Hierarchical model fitting to 2D and 3D data

A. van den Hengel, A. Dick, T. Thormählen, B. Ward
School of Computer Science,
University of Adelaide,
Australia
Anton.vandenHengel@adelaide.edu.au

P. H. S. Torr
Department of Computing,
Oxford Brookes University,
United Kingdom
philiptorr@brookes.ac.uk

Abstract

We propose a method for interactively generating a model-based reconstruction of a scene from a set of images. The method facilitates the fitting of multiple object models to the data in a manner that provides the best overall fit to the image set. This requires that models are not fit independently, but rather collectively, each potentially impacting upon the fit of the other.

1. Introduction

We present a method for generating a model-based description of a scene, based on both 2D and 3D information. The 2D information consists of a set of images of the scene, obtained using a still or video camera. The 3D information is an incomplete (and possibly noisy) specification of the 3D structure of the scene. This information usually consists of a sparse set of 3D points either reconstructed from the images using a structure and motion technique [5, 7], or obtained using a laser range finder. Our goal is to recover a 3D model of the scene from this information that not only has the correct shape and appearance, but also identifies objects in the scene and their relationships. For example, in the scene shown in Figure 2, we aim to recover a model that includes the information that the scene contains a set of cubic objects resting on a ground plane.

Such an object-level interpretation of an image set has a number of advantages over approaches based on reconstructing points or other low-level features. An object-level scene description facilitates semantic interpretation and lighting calculations, for example, and enables a range of computer graphics techniques such as the insertion of computer generated characters and object removal.

The method we propose is interactive, in that it allows full user control, but it does so is a manner which requires minimal interaction. The user thus provides the minimal input required to achieve the desired accuracy of fit. Interaction is initiated by the user providing high level scene information which may include a specification of the relationship between objects. These relationships include basic geometric concepts such as ‘on top of’, ‘adjacent to’ or ‘within’.

A scene is defined as a set of models. Models are defined in terms of a vector of parameters. These parameters range from the very generic, such as the position of a model in world space, to the very specific, such as the branching frequency for a tree model. Thus, while some parameters are common to all models, others may apply only to one model. Models can have pairwise dependencies (for example, a cube can be resting on a plane, which constrains the positions of the cube and the plane). These dependencies are soft and are expressed through their parameters.

The main contribution of this paper is an algorithm for fitting models to a scene. The algorithm makes use of the varying specificity of model parameters to fit using a coarse to fine strategy. This allows an efficient search within a large volume of space, and allows the same search algorithm to be used for a wide variety of models. The algorithm also takes into account dependencies between models, to further refine the search space. It relies on some user interaction, but has been specifically designed to maximise the amount of information derived from each interaction, so that the burden placed upon a user is minimised.

The paper is organised as follows. Section 2 defines the way in which models and their dependencies are specified. Section 3 describes the likelihood functions that are used to fit the models to the 2D and 3D data. Section 4 shows how models are defined by parameters, while Sections 5 and 6 explain the process of fitting these parameters to the data. Some applications of this method are discussed and demonstrated in Sections 7 and 8.

2. Model Specification

Each model is defined by an identifying label (for example, it might be a cube or a sphere), and, by a slight abuse of

Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)
0-7695-2606-3/06 $20.00 © 2006 IEEE

notation, we identify with each instance of a model a vector of parameters sharing the same label. The form of this parameter vector depends on the model type. In the case of a cube, for example, the parameters define the cube's position, scale, and orientation in 3D space.

2.1. A hierarchical object graph

The relationships between objects are modelled using a Markov Random Field with a tree structure. This simplified network model captures pairwise inter-object relationships, without requiring complex techniques for propagating probabilities through more general graph structures. Relationships such as the fact that objects usually rest on one-another, for example, can be captured directly within this model, but more complex ideas involving groups of objects must be mediated through a common intermediary.

Relationships between models are defined in terms of their parameters. A relationship is formed between 2 models when there is a dependency between their parameter vectors. For example, if a cube is resting on a table, there is a dependency between the position of the cube and the table, and therefore they are related. If a cube is placed on top of another cube, which in turn is resting on a table, then all 3 are related, but the top cube and the table are conditionally independent given the middle cube.

