Datasets and Benchmarks for Densely Sampled 4D Light Fields

Sven Wanner, Stephan Meister and Bastian Goldluecke
Heidelberg Collaboratory for Image Processing

Abstract
We present a new benchmark database to compare and evaluate existing and upcoming algorithms which are tailored to light field processing. The data is characterised by a dense sampling of the light fields, which best fits current plenoptic cameras and is a characteristic property not found in current multi-view stereo benchmarks. It allows to treat the disparity space as a continuous space, and enables algorithms based on epipolar plane image analysis without having to refocus first. All datasets provide ground truth depth for at least the center view, while some have additional segmentation data available. Part of the light fields are computer graphics generated, the rest are acquired with a gantry, with ground truth depth established by a previous scanning of the imaged objects using a structured light scanner. In addition, we provide source code for an extensive evaluation of a number of previously published stereo, epipolar plane image analysis and segmentation algorithms on the database.

1. Introduction
The concept of a light field was originally used mainly in computer graphics as a powerful tool to describe scene appearance [AB91, LH96], but recently it is also getting more and more attention from the computer vision community. One of the likely reasons is the availability of cheap recording devices. While the first light field capturing techniques used large camera arrays [WJV∗05] which are expensive and not very practicable, hand-held light field cameras [Ng06, PW10] are now available on the consumer market.

However, the driving force for successful algorithm development is the availability of suitable benchmark datasets with ground truth data in order to compare results and initiate competition. The current public light field databases we are aware of are the following.

- **Stanford Light Field Archive**
  http://lightfield.stanford.edu/lfs.html
  The Stanford Archives provide more than 20 light fields sampled using a camera array [WJV∗05], a gantry and a light field microscope [LNA∗06], but none of the datasets includes ground truth disparities.

- **Synthetic Light Field Archive**
  The synthetic light field archive provides many interesting artificial light fields including some nice challenges like transparencies, occlusions and reflections. Unfortunately, there is also no ground truth depth data available for benchmarking.

- **Middlebury Stereo Datasets**
  http://vision.middlebury.edu/stereo/data/
  The Middlebury Stereo Dataset includes a single 4D light field which provides ground truth data for the center view, as well as some additional 3D light fields including depth information for two out of seven views. The main issue with the Middlebury light fields are that they are designed with stereo matching in mind, and thus the baselines are quite large and thus not representative for plenoptic cameras and unsuitable for direct epipolar plane image analysis.

While there is a lot of variety and the data is of high quality, we observe that all of the available light field databases either lack ground truth disparity information or exhibit large camera baselines and disparities, which is not representative for plenoptic camera data. Furthermore, we believe that a
large part of what distinguishes light fields from standard multi-view images is the ability to treat the view point space as a continuous domain. There is also emerging interest in light field segmentation [KSS12, SHH07, EM03, WSG13], so it would be highly useful to have ground truth segmentation data available to compare light field labeling schemes. The above datasets lack this information as well.

Contributions. To alleviate the above shortcomings, we present a new benchmark database which consists at the moment of 13 high quality densely sampled light fields. The database offers seven computer graphics generated datasets providing complete ground truth disparity for all views. Four of these datasets also come with ground truth segmentation information and pre-computed local labeling cost functions to compare global light field labeling schemes. Furthermore, there are six real world datasets captured using a single Nikon D800 camera mounted on a gantry. Using this device, we sampled objects which were pre-scanned with a structured light scanner to provide ground truth ranges for the center view. An interesting special dataset contains a transparent surface with ground truth disparity for both the surface as well as the object behind it - we believe it is the first real-world dataset of this kind with ground truth depth available.

We also contribute a CUDA C library with complete source code for several recently published algorithms to demonstrate a fully scripted evaluation on the benchmark database and find an initial ranking of a small subset of the available methods on disparity estimation. We hope that this will ease the entry into the interesting research area which is light field analysis, and are fully committed to increasing the scope of the library in the future.

