Journal of Geotechnical and Transportation Engineering
Volume 2 | Issue 2
An Intelligent System for Automatic Detection of Trafiic Rules Violation From Traffic Surveillance Camera Videos
M. Alavianmehr et al.
Received 10/2/2016 Accepted 11/12/2016
Published 12/1/2016
Corresponding Author:
M. Alavianmehr; alavianmehr@gmail.com
www.jgtte.com

An Intelligent System for Automatic Detection of Traffic Rules Violation From Traffic Surveillance Camera Videos

M. A. Alavianmehr
Mayor's Consultant in IT and Ph. D. Student at Electrical Engineering-Electronics, Shiraz Municipality, Iran
E-mail: alavianmehr@gmail.com

A. R. Pakfetrat
Mayor of Shiraz, Ph. D. Student at Faculty of Geographical Sciences and Planning, University of Isfahan, Isfahan, Iran
E-mail: Pakfetrat.mayor@gmail.com

Zohreh Karimian
Lecture in Ghazvin Azad University, Ghazvin, Iran
Email: zohreh_60@yahoo.com

Abstract

Checking traffic rule observation by vehicles play an important role in every day transportation handling either for intra- or inter-city travels. Also, image processing plays a great role in modern intelligent transportation system (ITS). One of the most advance ways for traffic management and rules observation studies is employing live surveillance camera videos. In this paper, a new approach toward automatic detection of traffic rules violation based on image processing techniques is proposed. The proposed method applies innovative image processing techniques for live traffic surveillance target. Based on these techniques, the moving objects including cars and pedestrians are detected, tracked and observed. At first, some preprocessing steps employed for discrimination of foreground from background of surveillance video frames. For tracking purpose, a modified Munkres' version of Hungarian algorithm is applied to Kalman filtering to provide tracking predictions for detected moving objects. The tracks of detected moving objects are analyzed and if any traffic rule violation takes place, they will be detected and reported automatically. The implementation results related to the proposed method demonstrates its high performance and applicability for real traffic rule violation detection.

Keywords: Gaussian Mixture Model; Kalman Filtering; Hungarian Algorithm;Intelligent Transportation System; Munkres' Assignment Algorithm.

1. Introduction

The escalating increase of contemporary urban and national road networks over the last three decades emerged the need of efficient monitoring and management of road traffic. Environmental pressures as well as socioeconomic problems are associated with this increase due to prolonged congestions and slowing down of the average highway speed. To deal with this problem, one option is to increase network capacity and the other one is to increase efficiency by investing in Intelligent Transportation Systems (ITS) technology. Intelligent Transportation System (ITS) is the applications of new information and communication technologies (ICT) into vehicles and roadways for monitoring traffic conditions, decreasing congestion, increasing mobility, improving the use of transport infrastructure and enhancing security. They also draw traffic predictions and suggest the best departure times in different areas, and alternative routes for important roads. They make road users aware in real time on their travel time and best routes to be taken given a destination, and so on. The technologies used in intelligent transportation systems include basic management systems such as variable message signs (VMS), automatic radar or video surveillance to applications more advanced that integrate data in real time with feedback from many sources, such as weather information, embedded navigation systems informing travel time in real-time.

Video surveillance on highway has been probed recently, but still it is a hot topic and a great challenge in ITSs. In such applications requiring objects extraction, cast shadows induce shape distortions and object fusions interfering performance of high level algorithms. Shadow elimination let enhance the performances of video object extraction, classification and tracking. In other hand, it is real vital to determine the type of a detected vehicles in order to track reliably and estimate traffic parameters accurately. Background subtraction is a technique to remove nonmoving components from a video frame. The main assumption for its application is that the camera remains still. The basic principle is to make reference frame of the stationary components in the frame and then each pixel of the prototype is compared with the actual frame color map. If the color difference goes beyond a predefined threshold value, it is assumed that this pixel is a part of the foreground frame. There are numerous challenges that a background model must solve correctly; such an illumination changes, changes in the background geometry (parked, slow and stationary vehicles…) and repetitive movements of contextual objects in the scene (leaves of trees…). Considering these challenges, the background model must be adaptive to the change of illumination and geometry of background and robust to the repetitive movements by learning an updated background

image. Detection of moving objects and motion-based tracking are important components of many computer vision applications, including activity recognition, traffic monitoring, and automotive safety. The problem of motion-based object tracking can be divided into two parts:
1- Detecting moving objects in each frame.
2- Associating the detections for the same object over time.

