Approaches to 3D Motion Estimation with Multiple Cameras *

Thomas Popham,
Department of Computer Science,
Warwick University,
tpopham@dcs.warwick.ac.uk

December 19, 2010

1 Introduction

The purpose of this article is to review the main approaches described in the literature to 3D motion estimation with multiple cameras. The review assumes that the initial 3D geometry of the scene has already been estimated using techniques such as laser-scanners [20], time-of-flight cameras [27], structured light measurements [44] or multiple view reconstruction [45]. This review concentrates on *passive* techniques that do not require manipulation of the scene, such as the addition of active LEDs markers or 'ultraviolet paint' [3], although some passive optical-marker techniques are included as a reference baseline. It is also important to note that this review focuses on recovering the dense motion of the scene, rather than tracking a single 3D object with methods such as [2, 22, 42].

The dense three-dimensional motion of the scene is often referred to as the *scene flow* [51], and is analogous to the concept of optical flow for describing the motions within a 2D image. There are several approaches to estimating scene flow: some

*This article is taken from chapter 2 of my thesis [41] available at: www.dcs.warwick.ac.uk/∼tpopham/thesis.pdf

methods require very specific conditions, whilst other methods are very general. This property forms the structure for this review: the most constrained approaches are reviewed first and the most general approaches are reviewed last.

The classical method used by the entertainment industry, *optical marker* motion capture, relies on placing reflective markers in the scene to be tracked using standard computer-vision techniques [56]. Although the motion can be tracked for long sequences, the approach is highly restrictive, since markers must be manually inserted into the scene. The next group of methods attempt to overcome this problem by using a *model-based approach* to achieve markerless motion capture, and although accurate results may be obtained for short sequences, even this method is restrictive: the scene motion model must be known beforehand. The most general set of motion estimation methods do not use a specific motion model of the scene, and instead rely on a more general assumption: motion coherency. This assumption means that we expect the neighbouring points of a scene (or image) to move smoothly with respect to one another. This leads to the *optical flow approaches*, which make the assumption that the neighbouring image pixels move together. Although the motion of general scenes may be estimated using this approach, the assumption does not always hold (e.g. at object boundaries). The last category of techniques to be reviewed are the *surface-based approaches*, which assume that neighbouring surface points move smoothly with respect to one another. Each of these motion estimation approaches is now reviewed in detail.

2 Passive Optical-Marker Approaches

Passive motion capture techniques [1] use multiple cameras to track a set of reflective markers, such as the ones shown in figure 1. Since the markers provide clear targets in each image, they can be tracked easily using standard computer

vision techniques [56]. The main problem in tracking the markers is ensuring that the 2D tracks from each camera are correctly triangulated, so that the full set of 3D tracks is recovered. Since a marker may only be visible from a narrow range of viewing angles, multiple cameras are required (up to a hundred) for tracking a wide range of motions. However, due to the data association problem [4], the number of markers is limited, which means motions may be only be captured at the general skeleton level. Despite these problems, optical-marker techniques are probably still the only current reliable vision-based solution for motion capture in 'real-world' applications such as the film and video-game industries.

(a) (b)

Figure 1: Two examples of a passive optical markers for motion capture [1].

3 Model Based Approaches

Since the motion of a scene can be very complex, introducing a specific motion model of the scene significantly reduces the number of parameters to be estimated. Of course, the accompanying limitation is that the scene motions can only be estimated if they are included in the motion model. Usually the model is in the

form of a human skeleton [34, 35], although there are other possibilities such as a hand model [32, 46]. There are three aspects to a model-based approach: the form of the model; the method of estimating the conditional probability of the model given the images (the measurement model); and the method of tracking the model with the given measurement model. The most common human-body model is composed of cylinders [25], although many other forms are possible, such as boxes, ellipsoids [38], cones [16], spheres [15] and superquadrics [21]. An obvious limitation of these models is that they assume that the subject is not wearing loose clothing, which would mean that the underlying cylinders or cones cannot be observed. More recently, attempts have been made to incorporate a mesh model, which is linked to the underlying skeleton model, in order to overcome the 'loose garment' problem [20, 9].

There are many methods of evaluating the conditional probability of the model pose given the available images. Edges are a commonly used cue [25, 32, 46], as they can usually be reliably extracted from the images and are invariant to lighting conditions. In order to derive a suitable measurement model, the chamfer distance between the actual edges and the predicted edges (from the model parameters) is frequently used [46]. Silhouettes are also a common image cue [20, 9], as they lead to a straightforward measurement model: the pixelwise difference between the real silhouettes and the predicted silhouettes from the model parameters. There are many other image cues for forming a measurement model, including colour [46], optical flow [32], stereo [38] and scale-invariant features [20].

Although the form of the human body model and measurement model are important factors in this approach, the key challenge is finding the most likely state (over time), given the measurement model. The challenge comes from the fact that even with a relatively simple model, the set of possible poses is very large, and there is often more than one model pose that explains the observed images.

Earlier solutions to the problem tended to use a gradient descent technique for updating the model parameters at every frame [35, 37]. This was then developed by taking a probabilistic approach to modelling the system state, using a Gaussian density to describe the uncertainty at each frame (the Kalman Filter) [24, 53]. The problem with these approaches is that there is no adequate handling of multiple explanations for the observed data. The use of a particle filter [28] for tracking the model overcomes this problem by using a set of weighted particles to model general state probabilities, including multimodal distributions [16, 20]. Other methods which use a multimodal model of the system state include the work of Han *et al.*, who model the state of each limb using a Gaussian Mixture Model, which is propagated to neighbouring skeleton nodes using belief propagation [23].

