www.ietdl.org

Published in IET Computer Vision
Received on 30th November 2012
Revised on 20th June 2013
Accepted on 20th June 2013
doi: 10.1049/iet-cvi.2012.0287

ISSN 1751-9632

Depth order estimation for video frames using motion occlusions

Guillem Palou, Philippe Salembier

Department of Signal Theory and Communications, Technical University of Catalonia (UPC), Barcelona, Spain
E-mail: guillem.palou@upc.edu

Abstract: This study proposes a system to estimate the depth order of regions belonging to a monocular image sequence. For each frame, the regions are ordered according to their relative depth using information from the previous and following frames. The algorithm estimates occlusions relying on a hierarchical region-based representation of the image by means of a binary tree. This representation is used to define the final depth order partition which is obtained through an energy minimisation process. Finally, to achieve a global and consistent depth ordering, a depth order graph is constructed and used to eliminate contradictory local cues. The system is evaluated and compared with the state-of-the-art figure/ground labelling systems showing very good results.

1 Introduction

Depth perception in human vision relies on several depth cues. For close objects, humans accurately estimate depth making use of both eyes and inferring disparity between two views. However, when objects are distant or when only one viewpoint is available, it is also possible to partially estimate the scene structure through the so-called 'monocular depth cues'. In static images, T-junctions or convexity cues are classical depth cues. In video sequences, motion information can also be used to obtain depth information. For example, occlusion of moving objects, size change or motion parallax are used to structure the scene [1].

Nowadays, motivated by the film industry, many research works are focusing on depth maps generation. Most approaches make use of several viewpoints to compute disparity as it offers a reliable cue for depth estimation [2]. However, disparity estimation assumes that two images captured at the same time instant are available but, in many situations, this assumption cannot be fulfilled. For example, most commercial cameras have only one photographic lens and only record monocular sequences. Moreover, one critical issue is the large amount of material which has already been acquired in the past as monocular sequences and which needs to be converted to some extent to a three-dimensional (3D) format. In such cases, depth information can only be inferred through monocular cues. The film industry is seriously tackling this problem. For example, Disney or Microsoft have designed supervised systems supporting the creation of depth maps for monocular sequences [3, 4]. These systems rely heavily on human interaction. However, there is a clear interest in defining unsupervised systems because of their reduced cost in time and money resources [5–7].

Depth order maps can be seen as an intermediate state between 2D images where no depth information is defined and full 3D maps. The depth order map specifies an image partition where regions are ordered by their relative depth. State-of-the-art depth ordering systems include [5, 7] in which a layered representation of a sequence is obtained by finding occlusions between a pair of regions. However, the final depth order is obtained by a simple aggregation of local cues with no global reasoning. As a result, the final map is not globally consistent. In [8], a global depth order is obtained through the estimation of 3D movements. The approach processes pixels individually and lacks the concept of regions. Therefore the resulting partitions involve many small regions and the decision process is not robust. Karsch *et al.* [6] attempt to find a full depth map by matching parts of the input video to similar videos and then by propagating depth information to unmatched regions. This approach works well for known scenes but its generalisation to arbitrary scenes is very difficult. There is an attempt in [9, 10] to retrieve a full depth map from a monocular image sequence. However, they involve important assumptions and restrictions about the scene structure which may not be fulfilled in many typical situations.

Other state-of-the-art systems do not try to create a depth partition but focus on the estimation of the depth order around contours. In this context, the contours may not be closed and therefore do not specify regions. For example, assuming that the scene is still and the occlusions are because of disparity, the contours are detected in [11]. Interesting detection results are shown but, if relative depth is needed, another approach should be followed. Sundberg *et al.* [12] defines a figure/ground (f/g) labelling on occlusion contours by computing the motion boundaries and assigning the closer (figure) side to the region that moves similar to the contour. The main drawback of this scheme is that the relative depth is assigned based on a set of local characteristics of the contour and avoids global

reasoning on the depth structure of the scene. However, the f/g labels are attractive because they offer a good way to compare systems by taking into account the correct number of labelled contour pixels and an objective evaluation methodology is defined in [12].

The system proposed here addresses the main problems of the state-of-the-art solutions dealing with depth order estimation in video sequences. The three main challenges we address are: to allow moving objects to be present in the scene, to provide a complete depth order partition in contrast to defining depth order only on occlusion boundaries and to ensure that this partition is globally coherent.

Our approach is to use motion occlusion to determine the depth order within a frame given its previous and next frames. We make no assumptions on scene stillness. Therefore the main cue we can rely on is motion occlusion. When the objects move relative to the camera, background areas may appear and disappear, providing a reliable cue to determine the depth order. Note that motion occlusion appears when the apparent motion of two overlapping objects/regions is different. This situation occurs either when:

- The real motion of the two objects is different (e.g. two cars in a road).
- The scene is static and the object depths are different (e.g. a building occluding the sky).

To exploit this idea, the system first computes the forward and backward optical flows ('optical flow estimation' block of Fig. 1). Then, a hierarchical region-based representation of the image is computed and stored in a binary partition tree, BPT ('tree construction' block). The goal of this representation is to support robust estimation and global reasoning about relative depth. The use of such representation is essential in our approach. In this paper, we will use and compare two ways to construct this representation: one based on colour, shape and motion features (CSM) [13] and one based on ultrametric contour map (UCM) [14]. The created BPT is used to retrieve two partitions using specific graph cut techniques called 'pruning'. The first partition allows us to fit parametric flow models to regions, finding reliable flow values at occlusion points ('parametric flow fitting pruning' block) and then obtaining occlusion relations. The second partition is obtained by exploiting these occlusion relations and defines regions which can be depth ordered ('depth ordering pruning' block). Since occlusion relations provide depth relations between a pair of regions, a final step is needed to ensure global consistency and to obtain a final depth order map. Besides the algorithm definition, this work's contributions concern the formalisation of the energy minimisation originally presented in [15] as an efficient way to retrieve partitions from BPTs and the study of motion occlusions as a reliable cue for depth ordering on video frames, showing that dynamic cues perform better than static ones [16].

