



Published in Image Processing On Line on 2019-10-09.  
 Submitted on 2017-10-05, accepted on 2019-01-11.  
 ISSN 2105-1232 © 2019 IPOL & the authors CC-BY-NC-SA  
 This article is available online with supplementary materials,  
 software, datasets and online demo at  
<https://doi.org/10.5201/ipo1.2019.217>

# Comparison of Optical Flow Methods under Stereomatching with Short Baselines

Tristan Dagobert<sup>1</sup>, Nelson Monzón<sup>1</sup>, Javier Sánchez<sup>2</sup>

<sup>1</sup> CMLA, ENS Cachan, France ([tristan.dagobert@cmla.ens-cachan.fr](mailto:tristan.dagobert@cmla.ens-cachan.fr))

<sup>2</sup> CTIM, University of Las Palmas de Gran Canaria, Spain ([jsanchez@ulpgc.es](mailto:jsanchez@ulpgc.es))

*Communicated by* Gabriele Facciolo

*Demo edited by* Nelson Monzón and Tristan Dagobert

## Abstract

This article studies the effectiveness of optical flow methods applied to short baseline image pairs under different noise levels. New metrics have been developed to analyze the results because the usual metrics are inadequate in a subpixel context. We have used the implementation of some standard optical flow methods adapted to the stereo problem. Our experiments show that the Brox et al. method produces the least errors, with a 60% success rate and a relative precision at 1/100th of a pixel. On the other hand, our comparison shows that a discontinuity preserving method, derived from Brox et al., also provides competitive results at the same time that it yields disparities with more details and correct contours.

## Source Code

Source codes of Lucas-Kanade 1D, Robust Optical Flow 1D and Robust Discontinuity Preserving 1D algorithms are provided in [the web page of the article](#)<sup>1</sup>.

**Keywords:** stereovision; subpixel; metric; optical flow; ROC curve

## 1 Introduction

Stereoscopic vision plays an important role in many areas such as autonomous robots, remote sensing or industrial control processes. It relies on matching two views observed from two different sensors or one sensor in motion.

In the case of fronto-parallel camera displacements, which is the setting that we will use in the sequel, stereoscopic image pairs can always be rectified so that matching reduces to estimating movements  $\mathbf{d}$  (at each pixel) along the  $\vec{x}$  axis between two images  $u$  and  $v$

$$u(\mathbf{x}) = v(\mathbf{x} + \mathbf{d}), \quad (1)$$

where  $\mathbf{x} = (x, y)^T$  and  $\mathbf{d} = (d, 0)^T$ .

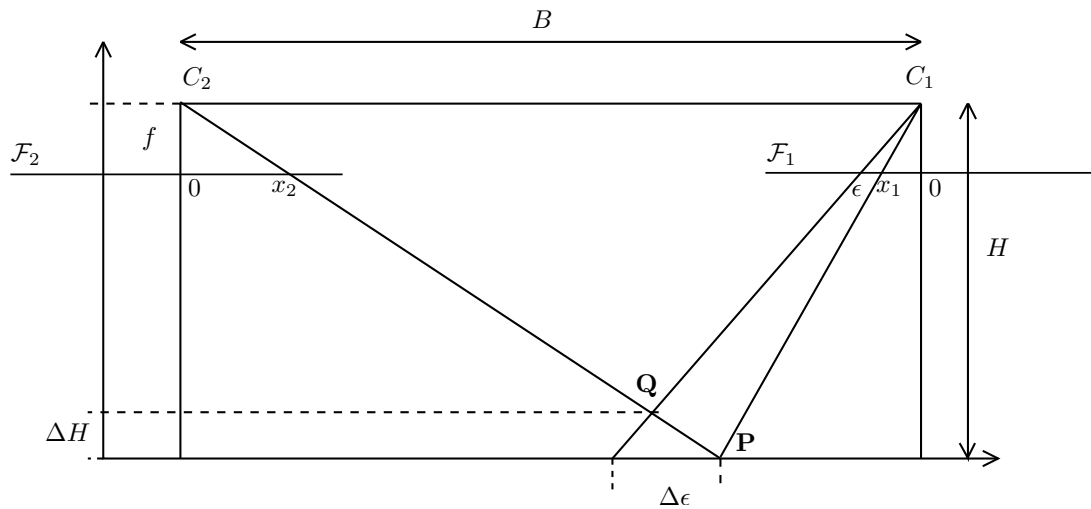


Figure 1: Estimate of the depth  $H$ , of a point  $\mathbf{P}$ , from two cameras  $C_1$  and  $C_2$ , fronto-parallelly shifted by a distance  $B$ . The knowledge of the focal length  $f$ , and the projections of  $\mathbf{P}$  on the focal planes  $\mathcal{F}_1$  and  $\mathcal{F}_2$ , through the principal points  $C_1$  and  $C_2$ , at points  $x_1$  and  $x_2$ , respectively, allows estimating the depth. When an error  $\epsilon$  is introduced in the estimate of  $x_2$  this yields a depth error  $\Delta H$  in the reconstruction of the 3D point  $\mathbf{Q}$ .

Delon and Rougé [6] and Sabater et al. [20, 22, 21] analyzed the sources of the 3D reconstruction imprecisions. The estimate of the triangulation accuracy is conventionally formulated from the pinhole camera model assumption [20, chap. 1], which relies on Thales' theorem. Based on the notation introduced in Figure 1, we have the following relations

$$\frac{B}{\Delta\epsilon} = \frac{H - \Delta H}{\Delta H} \text{ and } \frac{H}{f} = \frac{\Delta\epsilon}{\epsilon}, \quad (2)$$

which lead to

$$\epsilon = \frac{B\Delta H}{H - \Delta H} \frac{f}{H}. \quad (3)$$

When  $\Delta H$  is small compared to  $H$ , the error is approximated as

$$\epsilon \simeq \frac{B}{H} \cdot \frac{f}{H} \Delta H, \quad (4)$$

meaning that it linearly depends on the ratio  $B/H$ . In order to reduce the value of  $\Delta H$  for a given  $\Delta\epsilon$ , Equation (4) leads to choose stereo pairs acquired in long baselines. This is done at the expense of a loss of information due to occlusions, geometric deformations or illumination changes between the images. However, choosing short baselines alleviates these problems because both images become more similar, but may increase  $\Delta H$ . The works by Delon and Rougé [6] and Sabater [20] have theoretically shown the advantages of such an approach: the imprecision of the matching algorithms is mainly due to noise.

The purpose of this study is to compare the previous theoretical results with realistic cases in short baselines and for different levels of noise, in the context of fixed cameras. To that end, we compare several algorithms and evaluate their precision. On the one hand, this study implies the use of image databases with negligible noise levels and very accurate ground truths. Dagobert [5] found that commonly used databases poorly met these requirements and proposed a new one, the CMLA dataset. This database has the advantage of providing pairs of images created with different baselines, which have virtually no residual noise and dense disparity maps, whose accuracy is very high.

<sup>1</sup><https://doi.org/10.5201/ipol.2019.217>

On the other hand, the comparisons require the use of metrics taking into account the influence of different baselines and noise levels, as well as the density of the disparity maps. By density, we refer to the proportion of points for which there exists a solution that is close to the ground truth. That is the reason for using new metrics better suited for this study, instead of the metrics used by other evaluation platforms.

This article is organized as follows: Section 2 describes and justifies our assessment methodology and the new metrics employed. Section 3 describes the 1D optical flow algorithms employed in this study, in particular, the methods of Lucas and Kanade [13], Brox et al. [3] and Monzón et al. [17]. Section 4 deals with the set up of the experimental protocol. Then, we present the results obtained by the stereoscopic and optical flow algorithms, using the CMLA database and a few pairs from the Middlebury dataset [26]. Finally, we present the conclusions in Section 5.

## 2 Evaluation Methodology

### 2.1 Notation

We denote by  $D_{truth,l}$  the exact dense disparity map,  $O_{truth,l}$  the exact occlusion map defined by  $O_{truth,l}(\mathbf{x}) = 0$  if the pixel  $\mathbf{x} = (x, y)^T$  is visible in both images and  $O_{truth,l}(\mathbf{x}) = 1$ , otherwise. The exact sparse disparity map is then defined by

$$\mathring{D}_{truth,l}(\mathbf{x}) = \begin{cases} D_{truth,l}(\mathbf{x}) & \Leftrightarrow O_{truth,l}(\mathbf{x}) = 0 \\ NaN & \Leftrightarrow O_{truth,l}(\mathbf{x}) = 1, \end{cases} \quad (5)$$

with the convention  $D_{truth,l}(\mathbf{x}) = NaN \Leftrightarrow D_{truth,l}(\mathbf{x}) \notin \bar{\mathbb{R}}$ . Equivalently, we denote by  $D_l$  the estimated dense disparity map,  $\mathring{D}_l$  the sparse estimated disparity and  $O_l$  the estimated occlusion map obtained from the previous map by the following relation

$$O_l(\mathbf{x}) = \begin{cases} 1 & \Leftrightarrow \mathring{D}_l(\mathbf{x}) \notin \bar{\mathbb{R}}, \\ 0 & \Leftrightarrow D_l(\mathbf{x}) \in \bar{\mathbb{R}}. \end{cases} \quad (6)$$

In a similar way,  $D_r$ ,  $O_r$  are the maps associated with the right image  $v$ . The maps size is denoted  $N_I$ .

### 2.2 Classical Approach of Assessment

#### 2.2.1 The Middlebury Example

The definition of objective metrics to evaluate and rank the effectiveness of a particular stereo algorithm is a complex problem. The Middlebury benchmark database [27], which is the *de facto* standard in the evaluation of stereo algorithms, is very representative:

- It offers no less than forty metrics: ten formulas, such as success rates with four precision thresholds, RMSE, mean error and four quantiles of percentage of error;
- It contemplates two algorithmic properties. The first one based on density: the algorithm indicates, for each pixel, the disparity associated to the left image and denoted as  $D_l(\mathbf{x}) \in \mathbb{R}$ . The second one based on sparsity: the algorithm indicates a non numeric value at pixels where it cannot evaluate the disparity, i.e.  $D_l(\mathbf{x}) \notin \mathbb{R}$ ;
- It considers two configurations, depending on whether or not visibility or occlusion of pixels are taken into account in both views.

Among these metrics, as noted by Kostlivá et al. [12], the first family associated with a precision threshold of one pixel and the two aforementioned configurations are typically the most used. These precision criteria are very simple to use and give a relatively good idea about what an algorithm must do: to accurately estimate displacements, or disparities in the stereo context, and provide dense data.

### 2.2.2 Existing Assessment Metrics

Among the existing metrics, the success rate (or the mismatch rate) considered by Scharstein et al. [26] with a fixed a priori accuracy threshold  $s$  is widely adopted. Two categories of pixels are taken into account. The first one is the number of visible pixels defined by

$$N_{vis} = \sum_p \mathbb{1}(O_{truth,l}(\mathbf{x}) = 0). \quad (7)$$

By convention, the authors consider that a disparity with a non numeric value (NaN) represents either an occlusion or the impossibility for the algorithm to estimate the disparity. The second one concerns the valid number of pixels defined by

$$N_{val} = \sum_p \mathbb{1}(O_{truth,l}(\mathbf{x}) = 0) \cdot \mathbb{1}(D_l(\mathbf{x}) \in \mathbb{R}). \quad (8)$$

Finally, the total number of pixels with a mismatch error below  $s$  is

$$S_a(D_{truth,l}, D_l) = \sum_p \mathbb{1}(O_{truth,l}(\mathbf{x}) = 0) \cdot \mathbb{1}(D_l(\mathbf{x}) \in \mathbb{R}) \cdot \mathbb{1}(|D_{truth,l}(\mathbf{x}) - D_l(\mathbf{x})| < s). \quad (9)$$

Additionally, the authors define dense  $S_a/N_{vis}$  and non dense  $S_a/N_{val}$  success rates. The second ratio is necessarily higher than the first one.

According to the Thales theorem, the reconstruction of point clouds from a disparity map  $D_{truth,l}$  under the hypothesis of a pinhole camera is

$$z_C(\mathbf{x}) = \frac{fB}{D_{truth,l}(\mathbf{x})}, \quad (10)$$

where  $z_C(\mathbf{x})$  is the distance from point  $\mathbf{x}_C$  to the plane  $(O_C, \vec{x}_C, \vec{y}_C)$ , containing the focal point  $O_C$ , and  $f$  is the focal length. Cabezas et al. [4] identified two elements that lack in the definition of metric (9). The first is that it does not take into account the a priori error made on the depth  $z_C$  during the 3D reconstruction. The precision error  $\mathbb{1}(|D_{truth,l}(\mathbf{x}) - D_l(\mathbf{x})| < s)$  implies that disparity maps having similar success rates after equation (9), and using the same threshold  $s$ , can produce 3D reconstructions of very different qualities. The second shortcoming is that the metric (9) is a function that depends on a threshold  $s$  whose value can influence the results.

To overcome these problems, Cabezas et al. derived from (10) the following metric, called SZE (Sigma-Z-Error), that sums the differences between the estimated and exact depths on the entire disparity map

$$SZE = \sum_{\mathbf{x}} \left| \frac{fB}{D_{truth,l}(\mathbf{x}) + \mu} - \frac{fB}{D_l(\mathbf{x}) + \mu} \right|, \quad (11)$$

where  $\mu$  is a constant that avoids numerical instabilities due to missing disparity estimates. According to the authors, this metric has the advantage of measuring the impact of the disparity error upon the error depth and does not resort to any threshold that could introduce a bias. However, they do not specify the value of  $\mu$ . Their experiments on the Middlebury database and the Tsukuba sequence

showed that the SZE metric is more efficient and accurate than metric (9) because it better reflects the errors in the observed reconstruction.