2. The light field archive

Our light field archive (www.lightfield-analysis.net) is split into two main categories, Blender and Gantry. The Blender category consists of seven scenes rendered using the open source software Blender [Ble] and our own light field plugin, see figure 2 for an overview of the datasets. The Gantry category provides six real-world light fields captured with a commercially available standard camera mounted on a gantry device, see figure 5. More information about all the datasets can be found in the overview in figure 1.

Each dataset is split into different files in the HDF5-format [The10], exactly which of these are present depends on the available information. Common to all datasets is a main file called lf.h5, which contains the light field itself and the range data. In the following, we will explain its content as well as that of the different additional files, which can be specific to the category.

<table>
<thead>
<tr>
<th>dataset name</th>
<th>category</th>
<th>resolution</th>
<th>GTD</th>
<th>GTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>buddha</td>
<td>Blender</td>
<td>768x768x3</td>
<td>full</td>
<td>yes</td>
</tr>
<tr>
<td>horses</td>
<td>Blender</td>
<td>576x1024x3</td>
<td>full</td>
<td>yes</td>
</tr>
<tr>
<td>papillon</td>
<td>Blender</td>
<td>768x768x3</td>
<td>full</td>
<td>yes</td>
</tr>
<tr>
<td>stillLife</td>
<td>Blender</td>
<td>768x768x3</td>
<td>full</td>
<td>yes</td>
</tr>
<tr>
<td>buddha2</td>
<td>Blender</td>
<td>768x768x3</td>
<td>full</td>
<td>no</td>
</tr>
<tr>
<td>medieval</td>
<td>Blender</td>
<td>720x1024x3</td>
<td>full</td>
<td>no</td>
</tr>
<tr>
<td>monasRoom</td>
<td>Blender</td>
<td>768x768x3</td>
<td>full</td>
<td>no</td>
</tr>
<tr>
<td>couple</td>
<td>Gantry</td>
<td>898x898x3</td>
<td>cv</td>
<td>no</td>
</tr>
<tr>
<td>cube</td>
<td>Gantry</td>
<td>898x898x3</td>
<td>cv</td>
<td>no</td>
</tr>
<tr>
<td>maria</td>
<td>Gantry</td>
<td>926x926x3</td>
<td>cv</td>
<td>no</td>
</tr>
<tr>
<td>pyramid</td>
<td>Gantry</td>
<td>898x898x3</td>
<td>cv</td>
<td>no</td>
</tr>
<tr>
<td>statue</td>
<td>Gantry</td>
<td>898x898x3</td>
<td>cv</td>
<td>no</td>
</tr>
<tr>
<td>transparency</td>
<td>Gantry</td>
<td>926x926x3</td>
<td>cv</td>
<td>no</td>
</tr>
</tbody>
</table>

Figure 1: Overview of the datasets in the benchmark. dataset name: The name of the dataset. category: Blender (rendered synthetic dataset) or Gantry (real-world dataset sampled using a single moving camera). resolution: spatial resolution of the views, all light fields consist of 9x9 views. GTD: indicates completeness of ground truth depth data, either cv (only center view) or full (all views). A special case is the transparency dataset, which contains ground truth depth for both background and transparent surface. GTL: indicates if object segmentation data is available.

2.1. The main file

The main file lf.h5 for each scene consists of the actual light field image data as well as the ground truth depth, see figure 1. Each light field is 4D, and sampled on a regular grid. All images have the same size, and views are spaced equidistantly in horizontal and vertical direction, respectively. The general properties of the light field can be accessed in the following attributes:

<table>
<thead>
<tr>
<th>HDF5 attribute</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yRes</td>
<td>height of the images in pixel</td>
</tr>
<tr>
<td>xRes</td>
<td>width of the images in pixel</td>
</tr>
<tr>
<td>vRes</td>
<td># of images in vertical direction</td>
</tr>
<tr>
<td>hRes</td>
<td># of images horizontal direction</td>
</tr>
<tr>
<td>channels</td>
<td>light field is rgb (3) or grayscale (1)</td>
</tr>
<tr>
<td>vSampling</td>
<td>rel. camera position grid vertical</td>
</tr>
<tr>
<td>hSampling</td>
<td>rel. camera position grid horizontal</td>
</tr>
</tbody>
</table>

The actual data is contained in two HDF5 datasets:

<table>
<thead>
<tr>
<th>HDF5 dataset</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF</td>
<td>vRes x hRes x xRes x yRes x channels</td>
</tr>
<tr>
<td>GT_DEPTH</td>
<td>vRes x hRes x xRes x yRes</td>
</tr>
</tbody>
</table>

These store the separate images in RGB or grayscale (range 0-255), as well as the associated depth maps, respectively.