Modern model-based human trackers are usually demonstrated on sequences that are several hundred frames long, and may even contain subjects wearing loose garments [20]. However, there are several difficulties with model-based approaches. The first problem is that the initial subject pose has often to be entered manually, although some methods to find the human pose in images do exist [17]. The second problem is that, due to the large variation in subject appearance and size, the human-skeleton model must be customized to each subject [20]. A third problem is that the subject must not interact with any other objects (such as props or equipment), as these are unaccounted for in the model.

In recent years, techniques have emerged for estimating a scene motion model by finding an articulated model (such as a skeleton) that explains the observed motions in the scene [49, 10, 11]. Many of these methods tend to adopt a volumetric approach, which involves estimating the object volume at each time point, and then corresponding the estimated volumes to provide the motion model. The visual hull is a common choice for the modelling the scene volume, presumably due to its ease of computation. However, other methods attempt to segment a mesh model in

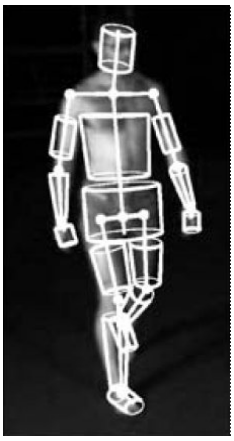

Figure 2: An example skeleton from the model-based approach of Deutscher and Reid [16].

order to find the articulated components.

Theobalt *et al.* fit a set of superquadric shapes to a set of voxels at each time step [49]. A superquadric is able to model cubes, spheres, ellipsoids and octahedrons, making it a very flexible tool for modelling the scene components. A hierarchical, top-down approach is taken for fitting the superquadrics to the voxel model at each frame. The set of superquadrics at time t is then matched with the set of superquadrics at time $t+1$ by minimising the distance between the matched superquadric centres. Since the superquadrics are independently fitted to the voxels at time t, a one-to-one mapping between the superquadrics is not guaranteed, and therefore some superquadrics are either split or merged to ensure a one-to-one mapping. Since a rigid body part may be composed of more than one superquadric, the paths of superquadrics are compared and if they are sufficiently similar, then they are labelled as belonging to the same body part and are therefore merged.

In [49], results for the method are demonstrated on 3 synthetic sequences and one 'real' sequence. The fitted superquadrics only provide a very high-level approximation to voxel models and do not provide an accurate representation of the

voxels (e.g. a snowman comprised of 3 spheres and a hat is approximated by 2 superquadrics). Since the joint positions are only estimated for the upper body, it is questionable whether the proposed method would be able to estimate a full human skeleton from voxel data.

Cheung *et al.* also use the visual hull to estimate the components of an articulated object but also incorporate coloured surface points as an additional cue for resolving correspondences between body parts [10]. Expectation Maximisation is used to simultaneously cluster and estimate the motion parameters of each cluster component. Since only two body parts may be segmented at once, a separate sequence must be captured for each body joint, whereby all other body joints remain rigid. Once the body model is estimated, it may be tracked with full movement. The tracking is performed in a hierarchical manner, starting with estimating the torso position, and then proceeding to estimate the position of neighbouring limbs. The position of each body part is estimated by maximising a consistency score for the body part's coloured surface points and the input images. As with many methods, successful tracking is demonstrated on real-world sequences, but short sequences are used (up to 200 frames), which shows the potential difficulty of long-term human motion capture, even when a strong prior model is available.

4 Optical Flow Approaches

This class of methods estimate the scene flow by combining optical flow estimates from several cameras. Scene flow is defined as a three-dimensional flow field describing the motion at every point in the scene [51]. It is necessary to define some notation in order to explain this concept. A three-dimensional point in the scene is $\mathbf{q} = (x, y, z)^T$ and a two-dimensional point on a camera image plane is $\mathbf{p} = (m, n)^T$. The scene flow is then: $\frac{d\mathbf{q}}{dt} = \left(\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}, \frac{\partial z}{\partial t}\right)^T$ and the optical flow is

$\frac{\partial \mathbf{p}}{dt} = \left(\frac{\partial m}{\partial t}, \frac{\partial n}{\partial t}\right)^T$. The optical flow is therefore a projection of the scene flow onto an image plane:

$$\frac{d\mathbf{p}}{dt} = \frac{\partial \mathbf{p}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{dt} \tag{1}$$

where $\frac{\partial \mathbf{p}}{\partial \mathbf{q}}$ is the effect of a change of a scene point upon the projected image point. This leads to a system of linear equations for each point in the scene:

$$B\frac{d\mathbf{q}}{dt} = U \tag{2}$$

where:

$$B = \begin{pmatrix} \frac{\partial m_1}{\partial x} & \frac{\partial m_1}{\partial y} & \frac{\partial m_1}{\partial z} \\ \frac{\partial n_1}{\partial x} & \frac{\partial n_1}{\partial y} & \frac{\partial n_1}{\partial z} \\ . & . & . \\ & & \\ . & . & . \\ \frac{\partial m_N}{\partial x} & \frac{\partial m_N}{\partial y} & \frac{\partial m_N}{\partial z} \\ \frac{\partial n_N}{\partial x} & \frac{\partial n_N}{\partial y} & \frac{\partial n_N}{\partial z} \end{pmatrix}, \tag{3}$$

and:

$$U = \begin{pmatrix} \frac{\partial m_1}{\partial t} \\ \frac{\partial n_1}{\partial t} \\ . \\ \\ . \\ \frac{\partial m_N}{\partial t} \\ \frac{\partial n_N}{\partial t} \end{pmatrix}. \tag{4}$$

The matrix B may be obtained either numerically or analytically from the camera projection matricies, and matrix U is the optical flow from each camera.