The paper is organised as follows: Section 2 defines the optical flows used in the system. An overview on the hierarchical segmentation tools used is explained in Section 3 whereas Section 4 discusses the specific graph cut technique, called pruning, used to extract an optimal partition from the trees. The motion occlusion estimation is presented in Section 5. Finally, the definition of the partition involving the regions to be ordered and the global reasoning leading to a complete depth order map are detailed in Section 6. The evaluation of the prosed scheme is performed in Section 7. Finally, Section 8 concludes the paper.

2 Optical flows

To determine the depth order of a frame I_t, the previous and following frames I_{t-1}, I_{t+1} are used. The forward flows $\boldsymbol{w}^{t-1,\ t}$, $\boldsymbol{w}^{t,\ t+1}$ and backward flows $\boldsymbol{w}^{t,\ t-1}$, $\boldsymbol{w}^{t+1,\ t}$ (see Fig. 2) are estimated using the technique presented in [17]. This is a classical motion estimation algorithm which provides good results with a reduced computational load. The optical flow $\boldsymbol{w}^{t_a,\ t_b}$ maps each pixel of I_{t_a} to one pixel in I_{t_b}. The flows $\boldsymbol{w}^{t,\ t+1}$ and $\boldsymbol{w}^{t,\ t-1}$ are used with colour information to create the BPT (Section 3). The two remaining flows are also used to estimate the occlusions (Section 5). Let us discuss now the construction of the BPT.

3 BPT to represent hierarchical segmentations

The algorithm developed in this paper relies on a hierarchical region-based representation of the image using a binary tree structure. A classical way to build such a BPT is using a bottom-up region merging technique. A region tree structure is attractive as it allows a more global and robust image interpretation to be performed compared with the original pixel-based representation. Moreover, the

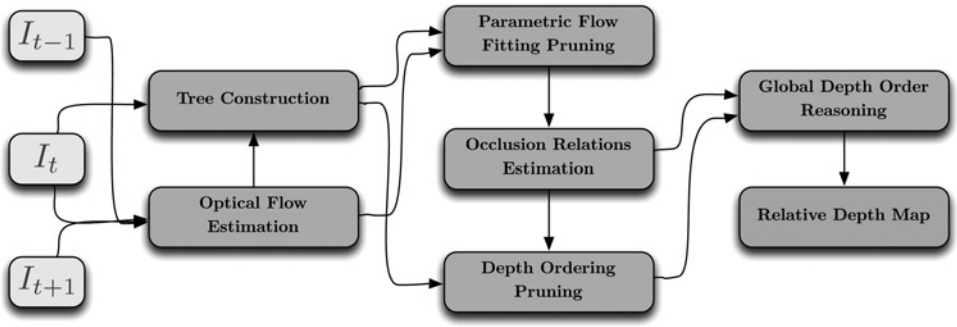


Fig. 1 *Proposed system: three consecutive frames are used to estimate a depth order map*

System involves an optical flow estimation step and a tree construction
Then, two pruning (graph cut) strategies are applied to extract one partition providing a region-based representation of the optical flow and a second partition involving regions which can be depth ordered
Finally, a global reasoning is used to define a consistent depth order map

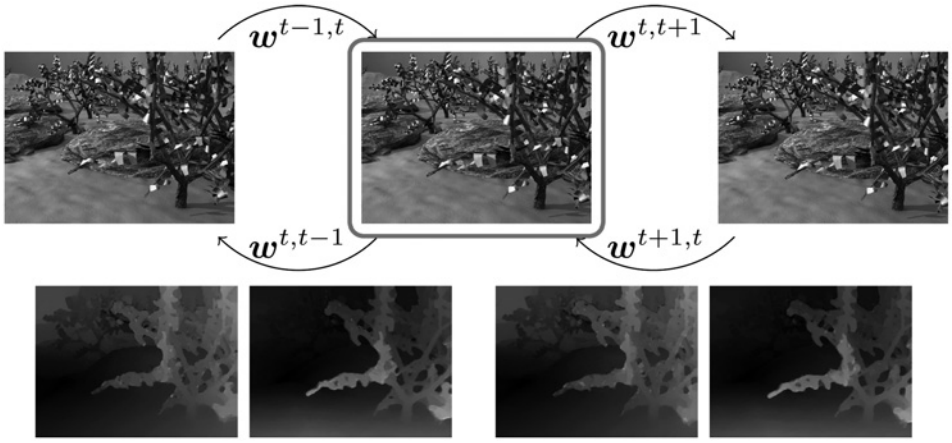


Fig. 2 *Top row: three consecutive frames I_{t-1}, I_t (outlined) and I_{t+1}*
Bottom row: from left to right, $w^{t-1,\,t}$, $w^{t,\,t-1}$, $w^{t,\,t+1}$ and $w^{t+1,\,t}$ flows

representation is multi-scale and small details as well as very large areas are described by the tree. Note that arbitrary trees could be used, but we restrict ourselves to the binary case for two reasons: (i) binary trees allow a fine control of the image under/over segmentation and (ii) pruning algorithms on these kinds of trees are easier to define and to handle than arbitrary trees.

The BPT construction begins with an initial partition of the image and iteratively merges a pair of neighbouring regions until only one region is left. The merging order is defined by a region distance describing the similarity between two regions. In the case of static images, the distance is usually a combination of similarity measures relying on simple characteristics such as colour, area, shape or contour strength. In the video case, using motion flows, it is also possible to differentiate between regions of similar colour which move in different directions. In any case, the resulting BPT is composed of nodes representing image regions and edges describing the inclusion relationship between regions, whereas the leaves represent regions belonging to the initial partition. Any oversegmentation can be used as initial partition. In the following, we will assume that the initial partition is made of regions involving only one pixel.

