2012 15th International IEEE Conference on Intelligent Transportation Systems
Anchorage, Alaska, USA, September 16-19, 2012

Monocular target detection on transport infrastructures with dynamic and variable environments

S. Álvarez, D. F. Llorca, M. A. Sotelo, A. G. Lorente

Abstract— **This paper describes a target detection system on transport infrastructures, based on monocular vision, for applications in the framework of Intelligent Transportation Systems (ITS). Using structured elements of the image, a vanishing point extraction is proposed to obtain an automatic calibration of the camera, without any prior knowledge. This calibration provides an approximate size of the searched targets (vehicles or pedestrians), improving the performance of the detection steps. After that, a background subtraction method, based on GMM and shadow detection algorithms, is used to segment the image. Next a feature extraction, optical flow analysis and clustering methods are used to track the objects. The algorithm is robust to camera jitter, illumination changes and shadows. Therefore it can work indoor and outdoor, in different conditions and scenarios, and independent of the position of the camera. In the paper, we present and discuss the results achieved up to date in real traffic conditions.**

Index terms— **Vanishing points, Camera calibration, Background subtraction, shadow detection, flock of features.**

I. INTRODUCTION

In recent years, the use of cameras for traffic scene analysis has greatly promoted the development of intelligent transportation systems. In result, video sequences are used to detect vehicles and pedestrians for traffic flow estimation, signal timing, safety applications or video surveillance, among others. The challenge and the main task to solve are the object segmentation and tracking.

As the traffic monitoring systems often use fixed cameras, most of the named applications above are based on the *background subtraction* algorithm, as it is referenced in the related work section. The idea is to subtract the current image from a reference image, which is a representation of the scene background, to find the foreground objects. The technique has been used for years in many vision systems as a preprocessing step, and the results obtained are fairly good. However the algorithm is susceptible to several problems such as sudden illumination changes, cast shadows, camera jitter or image noise; which often cause serious errors due to misclassification of moving objects. Moreover, the size of the targets is very dependent of the position of the camera.

In this paper, an approach for detecting moving objects from a static background scene is presented. The idea is to develop a "plug&play" system able to work in a wide range of environments and illumination conditions, without modifying the algorithm. To reach that purpose, all modules have an adaptive component to adjust the system to changes in the scene: adaptive background subtraction, which updates continuously the background model; image stabilization, to minimize the effect of camera vibrations and jitter; shadow and highlight detection, to remove non-permanent illumination changes; and an online automatic camera calibration to overcome problems due to the unknown position of the camera and therefore the sizes of the objects.

The authors are with the Computer Engineering Department, University of Alcalá, Madrid, Spain. e-mail: [*sergio.alvarez*, *llorca*, *sotelo*] *@aut.uah.es*

II. RELATED WORK

The main related work in traffic monitoring, using vision-based systems with fixed cameras, is based on the background subtraction method. The pixel-level Gaussian Mixture Model (GMM) background model has become very popular due to its efficiency working with multi-modal distributions, and the possibility of updating the model as times goes by. Stauffer et al. [1] present a method that models each pixel by a mixture of K Gaussian distributions, and Zivkovic [2] improve the method incorporating a model selection criterion to choose the proper number of components for each pixel on-line. These methods show interesting results in good illumination conditions, and can handle progressive illumination changes. Nevertheless, they are vulnerable to sudden changes, and shadows cast by moving objects can easily be misinterpreted as foreground.

Many efforts have been made to solve the problem of illumination changes. Algorithms can be classified as model-based or property-based. On the one hand, model-based methods use prior knowledge of scene geometry, target objects or light sources to predict and remove shadows. Joshi et al. [3] propose an algorithm which detects shadows by using Support Vector Machine (SVM) and a shadow model, learned and trained from a database. Reilly et al. [4] propose a method based on a number of geometric constraints obtained from meta-data (latitude, longitude, altitude, as well as pitch, yaw and roll). The problem of these methods is they need prior information.