An advantage of this approach is that state-of-the-art optical flow techniques can easily be applied, and the algorithm is ideal for parallel implementation. A disadvantage is that any regularization must be applied in the images without any knowledge of the depth. This is a suboptimal approach for three reasons: first, without the knowledge of depths, motion smoothing across object boundaries becomes unavoidable unless robust cost functions are applied; secondly, a uniform regularization in the image space is likely to result in a non-uniform regularization in the scene, due to the fact that objects lie at different depths from each camera; and thirdly, the approach ignores that the fact that the optical flows should be consistent with each other.

Several methods ensure that the optical flow estimates are consistent with one another, by jointly estimating the optical flow in both a left image and a right image [54, 26, 29]. This leads to a 2.5D framework, in which the motion of a 3D point is parameterised by a change in image co-ordinates plus a change in depth. The scene flow is estimated using an energy model consisting of a data term and a smoothness term:

$$E(u, v, d') = E_{data}(u, v, d') + E_{smooth}(u, v, d'). \tag{5}$$

where (u, v) are the motions in the m- and n-directions, and d' is the change in disparity. The data term consists of three components[1] : (1) the brightness consistency in the left image; (2) the brightness consistency in the right image; and (3) the brightness consistency between left and right images for the moved

[1] Isard and MacCormick [29] use three slightly different data terms, but the principle remains the same.

point. This is expressed mathematically as follows:

$$
\begin{aligned}
E_{data} = & \int_{\Omega} \Psi \left((\mathcal{I}_{l,t+1}(m+u, n+v) - \mathcal{I}_{l,t}(m,n))^2 \right) dmdn \\
& + \int_{\Omega} \Psi \left((\mathcal{I}_{r,t+1}(m+d+u+d', n+v) - \mathcal{I}_{r,t}(m_d,n))^2 \right) dmdn \\
& + \int_{\Omega} \Psi \left((\mathcal{I}_{l,t+1}(m+d+u+d', n+v) - \mathcal{I}_{r,t+1}(m+u,n+v))^2 \right) dmdn,
\end{aligned} \tag{6}
$$

where $\mathcal{I}_{l,t}(m,n)$ and $\mathcal{I}_{r,t}(m_d,n)$ are the intensity values in the left and right images at time t, Ω is the domain of each image, Ψ is a robust cost function and d is the disparity between the left and right images (which are assumed to be rectified, so that changes in depth only relate to a change in the m image co-ordinate). The smoothness term penalizes differences in neighbouring image motions:

$$
E_{smooth} = \int_{\Omega} \Psi \left(\lambda(|\nabla u|^2 + |\nabla v|^2) + \gamma |\nabla d'|^2 \right) dmdn. \tag{7}
$$

where λ controls the strengths of smoothness constraint for changes in image co-ordinates and γ controls the strength of the smoothness constraint for changes in depth. Note that robust cost functions are required for both the data and smoothness terms to prevent smoothing across object boundaries. This is necessary, since the same level of motion smoothing is applied between two pixels, even if they have totally different disparities.

There are two main approaches to minimizing the energy in equation (5): a continuous variational approach [54, 26] and a discrete Markov Random Field [29]. For the variational approach, the use of robust cost functions leads to a rather complicated energy minimization, with three nested loops: the outer loop implements the multiresolution aspect of the minimization strategy; the middle loop warps the images towards the final solution; and the inner loop uses successive over-relaxation alongside a linearization of the robust cost function to calculate

the parameters for the next warp. Such an approach is strongly related to modern optical flow techniques such as [7, 47]. For a discrete Markov Random Field approach [29], a relatively straightforward MAP estimation technique such as Belief Propagation can be used. The disadvantage of the discrete approach is that it is computationally expensive, as a set of data term costs has to be evaluated at fixed intervals within the solution space.

5 Surface Based Approaches

Several methods estimate the scene motion using a surface model, which can be represented using either a piecewise model [8, 36], a mesh model [18, 19, 12, 14] or a level-set framework [39]. The major difference between these methods is the notion of *connectivity*: with a piecewise description, there is no knowledge of which elements are connected to each other, but for a mesh model this knowledge is made explicit. There are many approaches to estimating the surface motion, which include: comparing images of the surface at time t and $t+1$ [8, 36, 39]; using the silhouettes at time t and time $t+1$ to derive a set of constraints upon the surface motion [12]; feature and descriptor based approaches [12, 57]; and optical flow constraints [14]. We begin by reviewing the methods that use a piecewise surface description, and then proceed to the level-set and mesh techniques.

Carceroni and Kutulakos [8] use a set of surface elements or *surfels* to form a piecewise description of the surface. Each surfel is represented by the following components: the time t, the shape $\mathcal{S}$, a reflectance model $\mathcal{R}$, the curvature $\mathcal{B}$ and a motion component $\mathcal{M}$. It is immediately obvious that this is a complex description: 25 parameters are necessary to describe each surfel. Each surfel is fitted to the surface using a volumetric approach, in which every voxel in the model is tested for the presence of a surfel using a combination of sampling an

optimization steps.

Once a surfel has been fitted to the surface, the motion across a single frame is estimated. The motion $\mathcal{M}$ consists of nine components: three for translation; three for rotation and three for shearing and scaling effects. These parameters are estimated using least-squares to minimize $||A\mathbf{x} - b||^2$, where A contains the image gradients with respect to each motion component, $\mathbf{x}$ contains the motion parameters to be estimated, and b contains the change of image on the surfel surface with respect to time. Since the reflectance properties of each surfel are estimated, it is possible to consider the effect of changing surface normal upon the change of illumination of the patch surface. For a Lambertian surface, the intensity of an imaged point $\mathbf{q} \in \mathbb{R}^3$ as a function of the point surface normal $\mathbf{n}(\mathbf{q};t) \in \mathbb{R}^3$, albedo a and net illumination radiance $\mathbf{r}(\mathbf{q};t) \in \mathbb{R}^3$ is:

$$\mathcal{I}(\mathbf{p}) = -Ka[\mathbf{n}(\mathbf{q};t).\mathbf{r}(\mathbf{q};t)], \tag{8}$$

where the image point $\mathbf{p}$ is obtained by the projection of the scene point $\mathbf{q}$. Now differentiating equation (8) with respect to time gives:

$$\frac{d\mathcal{I}}{dt} = -Ka\frac{d}{dt}[\mathbf{n}.\mathbf{r}] = \frac{\partial\mathcal{I}}{\partial\mathbf{p}}\frac{d\mathbf{p}}{dt} + \frac{\partial\mathcal{I}}{\partial t}. \tag{9}$$