For the purposes of this work, two possible trees have been considered: the $\mathrm{BPT}_{\mathrm{CSM}}$ created using [18] and the $\mathrm{BPT}_{\mathrm{UCM}}$ using the UCM of [14]. The only difference in the construction of the two BPTs is region distance $d(R_i, R_j)$. The $\mathrm{BPT}_{\mathrm{CSM}}$ uses a combination of colour, shape and motion information, whereas the $\mathrm{BPT}_{\mathrm{UCM}}$ considers the mean strength of the common contour between R_i and R_j. The formal expressions are

$$d_{\mathrm{BPT}_{\mathrm{CSM}}}\left(R_i,\ R_j\right) = d_a\left(\alpha d_{\mathrm{cm}} + (1-\alpha)d_s\right) \tag{1}$$

$$d_{\mathrm{BPT}_{\mathrm{UCM}}}\left(R_i,\ R_j\right) = \sum_{x\in\Gamma_{ij}} \frac{gPb(x)}{|\Gamma_{ij}|} \tag{2}$$

For the $\mathrm{BPT}_{\mathrm{CSM}}$ distance, d_s is the shape distance defined as in [19], d_a is a logarithmic weight of the area as defined in [16]. d_{cm} stands for a distance measuring the region similarity in terms of colour and motion. Essentially, each region is represented by a limited number of dominant colours and motion vectors and the Earth Mover's distance is used to compare the descriptions of two regions. See [13] for more details. For the $\mathrm{BPT}_{\mathrm{UCM}}$ distance, Γ_{ij} is the common region contour and *gPb* is the contour detector of [14]. Once the BPT has been constructed, it can be used to retrieve many different partitions. The next section discusses this point.

4 Optimum tree pruning

Independent of the distance used to create the tree, the technique extracting a partition from it can be viewed as a 'pruning' [15, 19, 20]. The BPT is a particular graph where each node represents a region and the tree branches the region inclusion relationship. A partition can be naturally defined from a BPT by selecting the regions represented by the tree leaves. If this is done on the original tree, the initial partition where each region involves only one pixel is extracted. However, if we prune the tree, that is, if we cut branches at one location to reduce their length, a new tree, called a 'pruned BPT' is created. The leaves of the pruned BPT define a non trivial partition. This pruning is a particular graph cut: if the tree root is the 'source' of the graph and the leaves are connected to a 'sink' node, the pruning cuts the tree in two connected components, one including the source and the other the sink. Note that following this approach, partitions observed during the merging sequence can obviously be obtained but the interest of the pruning is that a much richer set of partitions can be extracted. Of course, the key point is to define an appropriate pruning rule. Here, an optimum pruning based on energy minimisation is proposed.

A partition P extracted by pruning can be represented by a 'partition vector' $\boldsymbol{x}$ of binary variables $x_i = \{0, 1\}$ with $i = 1, \ldots, N$ assigned to each BPT region R_i. If $x_i = 1$, R_i belongs to the partition, otherwise $x_i = 0$. Only a reduced subset of vectors, called 'valid' vectors, actually represents a partition extracted by pruning. A vector $\boldsymbol{x}$ is valid if one and only one region in every BPT branch involves only one $x_i = 1$. A branch is a sequence of regions from a leaf to the root of the tree. For example, the tree of Fig. 3 involves four branches.

Each branch l can be represented by a 'branch vector' $\boldsymbol{b}_l = \left(b_1^l, \ \ldots, \ b_N^l\right)^\top$ where $b_i^l = 1$ if region R_i is in the branch and $b_i = 0$ otherwise. In the example of Fig. 3, the

doi: 10.1049/iet-cvi.2012.0287

Set of valid $\boldsymbol{x} = (x_1, \ldots, x_7)^T$:

$\boldsymbol{x}_1 = (1,1,1,1,0,0,0)^T$
$\boldsymbol{x}_2 = (0,0,1,1,1,0,0)^T$
$x_3 = (1,1,0,0,0,1,0)^T$
$\boldsymbol{x}_4 = (0,0,0,0,1,1,0)^T$
$\boldsymbol{x}_5 = (0,0,0,0,0,0,1)^T$

Invalid:

$\boldsymbol{x}_I = (1,1,0,0,1,0,0)^T$

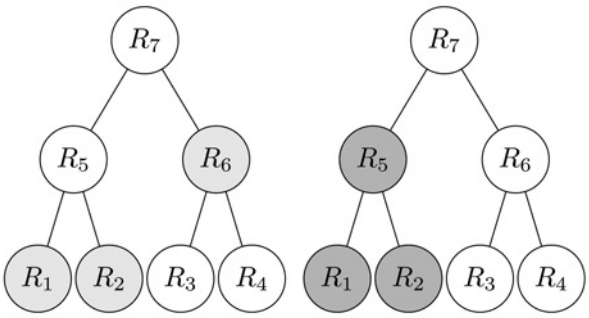


Fig. 3 *Left: set of valid partition vectors representing a pruning and an invalid partition vector*
Centre: BPT with light grey nodes indicating the cut described by $\boldsymbol{x}_3$
Right: BPT with grey nodes representing the regions described by $\boldsymbol{x}_I$ which does not define a pruning

four branch vectors are: $\boldsymbol{b}_1 = (1, 0, 0, 0, 1, 0, 1)^\top$, $\boldsymbol{b}_2 = (0, 1, 0, 0, 1, 0, 1)^\top$, $\boldsymbol{b}_3 = (0, 0, 1, 0, 0, 1, 1)^\top$ and $\boldsymbol{b}_4 = (0, 0, 0, 1, 0, 1, 1)^\top$. With this notation, a partition vector $\boldsymbol{x}$ is valid if, for every branch l, $\boldsymbol{b}_l^\top \boldsymbol{x} = 1$. In Fig. 3, $\boldsymbol{x}_I = (1, 1, 0, 0, 1, 0, 0)^\mathrm{T}$ is not valid because $\boldsymbol{b}_l^\top \boldsymbol{x}_I = 2$. The constraint can be globally expressed as a matrix product $\boldsymbol{A}\boldsymbol{x}$. In the case of Fig. 3, the constraint is

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{b}_1^\top \\ \boldsymbol{b}_4^\top \\ \boldsymbol{b}_3^\top \\ \boldsymbol{b}_4^\top \end{pmatrix} \boldsymbol{x} = \mathbf{1} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \boldsymbol{x} = \mathbf{1} \quad (3)$$

where $\mathbf{1}$ is a vector containing all ones.

