Journal of Engineering Science and Technology
Vol. 12, No. 12 (2017) 3215 - 3225

DEVELOPING AND VALIDATING A REAL TIME VIDEO BASED TRAFFIC COUNTING AND CLASSIFICATION

ALI E. JEHAD[1,*], RIZA A. O.K. RAHMAT[2]

[1, 2] Department of Civil & Structural Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia, Bangi, Malaysia
*Corresponding Author: aliemad87@gmail.com

Abstract

An algorithm program was developed to detect vehicles in traffic videos and get the vehicle count for the small time period as a tool that can assist researchers who seek vehicle counting. This system approach has been presented for extracting traffic data using video image processing. Meanwhile, an offline program focuses on extracting vehicles, tracking them and providing the vehicle count for a short period of time. It uses background subtraction, shadow removal, and pixel analysis for extracting moving objects. The results show that the algorithm is capable of counting 95% of the vehicles due to some shaking in the video feed. These data have been analysed by the paired samples *t*-test to show the credibility of the results which have been approved to be useful according to the values of *correlation* and *P-value* compared with the values of the observation method. Also, the classification of vehicles was performed using *the improfile command* in Matlab-Video Image Processing that computes the colours intensity values along a line or a multiline path in an image.

Keywords: Video image processing, Algorithm system, Matlab, optical flow model, Vehicles classification.

1. Introduction

Speed, flow, and density are macroscopic parameters for characterizing the traffic stream as a whole, while headway and spacing are microscopic measures for distinguishing individual vehicles [1]. High demands for computer algorithms and technological solutions have been raised due to traffic analysis and monitoring in a real-time mode via computer vision techniques [2]. Vehicle tracking is one of the most convincing applications lies in initiating a track automatically.

During the last years, there have been tremendous increases in road traffic flow. This increasing had brought into focus the best tool for measuring the traffic volume

Nomenclatures

>=	Relational operator
$\|u\|^2$	Orders of magnitude function
B	Blue colour
B'	Blue colour intensity
$B(i, j)$	Background image frame
$B(p)$	Background vector
d	Mean difference between two samples
d_k	Tracking vehicle
$d(i,j)$	Foreground difference
G'	Green colour intensity
G	Green colour
h	Change in light
Improfile	Pixel-value cross-sections along line segments code
I	Input matrix
Img	Image
$I_t(i,j)$	Current pixel frames
$I_{t-1}(i,j)$	Previous pixel frames
I_x,I_y,I_t	Spatiotemporal image brightness derivatives
$I(i,j)$	Foreground pixel
$I(p)$	Colour vector
N	Sample size
n	Frame
n_o	Number of observations (experimental data points)
P-value	Function of the observed sample results
R	Red colour
R'	Red colour intensity
r	Light ratio
T	Pre-set threshold
t	Size of the difference relative to the variation of samples data.
u	Horizontal optical flow
V	Output matrix
Vel	Velocity
v	Vertical optical flow
x_c,y_c	Vehicle centre coordinates
Xj,Yj	Coordinates of pixel lying within the boundary of the target vehicle
x_k,y_k	Vehicle tracking coordinates

Greek Symbols

ϕ	Angle between background vector and colour vector
ϕ_B	Angle between background vector and (1,1,1) vector
ϕ_o	Maximum angle separation

Abbreviations

BBox	Bounding box
BW	Binary image
FPS	Frame Per Second
RGB	Red Green Blue

with less effort and time instead of manual counting. Also, in developing countries, traffic is composed of different types of vehicles such as cars, buses, trucks and motorized vehicles, etc. Therefore, manual counting of vehicles may not be useful to collect data under these conditions due to confusion in vehicles sizes.

Vehicle detection and classification through image processing program, regarding to capability and resourcefulness in computing data collection has been an area of interest in both transportation as well as computer vision communities for the past few years.