The drawback of measure (11) is that it is absolute and depends on the scene itself. In fact, it represents a kind of volume since the difference of depths is integrated over the entire surface of the image. This makes it difficult to compare the results obtained from the same scene with various baselines, or to average the results of different scenes to compute statistics. Furthermore, this metric requires the use of an unspecified smoothing term  $\mu$ .

### 2.3 Proposed Methodology and Metrics

Our goal is to compare the quality of disparity maps obtained by several algorithms according to their density and accuracy with respect to varying baselines. A downside of this approach is that we will compare success criteria obtained by the same algorithm but with different input data. Indeed, if we denote by  $u$  a left reference image and by  $B_1$  and  $B_2$  two shifts between two optical centers for the production of the respective right images,  $v(B_1)$  and  $v(B_2)$ , test pairs  $(u, v(B_1))$  and  $(u, v(B_2))$  have a different informational content. This criticism is legitimate but must be minimized: we do not only seek to assess algorithmic qualities but also two types of stereoscopic acquisition systems, namely pairs with a small or a large baseline. In what follows, we present these metrics and their suitability to this specific problem, together with the available ground truths.

In our opinion, the general behavior of an algorithm is better described using a range,  $S = [s_{\min}, s_{\max}]$ , of values rather than a fixed threshold  $s$  in Equation (9). As a consequence, the evaluation should be represented in the less condensed form of ROC curves, where success rate is expressed in terms of accuracy thresholds. The metrics we propose in what follows take into account a criterion precision incorporated into a density measurement. These considerations let us propose metrics which deal simultaneously with the accuracy and the density of the disparity maps.

**Precision criterion.** Equation (10) introduces the measure of the relative error on the depth defined by

$$\frac{|\tilde{z}_C(\mathbf{x}) - z_C(\mathbf{x})|}{|\tilde{z}_C(\mathbf{x})|} = \left| \frac{D_{truth,l}(\mathbf{x}) - D_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right|, \quad (12)$$

with  $\tilde{z}_C(\mathbf{x}) = fB/D_l(\mathbf{x}) = fB/(D_{truth,l}(\mathbf{x}) + \eta)$ , the term  $\eta$  being the error. This leads us to define the indicator function of accuracy as

$$\mathbb{1} \left( \left| \frac{D_{truth,l}(\mathbf{x}) - D_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right| < s \right), \quad \forall s \quad (13)$$

with the convention that the ratio is zero if the terms  $D_{truth,l}(\mathbf{x})$  and  $D_l(\mathbf{x})$  are zero simultaneously. These relative precision criteria allow us to be independent of the considered baseline and to express it as a percentage.

**Density criterion.** The definition of the metrics is based on the assumption that the area under the pixel can only have two states: being visible on both images or hidden in one. It follows that the exact and estimated occlusion maps present four possible Boolean relationships,  $\mathbf{r}$ , namely

$$\mathbf{r}_{kl}(\mathbf{x}) = (O_{truth,l}(\mathbf{x}) = k) \wedge (O_l(\mathbf{x}) = l), \quad \forall (k, l) \in \{0, 1\} \times \{0, 1\}, \quad (14)$$

that verifies

$$\sum_{\mathbf{x}} [\mathbb{1}(\mathbf{r}_{00}(\mathbf{x})) + \mathbb{1}(\mathbf{r}_{10}(\mathbf{x})) + \mathbb{1}(\mathbf{r}_{01}(\mathbf{x})) + \mathbb{1}(\mathbf{r}_{11}(\mathbf{x}))] = N_I. \quad (15)$$

State  $\mathbf{r}_{00}$  is associated to disparities that, if they exist, are estimated in non-occluded areas by a matching process specific to the algorithm. On the other hand, state  $\mathbf{r}_{10}$  corresponds to disparities that, if they exist, are estimated by an inference process. More specifically, it is possible that, in order to calculate the disparities of pixels visible in both images, the algorithm involves an interpolation process, for example on a constant color area, and not only a matching process. In contrast, to compute disparities of occluded pixels, the algorithm can only proceed from inference or interpolation. This leads us to define the following three metrics.

### 2.3.1 Density Precision Metrics

**Definition 1.** *The average density precision (ADP) metric, that compares the estimated disparity  $\mathring{D}_l$  with the true disparity  $D_{truth,l}$ , is the success rate*

$$T_{ADP}(\mathring{D}_l, D_{truth,l}, s) = \frac{1}{N_I} \sum_{\mathbf{x}} \mathbb{1} \left( \left| \frac{D_{truth,l}(\mathbf{x}) - \mathring{D}_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right| < s \right), \quad \forall s, \quad (16)$$

with

$$\mathbb{1} \left( \left| \frac{D_{truth,l}(\mathbf{x}) - \mathring{D}_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right| < s \right) = 0, \quad (17)$$

if  $\mathring{D}_l(\mathbf{x}) \notin \bar{\mathbb{R}}$  or if  $\mathring{D}_l(\mathbf{x})$  and  $D_{truth,l}(\mathbf{x})$  are null simultaneously.

This definition represents the proportion of points that are close to the ground truth disparity map. The ADP metric is computed over the non null regions of image  $D_{ADP} = D_{truth,l}$  (see Figure 2).

**Definition 2.** *The matching density precision (MDP) metric, relating to precision threshold  $s$ , comparing the estimated disparity  $\mathring{D}_l$  with the exact dense disparity  $D_{truth,l}$  and knowing the exact occlusions  $O_{truth,l}$  is the success rate*

$$T_{MDP}(\mathring{D}_l, D_{truth,l}, s) = \frac{\sum_{\mathbf{x}} \mathbb{1}(O_{truth,l} = 0) \cdot \mathbb{1} \left( \left| \frac{D_{truth,l}(\mathbf{x}) - \mathring{D}_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right| < s \right)}{\sum_{\mathbf{x}} \mathbb{1}(O_{truth,l} = 0)}, \quad \forall s, \quad (18)$$

with the same convention as Definition 1.

In this case, the density is calculated as the proportion of points that are visible in both images and their solution is close to the ground truth. We denote the visible ground truth disparity as  $D_{MDP} = \mathbb{1}(O_{truth,l} = 0) \cdot D_{truth,l}$ , where the MDP metric is computed over its non null regions (see Figure 2).

**Definition 3.** *The inference density precision (IDP) metric, relating to the precision threshold  $s$ , comparing the estimate disparity  $\mathring{D}_l$  with the exact dense disparity  $D_{truth,l}$ , and knowing the exact occlusions  $O_{truth,l}$ , is the success rate*

$$T_{IDP}(\mathring{D}_l, D_{truth,l}, s) = \frac{\sum_{\mathbf{x}} \mathbb{1}(O_{truth,l} = 1) \cdot \mathbb{1} \left( \left| \frac{D_{truth,l}(\mathbf{x}) - \mathring{D}_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right| < s \right)}{\sum_{\mathbf{x}} \mathbb{1}(O_{truth,l} = 1)}, \quad \forall s, \quad (19)$$

with the same conventions as in Definition 1.

This definition stands for the proportion of occluded pixels for which the calculated disparity is close to the ground truth. We denote the non-visible ground truth disparity map as  $D_{IDP} = \mathbb{1}(O_{truth,l} = 1) \cdot D_{truth,l}$ , where the IDP metric is computed over its non null regions (see Figure 2).

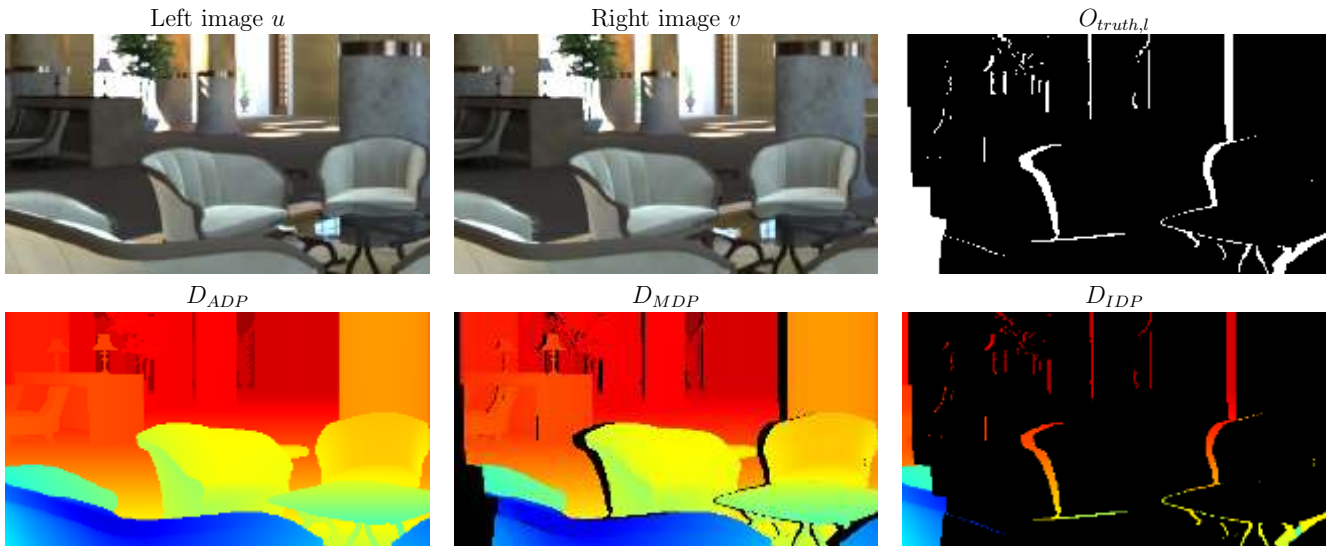


Figure 2: Given a stereo pair  $(u, v)$ , the associated occlusion map,  $O_{truth,l}$ , and the disparity map  $D_{truth,l}$ ; the ADP, MDP and IDP metrics are respectively computed over the non-black areas of the  $D_{ADP}$ ,  $D_{MDP}$  and  $D_{IDP}$  maps.

**Remarks.** To our knowledge, the assumption of the two-state visibility is valid in all current academic test databases, i.e. there is no concept of transparency. However, the notion of occlusion becomes quite unclear if there is an attempt to take the transparency phenomenon into account. The formalization of this problem was deemed too complex to be treated here.

Some stereoscopic algorithms distinguish two categories of non-assessed pixels: occluded pixels and non-assessable pixels. The latter property indicates that information is lacking to estimate their disparity. The proposed metrics do not differentiate these algorithms from those that make no discrimination. In both cases, the metrics consider that  $\hat{D}_l(\mathbf{x}) \notin \mathbb{R}$ .

Metrics (16) and (18) apply equally to dense or sparse algorithms. Unlike (9), the metric (18) includes all the pixels of the images and not only the visible or valid pixels. This is consistent with our goal to penalize the lack of information. The metric (19) only makes sense when assessing algorithms providing dense maps and should only be used (in order to be statistically relevant) in cases where the hidden pixels are numerous.

### 2.3.2 Consistency Criterion for Optical Flow Algorithms

In this work, we have included a post-processing to create sparse disparity maps  $\hat{D}_l(\mathbf{x})$  from dense ones. We did this due to the fact that some stereoscopic algorithms may provide non-dense maps, but this is not the case of the optical flow algorithms. Therefore, our intention is to avoid penalizing them by introducing a disparity rejection criterion that corresponds to the left-right consistency check [9] and that is based on the comparison between disparities estimated from the pairs  $(u, v)$  and  $(v, u)$ . We have, for all  $\mathbf{x}$ , the condition

$$\hat{D}_l(\mathbf{x}) \notin \bar{\mathbb{R}} \text{ if } \begin{cases} p + D_l(\mathbf{x}) < 0, \\ p + D_l(\mathbf{x}) > W_I, \\ \mathbb{1}(|D_l(\mathbf{x}) + \tilde{D}_r(\mathbf{x} + D_l(\mathbf{x}))| > \theta), \end{cases} \quad (20)$$

where  $W_I$  is the image width and  $\tilde{D}_r$  is the map of estimated and interpolated disparities at non integer positions  $\mathbf{x} + D_l(\mathbf{x})$ .

In the case of the Middlebury or Tsukuba [14] datasets for example, this consistency threshold is usually set to 0 or 1, this is because the considered disparity maps take their values in the range

$[a, b]$ , with  $1 < a$  and  $b \lesssim 300$ . The choice of a  $\theta$  threshold is more difficult in the subpixel context, especially as the CMLA dataset provides disparity maps such that  $0 \leq a \ll 1$  and  $b \leq 50$ . The orders of magnitude are thus very variable. In order to have a moderate rejection criterion, we propose to use the relative error defined by

$$\mathbb{1} \left( 2 \frac{|D_l(\mathbf{x}) + \tilde{D}_r(\mathbf{x} + D_l(\mathbf{x}))|}{|D_l(\mathbf{x}) - \tilde{D}_r(\mathbf{x} + D_l(\mathbf{x}))|} > \theta \right), \quad (21)$$

where the ratio is set to zero if  $D_l(\mathbf{x})$  and  $\tilde{D}_r(\mathbf{x} + D_l(\mathbf{x}))$  are zero simultaneously.

### 3 Description of Optical Flow Algorithms

In this section, we detail the 1D optical flow algorithms, which calculate non-rigid motions between two consecutive images.