An efficient way to extract a partition from the BPT is to find the one that minimises an energy function of the type

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} E(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}} \sum_{R_i \in \mathrm{BPT}} E_r(R_i) x_i \quad (4)$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \mathbf{1} \quad x_i = \{0,\ 1\} \quad (5)$$

where $E_r(R_i)$ is a function which only depends of the internal characteristics of R_i. In that case, the optimum partition $\boldsymbol{x}^*$ can be efficiently found by the dynamic programming Algorithm 1 (see Fig. 4). The algorithm benefits from the fact that the energy $E_r(R_i)$ does not dependent on regions $R_{j \neq i}$ and that the global energy is the sum of the energy values assessed on each region. Therefore locally optimum decisions lead to a global optimum. More precisely, if R_i is a region which has two child regions R_l and R_r, the local decision which has to be taken is to know whether R_i or $R_l \cup R_r$ has to belong to the partition as both solutions cover the same image area. If $E_r(R_i)$ is smaller (larger) than $E_r(R_l) + E_r(R_r)$, the locally optimum solution selects $R_i(R_l \cup R_r)$. The complete tree is analysed in a bottom-up fashion (from the leaves to the root) to define the complete partition as outlined in Algorithm 1 (see Fig. 4). This algorithm is going to be used twice in the proposed system. Once for the identification of occluded and disoccluded areas and once for the extraction of regions to be depth ordered. The first issue is discussed in the next section.

Algorithm 1

function OPTIMALSUBTREE(Region R_i)
  $R_l, R_r \leftarrow$ (LEFTCHILD(R_i),RIGHTCHILD(R_i))
  $(\boldsymbol{o}_i, c_i) \leftarrow (R_i, E_r(R_i))$
  $(\boldsymbol{o}_l, c_l) \leftarrow$ OPTIMALSUBTREE(R_l)
  $(\boldsymbol{o}_r, c_r) \leftarrow$ OPTIMALSUBTREE(R_r)
  if $c_i < c_r + c_l$ **then**
    OPTIMALSUBTREE(R_i) $\leftarrow (\boldsymbol{o}_i, c_i)$
  else
    OPTIMALSUBTREE(R_i) $\leftarrow (\boldsymbol{o}_l \bigcup \boldsymbol{o}_r, c_l + c_r)$
  end if
end function

Fig. 4 ***Algorithm 1*** *Optimal partition selection: OPTIMALSUBTREE (region R_i) contains the set of regions belonging to the subtree rooted at R_i which have been selected to be part of the partition and the sum of their associated energy*

5 Estimation of occlusion relations

5.1 Occluded and disoccluded areas

As discussed in the introduction, motion occlusion is used here as the basis of depth order estimation. Using three frames I_{t-1}, I_t, I_{t+1}, it is possible to detect pixels becoming invisible from I_t to I_{t+1}, called 'occluded pixels' and pixels becoming invisible from I_t to I_{t-1} called 'disoccluded pixels'. Here, we describe the detection of occluded pixels as the detection of disoccluded pixels can be performed similarly by working on the past frame I_{t-1} instead of the next frame I_{t+1}.

When there is no occlusion, the optical flow $\boldsymbol{w}^{t,\,t+1}$ creates locally a bijection between I_t and I_{t+1}. However, in case of occlusion, two different pixels from I_t, $\boldsymbol{p}_a$ and $\boldsymbol{p}_a$, are projected at the same location $\boldsymbol{p}_m$ in I_{t+1}. This situation is illustrated in the left part of Fig. 5. Therefore an occlusion is detected if

$$\begin{aligned} &\boldsymbol{p}_a + \boldsymbol{w}^{t,\,t+1}(\boldsymbol{p}_a) = \boldsymbol{p}_b + \boldsymbol{w}^{t,\,t+1}(\boldsymbol{p}_b) \\ &= \boldsymbol{p}_m \text{ assuming that } \boldsymbol{p}_a \neq \boldsymbol{p}_b \end{aligned} \quad (6)$$

This equation explains that either $\boldsymbol{p}_a$ or $\boldsymbol{p}_b$ is an occluded pixel. To decide which one is the actual occluded pixel, we rely on a comparison of patches centered around $\boldsymbol{p}_a$, $\boldsymbol{p}_b$ and $\boldsymbol{p}_m$. Indeed, it is likely that the patch around the non-occluded pixel ($\boldsymbol{p}_b$ in Fig. 5) is very similar to the

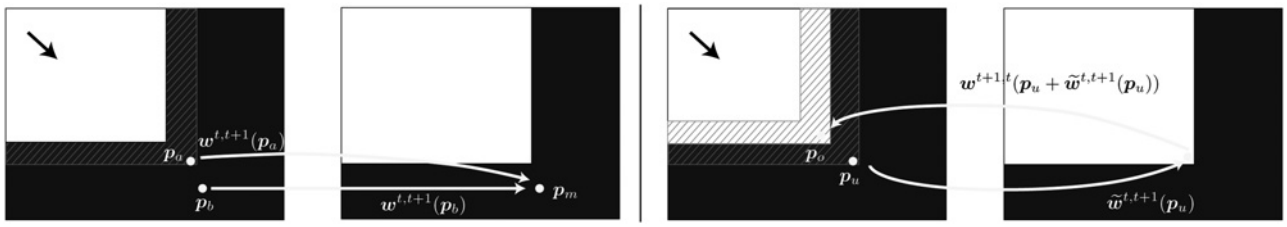


Fig. 5 *Left: detection of occluded pixels (black area)*
Right: detection of occluding pixels (white area)
In both cases, the image on the left (right) is I_t (I_{t+1})

patch centred around its projected point ($\boldsymbol{p}_m$). Therefore the decision is based on the distance between patches

$$D(\boldsymbol{p}_x,\ \boldsymbol{p}_m) = \sum_{\boldsymbol{d}\in\Gamma} \left(I_{t+1}(\boldsymbol{p}_m + \boldsymbol{d}) - I_t(\boldsymbol{p}_x + \boldsymbol{d})\right)^2 \quad (7)$$

with $\boldsymbol{p}_x = \boldsymbol{p}_a$ or $\boldsymbol{p}_b$ and Γ is a 5×5 square window. The pixel with highest $D(p_x, p_m)$ is declared to be the occluded pixel. Following this strategy, all pixels belonging to the black area of Fig. 5 are defined as occluded pixels. An example on a real image can be seen in the central image of Fig. 6 where occluded pixels are shown.