Lately, queue detection, incident detection, vehicle classification, and vehicle counting were included to traffic research field by applying image processing [3, 4]. Many researchers like [5 - 7] have been trying to develop methods that can be applied in video-based traffic surveillance. Vehicle tracking, counting the number of vehicles, Calculating vehicle velocity, finding vehicle trajectory, classifying the vehicles, estimating the traffic density, finding the traffic flow, and license plate recognition, etc. were included in applications of video base surveillance [8]. Morris and Trivedi [9] stated that tracking is the process of measurements obtained from a target in order to maintain an estimate of its current state. A track is a state trajectory estimated from a set of measurements that have been associated with the same target. In the tracking targets, there can be two major problems: an uncertainty associated with the measurements in addition to their accuracy, which is usually modelled by additive noise, and the uncertainty of the measurement origin, a measurement that is to be used in the tracking algorithm may not have originated from the target of interest.

2. Theoretical Work

In this paper, the frame difference method is used to detect and count the number of vehicles. The whole method can be divided into three processes. These processes are discussed in the following:

2.1. Pixel Analysis

Recognizing the absolute difference between the two respective images, pixel-by-pixel with the present threshold is one of the basic methods of change detection. A pixel, I(i,j), is considered as foreground pixel under the condition that the foreground difference, d(i,j) is defined in the following equation:

$$d(i,j) = [I_t(i,j) - I_{t-1}(i,j)] > T \tag{1}$$

where T denotes the pre-set threshold and $I_t(i,j)$ and $I_{t-1}(i,j)$ are pixels at current and previous frames.

Wei et al. [10] and Avery et al. [11] noted that the difference of two corresponding pixels in the subsequent frames is the main problem in this simple approach. The absolute difference could be less than the threshold and if the pixel under consideration lacks texture and is part of the moving object, it will be considered as a background pixel. Also, random camera noise can further confuse the decision to be made.

To overcome this problem a background image is constructed using several frames from the video. Now, instead of making a comparison with the previous frame, the current frame is compared with the background image $[B(i,j)]$.

$$d(i,j) = [l_t(i,j) - B(i,j)] > T \quad (2)$$

For background image construction, the histogram is constructed at each pixel location. A pixel can have a value between (0) and (255) colour value. Frames with the same pixel value are grouped together to make the histogram. Since the foreground region will be covered by the moving vehicles with different colours, it can be assumed that the pixel value that occurs most frequently in the histogram is the background pixel. This way a background image can be generated.

2.2. Shadow Removal

Removing cast shadow is a proposed edge-based algorithm by Xiao et al. [12]. They base their algorithm on three observations. They found that after generating edges on foreground images:

- the cast shadows present sharp edges because the illumination source is far from the objects;
- the vehicle has significant edges; however the corresponding shadow is generally edgeless;
- the edge of cast shadow fastens on the boundary region of the moving foreground mask.

According to Porikli and Thornton [13] the assumption method stated that: "shadow decreases the luminance and changes the saturation, yet it does not affect the hue". The colour vector *I(p)* pointed to the background vector *B(p)* to get the changes from light to shadow for (*h*) as shown in Fig. 1 and Eq. (3).

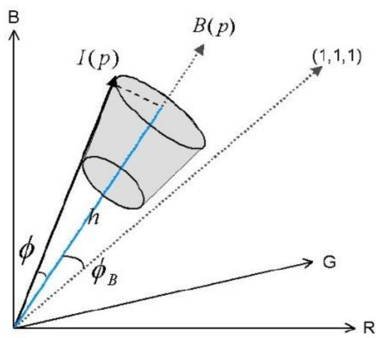

Fig. 1. Weak shadow is defined as a conic volume around the corresponding.

$$h = [I(p)] \cos \emptyset \quad (3)$$

where $\emptyset$ is the angle between *B(p)* and *I(p).* Another angle $\emptyset_B$ is computed between background vector and (1,1,1) vector. They, further, define the light ratio as r = *[I(p)]h.* Pixels satisfying the following criteria are considered as the shadow pixel.

$$\emptyset < min\,(\emptyset_B, \emptyset_o), \qquad r_1 < r < r_2 \tag{4}$$

where $\emptyset_o$ is the maximum angle separation, $r_1 < r_2$ determines the maximum darkness and brightness allowed respectively. In this research, this approach is used to eliminate all shadow pixels.