On the other hand, property-based approaches use features like geometry, brightness or color to detect illumination changes. Horprasert et al. [5] propose a color model to classify each pixel as foreground, background, shadowed background, or highlighted background. The algorithm performs well in indoor environments or under certain illumination conditions, but not for the variability of traffic scenes. In [6], Cucchiara et al. use the hypothesis that shadows reduce surface brightness and saturation while maintaining hue properties in the HSV color space. These methods can deal with illumination noises and soft shadows but they fail when color and chromaticity information are totally lost.

978-1-4673-3063-3/12/$31.00 ©2012 IEEE

There are also statistical approaches such as [7] that uses Gaussian Mixture Model to describe moving cast shadows, or [8] which models shadows using multivariate Gaussians. These methods can adapt to changing shadow conditions and provide a low number of false detections. However, the hypothesis is not effective with soft shadows and if the shadowed pixels are seldom or they have never been taken up by the algorithm.

Related to tracking, Bayesian filtering, and in particular Kalman filter, is extensively used to predict the position of the targets. The state vector can be modelled with data directly available from blobs such as kinematic parameters [9]. However, the most interesting works combine background subtraction and feature tracking to take advantage when partial occlusion occurs, since some of the features of the object remain visible. Kanhere et al. [10] use the background subtraction result to estimate the 3D height of corner features by assuming that the bottom of the foreground region is the bottom of the object. The problem is the assumption fails in case of occlusions.

The algorithm proposed addresses these drawbacks by knowing an approximate size of the objects searched. To obtain it, intrinsic and extrinsic parameters of the camera are needed. One possible method is to use a planar checkerboard to extract the corners and calibrate the camera with one of the multiple toolboxes available. However, there are many problems associated like the need to be in the scene with the board for any change of the position of the camera; or the resolution of the board in case the camera is located very high, among others. In this context, vanishing points calibration, as proposed in [11], seems to be an interesting solution to the problem. There are many works dealing with vanishing points in architectural scenarios [12][13], where the large number of orthogonal lines provide good results. The difficulties in traffic scenes come from the lack of structured orthogonal elements. Hodlmoser et al. [14] uses zebra-crossings to obtain the ground plane, and pedestrians to obtain the vertical lines. The problem is the maximum distance the camera can be from the ground and the dependence on real measures from the scene. Zhang et al. [15] proposes a method to calibrate the camera based on object motion and appearance. It seems to work well on straight roads (straight and parallel motion), but the purpose of the author's method is to cover more transport infrastructures like intersections and roundabouts with different motion patterns.

Due to the wide range of possible scenes and the difficulty of extracting vanishing points automatically in most of them, the implemented method proposes an online vanishing points extracting tool. The user provides three sets of orthogonal lines at the beginning of the program (two from the ground plane and one vertical to it), and the system computes the calibration parameters to start with the image processing.

III. PROPOSED METHOD

In this section, the implemented method is described in separated subsections. An automatic calibration tool, executed in the first frame from a set of orthogonal lines, is explained in III-A; and the main process, summarized in the diagram of the Figure 1, is explained afterwards.

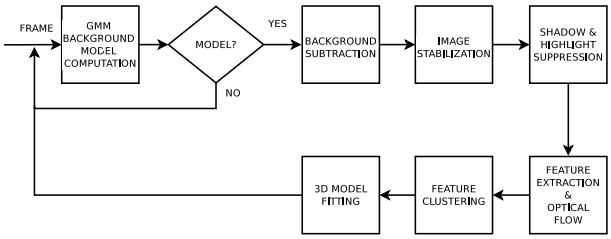


Fig. 1: Steps of the proposed method.

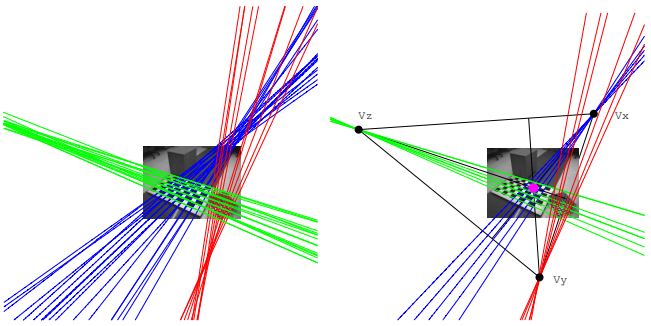
A. Camera calibration from vanishing points

In this subsection, an approach to recover the camera parameters from vanishing points is introduced. Initially the method is tested with an ideal scenario, like the one shown in Figure 2. The obtained auto calibration is compared to a supervised calibration in order to validate the method before using it in real traffic scenarios.

