Proc. SPIE Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision, D.P. Casasent and E.L. Hall, Eds., Vol. 4572, pp. 165-173, 2001.

3D Object Recognition Based on Hierarchical Eigen-Shapes and Bayesian Inference

Timo Kostiainen, Ilkka Kalliomäki, Toni Tamminen, Jouko Lampinen

Laboratory of Computational Engineering
Helsinki University of Technology,
P.O.Box 9400, FIN-02015
ESPOO, FINLAND

ABSTRACT

We present results of using Bayesian inference for recovering the 3-D shape and texture of an object based on information extracted from a single 2-D image. We are using a number of different models for specific object classes. The goal is to combine the classes to a hierarchical structure. Instead of searching for the most probable explanation we estimate the entire posterior distribution of the model parameters using Markov chain Monte Carlo methods. The evaluation of model fit is based on combining edge information with intensity difference between the model and the target image.

Keywords: 3-D object representation, Bayesian inference, eigen-shapes

1. INTRODUCTION

Our goal is to be able to recognize 3-D objects in a 2-D scene using Bayesian inference. We start from a number of basic shapes suitable for the representation of certain classes of objects and we envision modeling a wider range of object classes as linear combinations of these basic shapes. This leads to a hierarchical structure for the object models and it could be extended further by representing composite objects on a yet higher level. On all levels of the hierarchy, prior distributions are associated with parameters which control the models. The prior distributions determine the probability distributions of the objects within a class or, at a higher level, the probabilities of different classes.

In this paper we present results of using Bayesian inference in the recognition of 3-D objects from a single 2-D image. The inference of shape based on one 2-D image is basically an ill-posed problem; there are infinitely many solutions. Therefore the set of possible shapes must be constrained one way or another. In practical applications, the objects to be recognized can often be assumed to belong to a known set of distinctive classes. The Bayesian approach allows the representation of such classes as prior information in terms of probability distributions. In Bayesian inference, observations are combined with the prior information to find out the probability distribution of different explanations.

We have experimented with different ways of representing 3-D objects. We do not yet have a model of any object class for which we can define a prior parameter distribution that would realistically represent the object class, so all the advantages of the Bayesian approach are not present. However, the results that we have obtained using manually crafted models indicate that Bayesian inference is well suited to this problem.

The practical implementation of Bayesian inference requires that we specify an error function or likelihood function which determines how well given 3-D model parameter values fit to the observed image. Typically, such a function cannot be integrated over analytically, so we use Markov chain Monte Carlo (MCMC) methods to pick samples from the distributions.

Further author information: (Send correspondence to T. Kostiainen)
T. Kostiainen: E-mail: Timo.Kostiainen@hut.fi, I. Kalliomäki: E-mail: Ilkka.Kalliomaki@hut.fi
T. Tamminen: E-mail: Toni.Tamminen@hut.fi, J. Lampinen: E-mail: Jouko.Lampinen@hut.fi

2. BAYESIAN INFERENCE

Bayesian inference is based on defining the probability space for all events (joint probability of all observed and unobserved variables) and inferring the conditional distribution of the variables of interest, given the observations. The idea is to estimate the entire distribution of those variables, instead of just points estimates. Various statistics, such as mean and modes, can be computed from the distribution.

In object recognition, an image I is observed and there is a model for the process that is assumed to have produced that image. If we denote the parameters of this process by θ, the quantity that we are interested is $P(\theta|I)$, the probability distribution of process parameters given the observed image. In other words, we want to find out what can be said about θ based on the observation I. This quantity is given by Bayes' rule as follows:

$$P(\theta|I) = \frac{P(I|\theta)P(\theta)}{P(I)} = \frac{P(I|\theta)P(\theta)}{\int P(I|\theta)P(\theta)d\theta}, \tag{1}$$

where $P(I|\theta)$ is the likelihood that the process θ has produced the observed image I and $P(\theta)$ is the prior distribution of θ, that is, all the information we have of θ prior to observing the image. In the recognition of a 3-D model, the parameters θ would include those which control the shape and texture of the object, as well as parameters of the camera and lighting conditions.

2.1. Markov chain Monte Carlo sampling

In practice it is rarely possible to compute the high dimensional integrals which are needed to evaluate the posterior distribution. Monte Carlo integration is a stochastic technique for the approximation of integrals by drawing random samples from a distribution and evaluating the integrand in those sample points. The idea is that the samples should be concentrated on areas of interest with regard to the values of the integrand, such as peaks. In Bayesian data analysis, the target function is the posterior distribution $p(\theta|X)$.