Some explanation of the notation is helpful: $\frac{d\mathcal{I}}{dt}$ means the change in brightness of the moving image point, whereas $\frac{\partial\mathcal{I}}{\partial t}$ means the change in brightness at a fixed image point. In most motion estimation algorithms, it is assumed that the brightness of the same point in the image remains constant, to give the optical flow constraint:

$$\frac{\partial\mathcal{I}}{\partial\mathbf{p}}\frac{d\mathbf{p}}{dt} + \frac{\partial\mathcal{I}}{\partial t} = 0, \tag{10}$$

This makes the assumption that the surface normal and illumination conditions

are not changing, since $\frac{d}{dt}[\mathbf{n}.\mathbf{r}] = 0$. However, since the surfel models the surface normal and radiance, then a more general constraint can be made:

$$-Ka\mathbf{r}.\frac{\partial \mathbf{n}}{\partial t} = \frac{\partial \mathcal{I}}{\partial \mathbf{q}}\frac{d\mathbf{q}}{dt} + \frac{\partial \mathcal{I}}{\partial t}. \tag{11}$$

Very few motion estimation methods incorporate the effect of changing surface normal upon the surface illumination, and perhaps this goes some way to justifying the complex nature of the model.

Mullins *et al.* use a simpler piecewise description of the scene surface: a set of planar patches, where each patch is description by a centroid $\mathbf{q}$ and a surface normal $\mathbf{n}$ [36]. In order to track these patches over time, they are clustered using a Gaussian Mixture Model, with the assumption that the patches within a component move in a rigid manner. The clustering is achieved through the use of a Multiresolution Gaussian Mixture Model [55], which means that the patches are clustered using a hierarchical structure. This hierarchical nature of the model implies that rigid-body motion can expected at various 'resolutions' of the scene, whether at the global object level or at the finer detail level. Each component is then tracked using a particle filter [43], which is a numerical method for tracking 'targets' with nonlinear measurement functions and non-Gaussian noise densities. The final motion field is obtained by interpolating between the estimated motions at neighbouring cluster components.

Applying a local rigid-body motion model to a scene is potentially a powerful method of constraining the range of possible motions, since it ensures that the 3D tracking problem is not ill-posed. However, in reality, finding such a motion model is a 'chicken-and-egg' problem: the scene motions are required to build the rigid-body model, but the rigid-body model is required to estimate the scene motions. The result of this limitation is that the clustered patch components do not necessarily reflect the rigid components within the scene, which means the

Gaussian mixture model must be reinitialized at regular time intervals.

We now consider the algorithms using mesh and level-set representations. Pons *et al.* estimate the scene flow using a variational approach to 'evolve' a surface model to its position at the next frame [39]. This is achieved using an image-based matching score, which is summed over all cameras. Many surface-based methods estimate the motion at each surface point, but Pons *et al.* take a different approach by directly estimating the dense scene flow field. Let l be a function that maps every point in the scene to a 3D motion vector: $l : \mathbf{q} \in \mathbb{R}^3 \rightarrow \mathbf{f} \in \mathbb{R}^3$, where $\mathbf{f}$ is the motion vector $(u, v, w)^T$. The cost function to be minimized as a function of the scene flow l is:

$$E(l) = \sum_{c \in \mathcal{C}} f(\mathcal{I}_{c,t}, \mathcal{I}_{c,t+1} \circ \Pi_c \circ d(l) \circ \Pi^{-1}_{c,S^t}), \tag{12}$$

where $\circ$ denotes function composition, $\mathcal{I}_{c,t}$ is the image in camera c at time t, Π_c is a function mapping a point on the surface into camera c, $d(l)$ is a function applying the effects of the scene flow, Π^{-1}_{c,S^t} is a function mapping a point from the camera image plane onto the surface S, and $f(\mathcal{I}_i, \mathcal{I}_j)$ is a general matching function (such as cross-correlation), between images i and j. This is an interesting approach, since a surface model is used with an image-based matching to score to estimate a dense motion field. In order to regularize the solution, the Laplace-Beltrami operator is used on the surface. As with many other scene flow papers, only a single frame of motion is estimated.

The use of a discrete Laplacian operator for ensuring motion smoothness is typical for mesh-based algorithms since it preserve the fine mesh details [14, 12, 5, 50, 52]. If the vertices belonging to a 3D triangular mesh are denoted $\{\mathbf{q}_i\}_{i=1}^N$,

then we may define the Laplacian mesh operator as follows:

$$D(\mathbf{q}_i) = \mathbf{q}_i - \frac{1}{N_m} \sum_{j \in \mathcal{M}_i} \mathbf{q}_j, \tag{13}$$

where $\mathcal{M}_i$ is a set containing the indexes of the neighbours to mesh node i and N_m is the number of elements in $\mathcal{M}_i$. The Laplacian mesh operator is a linear transformation and can be described by the matrix L:

