Spherical Light Field Rendering in Application for Analysis by Synthesis

Todt, S.; Langer, M.; Rezk-Salama, C.; Kolb, A. and Kuhnert, K.-D.

Institute for Vision and Graphics, University of Siegen, Germany
E-mail: {Kolb,Kuhnert,Rezk,Todt,Langer}@fb12.uni-siegen.de

Abstract: This paper presents a novel approach for object classification and pose estimation which employs spherical light field rendering to generate virtual views based on synthesis parameters determined and successively refined in a two-stage analysis by synthesis process. Compared to previous object recognition techniques the presented approach provides a significant improvement in terms of object classification quality and computational efficiency.

Our GPU based light field renderer exploits per-pixel depth information available with modern time-of-flight sensors such as the PMD camera for high-quality image synthesis in real-time. The renderer uses combined per-pixel RGB and depth values to minimize ghosting artefacts to a non noticeable amount and employs a spherical parameterisation to ensure full six degrees of freedom for virtual view synthesis. Synthetic views are used in our two-stage analysis by synthesis technique which implements a pre-classification and pre-pose-estimation to reduce parameter search space, thus providing improved object classification quality at less computation time.

Keywords: Analysis by synthesis; Image-Based Rendering; Spherical Light Field

1 Introduction

Object classification and 6D pose estimation forms a ubiquitous problem in mobile robot and automotive navigation. Environmental knowledge allows a flexible and autonomous behaviour in different situations. Various techniques in sensorics and algorithms have been exploited to gather such information. The most popular sensors aiming at collecting information about the environment are ultrasonic sensors, laser scanners, monocular and binocular camera systems. Lately a new sensor type has been developed: the time–of–flight camera. These cameras provide intensity, depth and modulation images in real–time. The PMD (*photonic mixer device*) camera (Kraft et al., 2004 [3]) developed by the ZESS institute at the University of Siegen is an example for this camera type.

We present a new approach for 3D object classification and pose estimation, that exploits the advantages of combined color and distance data (RGBz data) available with such devices. A sparse set of RGBz images is acquired employing PMD technology to define an object representation. Based on this representation new virtual photorealistic views are generated in real–time according to varying synthesis parameters using our new spherical light field rendering technique. Synthetic views are evaluated in the successive object classification process.

The object classification and pose estimation is performed using an *analysis by synthesis* approach, in which object images are iteratively matched against synthetic images created by applying specific parameters. The goal is to find optimal synthesis parameters for an exact match with the object image, that is to be analyzed. In contrast to classic approaches we use a pre-classification and pre-pose-estimation stage to significantly reduce the search space and gain a coarse approximation of the initial synthesis parameters. The synthesis parameters are iteratively refined based on the evaluation of the generated views and used to steer the image synthesis process.

The remainder of the paper is structured as follows. In Section 2 we give an overview of the image acquisition process and introduce our image based rendering method. Object classification and pose estimation is explained in detail in Section 3. Section 4 evaluates our approach with respect to object classification and pose estimation quality. In Section 5 we draw conclusion and comment on future work.

2 Spherical Light Field Rendering

Light field rendering approaches are targeted at the continuous reconstruction of the *plenoptic function* from a set of input images. The 7D *plenoptic function* $\rho = P(\theta, \phi, \lambda, V_x, V_y, V_z, t)$ describes all of the radiant energy that can be received at a certain point in space (V_x, V_y, V_z) from any possible direction (θ, ϕ) at a certain moment in time t for a chosen band of wavelengths λ (Adelson and Bergen, 1991 [1]). The plenoptic function can be reduced to a 5D function by restricting it to a certain moment, eliminating t and reducing the wavelength λ using the primary values RGB. Under the assumption of the observation space to be free of occluders and the object being observed from positions completely outside its convex hull the 5D function is further reduced to a 4D function of position and direction. This 4D representation is described as *light field* (Levoy and Hanrahan, 1996 [4]).

2.1 Spherical Parameterisation

For our light field rendering technique a spherical parameterisation is chosen to discretise the 4D plenoptic function. The parameterisation is based on a polygonal approximation of the sphere which is generated by applying a regular one-to-four interpolatory subdivision scheme on a mesh generated from an icosahedron (see Fig. 1 (a)). The generated vertices define the sample positions of individual virtual cameras (see Fig. 1 (b)). For each camera an RGBz raster image of the opposite hemisphere is stored using a parabolic parameterisation (see Fig. 1 (c)).