In the context of Monte Carlo integration, Markov chains are used to simulate random walk in the space that converges to the posterior distribution $p(\theta|X)$. In a Markov chain, each state θ^t depends only on the previous state, and the consecutive states are related by a transition distribution $T_t(\theta^t|\theta^{t-1})$. Markov chain simulation provides efficient means of producing samples from arbitrary distributions. Two common methods, which were used in this paper, for the construction of the associated transition distributions are reviewed below.

2.1.1. Metropolis algorithm

The Metropolis algorithm produces a random sequence which converges to a given target distribution $p(\theta|X)$. The Metropolis algorithm samples candidate points θ^* from a so called jumping distribution $J_t(\theta^*|\theta^{t-1})$, which is required to be symmetrical, that is, $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$ for any θ_a, θ_b, and t. The sample θ^* is accepted with probability $r = p(\theta^*|X)/p(\theta^{t-1}|X)$. The jumping distribution J_t has to be able to eventually reach all states with a finite probability.[1]

2.1.2. Gibbs sampling

The Gibbs sampler picks samples from conditional distributions of the target distribution $p(\theta|X)$. Each iteration cycles through all the components of θ in random order. At each step in the cycle, the new state θ^t is the previous state θ^{t-1} with the component $\theta^{t-1}_{\{j\}}$ substituted with a sample from the conditional distribution $p(\theta_{\{j\}}|\theta^{t-1}_{\{1\dots d\setminus j\}}, X)$. So the components of θ are updated one at a time. Since the samples are taken directly from the conditional posterior density, all samples are always accepted. In order for Gibbs sampling to work, sampling from all conditional distributions of $p(\theta|X)$ must be possible.

3. EXPERIMENTS OF MCMC SAMPLING OF 3-D OBJECTS

In this section we discuss ways of representing 3-D objects and the results which we have obtained in estimating model parameters using Metropolis and Gibbs sampling, based on different image information.

3.1. 3-D object representation

Choosing a scheme for representing 3-D objects is anything but straightforward. Psychological studies do not offer much help, as there is no consensus on how human beings represent objects. There is evidence to support both view based and model based representations.[2] The view based approach is very effective in 2-D image recognition, but it is difficult to implement the interpolation of different views, which is essential in 3-D modeling. Model-based approaches are well suited to computer graphics and the rendering of images from 3-D models can be performed very quickly. Another reason to prefer the model-based approach is the fact that it provides a setting for the understanding the scene in the image as consisting of 3-D objects.

Models for 3-D objects can be constructed in very different ways. One extreme would be to have one universal, highly flexible model to describe all the different objects. The number of free parameters in such a model would be very high and objects, once recognized, could be classified based on the parameter values. Another extreme is to have a great number of specific models for different strictly defined object classes. In this case, the models can be very restricted and only a few parameters are needed. If the correct class is not known beforehand, classification must be done in a separate step, or many different models must be fitted to see which of them are most likely solutions. The universal model approach appears conceptually elegant but is very challenging in practice. It is not easy to see how such a model should be represented mathematically. It could be a set of vertices whose order if fixed but which are otherwise very free to move. A covariance model for the vertex locations could be defined, perhaps based on samples, but it still would be difficult to represent, say, both smooth and rectangular objects in an effective way. Inclusion of new object types could be done simply by introducing samples of a new object class, whereas in the latter approach this would require choosing a new representation for the new class.

The fundamental problem is that there are more degrees of freedom in the model than in the image and therefore the parameter space must be constrained. Many man-made objects are shaped as rectangular solids, or combinations of them. It seems extremely cumbersome to attempt to model those as arbitrary shapes with a certain correlation matrix. This line of reasoning leads us to favor the approach where different classes are modeled with reasonably specialized models. Since our goal is to be able to represent a wide variety of different objects, we would still like to find a middle ground between the two extremes. One possible way to accomplish this would be to represent all objects as linear combinations of basic shapes.[3] These basic shapes would be specific enough for the accurate modeling of various object classes, and for such known object class the weights of other basic shapes should be zero. The prior distribution of these weights should be carefully defined such that known object classes would have high prior probabilities, so strong evidence from the observed image is needed to divert from these known shapes. This leads to a hierarchical structure for the representation of all possible objects. Principal component analysis could be applied to eliminate redundancies in the set of basic shapes. Addition of a new object to such a model could then be accomplished either by defining a new basic shape model or automatically as a combination of other basic shapes if the existing models are sufficiently versatile. The latter only involves a change in the prior distribution of parameters on the higher level of the hierarchy.

