# Summary Page

**Do not exceed one page. Failure to include the summary page will result in the automatic rejection of the paper.**

Answer each of the following questions in no more than 2-3 sentences:

1. Is this a system paper or a regular paper?

   **This is a regular (evaluation) paper.**

2. If it is a system paper, please explain the contribution.

   **N/A.**

3. If it is a regular paper:

   (a) What is the main contribution in terms of theory, algorithms and approach.

   **Our main contribution is to develop a new collection of test sequences and metrics for evaluating optical flow algorithms. We will be distributing the data freely and will have a database website up before the conference with scoring scripts and upload capabilities. We include a preliminary version as supplemental material.**

   (b) Describe the types of experiments and the novelty of the results. If applicable, provide comparison to the state of the art in this area.

   **We are the first paper since Barron *et al*. [2], Otte *et al*. [15], and McCane *et al*. [12] to provide a significant new dataset and evaluation methodology for optical flow. Our dataset includes a variety of different sequences, both real and synthetic, of both rigid and non-rigid scenes, and with both dense flow ground-truth and frame interpolation ground-truth. We compute a wide variety of different error measures and statistics over a variety of different image regions.**

# A Database and Evaluation Methodology for Optical Flow

Anonymous ICCV submission

Paper ID 1675

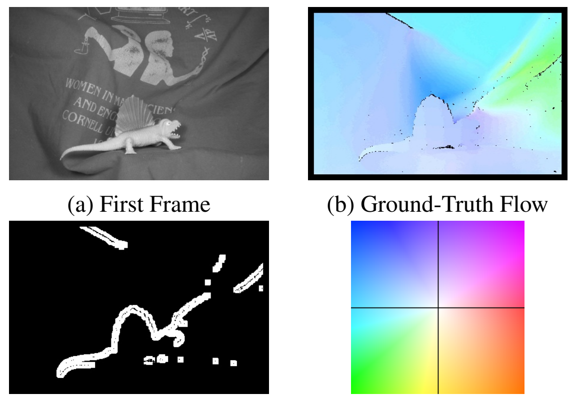
## Abstract

*The quantitative evaluation of optical flow algorithms by Barron* et al. *led to significant advances in the accuracy and robustness of optical flow methods. The challenges for optical flow today go beyond the datasets and evaluation methods proposed in that paper and center on problems associated with non-rigid motion, real sensor noise, complex natural scenes, motion boundaries, and frame interpolation. Our goal is to establish a new set of benchmarks and evaluation methods for the next generation of optical flow algorithms. To that end, we contribute four types of high-quality datasets to test different aspects of subpixel-accurate motion estimation algorithms: sequences with non-rigid motion where the ground-truth flow is determined by tracking hidden fluorescent texture; realistic synthetic sequences; high frame-rate video used to study interpolation error; and modified stereo sequences of static scenes. In addition to the average angular error used in Barron* et al.*, we compute the absolute flow endpoint error, frame interpolation error, improved statistics, and flow accuracy at motion boundaries and in textureless regions. We evaluate the performance of several well-known methods on this data to establish the current state of the art. We will make the database freely available on the web, together with scoring scripts and results upload capabilities.*

## 1. Introduction

As a sub-field of computer vision matures, datasets for quantitatively evaluating algorithms are essential to ensure continued progress. Many areas of computer vision, such as stereo [18], face recognition [16], and object recognition [8], have challenging datasets to track the progress made by leading algorithms, and to stimulate new ideas.

Optical flow was actually one of the first areas to have such benchmark datasets for quantitative comparison [2]. The field benefited greatly from this study, which led to rapid and measurable progress. When the Barron *et al.* [2] evaluation first appeared, the state of the art was quite poor and most algorithms had difficulty with even simplest sequences. Today the story is quite different. Numerous flow



(a) First Frame (b) Ground-Truth Flow

(c) Discontinuity Mask (d) Flow Field Color Coding

Figure 1. **Dimetrodon:** An illustrative example of one of the 4 types of data in our database. The dense ground truth for this non-rigid scene was obtained using hidden fluorescent texture. See Section 3.1 for the details. (a) The first image. (b) Ground truth flow field. (c) Motion discontinuity mask. (d) Flow color coding.

algorithms are in regular use and performance on the classic ground truth datasets such as the **Yosemite** sequence have largely saturated. State of the art algorithms obtain average angular errors (AAE) of less than $2.0°$ (equivalent to around $0.1$ pixels) with essentially no outliers.

To continue this rapid progress, new and more challenging datasets are needed to push the limits of current technology, reveal where current algorithms fail, and evaluate the next generation of optical flow algorithms. Such an evaluation dataset for optical flow should ideally consist of complex real (or photo-realistic) scenes with all the artifacts of real sensors (noise, motion blur, *etc*.). They should also contain substantial motion discontinuities as well as non-rigid motion. Of course, the image data must be paired with dense, subpixel accurate, ground-truth flow fields.