Once the occluded pixels have been defined, we need to find the 'occluding pixels', that are the pixels which will cover the occluded pixels in the next frame. Indeed, it is the relation between occluded and occluding pixels that provides a depth cue. However, as can be seen in Fig. 5, the optical flow associated with occluded pixels (P_a) is particularly unreliable. To deal with this issue, the previously created BPT is used to define an optical flow partition P_f where a parametric motion model is assigned to each region allowing us to obtain a reliable flow for occluded pixels. Of course, a similar detection has to be performed for disoccluded and disoccluding pixels.

5.2 Pruning for parametric flow fitting

To obtain a region-based modelling of the optical flow, a parametric projective model [21] is used. The flows $\tilde{\boldsymbol{w}}_{R_i}^{t,\,q} = (\tilde{u},\ \tilde{v})$ with $q = t \pm 1$, associated with region R_i can be expressed as a quadratic model on the x and y coordinates

$$\tilde{u}(x, y) = a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \quad (8)$$

$$\tilde{v}(x, y) = a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \quad (9)$$

where $(x, y) \in R$. The $a_1, \ldots, a_8$ parameters are estimated with robust regression using iterative least squares [22]

$$\tilde{\boldsymbol{w}}_{R_i}^{t,\,q}(\boldsymbol{p}) = \arg\min_{\tilde{\boldsymbol{w}}^{t,q}} \sum_{\boldsymbol{p}=(x,\,y)\in R_i} \Psi\left(\left\|\boldsymbol{w}^{t,\,q}(\boldsymbol{p}) - \tilde{\boldsymbol{w}}^{t,\,q}(\boldsymbol{p})\right\|^2\right) \quad (10)$$

with the robust penaliser $\Psi(z) = \sqrt{z^2 + \epsilon^2}$ with $\epsilon \ll 1$. An example of flow fitting can be seen in the right part of Fig. 6. To limit the computational load, this flow fitting is applied on the tree nodes that are close to the tree root. Typically, the nodes corresponding to the last thousand merging steps are kept and the remaining nodes corresponding to earlier merging steps are discarded. Once the parametric flow is estimated, a partition P_f representing the regions that best fit to these models is computed using the optimal pruning Algorithm 1 (see Fig. 4) with the following energy $E_r(R_i)$

$$E_r(R_i) = \sum_{q=t\pm 1} \sum_{x,\,y\in R_i} \left|\boldsymbol{w}^{t,\,q}(x,\ y) - \tilde{\boldsymbol{w}}_{R_i}^{t,\,q}(x,\ y)\right| + \lambda_f \quad (11)$$

The constant $\lambda_f = 4 \times 10^3$ is used to prevent oversegmentation. It was found experimentally and proved not to be crucial for overall system performance.

5.3 Occlusion relation estimation

Once the partition P_f has been defined and a parametric optical flow model is available for each region, the occluding pixels can be defined by projecting the occluded pixels in I_{t+1} with $\tilde{\boldsymbol{w}}^{t,\,t+1}$ and by going back to the current frame following the backward flow $\boldsymbol{w}^{t+1,t}$. This is illustrated in the right part of Fig. 5 where occluding pixels appear in the white area. Hence, for each occluded pixel $\boldsymbol{p}_u$, the

Fig. 6 *Example occlusion estimation for two regions*
From left to right: original frame with the region contours in white
Frame with occluded and occluding pixels
Forward (top) and backward (bottom) estimated and modelled flows

doi: 10.1049/iet-cvi.2012.0287

corresponding occluding pixel $\boldsymbol{p}_o$ is given by

$$\boldsymbol{p}_o = \boldsymbol{p}_u + \tilde{\boldsymbol{w}}_{R_i}^{t,\,t+1}(\boldsymbol{p}_u) + \boldsymbol{w}^{t+1,\,t}\left(\boldsymbol{p}_u + \tilde{\boldsymbol{w}}_{R_i}^{t,\,t+1}(\boldsymbol{p}_u)\right) \quad (12)$$

The central image of Fig. 6 also shows these occluding pixels. At this point, we know that the occluding pixels are in front of the occluded pixels and similarly, the disoccluding pixels are in front of the disoccluded pixels. This information is used in the second BPT pruning described in the following section.

6 Depth order map definition

6.1 Depth ordering pruning

Equation (12) creates a set of pixel pairs ($\boldsymbol{p}_u$, $\boldsymbol{p}_o$) for which depth information is available. If both pixels belong to the same region, they are discarded but if they belong to two different regions, we can conclude that there is one evidence that the two regions belong to different depth planes. In the context of regions described by the BPT, if we deal with regions that are close to the root, many ($\boldsymbol{p}_u$, $\boldsymbol{p}_o$) pairs are discarded because the regions are very large. By contrast, if the regions are close to the leaves, many ($\boldsymbol{p}_u$, $\boldsymbol{p}_o$) pairs will be preserved.

To extract from the BPT a partition P_d involving regions which can be depth ordered, an optimal pruning is used. Here, the energy to be optimised should be a compromise between the number of occlusion relations, that is of ($\boldsymbol{p}_u$, $\boldsymbol{p}_o$) pairs, that are kept and the simplicity of the partition in terms of region number. As a result, the pruning is performed with Algorithm 1 (see Fig. 4) with the following energy

$$E_r(R_i) = \sum_{(\boldsymbol{p}_u, \boldsymbol{p}_o) \in R_i} \frac{1}{N_o} + \lambda_o \quad (13)$$

where N_o is the total number of estimated occlusion relations. To avoid oversegmented solutions, $\lambda_o = 4 \times 10^{-3}$ is used (see Section 7).

6.2 Final depth ordering

Once the final partition P_d is obtained through BPT pruning, a global ordering can be computed. The problem could be viewed as a rank aggregation problem which is used for web ranking [23] or photosequencing [24]. Here, the goal is to achieve a fully ordered list from a set of partial orders by minimising a given cost function. Normally, rank aggregation works with fully ordered lists, where two elements cannot have the same order. Since, in an image, two different regions may be at the same depth (thus have the same order), we state the problem as a network reliability problem [25].