The uniform spherical arrangement guarantees full 6DOF for virtual view point selection in the reconstruction process and reduces discontinuity render artefacts to a minimum (see Sec. 5).

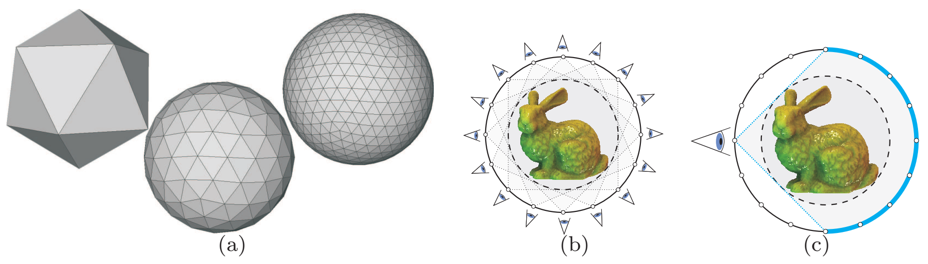

Figure 1 a: Spherical proxies are generated by successively subdividing the 20 faces of an icosahedron(left) into spherical approximations with 162 vertices (2 recursions, middle) and 642 vertices (3 recursions, right). b: The light field consists of a predefined set of camera positions uniformly distributed on a sphere around the object. c: Only the opposing hemisphere is recorded for each camera.

2.2 Light Field Acquisition

The combination of a high-resolution RGB camera and a PMD camera device in a bifocal setup yields combined per-pixel depth and RGB information. Thus such a setup provides the ideal solution for hand-held light field acquisition (Lindner et al., 2007 [7]). Lateral and depth calibration of PMD distance sensors and the fusion of high resolution RGB images and per pixel depth information however is still an active research field (Lindner and Kolb, 2006 [6]).

A time-of-flight simulation framework that emulates the characteristics of the PMD camera (Keller et al., 2007 [8]) is employed for light field acquisition at the current state of development. The framework works on a synthetic 3D model which in our case was generated manually from a physical object of daily usage (see Figure 6). For each virtual camera, an RGBz image of the 3D model is captured and stored in a 2D texture. In practice we chose a spherical approximation yielding 642 or 2562 camera positions for the acquisition task and store individual samples with a resolution of 256×256. Using S3TC DXT3 texture compression the light field images can be stored efficiently (S3TC, 1999 [11]). With the DXT3 compression algorithm a compression ratio of 4:1 is achieved reducing the data size of a complete light field to 41.1 MB for 642 samples and 164.2 MB for 2562 samples.

2.3 Light Field Rendering

Light field reconstruction is performed by rendering the smooth shaded spherical proxy. In the rendering process the front faces of the proxy are rasterized with respect to the virtual viewpoint as defined by the synthesis parameters (position and orientation of the virtual camera). For each triangle the RGBz texture images and the viewing matrices $\mathbf{M}$ of the three cameras located at the triangle's vertices are used to calculate per pixel color information. Calculations are performed on the *Graphics Processing Unit (GPU)* using a customized pixel program. The pixel program implements our new light field rendering technique exploiting per-pixel depth information for high-quality image synthesis in real–time.

The rendering algorithm implements a ray casting approach. When a triangle is rasterized, the pixel program calculates the viewing ray, which corresponds to the pixel. We then calculate the intersection point of the viewing ray with the bounding

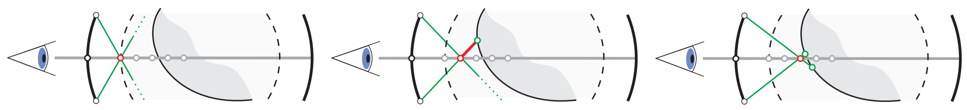

Figure 2 The rendering algorithm samples the viewing ray stepwise at adjacent positions starting at the intersection point with the object's bounding sphere.

sphere of the object. From this position, we sample the viewing ray iteratively at a fixed step size, as shown in Fig. 2. At each step, we validate the assumption that the current ray position is the actual object intersection point. We project this point onto the camera sphere using the adjacent camera positions as center of projection. The resulting three intersection points with the sphere are transformed into the viewing coordinate system of the corresponding camera and converted to parabolic coordinates. Depth values are obtained from the corresponding parabolic texture maps and local estimates of object surface points are calculated for each camera. We then compare these points to the position of the current ray sample. If one of the local estimates is equal to the ray position within a given tolerance, we immediately stop the ray sampling. This means that we have found at least one camera that reliably observes an object intersection at exactly the ray position.

