are occupied by other issues. However, danger happens any time in any situations. It is necessary for a baby monitoring device to have a motion detection function to trigger recording functions or alert the guardians. This paper introduces a solution for the motion detection problem. The solution combines statistical methods, kernel density estimation and histogram analysis methods, and deep learning method for detecting the salient object. The final outputs are both the motion levels indicating the severity of the motions and the bounding boxes indicating the location of the motion happening in the video frame. The different levels of motion can later be used as triggers for different functions built into the device, such as starting video recording, playing sirens, etc., and the bounding boxes can provide reference focus areas for the guardians to check the details of the motion to decide whether to take actions or not.

Introduction
Background

People care about the safety of their babies. Danger can happen any time to anyone. Due to babies' natural naivete and innocence, babies themselves may not notice that danger is close and they may not know the proper ways to take actions. However, people are unable to ensure that babies are under supervision 24 hours 7 days all year round. Even within one house, it is not possible for guardians to be around the babies all the time. For the sake of safety, any kind of personal real-time video monitoring device is a necessary product. On the other hand, babies do not always need an adult's full attention, especially when babies are sleeping or staying within their crib. For these situations, it is reasonable to leave babies attended only through a personal real-time video monitoring device as long as the guardians stay in a reachable distance, such as another room but in the same house. In this case, safety and emotional fluctuation of the babies can be noticed through a real-time video monitoring device. If needed or any kind of danger happens, guardians can make their way to the babies and take action immediately. Guardians can work on other household issues while watching their babies with the assistance of the baby monitoring device. This way, people will save their time and efforts from watching babies sleeping only for safety concerns.

*Research supported by Sunvalleytek International Inc., Shenzhen, Guangdong, China

Not like other security surveillance video monitoring devices, baby monitoring devices mainly focus on real-time video displaying and recording only the issues happening around babies. A baby monitoring device does not need to record 24-7. Then, the motion detection algorithm becomes the very first activation algorithm for triggering the recording function and even other functions built-in the monitoring device. With the motion detection algorithm acting as an activation, only the meaningful clips will be recorded. In this way, the disk storage space can be more efficiently used and the device does not need as large a storage as before. With the same size of disk storage, the device can store videos that happened much earlier and are more meaningful. The important recorded videos will not be easily missed or wasted by meaningless and repetitive recordings. It is feasible for users to find their wanted video clips.

The motion detection algorithm detects motions. Also, it works as an activation for other built-in functions in the system. Once the motion detection algorithm gives out a positive signal indicating temporal changes detected by the system, the system then enables the other built-in functions including recording, alarming, etc. Not all the temporal changes are considered vital for baby monitoring situations, for example, swinging trees due to strong winds, blinking of the LED lights on Ethernet routers, etc. So, for these cases, the motion detection algorithm is not supposed to report motions to the system. The motion detection algorithm not only detects any motions in the video but also makes decisions as to whether to report motions to the system. An additional function that can be added to a motion detection algorithm is reporting the types and severeness of motions so that the system can decide which functions are to be activated.

Related Works

Multiple and various methods have been applied for solving the motion detection problem recent years. Initially, most of motion detection methods are set up based on background detection and subtraction [2][3]. Based on a reference image, Singla [1] described a motion detection method by applying Otsu's method [5] to the differences between two adjacent frames captured by a camera. Later on, Zhao [4] proposed a target motion detection algorithm that is not restricted by using Otsu's method [5] to determine the threshold for background detection and subtraction. The thresholds are set with the help of the information between frames. As the current time stamp moves, the current frame moves to the next frame; the information between frames changes; and the thresholds vary.

Researches have also taken advantage of statistical tools and applied them to background detection and subtraction. As reviewed by Piccardi [6], parametric and non-parametric approaches have been applied in the methods of background detection and subtraction. Parametric approaches include setting a threshold separating background and foreground based on a running Gaussian average, medians, mixture of Gaussians, etc. Non-parametric approaches include Kernel density estimation, sequential Kernel density approximation, etc. Lu, et al. [17] moves further by applying a mixture of Gaussian models together with a frame difference for motion detection. In this way, combining the statistical methods on background subtraction with frame difference techniques becomes a common techniques in solving motion detection problems. According to the investigation of Soh, et al. [14], among all of these statistical methods combined with a frame difference technique, kernel density estimation [23] has a better overall performance for videos in various scenes with different levels of motions and background of different levels of noises.

