

Integrated Segmentation and Depth Ordering of Motion Layers in Image Sequences

David Tweed and Andrew Calway
Department of Computer Science
University of Bristol, UK
{tweed, andrew}@cs.bris.ac.uk

Abstract

We describe a method to segment and depth order motion layers simultaneously in an image sequence. Previous approaches have tended to ignore the depth ordering issue or treat it as a post-processing operation. We argue here that motion estimation and segmentation are crucially dependent on depth order and hence that the latter should form an integral part of any layering scheme. Using an explicit model of boundary ownership allowing simultaneous assignment of motions to regions and extraction of depth order, the method fuses colour region segmentations with motion estimates obtained via block correlation. The motion estimates are then updated using a depth-dependent partial correlation. Experiments show the approach is effective.

1 Introduction

The problem addressed in this paper is how to segment frames of an image sequence into regions moving with different motions and to order the regions in terms of their relative depth in the scene, producing a *layered* representation [2]. Such layers correspond to the background and foreground objects and provide an effective means of processing sequences, able to account for effects such as occlusion, without the need to compute a full 3-D description. Layered representations have applications in a number of areas, most notably data compression where they are likely to play a key part of future video compression standards such as MPEG-4 and MPEG-7. Extracting layers is a difficult task, sharing many of the problems associated with still image segmentation and having the additional complications associated with motion ambiguity and depth ordering. These include those caused by aperture problems, the dependence of the link between boundary and interior motions on 3-D structure and motion, and occlusion/background uncovering.

Previous approaches to the problem have been based on using colour segmentation, motion estimation, region or edge tracking, and various combinations of one or more of these. There is a large body of work on using only motion information in the form of optical flow fields, based on a variety of segmentation techniques [1, 2, 3]. However, these approaches all suffer from the problem that reliable motion estimation requires model fitting over ‘within region’ pixels, which of course are unknown. Although techniques such as robust statistics [3] can overcome this to some extent, it remains a significant limitation

of such schemes. One way to address this is to incorporate colour information to adapt the motion estimation to the underlying image structure and this has been the approach of most recent work in the area. For example, in [4], motion and colour measurements are combined using an MRF framework, whilst in [5, 6] colour segmented regions are used to constrain the motion fitting. Alternatively, the segmented regions and their boundaries can be tracked over time to give motion estimates and hence segmentations as in [8] and [7]. In the main, these combined techniques have proved more effective.

The approach described here also combines colour and motion information in that it attempts to fuse colour segmentations with motion descriptions obtained using block correlations. The basic principle is to use the block estimates to assign motions to each of the segmented regions so as to minimise a measure of support based on motion compensated differences (MCD) and then to group regions exhibiting coherent motion to form the required layers. The key component is the method used for motion assignment. This is based on an explicit model of the relative depth of adjacent regions, and hence of *boundary ownership* [10], which predicts the expected distribution of MCD energy for a given motion and depth configuration. Testing these against the data then allows the ‘most likely’ configuration to be selected. In effect, the depth ordering constrains possible interpretations of the data, resulting in more robust motion assignment. It also means that depth ordering is automatically generated as part of the segmentation process, as in the recent approaches of [8] and [7], but in contrast to previous approaches in which it has been treated as a post-processing operation [9, 10]. The resulting depth ordering is also used to refine the motion estimates within blocks straddling motion boundaries by employing a depth-dependent partial correlation technique in which the knowledge of boundary ownership is used to reduce bias when estimating the motion of occluded regions. These estimates are fed back into the assignment and grouping process to give updated layers.

A motion segmentation algorithm based on the boundary ownership model was originally described in [11]. In this work we report significant improvements to the motion assignment and layering processes, and describe the depth dependent partial correlation for improving the motion estimates on each layer. Details of the motion assignment technique are given in the next section, followed in Section 3 by an overview of the layering algorithm. Results of experiments on real sequences are presented in Section 4.

