Journal of
Electronic Imaging

SPIEDigitalLibrary.org/jei

Camera that takes pictures of aircraft and ground vehicle tires can save lives

Yair Wiseman

SPIE

IS&T
imaging.org

Camera that takes pictures of aircraft and ground vehicle tires can save lives

Yair Wiseman
Holon Institute of Technology
Computer Science Department
Holon 58102, Israel
E-mail: wiseman@cs.biu.ac.il

Abstract. *Damaged tires cause many fatal car accidents that claim lives worldwide. The use of a regular digital camera that can generate JPEG images, so as to become aware of damaged tires, is proposed. The digital camera repeatedly takes pictures of the car's tires. When there is a drastic change in a particular block of JPEG, the value of many of its frequency coefficients will be high and it will be compressed into many more bits; therefore, if the image is overly large, an embedded computer system will turn on a noticeable light-emitting diode in the car. The driver will be aware of the damaged tire before a severe accident would possibly happen.* © 2013 SPIE and IS&T *[DOI: 10.1117/1.JEI.22.4.041104]*

1 Introduction

Damaged tires are a critical reason for many fatal accidents. A new warning from the National Highway Traffic Safety Administration from June 2011 notifies the drivers that "The latest data from the Department's National Highway Traffic Safety Administration show that over the five-year period from 2005 to 2009, nearly 3,400 people died, and an estimated 116,000 were injured, in tire-related crashes."[1]

These worrisome numbers motivated researchers to find ways to detect below-standard tires. Some of them used special equipment like an eyefish lens[2] or a complex intelligent wireless sensor network inside the tire.[3–5]

Actually, automatic tire inspection is a very old objective and a patent for using an x-ray device for tire inspection can be found at the very old date of 1939.[6] Some other devices have been presented during the years, which have used many other mechanisms like a hardware for interferometric analysis of the tires,[7] thermal cameras,[8] acoustic emission sensors,[9] or a laser-based illumination device.[10]

One more work that deals with automatic tire inspection is described in Ref. 11. The objective of this work is classification of tires according to their type, i.e., all-season, snow, studded, summer. However this is an easier task because they find out the shape of the entire tire. Our purpose is detecting spot damage in the tire.

Some other works aim at simulating accidents and estimating the potential damage.[12–14] This is not the aim of our work. Our work aims at detecting a potential exposure to danger and notifying drivers about it.

Paper 13195SS received Apr. 16, 2013; revised manuscript received May 28, 2013; accepted for publication May 30, 2013; published online Jul. 9, 2013.

Another work that is noteworthy is Ref. 15. This work deals with JPEG pictures of tires that were involved in a car accident. The marks on the tire are usually unnoticeable and it makes the picture ineffective, so the authors suggest employing some histogram equation so as to make the picture clearer. However, their method is unsuitable for this paper because they aim at human eye's comprehension while this paper aims at automatic machine's comprehension.

We suggest using an ordinary digital camera to find below-standard tires. Nearly all digital cameras can produce JPEG pictures. JPEG is a very common method for image compression and it is also extensively used by electronic devices like scanners and digital cameras[16] as well as by vehicle equipment like global positioning system.[17]

JPEG images have many advantages like the ability of being decoded in parallel,[18] the straightforwardness of adaptation for new compression methods,[19,20] and the capability of flexible implementation for hardware from different vendors.[21]

Images are often stored in a compressed standard. A naive approach for image processing on compressed images would be to decompress the image and then run the image processing algorithm on the original image data. Instead, for some image operations, we can act on the compressed data directly. This gives us two benefits: first, we can use the standard digital cameras without a need to adjust the digital camera; second, we can use the frequency information embedded in the compressed data.

The rest of the paper is organized as follows: Sec. 2 explains how JPEG can be used for detecting damaged tires, Sec. 3 gives some results, Sec. 4 explains how the method can be extended to motion JPEG-2000, and Sec. 5 concludes the paper.

2 Using JPEG for Damaged Tire Detection

JPEG is a well-known standardized image compression technique. JPEG loses information, so the decompressed picture is not the same as the original one. By adjusting the compression parameters, the degree of loss can be adjusted. The wide use of JPEG is because of two fundamental reasons: reducing the size of image files and storing full color information. JPEG is an eminent format and is described in many places, e.g., Refs. 22 and 23.