These methods are based on two general assumptions: the luminance is conserved between the two images and their digital versions are correctly sampled in the sense of Shannon. It follows that, in the case of horizontal displacements  $\mathbf{d}$  between two images  $u$  and  $v$ ,

$$u(\mathbf{x}) = v(\mathbf{x} + \mathbf{d}), \quad \forall \mathbf{x} \in \Omega. \quad (22)$$

Applying first order Taylor expansions to the second member, and neglecting its residue because of the correctly sampled assumption, the gradient equation is

$$d \simeq \frac{u(\mathbf{x}) - v(\mathbf{x})}{v_x(\mathbf{x})}. \quad (23)$$

Raudies [19] has compiled a fairly comprehensive overview of optical flow methods — accompanied by their code — that rely on this constraint and possibly adding others.

#### 3.1 Lucas-Kanade Optical Flow

The Lucas and Kanade’s algorithm [13], in addition to the optical flow constraint (23), adds the supplementary assumption that movement  $\mathbf{d}(\mathbf{x})$  is constant in a neighborhood  $\mathcal{V}$  of  $\mathbf{x}$ . The authors define an energy functional based on the quadratic error between the images, as

$$E(d) = \sum_{\mathbf{w} \in \mathcal{V}(\mathbf{x})} (v(\mathbf{w} + \mathbf{d}) - u(\mathbf{w}))^2. \quad (24)$$

By differentiating this functional, they get the following relation:

$$d \simeq \frac{\sum_{\mathbf{w} \in \mathcal{V}(\mathbf{x})} \partial_x v(\mathbf{w})(u(\mathbf{w}) - v(\mathbf{w}))}{\sum_{\mathbf{w} \in \mathcal{V}(\mathbf{x})} (\partial_x v(\mathbf{w}))^2}. \quad (25)$$

Over the decades, many contributions have appeared to improve the estimation of  $d$ . Baker et al. [1] compile many of these variants; see also the implementation in [23].

The algorithm proposed in the evaluation is primarily used as a reference algorithm due to its simplicity. This is a 1D multiscale version similar to the one by Bouguet [2]. A Gaussian filter,  $G_{\sigma_1}$ , is applied at each scale to reduce image noise, as advocated by Rais et al. [18]. The algorithm, however, differs from these methods in the consideration of the chromatic channels. In both previous



cases, the input color images are converted to grayscale before being processed. The argument is that the averaging of the RGB channels, resulting in the grayscale image, reduces noise and increases the accuracy of the displacements estimate. Nevertheless, in the presence of very low noise, this averaging mitigates the signal and uniformizes it. Our algorithm retains the three channels to calculate displacements. Regarding the iterative process to estimate  $d$ , we opted for the method of Hager et al. [10], called the inverse additive algorithm [1, Section 3.3], which accelerates calculations. The algorithm is detailed in Algorithms 2, 3 and 4. We explain the main numerical schemes in the next section.

### 3.2 Robust Optical Flow in 1D

The Brox et al. method [3] formulates the optical flow estimation as a continuous optimization problem with a global energy functional that includes constraints for computing piecewise-continuous optical flow fields. In the case of disparity maps, this energy is given by

$$\begin{aligned}
 E(d) = & \int_{\Omega} \Psi \left( (v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x}))^2 \right) dx \\
 & + \gamma \int_{\Omega} \Psi \left( |\nabla v(\mathbf{x} + \mathbf{d}) - \nabla u(\mathbf{x})|^2 \right) dx \\
 & + \alpha \int_{\Omega} \Psi \left( |\nabla d|^2 \right) dx, \tag{26}
 \end{aligned}$$

where  $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ ,  $\epsilon := 0.001$  a small constant and  $\mathbf{d} = (d, 0)^T$ . The first term can be seen as a continuous formulation of the Lucas-Kanade matching function. The second term allows to find a solution where the gradients of both images are similar. This is interesting if there exist constant brightness changes between the two images. The last term is a regularization constraint that enforces the solution to be smooth. The use of robust functions,  $\Psi$ , allows to deal with problems like image noise, occlusions and brightness changes, and, in the case of the regularization term, to create piecewise-continuous motions with sharp edges. Parameters  $\gamma$  and  $\alpha$  weight the gradient and the smoothness terms, respectively.

The minimum of the previous energy functional can be found by solving the associated Euler-Lagrange equations, given by

$$\begin{aligned}
 0 = & \Psi'_D \cdot (v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x})) \cdot v_x(\mathbf{x} + \mathbf{d}) \\
 & + \gamma \Psi'_G \cdot [ (v_x(\mathbf{x} + \mathbf{d}) - u_x(\mathbf{x})) \cdot v_{xx}(\mathbf{x} + \mathbf{d}) \\
 & \quad + (v_y(\mathbf{x} + \mathbf{d}) - u_y(\mathbf{x})) \cdot v_{xy}(\mathbf{x} + \mathbf{d}) ] \\
 & - \alpha \operatorname{div} (\Psi'_S \cdot \nabla d), \tag{27}
 \end{aligned}$$

with  $\Psi'(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon^2}}$  and

$$\Psi'_D := \Psi' \left( (v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x}))^2 \right), \tag{28}$$

$$\Psi'_G := \Psi' \left( |\nabla v(\mathbf{x} + \mathbf{d}) - \nabla u(\mathbf{x})|^2 \right), \tag{29}$$

$$\Psi'_S := \Psi' \left( |\nabla d|^2 \right). \tag{30}$$

First order Taylor expansions are used to linearize  $v(x + d, y)$  and its derivatives as follows

$$v(\mathbf{x} + \mathbf{d}^{k+1}) \approx v(\mathbf{x} + \mathbf{d}^k) + v_x(\mathbf{x} + \mathbf{d}^k) \Delta d^k, \tag{31}$$

$$v_x(\mathbf{x} + \mathbf{d}^{k+1}) \approx v_x(\mathbf{x} + \mathbf{d}^k) + v_{xx}(\mathbf{x} + \mathbf{d}^k) \Delta d^k, \tag{32}$$

$$v_y(\mathbf{x} + \mathbf{d}^{k+1}) \approx v_y(\mathbf{x} + \mathbf{d}^k) + v_{xy}(\mathbf{x} + \mathbf{d}^k) \Delta d^k, \tag{33}$$

with  $\Delta d^k = d^{k+1} - d^k$ . We assume that  $d^k$  is a close approximation to our unknown  $d^{k+1}$ . The disparity map is incrementally updated as  $d^{k+1} = d^k + \Delta d^k$ .

For more numerical details see Appendix A.2.

### 3.3 Discontinuity-Preserving Robust 1D Optical Flow

One of the problems of the Brox et al. method is that it often produces motion contours that do not coincide with the object contours. A standard approach to overcome this problem is to include a scalar decreasing function [16] in the regularization term in order to mitigate the smoothing when the gradient of the image is high.

The energy functional is

$$E(d) = \int_{\Omega} \Psi((v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x}))^2) + \gamma \int_{\Omega} \Psi(|\nabla v(\mathbf{x} + \mathbf{d}) - \nabla u(\mathbf{x})|^2) + \alpha \int_{\Omega} \Psi(\Phi(\nabla u) \cdot |\nabla d|^2) dx, \quad (34)$$

where  $\Psi(s^2)$  is defined as in the previous section. The scalar decreasing function,  $\Phi(\nabla u_r)$ , is the only difference with respect to the previous model. It can be chosen as

$$\Phi(\nabla u) = e^{-\lambda(\mathbf{x})|\nabla u|}. \quad (35)$$

For more numerical details see Appendix A.3.

## 4 Evaluation

### 4.1 Experimental Protocol

In our experiments, we used the optical flow methods described in Section 3 and several methods specific to stereo. Among these, we selected those with representative and reproducible properties and whose theoretical descriptions and digital implementations are very well detailed. By convention, we will use the following acronyms:

- LKM: Lucas-Kanade 1D multiscale method;
- ROF1D: Robust Optical Flow 1D method;
- RDPOF1D: Robust Discontinuity Preserving Optical Flow 1D method;
- KZ: Kolmogorov-Zabih method [11];
- GF: Guided filter method [28];
- BWP: Bilaterally Weighted Patches method [29, 8];
- MGM: More Global Matching method [7].

The sequences used in this section belong to the CMLA dataset [5] and were produced by simulating a fronto-parallel camera motion. The images have near-zero noise and their disparity accuracy is in the order of  $10^{-6}$ . This dataset particularly includes four scenes with and without light reflections. We will call these two configurations “case A” and “case B”, respectively. From this dataset, we have extracted three groups of pairs, each consisting of 11 synthetic scenes:



Figure 3: The above seven scenes of the Middlebury 2014 dataset were used for the benchmark. Their disparity maps were obtained by laser scanning. They have a maximal precision of about 0.2 pixel.

- the group of pairs in “large” baseline, composed of viewpoints +000 and +050, where the maximum disparity is 50 pixels and the baseline is about 1/50;
- the group of pairs in “moderate” baseline, composed of viewpoints +000 and +010, where the maximum disparity is 10 pixels and the baseline is about 1/500;
- the group of pairs in “short” baseline, composed of viewpoints +000 and +001, where the maximum disparity is 1 pixel and the baseline is about 1/2500.

We also compared the algorithms for seven stereo pairs (shown in Figure 3) from the Middlebury 2014 dataset [26], which we previously rectified and corrected (see Section 4.2).

## 4.2 Preprocessing of Middlebury 2014 Pairs

The stereo pairs of the Middlebury 2014 dataset [26] have the following characteristics:

- size of about  $2800 \times 2000$  pixels;
- disparities varying from 30 to 250 pixels;
- maximal disparity precisions of about 0.2 pixel;
- unknown but weak noise values.

We selected the seven stereo pairs presented in Figure 3. In order to compare the results from this database with those of the synthetic pairs, both the pairs and the ground truths needed to be preprocessed (see Algorithm 1).

We reduced the size of the images by a factor of 5, thereby further reducing the image noise, whose left disparity maps have then a precision error of 0.04 consistent with a 0.075 precision limit (1/16th pixel), and whose 50 pixel maximal disparities are comparable with those of the large baseline pairs of the CMLA dataset. The raw disparity maps provided by Middlebury 2014 needed to be preprocessed before use, because they contained invalid values that made it difficult downscaling the images and computing the occlusion maps.

The occlusion map, denoted by  $O_{truth,l}$ , is calculated according to the formula

$$O_{truth,l}(\mathbf{x}) = \mathbb{1} (|D_{truth,l}(\mathbf{x}) + D_{truth,r}(\mathbf{x} + D_{truth,l}(\mathbf{x}))| \geq \alpha), \quad (36)$$

where the threshold  $\alpha$  is set to 0.2, to be consistent with the precision disparity maps provided by Middlebury 2014. The value of  $D_{truth,r}$  at the non-integer position  $(\mathbf{x} + D_{truth,l}(\mathbf{x}))$  is obtained by a spline interpolation of order 5.

Taking into account invalid disparities in the success rate computation implies rewriting relation (16) as

$$T'_M(\mathring{D}_l, D_{truth,l}, s) = \frac{1}{N_V} \sum_{\mathbf{x}} V_l(\mathbf{x}) \mathbb{1} \left( \left| \frac{D_{truth,l}(\mathbf{x}) - \mathring{D}_l(\mathbf{x})}{D_{truth,l}(\mathbf{x})} \right| < s \right), \quad \forall s, \quad (37)$$

where  $N_V = \sum_{\mathbf{x}} V_l(\mathbf{x})$  is the number of valid pixels.

---

**Algorithm 1:** PREPROCESS — Preprocessing applied on the images of the Middlebury 2014 dataset to obtain tractable stereo pairs and ground truths.

---

**Input:**  $(u, v)$  stereo pair of dimensions  $H_I \times W_I$ ,  $(D_l, D_r)$  disparity maps related to  $(u, v)$   
**Output:**  $(u^c, v^c)$  processed stereo pair,  $(V_l, O_l, D_l)$  left invalid, occluded and disparity maps

```

1 // REPLACEMENT OF INVALID VALUES BY MEDIAN
2 foreach  $w \in \{D_l, D_r\}$  do
3   if  $w(\mathbf{x}) \notin \mathbb{R}$  then  $w^a(\mathbf{x}) \leftarrow \text{median}(w)$  else  $w^a(\mathbf{x}) \leftarrow w(\mathbf{x})$ 
4 end
5 // LEFT VALIDITY MAP CREATION
6  $V_l(\mathbf{x}) \leftarrow \mathbb{1}(D_l^a(\mathbf{x}) \in \mathbb{R})$ 
7 // LEFT OCCLUSION MAP CREATION
8  $O_l(\mathbf{x}) \leftarrow \mathbb{1}(|D_l^a(\mathbf{x}) + D_r^a(\mathbf{x} + D_l^a(\mathbf{x}))| \geq 0.2)$ 
9 // PRESMOOTHING BY A GAUSSIAN BLUR
10 foreach  $w \in \{u, v, D_l^a, V_l, O_l\}$  do  $w^b \leftarrow \text{BLUR}[w, \sigma = 4]$ 
11 // DOWNSAMPLING BY A FACTOR  $S = 5$ 
12 foreach  $w \in \{u^b, v^b, D_l^b, V_l^b, O_l^b\}$  do
13   for  $\mathbf{x} \in \llbracket 0, W_I/S \rrbracket \times \llbracket 0, H_I/S \rrbracket$  do  $w^c(\mathbf{x}) \leftarrow w^b(S\mathbf{x})$ 
14 end
15 // BINARIZATION
16 foreach  $w \in \{V_l^c, O_l^c\}$  do  $w^d \leftarrow \mathbb{1}(w^c < 0.5)$ 
17  $\{V_l, O_l, D_l\} \leftarrow \{V_l^d, O_l^d, D_l^c\}$ 

```

---

### 4.3 Algorithms Parameters

Stereo algorithms were tested with their default settings. While the GF, BWP and KZ methods have pixel accuracy, MGM was adapted to reach a 1/4th pixel accuracy. The optical flow algorithms were also applied with fixed parameters. The LKM algorithm was configured with a  $5 \times 5$  pixel neighborhood and a Gaussian blur kernel of standard deviation  $\sigma = 0.4$ . ROF1D was used with  $\alpha_{ROF} = 2$  and  $\gamma_{ROF} = 2$  (respectively corresponding to parameters  $\alpha$  and  $\gamma$  in [25, Eq. (3)]) from observations made by Sánchez et al., while other parameters take default settings. RDPOF1D was used with parameters  $\alpha_{RDP} = 25$  and  $\gamma_{RDP} = 1$  in automatic mode, the other parameters take the default values. The consistency threshold  $\theta$  of (21) applied to LKM, ROF1D and RDPOF1D was set to 0.2.

