Occlusion Analysis: Learning and Utilising Depth Maps in Object Tracking

D. Greenhill, J. Renno. J. Orwell and G.A. Jones
Digital Imaging Research Centre, Kingston University
Penrhyn Road, Kingston upon Thames, Surrey, KT1 2EE
{d.greenhill,j.renno,j.orwell,g.jones}@kingston.ac.uk

Abstract

Complex scenes such as underground stations and malls are composed of static occlusion structures such as walls, entrances, columns, turnstiles, barriers, *etc*. Unless this occlusion landscape is made explicit such structures can defeat the process of tracking individuals through the scene. This paper describes a method of generating the probability density functions (PDFs) for the depth of the scene at each pixel from a training set of detected blobs *i.e.* observations of detected moving people. As the results are necessarily noisy, a regularization process is employed to recover the most self-consistent scene depth structure. An occlusion reasoning framework is proposed to enable object tracking methodologies to make effective use of the recovered depth.

1 Introduction

Accurately tracking moving objects within monitored scenes is crucial to a range of surveillance tasks. By far the most common approach to monitoring in typical CCTV imagery uses *pixel differencing* in static scenes[8], *blob analysis* to extract observations of moving objects, and *trajectory tracking* to establish the temporal history of individual scene events. The most significant challenge to this otherwise successful approach is the frequent problem of occlusion. Complex scenes such as underground stations and malls are composed of static occlusion structures such as walls, entrances, columns, turnstiles, barriers, *etc*. These structures induce *partial occlusion* which significantly distorts the expected position and appearance of tracked object, or *full occlusion* for different lengths of time during which the object completely disappears (and need not necessarily reappear).

This paper describes a method of establishing the depth of the scene structure by generating the depth probability density functions (PDFs) at each pixel from a training set of detected blobs *i.e.* observations of detected moving people. The connected region of pixels associated with each observation of a person occludes some static scene element such as a wall whose depth is unknown but required. Such an observation constrains the occluded structure to lie at some distance beyond the observation. Thus, assuming that the training set of observed people explore all the navigable space in the scene, the union of their 3D trajectories will approximate to the depth structure of the scene. In particular, the occlusion surfaces will appear as depth discontinuities. After regularisation, the depth map will support reasoning about static occlusions. As an object proceeds through the

BMVC 2004 doi:10.5244/C.18.49

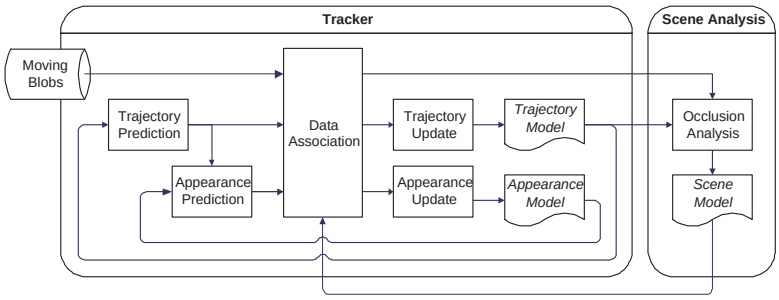


Figure 1: Tracking Architecture

scene, inter-frame correspondence - which typically involves a spatio-chromatic comparison of observation pixels with an *appearance model* - becomes problematic where there has been significant occlusion of the observation. The depth map allows the visibility of the object to be predicted and the accuracy of correspondence to be improved.