2 Boundary Ownership and Depth

Our starting point is that want to assign motions to pre-segmented regions based on how well their interior pixels match those in the next frame if the motions are applied. The difficulty is that in real sequences interior pixels will often have low intensity variation, making comparison between motions problematical as many motions will fit the data equally well. This is *not* the case for pixels in the vicinity of boundaries - application of a motion may take some pixels into an adjacent region in the next frame, giving larger intensity differences. Hence, for such low variance regions, the motion can sometimes be deduced from pixels near the boundary. However, as illustrated below, the location of such pixels and the expected MCDs depends not only on the motions assigned to the regions but also their relative depth. This observation underlies our technique.

Consider the example in Fig. 1a, in which two constant intensity regions Λ_1 and Λ_2 have the motions \mathbf{v}_1 and \mathbf{v}_2 , and region Λ_1 corresponds to a surface nearest the camera, ie

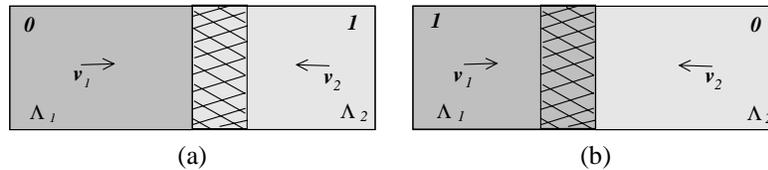


Figure 1: Boundary support regions (hashed) for both depth orderings of two constant intensity regions having different motions. The regions indicate non-zero MCD values for (a) motion v_2 and (b) motion v_1 , and thus determine boundary ownership.

occluding that associated with Λ_2 . Comparing MCD values for both motions at all pixels reveals that they are zero everywhere for motion v_1 and only non-zero for v_2 within the hashed region to the right of the boundary. The latter is $|(\mathbf{v}_1 - \mathbf{v}_2) \cdot \mathbf{n}|$ pixels wide, where \mathbf{n} is the exterior normal to the boundary of region Λ_1 , and it results both from the motions *and* the depth ordering. As shown in Fig. 1b, the reverse depth ordering gives a similar region but on the other side of the boundary and in favour of v_2 instead of v_1 . Thus, given the motions, the depth ordering is indicated by the location of the MCD energy on either side of the boundary. As pointed out by Darrell and Fleet [10], the key here is *boundary ownership* - occluding regions ‘own’ the boundary and this is reflected by ‘support’ for their motion within the hashed regions. We call these *Boundary Support Regions* (BSRs).

However, the BSRs and supported motions aren’t unique. For example, reversing the motion/depth configuration for both cases in Fig. 1 results in the same BSRs and supported motions. As discussed in [11], if we allow the possibility of the regions having the same motion (but *not* different motions when they’re at the same depth, which will be rare in real sequences), there are two possible BSR patterns, each corresponding to 5 possible motion/depth configurations. Of the latter, 3 correspond to both regions having the same motion but a different depth configuration; in the context of motion segmentation these can be regarded as equivalent. It also turns out that for a given motion assigned to one region, one of the two possible BSR patterns uniquely defines both the motion and the relative depth of the other region. This gives an ambiguity of two possible configurations for the remaining BSR state. Thus, although the BSRs are not uniquely associated with particular motion/depth configurations, they provide constraints on the possible interpretations of the MCD energy and it is this we use in the motion assignment.

2.1 Motion Support and Assignment

Motivated by the above, we adopt a support measure for assigning motions to regions based on two complementary terms: an *intra-region* term which represents support for a motion assignment amongst interior pixels; and an *inter-region* term which represents support from pixels within the predicted BSR given a depth ordering. In other words, we test a given motion assignment by looking for support both from within the regions themselves and within the BSR for each possible depth ordering. This approach has two advantages: (i) If the intra-region support is ambiguous, then the inter-region contribution will aid in determining motion assignments which are ‘valid’ interpretations of the data, ie supportive MCD values will only be found in the BSR corresponding to the ‘correct’ motion assignment and depth ordering. In effect, each motion/depth configuration pre-

dicts where to look for supportive MCD values, and the configuration with best overall support is selected; (ii) as depth ordering is an integral part of the assignment process, the scheme automatically gives the most likely depth ordering for the best configuration.