2.2. The joint probability

We aim to find the set of models $\mathcal{M} = \{M_\xi\}$ where $\xi = 1 \ldots \Xi$ that are most probable given the data $\mathcal{D}$ (images and 3D points) and any prior information $\mathcal{I}$. We represent the estimation problem as a Markov Random Field with a hidden node corresponding to each object and an observed node for each (object-based) measurement. Observed nodes are linked to the corresponding model nodes, as would be expected, with the relationships between models providing the links between model nodes. The relationships between models have been constrained such that the graph has a tree structure, which simplifies the calculation of the joint probability function.

The Hammersley-Clifford theorem states that we can factorise the joint probability over the model set $\mathcal{M}$ as the (normalised) product of the individual clique potential functions[2] of the graph. The cliques in this case are all of size 2. The potential function adopted for the cliques containing an observed node and a model node is based on the probability of the model given the observation and the prior. For a model X,

$$\Pr(X|\mathcal{D}\mathcal{I}) \propto \Pr(\mathcal{D}|X\mathcal{I}) \Pr(X|\mathcal{I}) . \qquad (1)$$

It is the right hand side of this expression which forms the clique potential function.

The potential function for cliques containing only nodes representing models (models X and Y for instance) is the joint probability of the 2 models $\Pr(X, Y)$. The full joint probability of the set of models $\mathcal{M}$ given the data set $\mathcal{D}$ and the prior information $\mathcal{I}$ is thus

$$\Pr(\mathcal{M}|\mathcal{D}\mathcal{I}) = \frac{1}{Z} \prod_{M \in \mathcal{M}} \Pr(\mathcal{D}|M\mathcal{I}) \Pr(M|\mathcal{I}) \prod_{N \in \mathcal{D}_M} \Pr(M, N), \qquad (2)$$

where $\mathcal{D}_M$ represents the set of descendants of M in the tree. The descendants are chosen here rather than the full neighbourhood in order to ensure that each model-model probability is counted only once as is required under the Hammersley-Clifford Theorem.

Because the definition of a model is quite general, the method is naturally capable of fitting a range of models, and also of fitting families of models. A simple model, for example, might be a plane or sphere. More complex models might involve objects with non-parametric descriptors, or families of objects and the relationships between them.

3. Model Observations

We can partition the data into 2D and 3D feature sets $\mathcal{D}_2$ and $\mathcal{D}_3$. By assuming that these data sets are independently acquired, we can also factorise the likelihood terms as $\Pr(\mathcal{D}|M\mathcal{I}) = \Pr(\mathcal{D}_3|M\mathcal{I}) \Pr(\mathcal{D}_2|M\mathcal{I})$. This assumption may not always be justified; for example when the 3D data is generated by performing structure and motion estimation based on the same image set as that from which the 2D data is generated. However, an analytic form for the dependence between $\Pr(\mathcal{D}_3|M\mathcal{I})$ and $\Pr(\mathcal{D}_2|M\mathcal{I})$ is very difficult to derive, and we thus assume the uninformative uniform model prior in this case. It is thus always the case that

$$\Pr(\mathcal{D}|M\mathcal{I}) \propto \Pr(\mathcal{D}_3|M\mathcal{I}) \Pr(\mathcal{D}_2|M\mathcal{I}) \quad . \qquad (3)$$

As part of the definition of a particular object type we define zero or more of these likelihood functions, and the prior information $\mathcal{I}$. 3D likelihood functions define the probability of a set of model parameters given a set of 3D points, and typically favour parameters that result in many 3D points lying close to or on the model surface. 2D likelihood functions define the probability of model parameters given the images—this typically favours image edges near the projections of model edges, and incorporates any appearance information that is known about the model. We give examples of such functions in the following sections.

3.1. 3-Dimensional Likelihood Functions

We now describe one possible 3D likelihood function which is particularly suitable for point clouds occurring as a result of a bundle adjustment procedure. Given that bundle adjustment minimises the reprojection error we use a likelihood for each point is which is closely related to this error measure. Assuming that the reconstructed points are conditionally independent given the model M, the 3D component of the model likelihood can be written as $\Pr(\mathcal{D}_3|M\mathcal{I}) = \prod_{\mathbf{P}\in\mathcal{D}_3} \Pr(\mathbf{P}|M\mathcal{I})$.