Review of Object Tracking

The typical object tracking architecture is presented in Figure 1. *Motion detection* processes locate *blobs* (connected regions of moving pixels) to create a candidate list of observations of the current active scene *objects*. Typically these blobs are recovered by pixel differencing against a reference frame of the static scene, usually attributed with their bounding box, orientation and centroid. The *Tracker* module is implemented using a *hypothesize, validate and update* approach. Each active scene *Object* has an associated *Trajectory Model* which describes the current position, velocity and possibly acceleration of the object in - typically - image coordinates. In addition, each *Object* has an associated *Appearance Model* which may be used to identify those blobs with the most similar shape and/or chromatic structure. Such appearance models may simply describe the expected width and height of the object's bounding box, or record the pixel greylevels within the last bounding box. Dynamic variants of these may also describe the rate of change of these dimensions[4]. More sophisticated models may record the contour, binary pixel shape, or spatio-chromatic structure. *Active* or statistical appearance models attempt to learn the allowable variation in object appearance[7]. In the *hypothesis* phase of the procedure, the object position and appearance is predicted from the trajectory and appearance models. Each active scene object is then *validated* by locating an appropriate corresponding observation from the list of candidate observations in the *Data Association*[1] step. *Greedy matching* is a common local approach to establishing correspondences in which the observation closest (using the Mahalanobis distance metric) to the predicted position of an object is selected. In addition to incorporating appearance information, more sophisticated *global* matching approaches attempt to enforce the uniqueness constraint by considering all possible object-observation pairings[2]. Unmatched observations may be used to hypothesise new objects appearing within the scene. In the *update* phase, the position and appearance of each corresponding observation is used to update the trajectory and appearance model of validated scene objects. Typical update mechanisms include the $\alpha - \beta$ and Kalman filter. Fundamentally, the tracker maintains the temporal coherence of object identities. Trajectory and appearance models maximise this temporal coherence.

Within tracking methodologies, *occlusion analysis* refers to two largely distinct processes. First, the occlusion of one moving object by other moving objects which causes particular difficulties in continuing to establish the temporal identity. A common but by no means satisfactory solution is to suspend both the correspondence and update phase of the tracker until the objects are predicted to reappear. Occlusions are *predicted* by checking for pairwise overlap of bounding boxes at predicted positions. Suspending the update phase for any length of time, however, is problematic since motions (particularly of people) can rapidly evolve. A recent simple but effective approach is to track the boundaries of bounding boxes separately which results in at least some updating evidence recovered for a substantial proportion of the occlusion event[9]. Another recent approach is to match the appearance model to partially obscured observations using depth to control the order in which appearance models are matched[7].

The second type of occlusion analysis involves the interaction between moving objects and static occluding structures with in the scene such as walls, entrances, columns, turnstiles, barriers *etc* or the image boundary itself. Recently Senior[7] automatically identified an *occlusion map* defined as pixels modelled as part of the reference image but which were never occluded. Of course, occluding surfaces can be also be occluded by moving objects. Using bounding boxes, Ellis and Xu manually identified *long-term*, *short-term* and *border* occlusions, and employed a Bayesian network to infer about the status of unmatched blobs as they interacted with these occlusion structures. However, the most effective occlusion representation is in fact a depth map. Using a similar philosophy to that proposed in this paper, Schödl and Essa used detected moving objects to infer the relative depth structure[6]. Since these blobs lacked any depth assignment, they were forced to employ an extremely time-consuming search process based on minimum description length to partition the image into relatively few depth planes. No specific mechanism was proposed to make use of this depth representation.

2 Constructing Depth Probability Density Functions

Complex scenes such as underground stations and malls are composed of static occlusion structures such as walls, entrances, columns, turnstiles, barriers, *etc*. Unless this occlusion landscape is made explicit such structures can defeat the process of tracking individuals through the scene. This section describes a method of generating the probability density functions (PDFs) of depth at each pixel location from a training set of detected blobs *i.e.* observations of detected moving people whose depth from the camera is known.

The connected region of pixels associated with each observation of a person occludes some scene structure such as a wall whose depth is unknown but required. Such an observation constrains the occluded structure to lie at some distance beyond the observation. Thus, assuming that the training set of observed people explore all the navigable space in the scene, the union of their 3D trajectories will approximate to the depth structure of the scene. In particular, the occlusion surfaces will appear as depth discontinuities.

2.1 Defining a depth measure

The *ground plane* relates the depth of a person to the projected image position of his/her feet. Thus assuming a point belongs to the ground plane, its row position correlates with

depth from the camera. Rather than relying on performing the necessary calibration, a shifted version of the row position is used as a non-linear proxy D for this depth.

The projected pixel height h of a person in an $H \times W$ image plane is a function of the vertical position of the person in the image *i.e.* distant people standing on the ground plane are both higher up the image and smaller than nearby objects. For typical camera installations, this relationship is practically linear[5] *i.e.*

$$h = \alpha(i_b - i_h) \qquad \text{or} \qquad h = \frac{\alpha}{1-\alpha}(i_t - i_h) \quad \text{where} \quad i_b = i_t + h \tag{1}$$

where i_b - the position of the feet (or alternatively i_t the top of the person's head) - is related to the *height expansion rate* α and horizon i_h. These latter constants must be manually (or automatically[5]) learnt from observations of people events. Assuming that a person's head is unoccluded in typical scenes, it is possible to estimate the image location of the feet of an average person i_b from the head using equation 1.