### 4.4 Numerical Results

**Robustness evaluation.** We tested the robustness of the algorithms against the influence of a growing additive Gaussian noise from  $\sigma = 0$  to  $\sigma = 10$ . We used the metric ADP with a fixed accuracy threshold  $s = 1$ . This choice must be explained. The relation (13) defining the relative precision can be reformulated as  $|D_{truth,l}(\mathbf{x}) - \hat{D}_l(\mathbf{x})| < s|D_{truth,l}(\mathbf{x})|$ , but the short baseline CMLA pairs have disparities of approximatively one pixel, that is to say  $|D_{truth,l}(\mathbf{x})| \approx 1$ .

On the one hand, since discrete and pixel precise algorithms have no precision below one pixel, the minimal errors  $|D_{truth,l}(\mathbf{x}) - \hat{D}_l(\mathbf{x})|$  are larger than 1. It is therefore not appropriate to set a precision threshold  $s$  below 1. On the other hand, this threshold corresponds to a relative error of 100%. In the case of the long baselines CMLA pairs, there is no reason to increase the threshold. Therefore, this choice is a compromise that allows to compare pixel and subpixel algorithms together and under different baselines configurations. The results are shown in Figures 4 and 5, for the CMLA dataset, and in Figure 9, for the Middlebury dataset.

Globally, according to the graphics in Figures 4 and 5, we can observe behaviors depending on the noise level. For  $\sigma$  in  $[0, 2]$ , the results are in descending order from low to large baselines. Indeed, for low noise levels the error caused by occlusions dominate, and this error is bigger for larger baselines. When  $\sigma$  exceeds 2, the larger baselines become more robust. Thus, moderate baselines represent the best compromise. The Middlebury 2014 and CMLA databases give, for large baselines, almost the same algorithm ranking (see Figures 5 and 9). In descending order, we have: KZ, RDPOF1D, ROF1D, MGM, LKM, BWP and GF. All baselines combined, RDPOF1D, ROF1D and MGM are better than the others. The RDPOF1D rates are never lower than 90% on the CMLA dataset and are at 85% on Middlebury, regardless of noise and baseline.

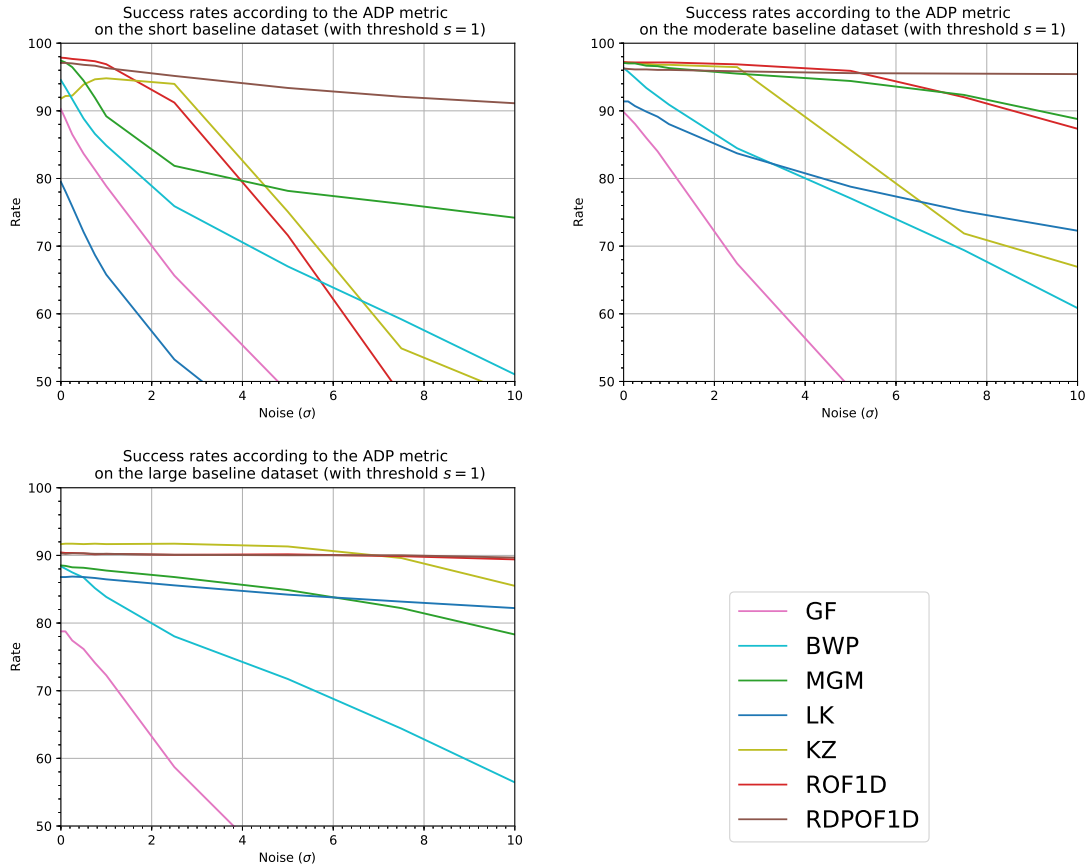


Figure 4: Top line, success rates of each algorithm according to the ADP metric, averaged on the 11 CMLA scenes, expressed as functions of an additive Gaussian blur  $\sigma$  and for a precision threshold of  $s = 1.00$ . This threshold is the smallest to allow simultaneous comparison of pixel and sub-pixel algorithms. The settings for ROF1D, RDPOF1D and LKM were set to identical values for the three baselines. Moderate baselines globally outweigh the large and low ones: occlusions in large baselines are penalizing, while noise rapidly degrades the short baselines. The three best algorithms are ROF1D, RDPOF1D and MGM whose success rates decrease more slowly than others depending on the noise.

We did the same tests for the sub-pixel algorithms LKM, ROF1D, RDPOF1D and MGM, with an accuracy threshold  $s = 0.25$ . The results are shown in Figure 6 for the CMLA dataset and Figure 9, for the Middlebury base. We observe the same phenomena as above: moderate baselines perform better than the low and large ones, for a noise smaller than  $\sigma = 2$ . For higher values, high and moderate baselines are similar, while in low baselines, only RDPOF1D is robust. ROF1D and RDPOF1D are better than MGM and the LKM, on both datasets.

**Accuracy evaluation.** We investigated the accuracy limits of the optical flow algorithms, LKM, ROF1D and RDPOF1D, with and without noise, by varying the threshold  $s$  from 1 to  $10^{-4}$ , combined

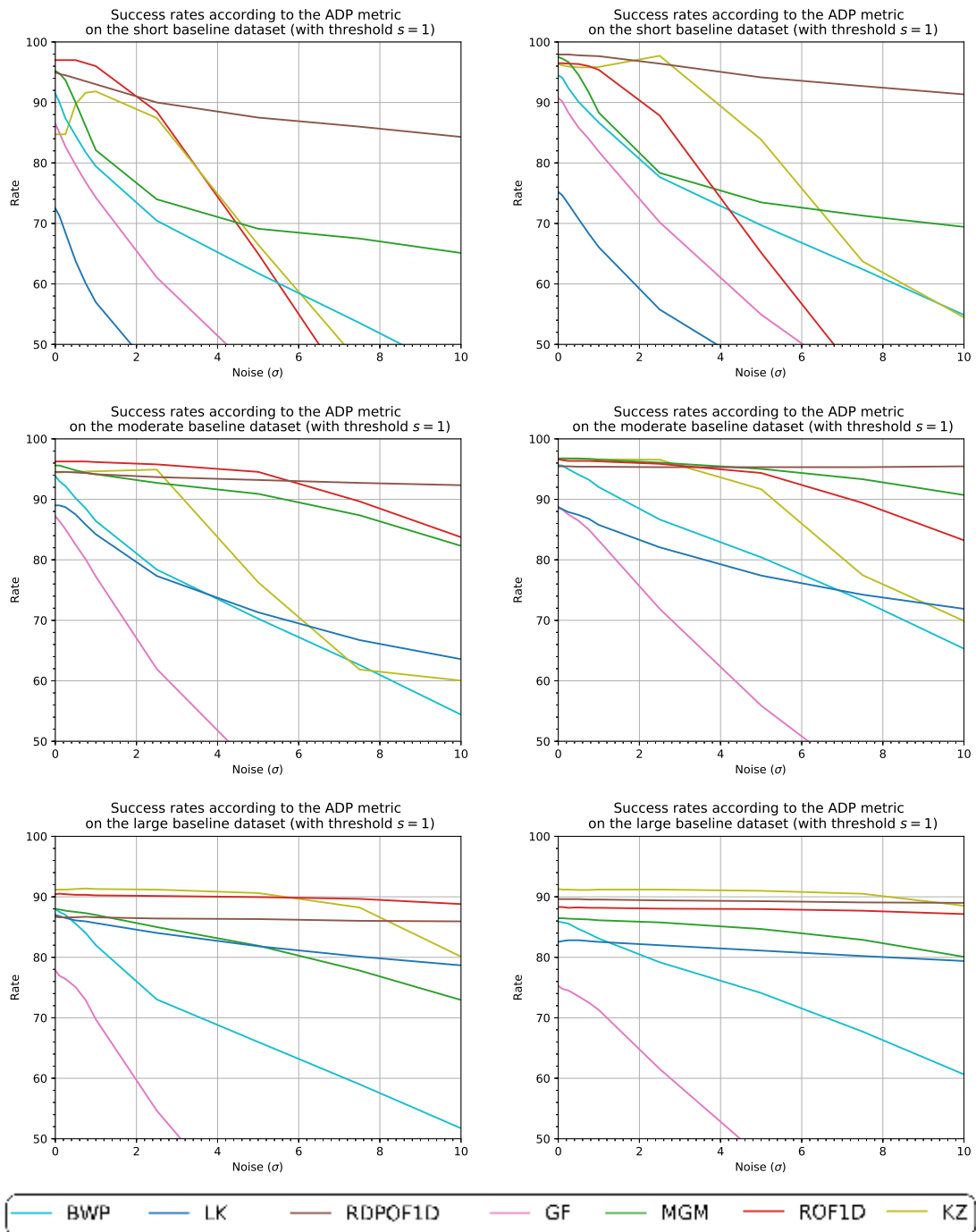


Figure 5: Left column (resp. right), success rates of each algorithm according to the ADP metric, averaged on the 4 scenes of the hotel without (resp. with) reflections, expressed as functions of an additive Gaussian blur  $\sigma$  and for a precision threshold of  $s = 1.00$ . We could have expected the results with the reflections to be worse than without, in large baselines, due to the areas of reflection that do not follow in appearance the real geometric displacements. However, this behavior does not apply to all algorithms.

with the ADP and IDP metrics. The results are shown in Figures 7, 8 and 10.