Another important thing to consider is the representation of composite objects. In our view, composite objects should be modeled according to the principle that the average code length needed to represent any object should be minimal. This leads to the division of composite objects to relatively small parts which can be modeled in terms of a small number of parameters – the basic shapes. The composite object would thus appear on the highest level of our object hierarchy.

Here we discuss experiments on MCMC sampling with a small number of different object models. Dealing with composite objects remains a future challenge.

3.1.1. Eigen-shapes

Principal component analysis (PCA) is a popular technique for dimensionality reduction. It has been used with considerable success in both 2-D[3] and 3-D[4,5] modeling of human faces (or heads). Principal component analysis produces orthogonal eigenvectors which explain the variance of the data most effectively. The eigenvectors can be ordered according to respective eigenvalues, which tell the amount of variance explained by each eigenvector. If the sample set is a good representation of the object class, then the first few eigenvectors (with greatest eigenvalues) make up a good linear model of the class. This model can be treated as a Gaussian distribution, whose mean is given by the eigenvectors and variance by the eigenvalues. Using only part of the eigenvectors causes the model to be constrained to a subspace and the residual must be taken into account in the model for image residual. The

Gaussian model makes it possible to compute the likelihood that a sample belongs to the object class that this model describes. This is a highly favorable property from the point of view Bayesian analysis, as a means of expressing the the prior distribution of the object class. Another advantage is the orthogonality of the eigenvectors, which enables model fitting one component at a time.

In image analysis, PCA is dependent on a large number of accurate sample images. In the 3-D case, the samples can be obtained for example using a laser or MRI scans or a stereo imaging system. If the the model class is very restricted, such as the human face, for example, the quality of PCA reconstructions is limited only by the quality and number of sample images.

3.1.2. Generalized cylinders

One of the models we have experimented with describes generalized cylinders. Their median axis is the linear combination of spline curves and the cross section is a circle or ellipse. The linear basis for the median axis is created from B-spline approximations to random curves. The random curves are generated from a Markov model, such that each increment to the curve is correlated with the previous increment. The median axis is a parametric curve in the sense that $x = x(z)$ and $y = y(z)$. The x and y curves share the same basis but have different parameters. One of the camera angle parameters becomes redundant because the object can be rotated by changing the scaling of the x and y curves. The radius can share the same model with x and y curves, although it is favorable that the radius is always positive. This model lends itself well to the representation of various man-made objects, especially if the definition of the cross-section is changed to allow rectangular shape.

3.2. Integration of different types of image information

There are various different types of information that can be extracted from a scene. Typical features of interest in image processing are edges, homogeneous regions and fiducial points or areas which match a particular template or filter. There exist good methods for extracting different information including such features but there is no obvious way to analyze the combination. Apparently, the whole is much greater than the sum of its parts.

We have considered how to take advantage of edge information in the fitting of a 3-D model to an image. In theory, Bayesian inference provides means of combining different sources of information. If we have two observations of the same image, I_A and I_B, we can compute the joint posterior distribution as $P(\theta|I_A, I_B) = P(\theta|I_A)P(\theta|I_B)$. In practice this is no simple task. We cannot express any such distributions in closed form; instead, we have to represent them as samples, obtained using MCMC methods. Attempts to compute products of high dimensional distributions represented this way will yield very inaccurate results. Another problem is that the less information we use in the sampling, the wider and more complicated will the distributions be. This means that the number of samples required will be greater, too. Another problem is that there is no obvious answer to how different types of information should be transformed into probabilities. As a result, in practice we must resort to combining different error measures to a single likelihood value using ad hoc methods.