The presence of non-rigid or independent motion makes collecting a ground-truth dataset for optical flow far harder than for stereo, say, where structured-light [18] or range-scanning [20] can be used to obtain dense, pixel-accurate ground truth. Our solution is to collect four different datasets (Figure 1 illustrates one of the datasets), each of which satisfies a different subset of the desirable properties described above. The combination of these datasets provides a basis for a rigorous evaluation of current optical flow algorithms. Moreover, the relative performance of algorithms on the different sequence may stimulate further research in the field.

In particular, we collected the following data:

1. Real imagery of non-rigidly moving scenes where the dense ground truth is obtained using hidden fluorescent texture painted on the scene. We slowly move the scene, at each point capturing separate test images (in visible light) and ground-truth images (in UV light). Note that Tappen *et al*. [26] recently used certain wavelengths to hide ground truth for intrinsic images.
2. Realistic synthetic imagery. We address the limitations of sequences such as **Yosemite** [2] by rendering more complex scenes with significant motion discontinuities, textureless regions, motion blur, *etc*.
3. Imagery for frame interpolation where intermediate frames are withheld and used as ground truth. In a wide class of applications such as novel view generation and motion-compensated compression, what is important is not how well the flow field matches the ground-truth motion, but how well intermediate frames can be predicted using the flow [25]. We include this component to be more predictive of such tasks.
4. Real stereo imagery of rigid scenes where dense ground truth is captured using the procedures in [18, 19]; these datasets are modified for the purposes of optical flow.

We also extend the set of performance measures and the evaluation methodology to focus attention on current algorithmic problems. We evaluate a number of common flow algorithms to characterize the current state of the art. Our focus is on developing the database and evaluation methodology; the particular comparisons provided here are in no way meant to be exhaustive or fully representative of the state of the field. They do however, highlight many limitations of current techniques and challenges for the field.

To foster a community and enable continued development the database will be placed on the World Wide Web (before the conference) with scripts allowing researchers to evaluate their algorithms and upload their results, as in the Middlebury stereo datasets [18]. A preliminary version of this website is included in the supplemental material.

## 2. Related Work: Optical Flow Evaluation

A full review of optical flow algorithms is beyond the scope of this conference paper. Interested readers are referred to previous surveys by Aggarwal and Nandhakumar [1], Barron *et al*. [2], Otte and Nagel [15], Mitiche and Bouthemy [14], and Stiller and Konrad [22]. Instead we focus here on the evaluation of optical flow algorithms.

We must first define what we mean by optical flow. Following Horn's [10] taxonomy, the *motion field* is the 2D projection of the 3D motion of surfaces in the world, whereas the *optical flow* is the *apparent motion* of the brightness patterns in the image. These two are not always the same and, in practice, the goal of optical flow recovery is application dependent. In frame interpolation ("slow–mo"), it may be preferable to estimate apparent motion so that, for example, specular highlights move in a realistic way. In this paper we present two kinds of ground-truth; ground truth motion fields and intermediate images for the evaluation of apparent motion. We also assume that the true flow can be modeled by a single flow vector at each point in the scene; that is, we exclude transparency for now.

There have been three major previous attempts to quantitatively evaluate optical flow algorithms, each proposing sequences with ground truth. The work of Barron *et al*. [2] has been so influential that essentially all published methods today compare with it. The synthetic sequences used there are now too simple, however, to make meaningful comparisons between modern algorithms. Otte and Nagel [15] introduced ground truth for a real scene consisting of polyhedral objects. While this provided "real" image data, the images were still extremely simple. Most recently McCane *et al*. [12] provided more ground truth for real polyhedral scenes as well as graphics scenes of various levels of realism.

In this paper, we go beyond these studies in several important ways: First, we provide ground truth motion for much more complex real and synthetic scenes. Specifically we include ground truth for scenes with non-rigid motion. Second, we provide ground truth motion boundaries and extend the evaluation methods to these areas where many flow algorithms fail. Finally, we provide a web-based interface which will facilitate ongoing comparison of methods.

Our goal is to push the limits of current methods and, by exposing where and how they fail, focus attention on the hard problems. In general all flow algorithms have some matching criterion, some method for combining measurements spatially, and some optimization algorithm for computing the flow field. Regardless of which matching criteria and optimization algorithms are chosen, optical flow algorithms must somehow deal with all of the phenomena that make the problem intrinsically ambiguous and difficult. These include the aperture problem, textureless regions, motion discontinuities, occlusions, large motions, small objects, non-rigid motion, mixed pixels, changes in illumination, non-Lambertian reflectance, motion blur, and camera noise. Our goal is to provide ground truth data containing all these components and to provide information about their location in images. By so doing we can evaluate which phenomena pose problems for which methods.