As said in [11], one of the distinguishing features of perspective projection is that the image of an object that stretches off to infinity can have finite extent. Particularly, parallel world lines are imaged as converging lines, and their image intersection is the vanishing point. With the assumption of camera zero skew and unit aspect ratio, the intrinsic parameter matrix K is simplified to have only 3 degrees of freedom. The three vanishing points corresponding to the three orthogonal directions in the 3D space will provide the information needed to obtain all the parameters searched.

The calibration algorithm, used at the beginning of the application, consist on the following steps:

- **Line extraction:** Interactive tool to draw and extract three sets of parallel lines, to get the three orthogonal vanishing points.
- **Vanishing points estimation:** The common image intersection points are estimated. Due to noise in the parallelism of the sets, line segments will generally not intersect in a unique point, as can be seen in Figure 2(a). A Random Sample Consensus (RANSAC) solution is then proposed to find the point which minimizes the sum of orthogonal distances to the lines. Outliers are discarded. Figure 2(b) shows how the problem is solved with a clear intersection point obtained.
- **Locate the principal point:** Under the assumption named previously, the orthocentre of the triangle formed by the three orthogonal vanishing points as vertices is the principal point (u_0, v_0). Figure 2(b) depicts an example of the vanishing point extraction result and the principal point.
- **Compute focal length and rotation angles:** as explained in [16], the focal length (f) and extrinsic parameters roll (r), pitch (p) and yaw (y), can be estimated with the following expressions. (u_{Vy}, v_{Vy}) is the image coordinate of the vertical vanishing point, and (u_{Vx}, v_{Vx}) is the image coordinate of one of the plane vanishing points.



(a) (b)

Fig. 2: (a) Initial lines extracted; (b) Lines, vanishing points and orthocentre after RANSAC.

$$r = tan^{-1}\left(\frac{u_{Vy} - u_0}{v_{Vy} - v_0}\right) \quad (1)$$

To express the following equation clearer the terms of f are split into f_1 and f_2:

$$\begin{aligned} f_1 &= \sin(r)(u_{Vx} - u_0) + \cos(r)(v_{Vx} - v_0) \\ f_2 &= \sin(r)(u_0 - u_{Vy}) + \cos(r)(v_0 - v_{Vy}) \end{aligned} \quad (2)$$

$$f = \sqrt{f_1 f_2} \quad (3)$$

Finally the pitch and yaw angles are computed by:

$$p = tan^{-1}\left(\frac{(\sin(r)(u_{Vx} - u_0) + \cos(r)(v_{Vx} - v_0)}{f}\right) \quad (4)$$

$$y = tan^{-1}\left(\frac{f}{\cos(p)(cos(r)(u_{Vx} - u_0) - \sin(r)(v_{Vx} - v_0))}\right) \quad (5)$$

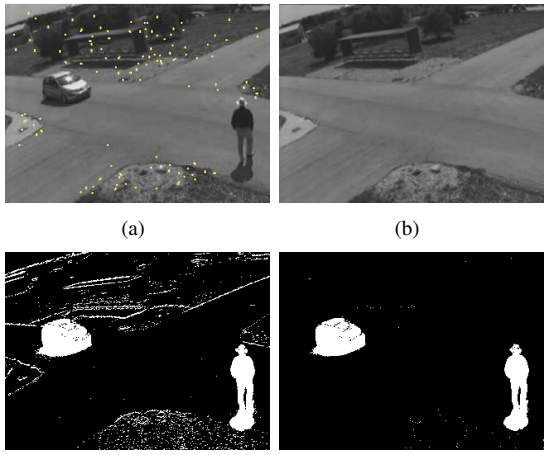
B. Background subtraction

The basic idea of background subtraction is to subtract the current image from a reference image that models the background scene. Obviously the capturing system has to be fixed and the background static. Although pedestrians and vehicles are the only objects which are moving in the field of view, the algorithm is susceptible to both global and local illumination changes such as shadows, so a detection of these problems is needed to achieve satisfying results.