We assume that the error on the observations conforms to a Gaussian distribution. Because the observations are measured in the image domain, the likelihood measurements must also be made in this domain in order to be statistically justified. Let $\mathbf{P}_M$ be the point on the surface of the model M which is closest to the reconstructed data point $\mathbf{P}$. If we label the projection of a 3D point $\mathbf{P}$ into image K as $\mathbf{p}(\mathbf{P}, K)$ then we wish to measure the distance between $\mathbf{p}(\mathbf{P}, K)$ and $\mathbf{p}(\mathbf{P}_M, K)$ in each of the images that were used in the estimation of $\mathbf{P}$. The distance in image K is $d_2(\mathbf{p}(\mathbf{P}, K), \mathbf{p}(\mathbf{P}_M, K))$ where $d_2(\cdot, \cdot)$ represents the 2D image-based distance. Not all points in the reconstruction necessarily belong to the model that is being fitted, so a Huber function [3] $h(\cdot)$ is applied to the distance measure, to diminish the influence of points far from the model. This also has the effect of segmenting the point cloud into those points belonging, and not belonging, to the object according to their position. The distance measure for a 3D point $\mathbf{P}$ thus becomes $h(d_2(\mathbf{p}(\mathbf{P}, K), \mathbf{p}(\mathbf{P}_M, K)))$.

If $\mathbf{P}$ from a set of points $\mathcal{P} = \{\mathbf{P}_i\}$ where $i = 1 \ldots n$ was originally calculated from observations in images $\mathcal{K}_{\mathbf{P}} = \{K\}$ then the negative log likelihood of $\mathcal{P}$ given a model M is

$$\mathcal{J}_3(\mathcal{P}, M) = f_3 \sum_{\mathbf{P}\in\mathcal{P}} \sum_{K\in\mathcal{K}_{\mathbf{P}}} \frac{h(d_2(\mathbf{p}(\mathbf{P}, K), \mathbf{p}(\mathbf{P}_M, K)))}{\|\mathcal{K}_{\mathbf{P}}\|}. \tag{4}$$

where f_3 is a constant scale factor.

3.2. 2-Dimensional Likelihood Functions

One possible 2-dimensional likelihood is that based on the assumption that edges in the model will give rise to intensity gradients in the image. Edges have a number of advantages over corners or other features that might be used to guide model fitting. These advantages have been well discussed in the tracking literature (see [6] for example) but include rapid detection and relative robustness to changes in lighting.

In order to calculate the degree to which a hypothesised model is supported by the image intensities the visible edges are projected back into the image set and a measure is taken of the corresponding intensity gradients. The measure is the same as that described in [8] and similar to that used in [6] amongst others.

Some models, however, do not contain geometry with prominent edges. In this case a different 2D likelihood is defined as part of the model. This likelihood is based on the assumption that the surface of the plane is largely unoccluded by objects not modelled and that it is a Lambertian surface and will therefore have the same appearance in each image. The projections of a point on the surface into each image are related by homographies, which can be calculated analytically from the camera projection matrices and the plane parameters (for example, see [1]). The likelihood of each point on the surface of a hypothesised plane model is therefore defined by the variance of pixel values at the projection of that point into each image in which it is visible. More details are available in [8].

3.3. Relations between models

Model parameters from separate models can be linked together to express dependencies between them. For example, a sphere lying on a table, then the position parameters $\mathbf{T}$ of the sphere and table plane are linked. Rather than simply forcing these parameters to be the same (or have a fixed difference) we link them probabilistically, through the term $\Pr(M, N)$ in Equation 2.

4. Model parameters

In general, the definition of any model includes a position $\mathbf{T}$ and a scale S. For a simple model such as a sphere, this may be all that is required. However most models will also contain other parameters specifying their orientation, elongation and other relevant geometric properties. We can create a hierarchy of models according to their parameters, a child model inheriting the parameters of its parent, and adding extra parameters specific to it. This allows us to formulate a general strategy for fitting models to data, as will be described later.

4.1. Example: Cube Model

As well as a position $\mathbf{T}$, which is the position of one corner of the cube, and a scale S which is the side length of the cube, the cube model has 3 orientation parameters. These parameters are used to derive two orthogonal unit vectors $\mathbf{U}$ and $\mathbf{V}$ that intersect at $\mathbf{T}$, and a normal vector $\mathbf{N} = \mathbf{U} \times \mathbf{V}$. The cube is then defined by 6 faces bounded by the vertices $\mathbf{T}$, $\mathbf{T} + S\mathbf{U}$, $\mathbf{T} + S\mathbf{U} + S\mathbf{V}$, $\mathbf{T} + S\mathbf{V}$, $\mathbf{T} + S\mathbf{N}$, $\mathbf{T} + S\mathbf{N} + S\mathbf{U}$, $\mathbf{T} + S\mathbf{N} + S\mathbf{U} + S\mathbf{V}$, and $\mathbf{T} + S\mathbf{N} + S\mathbf{V}$.