These statistical methods are actually doing modeling of the background in some degree. Some modeling methods are simple and direct with one or a few thresholds [10]. For video or complicated background scenarios, adaptive background modelings [15][16] have been investigated. Stauffer and Grimson [15] are using mixture of Gaussians to generate mltiple values for multiple background objects when necessary. While McFarlane and Scholield [16] take advantage of traditional image processing tools such as connected components, etc. to handle the slight changes in background. Hamad and Tsumura [11] perform background modeling by recursively updating the background threshold with clustering results and the former threshold results. Hwang, et al. [7] combine statistics tools to update the paremeters for mixture of Guassian modeling of the background of urban traffic videos.

Other than the statistical tools, multiple machine learning methods have also been applied to solve the motion detection problem. For the purpose of detecting and subtracting background, the clustering based methods such as sequential clustering [8], and codebook [9] have been applied. As for deep learning methods, optical flow [18] plays an important part in solving motion detection with a static background [19][20]. Optical flow tracks the moving object locations between two frames and separates the moving objects from the static background. Optical flow basically calculates the motions into a two-dimensional vector at each pixel location to form a vector field over the frame image. Optical flow takes advantage of the motion trajectory vectors, however, introducing abundant noise at the same time.

Since background separation cannot be achieved perfectly with a single method, some traditional image processing methods have been combined and applied as pre-processing procedures or post-processing procedures. For example, Han, et al. [21] applied noise filtering after optical flow as a post processing for removing abundant noise.

For some cases with specific characteristics, some targeted techniques have been applied. According to the periodic behavior [12] of the given moving objects, periodic analysis methods have been applied to solving the problem of motion detection [13].

For baby monitoring devices, the main purpose of the motion detection function is to save power and resources while capturing meaningful motions. In this case, generally, our method combines the statistical method [23] for motion detection and utilizes a saliency detection map [22] for only focusing on meaningful foreground area.

Methods

Data Collection

Since this algorithm is developed specifically for a baby monitor product, sample videos are firstly generated with the target baby monitor device. With threads sewed on the clothes in which the baby doll is dressed, the baby doll's legs and arms can be lifted manually to create meaningful motions while shooting videos. The baby doll is moved to different scenes with diverse settings including indoor scenes with sofa and decorations, and outdoor scenes with lawn and trees (Figure 1).

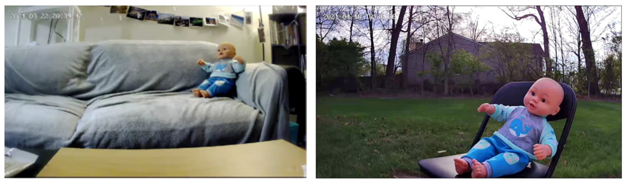

Figure 1. *(left) A sample video frame of the video shot with a baby doll and indoor settings.*
(right) A sample video frame of the video shot with a baby doll and outdoor settings.

To increase the varieties of the videos and enlarge the scene complexity, some publicly accessible video clips that people uploaded to public websites are used for testing the algorithm. These color videos are shot with fixed angle, fixed position and minimum 30 fps (frame per second) by baby monitors sold on the market with a baby or a toddler that stays in the center area of the video frames (Figure 2).

Figure 2. *Sample video frames of the videos shot with fixed angel and fixed position baby monitors sold on the market with a baby or a toddler stays in the center area of the video frames.*

Overview

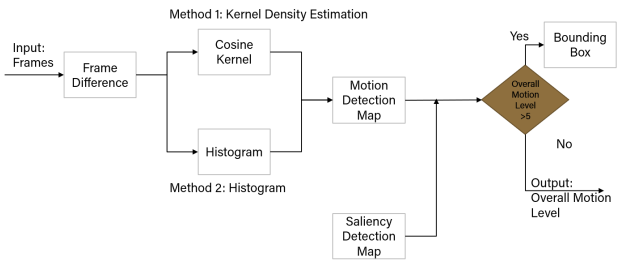

Figure 3. *Block diagram of the overall algorithm.*

As shown in the block diagram in Figure 3, the frame differences are firstly calculated based on input frames, then the motion detection map is produced by either a kernel density estimation method [23] or a histogram. With the motion detection map and the Saliency detection map [22], the overall motion level of the frame is calculated. Finally, the bounding box is printed on the frame when the overall motion level is over 5 or the overall motion level is outputted when the overall motion level is equal or under 5.

