Simultaneous Localization and Mapping in Geo-referenced Images.

S.A..Berrabah, and Y.Baudoin,

Royal Military Academy,
30 Av de la Renaissance, 1000 Brussels, Belgium
sidahmed.berrabah@rma.ac.be

I. ABSTRACT

In the event of an emergency due to a fire or other crisis, a necessary but time consuming pre-requisite, that could delay the real rescue operation, is to establish whether the ground or area can be entered safely by human emergency workers. The objective of the VIEW-FINDER research project is to develop robots which have the primary task of gathering data. The robots are equipped with sensors that detect the presence of chemicals and, in parallel, image data is collected and forwarded to an advanced Control station (COC).

One of the problems tackled in this project is the robot navigation. The used robot for the outdoor scenario is equipped with a set of sensors: stereo-camera, GPS, inertial navigation system (INS), wheel encoders, and ultrasounds sensors. The robot uses a SLAM approach to combine data from different sensors for an accurate positioning in geo-referenced images. This paper gives an overview on the used algorithms for robot positioning.

Acknowledgment: This research is founded by the View-Finder FP6 IST 045541 project.

II. Introduction

Using robotics in an incident scene needs to be with high precision. This contribution introduces the increase of mobile robot positioning accuracy using a SLAM approach. The SLAM algorithm uses data from a single monocular camera together with data from other sensors (Global Positioning System (GPS), Inertial Navigation System (INS) and wheel encoders) for robot localization in large-scale environments.

The SLAM problem is tackled as a stochastic problem and it has been addressed with approaches based on Bayesian filtering. The main problem of those approaches is that the computational complexity growth with the size of the mapped space, which limits their applicability in large-scale areas. In the case of vision based SLAM approaches, other challenges have to be tackled, as the high rate of the input data, the inherent 3D quality of visual data, the lack of direct depth measurement and the difficulty in extracting long-term features to map.

The well known vision-based approach is the MonoSLAM algorithm of Davison et al. [1]. This is a real-time SLAM approach for indoors in room-size domains, which recover the 3D trajectory of a monocular camera, moving rapidly through an unknown scene. Davison's algorithm is not suitable in larger environments.

To be able to use the monoSLAM algorithm in large areas, in this project we propose to build several size limited local maps and combine them into a global map using an 'history memory' which accumulates sensory evidence over time to identify places with a stochastic model of the correlation between map features. In our implementation, the dynamic model of the camera takes into account that the camera is on the top of a mobile robot which moves on a perfect ground-plane at all times and the SIFT feature detector [10] is used instead of Shi and Thomasi algorithm as in [1]. The SIFT features are proved to remain stable to affine distortions, change of viewpoint, noise and change in illumination [11]. Using SIFT features allows also a more reliable feature matching by using the advantage of the space-scale invariance parameters of the SIFT features.

The data from GPS are used to help localizing the robot and features in satellite images. But in some cases, the vehicle may lose the GPS due to buildings or tree canopies, and we seek to maintain an accurate robot positioning even in this case..

III. System modeling and Feature extraction

In our application, a (stereo-)camera is fixed on the top of a mobile car-like robot "ROBUDEM" (figure 1). The vehicle travels through the environment using its sensors to observe features around it. A world coordinate frame W is defined such that its X and Z axes lie in the ground plane, and its Y axis point vertically upwards.

Figure 1: The used robot in the VIEW-FINDER project.

The system state vector of the stereo-camera $\mathbf{y}_R$ in this case is defined with the 3D position vector $r = (y_1, y_2, y_3)$ of the head center in the world frame coordinates and the robot's orientations roll, pitch and yaw about the *Z*, *X*, and *Y* axes, respectively $(\gamma, \theta, \varphi)$.

$$\mathbf{y}_R = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \gamma \\ \theta \\ \varphi \end{bmatrix}$$

The dynamic model or motion model is the relationship between the robot's paste state, $\mathbf{y}_R^{t-1}$, and its current state, $\mathbf{y}_R^t$, given a control input u^t

$$\mathbf{y}_R^t = \mathbf{f}(\mathbf{y}_R^{t-1}, u^t, \mathbf{v}^t) \qquad (1)$$

Where $\mathbf{f}$ is a function representing the mobility, kinematics and dynamics of the robot (transition function) and $\mathbf{v}$ is a random vector describing the unmodelled aspects of the vehicle (process noise such as wheel sleep or odometry error).