$$L_{ij} = \begin{cases} 1 & i = j \\ -\alpha & j \in \mathcal{M}_i \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The *differential co-ordinates* of the mesh are now obtained using $L\mathbf{m} = \mathbf{k}$, where $\mathbf{m} = (\mathbf{q}^{(x)}, \mathbf{q}^{(y)}, \mathbf{q}^{(z)}))$ is a vector of mesh vertices and $\mathbf{k} = (\mathbf{k}^{(x)}, \mathbf{k}^{(y)}, \mathbf{k}^{(z)})$ is a vector of differential co-ordinates. Note that by solving the linear system, it is possible to recover the original mesh co-ordinates, given the differential co-ordinates and the Laplacian operator. It is also possible to introduce some constraints upon the final mesh vertex positions by adding a second set of equations: $w_i\mathbf{q}_i = w_i\mathbf{b}_i$, where w_i is a weight, $\mathbf{q}_i$ is the vertex of mesh node i to be constrained and $\mathbf{b}_i$ is the constraint. This results in the following least-squares minimization:

$$\arg\min_{\mathbf{m}} ||L\mathbf{m} - \mathbf{k}||^2 + ||A\mathbf{m} - \mathbf{b}||^2 \tag{15}$$

where A is a matrix of constraint weights (w_i) and $\mathbf{b}$ is a vector of the mesh-node constraints.

A major question is how the predicted mesh positions should be obtained. Furukawa and Ponce [18, 19] use texture-based constraints to estimate the motions at each node of a mesh model. Their algorithm begins by estimating the rigid-body motion at each node i of a mesh. This is achieved by using a conjugate gradient

method to maximise the normalized cross-correlation score between a reference texture and a predicted texture from the images, given the motion of mesh node i. The reference texture is mapped onto the mesh faces surrounding the node i at the beginning of the sequence and remains fixed throughout the sequence.

The next step of the algorithm is to perform a global mesh deformation, based upon the local motions calculated at the first step. The global energy to be minimized consists of three terms: a data term, a motion smoothness term, and a local rigidity term:

$$E = \sum_i ||\mathbf{q}_i - \hat{\mathbf{q}}_i||^2 + \alpha|[\zeta_2 D^2 - \zeta_1 D]\mathbf{q}_i|^2 + \beta[\varepsilon(i) - \varepsilon(i^0)]^2, \tag{16}$$

where $\{\mathbf{q}_i\}_{i=1}^N$ is the set of mesh node locations to be globally optimized, $\hat{\mathbf{q}}_i$ is the mesh node location estimated at the first stage of the algorithm, $\alpha, \beta, \zeta_1, \zeta_2$ are 'tunable' parameters of the algorithm, D is the discrete Laplacian operator, D^2 is the discrete Laplacian operator applied twice, $\varepsilon(i)$ is the mean length of the edges connected to node $\mathbf{q}_i$ and $\varepsilon(i^0)$ is the mean length of the edges connected to node i at the initial frame. The final step of the algorithm is a filtering stage, which involves repetitions of the following steps: (i) detecting and removing erroneous motion estimates and (ii) re-running of the global optimization process described at the second stage. With this approach, a sequence of high-quality meshes is generated across time for sequences of around 100 frames. It is worth noting that the method seems to require highly textured scenes, such as fabrics containing fine patterns and human faces that have been painted with fine textures.

In a similar series of papers that all used Laplacian mesh deformation [5, 30], Aguiar *et al.* presented a set of methods that did not estimate the motions directly from the image textures, but instead used optical flow [14], SIFT features [13, 31], silhouettes [12], and stereo [12] constraints. These constraints may be divided into two groups: *motion* constraints and *position* constraints. Although the silhouette

and stereo positional constraints obviously help to determine the mesh positions at each step, it is questionable whether they lead to accurate motion estimation. For example, a rotating sphere or cylinder (rotating around its centre axis) will be considered to be stationary by stereo and silhouette algorithms. For this reason it is more likely that these methods are producing time-consistent meshes rather than accurate motion estimates.

6 Discussion

A wide range of approaches has been reviewed and there seems clearly to be a trade-off between performance and generality. This is borne out in figure 3 which shows the number of frames that can be tracked against the generality of each method. Table 1 shows some of the limitations imposed on the scene by each method. The optical-marker approaches offer the best performance but require the most restrictive set-up. The next best performing approach is the model based one, which has been demonstrated on sequences of up to 1000 frames, yet still requires specific knowledge of the subject to be tracked. This leaves the general scene motion estimation algorithms, which vary a great deal in terms of their form and performance; some algorithms are demonstrated for a single frame of motion, whereas others are tested on sequences containing more than 100 frames. Some detailed analysis is required to explore this discrepancy, and to this end, the *general scene* motion estimation algorithms will be examined from a Bayesian point of view. This means that we can view the algorithms according to the following factors: the form of the state model ($\mathbf{x}$), the prior knowledge that is used $p(\mathbf{x})$, the measurement likelihood $p(\mathbf{z}|\mathbf{x})$, and finally the MAP estimation method. Table 2 shows the details of each factor for the approaches reviewed in this article. The methods are ordered according to the number of frames for which they can track

sequences (without performing a reconstruction).

The form of the model is extremely important, since it determines which parameters must be estimated. There is a clear link between the form of the model and the performance and it is interesting to note that only surface based models have been demonstrated on more than a single frame of motion. The likely explanation for this trend is that only a surface model allows accurate enough data likelihood and prior models (modelling a surface with anything but a surface will be sub-optimal). Even within the surface model category, there are some differences. The mesh model is probably the most informative, since it not only describes the surface but also describes the connections between surface points. The patch-based models are probably less informative as the notion of connectivity is lost, although the surfel model of Carceroni and Kutulakos [8] is unique in that a surface lighting model is also estimated.