The origin of the depth axis D is defined by the row position i_0 of the feet of an average-sized person whose head is located just below the bottom edge of the image. Thus D which increases as $i_0 - i_b$ increases (or alternatively $i_0 - i_t$) can be defined as

$$D_b(i_b) = \frac{H - \alpha i_h}{1-\alpha} - i_b \qquad \text{or} \qquad D_t(i_t) = \frac{H - i_t}{1-\alpha} \tag{2}$$

The most distant object would be located at the horizon i_h corresponding to a *depth* $D_{\max} = (H - i_h)/(1-\alpha)$. Points above the horizon are assumed to lie at this maximum depth.

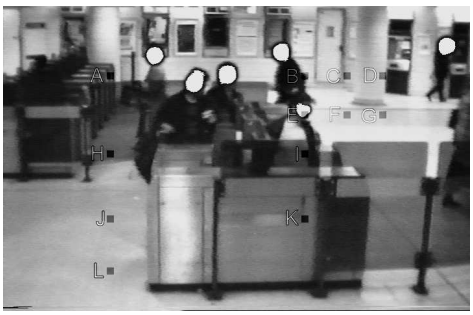
2.2 Recovering the pixels depth PDFs

A priori, the depth D of any pixel ϕ at row i in the image plane is assumed to belong to a uniform density function between the limits $[0, \min(D_b(i), D_{\max})]$ *i.e.* it cannot lie at a distance $D_b(i)$ deeper than the ground floor element that projects to that pixel. We can use observations of moving people (*i.e.* blobs) to eliminate depths closer than the observed object. All pixels belonging to a single moving person can be assigned a proxy distance $D_t(i_t)$ given by the topmost row location of the person's head. For each pixel ϕ we maintain a depth histogram $z_\phi(D)$ in the range $[0, \min(D_b(i), D_{\max})]$ which we increment at the depth associated with any blob which contains the pixel. From this histogram we generate a PDF $\Pi_\phi(D)$ by accumulating over the training set and normalising as follows

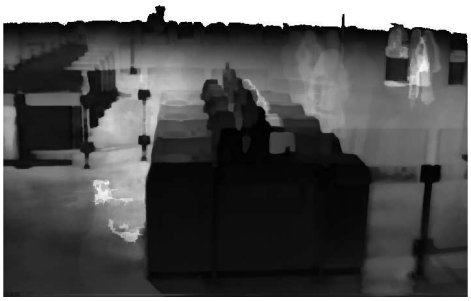
$$\Pi_\phi(D) = \begin{cases} \frac{1}{A_\phi} \sum_{d=0}^{D} \left\{ z_\phi(d) - \delta \right\} & \text{if } D < \min(D_b(i), D_{\max}), \\ 0 & \text{else.} \end{cases} \tag{3}$$

$$A_\phi = \sum_{d=0, \min(D_b(i), D_{\max})} \left\{ z_\phi(d) - \delta \right\} \tag{4}$$

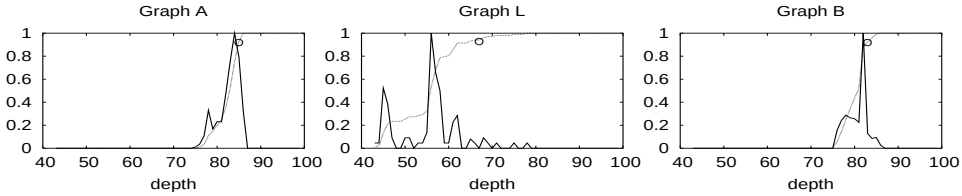
Typically these PDFs exhibit a plateau beyond the deepest occluding observation. The subtraction of a small constant δ biases the result towards the closest viable depth. Equation 4 generates an activity map for the scene. Figure 2(b) illustrates the activity recovered from 15,000 frames of the *Underground* scene in Figure 2(a). Figures 2(c) to (h) present a set of such PDFs for a number of positions in the original underground scene in Figure 2(a). An initial depth field $Z_\phi = \text{argmax}_D \, \Pi_\phi(D)$ is shown in Figure 2(a). While noisy, the depths and occlusion edges are nonetheless evident.