If we know the locations of a number of likely edges in an image, we have no effective way of finding out the parameter distribution of an object, given the information that those edges (better yet, some of those edges) are part of the object. The difficulty is in the complicated relationship between the model parameters and the corresponding edge locations. This causes the parameters to have strong mutual nonlinear correlations. The edges can be computed for given model parameter values, but the inverse relationship is intractable in anything but the simplest cases. Even for very simple models the possibility of rotation ensures that each edge location can be dependent on every single model parameter. That is why in the general case, the conditional model parameter distribution given the edge locations cannot be computed or sampled from. This means that Gibbs sampling, where samples are drawn from the conditional distributions, is not likely to work well in this case. Specific strategies for computing the conditional distributions can be designed for some simple object models, but they do not really generalize to other models.

Knowledge of the location of one fiducial point can easily be exploited in the fitting procedure by making sure that the same point in the model is projected to that location, for example using that information to define a tight prior distribution for the coordinates of the point in the Bayesian model fitting approach. Then the distribution of the rest of the parameters will be conditioned on that information. That is easy because only the translation parameters are involved. An example is shown in Fig. 1*d*. To include another parameter in the model is much more difficult because rotation makes all parameters correlated.

3.3. Fitting a rectangular solid using only edge information

Our wire-frame model of a rectangular solid has seven free parameters: edge lengths, coordinates of 2-D projection and camera angles. We use the Metropolis algorithm to estimate the position of the object in an image using edge information.

The likelihood value for a given position is computed as the product of pixelwise Gabor filter responses along the wire-frame. We use filters of eight different orientations and the nearest orientation is used for each segment of the wireframe model. We have used a gamma distribution as the prior for the length of one of the edges, to ensure that all edge lengths cannot go to zero. The camera angles have uniform prior distributions, and the rest of the parameters have Gaussian priors.

Results obtained with a test image of a relatively cluttered scene are illustrated in Fig. 1. Different initializations were used for the Metropolis chains. Two positions where most of the chains converged with small error are shown in image *a)*. The sketches represent medians of the parameter distributions obtained from two sample chains. Variance of the distributions was very small in this case. The cardboard box cannot be matched more accurately because the wire-frame model does not take perspective deformation into account. Images *b)* and *c)* display typical values from chains which failed to converge in the course of a few hundred samples.

In image *d)*, the location of the strongest edge observation was used as prior information of the location of one of the edges of the object. This was implemented by assigning the edge center a Gaussian prior distribution, centered on where the strongest edge filter response was obtained. In this case all the Metropolis chains converged to the cardboard box very quickly. We found that use of such prior information usually leads to savings in computation time, even if there are many different candidates for the edge location.

3.4. Estimation of shape using the generalized cylinder model

In the estimation of 3-D shape we use a likelihood function which includes two different error measures:

$$P(I|\theta) = \exp\left(-\frac{1}{\sigma^2}\left(\beta E_{edge} + \sum \|I_{ref} - I_m(\theta)\|^2\right)\right). \qquad (2)$$

E_{edge} is a measure of edge match and the other error term is the pixel-wise intensity difference between the target image I_{ref} and the rendered model I_m. σ denotes the error variance and β is a heuristically chosen constant. Since at this stage we do not attempt to estimate the texture of the object, the edge error term is allowed to dominate and the intensity error is basically only used to separate the object from the background. We compute the edge error from smoothed edge filter responses at vertices which lie on the edge of the rendered model. Whether the end points of the cylindrical object are visible or not can be computed using knowledge of the tangent vector of the model axis at the end points and the location of the camera. The edge vertices on the side of the model will be connected to triangles which are approximately perpendicular to the camera angle. In the example illustrated in Fig. 2, we used Metropolis sampling to estimate 11 parameters: eight for the model, three for the camera angle and scale and two for the alignment of the target image with the image of the rendered model (translation). There were no local minima because the target object was well separated from the background. Since we did not use a prior which would favor symmetric or linear objects the distribution of the side axis profile is wider than that of the front profile. This means that there is more ambiguity in the side view of the object than the front view.

3.5. Texture estimation

Texture has a decisive effect on the appearance of the 3-D object. Texture accounts for details that are not explained by the model geometry and for changes in the surface color. Ideally the estimation of texture should be done jointly with the rest of the model parameters. This requires a relatively constrained model for the texture (or the object) because, as an extreme example, any scene could be explained as a flat object where the texture accounts for every detail, like a photograph. Actually this is a good explanation and it is one mode of the posterior distribution. A realistic model for texture is difficult to define and therefore we have kept texture separate from the shape estimation. We first estimate the position, shape and orientation of the object and then, keeping those parameters constant and assuming a suitable lighting model, estimate the texture of the object.