Rather than explicitly modelling the values of the pixels as one particular kind of distribution, each pixel is modelled by a mixture of *K* Gaussian distributions, whose mean and variance is adapted over time. See the author's work in [17] for a complete description of the algorithm.

C. Image stabilization

Most of the traffic monitoring systems entail the use of cameras in outdoor environments. Because of that, they are exposed to vibrations and shaking due to wind, among others; which can cause visible frame-to-frame jitter and therefore foreground errors. To avoid these problems, a jitter estimation module has been developed. It captures the movement of static feature points (SURF [18]) between the current image and the background model, to estimate the camera displacement. After extracting static points, the neighbourhood of each one is represented by a feature vector and matched between the images, based on Euclidean distance. In case of having noise, erroneous measurements or incorrect hypotheses about the interpretation of data (outliers), RANSAC is used. After RANSAC has removed outliers, SURF feature pairs are used to compute the homography matrix between both images. Finally a perspective transformation based on this homography matrix is applied to the current image to compensate the movement. The result of this step can be seen in Figure 3.



(a) (b) (c) (d)

Fig. 3: Result of image stabilization step. (a) Original image with camera jitter, and SURF points; (b) modelled background; (c) extracted foreground without stabilization; (d) extracted foreground with stabilization.

D. Shadow and highlight detection

Background subtraction step detects all the moving objects that do not belong to any component of the mixture. Despite the robust detection in easy conditions, the algorithm suffers with the presence of shadows and sudden illumination changes. For this reason, a shadow and highlight detection algorithm is implemented, based on texture matching.

The technique used is the normalized cross correlation, and particularly *color normalized cross correlation* (CNCC). The idea is based on the fact that a shadow or a highlight changes color properties of the objects, but not their surface properties such as texture. The algorithm uses this method to compare the texture of every foreground pixel, by a neighbourhood window, with the correspondent one in the background model.

Let B be the background image and I an image of the video sequence. Then, considering for each foreground pixel a $(2N + 1)$ window, the NCC between the image and the background is given by (6). In the case of a color image, template summation in the numerator and each sum in the

Fig. 4: Result of shadow removal. First row: original images. Second row: foreground extraction. Third row foreground after shadow detection.

denominator is done over all of the channels, with separate mean values used for each channel.

$$NCC = \frac{E_t}{E_B E_I} \tag{6}$$

$$E_t = \sum_{n=-N}^{N} \sum_{m=-N}^{N} B(n,m) I(n,m) \tag{7}$$

$$E_B = \sqrt{\sum_{n=-N}^{N} \sum_{m=-N}^{N} B(n,m)^2} \tag{8}$$

$$E_I = \sqrt{\sum_{n=-N}^{N} \sum_{m=-N}^{N} I(n,m)^2} \tag{9}$$

For a pixel with an illumination change but similar texture, correlation is very close to 1. And in the case of shadows, the energy E_I has to be lower than E_B.

A combination of two two different space colors is used to compute the correlation. On the one hand, RGB works for soft shadows and sudden illumination changes; and on the other hand, for strong shadows the international standard CIE 1931 XYZ color space has been tested empirically with better results. Hence two different matching analysis are done together. Figure 4 shows the result of removing shadows in different conditions. Figure 5 shows the result of removing a sudden illumination change.

(a) (b) (c)

Fig. 5: Sudden global illumination change managed. (a) Original image with illumination change; (b) GMM initial foreground; (c) final foreground.

E. Feature extraction and optical flow tracking

After extracting correctly the image foreground, a new step to distinguish between different objects is done. Moreover, due to partial and global occlusions, detected objects could be fragmented, joined with a close one or even lost; so a tracking algorithm is needed. Feature-based tracking gives up the idea of tracking objects as a whole, after obtain the different regions through background subtraction. The idea of these algorithms is to extract and track foreground features and cluster them into objects using proximity, motion history, speed, orientation and the size constrains provided by the calibration.