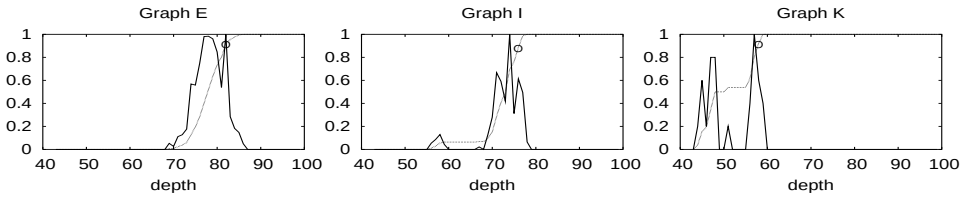
(a) Underground Scene



(b) Activity Map

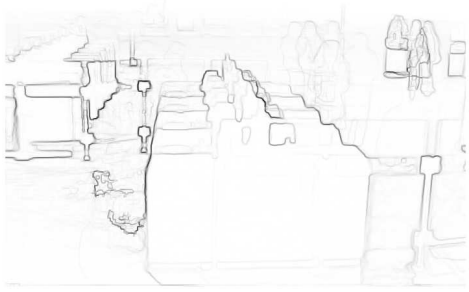


(c) Sample A (d) Sample L (e) Sample B



(f) Sample E (g) Sample I (h) Sample K

(i) Scene Depth Map



(j) Occlusion Map

Figure 2: Initial Depth Map and Occlusion Structure

3 Regularization of Depth

The depth probability density functions for each pixel are noisy for a number of reasons. First the paths taken by individuals do not necessarily visit all the available ground plane. Second, fragmentation and merging of detected regions generates erroneous depth estimates. For this reason it is necessary to regularize the depth fields *i.e.* spatially smooth the depth field while preserving the depth discontinuities. One method is to cast the regularization as an optimization problem which, on minimization, generates the optimal assignment of depth labels to all pixels in the image.

3.1 Defining the Optimization Functional

In the following functional, each pixel ϕ in an image $\mathscr{I}$ has a label θ_ϕ drawn from the set of integer depths $\Lambda = \{0, D_{\max}\}$. The goal is to recover the most probable depth label given the pixel's PDF and contextual information. Each possible label assignment $\theta_\phi = \lambda$ has an associated probability $p(\theta_\phi = \lambda)$ such that $\sum_{\lambda\in\Lambda} p(\theta_\phi = \lambda) = 1$.

$$\begin{aligned} F\left(\, p(\theta_\phi = \lambda); \forall\{\phi,\lambda\} \in \mathscr{I}\times\Lambda \,\right) &= \sum_{\phi\in\mathscr{I},}\ \sum_{\lambda\in\Lambda} \Pi_\phi(\lambda)\ p(\theta_\phi = \lambda) \\ &+ \beta \sum_{\phi\in\mathscr{I},}\ \sum_{\lambda\in\Lambda,}\ \sum_{\phi'\in\mathscr{N}_\phi,}\ \sum_{\lambda'\in\Lambda} C(\theta_\phi = \lambda, \theta_{\phi'} = \lambda') p(\theta_\phi = \lambda) p(\theta_{\phi'} = \lambda') \\ &- \gamma \sum_{\phi\in\mathscr{I}} \left\{1 - \sum_{\lambda\in\Lambda} p(\theta_\phi = \lambda)\right\}^2 \end{aligned} \tag{5}$$

where $\mathscr{N}_\phi$ is the 4-connected pixel neighbourhood of a pixel ϕ. The first term of this functional biases the functional towards solutions which select high $\Pi_\phi(\lambda)$ PDF values *i.e.* faithful to the observation information. The second term is designed to favour mutually consistent labelling of neighbouring pixels by returning high values for consistent neighbouring labels *i.e.* it tends to favour locally smooth interpretations. The final penalty term imposes the requirement that a pixel's label probabilities sum to one. The constants β and γ encode the influence of the three information sources. Expanding this third term and re-arranging, generates the following related quadratic