Texture coordinates which establish the correspondence between the model geometry and the texture map are computed using projective texturing. The same 3-D-to-2D projective transformation that transforms the 3-D coordinates of the model to 2-D points on the screen is used to compute 2-D texture coordinates associated with each

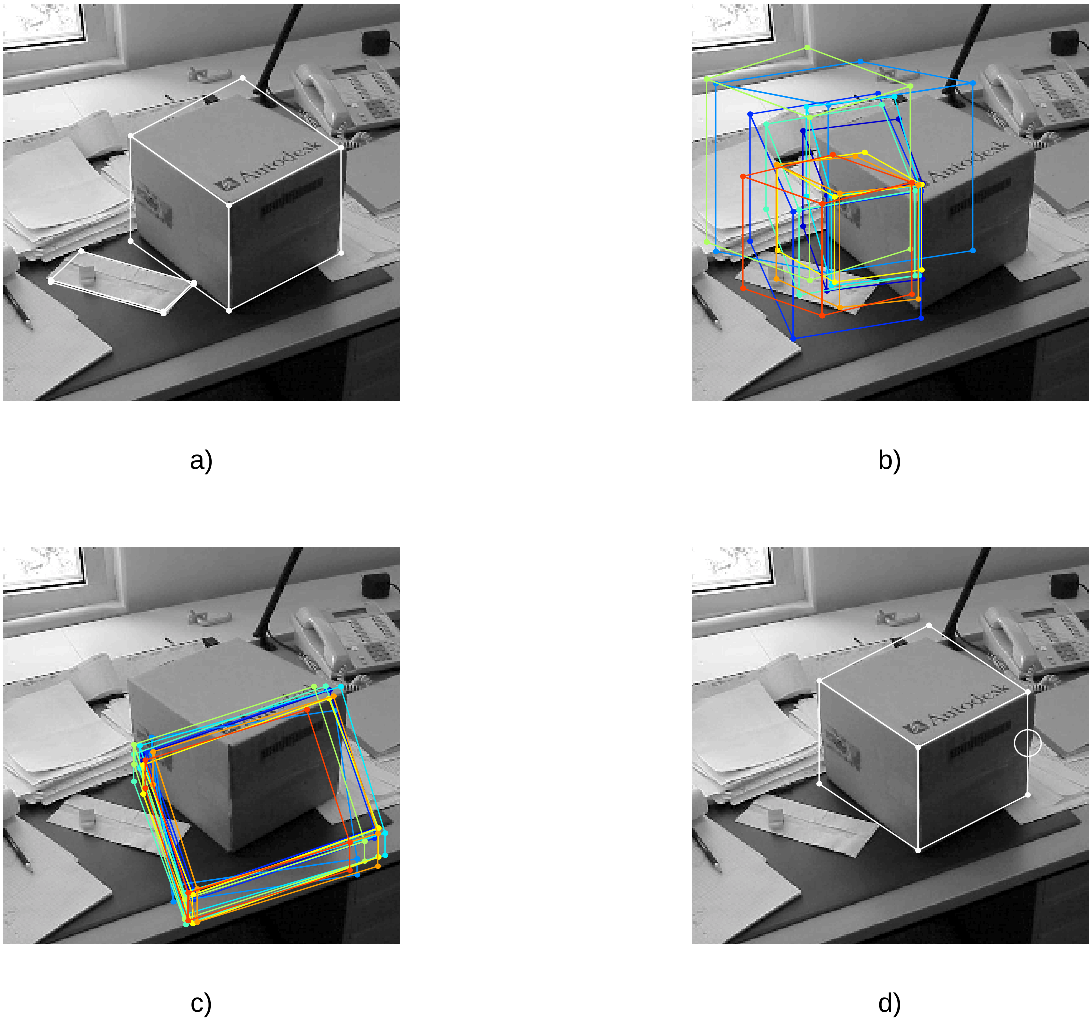

Figure 1. Estimating parameters of a rectangular solid based on edge information. *a)* Two modes of the distribution with small error. *b)* and *c)* Typical samples of MCMC chains which did not converge well and error remained high. *d)* Median of distribution obtained when using the information that the middle of one edge is located near the area marked by the circle.

3-D triangle in the model. Projective texture mapping can be performed efficiently using current graphics hardware. The approach is similar to,[6] but only a single texture is used on the object.