The proposed method is called *flock of features* and it is based on the work of Kölsch et al. [19]. The concept comes from natural observation of flocks of birds or fishes. It consists of a group of members, similar in appearance or behaviour to each other, which move congruously with a simple constraint: members keep a minimum safe distance to the others, but not too separated from the flock. This concept helps to enforce spatial coherence of features across an object, while having enough flexibility to adapt quickly to large shape changes and occlusions.

Pyramid-based KLT feature tracking (Kanade, Lucas and Tomasi [20]), based on "good features to track" [21], is chosen as the main tracker where the flock constrains are applied. Features are extracted from the foreground regions and tracked individually frame to frame. Figure 6 depicts an example of the feature tracking step with a truck in a highway.

F. Feature clustering

To group all the features from the same object, mean shift clustering algorithm is used [22]. This method is a non-parametric technique which does not require prior knowledge of the number of clusters, and does not constrain their shape (it is fitted in the next module by the 3D information).

The data introduced to the function is composed by two structures: position (x,y coordinates), and motion information (speed and direction). To avoid problems with similar directions but different angle, like $359°$ and $1°$, the optical flow vector is consider as color in the HSV space (Hue=direction, Saturation=speed, Value=1), and then converted into RGB space (RBG wheel). The inputs are then position (x and y) and the three components of the RGB feature. Figure 7 shows an example of feature clustering.

Fig. 6: Feature tracking sequence.

Fig. 7: Example of feature clustering.

G. 3D model fitting

Doing feature clustering without size constrains, usually derives into overlapping cases. A common example is when two vehicles (overlapped in the foreground image) drive through the road with alike directions and speed. In that cases, position coordinates and rgb velocity components are very similar; therefore mean shift clustering can consider all features belong to the same object. To solve this kind of problems, 3D information is needed to know an approximate size of the vehicles. Then, the algorithm is able to separate the whole cluster into several subclusters corresponding to each object. Analysing the cluster shape, size and its motion pattern the algorithm finds the optimum number of objects included and its position.

IV. IMPLEMENTATION AND RESULTS

This section demonstrates the performance of the proposed algorithm in different day times and illumination conditions. Other results of the algorithm have been depicted in previous pages of the paper, in different environments and conditions, trying to cover as many situations as possible.

The system has been implemented on a Intel Core Duo CPU T2450 at 2.00GHz, running Kubuntu/Linux O.S and OpenCV libraries, with a 640x480 CMOS camera.

TABLE I: Calibration RMSE

Parameter	ROLL	PITCH	YAW	FOCAL	OP. CENTRE
RMSE	1.22°	0.71°	1.79°	14.33 pix	7.20 pix

Fig. 8: Projected volume of two cars provided by the camera calibration.

Fig. 9: Overlapping solved with 3D information.

To check if the calibration method works, the algorithm is compared with a ground-truth calibration provided by the Matlab toolbox. 15 images with similar environment but changing randomly the position and orientation of the camera, have been tested. The root mean square error (RMSE) obtained for each parameter is presented in the Table I.

After testing the method in an ideal situation, a real scene is used to check its performance and see if the obtained error in the calibration step can be assumed. A projection of the approximate volume of a vehicle is estimated and placed on top of two cars, following different directions. Figure 8 depicts the result of the estimation. Image has been edited to have different samples of both cars path, not to need multiple figures. As can be seen, the projected volume fits perfectly in both vehicles in the whole paths.

An example of the benefits of the calibration parameters obtained is explained in Figure 9. Due to similar position, speed and direction, the clustering algorithm groups two different vehicles into the same object (green vectors). Only due to the 3D information estimated by the algorithm, from structured elements of the scene, the system is able to fit the sizes of the objects and separate correctly the vehicles.

To end this section, Figure 10 demonstrate the system works in different scenarios and different light conditions: daytime, dusk and night (artificial illumination). Moreover table II shows the numerical results obtained: Number of

Fig. 10: Result of the system in different conditions.