2.3. Vehicle Tracking

The distance among all vehicle images was used by Atkociunas et al. [14] using the coordinates of their centre to find the tracking vehicles in the subsequent frames. Firstly, the marked geometric centres of each vehicle are calculated as follows:

$$x_c = \frac{\sum x_j}{n}, y_c = \frac{\sum Y_j}{n} \tag{5}$$

where x_c and y_c are vehicle center coordinates and Xj and Yj are the coordinates of pixel lying within the boundary of the target vehicle. Their assumption on tracking vehicles was that the "displacement of image centre of the observed vehicle in two neighbouring frames is less than the distance between it and another vehicle's centres in the same or neighbouring frames". The calculation distances between all vehicles in frame *n* and all vehicles in frame *(n+1)* using coordinates (*x, y*) of their vehicles' centres are to find the tracking vehicle:

$$d_k = \sqrt{(x_k - x_c)^2 + (y_k - y_c)^2} \tag{6}$$

Finding the minimum of d_k gives away the tracking vehicle. Applying this method on all the vehicles present in the current image will keep them tracked.

3. Methods of Traffic Counting

the optical flow estimation model was used for estimation the motion vectors in each frame of the video sequence [15]. Binary feature images were produced by Thresholding and performing morphological closing on the motion vectors model. The model that locates the cars in each binary feature image using the Blob Analysis block is presented in Fig. 2.

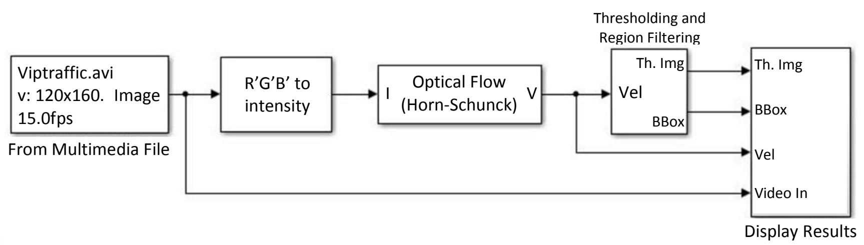

Fig. 2. Tracking cars using optical flow model.

The "From Multimedia File block" reads audio samples, video frames, or both from a multimedia file. The block imports data from the file into a Simulink model while The “Colour Space Conversion block” converts colour information between colour spaces. The conversion from the R',B',G' colour space to intensity is the international standard for digital coding of TV pictures at 525 and 625 line rates and it deals only with the digital representation of R',B',G' signals [16]. It is defined by the following equation:

$$intensity = 0.299R' + 0.587B' + 0.114B' \quad (7)$$

The “Optical Flow block” estimates the direction and speed of the object motion from one image to another or from one video frame to another using either the Horn-Schunck or the Lucas-Kanade method [17, 18]. The following optical flow constraint equation is to compute the optical flow between two images:

$$I_x u + I_y v + I_t = 0 \quad (8)$$

where I_x, I_y, and I_t are the spatiotemporal image brightness derivatives. *u* is the horizontal optical flow and *v* is the vertical optical flow.

The “Thresholding and Region Filtering Block” represents a series of subsystems:

1) Provide an input port for a subsystem or model.
2) Represent mathematical functions including logarithmic, exponential, power, and modulus functions.
3) Compute the mean value along the specified dimension of the input or across time (running mean).
4) Use the Neighborhood size parameter (the dimensions size in a form of a matrix) to specify the size of the neighborhood over which the block computes the median e.g. [3 3].
5) Perform morphological closing on an intensity or binary image.

Figure 3 shows the subsystem block of the Thresholding and Region Filtering model.

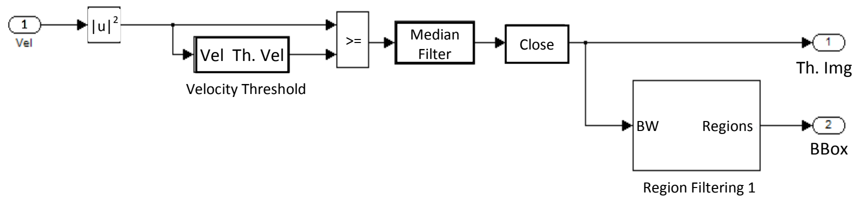

Fig. 3. Thresholding and region filtering model.

Finally, the Draw Shape block was used to draw a green rectangle around the cars that pass beneath the white line. Figure 4 shows four windows which represent the input screen along with the vector analysis for the vehicles beside the Thresholding analysis to conclude with the result window.

Fig. 4. Display results windows.