The system dynamic model in our case, considering the control u as identity, is given by:

$$\mathbf{y}_R^t = \begin{bmatrix} y_1^t \\ y_2^t \\ y_3^t \\ \gamma^t \\ \theta^t \\ \varphi^t \end{bmatrix} = \begin{bmatrix} y_1^{t-1} + (\mathbf{v}^{t-1} + \mathbf{V})\cos(\gamma^{t-1})\Delta t \\ y_2^{t-1} + (\mathbf{v}^{t-1} + \mathbf{V})\sin(\gamma^{t-1})\Delta t \\ y_3^{t-1} \\ \gamma^{t-1} + (\omega^{t-1} + \mathbf{\Omega})\Delta t \\ \theta^{t-1} \\ \varphi^{t-1} \end{bmatrix} \qquad (2)$$

$\mathbf{v}$ and ω are the linear and the angular velocities, respectively. $\mathbf{V}$ and $\mathbf{\Omega}$ are the Gaussian distributed perturbations to the camera's linear and angular velocity, respectively.

Usually the features used in vision-based localization algorithms are salient and distinctive objects detected from images. Typical features might include regions, edges, object contours, corners etc. In our work, the map features are obtained using the SIFT feature detector [10], which maps an image data into scale-invariant coordinates relative to local features (e.g for detected SIFT features in figure 1) . These features were contemplated to be highly distinctive and invariant to image scale and rotation. The work of Mikolajczyk and Schmid [11] proved that SIFT features remain stable to affine distortions, change of viewpoint, noise and change in illumination.

Figure 1: Features detected using the SIFT algorithm

To deal with the problem of SLAM in dynamic scenes with moving object we use an algorithm for motion segmentation [17] to remove the outliers features which are associated with moving objects. In other words, the detected features which correspond to the moving parts in the scene are not considered in the built map. For more security we use a bounding box around the moving objects (figure 2). Another marge of security is used; the newly detected features are not added directly to the map but they should be detected and matched in at least *n* consecutive frames (in our application n=5).

Figure 2: Features detected in a scene with moving objects

Features are represented in the system state vector by their 3D location in the world coordinate system W:

$$\mathbf{x}_i = (x_{1,i}, x_{2,i}, x_{3,i})^T$$

The observation model describes the physics and the error model of the robot's sensor.The observations are related to the system state according to:

$$\mathbf{z}^t = \mathbf{h}(\mathbf{x}^t) + \mathbf{w}^t \qquad (3)$$

where $\mathbf{z}^t$ is the observation vector at time t and $\mathbf{h}$ is the observation model. The vector $\mathbf{z}_i^t$ is an observation at instant t of the i'th landmark location $\mathbf{x}_i^t$ relative to the robot's location $\mathbf{y}_R^t$.

Making a measurement of a feature i consists of determining its position in the camera image. Using a perspective projection, the observation model in the robot coordinate system obtained as follows:

$$\mathbf{z}_i^t = \mathbf{h}(\mathbf{x}_i^t) = \begin{bmatrix} x_0 + f\frac{{}^R x_{1,i}}{{}^R x_{3,i}} \\ y_0 + f\frac{{}^R x_{2,i}}{{}^R x_{3,i}} \end{bmatrix} \quad (4)$$

where x_0 and y_0 are the image center coordinates and f is the focal length of the camera.

${}^R\mathbf{x}_i = ({}^R x_{1,i}, {}^R x_{2,i}, {}^R x_{3,i})^T$ are the coordinates of the feature i in the robot coordinate frame R. They are related to $\mathbf{x}_i$ by:

$${}^R\mathbf{x}_i = \begin{pmatrix} cos(\gamma) & 0 & -sin(\gamma) \\ 0 & 1 & 0 \\ sin(\gamma) & 0 & cos(\gamma) \end{pmatrix} \begin{pmatrix} x_{1,i}^t - y_1 \\ x_{2,i}^t - h \\ x_{3,i}^t - y_2 \end{pmatrix} \quad (5)$$

h is the high of the camera.