Data Preparation: Frame Sampling

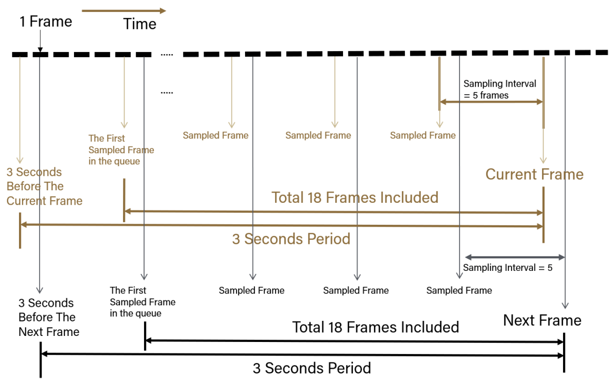

Figure 4. *Frame samplings before doing frame difference calculations.*

The algorithm will be very slow and cost too much resources if the frame differences are calculated between every frame within the time window within which we would like to capture the motions. And the frame difference between the adjacent frames does not show a strong signal when the video is captured with 30 fps. The object movement happening within $1/30$ second is not noticeable. For the sake of saving energy and resources and at same time increasing calculation efficiency, frame sampling is performed before frame differences calculation. As shown in the Figure 4, every time a new frame inputted, the frame differences are only calculated with 18 frames sampled from last 3 second time period. The 17 frame differences data calculated with these 18 frames form a frame difference sequence. When time moves forward 1 frame, the selected sampling frames also move forward 1 frame ahead. This guarantees the fluency of the output video and the accuracy of movement captures. In this way, the frame difference sequence is regenerated every time the algorithm gets a new video frame inputted.

Motion Detection Map

With the frame difference sequence, two different motion detection methods, kernel density estimation [23] and histogram-based method, are experimented with to get a motion detection map. Kernel density estimation is selected to be implemented because it is the most accurate and adaptive to various scenes among all the statistical methods according to Soh, et al. [14]. For the histogram-based method, we found it to be the most straight forward and direct method for solving the motion detection problem.

For a single-channel:

$$\begin{aligned} p_h(f_t) &= \frac{1}{N}\sum_{i=1}^{N} K_h(f_t - f_i) \\ &= \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{f_t - f_i}{h}) \end{aligned} \tag{1}$$

For the multi-channel case:

$$\begin{aligned} p_h(f_t) &= \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d} K_h(f_{t_j} - f_{i_j}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d} \frac{1}{h} K(\frac{f_{t_j} - f_{i_j}}{h}) \end{aligned} \tag{2}$$

According to Elgammal, et al. [23], we implement the kernel density estimation method as follows. First, let us introduce some notation. We let f_1, f_2, $\cdots$,f_N be the previous N samples of intensity values for some pixel x. Given these samples, kernel density estimation is used to estimate the probability density at any intensity value of the pixel x. Let x_t be an intensity value of the pixel at time t. Then, we can estimate probability density $p_h(f_t)$ for pixel value f_t as shown in Equation 1 for single- channel situations or grey scale videos in our case and in Equation 2 for multi-channel situations or color videos in our case. In these equations, h is a bandwidth adapting to different scenes. Large bandwidth leads to under-smoothing results, while small bandwidth leads to over-smoothing results. d is the total channel number for the multi-channel case. And the kernel function utilized here is a cos kernel as shown in Equation 3, which exhibits the best performance according to Soh, et al. [14].

$$K(u) = \begin{cases} \cos(\frac{\pi}{2}\cdot\frac{u}{256}) & |u| \le 255 \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