The JPEG standard is based on the discrete cosine transform (DCT) paradigm.[20,21] The DCT changes the picture into frequency space. The frequency coefficients, which are very low magnitude, are rounded to zero. When most of the coefficients in a block are zero or of very low magnitude, the compression algorithm will give a very short bit sequence. Zero sequences are treated very efficiently by JPEG compression and the results will be only a few bytes.

When there is a drastic change in a block of 8×8, the value of many frequency coefficients will be high. Such a sequence will be compressed into many more bits. JPEG's standard stipulates that the block's size will be 8×8 pixels, but the algorithm will be obviously good for other small $N \times N$ pixel sizes, too.

When there is a good algorithm that is very common in the market and does a good job, it seems to be the logical choice. There might be some other good algorithms, but using those algorithms means you have to push a new standard, which can be a very hard task.

When looking for the contour of an object, the goal will be to find the object's border. The idea is to break the compressed file into its original blocks and then look in the compressed file for long bit sequences. The blocks that are compressed into long bit sequences are presumed to be the object's border. In our implementation we took a simpler approach. We take many pictures of each tire. We actually take a close picture of each part of the tire. If the entire size of a picture is above a certain threshold, we will consider this part of the tire as a damaged part.

If we have no idea what the threshold value should be, we can examine the probability density function (PDF) of the block representation to select a suitable value. In the uncomplicated case, the PDF should be mono-modal and we set the value in the inflection point.

Many pictures are taken by rolling the tire, and in each picture, suspected objects are searched.

Note the method does not use the spatial information in the blocks, so it would identify any sequence of bits as a contour of the object, which is a weakness of the algorithm. Our assumption is that the resolution is large enough to describe the shape's character.

Figure 1 demonstrates how JPEG is used for contour extraction. The original image was compressed in grayscale baseline JPEG format with 75% quality. The left image shows the original image, which is a high-resolution picture of 1000×1000 pixels. The upper-right image shows the JPEG format in the white or black area. The lower-right image shows the JPEG format in the upper-left corner of the black square. The size of the black square is 200×200 pixels and the square is not aligned relative to JPEG's 8×8 blocks.

The JPEG file reports the difference in magnitude between the DC's coefficients of a previous block relative to the current block. In the case of a white or black area, there are no changes in the coefficients' magnitudes. This type of block is encoded as 6 bits by the JPEG standard. The output of the JPEG file is shown in Fig. 1. The "00" reflects that there are no differences between the values of the previous and the current block's DC coefficients, and "1010" symbolizes the end of the block. If there is a difference between the intensity of the DC coefficients of the previous and the current blocks, the size of the encoding block will be slightly bigger. For example, a block that encodes a sharp change from white to black is represented by a wide range of frequency coefficients. It is easy to select a threshold that delimits the edges of the shape from the rest of the image.

Figure 1 shows a sample of the block, which contains the upper-left corner of the black square. In order to compress these values in JPEG standard, 243 bits are needed. The difference between 6 and 243 is obviously significant. By using three parameters—the length of the block, its magnitude, and the number of consecutive blocks—the threshold can extract the contour with a range of scalar values.[24,25] The extra parameters allow more control over the resulting mechanism.

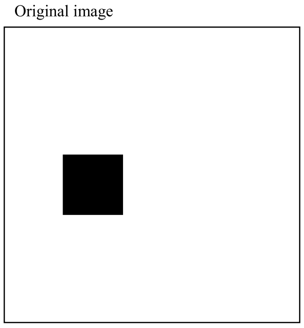
The algorithm is very simple and can be described as follows:

- Take a picture of the tire and create an image I
- Set $L = \text{size\_of\_image}\ (I)$
- If $(L > T)$
- then the tire is deemed to be damaged and should be checked.

JPEG format of uniform area

0	0	1	0	1	0

Original image



JPEG format of the upper left corner of the black square

-12	18	-8	-2	3	-1	-1	1
22	-25	8	2	-3	0	1	-1
-18	20	-7	-2	2	0	-1	1
16	14	4	1	-1	0	1	1
-11	10	-2	-1	0	-1	-1	0
6	-5	1	-3	0	0	0	0
-2	2	-1	0	0	0	0	0
1	-1	0	0	0	0	0	0

Fig. 1 Sample image and how JPEG can be used for contouring.

Table 1 Number of inspected damaged tires.