ICCV #1675

Figure 2. Our setup for obtaining ground-truth flow using hidden fluorescent texture, including computer-controlled lighting and motion stages for camera and scene. The small images show visible light illumination (top row) and UV illumination (bottom row); the middle column shows the high-resolution images taken by the camera, and the right column shows a zoomed portion. The high-frequency fluorescent texture in the UV images allows accurate tracking, but is largely invisible in the low-resolution test images (see Figure 1a).

## 3. Database Design

Collecting a ground-truth database for optical flow is difficult. For stereo, structured-light [18] or range-scanning [20] can be used to obtain dense, pixel-accurate ground-truth. For optical flow, the scene may be moving non-rigidly making such techniques inapplicable in general. Ideally we would like imagery collected in real-world scenarios with real cameras, which furthermore contains substantial non-rigid motion. We would also like dense, subpixel accurate ground-truth. Unfortunately, we are not aware of a practical technique that can be used to satisfy all of these goals.

Rather than collecting a single benchmark dataset (with its inherent limitations) we instead collect four different sets, each satisfying a different subset of desirable properties. As we will see, the relative performance of algorithms on the different types of data is itself interesting and may provide insights into future algorithm development.

### 3.1. Dense GT Using Hidden Fluorescent Texture

We have developed a technique for capturing imagery of non-rigid scenes with ground-truth flow. We build a scene that can be moved in very small steps by a computer-controlled motion stage. We apply a fine spatter pattern of fluorescent paint to all surfaces in the scene. The computer repeatedly takes a pair of high-resolution images both under ambient lighting and under UV lighting, and then moves the scene (and possibly the camera) by a small amount.

In our current setup, shown in Figure 2, we use a Canon EOS 20D camera to take images of size $3504\times2336$, and make sure that no scene point moves by more than 2 pixels from one frame to the next. We obtain our test sequence by downsampling every 20th image taken under visible light by a factor of 8, yielding images of size $438\times292$, with a maximum motion of about 5 pixels between frames.

Since fluorescent paint is available in a variety of colors, the color of the objects in the scene can be closely matched. In addition, it is possible to apply a fine spatter pattern, where individual droplets are about the size of 1–2 pixels in the high-resolution images. This high-frequency texture then effectively disappears in the low-resolution images, while the fluorescent paint is very visible in the high-resolution UV images (see Figure 2, rightmost column).

The ground-truth flow is computed by tracking small windows in the sequence of high-resolution UV images. We use a simple sum-of-absolute-difference (SAD) tracker with a window size of $15\times15$, corresponding to a window diameter of less than 2 pixels in the downsampled images. We perform a brute force search and use each frame to initialize the next. We also crosscheck the results by tracking each pixel both forwards and backwards through the sequence and require perfect correspondence. The chances that this check would yield false positives after tracking for 20 frames are very low. Crosschecking identifies the occluded regions, whose motion we mark as "unknown"; it also helps identifying regions with insufficient texture, which we can eliminate by applying more paint.

Using this combination of fluorescent paint, downsampling high resolution images, and sequential tracking of small motions, we are able to capture dense ground-truth for a non-rigid scene. The main limitations of our approach are (1) it can only be applied in a lab setting with controlled lighting and motion, (2) it does not capture effects such as motion blur, and (3) the accuracy of the flow field is restricted to 1/8=0.125 pixels in the downsampled sequence (although subpixel tracking could be used in the high resolution sequence to improve on this accuracy.)

We include two sequences in our database. **Dimetrodon** contains non-rigid motion and large areas with little texture. One image and the color-coded ground-truth flow are included in Figure 1. **Seashell** contains several objects undergoing independent motion and is illustrated in Figure 3.

Figure 3. **Seashell:** A second example of a sequence captured using hidden fluorescent texture. We display the first frame (left) and the color-coded (see Figure 1) ground-truth flow (right).

### 3.2. Realistic Synthetic Imagery

Synthetic scenes generated using computer graphics are often indistinguishable from real ones. For the study of optical flow, synthetic data offers a number of benefits. In particular, it provides full control over the rendering process and allows us to explore different sources of "noise" and their effects on flow algorithms. For example, we can generate scenes with varying amounts of motion blur to asses whether performance degrades with increasing blur. It also allows control over the material properties of the objects and provides precise ground truth motion and object boundaries.