Concerning the results of the latter, they must however be taken with some caution, because the disparity errors of the “exact” disparity maps are about 0.04 pixel. Figure 7: in the absence of noise, we clearly observe a threshold effect  $s_e$ , whatever the algorithm we choose. For precision values  $s \geq s_e$ , moderate baselines dominate whereas, for precision values  $s \leq s_e$ , large baselines dominate. This corresponds to the occlusion effects of the large baselines penalizing information density in

COMPARISON OF OPTICAL FLOW METHODS UNDER STEREOMATCHING WITH SHORT BASELINES

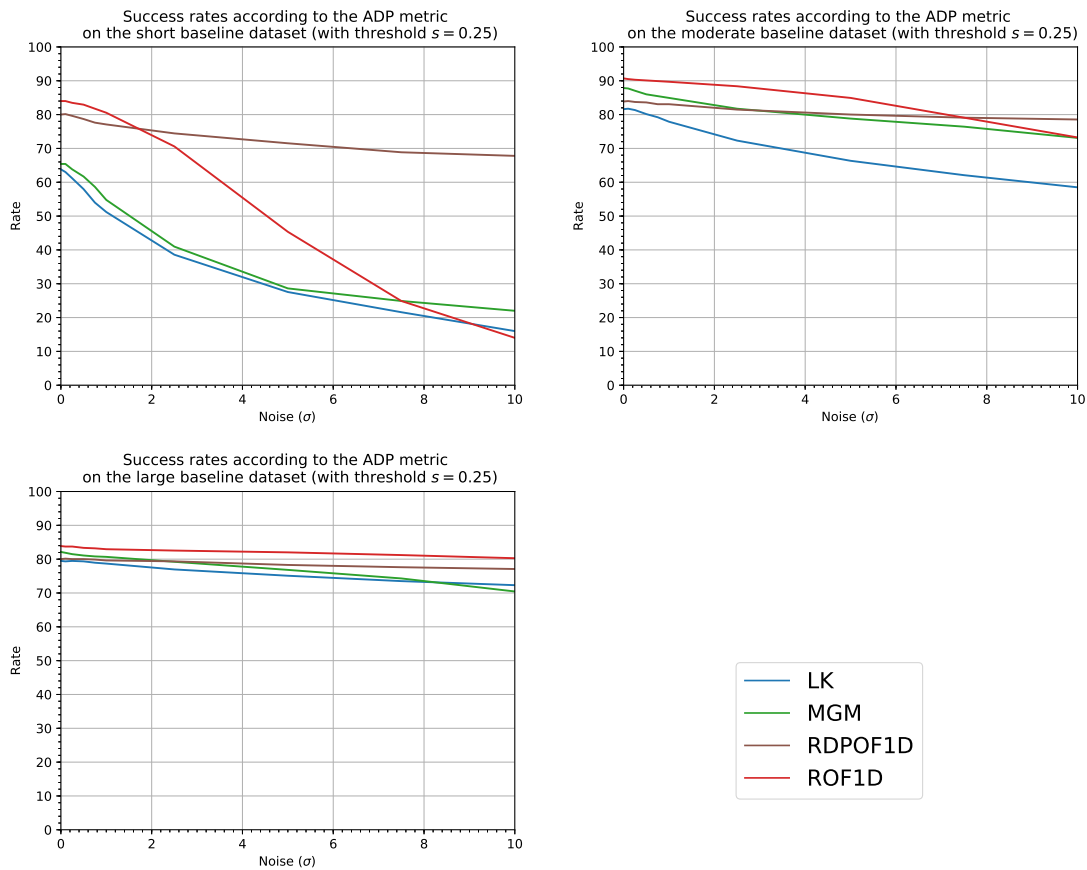


Figure 6: Success rates of each algorithm according to the ADP metric and the different baselines, averaged on the 11 CMLA scenes, expressed as functions of an additive Gaussian noise  $\sigma$  and for a precision threshold of  $s = 1.00$ . This threshold is the smallest to compare simultaneously all sub-pixel algorithms. The settings for ROF1D, RDPOF1D and LKM were set to identical values for the three baselines. Moderate and large baselines globally outweigh the short ones. The decrease in their success rate is quite slow. They only fall by 10 points between  $\sigma = 0$  and  $\sigma = 10$ , while in the short baseline case, success rates of LKM, MGM and ROF1D drop by more than 50 points. For a given baseline, ROF1D and RDPOF1D algorithms are more robust to noise than MGM.

the first case, and to the triangulation improvement by these same baselines, in the second. The thresholds of  $s_e$  of ROF1D, RDPOF1D and LKM are about 0.03, 0.01 and 0.15 respectively. The low baselines here may have little interest because they decrease very rapidly whatever the algorithms and the noise level.

In the presence of noise, large baselines globally dominate, whatever the algorithms. The effect of threshold is observable only for RDPOF1D and is worth more than 0.1. In addition, RDPOF1D is very robust. Regardless of the baseline considered, the drop in rates is much lower than the drop observed for the LKM and ROF1D. In both cases, regardless of the baselines, ROF1D is better than RDPOF1D and LKM without noise; it outweighs RDPOF1D by 10 points on average. It reaches 85% at  $s = 10^{-1}$  and 55% at  $s = 10^{-2}$ .

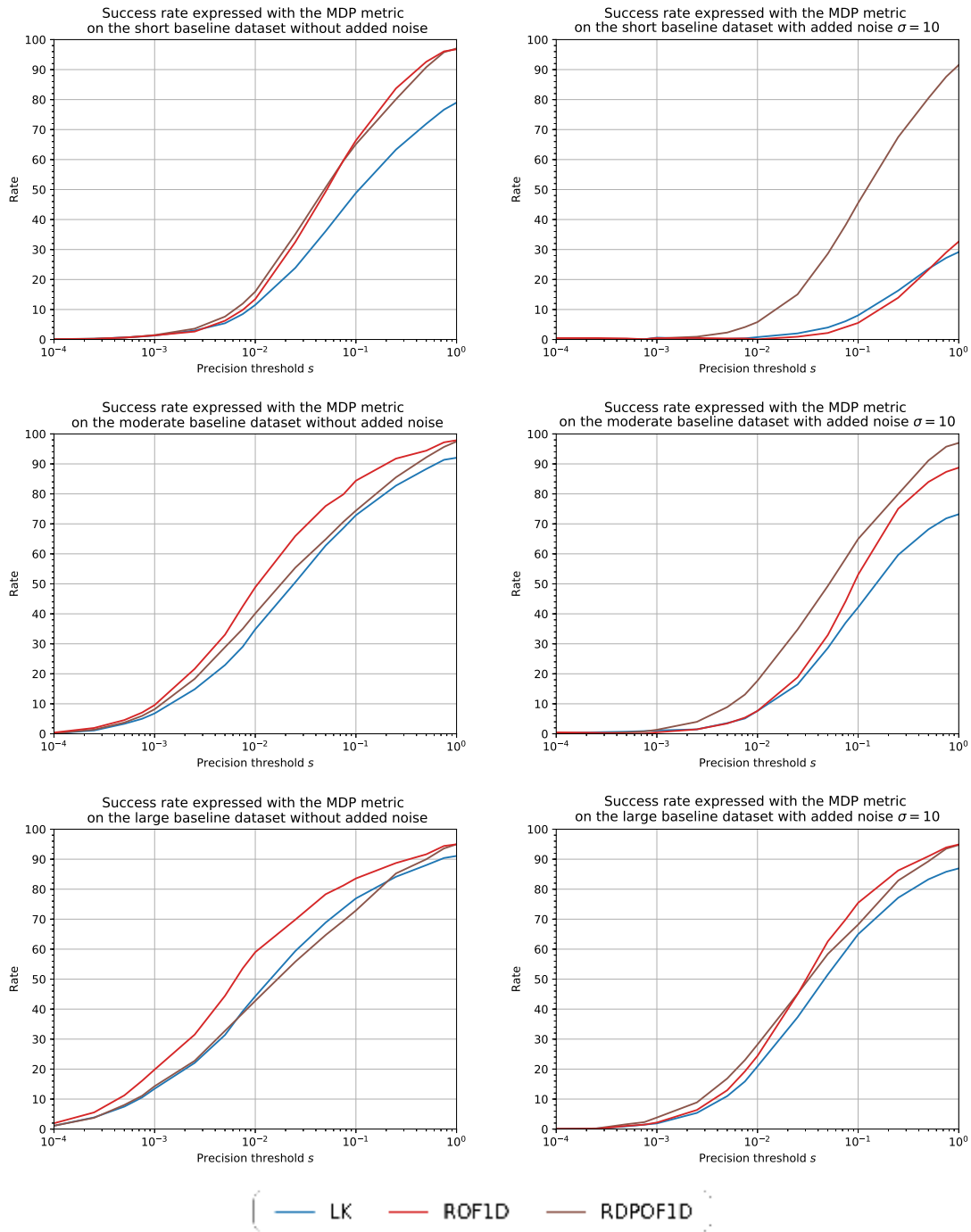


Figure 7: Success rates of optical flow algorithms according to the MDP metric, averaged on the 11 CMLA scenes, expressed as functions of the precision threshold  $s$ . On the left, with no added noise, and, on the right, with a Gaussian noise of  $\sigma = 10$ . In the absence of noise, optical flow methods get around 80% success at precision  $s = 0.1$  and around 50% at  $s = 0.01$ . The best rate is obtained by ROF1D with 85% and 60% for the previously mentioned thresholds. In the case of a noise with  $\sigma = 10$ , large baselines win whatever the method. ROF1D and RDPOF1D are ahead but they do not exceed 75% to 25% success at  $s = 1/10$  and  $s = 1/100$ , respectively.



COMPARISON OF OPTICAL FLOW METHODS UNDER STEREOMATCHING WITH SHORT BASELINES

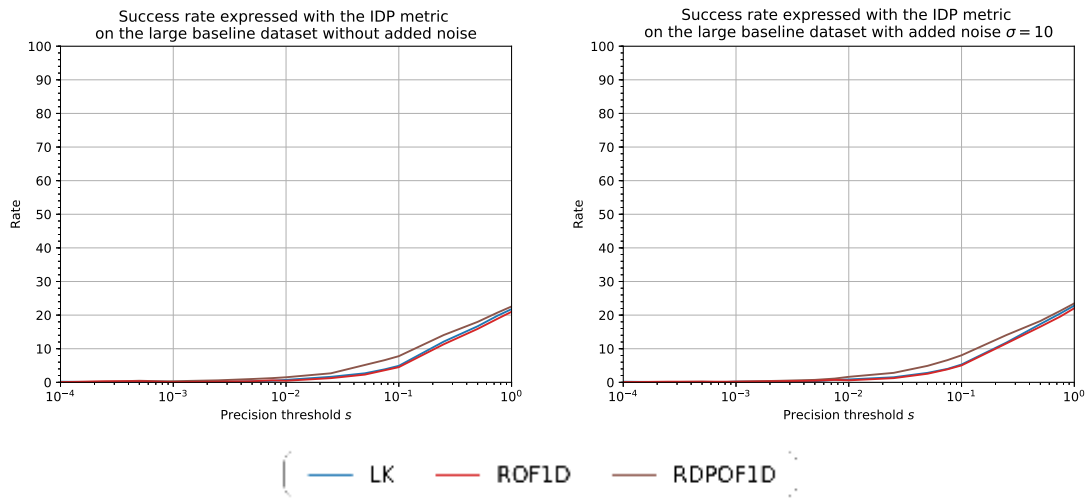


Figure 8: Success rate of optical flow algorithms according to the IDP metric averaged over the simulated pairs expressed as functions of the precision threshold  $s$ . On the left, with no added noise, and, on the right, with a Gaussian noise of  $\sigma = 10$ . The rates are very low because the algorithms do not use any inference method. However it is reasonable to think that if this were the case, they would not exceed the visible pixel rate.

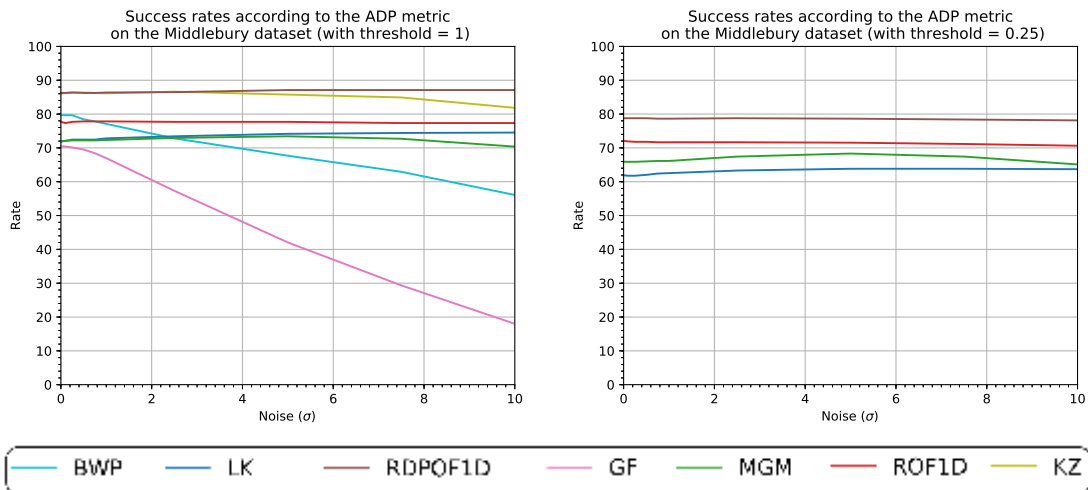


Figure 9: Left, success rates of each algorithm according to the ADP metric, averaged on the 7 Middlebury 2014 scenes, expressed as functions of an additive Gaussian noise and for a precision threshold of  $s = 1.00$ . This threshold is the smallest that allows to compare simultaneously pixel and sub-pixel algorithms. The baselines of these scenes are comparable to the large baselines of the CMLA dataset. Algorithms ROF1D, RDPOF1D, LKM and MGM remain very stable against increasing levels of noise, while KZ slightly decreases and the BWP and GF decrease very rapidly. Right, averaged success rates on the set of Middlebury 2014, of the four sub-pixel algorithms for a precision threshold of  $s = 0.25$ . RDPOF1D won with around 7 points more than ROF1D and almost 15 points higher than the MGM.