Damage size	Number of instances
<1	125
1 to 1.5	242
1.5 to 2	217
2 to 2.5	327
2.5 to 3	84
>3	37

Here T is a threshold and its value will be discussed below.

3 Results

We examined our technique on 1032 tires and succeeded in finding most of the damaged tires. Clearly, obvious cases, as in Fig. 1, do not exist in real tires, but we still succeeded in finding most of the damaged tires by rolling the tire and taking many pictures of it in all of its parts.

Table 1 details the number of tires that were examined, listed by the magnitude of the damage.

We used an Olympus FE-170 digital camera with resolution of 2112×2816. The images with no damage in the tire produced images of size less than 1.3 MB; whereas images of size more than 1.5 MB usually had damage in the tire.

Figure 2(a), 2(b), and 2(c) shows some damaged tires that our system has found. All of these pictures were on vehicles of volunteers. Driving a vehicle with such damages can be a substantial risk of life.

The images between 1.3 and 1.5 MB sometimes had a small scratch and sometimes it was only a change in the color shade. Also the caption on the tire can increase the size of the image; however, the caption typically increases the image size less than the increase that a real damage makes, but it may make the size above 1.3 MB.

Figure 3(a) and 3(b) shows two cases where a change in the color shade of the tire made the result above the threshold of 1.3 MB and caused an uncertain result.

There are also cases where our system has produced false alarms. Figure 4 shows an example of such a false alarm. The dirt on the tire in Fig. 4 can be easily noticed. This dirt made the system assume that the tire is damaged, whereas this tire was undamaged and the vehicle was safe for driving.

Therefore, we recommend washing the tire before the system tests the vehicle. Dust, dove feces, mud, and any other dirt can mislead the system, especially if the dirt is white.

Figure 5 shows the statistics of the experiment results. The graph includes all the cases where the image was larger than 1.3 MB. The graph details the cases of image size less than 1.5 MB (uncertain) and image size more than 1.5 MB (success or false positive). The graph specifies the instance distribution of each magnitude of the damage. The magnitude of the damage was calculated according to the longest measurement of damage regardless of its shape.

(a)

(b)

(c)

Fig. 2 Damaged tire.

It can be seen that large damages can be confidently detected. Smaller damages can be detected with a high probability of a correct detection, but the accuracy is less precise.

There were only four cases when the image was less than 1.3 MB, although there was a real damage in the tire. In all of these cases, the damage was very superficial and the magnitude of the damage was in three cases less than 1 cm and only in one case it was 1.2 cm.

Figure 6 details the reasons for images to be larger than 1.3 MB. Mainly the reason is a real damage, but in some cases the reason was something else. A dirty tire is the recurring explanation, but also a substantial caption can explain a

(a)

(b)

Fig. 3 Tire with a change in the color shade.

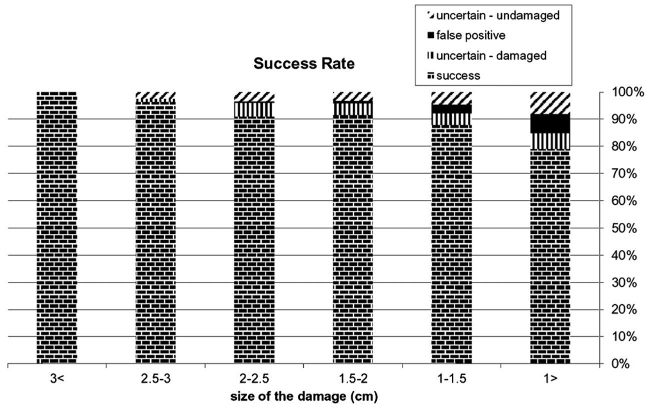


Fig. 5 Detection success rate.

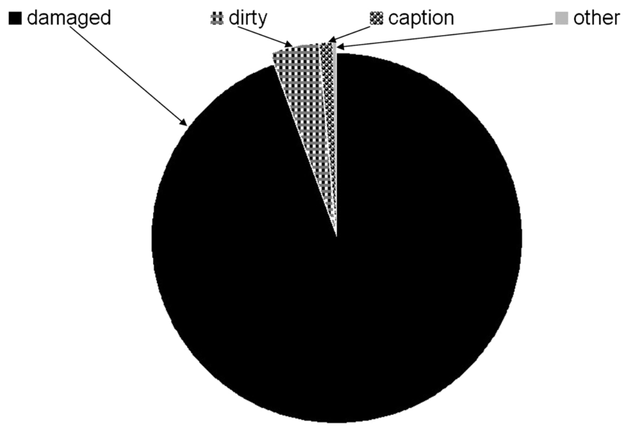


Fig. 6 Distribution of images larger than 1.3 MB.

larger image. In Fig. 6, if two reasons for a larger file come about, then the image will be added to the group with the higher possibility to occur.