2. Literature Review and Related Works

Analysis and understanding of video sequences is an active research field. Many applications in this research area (video surveillance [1–3], optical motion capture [4], multimedia application [5]) need in the first step to detect the moving objects in the scene. So, the basic operation needed is the separation of the moving objects called “foreground” from the static information called the “background”. The process mainly used is the background subtraction and recent surveys can be found in [6–8]. The simplest way to model the background is to acquire a background image which does not include any moving object. In some environments, the background is not available and can always be changed under critical situations like illumination changes, objects being introduced or removed from the scene. So, the background representation model must be more robust and adaptive. The basic technique is to model each pixel in a video frame with a Gaussian distribution [9]. The model does not work well in the case of dynamic natural environments, where the background is multimodal. A very popular technique is the Mixture of Gaussians Models (GMM) [10]. It is used to model complex background, and applied to the traffic monitoring problem. GMM is one of baseline algorithms for background modeling due to its effectiveness and efficiency for motion detection from video sequences. In [11], we presented an improved GMM model. In our algorithm in order to estimate the traffic status and to compute the velocity, receiving the images through video surveillance camera in first phase, we get use of GMM for each frame to achieve a precise background image. This process will be repeated as long as we seize an accurate background images. This phase is called training phase. In the second phase, the received images will be analyzed a long with the trained images to extract the vehicles (moving objects) based on this analysis. Also, a green block will surround each vehicles to enable us to count them. Either inaccurate training of the background images or the shadow of moving vehicles might cause problems in detecting vehicles in motion in the second phase. To solve these problems we used of merging algorithm. In [12], the authors argue that the GMM is not effective for outdoor scenes. They show that in outdoor scenes, the distribution of pixel intensity over a long period covers a wide range of intensity, and they choose to replace the GMM probability density function (PDF) with a kernel-based density estimation method. However, the cost to compute the kernel density, estimate at each pixel, is very high in term of memory requirement and time. Another efficient technique is the Codebook model proposed in [13] where the authors model each pixel with a codebook consisting of one or more codewords. The authors argue that GMM and Kernel method cannot handle rare background pixel values. To solve this problem, they propose to use a training phase to model these rare pixel values and every pixel value occurring during the training phase must pass a temporal test. If a pixel value passes the test successfully, this pixel value is considered as background. Another recent algorithm like the previous works is proposed in [14], it focuses on the calculation and reconstruction of the background template, based on scattered pixel level in formation obtained from a series of consecutive image frames. The idea is based on the notion that a specific location is occupied by moving objects for a time period shorter than that for which it remain unoccupied. The authors calculate histogram frequencies to choose the background pixel corresponds to high values at the side histograms. However, this approach neglects the temporal in formation that is interesting to model its basic idea. Others proposed works showing to enhance background model. In terms of background update; the background image is subsequently updated by the average [15] or the median [16] of the previous n frames. In [17], A. Tashk et al. presented an automatic traffic control systems which based on advance software architecture. In the proposed architecture, automatic car plate recognition and driver verification based on fingerprint biometric are mixed with each other. The license plate recognition part is adapted for Persian or Farsi characters. The Persian or Farsi characters recognition is done by a very simple Normalized cross correlation which is very analogous to Euclidian distance criterion. To improve the functionality of this platform, some special and innovative digital image processing are employed so that the program is able to extract and recognize the car plate and its related characters even if in the low light or shiny conditions. The fingerprint recognition system is also added to the proposed traffic control system to ensure the authority of the car driver to enter to the places with high privacy and security limitations. In [18], authors presented the dynamic Hungarian algorithm, applicable to optimally solving the assignment problem in situations with changing edge costs or weights. This problem is relevant, for example, in a transportation domain where the unexpected closing of a road translates to changed transportation costs. When such cost changes occur after an initial assignment has been made, the new problem, like the original problem, may be solved from scratch using the well-known Hungarian algorithm. However, the dynamic version of the algorithm which we present solves the new problem more efficiently by repairing the initial solution obtained before the cost changes. In table I, a pseudo code for Hungarian Algorithm is presented.