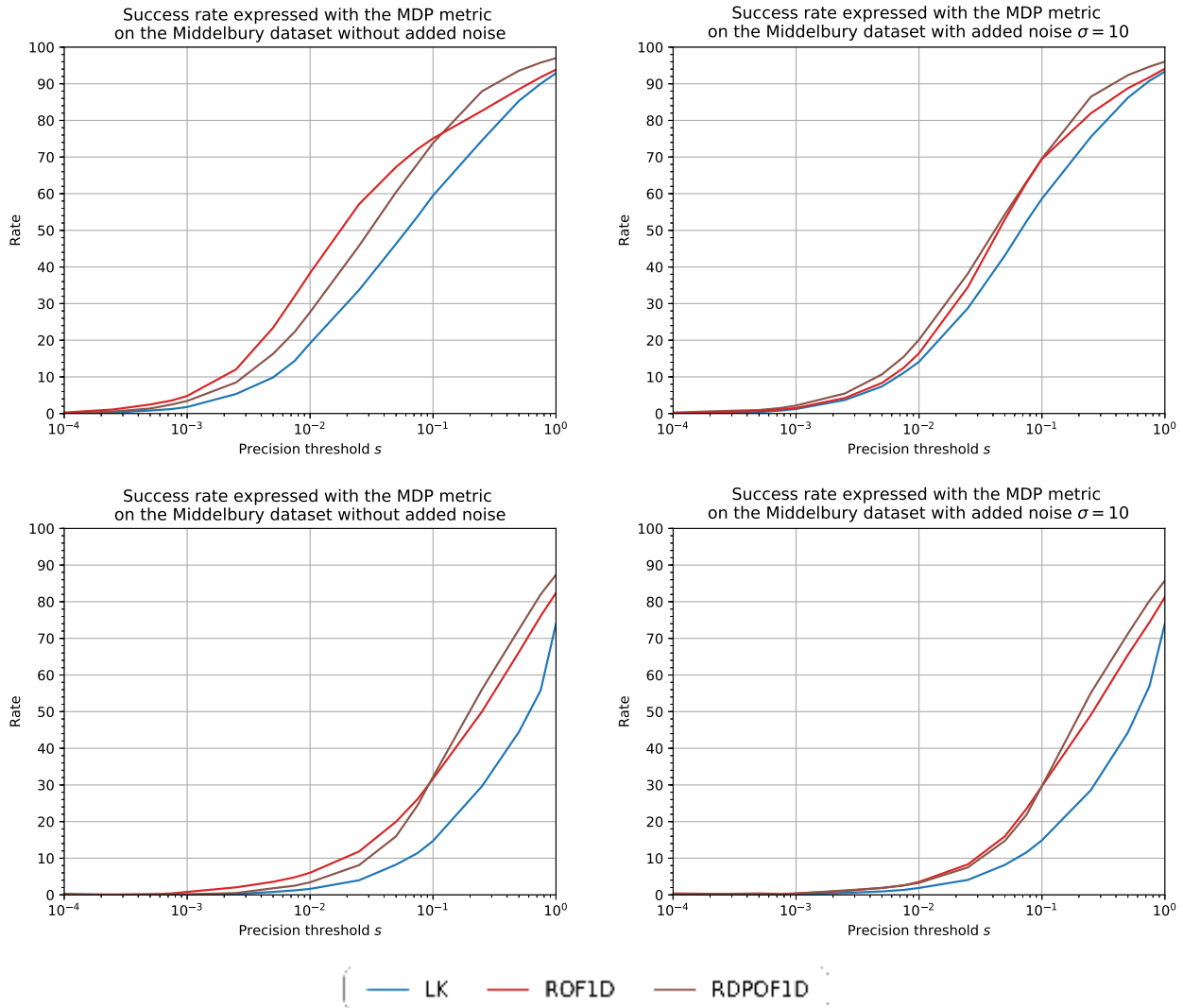


Figure 10: Top row, success rates of optical flow algorithms according to the MDP metric, averaged on the 7 Middlebury 2014 scenes, expressed as functions of the precision threshold  $s$ . On the left, with no added noise, and, on the right, with a Gaussian noise of  $\sigma = 10$ . In the absence of noise, ROF1D and RDPOF1D methods get around 70% of success at precision  $s = 0.1$  and around 30% at  $s = 0.01$ . The best rate is obtained by the ROF1D with 75% and 40% respectively, for the previously mentioned thresholds. In the case of a noise with  $\sigma = 10$ , large baselines win whatever the method. ROF1D and RDPOF1D are ahead but they do not exceed 70% to 20% success at  $s = 1/10$  and  $s = 1/100$ , respectively. Bottom row, the success rate according to the IDP metric averaged over the scenes. On the left, with no added noise, and, on the right, with Gaussian noise of  $\sigma = 10$ . The observed rates are higher than those of the synthetic scenes (see Figure 7). This could be explained by the fact that disparities in the occluded areas, for ground truths, are obtained after an interpolation process.

## 4.5 Comparison

In this section, we compare some disparity maps obtained from the LKM, ROF1D and RDPOF1D methods using some sequences from the moderate baseline of the CMLA dataset. Figures 11, 12 and 13 show the shiny versions of *bastet*, *saloon* and *shrub* sequences, while Figures 14 and 15 depict the *oranges* and *corridor* scenes. In each figure, the top row shows the original pair of images and the ground truth. The bottom row depicts, from left to right, the results for the LKM, ROF1D and RDPOF1D methods, respectively. The parameters have been set as in the previous experiments.

According to these results, LKM correctly detects the objects but it fails in several homogeneous areas. In contrast, the other two approaches achieve good results. On the one hand, the contour of the objects are better preserved by the RDPOF1D method, although it introduces some flow discontinuities in continuous regions that contain salient textures; on the other hand, ROF1D tends to obtain smoother solutions but with rounding shapes at the borders.

Figure 11 *bastet* is a good example of this behavior. RDPOF1D properly detects several details like the diadem, the pedestal and, the nose and the arms of the *bastet*. Besides, the lamp is well preserved while the other approaches fail. However, this method has more problems distinguishing the contours and the textures of the objects. This is more conspicuous in the floor of Figure 15. The smoothing behavior of the ROF1D method is better with textures. We also note that the leafs of the *shrub* (Figure 12) are better detected with ROF1D than with the other approaches. Nevertheless, the flowerpot is more similar to the ground truth using the RDPOF1D method and the regularization in the floor is more satisfactory. Unfortunately, the problems with specular lights appear again in this solution in front of the flowerpot.

In Figure 13 *saloon*, the details preserved with RDPOF1D are accurate, especially at the lamp and the arm-chairs. However, the motion of the center table is not completely correct and there are also some problems with the colored areas of the floor. This happens with the three methods. Interestingly, the solutions of the oranges sequence in Figure 14 are quite precise and the estimated motion in the background is very accurate for the RDPOF1D solution. Here, the LKM solution has conflicts dealing with the textured areas of the background.

From these experiments, we conclude that the RDPOF1D method is usually the best preserving the contours of the objects, but it also detect motion discontinuities inside some objects with prominent textures. The solutions of the ROF1D method tend to be smoother, but it creates rounding shapes at the contours and loses some details where the RDPOF1D method is successful. The lack of a regularization strategy in the LKM method causes this technique to fail in several situations, especially in homogeneous regions.

## 5 Conclusion

This study seems to support the hypothesis that, in the absence of noise, it is more advantageous to use stereoscopic pairs acquired in short baselines, and to apply optical flow algorithms on them. To this end, we proposed metrics to study the accuracy and density of stereo algorithms with pixel and sub-pixel accuracy. We then tested 1D optical flow and stereo algorithms on the Middlebury 2014 and CMLA datasets, up to their operating limits.

On the one hand, 1D optical flow methods, using regularization terms on the displacement fields, appear to be more accurate than stereo algorithms, regardless of the noise level and the baseline. In the absence of noise, it is possible to obtain a relative error of 1% with a pixel density of 50% for baselines around a few tens of pixels ( $\leq 50$ ) with optical flow algorithms.

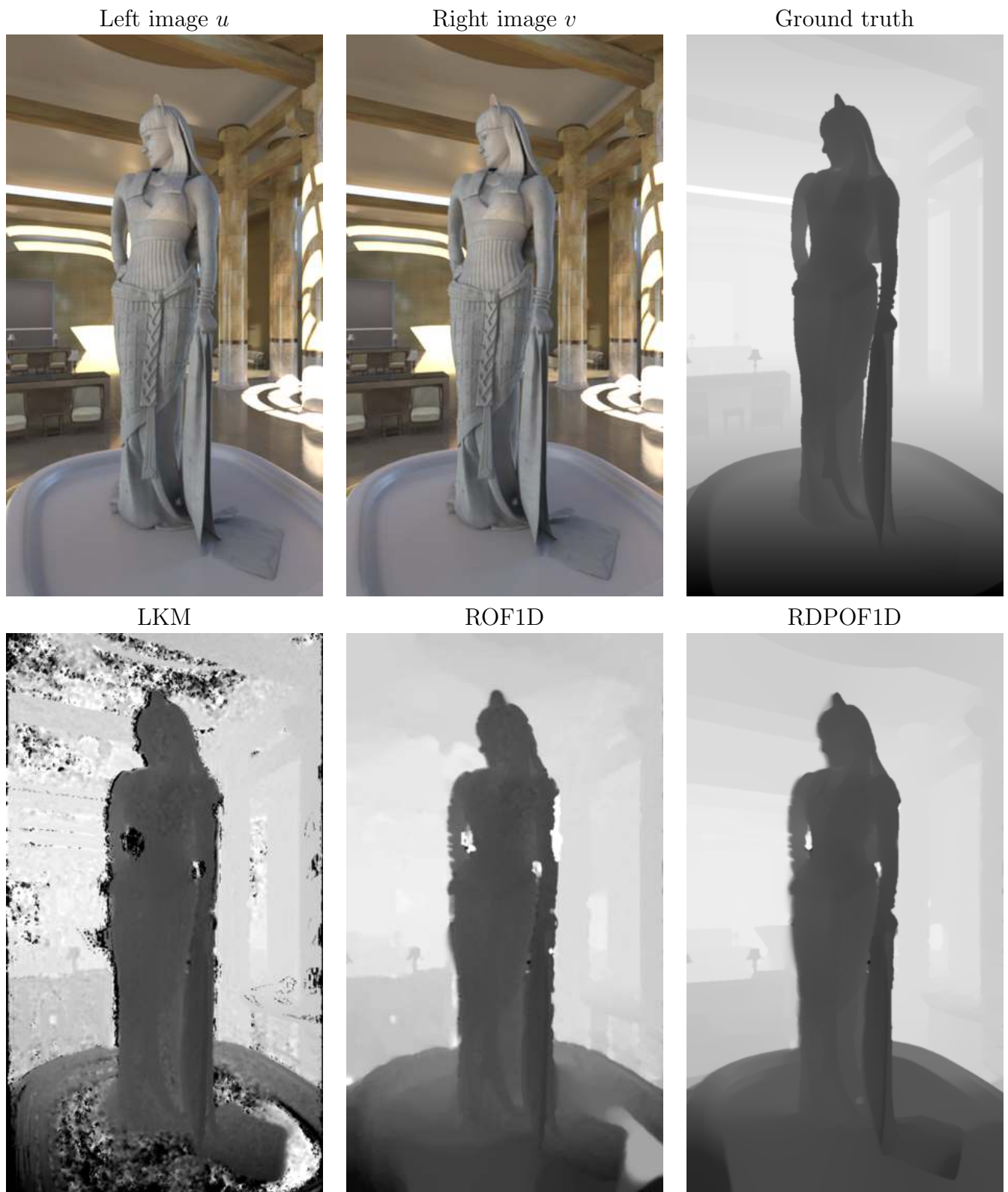


Figure 11: Results of the *bastet shiny* scene. Upper row: Left and right images using the moderate baseline and the ground truth. Bottom row: Disparity maps using LKM, ROF1D and RDPOF1D with their default parameters, respectively.

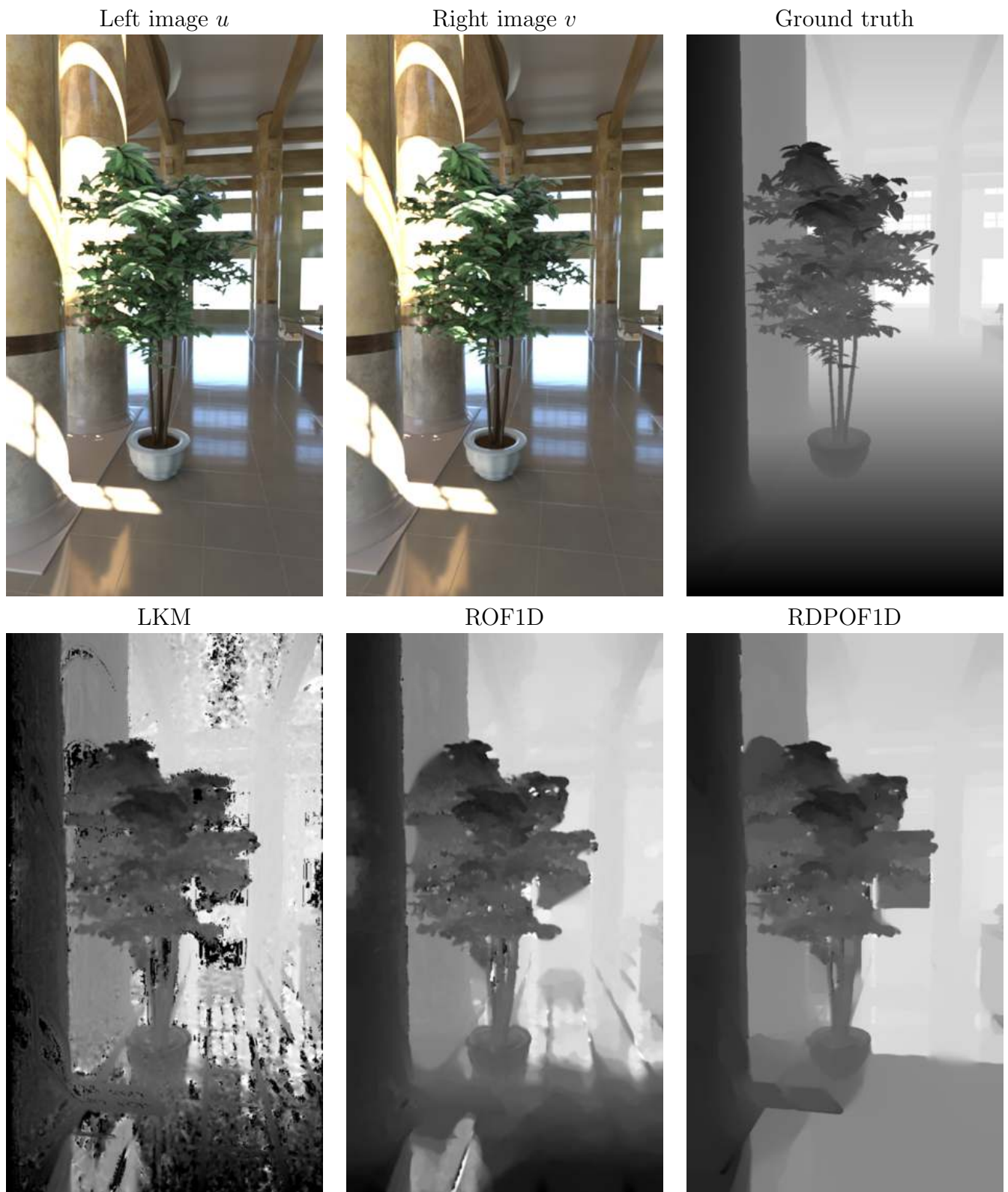


Figure 12: Results of the *shrub shiny* scene. Upper row: Left and right images using the moderate baseline and the ground truth. Bottom row: Disparity maps using LKM, ROF1D and RDPOF1D with default parameters, respectively.