Both of the above terms are based on a measure of the degree to which a motion best fits a given region of pixels. In the segmentation algorithm described in Section 3 we reduce the amount of computation and the risk of overfitting by making local assignments based on two candidate motions. The support measures are then defined in terms of the log ratio of the respective MCDs for within region pixels, ie we take less account of pixels where both motions fit equally well and more account of those for which one of the motions is significantly worse. For two motions \mathbf{v}_0 and \mathbf{v}_1 , the measure of support for motion i within some region Λ is defined as

$$\beta_i(\Lambda) = \sum_{\xi \in \Lambda, d_i(\xi) > 0} \frac{d_i(\xi)}{|\Lambda|} \sum_{\zeta \in \mathcal{N}_\xi \cap \Lambda} \frac{d_i(\zeta)}{|\mathcal{N}_\xi \cap \Lambda|} \quad (1)$$

where $d_i(\xi) = \log |MCD_{1-i}(\xi)/MCD_i(\xi)|$ measures the support at ξ for motion i based on the motion-compensated differences for the two motions, ie for frames $x_1(\xi)$ and $x_2(\xi)$

$$MCD_i(\xi) = x_1(\xi) - x_2(\xi - \mathbf{v}_i) \quad (2)$$

The inner summation term in eqn (1) represents the degree of within region support for motion i within the neighbours \mathcal{N}_ξ of ξ . This gives greater weight to contributions from clusters of supporting MCD values, whilst decreasing the influence of isolated support. Thus, $\beta_i(\Lambda)$ will be large if the MCD for motion i is significantly less than that for motion $1 - i$ over sets of spatially coherent pixels across the region.

Given P regions and the two candidate motions, we denote the combined support measure for a motion assignment $\mathbf{m} = (m_0, \dots, m_{P-1})$ and depth assignment $\mathbf{l} = (l_0, \dots, l_{P-1})$ by $\alpha(\mathbf{m}, \mathbf{l})$, where $m_p = i$ if motion i is assigned to region p and l_p is the depth index for region p . This is defined as

$$\alpha(\mathbf{m}, \mathbf{l}) = \sum_{p=1}^P \alpha_{pm_p} + \sum_{p=1}^{P-1} \sum_{q=p+1}^P \alpha_{pqm_p}^{l_{pq}} \quad (3)$$

where α_{pi} and $\alpha_{pqi}^{l_{pq}}$ denote the intra-region and inter-region support terms, respectively, and l_{pq} indicates the relative depth ordering of regions p and q , ie $l_{pq} = 0$ if $l_p \leq l_q$ and $l_{pq} = 1$ if $l_p > l_q$. The former are then defined using support measure $\beta_i(\Lambda)$ as

$$\alpha_{pi} = \beta_i(\Lambda_p) \quad \alpha_{pqi}^0 = \beta_i(\Lambda_{pqi}^0) \quad \alpha_{pqi}^1 = \beta_{1-i}(\Lambda_{pqi}^1) \quad (4)$$

where Λ_{pqi}^0 and Λ_{pqi}^1 are the BSRs for assigning motion i to region p and motion $1 - i$ to region q for the two possible depth orderings. Note within the BSRs the expected motion to be supported depends on the depth ordering, eg if region p is occluding then motion i should be supported within the BSR, whilst motion $1 - i$ should be supported for the reverse ordering. Using the above definitions, motion segmentation of the P regions is achieved by seeking the motion and depth assignments \mathbf{m} and \mathbf{l} which maximise $\alpha(\mathbf{m}, \mathbf{l})$ in eqn (3). This gives the ‘best’ motion assignment and the corresponding depth ordering. In the next section we describe how this is incorporated into the layering algorithm.