$$\begin{aligned} F'\left(\, p(\theta_\phi = \lambda); \forall\{\phi,\lambda\} \in \mathscr{I}\times\Lambda \,\right) &= \sum_{\phi\in\mathscr{I},}\ \sum_{\lambda\in\Lambda} \chi(\theta_\phi = \lambda) p(\theta_\phi = \lambda) \\ &+ \sum_{\phi\in\mathscr{I},}\ \sum_{\lambda\in\Lambda,}\ \sum_{\phi'\in\mathscr{N}'_\phi,}\ \sum_{\lambda'\in\Lambda} \Gamma(\theta_\phi = \lambda, \theta_{\phi'} = \lambda') p(\theta_\phi = \lambda) p(\theta_{\phi'} = \lambda') \end{aligned} \tag{6}$$

where $\mathscr{N}'_\phi = \mathscr{N}_\phi \cup \{\phi\}$, equations 5 and 6 are related as $F(.) = F'(.) + \sum_{\phi\in\mathscr{I}}$, and

$$\begin{aligned} \chi(\theta_\phi = \lambda) &= \Pi_\phi(\lambda) + 2\gamma \\ \Gamma(\theta_\phi = \lambda, \theta_{\phi'} = \lambda') &= \begin{cases} \beta C(\theta_\phi = \lambda, \theta_{\phi'} = \lambda') & \text{if } \phi \neq \phi', \\ -\gamma & \text{else.} \end{cases} \end{aligned}$$

3.2 Recovering the Regularised Depth

Minimisation is achieved using a *Hopfield neural network* formulation[3] as described in Section 3.4. The regularized depth field Z'_ϕ is recovered as the most probable label *i.e.*

$$Z'_\phi = \text{argmax}_{\lambda\in\Lambda}\ p(\theta_\phi = \lambda) \tag{7}$$

3.3 Compatibility of Neighbouring Depth Labels

The compatibility $C(\theta_\phi = \lambda, \theta_{\phi'} = \lambda')$ measures the consistency between pixel ϕ having depth λ and its neighbour ϕ' having depth λ'. In general, to encourage locally smooth interpretations, large incompatibility values should be generated for depth discontinuities **unless** these pixels straddle an occluding boundary. Such occluding boundaries can be signalled by discontinuities in the activity map of equation 4 - see Figure 2(b). Given the gradient magnitude E of the activity field A, compatibility ($-1 \leq C \leq 1$) is defined as

$$C(\theta_\phi = \lambda, \theta_{\phi'} = \lambda') = 2\exp\left\{-\frac{\sigma_E}{\sigma_\lambda}\frac{|\lambda - \lambda'|}{\left(\sigma_E + \max(E_\phi, E_{\phi'})\right)}\right\} - 1 \tag{8}$$

where σ_E and σ_λ are the standard deviations of the gradient magnitude E and depth map.

3.4 Minimizing the Optimisation Functional

Quadratic equations of the form given in equation 9 may be minimized using a *Hopfield neural network*[3]. In our problem, each pixel-label pair $\{\phi, \lambda\}$ may be viewed as a neuron. On comparison with equation 6, an external input potential I_i for each neuron may be derived from $\chi(\theta_\phi = \lambda)$ of equation 7. Each neuron is connected to other neurons by an interconnection potential Φ_{ij} given by $\Gamma(\theta_\phi = \lambda, \theta_{\phi'} = \lambda')$ of equation 7.

$$E = \sum_i I_i V_i - \frac{1}{2}\sum_i \sum_j \Phi_{ij} V_i V_j \tag{9}$$

where V_i, the output potential of the neuron, is equivalent to the pixel-label probability $p(\theta_\phi = \lambda)$. The dynamic update equation of the network is given by

$$\frac{du_i}{dt} = \frac{1}{C}\left[I_i - \frac{u_i}{R} + \sum_{j \neq i} \Phi_{ij} V_j\right] = \frac{1}{C}\left[I_i - \frac{u_i}{R} + \sum_{j \neq i} \Phi_{ij} g(u_j)\right] = F_i(\mathbf{u}) \tag{10}$$

where the internal state u_i of a neuron is related to output potential V_i by the function $V_i = g(u_i)$, and $\mathbf{u} = (u_1, \ldots, u_N)$. This non-linear function is often a sigmoidal function - in this case $V = 0.5\{1 + \exp(-\kappa u)\}^{-1}$ where κ is the gain of the sigmoid. A 4$^{\text{th}}$ order Runge-Kutta procedure is employed to iteratively and accurately solve equation 10.