Moreover, the results validate the use of a short baseline, up to a certain limit. In the absence of noise, and for a given accuracy, it may be more beneficial in terms of density, to work with a baseline of the order of  $1/250$  rather than  $1/50$ . However, the use of stereo pairs, whose disparities are less than, or equal to, the pixel (i.e. with baseline of about  $1/2500$ ), is ineffective when we want to get high accuracy. In the presence of noise, only large baselines ( $1/50$ ) offer the best compromise between density and precision.



Figure 13: Results of the *saloon shiny* scene. Upper row: Left and right images using the moderate baseline and the ground truth. Bottom row: Disparity maps using LKM, ROF1D and RDPOF1D with default parameters, respectively.

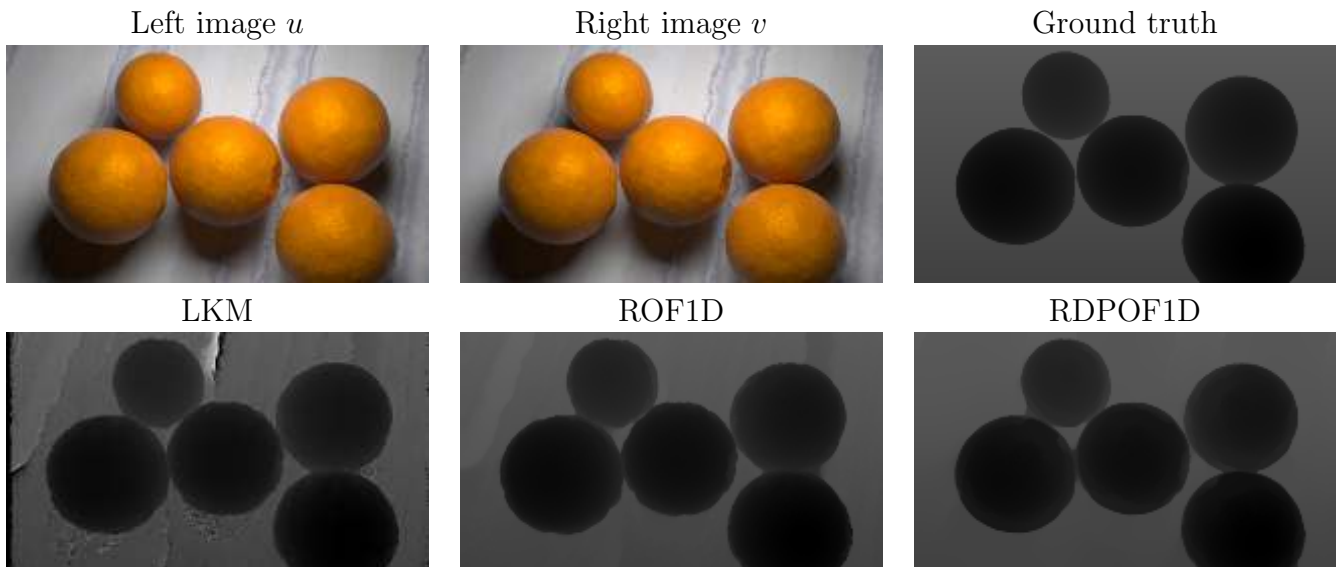


Figure 14: Results of the *oranges* scene. Upper row: Left and right images using the moderate baseline and the ground truth. Bottom row: Disparity maps using LKM, ROF1D and RDPOF1D with their default parameters, respectively.

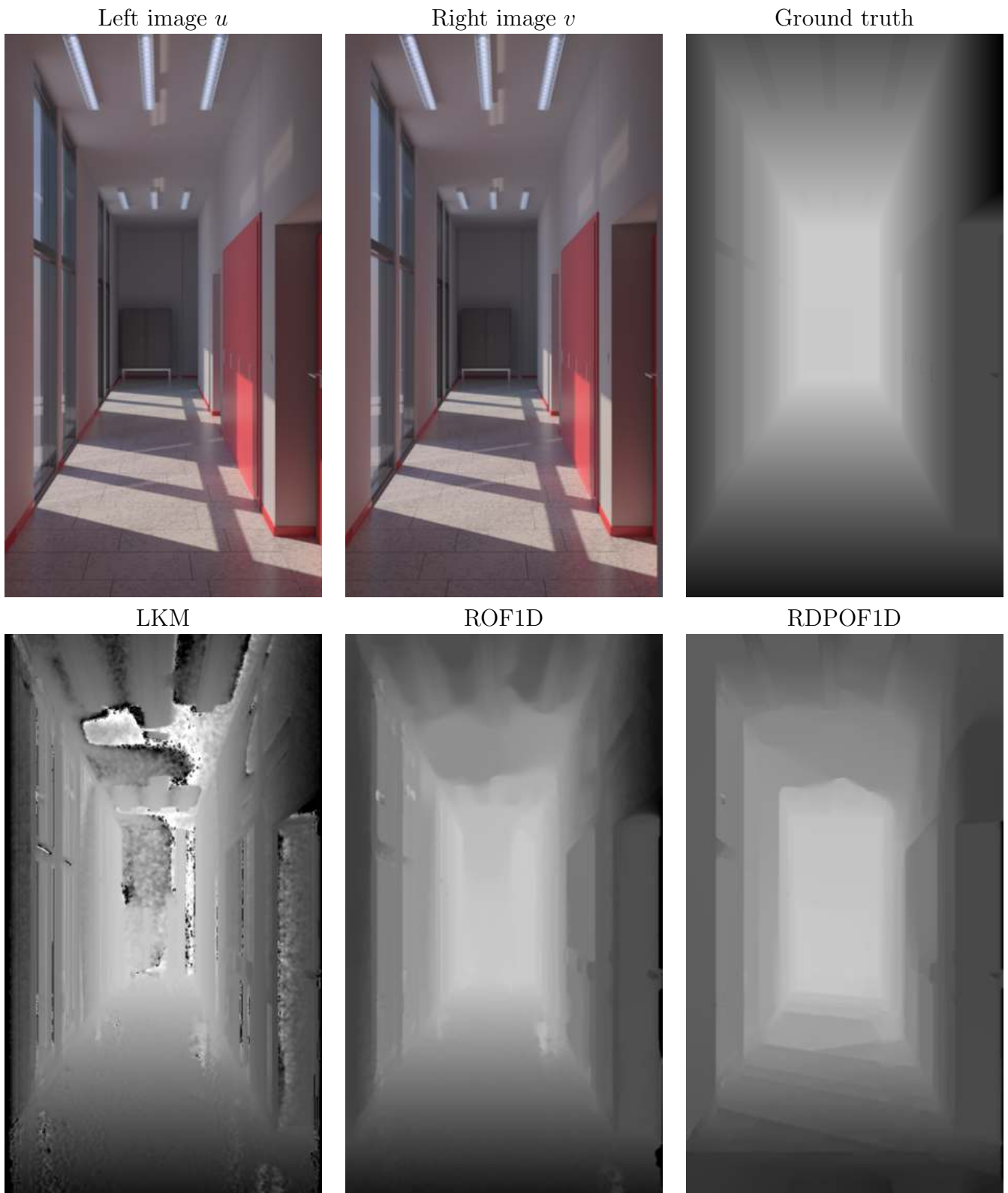


Figure 15: Results of the *corridor* scene. Upper row: Left and right images using the moderate baseline and the ground truth. Bottom row: Disparity maps using LKM, ROF1D and RDPOF1D with default parameters, respectively.

## A Numerical Schemes

### A.1 Numerical Scheme of the Lucas-Kanade Algorithm

Given  $S$ , the number of scales, we denote by  $u^s = (u_R^s, u_G^s, u_B^s)$  (resp.  $v^s$ ), the left color (resp. the right) image at scale  $s$ . These images have a size of  $W_s \times H_s$  pixels at scale  $s$ ; we denote by  $I^s = \llbracket 0, W_s \llbracket \times \llbracket 0, H_s \llbracket$  the discretized digital image.

**Neighborhood.** The neighborhood  $\mathcal{V}(\mathbf{x})$  consists of pixels  $\mathbf{w}$  such that  $\mathcal{V}(\mathbf{x}) = \{\mathbf{w} \in \llbracket 0, W_s \llbracket \times \llbracket 0, H_s \llbracket, \|\mathbf{x} - \mathbf{w}\|_\infty < \epsilon\}$  for a given  $\epsilon$ . The size of the neighborhood is fixed regardless of the considered scale.

**Multiscale structure.** The scaling of color images  $u$  and  $v$  from  $s$  to  $s + 1$  is done by blurring the images with a Gaussian kernel,  $G_{\sigma_2}$ , of standard deviation  $\sigma_2 = 1.2$ , to reduce aliasing, and by subsampling them using bicubic interpolation. This is formulated, considering only  $u$ , as

$$u^{s+1}(\mathbf{x}) = (K_1 * G_{\sigma_2} * u^s)(x/\rho_x, y/\rho_y) \quad \forall \mathbf{x} \in I^{s+1}, \quad (38)$$

where  $K_1$  is the bicubic interpolation kernel and  $\rho_x, \rho_y$  the scale factors set at  $\frac{\lfloor L_s/2 \rfloor - 1}{L_s - 1}$  and  $\frac{\lfloor H_s/2 \rfloor - 1}{H_s - 1}$ , respectively. Each image  $u^s$  is then blurred by  $G_{\sigma_1}$ , where  $\sigma_1 = 0.4$ , as

$$\check{u}^s(\mathbf{x}) = G_{\sigma_1} * u^s(\mathbf{x}), \quad \forall \mathbf{x} \in I^s. \quad (39)$$

This convolution is performed in the Fourier domain. The upscaling from the disparity map  $D_i^{s+1}$ , estimated at scale  $s + 1$ , to the disparity map  $D_{init}^s$  at scale  $s$ , is performed by bilinear interpolation as

$$D_{init}^s(\mathbf{x}) = \frac{1}{\rho_x} (K_2 * D_i^{s+1})(\rho_x x, \rho_y y), \quad \forall \mathbf{x} \in I^s, \quad (40)$$

where  $K_2$  is the bilinear interpolation kernel. (Bilinearity prevents the sign of the displacement direction to change whereas higher order interpolation may not.)

**Differentiation.** As the right images  $v^s$  are blurred at each scale  $s$ , the derivative  $\check{v}_x^s$  is calculated using  $\partial_x G_{\sigma_1}$  as

$$\check{v}_x^s(\mathbf{x}) = (\partial_x G_{\sigma_1} * v^s)(\mathbf{x}), \quad \forall \mathbf{x} \in I^s, \quad (41)$$

where

$$\partial_x G_{\sigma_1}(\mathbf{x}) = -\frac{x}{2\pi\sigma_1^4} \exp\left(-\frac{\|\mathbf{x}\|_2^2}{2\sigma_1^2}\right). \quad (42)$$

This convolution is performed in the Fourier domain.

**Iterative displacement estimation.** In the original article [13], the resolution of (25) is performed iteratively according to a Newton-Raphson scheme which has the disadvantage of having to calculate the interpolated values of (25) at each iteration and for all  $\mathbf{w}$  in  $\mathcal{V}(\mathbf{x})$ . Hager et al. [10] proposed an improvement by reformulating the energy (24) from the point of view of elementary displacements  $d_k$  [10, Eq. (7)] and showed that, in the particular case of translation along  $x$  [10, Eq. (26)], the scheme is

$$\begin{aligned} d_0 &= 0, \\ d_{k+1} &= d_k + \frac{A_k}{B}, \end{aligned} \quad (43)$$



where

$$A_k = \sum_{c \in \{R,G,B\}} \sum_{\mathbf{w} \in \mathcal{V}(p)} \check{u}_{x,c}^s(\mathbf{w}) [\check{u}_c^s(\mathbf{w}) - (K_1 * \check{v}_c^s)(\mathbf{w} + D_{init}(\mathbf{w}) + \mathbf{d}_k)], \quad (44)$$

$$B = \sum_{c \in \{R,G,B\}} \sum_{\mathbf{w} \in \mathcal{V}(p)} (\check{u}_{x,c}^s(\mathbf{w}))^2. \quad (45)$$

---

**Algorithm 2:** LKM — Multiscale Lucas-Kanade optical flow 1D algorithm.

---

**Input:**  $(u^1, v^1, \sigma_1, v_x, v_y, T, S)$  with :  $u^1$  reference image;  $v^1$  shifted image;  $\sigma_1$  standard deviation of Gaussian blur kernel  $G_{\sigma_1}$ ;  $T$  maximum number of iterations;  $S$  number of scales.

