Computerized Traffic Congestion Detection System

Yair Wiseman[1],

[1] Computer Science Department
Ramat-Gan 52900, Israel
wiseman@cs.biu.ac.il

Abstract. Congestion roads can cause many unwelcome results like making people late to work; deliveries do not arrive on time and more. One way to tackle this annoyance is letting the driver know which roads are congested, so they will be able not to go through them. This paper suggests a fast computerized method of concluding which roads are congested and which ones are vacant by automatic analyzing of the roads' images and getting a full map of the roads in no time.

Keywords: Traffic Congestion, Discrete Cosine Transform, JPEG

1 Introduction

Traffic congestion is generated when the number of vehicles passing through the road exceeds the road traffic capacity. Accordingly, traffic congestion can be produced because of an increase in the number of vehicles or a decrease in the capacity of the road.

An increase in the number of vehicles is generated when many vehicles go in a same road to the same direction. Such a circumstance happens for example in the morning when many employees go to employment centers, or in the evening when these employees go back from the employment centers to their homes. Also, a mass event can cause traffic congestion in the vicinity of the event.

A decline in road traffic capacity can be caused by various reasons like road construction, vehicle accidents or police checkpoints.

The intensity of traffic congestions in the roads is very essential information for drivers [1,2]. We suggest using a simple digital camera to locate the traffic congestions. Almost all digital cameras can produce JPEG images. JPEG is a very common method for image compression and it is also extensively used by electronic devices like scanners and digital cameras [3] as well as vehicle equipment like GPS [4].

JPEG images have many benefits like the capability of being decoded in parallel [5], the straightforwardness of adaptation for new compression techniques [6] and the potential of easy and flexible implementation for hardware from different vendors [7,8].

Since most of the data in an image is unnoticeable for a human eye and the difference between a bitmap image size and a compressed image size is enormous, images are typically stored in a compressed standard. The compression standards not only compress the data, but also discard some of the data that a regular human eye will not be aware of.

A naive approach for image processing of compressed images would be decompressing the image and subsequently running the image processing algorithm on the original image data. Instead, a better technique for some image operations, we can directly work on the compressed data. This direct work gives us two advantages: first, we can use a standard digital camera without a need to adjust the digital camera; second, the signal processing used by JPEG can let us use the frequency information embedded in the compressed data of JPEG.

The rest of the paper is organized as follow: Section 2 explains how JPEG can be used for detecting traffic congestions; section 3 describes the experiments we have performed while section 4 concludes the paper.

2 Computerized Traffic Congestion Dedection Using JPEG

JPEG is a well-known standardized picture compression technique. JPEG discards information, so the decompressed picture is not the same as the original one, but usually the changes are unnoticeable. By adjusting the compression parameters, the degree of discarding can be attuned. The wide usage of JPEG is because of two fundamental reasons: storing full color information and a significantly reduce of image file sizes, so as to avoid high traffic on networks and memory pressure [9]. JPEG is a popular format and is described in many places e.g. [10,11,12].

The JPEG standard is based on the DCT algorithm [13]. The DCT transforms the picture into frequency space. If there are frequency coefficients which have a low value, they will be rounded to zero. If most of the DCT coefficients in a block are zero or close to zero, JPEG compression algorithm will yield a very short bits sequence for this block. Zero sequences are treated very efficiently by JPEG compression algorithm and the compressed file will be only few bytes.

JPEG splits the picture into blocks of 8X8 pixels. The order of these blocks in the compression algorithm is line by line and each line is read from left to right [14,15]. When there is a drastic shade change in such a block of 8X8, the value of many DCT frequency coefficients will be high. Such a sequence will be compressed into many more bits than a sequence of zeros.

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

When looking for many vehicles in a picture vs. picture with few vehicles, the goal will be to detect the vehicles. The idea is to break the compressed file into its original JPEG blocks of 8X8, then looking in the size of the compressed file. In our implementation we took a straightforward approach. We take pictures of the road. Actually, we take a close picture of each segment of the road. If the entire size of a picture is above a certain threshold, we will consider this segment of the road as a congested segment.

If we are not sure what the threshold value should be, we can examine the probability density function (PDF) of the block representation in order to select a suitable value. In a simple case the PDF should be mono-modal and we can set the value in the inflection point [16,17].

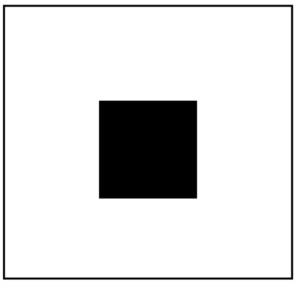

Fig. 1. Sample image and how JPEG can be used for contouring.

-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

Table 1. JPEG format of the upper left corner of the black square.