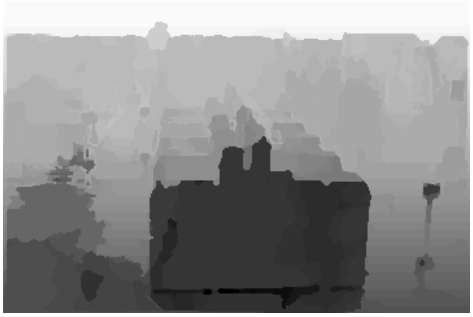
3.5 Results

Two image sequences have been used to evaluate the methodology - the *Underground* (8000 frames) and *Airport* scene (3000 frames) shown in Figures 2(a) and 3(a) respectively[1]. Both enjoy complex 3D structures, and despite being indoor scenes, both are highly problematic containing illuminance of high dynamic range and composed of highly reflective surfaces. Figures 3(c) and (d) present the regularized depth of these scenes. The *stepped* depth of the ticket barriers in the Underground has become spatially coherent. In both sequences, the ground plane has become much more evident. However both exhibit regions where there has been insufficient observations (see the activity maps) to resolve the depth structure. In particular, the depth of furthest regions is determined by the deepest observations. In addition, the clock signal appears as a nearby scene element.

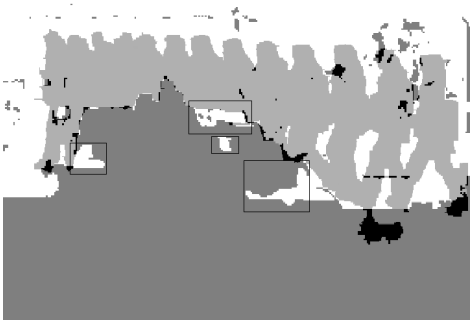
[1] Some editing of presented images has been necessary for legal and security reasons

(a) Airport Scene

(b) Airport Activity Map



(e) Tracked Object



(f) Occlusion Result

Figure 3: Building and using Scene Structure

No objective evaluation of the computed depths has yet been undertaken. Nonetheless the accuracy and utility of the method can be illustrated by generating, for the tracked person shown in Figure 3(e), the *occlusion mask* corresponding to its first frame depth. Black and dark grey colours refer to pixels whose depth is closer to the camera than the person (and white beyond). Black regions are pixels which incorrectly occlude the person (although one of these regions is a ground plane shadow). Boxes have been drawn around white regions which should have been identified as occluding. Despite these errors, the results generated from the approximately 800 people in the 8000 frames are very promising.

4 Occlusion Analysis

Data association (which establishes correspondence between observations and tracked objects) typically involves pruning observations whose area is significantly smaller than the expected area, and a chromatic (or spatio-chromatic) comparison of observation pixels with an *appearance model* maintained as part of the objects' representation. Such operations become problematic where there has been significant occlusion of the observation.

While there exists a number of approaches for modelling the appearance of tracked objects, it has been straightforward to embed our depth field within the the depth-dependent bounding box approach of Renno *et al*[5]. Here, for typical surveillance installations, projected height is modelled as a linear function of the image row position of the point of contact of a person with the ground plane - see equation 1. (The width of a person is assumed to be some fixed proportion of their height). Thus, assuming that the head of a person is never occluded, the unoccluded image-plane bounding box of any occluded observation is derived solely from the image location! The primary attraction of their method is the elimination of the need to temporally estimate the bounds of the object's appearance. Nonetheless, the following methodology could in principle be extended to other approaches. Using the work presented in this paper, this bounding box representation can also be attributed with an estimate of its depth which can be directly related to the depth field using the proxy depth measure defined in Section 2.1.

To illustrate the use of depth in the data association stage, we adapt the probabilistic cross-correlation filter of Senior[7]. Within a bounding box $\mathscr{B}_\psi$ for each object ψ, a simple probabilistic appearance model $M(\phi); \phi \in \mathscr{B}_\psi$ is generated from the temporally smoothed chromatic signature at each pixel. A probability map P^M_ψ defined over the bounding box $\mathscr{B}_\psi$ records the likelihood of a pixel belonging to the model.