It is necessary to select high level of traffic flow sites that give realistic results in order to be analysed statistically to collect corrected and sufficient data that satisfy the requirements of statistical calculations and representations. Selected sites should satisfy the following:

- The existence of an accessible vantage point allows for data collection to be made without giving any effect on the observed traffic behaviour.
- Vehicle flow varies over the times of the day.
- The range of the percentage of vehicle movement types and traffic compositions are to be considered.

4. Data Collection

Along the daytime of working days (7:30 – 10:30 am), visits were made to three highway sections for the two opposite directions along LEBUHRAYA KAJANG SILK which divides the multilane highway crossing KAJANG city, MALAYSIA. Based on surveys, the divided multilane highway sections were selected since these were found to satisfy the objectives and specifications of data collection. Fifteen minutes have been adopted as a period time along the specified working time which leads us to 12 time-segments per section per direction. In total, 72 time-segments have been recorded for all sections per directions.

The recorded data were abstracted from video films with the aid of a computer program called MATLAB® program. Developed by MathWorks, MATLAB® allows matrix manipulations, the plotting of functions and data, implementation of algorithms, as well as the creation of user interfaces.

Unnecessary objects have been removed from the video sections using the Shadow Removal technique based on the edge generation algorithm for foreground object’s images and omitting the background’s disruption by changing the saturation level.

5. Data Analysis And Results

The procedures presented in this section are performing a paired samples t-test analysis between observed and programed vehicles counting based on the survey to know the credibility of the counting vehicles programs instead of using the observed one, Table 1 shows the results for paired sample t-test analysis for both, then classifying the vehicles by using the video image processing technique.

5.1. Paired Samples t-test Analysis

The paired samples *t*-test compares two means that are from the same individual, object, or related units. The two means typically represent two different cases. To do so, firstly, we must state the two sets of variables for the analysis. Secondly, we set up two hypotheses. The first is the null hypothesis, which assumes that the mean of two paired samples are equal. The second hypothesis will be an alternative hypothesis, which assumes that the means of two paired samples are not equal. After making the hypothesis, we choose the level of significance. In most of the cases, significance level is 5%. To calculate the parameter we will use the following formula [19]:

$$t = \frac{\sum d}{\sqrt{\frac{N(\sum d^2)-(\sum d)^2}{N-1}}} \tag{9}$$

where *d* is the mean difference between two samples, *N* is the sample size and t is a paired sample t-test with *N-1* degrees of freedom.

Table 1. Paired Samples t-test analysis for section A (NB)

Time	Observation	Program	Paired Samples t-Test	
7:30	1753	1784	*N*	12
7:45	2641	2677	*Observation Mean*	1542.75
8:00	2258	2235	*Program Mean*	1540.83
8:15	2495	2441	*Correlation*	0.998529
8:30	1632	1682	*Paired Sample t-test Mean*	1.917
8:45	1361	1383	*Std. Error Mean*	9.705
9:00	1143	1112	*P-value*	0.847045
9:15	899	915	*t*	0.197491
9:30	1149	1124		
9:45	964	932		
10:00	1047	1020		
10:15	1171	1185		

From the results of both; observation and selected program, a comparison was made between the two results by using the paired samples t-test. The major measure of effectiveness, counting vehicles was considered. The results in Table 1 showed that the mean of difference in counting vehicles is 1.917 vehicles which considered too small, with a p-value of 0.85, with higher correlation of 0.998 and a small standard error in mean by 9.705 with *t* value of 0.197491 (the greater the evidence of *t* against the null hypothesis that there is no significant difference). This hypothesis concludes that there is no significant difference is accepted at 95% confidence. Figure 5 shows the line fit for both 12 readings of program and observation in Table 1.

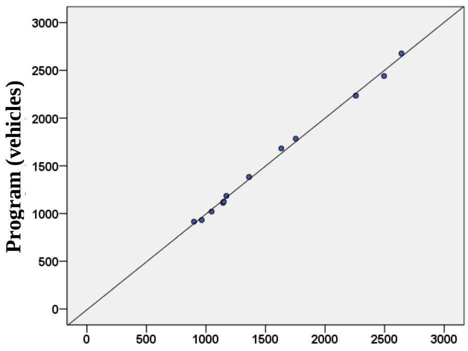

Fig. 5. Line fit plot for program and observation sets.