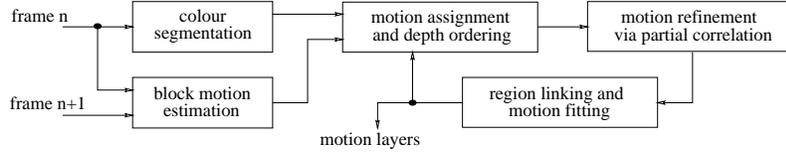


Figure 2: Block diagram of motion layering algorithm.

3 Integrated Layering and Depth Ordering

Fig. 2 shows the main components of the layering algorithm: colour segmentation and motion estimation, motion assignment, motion refinement, and region linking.

3.1 Colour Segmentation and Motion Estimation

Given two frames n and $n + 1$, the first stage consists of deriving a colour region segmentation for frame n and computing block based motions estimates between the two frames, where the latter take blocks in frame n into corresponding blocks in frame $n + 1$. The segmentation needs to provide good estimates of the colour region boundaries, and in particular to capture all of the motion boundaries. We assume that the latter are a subset of the former, ie regions of different colour can move together but not vice versa, and so slight over-fitting of the segmentation is acceptable and preferred to under-fitting. In the experiments we used the segmentation algorithm developed by Sinclair [13] (also used by Smith and Cipolla [7] in their boundary tracking technique). This combines a colour edge extraction process with region growing using seeds at peaks in the Voronoi image derived from the edge map. Essentially, regions are grown until meeting an edge or a pixel whose colour is too far from the local region mean. We found this gave reasonable segmentations over a range of different frames and sequences without excessive computational cost.

The block based motion estimates are obtained using normalised correlations performed via the frequency domain and applied to blocks with 50% overlap. Although the estimates have limited resolution around boundaries, they have the advantage being less sensitive to noise, than for example optical flow estimates, and are more likely to avoid aperture problems. They also provide a concise description of the motion field which has computational benefits. The present work uses a simple translational model of the block motion at a single resolution, although in principle the method could also use multiresolution and affine estimates obtained using the techniques described in [12] for example. Each correlation also provides both a certainty measure for the estimate via its peak value and an indication of its covariance via the energy distribution about the peak [12].

3.2 Motion Assignment

The next stage is to fuse the colour and motion information by assigning motions to colour regions using the set of estimates produced by the correlations. This is done within the same block structure used by the latter, ie given a set of colour regions $\Lambda = (\Lambda_1, \dots, \Lambda_P)$ and a set of blocks $B = (B_1, \dots, B_N)$, we seek to assign motions in block k to the set of regions

$$\Gamma_k = \{B_k \cap \Lambda_i \mid i = 0..N, B_k \cap \Lambda_i \neq \emptyset\} \quad (5)$$

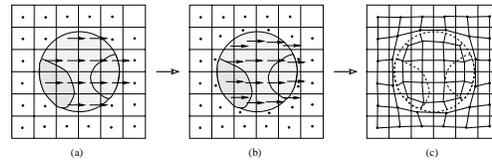


Figure 3: Layering process: (a) colour segmentation and block based motion estimates; (b) assignment of motions to regions; (c) layers formed by region linking.

This lets spatially variable motion fields for large colour regions span several blocks.

The assignment process involves two steps. First, an initial assignment is made based on the assumption that each block will contain at most two motion regions and that each motion can be obtained either from the block or one of its 8 neighbours. The latter tackles the limited resolution of the block estimates, whilst the former reduces both the computational cost of the assignment process and the risk of overfitting the motions, while being a reasonable assumption for a large majority of blocks if the block size is chosen carefully. Thus, for a given block k , we select the two ‘best’ motions from the 9 local estimates by comparing the intra-region support measures as defined by eqns (1) and (4) within the regions Γ_k for each distinct motion. The two motions which give the two largest support sums computed over all regions are then selected and used as candidate motions to assign to the colour regions using the method described in the previous section, ie we find the motion/depth configuration which maximises $\alpha(\mathbf{m}, \mathbf{l})$ in eqn (3). If only one distinct motion is present amongst the neighbouring blocks then the centre block is defined as a single motion block and all its regions assigned the same motion. The small number of regions in each block means that maximising eqn (3) can be done easily using an exhaustive search, with the intra- and inter- region terms being computed prior to the search and then the support for each configuration generated by the summation of the relevant terms.