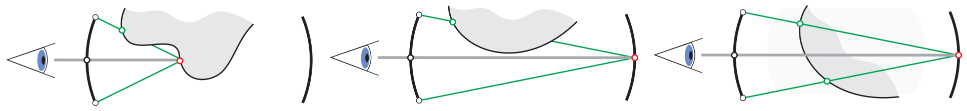

Figure 3 Left and Middle: At silhouette edges and locations where the correct object point is not visible from all adjacent cameras the camera with the most divergent intersection point is discarded. Right: Without depth correction the intersection points observed from adjacent cameras do not necessarily correspond to an identical surface point.

We now compare the intersection points obtained by the other two cameras. If they are also equal to the ray position within the given tolerance, we can assume that all cameras observe the same point and we use the barycentric weights of the pixel to calculate the final color for that pixel as outlined above. However, if the intersection point for one camera is far away from the ray position, we can assume that we are in one of the situations outlined in Fig. 3 (left and middle). In this case we discard the color information for the respective camera by setting the corresponding barycentric weight to zero. Afterwards, we normalize the weights again and calculate the final color as a weighted sum.

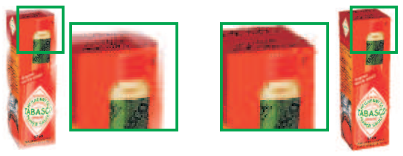

Figure 4 Left: Heavy ghosting artefacts appear without per-pixel depth correction. Right: With depth correction ghosting artefacts can be reduced to a minimum.

The raycasting approach represents a very accurate solution to per-pixel depth refinement providing high-quality images for object classification. Without depth correction a similar approach would sample the parabolic maps directly according to the spherical intersection point calculated for the viewing ray. Though computationally more efficient this assumption leads to heavy ghosting artefacts (see Fig. 4) in the image synthesis due to varying surface points being observed from adjacent cameras (see Fig. 3, right). Ghosting artefacts tremendously reduce object classification quality. Thus rendering techniques exploiting per-pixel depth information are superior to alternative approaches for usage in analysis by synthesis.

3 Analysis by Synthesis

Object classification and pose estimation is performed using an analysis by synthesis approach. This approach analyzes synthetic views generated by the light field renderer for a given viewpoint from the light field representation of an object (see Fig. 5). Steered by a similarity function F the viewpoint parameters are iteratively refined based on the comparison of the synthetic view and the image which is to be analyzed. Iteration is stopped if a certain similarity is achieved for the virtual view, compared to the analyzed image. There are various kinds of metrics the similarity function can use. One of the simplest and fastest is to take the difference of two measurement functions $f_{\mathrm{x}}(\boldsymbol{c}_{\mathrm{x}}, \boldsymbol{\theta}_{\mathrm{x}}, \boldsymbol{p}_{\mathrm{x}}) - f_{\mathrm{s}}(\boldsymbol{c}_{\mathrm{s}}, \boldsymbol{\theta}_{\mathrm{s}}, \boldsymbol{p}_{\mathrm{s}})$ for each image and let this difference converge to zero (see Fig. 5).

In our implementation we use a pre–stage synthesis parameter estimation step. This very efficient pre–stage reduces the search space for the optimal parameters significantly. The pre–stage facilitates a much faster approximation of the final accurate parameters. Our goal is to achieve an entire analysis by synthesis computation in real–time ($< 1s$).

We developed and validated our two–stage object classification based on the Coil–100 image database in the initial development phase. The Coil–100 database contains images of 100 objects of daily usage. For each object images for a full 360 degree rotation around the Y–axis are available at five degree steps. The Coil–100 database was consecutively supplemented by synthetic views of similar objects generated from the light field renderer to evaluate object classification quality working on our light field rendering in comparison to results from images commonly used for object recognition. Figure 6 shows some typical images of objects we use in our work from different angles. Figure 6 (c) and (d) were generated from light field representations using a spherical parameterisation with 642 camera positions and an image resolution of 256×256. They clearly resemble those objects, that can be found in the Coil–100 library.