5.2. Classification of vehicles

Improfile code was used to do the vehicles classification in the Matlab-Video Image Processing that determines the value of intensity along a path of line shown in an image. Equal spaced points along a specified path are selected and then interpolation is used to determine the colour intensity value for each point. *Improfile* works with both grayscale and colour intensity images. Figures 6 and 7 show the variation of colour images and a pixel-value cross-section along line segments for long and short vehicles snapshot images of a video feed.

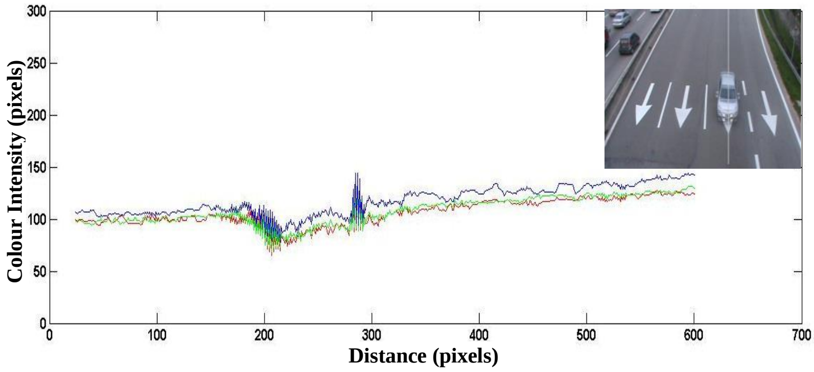

Fig. 6. Snapshot image for short vehicle and its pixel-value cross-sections along line segment.

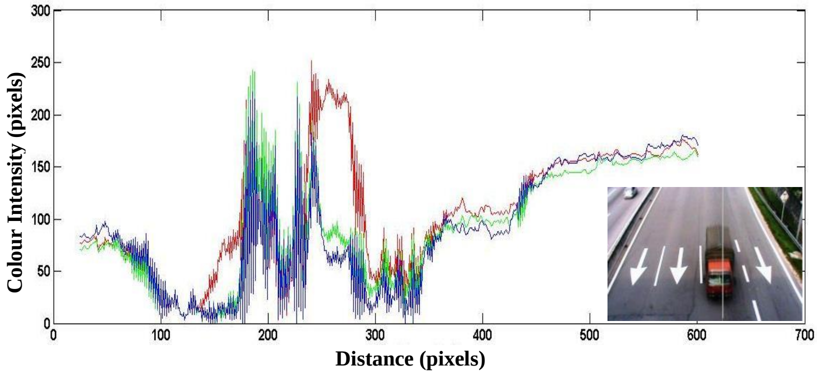

Fig. 7. Snapshot image for long vehicle and its pixel-value cross-sections along line segment.

Thus, the finding from this study suggests that the algorithm model can estimate the number of passing vehicles from recorded video segments using an optical flow model which represents the actual values of the real number of vehicles according to the paired sample *t*-test analysis. The result of *correlation* shows that we cannot reject the result of the model despite the minor error of reading between the observation and program variables which can be accepted, as there are no significance mean differences. The classification modeling has been made to classify the vehicles as a tool that can prevents miscounting the small vehicles like motorcycles by reading the intensity of the longitudinal line of the vehicle across video image pixels.

Sufficient data was gathered (72 time-segments) across the specified sections to show the credibility of the algorithm model against the manual counting which take a long time and much effort is needed to collect the data during bad weather. Therefore, if the algorithm model is to be applied, the resulting values need to be adjusted according to the methodology section.

6. Conclusions

In this paper, an algorithm program was developed to detect vehicles in traffic videos and to obtain the vehicle count for the small time period by a simple camcorder. The vehicle-counting was developed and implemented by MATLAB. The background was extracted, with an assumption that the camera position is still and stable. Since the extraction process was for small duration, it was assumed that there is no sudden change in the light (day time). With that assumption, almost 95% number of vehicles were been counted correctly and been validated by conducting paired sample *t*-test for both Observation and Program sets. During the shadow removal process, the pixel analysis, some of the vehicles (especially motorcycle) were miscounted due to the appearance of heavy vehicles in front of the motorcycles. This led us to carry out a conventional paired sample *t*-test analysis with underestimates of the coefficient for NB sections and overestimates of the coefficient for the South Bound (SB) sections. The purpose of this study was to count and classify the number of vehicles on a road, so an algorithm for vehicle counting and classification has been developed instead of counting and sorting the types of vehicles via observation. This algorithm program was tested successfully and the results were found to be close to the actual numbers of vehicles on the road.