Following this initial assignment, the assignments made in neighbouring blocks are merged. For each block, this is achieved by comparing each of the possible motion segmentations indicated by the 9 neighbours and selecting that which maximises the support measure used in the initial assignment. There are two benefits from this merging process: isolated assignment errors caused by bad motion estimates for example can be corrected if better assignments have been made in neighbouring blocks; and blocks containing more than two motion regions can be identified and correctly assigned if the additional motion regions have been detected in neighbouring blocks. (Note this process implicitly creates the linkages between neighbours which are used in section 3.4 to group consistently moving pixels into moving patches corresponding to objects in the scene.) The result for each block is a division of the regions Γ_k into two sets, each representing a distinct motion region with an associated depth order. Fig. 3 illustrates an example of the process, in which a circular region consisting of three colour sub-regions is moving to the right on a stationary background. The motion estimates obtained from the correlations are shown in their respective blocks in Fig. 3a and the result of the motion assignment is shown in Fig. 3b.

3.3 Motion Refinement

In the above process, regions within blocks which straddle motion boundaries may obtain their motion assignment from neighbouring blocks. Although this may be acceptable in some cases, it limits possible variation of the motion field and may lead to incorrect assignments. This is addressed by using the motion segmentations obtained in such blocks to perform partial correlations and so derive updated motion estimates [12]. In essence, for a two motion region, we set all the pixels in one region to 0 and correlate with the corresponding block in the next frame. However, we also make use of the depth order of the motion regions. For an occluded region, simple partial correlation may lead to biased estimates caused by the boundary, since the boundary carries the motion of the occluding region, not that of the occluded region. Hence, when estimating an updated motion for the latter, we attempt to reduce the effects of the boundary by smoothing off the intensity values at pixels near the region boundary prior to correlating. Specifically, denoting the two motion regions in a block by Π_1 and Π_2 , we generate ‘nulling’ functions

$$n_i(\xi) = \begin{cases} 1 & \text{if } \xi \in \Pi_i, |\xi - BND(\xi)| > D_i \\ r(|\xi - BND(\xi)|) & \text{if } \xi \in \Pi_i, |\xi - BND(\xi)| < D_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where $BND(\xi)$ is the closest pixel to ξ on the boundary between the two motion regions and $r(x)$ is a monotonically decreasing function such that $r(D_i) = 1$ and $r(0) = 0$. We set D_i to 0 when Π_i is occluding Π_{1-i} and to a non-zero value (depending on the size of Π_i) when Π_i is occluded. To get the new motion estimates for region i we correlate the nulled block $n_i(\xi)x_1(\xi)$ with the corresponding block in the next frame. This is repeated for the second motion region. The new motion estimates obtained from the partial correlations are used to derive updated assignments in the boundary blocks and their neighbours.

3.4 Region Linking

The final stage involves grouping together motion regions which exhibit coherent motion. This is done using a region-adjacency graph defined within the block structure. After the assignment process, each block has a motion segmentation – typically consisting of one, two or possibly 3 motion regions – and we seek to link these based on their assigned motions as follows: For a given block, each of its motion regions are compared with those in its four neighbours and a link is formed between two regions if they are contiguous, their depths compatible and their motions are sufficiently close. Where this gives a region linking to two different regions in the same block we keep the link to the most similar. This gives groups of connected motion regions with a common depth corresponding to the required layers. An example is shown in Fig. 3c. (The outer nodes of the graph for the circular region are linked to the boundary, indicating ownership of it and hence the depth order.)