The depth coordinate of the detected features is estimated by feature matching and tracking between the consecutive camera images. The matching is based on a hypothesis

$$\mathcal{H}^t = [\mathfrak{h}_1^t, \dots, \mathfrak{h}_m^t] \quad (6)$$

associating each measurement $\mathbf{z}_i^t$ with its corresponding map feature. $\mathfrak{h}_i^t = 0$ indicates that $\mathbf{z}_i^t$ does not come from any feature in the map. For data association a measure of the discrepancy between a predicted measurement that each feature would generate and an actual sensor measurement is measured by the innovation ε given by (16).

The measurement z_i^t can be considered corresponding to the feature j if the Mahalanobis distance $D_{ij}^{2\,t}$ satisfies:

$$D_{ij}^{2\,t} = \varepsilon^{T^t} \mathbf{S}^{-1^t} \varepsilon^t < th \quad (7)$$

Where the covariance $\mathbf{S}^t$ and the innovation ε^t are given by equations (15) and (16), respectively.

The state of the system at time t can therefore be represented by the augmented state vector, $\mathbf{x}^t$, consisting of the n_R states representing the robot, $\mathbf{y}_R^t$, and the n states describing the observed landmarks, $\mathbf{x}_i^t$, $i = 1, \dots, n$.

The GPS measurement, if existing, and measurement from encoders $\mathbf{v}$ and inertial sensor ω are integrated in the measurement block to produce the estimate of the state at time t based on measurements up to time t. The robot position and therefor the features position are measured in the universal GPS coordinate system (west-east, south-north).

III.1. Extended Kalman Filter for SLAM

Given a model for the motion and observation, the SLAM process consists of generating the best estimate for the system states given the information available to the system. This can be accomplished using a recursive, three stage procedure comprising prediction, observation and update of the posterior. This recursive update rule, known as filtering for SLAM, is the basis for the majority of SLAM algorithms.

Extended Kalman Filter (EKF) is the most well-known Gaussian filter for treating the SLAM problem, where the belief is represented by a Gaussian distribution. The Kalman Filter is a general statistical tool for the analysis of time-varying physical systems in the presence of noise. Its main goal is the estimation of the current state of a dynamic system by using data provided by the sensor measurements. Whenever a landmark is observed by the on-board sensors of the robot, the system determines whether it has been already registered and updates the filter. In addition, when a part of the scene is revisited, all the gathered information from past observations is used by the system to reduce uncertainty in the whole mapping, strategy known as closing-the-loop.

In EKF-based SLAM approaches, the environment is represented by a stochastic map $\mathcal{M} = (\hat{\mathbf{x}}, \mathbf{P})$, where $\hat{\mathbf{x}}$ is the estimated state vector (mean), containing the location of the vehicle R and the features of the environment $F_1 … F_n$, and **P** is the estimated error covariance matrix, where all the correlations between the elements of the state vector are defined. All data is represented in the same reference system. The map $\mathcal{M}$ is built incrementally, using the set of measurements $\mathbf{z_k}$ obtained by the camera. For each new acquisition, data association process is carried out with the aim of detecting correspondences between the new acquired features and the previously perceived ones.

$$\hat{\boldsymbol{x}}^t = E[\boldsymbol{x}^t] = \begin{bmatrix} \hat{\boldsymbol{y}}_R^t \\ \hat{\boldsymbol{x}}_1^t \\ \vdots \\ \hat{\boldsymbol{x}}_n^t \end{bmatrix}$$

$$P^t = E[(\mathbf{x}^t - \hat{\mathbf{x}}^t)(\mathbf{x}^t - \hat{\mathbf{x}}^t)^T] = \begin{bmatrix} \mathbf{P}_{RR}^t & \mathbf{P}_{R1}^t & \cdots & \mathbf{P}_{Rn}^t \\ \mathbf{P}_{1R}^t & \mathbf{P}_{11}^t & \cdots & \mathbf{P}_{1n}^t \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{nR}^t & \mathbf{P}_{n1}^t & \cdots & \mathbf{P}_{nn}^t \end{bmatrix}$$

The sub-matrices, $\mathbf{P}_{RR}^t$, $\mathbf{P}_{Ri}^t$ and $\mathbf{P}_{ii}^t$ are, respectively, the robot to robot, robot to feature and feature to feature covariances. The sub-matrices $\mathbf{P}_{ij}^t$ are the feature to feature cross-correlations. **x** and **P** will change in dimension as features are added or delated from the map.