Conversion between depth and disparity. To compare disparity results to the ground truth depth, the latter has to first be converted to disparity. Given a depth Z, the disparity
or slope of the epipolar lines $d$ in pixels per grid unit is

$$d = \frac{B \cdot f}{Z} - \Delta x,$$

where $B$ is the baseline or distance between two cameras, $f$ the focal length in pixel and $\Delta x$ the shift between two neighbouring images relative to an arbitrary rectification plane (in case of light fields generated with Blender, this is the scene origin). The parameters in equation 1 are given by the following attributes in the main HDF file:

<table>
<thead>
<tr>
<th>attribute</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>distance between to cameras</td>
</tr>
<tr>
<td>$f$</td>
<td>focal length</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>shift between neighbouring images</td>
</tr>
</tbody>
</table>

The following subsections describe differences and conventions about the depth scale for the two current categories.

### 2.2. Blender category

The computer graphics generated scenes consist without exception of ground truth depth over the entire light field. This information is given as orthogonal distance of the 3D point to the image plane of the camera, measured in Blender units $[BE]$. The Blender main files have an additional attribute `camDistance` which is the base distance of the camera to the origin of the 3D scene, and used for the conversion to disparity values.

#### 2.2.1. Segmentation ground truth

Some light fields have segmentation ground truth data available, see figure 1, and offer five additional HDF5 files:

- **labels.h5**: This file contains the HDF5 dataset `GT_LABELS` which is the segmentation ground truth for all views of the light field and the HDF5 dataset `SCRIBBLES` which are user scribbles on a single view.

- **edge_weights.h5**: Contains a HDF5 dataset called `EDGE_WEIGHTS` which
are probabilities for edges [WSG13] for all views. These are not only useful for segmentation, but any algorithm which might require edge information, and can help with comparability since all of these can use the same reference edge weights.

- **feature_single_view_probabilities.h5:**
  The HDF5 dataset Probabilities contains the prediction of a random forest classifier trained on a single view of the light field without using any feature requiring light field information [WSG13].

- **feature_depth_probabilities.h5:**
  The HDF5 dataset Probabilities contains the prediction of a random forest classifier trained on a single view of the light field using estimated disparity [WG12] as an additional feature [WSG13].

- **feature_gt_depth_probabilities.h5:**
  The HDF5 dataset Probabilities contains the prediction of a random forest classifier trained on a single view of the light field using ground truth disparity as an additional feature [WSG13].

### 2.3. Gantry category

In the Gantry category, each scene always provides a single main .h5 file, which contains an additional HDF5 dataset GT_DEPTH_MASK. This is a binary mask indicating valid regions in the ground truth GT_DEPTH. Invalid regions in the ground truth disparity have mainly two causes. First, there might be objects in the scene for which no 3D data is available, and second, there are parts of the mesh not covered by the structured light scan and thus having unknown geometry. See section 3.2.1 for details.

A special case is the light field transparency, which has two depth channels for a transparent surface and an object behind it, respectively. Therefore, there also exist two mask HDF5 datasets, see figure 3. We believe this is the first benchmark light field for multi-channel disparity estimation. Here, the HDF5 datasets are named

- **GT_DEPTH_FOREGROUND,**
- **GT_DEPTH_BACKGROUND,**
- **GT_DEPTH_FOREGROUND_MASK,**
- **GT_DEPTH_BACKGROUND_MASK.**

### 3. Generation of the light fields

The process of light field sampling is very similar for both the synthetic as well as the real world scenes. The camera is moved on an equidistant grid parallel to its own sensor plane and an image is taken at each grid position. Although not strictly necessary, an odd number of grid positions is used for each movement direction as there then exists a well-defined center view which makes the processing simpler. An epipolar rectification on all images is performed to align individual views to the center one. The source for the internal and external camera matrices needed for this rectification depends on the capturing system used.