TABLE II: Numerical results

Type of videos	Frames	Detected	Missed	False Positive
Day and dusk	52473	900	1	13
Night	2221	20	3	4

frames tested, detected objects, missed objects (not detected ones) and false positives (shadows at daytime and lighting reflections at night time). The system has been tested in day conditions (sunny, cloudy and dusk) and also in night conditions (with artificial illumination), although the last one was not the objective of this work. Nevertheless, the results are very interesting and give the chance to extend and improve the system for night conditions.

V. SUMMARY AND CONCLUSIONS

In this work, a monocular method has been developed to detect and track vehicles and other moving objects as pedestrians, for applications in the framework of ITS.

The algorithm requires no object model and prior knowledge (only an approximate size of the objects searched) and it is robust to illumination changes and shadows. Therefore it can work indoor and outdoor, in different conditions and scenarios. Moreover it is independent of the position of the camera due to the automatic calibration tool by vanishing point extraction.

The performance of the system is demonstrated via several images. Experimental results show different environments and illumination conditions and the proposed technique performs well in all of them, even with shadows.

Future work will include applying the algorithm to a larger number of data and performing comparative studies on various applications and public datasets. However the main effort will be done to improve the calibration tool to make it completely automatic. In that case, due to the adaptability of the whole algorithm, the proposed system will be able to work with variable pan-tilt-zoom cameras in fully self-adaptive mode.

VI. ACKNOWLEDGEMENTS

This work was financed by the Spanish Ministry of Economy and Competitiveness under Research Grant ONDA-FP TRA2011-27712-C02-02

REFERENCES

[1] C. Stauffer and W.E.L Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[2] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letter*, vol. 27, no. 7, pp. 773-780, 2006.

[3] A. J. Joshi and N. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *Transactions on Pattern Analysis and Machine Intelligence*, vol.30, no. 11, pp. 2055-2063, 2008.

[4] V. Reilly, B. Solmaz and M. Shah, "Geometric constraints for human detection in aerial imagery," *IEEE European Conference on Computer Vision (ECCV)*, 2010.

[5] T. Horprasert, D. Harwood and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *Proc. IEEE Int. Conf. Computer Vision FRAME-RATE Workshop*, 1999.

[6] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with HSV color information," *Proceedings of Intelligent Transportation Systems Conference*, 2001.

[7] N. Martel-Brisson and A. Zaccarin, "Learning and removing cast shadows through a multidistribution approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1133-1146, 2007.

[8] F. Porikli and J. Thornton, "Shadow flow: a recursive method to learn moving cast shadows," *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[9] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Detection and classification of highway lanes using vehicle motion trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 188-200, 2006.

[10] N. K. Kanhere, S. J. Pundlik, and S. T. Birchfield, "Vehicle segmentation and tracking from a low-angle off-axis camera," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[11] R. Hartley, A. Zisserman, "Multiple view geometry", *Cambridge university press*, 2000.

[12] C. Rother, "A new approach to vanishing point detection in architectural environments", *Image and Vision Computing*, vol. 20, no. 9, pp. 647-656, 2002.

[13] J.P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection", *IEEE International Conference on Computer Vision (ICCV)*, 2010.

[14] M. Hodlmoser, B. Micusik, M. Kampel, "Camera auto-calibration using pedestrians and zebra-crossings", *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2011.

[15] Z. Zhang, M. Li, K. Huang, T. Tan, "Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance", *CVPR*, 2008.

[16] L. Fengjun, T. Zhao, R. Nevatia, "Camera calibration from video of a walking human," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1513-1518, 2006.

[17] S. Álvarez. M.A. Sotelo, D.F. LLorca, "Monocular vision-based target detection on dynamic transport infrastructures," *Lecture Notes in Computer Science*, pp. 576-583, 2011.

[18] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346-359, 2008.

[19] M. Kölsch and M. Turk, "Fast 2D hand tracking with flocks of features and multi-cue integration," *IEEE Workshop on Real-Time Vision for Human-Computer Interaction*, 2004.

[20] B.D. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of Imaging Understanding Workshop*, 1981.

[21] J. Shi, C. Tomasi, "Good features to track," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[22] D. Comaniciu, P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619 2002.