The Extended Kalm Filter consists in two steps:

a) prediction step, which estimates the system state according to the state transition function f and the covariance matrix $\mathbf{P}$ to reflect the increase in uncertainty in the state du to noise Q (unmodelled aspects of the system) :

$$\mathbf{x}^{t|t-1} = \begin{bmatrix} f\left(\mathbf{y}_R^{t-1|t-1}, u=0\right) \\ \mathbf{x}_1^{t-1|t-1} \\ \vdots \end{bmatrix} \quad (8)$$

$$\mathbf{P}^{t|t-1} = \mathbf{F}\mathbf{P}^{t-1|t-1}\mathbf{F}^T + \mathbf{Q}^{t-1} \quad (9)$$

where

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\mathbf{x}^{t-1|t-1}} = diag\left(\frac{\partial \mathbf{f}}{\partial \mathrm{y_R}}\Big|_{\mathrm{y}_R^{t-1|t-1}}, \mathrm{I}\right) \quad (10)$$

is the Jacobian of $\mathbf{f}$ with respect to the state vector $\mathbf{x}$ and $\mathbf{Q}$ is the process noise covariance.

Considering a constant velocity model for the smooth camera motion:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{y}_R}\Big|_{\mathrm{y}_R^{t-1|t-1}} = \begin{bmatrix} 1 & 0 & -\sin(\gamma^{t-1})(\mathbf{v}^{t-1}+\mathbf{V})\Delta t \\ 0 & 1 & \cos(\gamma^{t-1})(\mathbf{v}^{t-1}+\mathbf{V})\Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

b) The Update step uses the current measurement to improve the estimated state, and therefor the uncertainty represented by P is reduced.

$$\mathbf{x}^{t|t} = \mathbf{x}^{t|t-1} + \mathbf{W}^t\varepsilon^t \quad (12)$$

$$\mathbf{P}^{t|t} = \mathbf{P}^{t|t-1} - \mathbf{W}^t\mathbf{S}^t\mathbf{W}^{t^T} \quad (13)$$

Where

$$\mathbf{W}^t = \mathbf{P}^{t|t-1}\mathbf{H}^T(\mathbf{S}^t)^{-1} \quad (14)$$

$$\mathbf{S}^t = \mathbf{H}\mathbf{P}^{t|t-1}\mathbf{H} + \mathbf{R}^t \quad (15)$$

$$\varepsilon = \mathbf{z}^t - \mathbf{h}(\mathbf{x}^{t|t-1}) \quad (16)$$

$\mathbf{Q}$ and $\mathbf{R}$ are block-diagonal matrices (obtained empirically) defining the error covariance matrices characterizing the noise in the model and the observations, respectively.

$\mathbf{H}$ is the Jacobian of the measurement model $\mathbf{h}$ with respect to the state vector. A measurement of feature $\mathbf{x}_i$ is not related to the measurement of any other feature so

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}} = [\frac{\partial \mathbf{h}_i}{\partial \mathbf{y}_R} \quad 0 \quad 0 \cdots \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_i} \quad 0 \quad \cdots] \quad (17)$$

where $\mathbf{h}_i$ is the measurement model for the i'th feature.

III.2. SLAM in Large-scale areas

The main open problem of the current state of the art SLAM approaches and particularly vision based approaches is mapping large-scale areas. Relevant shortcomings of this problem are, on the one hand, the computational burden, which limits the applicability of the EKF-based SLAM in large-scale real time applications and, on the other hand, the use of linearized solutions which compromises the consistency of the estimation process. The computational complexity of the EKF stems from the fact that covariance matrix $\mathbf{P}$ represents every pairwise correlation between the state variables. Incorporating an observation of a single landmark will necessarily have an effect on every other state variable. This make the EKF computationally infeasible for SLAM in large environment.