TABLE I
All steps of Hungarian algorithm for predicting tracks of multiple moving objects

Pseudo Code Hungarian Algorithm

```
inthungarian()
{
int ret = 0; //weight of the optimal matching
max_match = 0; //number of vertices in current matching
memset(xy, -1, sizeof(xy));
memset(yx, -1, sizeof(yx));
init_labels(); //step 0
 augment(); //steps 1-3
for (int x = 0; x < n; x++) //forming answer there
 ret += cost[x][xy[x]];
return ret;
}
```

2.1. Munkres’ Assignment Algorithm

Let C be an n×n matrix representing the costs of each of n workers to perform any of n jobs. The assignment problem is to assign jobs to workers so as to minimize the total cost. Since each worker can perform only one job and each job can be assigned to only one worker the assignments constitute an independent set of the matrix C. An arbitrary assignment is shown above in which worker a is assigned job q, worker b is assigned job s and so on. The total cost of this assignment is 23. A brute-force algorithm for solving the assignment problem involves generating all independent sets of the matrix C, computing the total costs of each assignment and a search of all assignment to find a minimal-sum independent set. The complexity of this method is driven by the number of independent assignments possible in an n×n matrix. There are n choices for the first assignment, n-1 choices for the second assignment and so on, giving n! Possible assignment sets. Therefore, this approach has, at least, an exponential runtime complexity. The following 6-step algorithm is a modified form of the original Munkres' Assignment Algorithm. This algorithm describes to the manual manipulation of a two-dimensional matrix by starring and priming zeros and by covering and uncovering rows and columns. The related procedures of employed Mukres' assignment algorithm are summarized in Table II.

TABLE II
All steps of Munkres' assignment algorithm for predicting tracks of multiple moving objects

Algorithm Munkres' Assignment Algorithm
Step 0: Create an n×m matrix called the cost matrix in which each element represents the cost of assigning one of n workers to one of m jobs. Rotate the matrix so that there are at least as many columns as rows and let k=min (n, m).
Step 1: For each row of the matrix, find the smallest element and subtract it from every element in its row. Go to Step 2.
Step 2: Find a zero (Z) in the resulting matrix. If there is no starred zero in its row or column, star Z. Repeat for each element in the matrix. Go to Step 3.
Step 3: Cover each column containing a starred zero. If K columns are covered, the starred zeros describe a complete set of unique assignments. In this case, Go to DONE, otherwise, Go to Step 4.
Step 4: Find a no covered zero and prime it. If there is no starred zero in the row containing this primed zero, Go to Step 5. Otherwise, cover this row and uncover the column containing the starred zero. Continue in this manner until there are no uncovered zeros left. Save the smallest uncovered value and Go to Step 6.
Step 5: Construct a series of alternating primed and starred zeros as follows. Let Z0 represent the uncovered primed zero found in Step 4. Let Z1 denote the starred zero in the column of Z0 (if any). Let Z2 denote the primed zero in the row of Z1 (there will always be one). Continue until the series terminates at a primed zero that has no starred zero in its column. Unstart each starred zero of the series, star each primed zero of the series, erase all primes and uncover every line in the matrix. Return to Step 3.
Step 6: Add the value found in Step 4 to every element of each covered row, and subtract it from every element of each uncovered column. Return to Step 4 without altering any stars, primes, or covered lines.
Assignment pairs are indicated by the positions of the starred zeros in the cost matrix. If C (i, j) is a starred zero, then the element associated with row i is assigned to the element associated with column j. Some of these descriptions require careful interpretation. In Step 4, for example, the possible situations are, that there is a no covered zero which get primed and if there is no starred zero in its row the program goes onto Step 5. The other possible way out of Step 4 is that there are no covered zeros at all, in which case the program goes to Step 6.

Receiving the images through video surveillance camera in first phase, we get use of GMM for each frame to achieve a precise background image. This process will be repeated as long as we seize an accurate background images. This phase is called training phase. In the second phase, the received images will be analyzed a long with the trained images to extract the vehicles (moving objects) based on this analysis. As the mention above, we may extract vehicles more accurately as long as, we have a more precise trained background images.