With the kernel density estimation results, each pixel is assigned an index number that ranges from 0 to 10, which reflects the motion levels (the severity of the motions). The motion levels are decided with the max-min method. First, the data range is calculated across the frame as shown in Equation 4. Then, the motion level $I(i, j; f)$ for each pixel location is assigned according to Equation 5. Basically, the kernel density estimation result for each pixel location is equally quantized into 10 levels according to the data range calculated over the current frame.

$$\begin{aligned} R(f) = &p_{h,max}(f) - p_{h,min}(f) \\ &\text{where } f \text{ is the frame index.} \end{aligned} \tag{4}$$

Motion level at location i, j:

$$\begin{aligned} &I(i, j; f) \in (0.9, 1.0] \cdot R(f) + p_{h,min}(f) \rightarrow \text{level } 0 \\ &I(i, j; f) \in (0.8, 0.9] \cdot R(f) + p_{h,min}(f) \rightarrow \text{level } 1 \\ &\qquad\vdots \\ &I(i, j; f) \in (0.1, 0.2] \cdot R(f) + p_{h,min}(f) \rightarrow \text{level } 9 \\ &I(i, j; f) \in (0.0, 0.1] \cdot R(f) + p_{h,min}(f) \rightarrow \text{level } 10 \end{aligned} \tag{5}$$

The second method we applied to detect motion is using the histogram of the frame difference. Similar to kernel density estimation method, a motion level is assigned to each pixel location according to the histogram of the frame difference. The range of ± 3 standard deviations (STDs) away from mean value is equally quantized into 9 levels. For example, if the histogram value at the pixel location is within the range of $[0, \pm 0.3] \cdot$STD away from the mean value, the motion level of the pixel location is assigned to be 0, etc. If histogram value at the pixel location is within the range of $(\pm 2.7, \pm 3] \cdot$STD away from mean value, the motion level is assigned to be 9. And if the histogram value at the pixel location is over $\pm 3\cdot$STD away from mean value, the pixel location will be assigned to be 10.

Overall Motion Level Calculation

With the motion detection map, each pixel location is assigned a motion level value from 0 to 10. The overall motion level needs to be calculated for each input frame. And we require the overall motion level to be focused on the meaningful movement of the main object, in our case, the baby. So, we take advantage of the saliency detection map generated by a deep learning neural network proposed by Hou, et al. [22]. The saliency detection map segments the image into a foreground area and a background area. The overall motion level M for the video frame is calculated as the maximum between the motion level of the foreground area M_O and the motion level of the background area M_B (Equation 6).

$$M = max(M_O, M_B) \tag{6}$$

The motion level of the background area M_B is simply assigned as the maximum motion level among the pixels belonging to the background area B according to the saliency detection map (Equation 7). Here, $I(i, j)$ is the motion level at pixel (i, j)

$$M_B = max(I(i, j) \in B) \tag{7}$$

The motion level of the foreground area M_O is assigned to be the maximum value between two values $M_{O,1}$ and $M_{O,2}$ (Equation 8). $M_{O,1}$ is calculated as the average motion level among the pixels belonging to the foreground area according to the saliency detection map (Equation 9). $M_{O,2}$ generally indicates the average of the motion level for big objects when the foreground area has multiple movements happening at the same time. For calculating $M_{O,2}$, the foreground area is binarized by assigning motion levels equal or over 5 as 1, and assigning motion levels below 5 as 0. Then, a connected component operation is performed. Each component that is larger than 1% of the foreground area is assigned its average motion level. And finally, the mean of component motion levels is assigned to $M_{O,2}$.

$$M_O = max(M_{O,1}, M_{O,2}) \tag{8}$$

$$M_{O,1} = \frac{1}{N_O} \sum_{I(i,j) \in O} I(i, j) \tag{9}$$

Again, $I(i, j)$ is the motion level at pixel (i, j)

Output Results

For each input frame, the overall motion level M is calculated as described in last section. If the overall motion level is equal to or greater than 5, the output video will be generated with a bounding box to indicate the location where the motion is happening and other system built-in functions will be activated. Otherwise, the output video will do nothing but report the overall motion level to the system. The video frames of a sample output video are shown in Figure 5. The provided video frames are demonstrative output video frames. When implemented in the system, the motion level values will not be displayed on the video frames, only the bounding boxes will be presented.