Lighting is computed using a simplified Phong lighting model[7] with four parameters, including k_a, k_d, k_s which control the contribution from ambient, diffuse and specular terms and ν which controls the size of specular highlights. The color value of a triangle after lighting calculation is determined by

$$C_L = k_a + k_d(\vec{N} \cdot \vec{L}) + k_s(\vec{R} \cdot \vec{V})^{\nu}, \tag{3}$$

where $\vec{N}$ is the surface normal, $\vec{L}$ is direction of illumination, $\vec{R} = 2\vec{N}(\vec{N} \cdot \vec{L}) - \vec{L}$ is the direction of reflection, and $\vec{V}$ is the view direction.

The color value of a pixel is generated as a product of the lighting and texture values, $C = C_L C_T$. Texturing is implemented in OpenGL graphics library using the GL_MODULATE texturing function, and the multiplications are done rapidly with the graphics hardware. Given the perceived image I and estimates of the light direction and

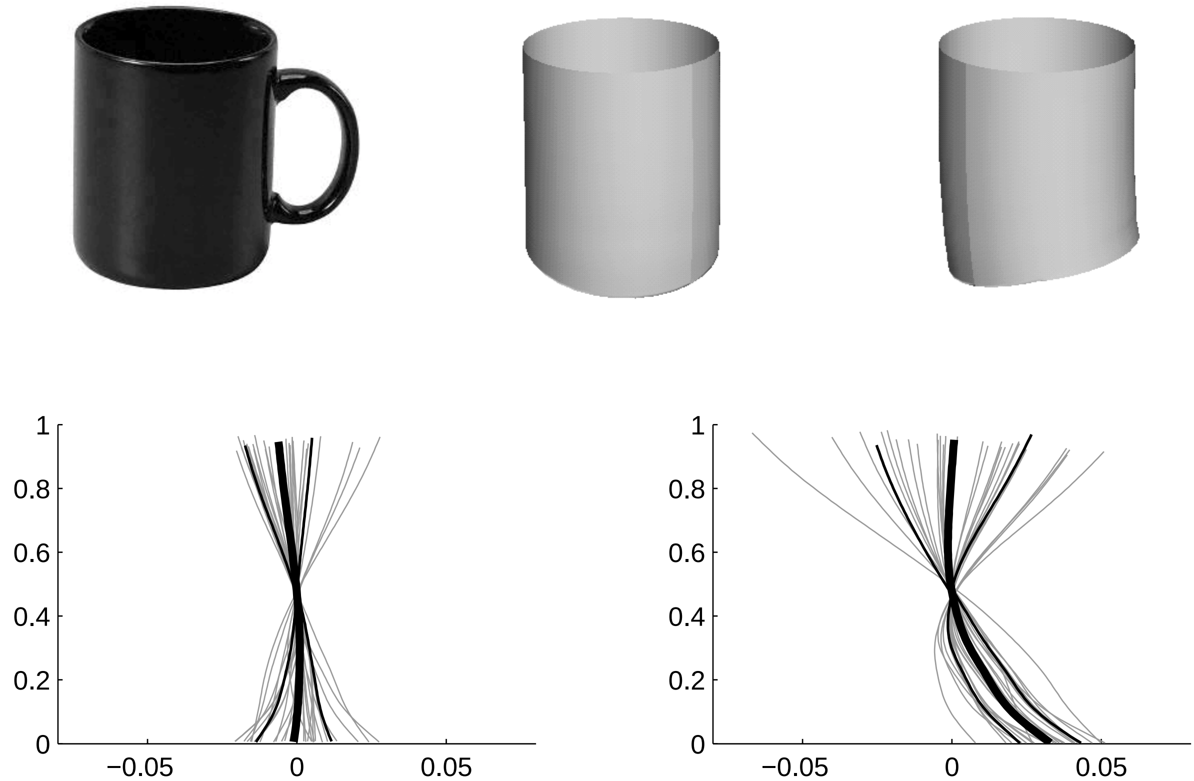

Figure 2. Results from fitting the generalized cylinder model to an image of a mug. Top left: target image. Middle: model which corresponds to the mean of estimated parameter distribution. Right: Side view of the same model. Neither direction of light nor texture was estimated in this example. The distribution of the estimated axis profile is shown in the bottom row; front view on the left and side view on the right. The black curves depict the mean and standard deviation of the distributions and the gray curves are a randomly chosen subset of individual samples. Note the stretched aspect ratio which makes the samples seem less linear than they actually are.