We have not been concerned about motion blur effects. A shutter speed of 1/60 s aims at capturing subjects moving slower than 60 miles per hour, while nowadays an average camera supports 1/1000 s. A better camera like Canon EOS 1D (announced in 2001) or Nikon D1 (announced in 1999) can support a shutter speed of 1/16,000 s, so the motion blur should not be a problem unless someone explicitly adjusts his camera to a low shutter speed.

Fig. 4 Dirt on the tire.

The quality factor of JPEG is used to produce two quantization tables—one for the luminance (brightness) information and the other for the chrominance (color) information. Applications that support Independent JPEG Group (IJG) produce quantization tables according to the IJG specification; however, many other applications employ different quantization tables. In this paper we have used IJG quantization tables.

In Fig. 7 the same scratch has been gone through several quantization tables. Figure 8 shows the file size of the same scratch of Fig. 7 for different quantization factors. A human eye can barely notice changes when the quantization factor is 50% or more. However, the image size is noticeably changed when the quantization factor is changed from 50% to a higher value as can be seen in Fig. 8.

When the quantization factor is low, the AC coefficients of the JPEG image will be divided by very large numbers and the system can take almost no notice of even fairly sharp changes. It can be seen in Fig. 7 how some 8×8 blocks become blocks of just one color (the average color of the original block).

In our system, we obviously preferred the highest quality factor (100%), so as to get the best distinguishability.

4 Motion JPEG-2000

JPEG-2000 is a new standard for image and video compression.[26–30] Several of the latest digital cameras support JPEG-2000 and motion JPEG-2000 (e.g., Refs. 31 to 34). Motion

Fig. 7 Pictures of the same scratch in various qualities. First row—original picture, 50%, 25%. Second row—12%, 6%, 3%.

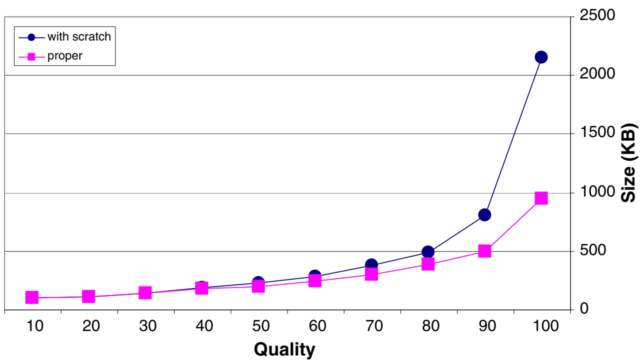


Fig. 8 Changes of image size of a proper/damaged tire as a function of the quantization factor.

JPEG is employed in numerous applications from professional video editing to capturing cards in many hardware settings.

By means of JPEG-2000, an image stored on a disk has lower quality levels embedded in the compressed file. If a user asks to preview a JPEG-2000 image, only a lower quality image will be restored. No modification of compressed data is needed. If the user decides to see a better quality image, further information will be restored from the JPEG-2000 image file. There will be no need to redownload the entire image. The overall quality of a JPEG-2000 image is typically higher than a traditional JPEG image. In addition to the advantages of improved quality and scalability, JPEG-2000 also generates about 20% better compression ratio than JPEG and in several cases JPEG-2000 even generates a considerably better compression ratio.

Unlike the Moving Picture Experts Group (MPEG), by means of motion JPEG-2000 encodes each frame independently. MPEG employs three kinds of frames for encoding.

- *I*-frames do not need other video frames to be decoded, but the compression ratio is not as good as the other frame types.
- *P*-frames employ data of previous frames to be decoded and the compression ratio is typically better than *I*-frames.
- *B*-frames employ data of both previous and forward frames for the decoding and typically obtain the best compression ratio.