The 3D likelihood of the cube is calculated as described in Section 3, by summing the Huber distances between each data point and the point on the model surface that is closest to it. The 2D likelihood is also calculated as in Section 3, by matching model edges to image gradients. Other regular polyhedra, such as a sphere and tetrahedron, can be defined, and their likelihoods computed, in a similar way.

4.2. Example: The Bounded Plane Model

The bounded plane has a position $\mathbf{T}$ which is a point on the boundary of the plane. The plane is further defined by two orthogonal unit vectors $\mathbf{U}$ and $\mathbf{V}$ that intersect at $\mathbf{T}$ and belong to the plane. The scale S has a different meaning depending on the shape of the plane. If it is a general shape, there are two scale factors $S_{i,u}$ and $S_{i,v}$ for each point $\mathbf{P}_i$ on the boundary. The boundary of the plane is defined as a sequence of points. The points are defined in terms of the plane position and scale, and the vectors $\mathbf{U}$ and $\mathbf{V}$: $\mathbf{P} = \mathbf{T} + S_{i,u}\mathbf{U} + S_{i,v}\mathbf{V}$. If it is a more regular shape, such as a square, the scale factors are shared between boundary points. In the case of a square there is only a single scale factor S which defines each of the 4 boundary points ($\mathbf{T}$, $\mathbf{T} + S\mathbf{U}$, $\mathbf{T} + S\mathbf{U} + S\mathbf{V}$, and $\mathbf{T} + S\mathbf{V}$).

The 3D likelihood of the plane is defined as for the cube model, by finding the sum of Huber distances from each point to the nearest point on the model. The 2D likelihood is based on the photoconsistency measure, due to the lack of edges expected in this model.

5. Model fitting

Having defined the model representation, and the associated likelihood functions, we now describe an algorithm for fitting such models to image data. It is not feasible to generate and evaluate a set of samples that would effectively explore $\Pr(\mathcal{D}|M\mathcal{I})$. Instead we use a coarse-to-fine strategy which exploits the nature of the functions $\Pr(\mathcal{D}_3|M\mathcal{I})$ and $\Pr(\mathcal{D}_2|M\mathcal{I})$ in order to guide our search for a suitable model. The function $\Pr(\mathcal{D}_3|M\mathcal{I})$ relates the model to a set of reconstructed points and is well suited to gross localisation of the object in the scene, due to the relatively smooth nature of the associated probability distribution. The function $\Pr(\mathcal{D}_2|M\mathcal{I})$ relates the model to the appearance of the object in the image set, and is typically only applicable when the model is very close to the true location of the object. When this criterion is satisfied, however, it can achieve very precise localisation, as the associated probability distribution is typically strongly peaked. Thus the 3D likelihood function is better suited to initial localisation, while the 2D likelihood is appropriate for further optimisation based on this initial estimate.

As described earlier, the definition of any model includes a position $\mathbf{T}$ and a scale S. All but the simplest models require more parameters; however even more complex models can be approximated by their position (e.g their centroid) and their size. When searching for a model in a 3D point cloud, a rough approximation of the likelihood of the model being of a certain location and size can be obtained by counting the 3D features that occur within a volume of space centred at $\mathbf{T}$ and with radius S. This process could equally be seen as that of eliminating areas in which further sampling for objects would be futile. In this sense the search process is preemptive, as a search for a model with many parameters (position, scale, orientation, aspect ratio, etc.) is carried out as a series of searches. Each search in the series is over an increasingly large numbers of parameters, later searches being more specific to the model type and refining results.

Given a region of space in which to search (the determination of this region is described later), we sample a range of positions from it. At each position, we search within a spherical region of space centred at that position. The search range is determined by the maximum extent expected of any model. Within this search space we calculate a histogram of radii of the point cloud. This forms a shape profile that can characterise a shape and be matched against profiles for known shape templates, such as cubes, pyramids etc. These profiles are not exact, but they are independent of shape orientation even for highly unsymmetrical shapes. They are therefore a useful first cut detector for a model. For example, the profile for a sphere is a single sharp peak (ideally a Delta function) at the scale equivalent to the sphere's radius. A cube is also a single peak but with a gradual dropoff. A cuboid (a box with different length along each axis) has 3 peaks. A planar surface has a profile that is a straight line. The closer the position is to the centre of the shape in the data, the more closely its profile will resemble the model profile.