model geometry, the texture C_T that is required to produce image I is computed as

$$C_T = \frac{C}{C_L} = \frac{I}{k_a + k_d(\vec{N} \cdot \vec{L}) + k_s(\vec{R} \cdot \vec{V})^{\nu}}. \tag{4}$$

A non-zero ambient term k_a ensures that the texture estimate remains finite everywhere. An example of the texture estimation procedure is shown in Fig. 3. The shape of the object has been estimated using a generalized cylinder model and Metropolis sampling as in section 3.4. The illumination-corrected texture, estimated using eq. (4), is shown in Fig 3*b*. Note how the texture remains uniformly white regardless of the shading present in the target image. The black stripe on the rightmost edge of the mug appears because the shape estimate was not quite accurate. The form of the texture is completely unconstrained, and thus any texture that produces, together with the shape, an image similar to the target image is considered plausible. One possible way to define a prior probability term for the texture is based on total variation. This technique is used in many inverse problems in medical imaging.

3.6. Pose and texture estimation of known shape

In this example we describe the estimation of pose and texture when the shape of the object is assumed to be known. This is because we do not yet have a parameterized model for human heads. The six parameters which control the pose are estimated using Gibbs sampling and the texture is extracted from the input image. No known correspondences between the images are used. The synthetic reference images were created by mapping a photograph

a) b) c) d)

Figure 3. Estimation of shape and texture. a) Target image. b) Estimated, illumination-corrected texture. c) Texture mapped shape estimate. The dark artifacts near the edge of the mug appear because of inaccuracies in the shape estimation stage. d) New view and illumination.

on a generic face model. The likelihood function of pose parameters is defined by

$$P(I_{ref}|\theta, \phi, \psi, \vec{r}) \propto \exp\left(-\frac{1}{\sigma^2}\sum ||I_{ref} - I(\theta, \phi, \psi, \vec{r})||^2\right), \tag{5}$$

where θ, ϕ and ψ denote the three unknown rotation angles, $\vec{r}$ is the center of rotation, I is the image rendered using specific parameter values and I_{ref} is the synthetic reference image. The value of σ^2, controlling the variance of assumed noise in the images and thus the width of peaks in the likelihood function, was fixed. With images of 400 by 400 pixels and gray scale values in the range [0..1], $\sigma^2 = 50$ was used. Uniform priors were assumed for all camera parameters. The image I created by the renderer is not texture mapped during evaluation of the likelihood.

Gibbs sampling is performed by evaluating the full conditional posterior distributions, for example $P(\psi|\theta, \phi, \vec{r}, I_{ref})$, one at a time in a tightly spaced grid. One sample of ψ is drawn from its full conditional distribution. When all the parameters have been updated once, a new MCMC sample from the joint posterior distribution $P(\psi, \theta, \phi, \vec{r}|I_{ref})$ has been obtained.

Due to the heavy coupling of the rotational and translational parameters, Gibbs sampling can take a large number of iterations before converging to a probability maximum. Peaks in the conditional distributions of the translation parameters are significantly sharper than those of the rotational parameters, which also makes the estimation task harder.

Fig. 4 shows the results obtained in a successful sampling run. All parameters in the initial pose are incorrect, but within a reasonable range from the correct values. The pose estimate is obtained using Gibbs sampling as the median of ten final samples. Because the error criterion in this example is only based on the pixelwise difference of the two images, the silhouette of the object is not entirely accurate despite complete knowledge of the shape. The frontal view was generated by simply assuming that the object is symmetrical along a known axis.

4. CONCLUSION AND FUTURE WORK

Our results indicate that Bayesian inference using MCMC sampling is a suitable approach for 3-D object recognition. We expect the results could be improved significantly by using more informative prior parameter distributions which describe known model classes.

Combining the estimation of shape and texture is an important step. In the examples presented in this paper, shape estimation is mostly based on edges and in some cases the clear separation of the object from the background. This is a severe limitation, and we plan to overcome it by including a measure of homogeneity to differentiate between background and foreground. One way to do this is the principle of minimizing total variation, used in medical imaging. This could be implemented by defining a suitable prior probability for the texture.