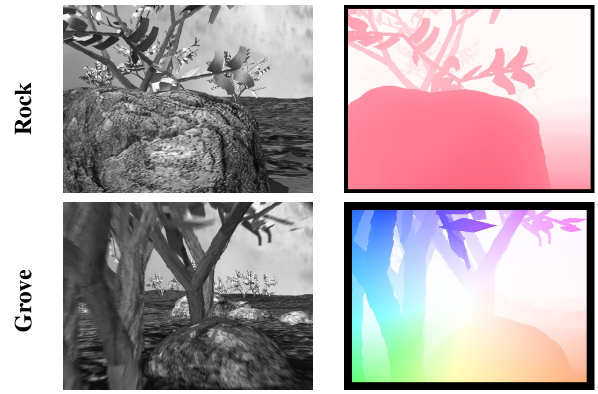
To go beyond previous synthetic groundtruth (*e.g.*, the **Yosemite** sequence) we generated fairly complex synthetic outdoor scenes with significant occlusion and a wide range of camera motions (see Figure 4). The scenes contain a random number of procedurally generated "rocks" (up to 40) and "trees" (up to 25) with randomly chosen ground texture and surface displacement. Additionally, the tree bark has significant 3D texture. The scenes are rigid and the camera motions include camera rotation and 3D translation.

These scenes were generated using the Mental Ray renderer [7]. The camera motion is sampled at the virtual shutter open and close times and hence is assumed linear during the open shutter interval. The virtual shutter is open for the full interval between frames (corresponding to a 360 degree shutter angle in film camera terminology). The scenes are computed at a resolution of 640x480 using linear gamma. Current rendered scenes do not include inter-reflections.

The ground truth was computed using a custom renderer ("lens shader" plugin) which projects the 3D motion of the scene corresponding to a particular image onto the 2D image plane. The resulting 2D flow vectors represent the motion of an image point from shutter open to shutter close.

### 3.3. Imagery for Frame Interpolation

In a wide class of applications such as novel view generation and motion-compensated compression, what is important is not how well the flow field matches the ground-truth motion, but how well intermediate frames can be predicted using the flow. To allow for measures that predict performance on such tasks we collected a variety of data suitable for frame interpolation. The relative performance of algorithms with respect to frame interpolation and ground-truth motion estimation is interesting in its own right.




(a) First Frame (b) Ground-Truth Flow

Figure 4. We include two synthetically generated sequences, **Rock** and **Grove:**. These sequences contain substantial motion discontinuities, motion blur, and larger motions that the **Yosemite** sequence. See Figure 1 for a the color coding of the flow.

We used a PointGrey Dragonfly Express to capture a number of sequences, acquiring frames at 100 frames per second. We provide every 4th image to the optical flow algorithms (*i.e.* 25Hz) and retain the remaining intermediate frames as ground truth for evaluating frame interpolation. This temporal subsampling means the input to the flow algorithms is captured at roughly the standard 25-30Hz while enabling generation of a $4\times$ slow-motion sequence.

We included 2 such sequences in the database: **Phone** and **Crumple.** In Figure 5 we show the first and second frames for these two sequences. We emphasize that there is no ground-truth *motion* for these sequences, only ground truth image data. In addition to this high-speed camera data, we also use some of the other other sequences for frame interpolation. We retain the middle frames for the hidden texture sequences **Dimetrodon** and **Seashell**, and so also compute the frame interpolation error for them. We also retain the middle image of the **Venus** and **Moebius** sequences described in the following section for the same purpose.

### 3.4. Modified Stereo Data for Rigid Scenes

Our final dataset consists of modified stereo data. Specifically we use the **Venus** dataset obtained by registering planes in the scene [18], and the **Moebius** dataset [17], which was obtained using the structured lighting technique of [19]. These datasets have an asymmetric disparity range $[0, d_{\max}]$ that is appropriate for stereo, but not for optical flow. We crop different subregions to convert this disparity range to $[-d_{\max}/2, d_{\max}/2]$ (see Figure 6). One benefit of using this modified stereo data is that it allows a comparison with state of the art stereo algorithms. Shifting the disparity range does not affect the performance of stereo algorithms so long as they are given the appropriate search range. One concern with this data is that algorithms may


(a) 1st Frame ($t = 0$) (b) 2nd Frame ($t = 4$ms)

Figure 5. The sequences **Phone** and **Crumple** are captured with a PointGrey Dragonfly Express camera at 100Hz. We provide every 4th frame to the optical flow algorithms (equivalent to 25hz.) The intermediate frames are retained as interpolation ground-truth.

take advantage of the knowledge that motions are horizontal. We could counteract this by adding additional non-rigid motion in future versions of the database.

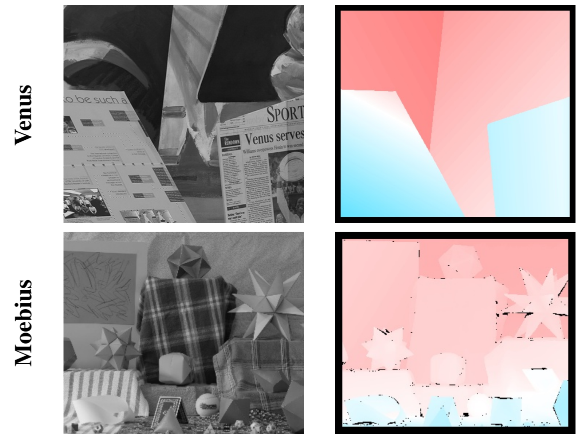
# 4. Evaluation Methodology

We refine and extend the evaluation methodology of [2] in terms of (1) the performance measures used, (2) the statistics computed, (3) the regions of the images computed over, and (4) the use of the World Wide Web for data distribution, results scoring, and results dissemination.