To solve the problem of SLAM in large spaces, in our study, we propose a procedure to break the global map into submaps by building a global representation of the environment based on several size limited local maps built using the previously described approach. The global map is a set of robot positions where new local maps started (i.e. the base references of the local maps). The base frame for the global map is the robot position at instant t_0. Each local map is built as follows: at a given instant t_k, a new map is initialized using the current vehicle location,$\mathbf{y}_R^{tk}$, as base reference B_k= $\mathbf{y}_R^{tk}$, $k=1, 2,... being the local map order. Then, the vehicle performs a limited motion acquiring sensor information about the L_i neighboring environment features.

The 'k'th local map is defined by:

$$\mathfrak{M}_k = (\mathbf{X}_k, \mathbf{P}_k)$$

where $\mathbf{X}_k$ is the state vector in the base reference B_k of the L_k detected features and $\mathbf{P}_k$ is their covariance matrix estimated in B_k.

The decision to start building a new local map at an instant t_k is based on two criteria: the number of features in the current local map and the scene cut detection result. The instant t_k is called a key-instant. In our application we defined two thresholds for the number of features in the local maps: a lower Th^- and a higher Th^+ thresholds. A key-instant is selected if the number of features n_l^k in the current local map k is bigger then the lower threshold and a scene cut has been detected or the number of features has reached the higher threshold. This allows kipping reasonable dimensions of the local maps and avoids building too small maps.

The global map is:

$$\mathfrak{M}_G^B = (\bar{\mathbf{y}}_R^0, \bar{\mathbf{y}}_R^1, \bar{\mathbf{y}}_R^2, \dots) \quad (18)$$

where $\bar{\mathbf{y}}_R^k$ are the robot coordinates in B_0, where it decides to build the local map $\mathfrak{M}_k$ at instant t_k.

$$\begin{pmatrix} \bar{\mathbf{y}}_R^k \\ 1 \end{pmatrix} = \mathcal{T}_{k\to 0} \cdot \begin{pmatrix} \mathbf{y}_R^{t_k} \\ 1 \end{pmatrix} \quad (19)$$

$$t_0 = 0 \text{ and } \bar{\mathbf{y}}_R^0 = \mathbf{y}_R^{t_0} = (0,0,0).$$

The transformation matrix $\mathcal{T}_{k\to 0}$ is obtained by successive transformations:

$$\mathcal{T}_{k\to 0} = \mathcal{T}_{1\to 0}.\mathcal{T}_{2\to 1}\ldots\mathcal{T}_{(k-1)\to(k-2)} \qquad (20)$$

where $\mathcal{T}_{i\to i-1} = (\mathcal{R}|\mathbf{t})$ is the transformation matrix corresponding to rotation $\mathcal{R}$ and translation $\mathbf{t}$ of frame B_i regarding to frame B_{i-1}:

$$\mathcal{T}_{i\to i-1} = \begin{pmatrix} \cos(\gamma^{t_i}) & 0 & -\sin(\gamma^{t_i}) & y_1^{t_i} \\ 0 & 1 & 0 & 0 \\ \sin(\gamma^{t_i}) & 0 & \cos(\gamma^{t_i}) & y_2^{t_i} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (21)$$

In this case, for feature matching at instant t, the robot uses the local map with the closest base frame to its current location:

$$\underset{i}{\operatorname{argmin}}(\mathrm{P}\bar{\mathbf{y}}_R^k - \bar{\mathbf{y}}_R^t\mathrm{P}) \qquad (22)$$

where $\bar{\mathbf{y}}_R^t$ is the robot position at instant t in B_0.

Figure 3 shows an example for the use of our algorithm for ROBUDEM localization in a real environment.