A graph $G = (V, E)$ is constructed where the vertices V represent the regions of P_d. A directed edge $e_i = (a, b, p_i)$ is defined between node a and node b if there are occlusion relations between region R_a and region R_b. The weight of e_i is $p_i = N_{ab}/N_o$ where N_{ab} is the number of pixels from R_a which have been estimated as occluding pixels of R_b and N_o is the total number of occluding pixels. The graph G can be seen as a network of (un)reliable links, with the edge $e_i = (a, b, p_i)$ connecting a and b with probability p_i. In this context, a precedes b in depth (a is in front of b) with probability p_i. For two arbitrary nodes of G, the probability of precedence (PoP) can be computed even if there are no edges directly connecting them. If there exists more than one path from a node a to b, the probability of 'a to precede b' is called ρ_{ab} and is the probability that at least one path between a and b is reliable. ρ_{ab} can be computed by complete state enumeration and the inclusion–exclusion principle [25].

To define a globally consistent depth order between regions, G should be acyclic. To break cycles in G (if any), the algorithm iteratively eliminates the edge of minimum PoP. Once all cycles have been removed in G, a topological partial sort [26] is applied and each region is assigned a depth order. Regions which have no depth relation, are assigned the depth of their most similar adjacent region according to the distance in the BPT construction. The complete process is illustrated in Fig. 7 with a simple example.

7 Results

System evaluation is performed on keyframes of several classical sequences. In order to obtain an objective evaluation, we propose two classes of experiments:

Figure/ground assignment: We assess the foreground/background (f/g) label assignment on contours as discussed in [11, 12]. In our context, the assignment is performed as

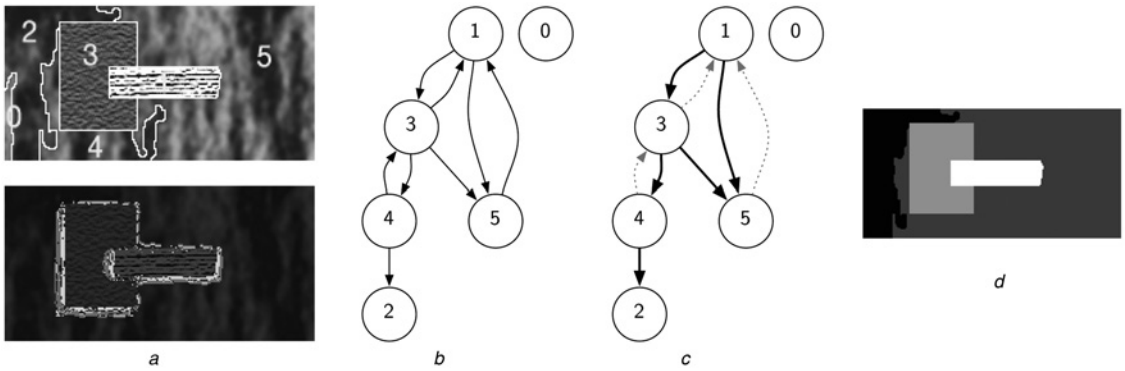


Fig. 7 *Depth ordering example*

a Depth order partition: top: region number and contours, bottom: estimated occluded points and occluding points
b Initial G graph with all occlusion relations
c Final graph where cycles have been removed. Removed edges are dashed
d Depth order map (brighter regions are closer to the viewer)

follows: when two depth planes meet, the part of the contour belonging to the closest region is assigned the foreground label and the other side of the contour is assigned the background label, see Fig. 8. It is important to note that the proposed system defines the depth information on a region basis, whereas the f/g algorithm of [12] only labels contour points. These contours are not necessarily closed and therefore no regions are defined and these f/g algorithms do not allow the creation of a complete depth order map. Nevertheless, the existence of a ground truth f/g database makes the comparison with these systems attractive. The used datasets are the Carnegie Mellon Dataset (CMU) [27] and the Berkeley Dataset (BDS) [12]. We follow the same evaluation procedure as [12] which essentially is the precision of f/g labels on matched contour pixels against a ground truth database containing depth order partitions.

Segmentation evaluation: Equation (13) establishes a region energy depending on a factor λ_o which has a direct effect on the granularity of the extracted partition P_d. For large values of λ_o, only prominent occlusion relations will be kept and thus only a few regions are conserved. On the contrary, for small values λ_o, the generated P_d also preserves low-confident occlusion relations, generating a partition with more regions. If P_d only includes regions corresponding to highly confident occlusion relations, it is expected to obtain a high f/g precision rate, at the expense of a low boundary recall (BR) on groundtruth segmentations. If P_d is formed by regions corresponding to low-confident occlusion relations, the BR is expected to improve, although the f/g precision can decrease. To this end, jointly with the f/g precision, we present the BR of the given algorithm.

Table 1 shows the performances of f/g assignment and BR on the CMU and BDS datasets for the BPT_{CSM} and BPT_{CMU} trees and compares it with the techniques proposed in [12, 16]. For [12], we only report the paper published f/g results as it was impossible to reproduce the complete algorithm. Therefore the BR for this technique is not available. For the remaining techniques, we have used λ_o values providing a similar BR to make a fair evaluation of the f/g results. From

Table 1 Percentage of correct f/g assignments and BR on the CMU and BDS datasets

Dataset	CMU [27]		BDS [12]	
	f/g, %	BR	f/g, %	BR
figure/ground from optical flow [12]	83.8	–	68.6	–
still image depth ordering [16]	63.4	0.5	63.6	0.4
depth ordering with BPT	88.0	0.48	80.9	0.37
depth ordering with UCM	67.9	0.48	68.9	0.37

all the presented techniques, the BPT_{CSM} is the one with the best performance on f/g assignment, outperforming [12] on both datasets. The one performing worst is [16] mainly because it does not use motion cues at all and its depth ordering is based only on monocular static cues (T-junctions and convexity). BPT_{UCM} has lower performances than BPT_{CSM}. Although UCMs have excellent performances in terms of defining distances between regions in static images, they do not involve motion features. The effects of introducing motion information in BPT construction can be seen in Fig. 9 on images with various objects of very similar colours. In cases where the colour information is ambiguous, motion successfully helps to identify regions moving coherently.