## 4.1. Performance Measures

The most commonly used measure of performance for optical flow is the angular error (AE). The AE between two flows $(u_0, v_0)$ and $(u_1, v_1)$ is the angle in 3D space between $(u_0, v_0, 1.0)$ and $(u_1, v_1, 1.0)$. The AE is usually computed by normalizing the vectors, taking the dot product, and then and then taking the inverse cosine of their dot product. The popularity of this measure is based on the seminal survey by Barron *et al.* [2], although the measure itself dates to prior work by Fleet and Jepson [9]. The goal of the AE is to provide a *relative* measure of performance that avoids the "divide by zero" problem for zero flows. Errors in large flows are penalized less in AE than errors in small flows. To this relative measure, we add the most natural *absolute* error; that is, the error in flow endpoint (EP) defined by $\mathrm{sqrt}[(u_0 - u_1)^2 + (u_0 - u_1)^2]$ as used in [15]. For many applications, endpoint error is probably more appropriate.

For image interpolation, we use the (square root of the) SSD between the ground-truth image and the estimated interpolated image. We also include a gradient-normalized SSD inspired by [25]. The (square root of the) normalized SSD between an interpolated image $I(x, y)$ and a ground-




(a) First Frame (b) Ground-Truth Flow

Figure 6. We cropped the stereo datasets **Venus** [18] and **Moebius** [17] to convert the asymmetric stereo disparity ranges into roughly symmetric flow fields. One additional reason for including this dataset was to allow direct comparison with state of the art stereo algorithms. See Figure 1 for the color-coding of the flows.

truth image $I_{\mathrm{GT}}(x, y)$ is given by:

$$\left[ \sum_{(x,y)} \frac{(I(x,y) - I_{\mathrm{GT}}(x,y))^2}{\|\nabla I_{\mathrm{GT}}(x,y)\|^2 + \epsilon} \right]^{\frac{1}{2}}. \tag{1}$$

In our experiments $\epsilon = 1.0$.

Naturally, an interpolation algorithm is required to generate the interpolated image from the optical flow field. In this paper, we use the baseline algorithm briefly described in Appendix A. Note that one area for future work is to develop better frame interpolation algorithms. We hope that our database can be used both by researchers working on optical flow and on frame interpolation algorithms.

## 4.2. Statistics

Although the full histograms are available in a longer technical report, Barron *et al.* [2] report averages (AV) and standard deviations (SD) of the error measures. This has led most subsequent researchers to only report these statistics. We also compute the popular robustness statistics used in the Middlebury stereo dataset [18]. In particular $\mathrm{R}X$ denotes the percentage of pixels that have an error measure above $X$. For AEs we compute R1.0, R3.0, and R5.0 (degrees). For EP errors we compute R0.1, R0.5, and R1.0 (pixels). For the SSD interpolation error and the normalized version of it, we compute R0.5, R1.0, and R2.0 (grey levels). We also compute robust accuracy measures similar to those in [20]: $\mathrm{A}X$ denotes the accuracy of the error measure at the $x^{\mathrm{th}}$ percentile. For all measures (AE, EP, SSD, and normalized SSD) we report A50, A75, and A95.

### 4.3. Region Masks

It is easier to compute flow in some parts of an image than in others. For example, computing flow around motion discontinuities is likely to be hard. Computing motion in textureless regions is also likely to be hard, although interpolating in those regions should be easier. Computing statistics over such regions may highlight areas where existing algorithms are failing and spur further research in these cases. We follow the procedure in [18] and compute the error measure statistics over 3 types of region masks: **all**, **motion discontinuities**, and **textureless regions**.

The **all** regions exclude boundary pixels around the edge of the image. Ideally we would like to include these pixels, but several of the algorithms that we tested had noticeable boundary effects. We did not want to penalize these legacy algorithms. In future versions of the data, we will include the boundary pixels. We did not remove semi-occluded pixels in the motion ground-truth datasets because we believe algorithms should be able to extrapolate into these regions. For the interpolation ground-truth, we did exclude these regions because the baseline interpolation algorithm does not reason about these areas. The **motion discontinuities** mask was computed by taking the gradient of the ground-truth flow field, thresholded the magnitude, and then dilated the resulting mask. If the ground-truth flow is not available, we used frame differencing to get an estimate of fast moving regions instead. The **textureless** regions were computed by taking the gradient of the image, thresholding, and dilating.