3.1 Pre Stage Part

Rough approximations of the synthesis parameters are determined in a pre-classification and pre-pose-estimation step. We concentrate on using *decision trees* for fast object classification.

Decision trees are built on a feature space defined by distinct features for class description. The feature space is recursively subdivided into sets of similar size and the subsets are fed to the decision tree nodes. To classify an object, the object features are matched against the class feature sets each node holds. An object can be classified to belong to one (or none) of an individual node's two class–feature subtrees. This process is repeated recursively on the selected subtree until a leaf–node is reached. The feature represented by this leaf node yields the class the object belongs to. Because the trees need not necessarily to be built up completely, whole

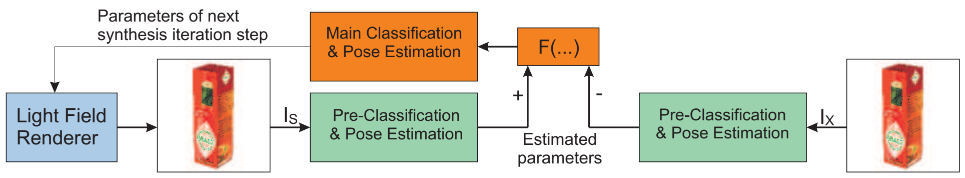
Parameters of next synthesis iteration step
Main Classification & Pose Estimation
F(...)
Light Field Renderer
Is
Pre-Classification & Pose Estimation
+
-
Estimated parameters
Pre-Classification & Pose Estimation
Ix

Figure 5 Analysis by Synthesis scheme. I_{x} denotes an arbitrary input image made from an object, that was previously synthesized, whereas I_{s} denotes a synthetic image requested from the synthetic renderer with given rendering–parameters. F represents the similarity function and is dependend on the estimated parameters. These parameters include the object classes $\boldsymbol{c}_{\mathrm{x}}, \boldsymbol{c}_{\mathrm{s}}$, the 3D Euler angles $\boldsymbol{\theta}_{\mathrm{x}} = (\alpha_{\mathrm{x}}, \beta_{\mathrm{x}}, \gamma_{\mathrm{x}}), \boldsymbol{\theta}_{\mathrm{s}} = (\alpha_{\mathrm{s}}, \beta_{\mathrm{s}}, \gamma_{\mathrm{s}})$ and the 3D object locations $\boldsymbol{p}_{\mathrm{x}} = (x_{\mathrm{x}}, y_{\mathrm{x}}, z_{\mathrm{x}}), \boldsymbol{p}_{\mathrm{s}} = (x_{\mathrm{s}}, y_{\mathrm{s}}, z_{\mathrm{s}})$ of I_{x} and I_{s} respectively. To find the optimal synthesis parameters $F(\boldsymbol{c}_{\mathrm{x}}, \boldsymbol{c}_{\mathrm{s}}, \boldsymbol{\theta}_{\mathrm{x}}, \boldsymbol{\theta}_{\mathrm{s}}, \boldsymbol{p}_{\mathrm{x}}, \boldsymbol{p}_{\mathrm{s}}) \rightarrow 0$ must be approximated.

class or feature sets can be hold by leaf nodes. (See Winston, 1993 [12], Russel, 2003 [10] for an in–depth look into decision trees.)

To gain the features for our decision tree, we use the following technique. We implemented an octave lowpass pyramid filter for recursively downsampling the input images, this goes for training image sets as well as the test image sets. Then we take the average RGB value of the downsampled image and mark that value in the RGB space yielding clusters for image sets that belong to one class. After that the clusters' centers of gravity points are calculated. This is done recursively computing new centers' of gravity using the previously gained points. These points are fed into the nodes of our decision tree with attached class sets. This is done in such a way that every node performs a decision between two centers of gravity for any given downsampled RGB value of a test image that is to be classified. The tree is of binary nature. Each node divides the center of gravity set (and so the associated class set) into half. This way we achieve a runtime complexity of $O(M \log(N))$ with N denoting the number of classes and M the number of objects to classify.