**Figure 3:** Dataset transparency. Top: center view, bottom left: depth of the background, bottom right: depth of the foreground.

**3.1. Blender category**

For the synthetic scenes, the camera can be moved using a script for the Blender engine. As camera parameters can be set arbitrarily and the sensor and movement plane coincide perfectly, no explicit camera calibration is necessary. Instead, the values required for rectification can be derived directly from the internal Blender settings.

**3.2. Gantry category**

For the real-world light fields, a Nikon D800 digital camera is mounted on a stepper-motor driven gantry manufactured by Physical Instruments. A picture of the setup can be seen in figure 4. Accuracy and repositioning error of the gantry is well in the micrometer range. The capturing time for a complete light field depends on the number of images, about 15 seconds are required per image. As a consequence, this acquisition method is limited to static scenes. The internal camera matrix must be estimated beforehand by capturing images of a calibration pattern and invoking the camera calibration algorithms of the OpenCV library, see next section.
for details. Experiments have shown that the positioning accuracy of the gantry actually surpasses the pattern based external calibration as long as the differences between the sensor and movement planes are kept minimal.

### 3.2.1. Ground truth for the Gantry light fields

Ground truth for the real world scenes was generated using standard pose estimation techniques. First, we acquired 3D polygon meshes for an object in the scene using a Breuckmann SmartscanHE structured light scanner. The meshes contain between 2.5 and 8 Million faces with a stated accuracy of down to 50 micron. The object-to-camera pose was estimated by hand-picking 2D-to-3D feature points from the light field center view and the 3D mesh, and then calculating the external camera matrix using an iterative Levenberg-Marquardt approach from the OpenCV library [Bra00]. This method is used for both the internal and external calibration. An example set of correspondence points for the scene pyramid can be observed in figure 6.

The reprojection error for all scenes was typically $0.5 \pm 0.1$ pixels. The depth is then defined as the distance between the sensor plane and the mesh surface visible in each pixel. The depth projections are computed by importing the mesh and measured camera parameters into Blender and performing a depth rendering pass. At depth discontinuities (edges) or due to the fact that the meshes’ point density is higher than the lateral resolution of the camera, one pixel can contain multiple depth cues. In the former case, the pixel was masked out as an invalid edge pixel and in the latter case, the depth of the polygon with the biggest area inside the pixel was selected. The error is generally negligible as the geometry of the objects is sufficiently smooth at these scales. Smaller regions where the mesh contained holes were also masked out and not considered for the final evaluations.

For an accuracy estimation of the acquired ground truth, we perform a simple error propagation on the projected point coordinates. Given an internal camera matrix $C$ and an external matrix $R$, a 3D point $\vec{P} = (X, Y, Z, 1)$ is projected onto the sensor pixel $(u, v)$ according to

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = CR \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}.$$ 

For simplicity, we assume that the camera and object coordinate systems coincide, save for an offset $t_z$ along the optical axis. Given focal length $f_x$, principal point $c_x$, and reprojection error $\Delta u$, this yields for a pixel on the $v = 0$ scanline

$$t_z = Z - \frac{f_x X}{u - c_x}.$$
resulting in a depth error $\Delta z_t$ of

$$\Delta z_t = \frac{\partial t_z}{\partial u} \Delta u = \frac{f_x X (c_x - u)}{(c_x - u)^2} \Delta u.$$

Calculations for pixels outside of the center scanline are performed analogously. The error estimate above depends on the distance of the pixel from the camera’s principal point. As the observed objects are rigid, we assume that the distance error $\Delta z_t$ between camera and object corresponds to the minimum observed $\Delta z_t$ among the selected 2D-3D correspondences. For all gantry scenes, this value is in the range of 1 mm so we assume this to be the approximate accuracy of our ground truth.