### 4.4. Distribution, Evaluation, and Dissemination

An important part of our evaluation methodology is to make the database freely available to researchers on the World Wide Web. Equally important, we will provide online scoring scripts and the ability for researchers to upload their scores, as in the Middlebury stereo evaluation [18].

## 5. Experimental Results

Our goal in this paper is to provide a set of baseline results to define the state of the art on the database and to allow researchers to get a sense of what is good performance on the data. To this end, we compared 5 algorithms:

**Pyramid LK:** An implementation [5] of the Lucas-Kanade algorithm [11] on a pyramid, subsequently refined in our lab. This implementation performs significantly better than the Lucas-Kanade code in Barron *et al.* [2]. It is included to give an idea of how the algorithms in [2] perform when implemented to today's standards.

**Black and Anandan:** We used Michael Black's implementation of this algorithm [4], with their default parameter settings.

**Bruhn *et al.*:** We implemented this highly regarded recent algorithm [6] (2D-CLG) ourselves, and tuned the parameters to (roughly) reproduce the results obtained by that algorithm on the **Yosemite** sequence (included in the results webpage.) Eventually we hope to replace these results with those obtained by the authors of this paper themselves.

**MediaPlayer$^{\mathrm{TM}}$:** As a baseline for interpolation we obtained results using the real-time flow algorithm used in Microsoft MediaPlayer 9 for video smoothing [13].

**Zitnick *et al.*:** We used the author's implementation of this algorithm [28] that uses consistent segmentation.

We have included the results for all of these algorithms in the form of a set of webpages in the supplemental material. We include results for all 4 measures (AE, EP, SSD, and normalized SSD), all the statistics, and for the 3 different masks. Mousing over any of the numbers pops up the estimated flow or interpolated image, and the error from the ground-truth. A screen shot of one of these pages is included in Figure 7 (left). The long-term goal of this work is to determine the best optical flow algorithms. We expect that to happen once researchers test their algorithms and upload their results to the database website. To this end, we will add the ability for researchers to score their algorithms and upload their result.

A preliminary analysis of the results points to the following major conclusions:

**Difficulty:** The data is considerably more challenging than **Yosemite**. For example, the AAEs for the Bruhn *et al.* algorithm are **Yosemite** 1.69, **Dimetrodon** 10.99, **Seashell** 11.09, **Venus** 8.73, **Moebius** 5.85, **Rock** 6.14, **Grove** 6.32. The disparity in performance around the motion discontinuities is higher still.

**Diversity:** There is substantial variation in difficulty across the datasets. For example, the average endpoint errors for the Black and Anandan algorithm are **Yosemite** 0.15, **Rock** 0.22, **Seashell** 0.30, **Dimetrodon** 0.39, **Venus** 0.55, **Moebius** 1.02, **Grove** 1.50. There is both variability across datatypes (hidden fluorescent texture, synthetic, and modified stereo), and within those types. This diversity is desirable because it means that as technology matures, some subset of the data will be at the appropriate level of difficulty. Moreover, the within type diversity indicates that we will be able to extend the dataset with a wide spread of difficulties by varying the complexity of the scene, and in the case of the synthetic data, the various rendering parameters.

**Region Masks:** A related point concerns the region masks. For the stereo datasets (**Venus** and **Moebius**) the untextured regions are not significantly more difficult than the textured regions. This is consistent with results obtained by stereo algorithms [18]. On the other hand, the results for the hid-