4 Experiments

Fig. 4 shows contributions to β_i (eqn (2)) from individual pixels ξ for the block shown in (a), with whiter indicating more certain of the choice of motion. It is clear the background concrete is in one motion class while the hat brim is in the other. The body of the hat is as

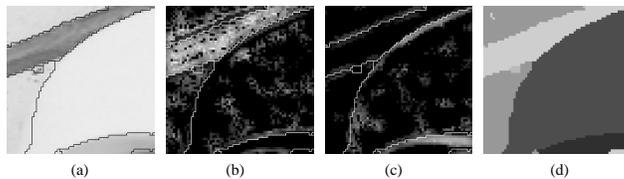


Figure 4: (a) Local grey level segmentation; (b) intra- and inter- region support values for background motion; (c) same as (b) for head motion; (d) final motion assignment.

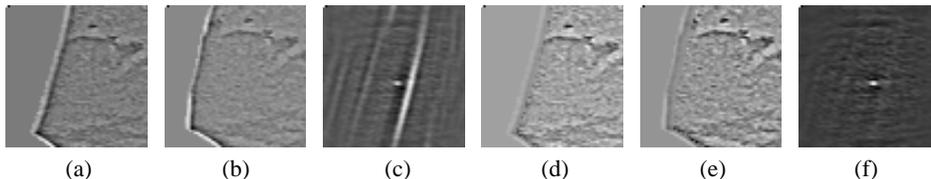


Figure 5: (a,b) 'Nulled' background blocks; (c) correlation has peak for foreground motion; (d,e) as (a,b) with smoothing from eqn (6); (e) correlation with correct peak

ambiguous (due to its extreme 'flatness'). Note this doesn't necessarily mean a constant MCD value; as here noise can have a disproportionate effect, as evidenced by the presence of small clusters within the hat for both motions. However the strong structures along the boundaries in (c) give BSRs showing it is moving with the brim above the background, as shown by motion segmentation (d).

Fig. 5a-c shows that direct partial correlation can produce the wrong result if sufficient hi-pass energy around a foreground edge is included in a background correlation: although there is a peak at the background motion position, it is dominated by the peak in the smear corresponding to the foreground 'edge' motion. Fig. 5d-f shows that smoothing removes enough energy around the edge to give a correct background motion estimate. Fig 6 shows examples for the foreman and hands sequences. For each row, image (a) shows the local segmentation results after neighbour correction; the two greylevels (which correspond to the local motion label) do not match as they are assigned purely locally; image (b) shows the region adjacency graph (black links) overlaying the block structure (straight white lines) and detected motion boundaries (white outline); image (c) shows the foreground component for the foreman examples and a the two depth components for the hands image, with the lighter one being the foreground region. The hand example shows the two foreground components. Note that they are judged to be at the same depth because depth cannot be determined using BSRs unless occlusion is occurring.

5 Conclusions

We gave a local layer model relating patterns of MCD energy along boundaries to motion/depth configurations, giving a motion segmentation algorithm using *interior* and *boundary* information for greater robustness. Joining local results using motion and 'pixel classification' similarity produces a global set of layers. Future work will investigate using a temporal window to gain robustness and produce a more compact representation.