In third phase, Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects. The detection of moving objects uses a background subtraction algorithm based on Gaussian mixture models. The association of detections to the same object is based solely on motion. The motion of each track is estimated by a Kalman filter. The filter is used to predict the track's location in each frame, and determine the likelihood of each detection being assigned to each track. Track maintenance becomes an important aspect of the proposed motion detection strategy. In any given frame, some detections may be assigned to tracks, while other detections and tracks may remain unassigned. The assigned tracks are updated using the corresponding detections. The unassigned tracks are marked invisible. An unassigned detection begins a new track. Each track keeps count of the number of consecutive frames, where it remained unassigned. If the count exceeds a specified threshold, the method assumes that the object left the field of view and it deletes the track. The whole procedures and steps of proposed algorithm are summarized in Table III.

TABLE III
Full Automatic Traffic Rule Observation and Violation Detection Algorithm

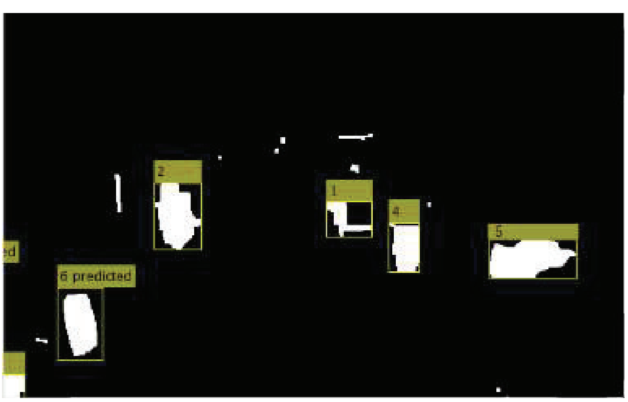
Algorithm Traffic Rule Violation Detection
0) Start
a) Create system objects used for reading video, detecting moving objects and displaying the results
b) Detect moving objects, and track them across video frames.
c) Create System objects used for reading the video frames, detecting foreground objects, and displaying results.
d) Initialize Video I/O: Create objects for reading a video from a file, drawing the tracked objects in each frame, and playing the video.
e) Create system objects for foreground detection and blob analysis: The foreground detector is used to segment moving objects from the background. It outputs a binary mask, where the pixel value of 1 corresponds to the foreground and the value of 0 corresponds to the background.
f) Connected groups of foreground pixels are likely to correspond to moving objects. The blob analysis system object is used to find such groups (called 'blobs' or 'connected components'), and compute their characteristics, such as area, centroid, and the bounding box.
g) Initializing Tracks: The initialize Tracks function creates an array of tracks, where each track is a structure representing a moving object in the video. The purpose of the structure is to maintain the state of a tracked object. The state consists of information used for detection to track assignment, track termination, and display. The structure contains the following fields: |id|: The integer ID of the track |bbox|: The current bounding box of the object; used for display |Kalman Filter|: a Kalman filter object used for motion-based tracking |age|: The number of frames since the track was first detected |total Visible Count|: The total number of frames in which the track was detected (visible) |consecutive Invisible Count|: the number of consecutive frames for which the track was not detected (invisible). Noisy detections tend to result in short-lived tracks. For this reason, the example only displays an object after it was tracked for some number of frames. This happens when |total Visible Count| exceeds a specified threshold. When no detections are associated with a track for several consecutive frames, the example assumes that the object has left the field of view and deletes the track. This happens when |consecutive Invisible Count| exceeds a specified threshold. A track may also get deleted as noise. If it was tracked for a short time, and marked invisible for most of the frames.
h) Detect Objects: The |detect Objects| function returns the centroids and the bounding boxes of the detected objects. It also returns the binary mask, which has the same size as the input frame. Pixels with a value of 1 correspond to the foreground, and pixels with a value of 0 correspond to the background. The function performs motion segmentation using the foreground detector. It then performs morphological operations on the resulting binary mask to remove noisy pixels and to fill the holes in the remaining blobs.
i) Predict New Locations of Existing Tracks: Use the Kalman filter to predict the centroid of each track in the current frame, and update its bounding box accordingly: - Predict the current location of the track. - Shift the bounding box so that its center is at the predicted location.
j) Assign Detections to Tracks: Assigning object detections in the current frame to existing tracks is done by minimizing cost. The cost is defined as the negative log-likelihood of a detection corresponding to a track. The algorithm involves two steps: Step 1: Compute the cost of assigning every detection to each track using the |distance| method of the |vision Kalman Filter| System object. The cost takes into account the Euclidean distance between the predicted centroid of the track and the centroid of the detection. It also includes the confidence of the prediction, which is maintained by the Kalman filter. The results are stored in an M×N matrix, where M is the number of tracks, and N is the number of detections. Step 2: Solve the assignment problem represented by the cost matrix using the |assign Detections to Tracks| function. The function takes the cost matrix and the cost of not assigning any detections to a track. The value for the cost of not assigning a detection to a track depends on the range of values returned by the |distance| method of the |vision. Kalman Filter|.