For future study, the algorithm program should be investigated more night time to study the effects for absence of sunlight in order to define the shadows around the corresponding colours extensively. Also, more calibrations are needed regarding the sensitivity of video feed shaking distortion due to the vibration of passing vehicles.

References

1. Transportation Research Board. (2010). *Highway capacity manual* (5^{th} ed.). Washington, D.C.: National Research Council.
2. Mathew, T.V.; and Rao, K.K. (2006). Introduction to transportation engineering. *Journal of Civil Engineering–Transportation Engineering.* IIT Bombay, Nptel Online, retrieved October 19, 2006, from http://www.cdeep.iitb.ac.in/nptel/Civil% 20Engineering.
3. Sun, Z.; Bebis, G.; and Miller, R. (2004). On-road vehicle detection using optical sensors: A review. *Journal of Intelligent Transportation Systems, Proceedings. The 7th International IEEE Conference on*, 585-590.
4. Dailey, D.J.; Cathey, F.W.; and Pumrin, S. (2000). An algorithm to estimate mean traffic speed using uncalibrated cameras. *Journal of Intelligent Transportation Systems,* 1(2), 98-107.
5. Oh, J.; and Leonard, J.D. (2003). Vehicle detection using video image processing system: Evaluation of PEEK VideoTrak. *Journal of Transportation Engineering* 129(4), 462-465.

6. McCall, J.C.; and Trivedi, M.M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE transactions on intelligent transportation systems*, 7(1), 20-37.
7. Qian, Z.; Shi, H.; Yang, J.; and Duan, L. (2013). Video-based multiclass vehicle detection and tracking. *IJCSI International Journal of Computer Science Issues*, 10(1), 570-578.
8. Trivedi, M.M.; Gandhi, T.; and McCall, J. (2007). Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *Journal of Intelligent Transportation Systems,* 8(1), 108-120.
9. Morris, B.T.; and Trivedi, M.M. (2008). Learning, modeling, and classification of vehicle track patterns from live video. *Journal of Intelligent Transportation Systems,* 9(3), 425-437.
10. Wei, J.; Ye, G.; Pickering, M.; Frater, M.; and Arnold, J. (2003). Robust and self-adaptive background extraction in video object change detection. *Journal of Optical Science and Technology, SPIE's 48th Annual Meeting*. San Diego, California, 142-147.
11. Avery, R.P.; Wang, Y.; and Rutherford, G.S. (2004). Length-based vehicle classification using images from uncalibrated video cameras. *Journal of Intelligent Transportation Systems, Proceedings. The 7th International IEEE Conference on*. 737-742.
12. Xiao, M.; Han, C.Z.; and Zhang, L. (2007). Moving shadow detection and removal for traffic sequences. *International Journal of Automation and Computing,* 4(1), 38-46.
13. Porikli, F.; and Thornton, J. (2005). Shadow flow: A recursive method to learn moving cast shadows. *Journal of Computer Vision, ICCV 2005. Tenth IEEE International Conference on*. Beijing, 891-898.
14. Atkociunas, E., Blake, R., Juozapavicius, A. & Kazimianec, M. (2005). Image processing in road traffic analysis. *Journal of Nonlinear Analysis: Modelling and Control,* 10(4), 315-332.
15. MathWorks. (2006). Tracking cars using optical flow. Retrieved 2006. from http://www.mathworks.com/examples/simulink-computer-vision/712-tracking-cars-using-optical-flow.
16. Ford, A.; and Roberts, A. (1998). *Colour space conversions*. Westminster University, London, 1-31.
17. Tomasi, C.; and Kanade, T. (1991). *Detection and tracking of point features*. Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, Pittsburgh.
18. Lucas, B.D.; and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence.* Vancouver, Canada, 674-679.
19. Mee, R.W.; and Chua, T.C. (1991). Regression toward the mean and the paired sample t test. *The American Statistician,* 45(1), 39-42.