| Average angle error | avg. rank | Dimetrodon (Hidden texture) all | disc | untext | Seashell (Hidden texture) all | disc | untext | Rock (Synthetic) all | disc | untext | Grove (Synthetic) all | disc | untext | Yosemite (Synthetic) all | disc | untext | Venus (Stereo) all | disc | untext | Moebius (Stereo) all | disc | untext |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bruhn et al. | 1.6 | 10.99 3 | **9.41** 1 | 14.22 3 | 11.09 2 | 19.48 2 | 16.21 2 | **6.14** 1 | **17.41** 1 | 12.86 2 | **6.32** 1 | **12.41** 1 | **10.98** 1 | **1.69** 1 | **2.86** 1 | **1.05** 1 | 8.73 2 | 31.46 2 | 8.15 2 | **5.85** 1 | 10.12 2 | 8.80 2 |
| Black and Anandan | 2.1 | **9.26** 1 | 10.11 3 | **12.08** 1 | 11.20 3 | 19.83 3 | 17.01 3 | 7.67 3 | 18.44 3 | 16.80 4 | 7.89 2 | 13.55 2 | 13.96 4 | 2.65 2 | 4.18 2 | 1.88 2 | **7.64** 1 | **30.13** 1 | **7.31** 1 | 7.05 2 | **10.02** 1 | **8.41** 1 |
| Pyramid LK | 2.8 | 10.27 2 | 9.71 2 | 13.63 2 | **9.46** 1 | **18.62** 1 | **12.07** 1 | 6.53 2 | 18.43 2 | **10.95** 1 | 8.14 3 | 15.08 3 | 12.78 2 | 5.22 3 | 6.64 3 | 4.29 3 | 14.61 4 | 36.18 4 | 24.67 5 | 12.98 5 | 13.85 4 | 20.61 5 |
| MediaPlayer™ | 4.1 | 15.82 4 | 26.42 4 | 16.96 4 | 23.18 4 | 27.71 5 | 21.78 4 | 9.44 4 | 22.25 4 | 15.03 3 | 10.99 4 | 18.15 5 | 13.64 3 | 11.09 4 | 17.16 4 | 10.66 5 | 15.48 5 | 43.56 5 | 15.09 4 | 9.98 4 | 15.04 5 | 9.47 3 |
| Zitnick et al. | 4.2 | 30.10 5 | 34.27 5 | 31.58 5 | 29.07 5 | 27.55 4 | 21.78 4 | 12.38 5 | 23.93 5 | 17.59 5 | 12.55 5 | 15.56 4 | 17.35 5 | 18.50 5 | 28.00 5 | 9.41 4 | 11.42 3 | 31.46 2 | 11.12 3 | 9.88 3 | 12.83 3 | 11.28 4 |

Figure 7. Left: A screen shot of one of the results webpages. This page shows the average angular error (AAE). The user can also select any of the other error metrics in Section 4.1 or any of the other statistics in Section 4.2. We display separate columns for each of the region masks. Mousing over any of the links bring up the flow image and error for the corresponding algorithm and dataset. Right: An explanation of the uncorrelated results in terms of ground-truth motion error and interpolation error. MediaPlayer$^{\rm TM}$ tends to overly extend the flow into textureless regions such as above the paper. Because these regions are textureless the interpolation error is not significantly affected.

den fluorescent texture (**Dimetrodon** and **Seashell**) and the synthetic data (**Rock** and **Grove**) show the textureless regions to be significantly more difficult. It is possible that the implicit assumptions constant or smooth flow in non-rigid scenes are less valid than the corresponding assumptions of planarity or constant disparity for stereo.

Perhaps unsurprisingly, performance around motion discontinuities is generally significantly worse than over the entire image. The notable exception is **Dimetrodon** where the major difficulty is the complex non-rigid flow in the textureless regions. This sequence is appropriate for researchers investigating these problems in isolation from discontinuities.

**Motion vs. Frame Interpolation Error:** As measured by average rank, the best performing algorithms for the motion ground-truth are Bruhn *et al.* and Black and Anandan. For the interpolation algorithm the Pyramid LK algorithm is the best. The results for MediaPlayer$^{\rm TM}$ are also significantly better for interpolation than for ground-truth motion. An explanation for this is illustrated in Figure 7 (right). MediaPlayer$^{\rm TM}$ tends to overly extend the flow into textureless regions such as above the paper. However, because these regions are textureless the interpolation error is not significantly affected. Because it does not need to be so careful in such regions the interpolation error can be improved elsewhere by increased regularization. For a visual assessment of the interpolation quality, please see the movies included with the supplementary materials.

**Comparison with Stereo:** The robustness results $\mathrm{R}1.0$ for **Venus** allow a comparison with stereo. The best performing stereo algorithms achieve an $\mathrm{R}1.0$ score of around $0.2$–$1.0$, whereas the best performing optical flow algorithm achieves $9.35$ (Bruhn *et al.*). Note, however, that the stereo algorithms use the epipolar constraint, which gives them a significant advantage. In addition, most stereo methods use color information, whereas all of the imagery in our dataset is greyscale. Zitnick *et al.*, which is similar in spirit to many segmentation-based stereo algorithms, performs relatively poorly overall. One reason might be the lack of color information to drive the segmentation. Another might be the focus in optical flow in sub-pixel accuracy, compared to the focus in stereo of robustness in labeling discrete disparities.

## 6. Conclusion

We have presented a collection of datasets for the evaluation of optical flow algorithms. Preliminary results show the data to be challenging and internally diverse, which facilitates interesting comparisons and insights. We have also extended the set of evaluation metrics and improved the evaluation methodology. Amongst other things, this allows an interesting comparison with stereo algorithms. As other researchers use the datasets, it should lead to a far better understanding of the relative performance of existing algorithms, and suggest interesting new directions for research.

We do not intend this dataset to be static. We will continue to extend the database by collecting additional sequences and refining the collection procedures. We restricted the current datasets to pairs of greyscale images because that is the "lowest common denominator" amongst current algorithm implementations. In future versions we plan to provide extended color sequences to see whether the additional channels and frames can be used productively.