This value must be tuned experimentally. Table I shows the proposed method.
Setting it too low increases the likelihood of creating a new track, and may result in track fragmentation. Setting it too high may result in a single track corresponding to a series of separate moving objects. The |assign Detections to Tracks| function uses the Munkres' version of the Hungarian algorithm to compute an assignment which minimizes the total cost. It returns an M×2 matrix containing the corresponding indices of assigned tracks and detections in its two columns. It also returns the indices of tracks and detections that remained unassigned.
k) Update Assigned Tracks: The |update Assigned Tracks| function updates each assigned track with the corresponding detection. It calls the |correct| method of |vision. Kalman Filter| to correct the location estimate. Next, it stores the new bounding box, and increases the age of the track and the total visible count by 1. Finally, the function sets the invisible count to 0.
l) Update Unassigned Tracks: Mark each unassigned track as invisible, and increase its age by 1.
m) Update Unassigned Tracks: Mark each unassigned track as invisible, and increase its age by 1.
n) Create New Tracks: Create new tracks from unassigned detections. Assume that any unassigned detection is a start of a new track. In practice, you can use other cues to eliminate noisy detections, such as size, location, or appearance.
o) Display Tracking Results: The |display Tracking Results| function draws a bounding box and label ID for each track on the video frame and the foreground mask. It then displays the frame and the mask in their respective video players. Note that: Noisy detections tend to result in short-lived tracks. Only display tracks that have been visible for more than a minimum number of frames. Display the objects. If an object has not been detected in this frame, display its predicted bounding box.

3. Implementation Results

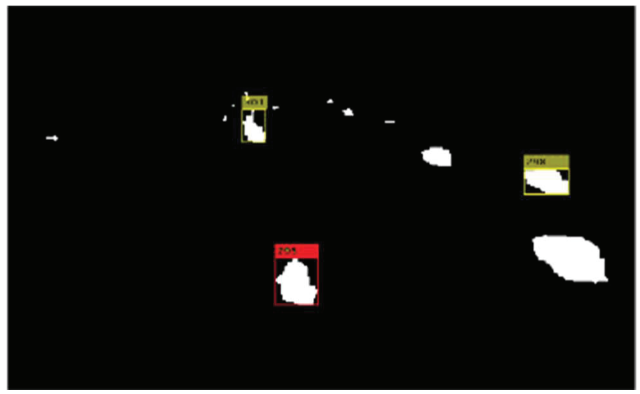
Using Matlab software, we analyzed traffic videos obtained from video surveillance camera. Fig. 1 shows two sample traffic rules violation detections and consequently tracking.

(a)



(b)

(c)



(d)

Figure 1. Detection and Tracking full path of two traffic law violation committed by (a) & (b) a driving car moving in reverse gear which is prohibited in high ways in both RGB and binarizedformats, and (c) & (d) a motorcycle driver driving in the wrong side and direction of a highway in both RGB and binarizedformats

As it is shown in fig.1, it is possible to detect a motorcycle driver driving in the wrong side and direction of a highway by means of our proposed method. According to proposed method, fig. 2 depicts the two non-consequent frames related to a rule violation committed by a driving car in a high way.

4. Conclusion

The tracking procedure of the proposed method was solely based on motion with the assumption that all objects move in a straight line with constant speed. When the motion of an object significantly deviates from this model, the method may produce tracking errors. Such mistakes are noticeable when moving objects occlude by each other. The likelihood of tracking errors can be reduced by using a more complex motion model, such as constant acceleration, or by using multiple Kalman filters for every object. Also, other cues for associating detections over time, such as size, shape, and color can be incorporated. In the proposed method, any kinds of traffic law violation can be detected. As an example even changes between lanes are possible to be determined by means of this proposed method.