An extension to normal decision trees are *decision forests*. Decision forests aggregate arbitrary decision trees, which have been trained with different data sets. To classify an object, the object is classified by each decision tree. After that a quorum is made amongst the trees in the forest to find the one tree, that produces the best results and thus determines the classification result. The runtime complexity rises to $O(KM \log(N))$ with K denoting the number of trees in the forest. Decision forests have proven to result in higher classification quality, compared to decision trees (see Section 4). Additionally decision forests provide the possibility to attach pose information to individual trees. Various trees in a forest can be trained with object sets captured from predefined positions and orientations. Hence the decision tree that wins the quorum narrows the pose parameters of an object in addition to it's classification.

3.2 Main Stage Part

The main stage of the object recognition is working on the pre-classified objects. In this stage we exploit *SIFT–features* (Scale Invariant Feature Transform) as proposed by Lowe (Lowe, 2003 [9]) for a most accurate object classification and pose

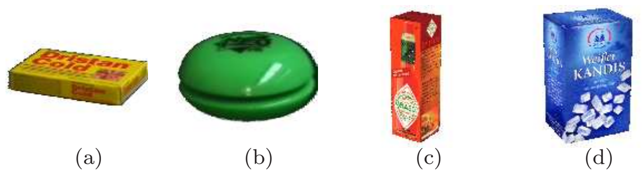
Figure 6 Sample images used in this work. Images (a) and (b) are taken from the Coil–100 library. Images (c) and (d) are synthesized by our light field renderer.

estimation. SIFT features describe scale-invariant local coordinates relative to local features extracted from sample image data. The features are highly distinctive and are invariant to image scaling and rotation[a]. There is also a partial invariance to illumination. For usage in object classification a database of SIFT features is built up from a training set of images. The feature database is generated in four steps:

1. Scale–space extrema detection
2. Keypoint localisation
3. Orientation assignment
4. Keypoint descriptor computation

The keypoint descriptors are stored in the SIFT database. The object recognition performs on these descriptors by matching keypoint descriptors of input images against the database. For the input image a SIFT feature extraction is performed as described above. For reliable results, clusters of at least three features are identified, that agree on the object and its pose. Each cluster is checked performing a detailed geometric fit to the model determining acceptance or rejection of the estimation. This reduces the incorrect recognized object rate drastically.

4 Results

The combination of our light field rendering approach with analysis by synthesis for object classification and pose-estimation as described in this paper shows promising results in performance and object recognition quality.

Synthetic views are rendered at a resolution of 128×128 from previously acquired object light fields. Using a spherical approximation of 642 and 2562 vertices high-quality virtual views are generated at frame-rates of 17fps and 10fps, respectively. Both parameterizations have been proven to generate synthetic images well suited for object classification. Classification quality based on synthetic images from the light field renderer has shown to have a slight variation (1,5% in average) compared to the classification quality based on the Coil–100 database. This low deviation proves the quality of the synthetic images regarding photorealism. Hence the synthetic images can be merged smoothly with the Coil-100 database without causing any significant tampering of the yielded results.

[a]for projections of 3D scenes Lowe states that the rotation invariance regarding viewpoint depth has a maximum of 30 degree

Table 1 Quality of different classification configurations. The columns show the number of decision trees used and their configuration. Quality is expressed as ratio of correctly classified objects to total amount of objects.

trees in forest	*tree configuration**	*classification quality*	*average quality*	*standard deviation*
1	0-355	63,75%		
1	0,10,20	64,03%		
1	0,45,90,135,180,225,270,305,355	66,39%		
1	0-45	72,36%		
1	0,35,70,105,140, . . . ,355	63,75%		
1	0,90,180,270,355	63,75%	65,67%	3,44%
2	{0-45},{325-355}	69,31%		
2	{0,15,30},{90,140}	74,86%		
4	{0-85},{90-175},{180-265},{270-355}	74,31%		
4	{0-30},{85-95},{170-205},{265-275}	73,47%		
5	{0,35},{70,105}, . . . , {315,355}	79,17%		
10	{0},{35},{70},{105},{140}, . . . ,{355}	77,22%	74,72%	3,38%

*The angles from which the pictures used to teach the decision trees are taken. Curly brackets denote angle–sets that are associated with one tree in a forest.