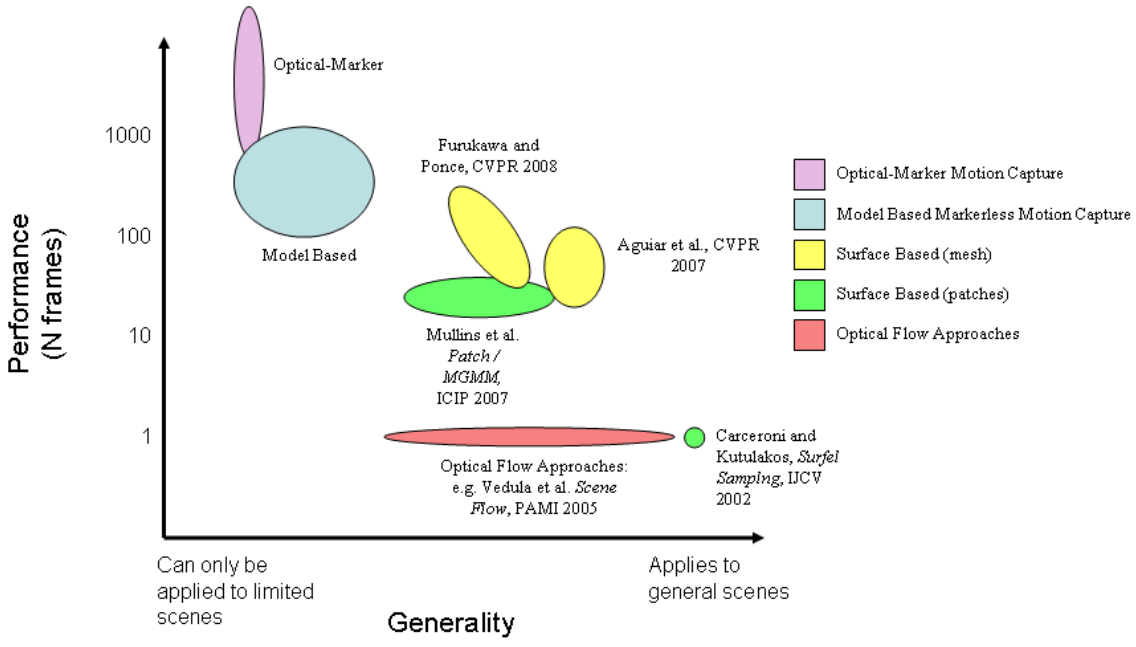

Figure 3: The trade-off between performance and generality.

Method	Authors	Limitations on scene
Optical-Marker	Many e.g [1]	Requires optical markers within scene
Model-based	Gall *et al.* [20]	Requires human skeleton model
Mesh (texture)	Furukawa and Ponce [18]	Highly textured scenes only
Mesh (optical flow)	Aguiar *et al.* [14]	Requires Laser-scan of subject
Mesh (optical flow and SIFT)	Aguiar *et al.* [13]	Requires Laser-scan of subject
Combine optical flows	Vedula *et al.* [51]	None
Surfels	Carceroni and Kutulakos [8]	None
Patches plus MGMM	Mullins *et al.* [36]	The clustered components are rigid

Table 1: Comparison of limitations placed on scene by the main tracking methods.

Authors	Form of model($\mathbf{x}$)	Data-Likelihood $p(\mathbf{z}\|\mathbf{x})$				Regularization $p(\mathbf{x})$						MAP estimation of $p(\mathbf{x}\|\mathbf{z})$				Approx N frames
		Image based texture score	Surface based texture score	Scale Invariant Features	Surface Colour Consistency	Global smoothness of image motion field	Global smoothness of scene motion field	Local smoothness of scene motion	Rigid clustered components	Laplacian mesh deformation	Constant velocity model	Variational	Linear Least Squares	Monte Carlo	Discrete MRF	
Aguiar *et al.* [13]	Mesh	X		X		X				X		X[a]	X[b]			10^2
Furukawa and Ponce [18]	Mesh		X					X		X		X				10^2
Aguiar *et al.* [14]	Mesh	X				X				X		X[a]	X[b]			10^2
Mullins *et al.* [36]	Patches plus MGMM				X			X	X					X		30
Pons *et al.* [39]	Level-Set	X					X					X				1
Wedel *et al.* [54]	Depth-map	X					X					X				1
Huguet and Devernay [26]	Depth-map	X				X						X				1
Isard and MacCormick [29]	Depth-map	X				X									X	1
Vedula *et al.* [51]	3D points	X				X						X[a]	X[b]			1
Carceroni and Kutulakos [8]	Surfels		X					X				X				1

[a] For optical flow estimates only.

[b] For mesh deformation only.

Table 2: Analysis of general motion estimation algorithms from a Bayesian perspective.

Nearly all of the reviewed methods use a texture-based data likelihood model, but there are several differences in how the model is implemented. Either an image-based or surface-based texture score can be used. The techniques which use an image-based score tend to use an optical flow algorithm for estimating the projected scene flows. The use of surface-based texture scores means a fixed texture model can be acquired at the beginning of the tracking process, and therefore can be used throughout the sequence to prevent tracking drift. However, assuming that exactly the same texture is projected into different cameras across time is problematic for several reasons, including non-Lambertian surfaces, changing lighting conditions, changing surface normal orientations, and the use of cameras which have not been photometrically calibrated. There are several strategies to improve the reliability of the data-likelihood model in the presence of these factors, including: photometric normalization of image textures; use of robust cost functions; and incorporation of an illumination model. It should be noted that the mesh-based approaches contain a significant number of 'robustness heuristics', which means that it is difficult to judge how well these techniques generalize to different datasets.

An alternative approach to texture-based scores is scale invariant feature matching. This approach is appealing, since the matching of feature descriptors is computationally inexpensive, due to the fact that relatively small numbers of features tend to be matched. Since the information obtained from feature matching is sparse, this approach only works well with a smooth motion prior, since motion estimates must be propagated into featureless regions.