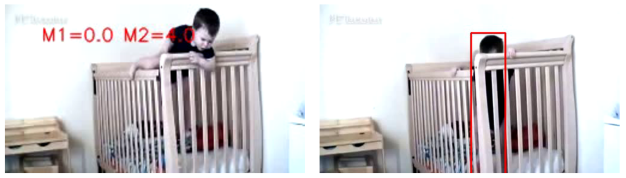

Figure 5. The two images are video frames of a sample output video. (left) M1 displayed in the video frame shows the value of M_B indicating the motion level of the background area. M2 displayed in the video frame shows the value of M_O indicating the motion level of the foreground area. (right) The output bounding box is shown in the video frame when the overall motion level is equal or greater than 5.

Conclusion

Generally speaking, this motion detection algorithm accurately detected motions with motion levels and bounding box outputs. When the algorithm was tested with frames of video inputted frame by frame, the overall motion levels can be measured and reported.

However, some issues remain to be solved. While we tested the algorithm with various videos, over-exposure and lightness changes can cause false alarms as shown in Figure 6. Further improvements can be work on over-exposure detection and auto-lightness adaption.

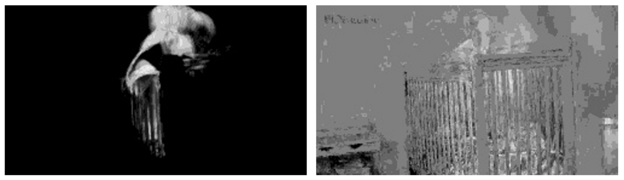

Figure 6. Exposure and lightness changes cause false motion detection signals. The images are the index image of the motion detection map. Due to the over exposure, the video frame shown on the right indicates that the false alarms appear throughout the frame. And the video frame shown on the left is the video frame that was captured one half second before the video frame shown on the right.

References

[1] Nishu Singla, Motion Detection Based on Frame Difference Method, International Journal of Information Computation Technology, 4, 15 (2014). pg. 1559-1565.

[2] Rupali S. Rakibe, Bharati D. Patil, Background Subtraction Algorithm Based Human Motion Detection, International Journal of Scientific Research Publications, 3, 5 (2013). pg. 2250-3153.

[3] K. Kavitha, A. Tejaswini, VIBE: Background Detection and Subtraction for Image Sequences in Video, International Journal of Computer Science and Information Technologies, 3, 5 (2012), pg. 5223-5226.

[4] Zeyi Zhao, Gang Lu, Target Motion Detection Algorithm Based on Dynamic Threshold, Journal of Physics: Conference Series, 1738, 1, (2021), pg. 012085.

[5] Nobuyuki Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man Cybernetics, 9, 1, (1979), pg. 62-66.

[6] Massimo Piccardi, Background Subtraction Techniques: A Review, 2004 IEEE International Conference on Systems, Man and Cybernetics, 4, (2004), pg. 3099-3104.

[7] Pyung-Soo Hwang, et al., A Statistical Approach to Robust Background Subtraction for Urban Traffic Video, 2009 Second International Workshop on Computer Science and Engineering, 2009, pg. 177-181.

[8] Mohcene Benalia, and Samy Ait-Aoudia, An Improved Basic Sequential Clustering Algorithm for Background Construction and Motion Detection, Image Analysis and Recognition, Berlin, Heidelberg:Springer Berlin Heidelberg, 2012, pg. 216-223.

[9] Kyungnam Kim, et al., Real-time Foreground-Background Segmentation Using Codebook Model, Real-Time Imaging, 11, 3, (2005), pg. 172-185.

[10] Shuming Jiang, et al., A New Algorithm for Background Extraction Under Video Surveillance, IEEE Conference Anthology, (2013), pg. 1-4.

[11] Ahmed M. Hamad, and Norimichi Tsumura, Background Subtraction Based on Time-Series Clustering and Statistical Modeling, Optical Review, 19, 2, (2012), pg. 110-120.