The three types of frames allow improved compression efficiency but the coding technique is more complex and therefore requires more computation time. Motion JPEG-2000 employs only intraframe blocks, which lets the user have the ability to randomly access a file at any fame. In addition, the usage of only intraframe blocks reduces the complexity of the compression and decompression processes and therefore the decoding time is shorter.

There is another major modification in JPEG-2000. The designers of JPEG-2000 came to a decision to depart from the block-based DCT coding used by the traditional JPEG and MPEG standards in favor of a wavelet-based compression—the discrete wavelet transform (DWT). The DWT provides improved quality than JPEG and also provides scalability without having to store surplus data.

The forward DWT at the encoder is successive functions of a pair of low-pass and high-pass filters, followed by division by a factor of two after each filtering function so that odd indexed samples will be discarded. The low-pass and high-pass filter pair is known as analysis filter-bank. The low-pass filter keeps the low frequencies, whereas the high frequencies are significantly diminished and therefore the result is only a vague form of the original signal. Conversely, the high-pass filter keeps only the high frequencies of a signal such as contours, whereas the low frequencies are significantly diminished.

The standard of motion JPEG-2000 is extremely suitable for the technique of this paper. Since the high-pass filter generates much more bits for contours, the compressed data of such a region in the picture will be larger. This feature can be used as it has been used with the DCT of traditional JPEG. Furthermore, there is no need to extract the high-quality picture. The high-pass filter detects a contour even in a low-quality picture. In addition, since motion JPEG-2000 employs only I-frame blocks, the technique becomes even more straightforward because there is no need to process the frame along with nearby frames. Each I-frame contains all the information needed for the damage detection.

However, there is a major difficulty to implement this method with contemporary video hardware. Some research has been conducted with the purpose of finding the threshold of rate for distinguishing successive still images from a proper continuous video that a human eye would not be able to detect (e.g., Refs. 35 and 36). Researchers are commonly consistent with the rate of 10 to 20 frames per second. Therefore, even though there are some exceptional plentiful frames per second cameras,[37] the common up to date hardware can support only about 30 frames per second in common image format size (352×288 pixels) and about 20 frames per second for standard definition television size resolution (760×480 pixels).[38] Pushing a new standard of digital cameras can be full of twists and turns; however, the common standard is not enough for a tire of a moving vehicle. For example, an everyday private car having ordinary tires of 225/45R17 traveling at a speed of 72 kilometers per hour will wholly rotate the tire 10 times each second. For such a system, 30 frames per second will not be enough. However, we believe the hardware will be improved and in the future this method will be feasible.

5 Conclusions and Future Work

Rollover accidents are very dangerous and cause too much loss of life. Since many of the rollover accidents are a result of damaged tires, keeping atire in good shape is more than important. Although there are reports arguing that the human factor is the main reason for rollover accidents,[39] we would not advise any driver to pay no attention to the shape of his tires. Additionally, automatic inspection of aircraft tires is critical and vital as was noted in Ref. 40. The suggested system in this paper can be mounted in an aircraft and it may also save lives.

In this paper, we explained how a system using simple equipment—a digital camera—can detect damages in a tire sidewall. In the future we would like to adapt our system to a moving vehicle. Conventional digital cameras cannot take a picture of such a fast-moving object, so we should find a practical way for the pictures to be taken. A possible solution might be taking the pictures when the vehicle moves slowly before or after stopping. In addition, we would like to adjust our system to be able to detect damages in the tire tread and also in the tire inner liner. We will also consider developing a more sophisticated algorithm that can not only identify damaged tires but also identify various kinds of damages. Looking at existing machine-vision[41–43] and machine-learning algorithms[44,45] may help in significantly extending the suggested method.

As a final point, our suggestion to mount a camera with the aim of becoming aware of below-standard tires is easy for implementation.[46] Such a simple device can be very beneficial because detecting damages in tires before an accident happens can save lives.

Acknowledgments

This work was supported in part by SUN Microsystems. We would like to thank SUN Microsystems for their donation.