In order for motions to be reliably estimated, it is important that the data likelihood model accurately reflects the real world processes. It is therefore important that visibility constraints are effectively handled. Ensuring that only the correct set of cameras is used strongly depends on the model that is used to represent the scene, and the mesh, voxel and level-set methods excel in this respect.

All methods employ some form of regularization to ensure that the problem is well-posed. Image-based regularizations are common [51, 54, 13] but are a sub-optimal approach, as smooth neighbouring pixel motions do not necessarily imply smooth surface motions. Patches are commonly used to ensure that the problem is well-posed by imposing a local smoothness prior on the solution [36, 8, 18]. However, for small patches containing no texture, it is still possible that the problem is ill-posed and a multiresolution technique is therefore often required. An alternative solution is to introduce a global regularization of the motion in the scene domain. This can be carried out in two ways: either the motion is regularized between neighbouring nodes of a mesh [13, 18] or the entire motion field is regularized [39].

It is obvious that the models discussed so far will be useless unless there is an efficient method of finding a Maximum A Posteriori (MAP) probability estimate, or its equivalent. Nearly all of the general motion estimation algorithms take a variational approach to finding the MAP estimate. Variational approaches tend to go hand-in-hand with motion estimation algorithms for the reason that, with small enough motions, there is always a close approximate solution to the problem. This is fortunate, since it means that the solution can often be found using a small number of iterations.

There are two other methods for MAP estimation. The first method is to use a discrete Markov Random Field technique such as graph-cuts or loopy belief propagation, which have been effectively used in applications such as stereo matching. The problem with these techniques is that they become computationally prohibitive as the number of dimensions increases. The second method is to use a Monte Carlo technique such as particle filtering, which is able to update a full state posterior density across time. The disadvantage is that it is difficult to incorporate global smoothness constraints, since more particles are needed as

the number of state parameters increases. The variational approach is well suited for surface tracking as it does not suffer from the disadvantages of the Markov Random Field and Monte Carlo techniques: it is efficient and a global motion smoothness prior can easily be accommodated.

Given the preceding analysis of general scene motion algorithms, it is worth noting some weaknesses in the presented methods, which will become important for the contributions to be made in this thesis [41, 40]. It seems that most methods take a deterministic approach to modelling the surface state over time, which is remarkable given the fact that in the object tracking community, a statistical model of the target state is very common [48, 28, 33, 56, 16, 6]. There are two main reasons for incorporating a probability model of the target state: the measurement noise may be accounted for; and prior statistical knowledge about the target movements can be incorporated. This insight leads to the contribution presented in chapter 4, which provides a stochastic model of the likely surface movements across time.

Another weaknesses arises when the motion smoothness constraints in the spatial domain are examined. First, the smooth motion constraint is broken under rotations, which leads to difficulties when regularizing the motion between neighbouring surface points. The other problem is that it is often assumed that each motion estimate (whether from comparing textures or matching features) on the surface model is disturbed by a uniform level of measurement noise. In other words, most authors ignore the fact that some surface textures provide more information than others. Chapter 5 presents a method that deals with both of these weaknesses: rotations are explicitly estimated at the surface and a probabilistic view of the motion estimation process is taken, to ensure that motion estimates are propagated to neighbouring surfaces in an optimal fashion.

It is possible at this stage to see two threads that will run through the whole thesis [41, 40]. First, probabilistic modelling techniques are central to the two

motion algorithms presented in this thesis. The method in chapter 4 takes account of the state uncertainties over time and incorporates a stochastic model of the likely surface movements. Chapter 5 takes a probabilistic approach to enforcing smooth surface motion. The second thread running through this thesis is the method for providing local motion estimates: a set of planar patches. This thesis therefore has similarities to the other works which also use a set of planar patches for tracking [18, 8, 36] and some differences need to be explained. Carceroni and Kutulakos [8] only estimate the motions for a set of surfels over a single frame, without using any temporal or neighbour statistical models. Furukawa and Ponce [18] estimate the rotations and translations at each node of a mesh using a local rigid texture model, but this is carried out in a deterministic fashion rather than the probabilistic methods presented in this thesis [41, 40]. Finally, Mullins *et al.* [36] use a particle filter to estimate the rigid body motion of some clustered components, which each consist of a set of planar patches. The method presented in chapter 4 also uses a particle filter, but this is applied at the patch level. The method presented in chapter 4 therefore estimates a motion for each individual patch, which is of course at a much finer level of detail than a clustered-component level. Furthermore, the method presented in chapter 5 includes a strong regularization of the motion between the individual patches, and this is much more general than the rigid body constraint provided by the Gaussian Mixture Model.

7 Summary

The aim of this article was to review the main approaches to 3D motion estimation using multiple cameras. The various methods described in the literature were grouped into four main approaches: passive optical-markers techniques; model-based approaches; optical flow approaches; and surface-based approaches. The optical-marker and model-based approaches are the most restrictive; either user intervention in the scene is required, or the scene motion model must be known beforehand. Only the optical flow and surface-based approaches are able to estimate the general motion of a scene. In order to understand the differences between these general motion estimation methods, a Bayesian perspective was taken to analyse the methods in terms of the form of the model $\mathbf{x}$, the data-likelihood model $p(\mathbf{z}|\mathbf{x})$, the prior model $p(\mathbf{x})$, and the estimation method.

This analysis led to the observation of two weaknesses of existing methods. For many of the optical flow and surface-based approaches, there is no model of the state uncertainty over time and there is no consideration of the likely surface dynamics. In the spatial domain, neighbourhood motion smoothness is commonly enforced, but this is done without any consideration of the measurement noise at each surface motion estimate. This paves the way to the contributions described in chapters 4 and 5 of this thesis [41, 40].