[12] Ross Cutler, and Larry S. Davis, Robust Real-Time Periodic Motion Detection, Analysis, and Applications, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22, 8, (2000), pg. 781-796.

[13] Fang Liu, and Rosalind W. Picard, Finding Periodicity in Space and Time, Sixth International Conference on Computer Vision(IEEE Cat. No.98CH36271), Bombay, India, 2002, pg. 376-383.

[14] Youngsung Soh, et al., Performance Evaluation of Various Functions for Kernel Density Estimation, Open Journal of Applied Sciences, 3, (2013), pg. 58-64.

[15] Christ Stauffer, and W.E.L Grimson, Adaptive Background Mixture Models for Real-Time Tracking, 1999 IEEE Computer Society Conference on Computer Vision Pattern Recognition (Cat. No PR00149), 2, (1999), pg. 246-252.

[16] Nigel J. McFarlane, et al., Segmentation and Tracking of Piglets in Images, Machine Vision and Applications, 8, 3, (1995), pg. 187-193.

[17] Xiaofeng Lu, et al., Improved Background Subtraction Method for Detecting Moving Objects Based on GMM, IEEJ Transactions on Electrical Electronic Engineering, 13, (2018), pg. 1540-1550.

[18] Joel Gibson and Oge Marques, Optical Flow and Trajectory Estimation Methods, Springer International Publishing, 2016, pg. 9-23.

[19] Simon Denman, Clinton Fookes, and Sridha Sridharan, Improved Simultaneous Computation of Motion Detection and Optical Flow for Object Tracking, 2009 Digital Image Computing: Techiniques applications, 2009, pg. 175-182.

[20] Li Dan, et al., Moving Object Tracking Method Based on Improved Lucas-Kanade Sparse Optical Flow Algoritm, 2017 International Smart Cities Conference (ISC2), 2017, pg. 1-5.

[21] Hong Han, and Minglei Tong, Human Detection Based on Optical Flow and Spare Geometric Flow, 2013 Seventh International Conference on Image and Graphics, 2013, pg. 459-464.

[22] Qibin Hou, et al., Deeply Supervised Salient Object Detection with Short Connections, in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pg. 3203-3212.

[23] Ahmed Elgammal, et al. Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, Proceedings of the IEEE, 90, 7, (2002), pg. 1151-1163.

Author Biography

Yang Yan *received her B.S. (2016) Electrical Engineering from Purdue University and currently is a Ph.D. student in the Purdue ECE department. Her primary area of research has been image processing, image quality evaluation, and machine learning.*

Qian(Rain) Guo *is an algorithm engineer in Sunvalleytek International Inc, Shenzhen, Guangdong, China. She received her B.S. in Light Chemical Engineering from Qufu Normal University, Rizhao, Shandong, China in 2013 and M.S. in Light Chemical engineering from South China University of Technology, Guangzhou, Guangdong, China in 2016.*

Wengui(White) He *is a senior director of embedded software team in Sunvalleytek International Inc, Shenzhen, Guangdong, China. He received his B.S. in Communication Engineering from Hunan University of Technology, Zhuzhou, Hunan, China in 2011.*

Peng(Davi) He *is a software manager in Sunvalleytek International Inc., Shenzhen, Guangdong, China. He received his B.S. in Communication Engineering from Hunan University of Arts and Science, Changde, Hunan, China in 2012.*

George T. Chiu *is a Professor of Mechanical Engineering with courtesy appointments in Electrical and Computer Engineering and Psychological Sciences at Purdue University. He received the B.S. degree from National Taiwan University and the M.S. and Ph.D. degrees from University of California at Berkeley. His research interests are mechatronics and control with applications to digital printing and imaging systems, digital fabrication and functional printing. He is a Fellow of ASME and IS&T.*

Jan P. Allebach *is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach was named Electronic Imaging Scientist of the Year by IS&T and SPIE, and was named Honorary Member of IS&T, the highest award that IS&T bestows. He has received the IEEE Daniel E. Noble Award, the IS&T/OSA Edwin Land Medal, the IS&T Johann Gutenberg Prize, is a Fellow of the National Academy of Inventors, and is a member of the National Academy of Engineering.*