(a)

(b)

Figure 4. Illustration of a of drivering car's path moving in reverse gear in a highway

References

[1] S. Cheung, C. Kamath, Robust background subtraction with foreground validation for urban traffic video, EURASIP J. Appl. Signal Process. (2005).

[2] Y. Tian, A. Senior, M. Lu, Robust and efficient foreground analysis in complex surveillance videos, Mach. Vis. Appl. 23 (5) (2012) 967–983.

[3] A. Senior, Y. Tian, M. Lu, Interactive motion analysis for video surveillance and long term scene monitoring, in: Asian Conference on Computer Vision, ACCV 2010, Workshops, 2010, pp. 164–174.

[4] F. El Baf, T. Bouwmans, Comparison of background subtraction methods for a multimedia learning space, in: International Conference on Signal Processing and Multimedia, SIGMAP, July 2007.

[5] J. Carranza, C. Theobalt, M. Magnor, H. Seidel, Freeviewpoint video of human actors, ACM Trans. Graph. 22 (3) (2003) 569–577.

[6] S. Elhabian, K. El-Sayed, S. Ahmed, Moving object detection in spatial domain using background removal techniques -state-of-art, Recent Patents Comput. Sci. 1 (1) (2008) 32–54.

[7] M. Cristani, M. Farenzena, D. Bloisi, V. Murino, Background subtraction for automated multisensory surveillance: A comprehensive review, EURASIP J. Adv. Signal Process. (2010) 24.

[8] T. Bouwmans, F. El-Baf, B. Vachon, Statistical background modeling for foreground detection: A survey, in: Handbook of Pattern Recognition and Computer Vision, vol. 4(2), World Scientific Publishing, 2010, pp. 181–199.

[9] N. A. Mandellos, I. Keramitsoglou, C. T. Kiranoudis. A background subtraction algorithm for detecting and tracking vehicle, Expert Syst. Appl38 (2011)1619–1631.

[10] C. Wren, A. Azerbaijani, T. Darrell, A. Pent land, P finder: real-time tracking of the human body, IEEE Trans. Pattern Anal. Mach. Intell.19 (1997)780–785.

[11] M. A. Alavianmehr, A. Tashk, A. Sodagaran. Video Foreground Detection Based on Adaptive Mixture Gaussian Model for Video Surveillance Systems. 4th International Conference on Traffic and Transportation Engineering (ICTTE 2015), Madrid, Spain.

[12] A. Elgammal, R. Duraiswami, D. Harwood, L. Davis, Background and foreground model in using nonparametric kernel density for visual surveillance, Proc.IEEE90(7)(2002)1151–1163.

[13] K. Kim, T. H .Chalidabhongse, D. Harwood, L. S. Davis, Real-time foreground-background segmentation using codebook model, Real- Time Imaging11(3)(2005)172–185.

[14] N. A. Mandellos, I. Keramitsoglou, C. T. Kiranoudis, A background subtraction algorithm for detecting and tracking vehicle, Expert Syst. Appl38 (2011)1619–1631.

[15] B. P. L. Lo, S. A. Velastin, Automatic congestion detection system for underground platforms, in: Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong,China,2000,pp.158–161.

[16] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, Detecting moving objects, ghosts and shadows in video streams, IEEETrans. Pattern Anal. Mach. Intell.25 (10) (2003)1337–1342.

[17] A. Tashk, and M. S. Helfroush, "An Automatic Traffic Control System based on Simultaneous Persian License Plate Recognition and Driver Fingerprint Identification," 20th Telecommunications forum TELFOR 2012 Serbia, Belgrade, November 20-22, 2012.

[18] G. Ayorkor Mills-Tettey, Anthony Stentz, M. Bernardine Dias. The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs. Robotics Institute

[19] Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, July 2007.

[20] H. W. Kuhn, "The Hungarian Method for the assignment problem", Naval Research Logistics Quarterly, 2: 83–97, 1955. Kuhn's original publication.

[21] H. W. Kuhn, "Variants of the Hungarian method for assignment problems", Naval Research Logistics Quarterly, 3: 253–258, 1956.

[22] I. H. TOROSLU and G. OLUK, "Incremental assignment problem. Information Sciences," 177, 6 (March 2007), pp. 1523–1529.