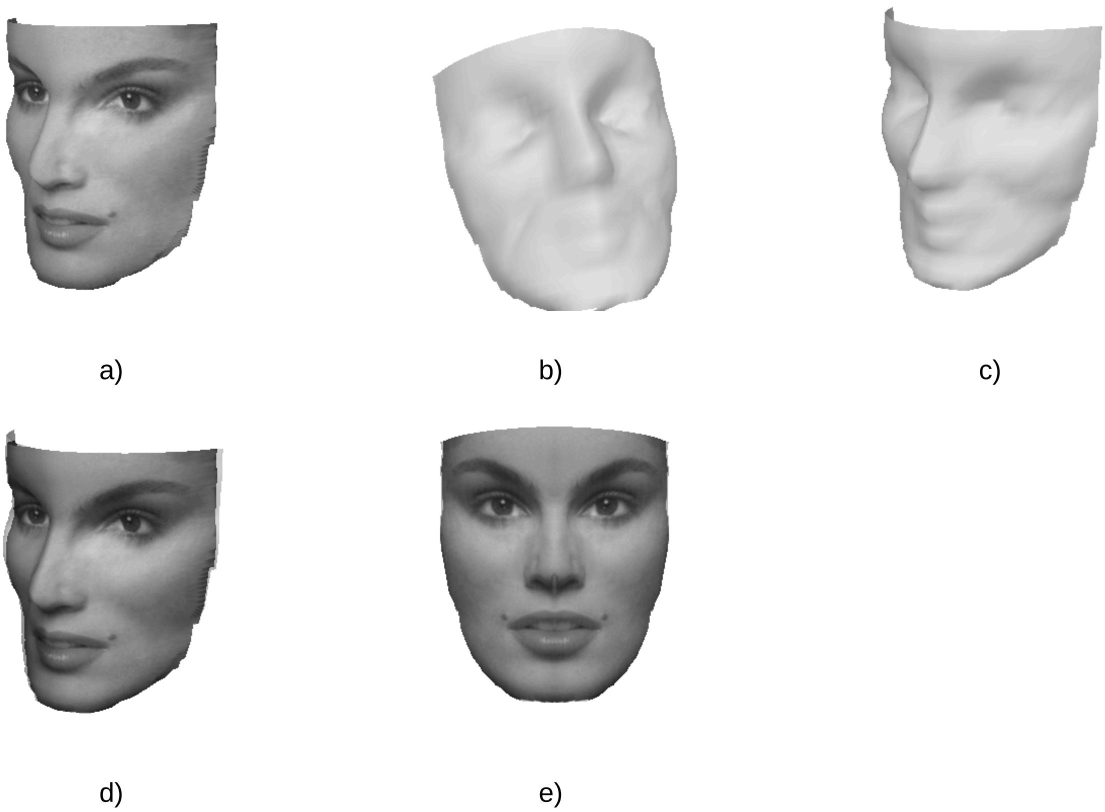

Figure 4. Results from pose and texture estimation of a known shape. a) Synthetic image which has been generated by texture mapping a photograph on a generic face model. b) Initial pose. c) Estimated pose using Gibbs sampling. d) Estimated texture applied on the model. e) Frontal view of the texture mapped model.

Other future challenges include creating new model classes with informative prior distributions and joining them together in a hierarchical structure, as well as the treatment of composite objects. Effective ways of learning of new object classes from 2-D sample images should also be considered.

REFERENCES

1. A. Gelman, J. B. Carlin, H. S. Stern, and D. R. Rubin, *Bayesian Data Analysis*, Texts in Statistical Science, Chapman & Hall, 1995.
2. H. H. Bülthoff, S. Y. Edelmann, and M. S. Tarr, "How are three-dimensional objects represented in the brain?," *Cerebral Cortex* **5**(3), pp. 247–260, 1995.
3. T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7), pp. 733–742, 1997.
4. J. J. Atick, P. A. Griffin, and A. N. Redlich, "Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images," *Neural Computation* **8**(6), pp. 1321–1340.
5. V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Siggraph 1999, Computer Graphics Proceedings*, A. Rockwood, ed., pp. 187–194, Addison Wesley Longman, (Los Angeles), 1999.
6. P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," *Computer Graphics* **30**(Annual Conference Series), pp. 11–20, 1996.
7. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, Reading, MA, 1990. Second edition.