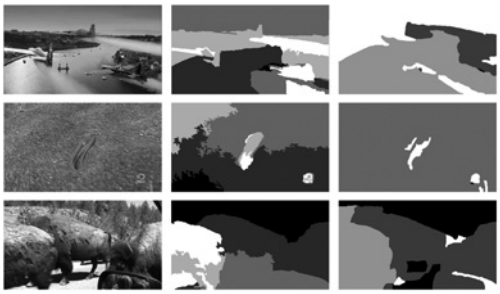
As stated in [28], prominent contours are easy to detect and to assign the correct depth gradient, whereas ambiguous contours are much more difficult to deal with. Therefore it is expected that, as the BR increases, the f/g assignment loses performance. This can be seen in the table reported in Fig. 9. These results are obtained on the BDS dataset, which we found more challenging than the CMU dataset, by varying λ_o on the BPT_{CSM} and BPT_{UCM} techniques. Depending on the application, one can set a high λ_o and let the system behave like a foreground/background segregation system with high f/g performance. If more complex scenes have to be processed, a low λ_o retrieves multiple regions although f/g assignment is less precise.

A subjective evaluation of the ability of the proposed system to create a depth order map can be seen in Figs. 8

Fig. 8 *Results of the CMU dataset and f/g assignment with the BPT_{CSM} system*

From left to right (on both columns)
(1) Processed keyframe
(2) Occlusion relations
(3) Estimated depth partition. White regions are closer and black regions are further
(4) f/g assignment on contours

doi: 10.1049/iet-cvi.2012.0287



BDS Dataset				
	BPT_{CMS}		BPT_{UCM}	
λ_o	f/g	BR	f/g	BR
$1 \cdot 10^{-2}$	92.5	0.33	73.9	0.14
$7 \cdot 10^{-3}$	80.9	0.37	71.1	0.2
$9 \cdot 10^{-4}$	74.2	0.49	68.9	0.37

Fig. 9 *Left: comparison of BPT_{CSM} and BPT_{UCM} for images with objects of similar colour: original frames shown on the left, the centre column shows the results of BPT_{CSM} and the right column shows the results of BPT_{UCM}*

Rightmost table shows the f/g assignment and the BR varying the parameter λ_o

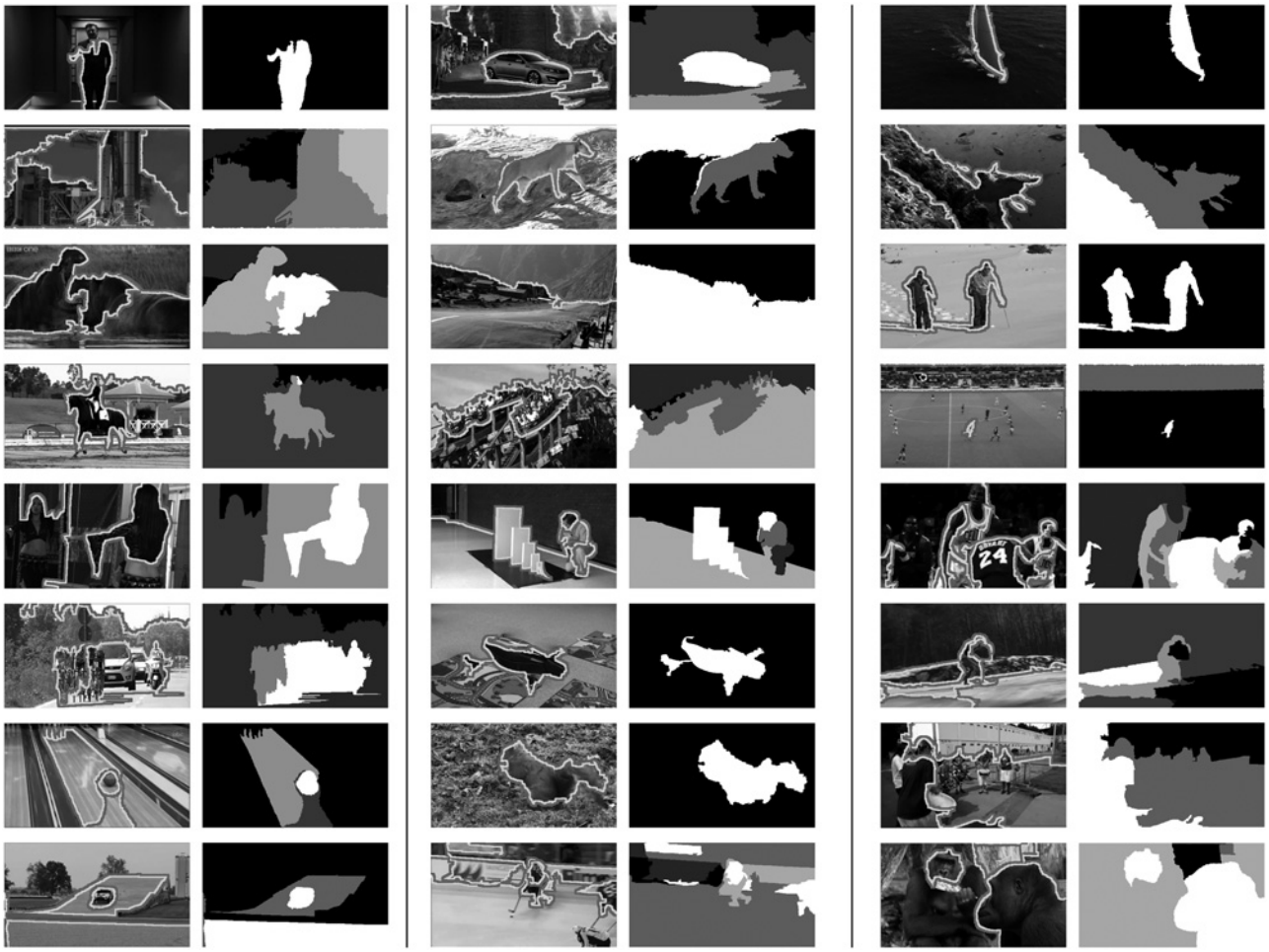


Fig. 10 *Results on a subset of the BDS dataset with the BPT_{CSM} system*

For each column, the right image corresponds to the analysed frame with f/g assignment overlaid on contours
Left image corresponds to the final depth order partition

and 10 showing that motion occlusions may work over a variety of situations: static scenes, moving foregrounds, moving backgrounds or even multiple moving objects.