The two-stage analysis by synthesis approach yields a very efficient object classification method providing high accuracy object classification results. The pre-stage of the algorithm drastically reduces search-space at minimal computational costs of a few milliseconds. Pre-stage results further positively affect the main-stage's computational effort. A typical feature match of an image with the SIFT feature database performs in a few hundred milliseconds.

As shown in Table 1 a considerable gain in object classification quality, expressed as ratio of correctly classified objects to the total amount of objects, is achieved by employing a decision forest for classification purposes instead of a single tree. For decision trees an average quality of 66% is achieved (see Table 1, upper half) whereas decision forests containing more than a single tree provide a classification quality of 75% in average (see Table 1, lower half). Quality however is not exclusively dependent on the tree-count. Best results can be achieved from carefully crafted decision forests of several properly configured decision trees.

5 Conclusion and Future Work

In this paper we provided insight in our object classification and pose estimation approach combining light field rendering and a two-stage analysis by synthesis technique. Our aim is to build an object classification system yielding high accurate classification results in real-time.

Light field rendering has been proven to be a powerful rendering technique for real–time generation of photorealistic images. Light field rendering techniques are based on a sparse set of input images. Thus no time consuming and computational complex pre-processing has to be performed to extract geometric information of the scene. Per-pixel depth information available with PMD technology however improve image synthesis quality of the light field renderer and directly leads to a significant improvement in object classification quality. In future development per-pixel depth information will be extracted from within the light field rendering process to provide additional features for object classification.

We are confident that per-pixel depth information in combination with SIFT image descriptors will increase object classification quality at lower computational costs in future development. Exploiting SIFT features offers a way to approximate the final synthesis parameters most accurately. This is due to the scale and rotation invariant nature of the image descriptors gained from SIFT. Even image distortions and partial occlusions are tolerated. In future work we will provide our own implementation of SIFT for reliable object classification and pose estimation. We will use our pre–classication stage to dramatically reduce the matching time of object features against the descriptor database, yet trying to preserve most accurate matching results.

Acknowledgements

This research project is partially funded by grant KO-2960/6-1 and KU-689/6-1 from the German Research Foundation (DFG).

References and Notes

1 E. H. Adelson; J. R. Bergen (1991) 'The Plenoptic Function and the Elements of Early Vision', *Computational Models of Visual Processing*, Cambridge, pp. 3–20

2 Columbia University Image Library (COIL–100), `http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php`

3 H. Kraft; J. Frey; T. Moeller; M. Albrecht; M. Grothof; B. Schink; H. Hess; B. Buxmbaum (2004) '3D-Camera of high 3D-frame rate, depth–resolution and background light elimination based on improved PMD (photonic mixer device)–technologies', *OPTO*, Nuernberg

4 M. Levoy; P. Hanrahan (1996) 'Light Field Rendering', *Proc. ACM SIGGRAPH*, pp. 31–42

5 T. Lindeberg (1998) 'Feature Detection with Automatic Scale Selection', *Int. Journal of Computer Vision*, Vol. 30, No. 2, pp. 77–116

6 M. Lindner; A. Kolb (2006) 'Lateral and Depth Calibration of PMD-Distance Sensors', *Advances in Visual Computing*, pp. 524-533

7 M. Lindner; A. Kolb (2007) 'Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images', to app. in *Proc. Int. Symp. on Signals, Circuits & Systems*

8 M. Keller; J. Orthmann; A. Kolb; V. Peters (2007) 'A Simulation Framework for Time-Of-Flight Sensors', to app. in *Proc. Int. Symp. on Signals, Circuits & Systems*

9 D.G. Lowe (2003) 'Destinctive Image Features from Scale-Invariant Keypoints', *Int. Journal of Computer Vision*, Vol. 20, pp. 91–110

10 S. Russel; P. Norwig (2003) 'Artificial Intelligence — A Modern Approach', *Prentice Hall*, ISBN 0130803022, pp. 653–664

11 K. I. Iourcha; K.S. Nayak; Z. Hong (1999) 'System and Method for Fixed-Rate Block-Based Image Compression with Inferred Pixel Values', *United States Patent 5,956,431*

12 P.H. Winston (1993) ‘Artificial Intelligence 3rd ed.’, *Addison Wesley*, ISBN 0-201-53377-4, pp. 423–442