4. Evaluation

The benchmark data sets are accompanied by a test suite which is coded in CUDA / C++ and which contains a number of implementations for recent light field analysis algorithms. Complete source code is available, and it is our intention to improve and update the test suite over the next years. As such, the list of algorithms and results below is only a snapshot of the current implementation. For example, an implementation of the multiview graph cut scene reconstruction [KZ02] is scheduled for an upcoming version. We will regularly publish benchmark results for the newest implementation.

The project will be hosted on SourceForge, and others are of course invited to contribute more algorithms for comparison, but this is not a requirement to get listed in the results. Instead, we will provide a method independent from the test suite code to submit results and have them ranked in the respective tables.

4.1. Disparity reconstruction algorithms

The following is a description of the methods for disparity reconstruction which are currently implemented in the test suite and ranked in the results. See figure 8 for an overview, and figure 7 for detailed results.

For all methods, we compute results from local data terms separately, and compared the same regularization schemes where possible. In the test suite, many more regularization schemes are implemented and are ready to use. However, comparisons are restricted to a subset of simple regularizers, since usually the data term has more influence on the quality of the result.

4.1.1. EPI-based methods

A number of methods exist which estimate disparity by computing the orientation of line patterns on the epipolar plane images [BBM87, CKS*05, WG12]. They make use of the well known fact that a 3D point is projected onto a line whose slope is related to the distance of the point to the observer.

For the benchmark suite, we implemented a number of schemes of varying complexity. First, we start with the purely local method EPI_L, which estimates orientation using an Eigensystem analysis of the structure tensor [WG12]. The second method, EPI_S, just performs a TV-$L^2$ denoising of this result [WG13], while EPI_G employs a globally optimal labeling scheme [WG12]. Finally, the method EPI_C performs a constrained denoising on each epipolar plane image, which takes into account occlusion ordering constraints [GW13].

Figure 8: Algorithms ranked by average accuracy over all data sets (mean squared error times 100). The computation time depends on exact parameter settings and can vary quite a bit, so we have only roughly classified the algorithms as fast (less than five seconds), medium (five seconds to a minute) and slow (more than a minute). The column all views indicates whether the algorithm computes the disparity for all views of the light field in the given time frame or only for the center view.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>accuracy</th>
<th>speed</th>
<th>all views</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_CH_G</td>
<td>1.01</td>
<td>slow</td>
<td></td>
</tr>
<tr>
<td>EPI_C</td>
<td>1.04</td>
<td>medium</td>
<td>yes</td>
</tr>
<tr>
<td>EPI_S</td>
<td>1.07</td>
<td>fast</td>
<td>yes</td>
</tr>
<tr>
<td>ST_AV_G</td>
<td>1.12</td>
<td>slow</td>
<td></td>
</tr>
<tr>
<td>ST_CH_S</td>
<td>1.14</td>
<td>fast</td>
<td></td>
</tr>
<tr>
<td>EPI_G</td>
<td>1.18</td>
<td>slow</td>
<td></td>
</tr>
<tr>
<td>ST_AV_S</td>
<td>1.19</td>
<td>fast</td>
<td></td>
</tr>
<tr>
<td>EPI_L</td>
<td>1.64</td>
<td>fast</td>
<td>yes</td>
</tr>
<tr>
<td>ST_AV_L</td>
<td>2.72</td>
<td>fast</td>
<td></td>
</tr>
<tr>
<td>ST_CH_L</td>
<td>3.54</td>
<td>fast</td>
<td></td>
</tr>
</tbody>
</table>
4.1.2. Multi-view stereo