In terms of the synthetic sequences we will add realistic sensor noise (including CCD radiometric falloff) to study the effects of noise on flow estimation. We will also add independently moving objects, use more complex reflectance models, and add transparency.

Another future direction is in terms of interpolation algorithms. The baseline algorithm that we used could be significantly improved if we had layering or depth information. We encourage authors to develop their own interpolation algorithms and submit interpolated images for direct comparison with the ground-truth, for example by looking at more than pairs of frames to estimate motion [23, 24].

## References

[1] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images—a review. *Proceedings of the IEEE*, 76(8):917–935, 1988. 2

[2] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994. 1, 2, 5, 6

[3] T. Beier and S. Neely. Feature-based image metamorphosis. *SIGGRAPH*, 26(2):35–42, 1992. 8

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 63(1):75–104, 1996. 6

[5] J. Bouguet. Pyramidal implementation of the Lucas-Kanade feature tracker: description of the algorithm. Technical report, OpenCV Document, Intel Microprocessor Research Labs, 2000. 6

[6] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *IJCV*, 61(3):211–231, 2005. 6

[7] T. Driemeyer. *Rendering with mental ray*. Springer-Verlag New York, Inc., New York, NY, USA, 2001. 4

[8] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *PAMI*, 28(4):594–611, 2006. 1

[9] D. Fleet and A. Jepson. Computation of component image velocity from local phase information. *IJCV*, 5:77–104, 1990. 5

[10] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, Massachusetts, 1986. 2

[11] B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In *IJCAI*, pages 674–679, 1981. 6

[12] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *CVIU*, 84:126–143, 2001. 1, 2

[13] Microsoft Corporation. Media player 9 video quality demos. *http://www.microsoft.com/windows/windowsmedia/demos/video_quality_demos.aspx*. 6

[14] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *IJCV*, 19(1):29–55, 1996. 2

[15] M. Otte and H.-H. Nagel. Optical flow estimation: advances and comparisons. In *ECCV*, 1994. 1, 2, 5

[16] P. Philips et al. FRVT 2006 and ICE 2006 large-scale results. Technical Report NISTIR 7408, National Institute of Standards and Technology, 2007. 1

[17] D. Scharstein and C. Pal. Learning conditional random fields for stereo. In *CVPR*, 2007. 4, 5

[18] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1–3):7–42, 2002. 1, 2, 3, 4, 5, 6

[19] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2003. 2, 4

[20] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 1, 3, 5

[21] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *SIGGRAPH*, pages 231–242, 1998. 8

[22] C. Stiller and J. Konrad. Estimating motion in image sequences: A tutorial on modeling and computation of 2D motion. *IEEE Signal Proc. Magazine*, 16(4):70–91, 1999. 2

[23] S. Sun, D. Haynor, and Y. Kim. Motion estimation based on optical flow with adaptive gradients. In *ICIP*, 2000. 8

[24] R. Szeliski. A multi-view approach to motion and stereo. In *CVPR*, volume 1, pages 157–163, 1999. 8

[25] R. Szeliski. Prediction error as a quality metric for motion and stereo. In *ICCV*, pages 781–788, 1999. 2, 5

[26] M. Tappen, E. Adelson, and W. Freeman. Estimating intrinsic component images using non-linear regression. In *CVPR*, volume 2, pages 1992–1999, 2006. 2

[27] C. L. Zitnick et al. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608, 2004. 8

[28] C. L. Zitnick, N. Jojic, and S. B. Kang. Consistent segmentation for optical flow estimation. In *ICCV*, 2005. 6

# A. Frame Interpolation Algorithm

We briefly describe the algorithm used to compute all the interpolation results in this paper. Our algorithm takes a single flow field $\mathbf{u}_0$ and constructs an interpolated frame $I_t$ that is a temporal distance $t \in (0, 1)$ between the first and second frames $I_0$ and $I_1$. We use both frames to generate the actual intensity values, as described below. In all the experiments in this paper $t = 0.5$. Our algorithm is closely related to previous algorithms for depth-based frame interpolation [21, 27] and performs the following steps:

1. Take the first flow field and *forward warp* (or *splat*) each flow value to the nearest destination pixel:

$$\mathbf{u}_t(\text{round}(\mathbf{x} + t\mathbf{u}_0(\mathbf{x}))) = \mathbf{u}_0(\mathbf{x}).$$

2. Fill in any holes in the extrapolated motion field $\mathbf{u}_t$. (We use a simple outside-in filling strategy.)

3. Fetch the corresponding intensity values from both the first and second image and blend them together [3],

$$I_t(\mathbf{x}) = (1-t) I_0(\mathbf{x} - t\mathbf{u}_t(\mathbf{x})) + t I_1(\mathbf{x} + (1-t)\mathbf{u}_t(\mathbf{x})).$$

Bilinear interpolation is used to sample the images.