References

[1] www.motionanaysis.com.

[2] S. Avidan. Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1064–1072, 2004.

[3] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[4] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[5] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008.

[6] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 8(1):90–99, 1986.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision*, pages 25–36, 2004.

[8] R.L. Carceroni and K.N. Kutulakos. Multi-View Scene Capture by Surfel Sampling: From Video Streams to Non-Rigid 3D Motion, Shape and Reflectance. *International Journal of Computer Vision*, 49(2):175–214, 2002.

[9] J. Carranza, C. Theobalt, M.A. Magnor, and H.P. Seidel. Free-viewpoint video of human actors. *ACM Transactions on Graphics (TOG)*, 22(3):569–577, 2003.

[10] K. Cheung, S. Baker, and T. Kanade. Shape-From-Silhouette across time part I: theory and algorithms. *International Journal of Computer Vision*, 62(3):221–247, 2005.

[11] K. Cheung, S. Baker, and T. Kanade. Shape-From-Silhouette across time part II: applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63(3):225–245, 2005.

[12] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. In *International Conference on Computer Graphics and Interactive Techniques*. ACM New York, NY, USA, 2008.

[13] E. de Aguiar, C. Theobalt, C. Stoll, and H.P. Seidel. Marker-Less 3D Feature Tracking for Mesh-Based Human Motion Capture. *Human Motion–Understanding, Modeling, Capture and Animation*, pages 1–15, 2007.

[14] E. de Aguiar, C. Theobalt, C. Stoll, and H.P. Seidel. Markerless deformable mesh tracking for human shape and motion capture. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Minneapolis, USA, June 2007. IEEE.

[15] Q. Delamarre and O. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *IEEE International Conference on Computer Vision*, volume 2, 1999.

[16] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61(2):185–205, 2005.

[17] M. Eichner, V. Ferrari, and S. Zurich. Better appearance models for pictorial structures. *British Machine Vision Conference*, 2009.

[18] Y. Furukawa and J. Ponce. Dense 3D motion capture from synchronized video streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[19] Y. Furukawa and J. Ponce. Dense 3d motion capture for human faces. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[20] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1753, 2009.

[21] D.M. Gavrila and L.S. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[22] J. Giebel, D.M. Gavrila, and C. Schnorr. A Bayesian framework for multi-cue 3D object tracking. In *European Conference on Computer Vision*, pages 241–252. Springer, 2004.

[23] T.X. Han, H. Ning, and T.S. Huang. Efficient nonparametric belief propagation with application to articulated body tracking. In *Computer Vision and Pattern Recognition Conference*, page 221. IEEE Computer Society, 2006.

[24] C. Harris. Tracking with rigid models. In *Active vision*, page 73. MIT Press, 1993.

[25] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[26] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *International Conference on Computer Vision*, pages 1–7, 2007.

[27] B. Huhle, S. Fleck, and A. Schilling. Integrating 3d time-of-flight camera data and high resolution images for 3dtv applications. *IEEE 3DTV conference*, 2007.

[28] M. Isard and A. Blake. CONDENSATION: Conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[29] M. Isard and J. MacCormick. Dense motion and disparity estimation via loopy belief propagation. *Asian Conference on Computer Vision*, pages 32–41, 2006.

[30] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. R
"ossl, and H.P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings of IEEE Shape Modeling International*, pages 181–190, 2004.

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

[32] S. Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. *IEEE Conference on Computer Vision and Pattern Recognition*, 2, 2003.

[33] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989.

[34] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[35] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.

[36] A. Mullins, A. Bowen, R. Wilson, and N. Rajpoot. Video based rendering using surfaces patches. *3DTV Conference, 2007*, pages 1–4, 2007.

[37] R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *International Conference on Computer Vision*, volume 1, page 394, 2001.

[38] R. Plankers, P. Fua, and N. DApuzzo. Automated body modeling from video sequences. In *Proc. IEEE International Workshop on Modelling People (mPeople), IEEE Computer Society Press, Corfu, Greece*, 1999.

[39] J.P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2):179–193, 2007.

[40] T. Popham, R. Wilson, and A. Bhalerao. A smooth 6DOF motion prior for efficient 3D surface tracking. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1–4. IEEE, 2010.

[41] T.J. Popham. *Tracking 3D Surfaces Using Multiple Cameras: A Probabilistic Approach*. PhD thesis, University of Warwick, December 2010.

[42] V.A. Prisacariu and I.D. Reid. PWP3D: Real-time segmentation and tracking of 3D objects. In *British Machine Vision Conference*, 2009.

[43] B. Ristic and S. Arulampalam. *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.

[44] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[45] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[46] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical Bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1372–1384, 2006.

[47] D. Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[48] D. Terzopoulos and R. Szeliski. Tracking with Kaiman snakes. *Active vision*, pages 3–20, 1992.

[49] C. Theobalt, M.A. Magnor, H. Theisel, and H.P. Seidel. Marker-free kinematic skeleton estimation from sequences of volume data. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 57–64. ACM New York, NY, USA, 2004.

[50] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud. Temporal surface tracking using mesh evolution. *European Conference on Computer Vision*, pages 30–43, 2008.

[51] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 475–480, 2005.

[52] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH Conference*, page 97. ACM, 2008.

[53] S. Wachter and H.H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3):174–192, 1999.

[54] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. In *European Conference on Computer Vision*, pages 739–751, 2008.

[55] R. Wilson. MGMM: multiresolution Gaussian mixture models for computer vision. In *International Conference on Pattern Recognition*, volume 15, pages 212–215, 2000.

[56] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4), 2006.

[57] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. *IEEE Computer Vision and Pattern Recognition*, 0:373–380, 2009.