References

1. K. Aldana, "Consumer advisory: motorists should check tires in hot weather to improve safety," http://www.nhtsa.gov/About+NHTSA/Press+Releases/2011/Consumer+Advisory:+Motorists+Should+Check+Tires+in+Hot+Weather+to+Improve+Safety (June 2011).
2. K. Hirose, T. Toriu, and H. Hama, "Detection of tire-road contact point for vehicle position estimate considering shape distortion in a circular fisheye image," in *2009 Fifth Int. Conf. on Intelligent Information Hiding, and Multimedia Signal Processing*, pp. 178–181, IEEE, Kyoto, Japan (2009).
3. S. C. Ergen et al., "The tire as an intelligent sensor," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **28**(7), 941–955 (2009).
4. F. Gustafsson et al., "Virtual sensors of tire pressure and road friction," presented at *SAE 2001 World Congress*, SAE International, Detroit (2001).
5. J. M. Giustino, "System and method for predicting tire forces using tire deformation sensors," U.S. Patent 6550320 (2000).
6. W. H. Capen et al., "Tire inspection apparatus," U.S. Patent No. 2301251 (1939).
7. F. S. Wright et al., "Tilt head camera for interferometric analysis of tires," U.S. Patent No. 4392745 (1981).
8. W. H. Gray, C. Dumont, and M. A. Abidi, "Simulation of a tire inspection system," in *Proc. of the 5th Conf. on Quality Control by Artificial Vision*, pp. 171–175, Canadian Image Processing and Pattern Recognition Society, Quebec, Canada (1999).
9. T. Schumacher, C. C. Higgins, and S. C. Lovejoy, "Detection of vehicles with studded tires using acoustic emission sensors mounted to highway bridges, *J. Transp. Eng.* **136**(5), 480–487 (2010).
10. A. Wilson, "3-D vision system checks tires," *Vis. Syst. Des.* **11**(2), 31–35 (2006).
11. B. Orhan et al., "Tire classification from still images and video," in *Proc. 15th Int. IEEE Conf. on Intelligent Transportation Systems*, pp. 485–490, IEEE, Shanghai, China (2012).
12. L. Feng, L. Chu, and S. Zechang, "Intelligent vehicle simulation and debugging environment based on physics engine," in *Proc. Int. Asia Conf. on Informatics in Control, Automation and Robotics*, pp. 329–333, IEEE, Bangkok, Thailand (2009).
13. I. Grinberg and Y. Wiseman, "Scalable parallel collision detection simulation," in *Proc. Signal and Image Processing*, pp. 380–385, IASTED, Honolulu, Hawaii (2007).
14. W. Yong, Z. Jing, and W. Ping-bo, "Dynamics simulation model of double united articulated container flat vehicle," *J. Traffic Transport. Eng.* **8**(1), 1–4 (2008).
15. C. Tao and W. E. I. Lang, "Study and implementation on JPEG image enhancement system of tire mark," in *Proc. Critical Issues in Transportation Systems Planning, Development and Management*, pp. 656–661, American Society of Civil Engineers, Harbin, China (2009).
16. N. Khanna et al., "Forensic techniques for classifying scanner, computer generated and digital camera images," in *Proc. of the 2008 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1653–1656, IEEE, Las Vegas (2008).
17. P. Hongyan, H. Hong, and J. Hengtian, "Drive design for ship GPS navigation equipment based on Linux operating system," in *Int. Conf. on Educational and Network Technology*, pp. 384–388, IEEE, Qinhuangdao, China (2010).
18. S. T. Klein and Y. Wiseman, "Parallel Huffman decoding with applications to JPEG files," *Comput. J.* **46**(5), 487–497 (2003).
19. Y. Wiseman, "Burrows-wheeler based JPEG," *Data Sci. J.* **6**, 19–27 (2007).
20. S. T. Klein and Y. Wiseman, "Parallel Lempel Ziv coding," *J. Discrete Appl. Math.* **146**(2), 180–191 (2005).
21. Y. Wiseman, "A pipeline chip for quasi arithmetic coding," *IEICE J. Trans. Fundamentals* **E84-A**(4), 1034–1041 (2001).
22. The International Telegraph And Telephone Consultative Committee, Information Technology—Digital Compression and Coding of Continuous-Tone Still Images Requirements and Guidelines—International Standard ISO/IEC 10918-1, (1993).
23. G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM* **34**(4), 30–44 (1991).
24. Y. Wiseman and E. Fredj, "Contour extraction of compressed JPEG images," *ACM J. Graphic Tools* **6**(3), 37–43 (2001).
25. E. Fredj and Y. Wiseman, "An O(n) algorithm for edge detection in photos compressed by JPEG format," in *Proc. IASTED International Conference on Signal and Image Processing*, pp. 304–308, IASTED, Honolulu, Hawaii (2001).
26. D. S. Taubman and M.W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards, and Practice, Kluwer International Series in Engineering, and Computer Science*, Springer, Norwell, MA (2002).
27. C. A. Christopoulos, A. N. Skodras, and T. Ebrahimi, "The JPEG 2000 still image coding system: an overview," *IEEE Trans. Consum. Electron.* **46**(4), 1103–1127 (2000).
28. A. N. Skodras, C. Christopoulos, and T. Ebrahimi, "JPEG 2000: the upcoming still image compression standard," *Pattern Recogn. Lett.*, **22**, 1337–1345 (2001).
29. A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Process. Mag.* **18**(5), 36–58 (2001).
30. B. E. Usevitch, "A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000," *IEEE Signal Process. Mag.* **18**(5), 22–35 (2001).
31. H. Yamauchi et al., "A single chip JPEG2000 encode procesor capable of compressing D1-images at 30 frame/sec without tile division," *IEICE Trans. Elec.* **E87-C**(4), 448–456 (2004).
32. H. C. Fang et al., "81 MS/s JPEG 2000 single-chip encoder with rate-distortion optimization," in *IEEE Int. Solid-State Circuits Conf.*, pp. 328–531, IEEE, San Francisco (2004).
33. L. Liu et al., "A VLSI architecture of JPEG2000 encoder," *IEEE J. Solid-State Circuits* **39**(11), 2032–2040 (2004).