Figure 1 demonstrates how different JPEG works for a block with a sharp change vs. a smooth block. The original picture in Figure 1 was compressed in grayscale baseline JPEG format with 75% quality. The figure shows the original picture, which is a high-resolution picture of 1000X1000 pixels. Table 1 shows the JPEG block containing the upper left corner of the black square. The size of the black square is 200X200 pixels and the square is not aligned relative to JPEG's 8X8 blocks.

JPEG does not save the DC values themselves, but rather JPEG saves the difference of magnitude between the DC's coefficient of a previous block relative to the DC of the current block [18,19]. In the case of a smooth white or a smooth black area there will be no changes in the coefficients' magnitudes. This type of block is encoded as six bits by the JPEG standard:

0, 0,1,0,1,0.

The "00" reflects that there is no difference between the value of the previous block's DC coefficient and the current block's DC coefficient, and "1010" denotes the end of the block. If there is a difference between the intensity of the values of the previous and the current blocks, the size of the encoded block will be considerably larger. For example, a block which encodes a sharp change from white to black is represented by a wide range of frequency coefficients as can be seen in Table 1.

Consequently, the threshold that delimits the edges of an object is usually clear-cut [20,21,22].

In the example of Figure 1 and Table 1 which contains the upper left corner of the black square, the threshold is uncomplicated. In order to compress the values of table 1 in JPEG standard, 243 bits are needed. The difference between 6 bits that are need to the smooth blocks to 243 bits that are needed to the blocks with the sharp changes is obviously significant. By using the length of the block, we can extract the contour of objects [23,24]. If there are many contours of vehicle in one picture, it will mean that there are many vehicles in the pictures.

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

- Take a picture of the road segment and create an image - I
  - Set L= size_of_image (I)
  - If (L > T)
    - then the road segment is deemed to be congested.

Where T is a threshold and its value is discussed above.

3 Experiments

We examined our technique on some road segments and tried to check whether we succeed to realize the intensity of congestion in each road segment. Clearly, smooth and sharp changes like in Figure 1 do not exist in real roads, but we have still succeeded realizing the roughly intense of the congestion in each of the segments we have examined.

Figure 2 is two picture of the same road in Los Angeles – one of the most congested cities in the world. The left picture is when the road was very congested and the right picture is when the traffic in the road was sparse.

Fig. 2. Congested road vs. vacant road

We used a Nikon D5300 digital camera with resolution of 24.2 MP. Images containing sparse traffic produced files in size of less than 4.2MB; whereas images of congested roads typically produced files in size of more than 5.7MB.

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 IJG (Independent JPEG Group) 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. Actually, in our system, we obviously preferred the highest quality factor (100%), so as to get the best distinguishableness, which means in IJG that the entire quantization table is filled with 1s i.e. the frequency coefficients are not divided.

There are some cases which are hard to detect. A jam in one lane which is usually the turn right lane as can be seen in Figure 3 is hard to detect. There are many vehicles on one side of the road but the other side of the road is vacant which leads to an ambiguous situation. The size of the image is quite similar to an image with a moderate congestion; however it is hard to find the difference only by the JPEG image file size.

Fig 3. Road with a congested lane

There are also cases where our system has produced false alarms. Figure 4 shows an example for such a false alarm. The hail on the road in Figure 4 can be easily noticed. This hail made the system assuming that the road is congested; whereas this road was actually vacant as can be seen in this figure. Light snow also can produce such a false alarm; however, heavy snow will make the entire road white and as a result the changes in the shade will be tiny and the image will be small.

Fig 4. Road with hail

4 Conclusions

Congestion road map is used by many application e.g. Waze [25] or Google Traffic [26]. Most of them use cellular phones [27]. This paper explains how a system using simple equipment – a digital camera that can generate JPEG images is able to detect traffic congestions and create such a map.

The signal processing method of JPEG assumes that the taken images are to a certain extent smooth. As a result when there is a sharp change in a particular block of an image, the value of many of its frequency coefficients will be high and it will be compressed into many more bits. If there are many such sharp changes, the size of the entire compressed image file will be significantly larger; therefore, if the image of a road is overly large, an embedded computer system can understand that this road is congested.