We compute a simple local stereo matching cost for a single view as follows. Let $V = \{(s_1, t_1), \ldots, (s_N, t_N)\}$ be the set of $N$ view pairs corresponding images $I_1, \ldots, I_N$, with $(s_c, t_c)$ being the location of the current view $I_c$ for which the cost function is being computed. We then choose a set $\Lambda$ of 64 disparity labels within an appropriate range. For our test we choose equidistant labels within the ground truth range for optimal results. The local cost $\rho_{AV}(x, l)$ for label $l \in \Lambda$ at location $x \in I_c$ computed on all neighbouring views is then given by

$$\rho_{AV}(x, l) := \sum_{(s_n, t_n) \in V} \min(\varepsilon, \|I_n(x + l v_n) - I_c(x)\|),$$

where $v_n := (s_n - s_c, t_n - t_c)$ is the view point displacement and $\varepsilon > 0$ is a cap on the error to suppress outliers. To test the influence of the number of views, we also compute a cost function on a “crosshair” of view points along the $s$- and $t$-axis from the view $(s_c, t_c)$, which is given by

$$\rho_{CH}(x, l) := \sum_{(s_n, t_n) \in V} \max(0, \|I_n(x + l v_n) - I_c(x)\|).$$

In effect, this cost function thus uses exactly the same number of views as required for the local structure tensor of the center view. The results of these two purely local methods can be found under $ST_{AV \_L}$ for all views, and $ST_{CH \_L}$ for all views or just a crosshair, respectively.

Results of both multiview dataterms are denoised with a simple TV-$L^2$ scheme, algorithms $ST_{AV \_S}$ and $ST_{CH \_S}$. Finally, they were also integrated into a global energy functional

$$E(u) = \int_\Omega \rho(x, u(x)) \, dx + \lambda \int_\Omega |Du|,$$

for a labeling function $u : \Omega \rightarrow \Lambda$ on the image domain $\Omega$, which is solved to global optimality using the method in [PCBC10]. The global optimization results can be found under algorithms $ST_{AV \_G}$ and $ST_{CH \_G}$.

4.1.3. Results and discussion

At the moment of writing, the most accurate method over all data sets is the straight-forward global stereo $ST_{CH \_G}$. Interestingly, using only a subset of input views gives more accurate results after global optimization, while the data term accuracy is clearly worse. However, global stereo takes several minutes to compute. Among the real-time capable methods, $EPI_S$ and the ST$_S$ methods perform comparably well. The difference is that the stereo methods only compute the disparity map for the center view in real-time, while $EPI_S$ recovers disparity for all views simultaneously, which might be interesting for further processing. Furthermore, $EPI_S$ and $EPI_C$ do not discretize the disparity space, so accuracy is independent of computation time.

As usual, overall performance depends very much on parameter settings, for the results here, we did a parameter search to find the optimum for all methods on each data set separately. In the future, we would like to do a second run with equal parameter values for each data set, which might turn out quite interesting. These results will also be available online.

4.2. Multiple layer estimation

One of our data sets is special in that it contains a transparent surface, so there are two disparity channels, one for the transparent surface and one for the object behind it, see figure 3. For this case, we currently only have one method implemented [Ano13], and we are quite interested in whether it can be done better.
4.3. Light field labeling algorithms

For labeling, we have implemented all algorithms which are described in [WSG13]. The results are equivalent, and we refer to this work for details. Test scripts to re-generate all their results are included with the source code.

5. Conclusion

We have introduced a new benchmark database of densely sampled light fields for the evaluation of light field analysis algorithms. The database consists of two categories. In the first category, there are artificial light fields rendered with Blender [Ble], which provide ground truth disparities for all views. For some of these light fields, we additionally provide ground truth labels for object segmentation. In the second category, there are real-world scenes captured using a single camera mounted on a gantry, for which we provide (partial) ground truth disparity data for the center view, generated via fitting a mesh from a structured light scan to the views. A large contribution is the source code for an extensive evaluation of reference algorithms from multi-view stereo, epipolar plane image analysis and segmentation on the entire database.

The paper only describes a current snapshot for the database, and explains only a subset of the available source code, which offers many more optimization models and data terms. Both will be regularly updated, in particular we plan new light fields recorded with a plenoptic camera and with available ground truth depth data, as well as extensions of the code base with reference implementations of more multi-view reconstruction algorithms.

6. Acknowledgement

We thank Julie Digne from the LIRIS laboratory of the Claude-Bernard University for providing multiple of the scanned objects. We also thank Susanne Krömker from the Visualization and Numerical Geometry Group of the IWR, University of Heidelberg as well as the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp) for providing us with high precision 3D scans and support.

References