34. Y. H. Seo and D. W. Kim, "VLSI architecture of line-based lifting wavelet transform for motion JPEG2000," *IEEE J. Solid-State Circuits* **42**(2), 431–440 (2007).
35. I. S. MacKenzie and C. Ware, "Lag as a determinant of human performance in interactive systems," in *Proc. of the INTERACT '93 and CHI '93 Conf. on Human Factors in Computing Systems*, pp. 488–493, ACM, Amsterdam, Netherlands (1993).
36. D. A. Kaff et al., "Application of the QuO quality-of-service framework to a distributed video application," in *Proc. of 3rd Int. Symp. on Distributed Objects and Applications*, pp. 299–308, IEEE, Rome, Italy (2001).
37. C. T. Chin et al., "Brandaris 128: a digital 25 million frames per second camera with 128 highly sensitive frames," *Rev. Sci. Instrum.* **74**(12), 5026–5034 (2003).
38. T.-H. Tsai and L.-T. Tsai, "An efficient design for motion-JPEG2000 system in real-time video encoding," *J. Circuits, Syst., Comput.* **17**(4), 597–610 (2008).
39. C. M. Farmer and A. K. Lund, "Rollover risk of cars and light trucks after accounting for driver and environmental factors," *Accid. Anal. Prev.* **34**(2), 163–173 (2002).
40. A. J. Stolzer, C. D. Halford, and J. J. Goglia, "Safety management systems in aviation," Chapter 3 in *Principles of Quality Management*, MPG Books Group, United Kingdom (2010).
41. E. Dickmanns, "The development of machine vision for road, vehicles in the last decade," in *Proc. IEEE Intelligent Vehicle Symp.*, IEEE, Versailles, France (2002).
42. J. Wang, G. Bebis, and R. Miller, "Overtaking vehicle detection using dynamic and quasi-static background modeling," in *Proc. IEEE Workshop Machine Vision for Intelligent Vehicles*, IEEE, San Diego, CA (2005).
43. Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(5), 694–711 (2006).
44. W. von Seelen et al., "Scene analysis and organization of behavior in driver assistance systems," in *Proc. IEEE Int. Conf. Image Processing*, pp. 524–527 (2000).
45. Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using evolutionary Gabor filter optimization," *IEEE Trans. Intell. Transp. Syst.* **6**(2), 125–137 (2005).
46. Y. Wiseman, "Take a picture of your tire!," in *Proc. IEEE Conf. on Vehicular Electronics and Safety*, pp. 151–156 (2010).

Yair Wiseman got his PhD from Bar-Ilan University and completed two postdocs—one at the Hebrew University of Jerusalem and one in Georgia Institute of Technology. His research interests include vehicular systems, intelligent transportation systems, process scheduling, hardware software codesign, memory management, and real-time operating systems.