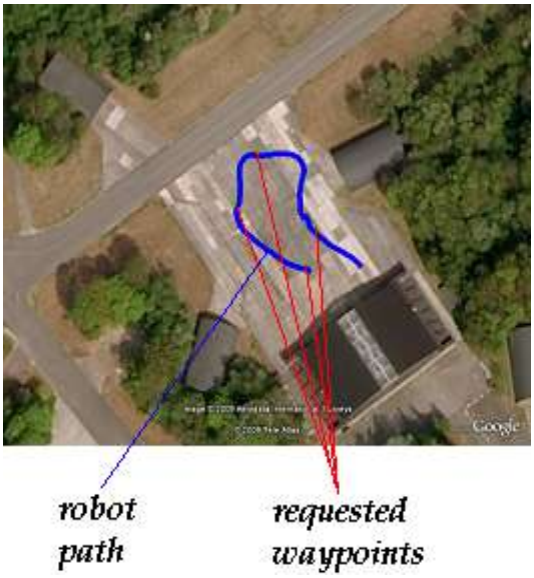

Figure 3: ROBUDEM localization in a real environment.

IV. References

[1] J. Davison, Y. G. Cid, N. Kita, *Real-time 3D SLAM with wide-angle vision*, In Intelligent Autonomous Vehicles, Lisboa-Portugal, July 2004.

[2] J. Folkesson, P. Jensfelt, H. Christensen, *Graphical SLAM using vision and the measurement subspace*, In IEEE/JRS -Intl Conf. on Intelligent Robotics and Systems (IROS), Edmundton-Canada, August, 2005.

[3] D. Wolf, G.S. Sukhatme, *Online Simultaneous Localization and Mapping in Dynamic Environments*, Proceedings of the Intl. Conf. on Robotics and Automation ICRA New Orleans, Louisiana, April, 2004.

[4] S. Se, D. Lowe, J. Little, *Local and Global Localization for Mobile Robots using Visual Landmarks*, Proceedings of the International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, Oct. 29 - Nov. 03, 2001, pp.414-420.

[5] J. A. Castellanos, J. Neira, J. D. Tards, *Multisensor fusion for simultaneous localization and map building*, IEEE Trans on Robotics and Automation, December 2001, Vol.17, N.6, pp.908-914.

[6] F. Andrade-Cetto, A. Sanfelin, *Concurrent Map Building and Localization with landmark validation*, 16th Intenational Conference on Pattern Recognition ICPR'02, 2002, vol.2.

[7] J. W. Fenwick, P. M. Newman, J. J. Leonard, *Cooperative Concurrent Mapping and Localization*, Proceedings of the 2002 IEEE International Conference on Robotics and Automation, May 2002, Washington, USA, pp.1810-1817.

[8] S. Thrun, D. Fox, W. Burgard, *A probabilistic approach to concurrent Mapping and Localization for Mobile Robots* , Machine Learning, 1998, Vol.31, N.1-3, pp.29-53.

[9] J. Davison, I. D. Reid, N. D. Molton, O. Stasse, *MonoSLAM: Real-Time Single Camera SLAM*, IEEE Transaction on Pattern Analysis and Machine Intelligence, JUNE 2007, Vol.29, N.6.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110.

[11] J. A. Castellanos, J. Neira, J. D. Tardos, Map Building and SLAM Algorithms, in S. S. Ge and F. L. Lewis (Eds), Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications, Series in Control Engineering, CRC, Taylor & Francis Group, May 2006, pp.335-371.

[12] K. Mikolajczyk, C. Schmid. *A performance evaluation of local descriptors*, Proceedings of Computer Vision and Pattern Recognition, 2003

[13] I. Bailey, Mobile robot localisation and maping in extensive outdoor environments. PhD thesis, Australian Centre for Field Robotics, University of Sydney, Australia, August 2002.

[14] J.D. Trados, J. Neira, P. Newman, J. Leonard, Robust mapping and localization in indoor environments using sonar data, International Journal of Robotics Research, 2002, N. 21, pp.311-330.

[15] S. B. Williams, Efficient Solutions to Autonomous Mapping and Navigation Problems, PhD thesis, Australian Centre for Field Robotics, University of Sydney, Australia, September 2001.

[16] C. Estrada, J. Neira, J. D. Tardos, Hierarchical SLAM: real-time accurate mapping of large environments, IEEE Transactions on Robotics, 2005, Vol.21, N.4, pp.588-596.

[17] S. A. Berrabah, G. De Cubber, V. Enescu, H. Sahli, "MRF-based foreground detection in image sequences from a moving camera", Accepted for the Thirteenth International Conference on Image Processing (ICIP 2006), which will be held in Atlanta, GA USA, in October 2006.