References

1. R. Bauza, J. Gozalvez and J. Sanchez-Soriano, "Road traffic congestion detection through cooperative vehicle-to-vehicle communications", Proc. 35th IEEE Conference on Local Computer Networks (LCN), pp. 606-612, 2010.
2. K. Mandal, A. Sen, A. Chakraborty, S. Roy, S. Batabyal and S. Bandyopadhyay, "Road traffic congestion monitoring and measurement using active RFID and GSM technology", Proc. 14th IEEE International Conference on Intelligent Transportation Systems (ITSC), pp. 1375-1379, 2011.
3. Khanna, N., Chiu, G. T. C., Allebach, J. P. and Delp, E.J., "Forensic techniques for classifying scanner, computer generated and digital camera images", Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008), pp. 1653-1656, 2008.
4. P. Hongyan, H. Hong, J. Hengtian, "Drive design for ship GPS navigation equipment based on Linux operating system", International Conference on Educational and Network Technology (ICENT), pp. 384-388, Qinhuangdao, China, June, 2010.
5. Klein S. T. and Wiseman Y., "Parallel Huffman Decoding with Applications to JPEG Files", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 46(5), pp. 487-497, 2003.
6. Wiseman Y., "Burrows-Wheeler Based JPEG", Data Science Journal, Vol. 6, pp. 19-27, 2007.
7. Wiseman Y., "A Pipeline Chip for Quasi Arithmetic Coding", IEICE Journal - Trans. Fundamentals, Tokyo, Japan, Vol. E84-A No.4, pp. 1034-1041, 2001.
8. R. Ben Yehuda and Y. Wiseman, "The Offline Scheduler for Embedded Vehicular Systems", International Journal of Vehicle Information and Communication Systems, Vol. 3(1), pp. 44-57, 2013.
9. Reuven M. and Wiseman Y., "Medium-term scheduler as a solution for the thrashing effect", The Computer Journal, Vol. 49(3), pp. 297-309, 2006.
10. Information Technology – Digital Compression and Coding of Continuous-Tone Still Images Requirements and Guidelines – International Standard ISO/IEC 10918-1, 1993.
11. Wallace G. K., The JPEG Still Picture Compression Standard Communication of the ACM 34, pp. 3-44, 1991.
12. Y. Wiseman, "The still image lossy compression standard – JPEG", Encyclopedia of Information and Science Technology, IGI Global, Third Edition, Vol. 1, Chapter 28, pp. 295-305, 2014.
13. R. Shan, C. Wang, W. Huang and X. Zhou, "DCT-JPEG Image Coding Based on GPU", International Journal of Hybrid Information Technology, Vol. 8, no. 5, SERSC, pp. 293-302, 2015.
14. Y. Wiseman, "Enhancement of JPEG Compression for GPS Images", International Journal of Multimedia and Ubiquitous Engineering, Vol. 10(7), pp. 255-264, 2015.
15. Y. Wiseman, "Improved JPEG based GPS picture compression", Advanced Science and Technology Letters, Vol. 85, pp. 59-63, 2015.
16. Y. Wiseman, "Device for Detection of Fuselage Defective Parts", Information Journal, Tokyo, Japan, Vol. 17(9(A)), pp. 4189-4194, 2014.
17. Y. Wiseman, "Fuselage Damage Locator System", Advanced Science and Technology Letters, Vol. 37, pp. 1-4, 2013.
18. Y. Wiseman, "Alleviation of JPEG Inaccuracy Appearance", International Journal of Multimedia and Ubiquitous Engineering, Vol. 11(3), pp. 133-142, 2016.
19. Y. Wiseman, "Diminution of JPEG Error Effects", The Seventh International Conference on Future Generation Information Technology, Vol. 117, pp. 6-9, 2015.

20. Y. Wiseman, "Take a Picture of Your Tire!", Proc. IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010) Qingdao, ShanDong, China, pp. 151-156, 2010.
21. Y. Wiseman, "The Effectiveness of JPEG Images Produced By a Standard Digital Camera to Detect Damaged Tyres", World Review of Intermodal Transportation Research, Vol. 4(1), pp. 23-36, 2013.
22 . Y. Wiseman, "Camera That Takes Pictures of Aircraft and Ground Vehicle Tires Can Save Lives", Journal of Electronic Imaging, Vol. 22(4), 041104, 2013.
23. Wiseman Y. and Fredj E., "Contour Extraction of Compressed JPEG Images", ACM - Journal of Graphic Tools, Vol. 6 No. 3, pp. 37-43, 2001.
24. Fredj E. and Wiseman Y., "An O(n) Algorithm for Edge Detection in Photos Compressed by JPEG Format", Proc. IASTED International Conference on Signal and Image Processing SIP-2001, Honolulu, Hawaii, pp. 304-308, 2001.
25. T. H. Silva, A. A. F. Loureiro, J. Salles, A. C. Viana, P. O. S. Vaz de Melo, J. M. Almeida, "Traffic Condition Is More Than Colored Lines on a Map: Characterization of Waze Alerts", Proc. International Conference on Social Informatics, pp. 309-318, Springer International Publishing, Kyoto, Japan, 2013.
26. NCTA – The Internet & Television Association, "How Google Tracks Traffic", available online at: https://www.ncta.com/platform/broadband-internet/how-google-tracks-traffic/ , 2013.
27. T. Roopa, A N. Iyer and S. Rangaswamy, "CroTIS-Crowdsourcing Based Traffic Information System", Proc. IEEE International Congress on Big Data, Santa Clara, CA, USA, pp. 271-277, 2013.