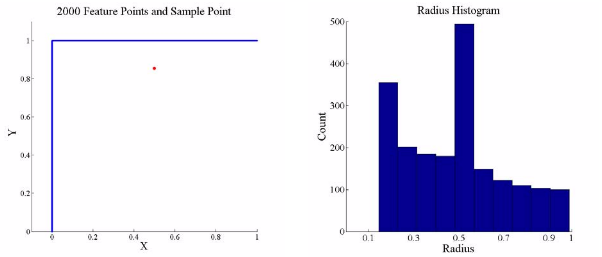

Figure 1. Profile of the number of points on a cube surface belonging to a region centred about a point in space (shown in red).

The result of this process is a set of samples, each at different locations, and each weighted according to how likely it is to correspond to a model of the type sought. The weighting is computed by comparing the radius histogram to a pre-computed (and appropriately scaled) profile for that model type, using the Kullback-Leibler divergence[4]. Samples with low weights can be eliminated from the search, and subsequent searches on the remaining model parameters applied only in areas of space near samples with high likelihood.

5.1. User Interaction

The fitting process is initialised by the user choosing an object type from the set of available models and selecting a point on one of the images. The point selected specifies a ray through the scene which is taken to intersect the desired object. A set of object hypotheses is generated on the basis of this ray and the scene information arising from the structure from motion process. At each sample point the inclusion of 3D points is tested for a set of spheres of increasing radii, in order to construct the shape profile as described in the previous section.

The search region need not be set interactively. Given a set of known models, and given the relationships between them, the search region for further models is defined by the parameters of those models already in the scene. For example, a plane in the scene defines a planar region in which other shapes are likely to occur. Given a plane, the volume of space immediately above it is automatically searched, for objects that lie on top of the plane. Given a cube, the planar regions aligned with sides of the cube are searched for further objects. Thus by interactively selecting one object, many objects whose positions are linked to that of the original object can be located.

In practice, these modes of specifying a search region are interleaved. The search region for the first shape to be modelled is specified interactively. Its position (and other parameters) are used to define further search regions for objects in the scene related to the initial shape. For instance, a ball is likely to be resting on a surface, so planes are sought that are tangential to a ball once it has been located.

6. Example: Fitting Cubes on a Table

In order to explain the operation of the method we now describe the process of fitting a set of cube models resting on a common plane model to the video shown in Figure 2.

6.1. Generating Initial Hypotheses

The cube profile function $T(\mathcal{P}, C, r)$ is used to calculate the likelihood of a set of points $\mathcal{P}$, forming a cube with centre C and radius (half side length) r, integrated over all cube orientations. Possible cube locations are sampled regularly within the search region. At each location, the profile function is evaluated for varying scale as described in Section 5. This profile is then compared with the precomputed cube profile by computing the KL divergence between the data and model histograms.

It is assumed that the object sought will fill at least 1% and less than 100% of the image used to identify it. This forms part of the cube model prior $\Pr(\mathcal{I})$, and provides a constraint upon the range of scales that should be evaluated at each point in the search region. Due to the effects of perspective, the range of scales increases for points in the search space that are further from the camera. The distance between template centres increases with the calculated radius, and thus also with the distance to the camera. The function $T(\mathcal{P}, C, r)$ is evaluated for each template and the parameter vectors corresponding to function evaluations above the 90th percentile retained. These parameter vectors are used to initialise the optimisation process.

6.2. Refining Hypotheses

Each initial parameter vector specifies the position and size of a hypothesised cube. This information initialises an iterative search for each cube's orientation based upon the likelihood function $\mathcal{J}_3(\mathcal{P}, \mathbf{C})$ specified in equation (4). The orientation of each cube is initially aligned with the camera coordinate system. A Levenberg Marquardt minimisation process is carried out on the cost function $\mathcal{J}_3(\mathcal{P}, \mathbf{C})$. The result of this minimisation is a parameter vector describing the location, radius, and orientation of a cube hypothesis. One such parameter vector is recovered for each initialisation. These vectors are checked to ensure that they are significantly different from each other and that they intersect the ray specified by the user. They may be interpreted as the identifying the local modes of the probability density function associated with $\Pr(\mathcal{D}_3|M\mathcal{I}) \Pr(\mathcal{I})$.