8 Conclusions

In this work, a system inferring the relative depth order of the regions of a frame has been described. Combining a variational approach for optical flow estimation and a hierarchical region-based representation of the image, we have developed a reliable system to detect occlusion relations and to create depth order partitions using only motion occlusion. The system also allows us to deal with the classical foreground/background contour labelling problem (f/g). In this context, comparison with the state-of-the-art shows that motion occlusions are very reliable cues. The presented approach, although using only motion information to detect boundaries, achieves better results on f/g assignment than the state-of-the-art technique [12].

Many extensions of the system are possible. First, a longer temporal window could be used to retrieve more precisely motion occlusions. Secondly, we can take advantage of other monocular depth cues, such as T-junctions and convexity to help in case of motionless depth relations. Although Table 1 shows that they are less reliable than motion occlusions, they could be useful when motion occlusions are not present (i.e. a static background), as in some cases in Fig. 10. We believe also that motion occlusions can be propagated throughout the sequence to infer a consistent depth order. Sequence depth ordering seems plausible because results on individual frames are promising.

9 References

1 Ono, M.E., Rivest, J., Ono, H.: 'Depth perception as a function of motion parallax and absolute-distance information', *J. Exp. Psychol., Hum. Percept. Perform.*, 1986, **12**, pp. 331–337
2 Qian, N., Qian, D.N.: 'Binocular disparity and the perception of depth', *Neuron*, 1997, **18**, pp. 359–368
3 Ward, B., Bing Kang, S., Bennett, E.P.: 'Depth director: a system for adding depth to movies', *IEEE Comput. Graph. Appl.*, 2011, **31**, (1), pp. 36–48
4 Wang, O., Lang, M., Frei, M., Hornung, A., Smolic, A., Gross, M.: 'StereoBrush: interactive 2D to 3D conversion using discontinuous warps'. Proc. Eighth Eurographics Symp. on Sketch-Based Interfaces and Modeling (SBIM'11), New York, NY, USA, 2011, pp. 47–54
5 Bergen, L., Meyer, F.: 'A novel approach to depth ordering in monocular image sequences'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2000, vol. 2, pp. 536–541
6 Karsch, K., Liu, C., Kang, S.B.: 'Depth extraction from video using nonparametric sampling'. ECCV, 2012
7 Turetken, E., Alatan, A.A.: 'Temporally consistent layer depth ordering via pixel voting for pseudo 3D representation'. 3DTV Conf., 2009, pp. 1–4
8 Chang, J.-Y., Cheng, C.-C., Chien, S.-Y., Chen, L.-G.: 'Relative depth layer extraction for monoscopic video by use of multidimensional filter'. Proc. IEEE Int Multimedia and Expo Conf., 2006, pp. 221–224
9 Li, P., Farin, D., Gunnewiek, R.K., de With, P.H.N.: 'On creating depth maps from monoscopic video using structure from motion'. Proc. 27th Symp. on Information Theory in the Benelux, 2006, pp. 508–515
10 Zhang, G., Jia, J., Wong, T.-T., Bao, H.: 'Consistent depth maps recovery from a video sequence', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (6), pp. 974–988
11 He, X., Yuille, A.: 'Occlusion boundary detection using pseudo-depth'. ECCV, 2010 (*LNCS*, 6314), pp. 539–552
12 Sundberg, P., Brox, T., Maire, M., Arbelaez, P., Malik, J.: 'Occlusion boundary detection and figure/ground assignment from optical flow'. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 2011, pp. 2233–2240
13 Palou, G., Salembier, P.: 'Depth ordering on image sequences using motion occlusions'. Proc. 19th IEEE Int. Conf. Image Processing, Florida, USA, September 2012, pp. 1217–1220
14 Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: 'Contour detection and hierarchical image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, **33**, (5), pp. 898–916
15 Salembier, P., Garrido, L.: 'Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval', *IEEE Trans. Image Process.*, 2000, **9**, (4), pp. 561–576
16 Palou, G., Salembier, P.: 'Monocular depth ordering using T-junctions and convexity occlusion cues', *IEEE Trans. Image Process.*, 2013, **22**, (5), pp. 1926–1939
17 Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: 'High accuracy optical flow estimation based on a theory for warping'. European Conf. Computer Vision, Prague, Czech Republic, May 2004, vol. 3024, pp. 25–36
18 Palou, G., Salembier, P.: '2.1 depth estimation of frames in image sequences using motion occlusions', in Fusiello, A., Murino, V., Cucchiara, R. (Eds.): 'ECCV Workshops' (Springer, 2012) (*LNCS*, 7585), pp. 516–525
19 Vilaplana, V., Marques, F., Salembier, P.: 'Binary partition trees for object detection', *IEEE Trans. Image Process.*, 2008, **17**, (11), pp. 2201–2216
20 Calderero, F., Marques, F.: 'Region merging techniques using information theory statistical measures', *IEEE Trans. Image Process.*, 2010, **19**, (6), pp. 1567–1586
21 Kanatani, K.: 'Transformation of optical flow by camera rotation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1988, **10**, (2), pp. 131–143
22 Andersen, R.: 'Modern methods for robust regression. Number 152 in quantitative applications in the social sciences' (Sage Publications, 2008)
23 Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: 'Rank aggregation methods for the web'. Proc. 10th Int. Conf. World Wide Web (WWW'01), New York, NY, USA, 2001, pp. 613–622
24 Basha, T., Moses, Y., Avidan, S.: 'Photo sequencing', in Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (Eds.): 'ECCV' (Springer Berlin, Heidelberg, 2012) (*LNCS*, 7577), pp. 654–667
25 Terruggia, R.: 'Reliability analysis of probabilistic networks'. PhD thesis, Universita degli Studi di Torino, 2010
26 Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: 'Introduction to algorithms' (MIT Press, 2001, 2nd edn.)
27 Stein, A.N., Hebert, M.: 'Occlusion boundaries from motion: low-level detection and mid-level reasoning', *IJCV*, 2009, **82**, (3), pp. 325–357
28 Maire, M.R.: 'Contour detection and image segmentation'. PhD thesis, University of California, Berkeley, 2009

doi: 10.1049/iet-cvi.2012.0287