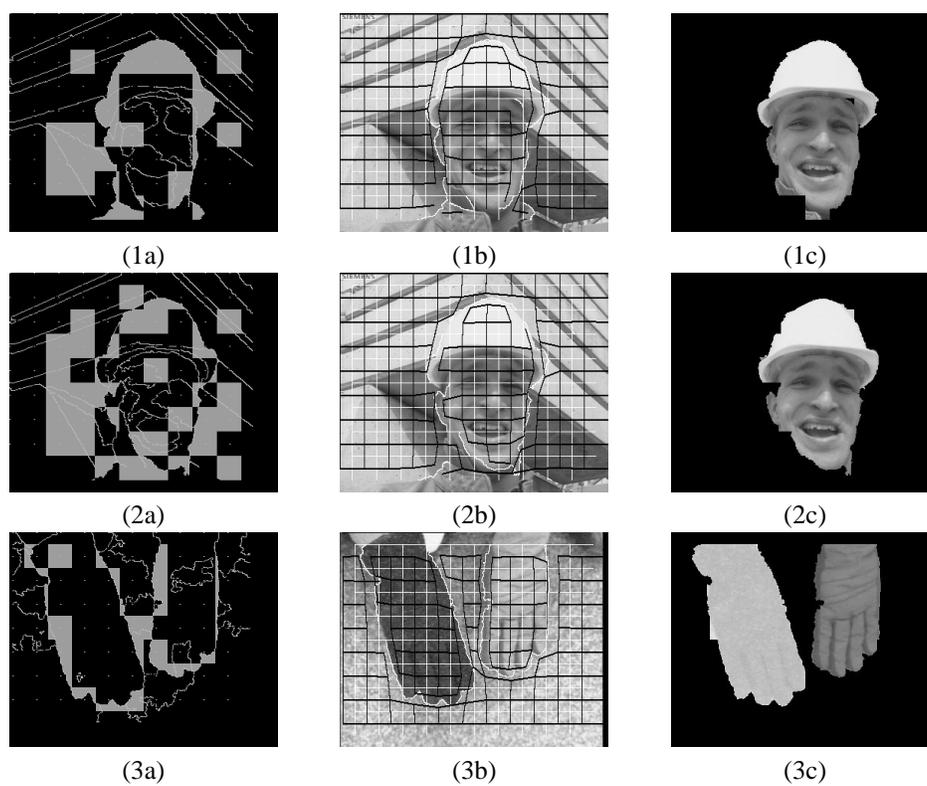


Figure 6: Examples from the foreman and hands sequences of (a) local motion segmentations; (b) region adjacency graphs (black links) overlaid on block structure and motion boundaries (white lines); (c) foreground component(s)

Acknowledgements

The authors are grateful to NDS Ltd for financial support of the work.

References

- [1] D.W.Murray and B.F.Buxton, "Scene segmentation from visual motion using global optimization", *IEEE PAMI*, 9(2), pp. 220-228, 1987.
- [2] J.Wang and E.H.Adelson, "Representing moving images with layers" *IEEE Trans Image Processing*, 3(5), pp. 625-638, 1994.
- [3] M.J.Black and P.Anandan, "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields", *Computer Vision & Image Understanding*, 63(1), pp.75-104, 1996.
- [4] M.J.Black, "Combining intensity and motion for incremental segmentation and tracking over long image sequences", *Proc. 2nd Euro Conf Computer Vision*, 1992.
- [5] Y.Altunbasak, P.E.Eren, and A.M.Tekalp, "Region-based parametric motion segmentation using color information", *Graphical Models and Image Processing*, 60(1), pp. 13-23, 1998.
- [6] M.J.Black and A.D.Jepson, "Estimating optical flow fields in segmented images using variable-order parametric models with local deformations", *IEEE Trans on Pattern Analysis and Machine Intelligence*, 18(10), 1996.
- [7] P.Smith, T.Drummond, and R.Cipolla, "Edge tracking for motion segmentation and depth ordering", *Proc British Machine Vision Conf*, 1998.
- [8] L.Bergen and F.Meyer, "Motion segmentation and depth ordering based on morphological segmentation", *Proc ECCV*, 1998.
- [9] M.Nitzberg and D.Mumford, "The 2.1-D Sketch", *Proc 3rd Int Conf on Computer Vision*, 1990.
- [10] T.Darrell and D.Fleet, "Second-order method for occlusion relationships in motion layers", MIT Media Lab tech Report 314.
- [11] D.S.Tweed and A.D.Calway, "Motion segmentation based on integrated region layering and motion assignment", *Proc Asian Conf on Computer Vision* p1002-1008, 2000.
- [12] Calway A.D, Krüger S.A, Tweed D.S, "Motion estimation using adaptive correlation and local directional smoothing", *Proc. IEEE ICIP*, 1998.
- [13] D.Sinclair, "Voronoi seeded colour image segmentation", Tech Report, AT&T Labs Cambridge, 1999.