Having explored $\Pr(\mathcal{D}_3|M\mathcal{I})$ we now incorporate $\Pr(\mathcal{D}_2|M\mathcal{I})$ in order to find the modes of $\Pr(D|M\mathcal{I})$. The 2D data likelihood of the model is described in Section 4.1. Recall that this cost function is based on the image distance between the projected edge of the model and the local intensity gradient maximum normal to the edge, summed across multiple points along each edge.

The 2D and 3D likelihood functions can now be combined to generate a complete data likelihood function. Because they are both log likelihoods, they are combined by addition; however because they are not normalised a scale factor is required to ensure that they each contribute appropriately to the final likelihood. As the 2D data likelihood is more sensitive to small changes in the cube parameters, it typically dominates this final optimisation stage.

6.3. Using Existing Information

Having generated a most likely cube hypothesis the system uses this information to find the plane. There are no reconstructed points associated with the ground plane. The only applicable likelihood is that described in Section 3.2 which is based on the assumption that the estimated camera parameters can be used to map the plane texture from one image to another. This, by itself, would not be enough to guide the fitting process, but the joint probability describing the relationship between the plane and the cube solves this problem. After sampling and optimising equation (2) for the cube-plane graph we have an estimate for the parameters of both. The plane estimate can then be used to guide the sampling process used to search for subsequent cubes as described in Section 5. The identified cubes are then optimised individually, and incorporated into the graph. The final joint is then maximised numerically over the parameters of all of the models.

7. Results

Figure 2 shows images of a set of cubes resting on a table. A semantic model for the scene has been generated by the method on the basis of only a single user mouse click on one of the cubes. Having a semantic model of the scene means that it can be manipulated in a straightforward manner. For example, the second row of Figure 2 is the result of a command to turn all cubes in the scene into Rubik's cubes. Because we know exactly where the cubes are, this can be done automatically. We also have an accurate geometric representation of the scene, which means that we can model physical interactions. In row 3 of Figure 2, a set of synthetic bouncing balls is dropped onto the scene, and interacts convincingly with the cube and the table top.

8. Conclusion

This paper has presented a method for the recovery of semantic and geometric structure of a scene, given images of the scene, a corresponding cloud of 3D points, and some semantic information provided interactively by the user. The interactive aspect of the system plays to the strength of a human observer at discerning overall content of a scene, and that of machine vision at accurately computing detailed scene structure.

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 434–441, 1998.

[2] J. Besag. Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 32(2):192–236, 1974.

[3] P. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:73–101, 1964.

[4] J. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

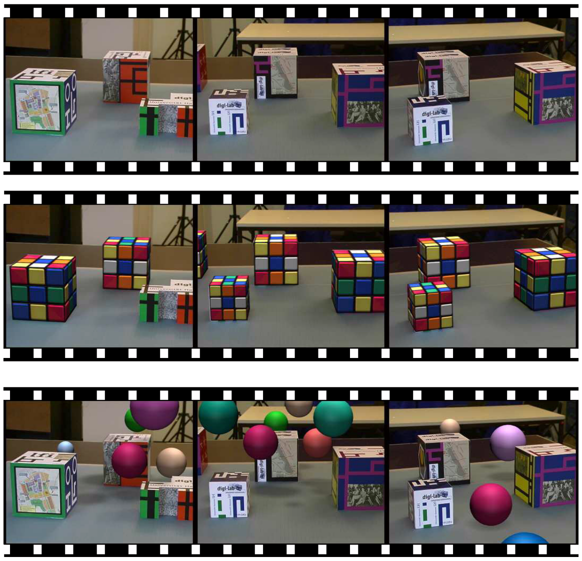

Figure 2. Reconstructed scene containing cubes on a tabletop. First row: original images of the scene. Second row: After modelling the scene, we can convert all cubes to Rubik's cubes. Third row: insertion of synthetic objects into the scene.

[5] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a handheld camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[6] P. Smith, T. Drummond, and R. Cipolla. Motion segmentation by tracking edge information over multiple frames. In *Proc. Sixth European Conference on Computer Vision*, pages 396–410, 2000.

[7] T. Thormählen. *Robust estimation of camera motion from image sequences*. PhD thesis, University of Hannover, 2006.

[8] A. van den Hengel, A. Dick, T. Thormählen, P. H. S. Torr, and B. Ward. Fitting multiple models to multiple images with minimal user interaction. In *Proceedings of the International Workshop on the Representation and use of Prior Knowledge in Vision (WRUPKV), in conjunction with ECCV'06*. (to appear), May 2006.