For each observation ω, the projected bounding box $\mathscr{B}_\omega$ is derived solely from its image location[5]. A pixel mask $\mathscr{O}_\omega$ whose size is given by the bounding box $\mathscr{B}_\omega$) is used to predict occluded pixels $\phi \in \mathscr{B}_\omega$ by comparing the depth estimate D_ω of the observation with the depth map Z'_ϕ of the scene.

The first (and original[7]) term of the cross-correlation filter expressed in equation 11, measures the log-likelihood of the greylevel structure in the current image I_t given the appearance model M_{t-1} over the predicted unoccluding pixels. The second term effectively adds a log-likelihood prior probability for predicted occluding pixels. The pixel by pixel comparison is performed on the aligned bounding boxes $\mathscr{B}_\omega$ and $\mathscr{B}_\psi$ over a search window $\mathscr{S}$ of displacements $\tau \in \mathscr{S}$ to allow for error in predicted object position.

$$L(\omega,\psi) = \min_{\tau \in \mathscr{S}} \left\{ \sum_{\phi \in \mathscr{B}_\psi \cap \mathscr{O}_\omega} \left[\frac{\left(I_t(\phi) - M_{t-1}(\phi+\tau)\right)^2}{2\sigma^2} + \log P^M_\psi(\phi+\tau) \right] + \sum_{\phi \in \mathscr{B}_\psi \cap \overline{\mathscr{O}}_\omega} \log P_O \right\} \tag{11}$$

The above formulation is presented to illustrate how easily the depth map can be incorporated into the data association stage of the tracking algorithm. Unfortunately no quantitative evaluation results were currently available at the time of writing.

5 Discussion

A method of generating the probability density functions (PDFs) of the likely depth of the scene at each pixel is presented. This learning approach uses a training set of observations of detected moving people, each of which constrains part of the occluded scene to lie at some distance beyond the observation. Since the results tend to be noisy, a regularisation process is required. Occlusion boundaries generate discontinuities in the activity map which can be used to prevent the smoothing of depth over possible depth boundaries.

Having extracted the depth scene, we have illustrated in Section 4 how the depth map can aid the inter-frame correspondence problem which is so highly sensitive to occlusion. In addition, the approach can also be integrated into *layered* approaches[7] for reasoning about dynamic occlusion between moving objects where depth clues order the data association process.

Each iteration of minimisation process is computationally burdensome, and typically takes several hundred iterations to converge. However computational speed is not really an issue as it is important to perform the accumulation of the pixel depth histograms until the scene population has explored all the available space. While it is attractive to imagine an online adaptive version, it is not clear what impact such a variant of the approach would have on the convergence properties of the minimisation process.

References

[1] Y. Bar-Shalom and T. Fortmann. *"Tracking and Data Association"*. Mathematics in Science and Engineering. Academic Press, 1988.

[2] T. Ellis and M. Xu. "Object Detection and Tracking in an Open and Dynamic World". In *Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Hawaii, December 2001.

[3] J. Hopfield. "Neurons with Graded Responses have Collective Computation Properties like those of Two-State Neurons". *Proceedings of the National Academy of Sciences, USA*, 81:3088–3092, 1984.

[4] Justus H. Piater, Stephane Richetto, and James L. Crowley. "Event-based Activity Analysis in Live Video using a Generic Object Tracker". In *Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–8, Copenhagen, June 1 2002.

[5] J. Renno, J. Orwell, and G.A. Jones. "Learning Surveillance Tracking Models for the Self-Calibrated Ground Plane". In *British Machine Vision Conference*, Cardiff, UK, September 2002.

[6] A. Schodl and I. Essa. "Depth Layers from Occlusions". In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 639–644, Kawai, Hawaii, December 2001.

[7] Andrew Senior. "Tracking People with Probabilistic Appearance Models". In *Third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 48–55, Copenhagen, June 1 2002.

[8] C. Stauffer and W.E.L. Grimson. "Learning Patterns of Activity using Real-Time Tracking". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

[9] M. Xu and T.J. Ellis. "Partial observation versus blind tracking through occlusion"'. In *British Machine Vision Conference*, pages 777–786, Cardiff, September 2002.