**Output:**  $D_l$  estimated disparity map of  $u^1$

```

1 // COMPUTATION OF  $(u^s, v^s)$  AT DIFFERENT SCALES  $s$ 
2 for  $s \leftarrow 2$  to  $S$  do
3    $u^s \leftarrow \text{REDUCE}[u^{s-1}]$ 
4    $v^s \leftarrow \text{REDUCE}[v^{s-1}]$ 
5 end
6 // INITIAL DISPARITY
7  $D_l^S \leftarrow \mathcal{M}_{(L_S \times H_S)}(0)$ 
8 for  $s \leftarrow S$  to 1 do
9   // IMAGE BLURRING
10   $\check{u}^s \leftarrow \text{BLUR}(u^s, G_{\sigma_1})$ 
11   $\check{v}^s \leftarrow \text{BLUR}(v^s, G_{\sigma_1})$ 
12  // COMPUTATION OF  $x$ -DERIVATIVE OF REFERENCE IMAGE
13   $\check{u}_x^s \leftarrow \text{BLUR}(\check{u}^s, \partial_x G_{\sigma_1})$ 
14  for  $\mathbf{x} \in \llbracket 0, W_s \rrbracket \times \llbracket 0, H_s \rrbracket$  do
15    // DISPLACEMENT ESTIMATION
16     $d_0 \leftarrow 0$ 
17     $d_{init} \leftarrow D_{init}^s$ 
18    for  $k \leftarrow 0$  to  $T$  do
19      
$$d \leftarrow \frac{\sum_{c \in \{R,G,B\}} \sum_{\mathbf{w} \in \mathcal{V}(p)} \check{u}_{x,c}^s(\mathbf{w}) [\check{u}_c^s(\mathbf{w}) - (K_1 * \check{v}_c^s)(\mathbf{w} + D_{init}(\mathbf{w}) + \mathbf{d}_k)]}{\sum_{c \in \{R,G,B\}} \sum_{\mathbf{w} \in \mathcal{V}(p)} (\check{u}_{x,c}^s(\mathbf{w}))^2}$$

20       $d_{k+1} = d_k + d$ 
21    end
22     $D_l^s(\mathbf{x}) \leftarrow d_T$ 
23  end
24  // DISPARITY MAP INITIALIZATION AT NEXT SCALE (40)
25   $D_{init}^{s-1}(x, y) \leftarrow \gamma_x(K_2 * D_l^s)(x/\gamma_x, y/\gamma_y)$ 
26 end
27 end
28 return  $D_l^1$ 

```

---

## A.2 Numerical Scheme of Robust Optical Flow Algorithm

An efficient way to solve (27) is through an iterative scheme like the Gauss-Seidel or SOR methods. The Brox method [3] is implemented by two fixed point iterations: a set of iterations is used to

---

**Algorithm 3:** REDUCE — Function of reduction used for multiscale construction.

---

**Input:**  $u$  image of dimension  $H_I \times W_I$   
**Output:**  $w$  reduced image by factor 4

```

1 // ALIASING ATTENUATION BY A GAUSSIAN BLUR
2  $v \leftarrow \text{BLUR}[u, \sigma = 1.0]$ 
3  $\rho_x \leftarrow \frac{\lfloor L_I/2 \rfloor - 1}{L_I - 1}$ 
4  $\rho_y \leftarrow \frac{\lfloor H_I/2 \rfloor - 1}{H_I - 1}$ 
5 // BICUBIC INTERPOLATION OF KERNEL  $K_1$ 
6 for  $(x, y) \in \llbracket 0, W_I \rrbracket \times \llbracket 0, H_I \rrbracket$  do
7    $w(x, y) \leftarrow (K_1 * v)(x/\rho_x, y/\rho_y)$ 
8 end
```

---



---

**Algorithm 4:** BLUR — Blur algorithm.

---

**Input:**  $u$  image of dimension  $H_I \times W_I$   
**Input:**  $G$  Gaussian blur image of dimension  $H_I \times W_I$   
**Output:**  $v$  blurred image

```

1 // SYMMETRIZATION OF  $u$  IN THE 4 DIRECTIONS
2 for  $(x, y) \in \llbracket 0, W_I \rrbracket \times \llbracket 0, H_I \rrbracket$  do
3    $U(W_I + x, H_I + y) \leftarrow u(x, y)$ 
4    $U(W_I + x, H_I - y) \leftarrow u(x, y)$ 
5    $U(W_I - x, H_I + y) \leftarrow u(x, y)$ 
6    $U(W_I - x, H_I - y) \leftarrow u(x, y)$ 
7 end
8 // CONVOLUTION USING FOURIER TRANSFORM  $\mathcal{F}$ 
9  $U \leftarrow \mathcal{R}e(\mathcal{F}^{-1}(\mathcal{F}(U)\mathcal{F}(G)))$ 
10 // SUITABLE PART OF THE IMAGE
11 for  $(x, y) \in \llbracket 0, W_I \rrbracket \times \llbracket 0, H_I \rrbracket$  do
12    $v(x, y) \leftarrow U(W_I + x, H_I + y)$ 
13 end
```

---

account for the Taylor expansions (index  $k$ ), and a set of iterations for the nonlinearities of the robust functions  $\Psi((v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x}))^2)$  (index  $l$ ); see [25] for more details. Using an index  $s$  to represent the iterations of the SOR method, the numerical scheme for updating  $d$  is given by

$$d^{k,l,s+1} := (1 - w) d^{k,l,s} + w \frac{\left( A + \alpha \operatorname{div}(\Delta d)_{i\pm 1, j\pm 1}^{k,l,s} \right)}{D}, \quad (46)$$

with  $w \in (0, 2)$  the SOR relaxation parameter ( $w := 1.9$  by default). In this equation,  $A$  and  $D$  are given by

$$\begin{aligned} A &:= - (\Psi'_D)^{k,l} (v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x})) v_x(\mathbf{x} + \mathbf{d}) + \alpha \operatorname{div}(d)_{i\pm 1, j\pm 1}^{k,l} \\ &\quad - \gamma (\Psi'_G)^{k,l} [ (v_x(\mathbf{x} + \mathbf{d}) - u_x(\mathbf{x})) v_{xx}(\mathbf{x} + \mathbf{d}) \\ &\quad \quad + (v_y(\mathbf{x} + \mathbf{d}) - u_y(\mathbf{x})) v_{xy}(\mathbf{x} + \mathbf{d}) ], \\ D &:= (\Psi'_D)^{k,l} v_x^2(\mathbf{x} + \mathbf{d}) + \gamma (\Psi'_G)^{k,l} (v_{xx}^2(\mathbf{x} + \mathbf{d}) + v_{xy}^2(\mathbf{x} + \mathbf{d})) \\ &\quad + \alpha C_{i,j}^{k,l}. \end{aligned} \quad (47)$$

Partial derivatives are approximated using central differences and expressions like  $v(\mathbf{x} + \mathbf{d}^{k,l})$  are computed using bicubic interpolation. The discretization of the divergence can be separated in three parts by

$$\begin{aligned} \operatorname{div}((\Psi'_S) \cdot \nabla(d + \Delta d)) &= \operatorname{div}((\Psi'_S) \cdot \nabla d) + \operatorname{div}((\Psi'_S) \cdot \nabla \Delta d) \\ &\approx \operatorname{div}(d)_{i,j}^{k,l} + \operatorname{div}(\Delta d)_{i\pm 1,j\pm 1}^{k,l+1} - C_{i,j}^{k,l} \cdot \Delta d_{i,j}^{k,l+1} \end{aligned} \quad (48)$$

where

$$\begin{aligned} \operatorname{div}(d)_{i,j}^{k,l} &:= \frac{(\Psi'_S)_{i+1,j}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} (d_{i+1,j}^{k,l} - d_{i,j}^{k,l}) \\ &\quad + \frac{(\Psi'_S)_{i-1,j}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} (d_{i-1,j}^{k,l} - d_{i,j}^{k,l}) + \\ &\quad \frac{(\Psi'_S)_{i,j+1}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} (d_{i,j+1}^{k,l} - d_{i,j}^{k,l}) \\ &\quad + \frac{(\Psi'_S)_{i,j-1}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} (d_{i,j-1}^{k,l} - d_{i,j}^{k,l}), \end{aligned} \quad (49)$$

$$\begin{aligned} \operatorname{div}(\Delta d)_{i\pm 1,j\pm 1}^{k,l+1} &:= \frac{(\Psi'_S)_{i+1,j}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} \Delta d_{i+1,j}^{k,l+1} + \frac{(\Psi'_S)_{i-1,j}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} \Delta d_{i-1,j}^{k,l+1} + \\ &\quad \frac{(\Psi'_S)_{i,j+1}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} \Delta d_{i,j+1}^{k,l+1} + \frac{(\Psi'_S)_{i,j-1}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} \Delta d_{i,j-1}^{k,l+1}, \end{aligned} \quad (50)$$

and

$$\begin{aligned} C_{i,j}^{k,l} &:= \frac{(\Psi'_S)_{i+1,j}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} + \frac{(\Psi'_S)_{i-1,j}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} + \\ &\quad \frac{(\Psi'_S)_{i,j+1}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2} + \frac{(\Psi'_S)_{i,j-1}^{k,l} + (\Psi'_S)_{i,j}^{k,l}}{2}. \end{aligned} \quad (51)$$

Equation (46) is iterated until convergence of the method to a steady state solution or until the number of iterations exceeds a maximum.

$$\frac{1}{N} \sum_{i,j,k} (du_{i,j,k}^{s+1} - du_{i,j,k}^s)^2 + (dv_{i,j,k}^{s+1} - dv_{i,j,k}^s)^2 < \varepsilon^2, \quad (52)$$

with  $N$  the number of pixels and  $\varepsilon$  the stopping criterion threshold. Upon convergence, the method passes to the next iteration,  $l + 1$ , and recomputes the variables in (47). A coarse-to-fine approach, similar to the one explained in Section 3.1 or in [15, 24], can be used to estimate large displacements. At the coarsest scale, the disparity map is initialized to zero and this numerical scheme is applied to find a solution for that scale. The disparity map is then upgraded and solved at each finer scale.

The work in [25] explains this method and its implementation in detail and analyzes its performance with respect to the parameters.

### A.3 Numerical Scheme of Discontinuity-Preserving Robust Optical Flow Algorithm

The associated Euler-Lagrange equation of (34) is

$$\begin{aligned}
 0 = & \Psi'_D \cdot (v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x})) \cdot v_x(\mathbf{x} + \mathbf{d}) \\
 & + \gamma \Psi'_G \cdot [ (v_x(\mathbf{x} + \mathbf{d}) - u_x(\mathbf{x})) v_{xx}(\mathbf{x} + \mathbf{d}) \\
 & \quad + (v_y(\mathbf{x} + \mathbf{d}) - u_y(\mathbf{x})) v_{xy}(\mathbf{x} + \mathbf{d})] \\
 & - \alpha \operatorname{div}(\Psi'_S \cdot \nabla d),
 \end{aligned} \tag{53}$$

with

$$\begin{aligned}
 \Psi'_D & := \Psi'((v(\mathbf{x} + \mathbf{d}) - u(\mathbf{x}))^2), \\
 \Psi'_G & := \Psi'(|\nabla v^c(\mathbf{x} + \mathbf{d}) - \nabla u^c(\mathbf{x})|^2), \\
 \Psi'_S & := \Phi(\nabla u) \cdot \Psi'(\Phi(\nabla u) \cdot |\nabla d|).
 \end{aligned} \tag{54}$$

Provided that  $\Psi'_S$  is the only place where the discontinuity-preserving function is used, we note that the numerical scheme for this method can be very similar to the Brox method. Therefore, the numerical scheme in (46) and the discretizations in (47) and (48) can be replicated.

As shown in [16], and implemented in [17], a good choice for  $\lambda(\mathbf{x})$  is given by

$$\lambda(\mathbf{x}) := \frac{-\ln(\xi) + \ln(\alpha)}{|\nabla u(\mathbf{x})|}. \tag{55}$$

with  $\xi$  a small constant and  $\alpha$  the regularization parameter. This choice allows us to avoid some instabilities that appear when the function approaches zero and adapts the smoothing strength based on local information.

However, when the gradient of the image is close to zero,  $\lambda$  tends to  $\infty$ . Thus, this formulation must discriminate whether a pixel belongs to a homogeneous regions or not. The value of  $\lambda$  is calculated in each pixel as

$$\lambda_f(\mathbf{x}) := \min\{\lambda_\Omega, \lambda(\mathbf{x})\}, \tag{56}$$

with

$$\lambda_\Omega := \frac{-\ln(\xi) + \ln(\alpha)}{|\nabla u(\mathbf{x}')|}, \tag{57}$$

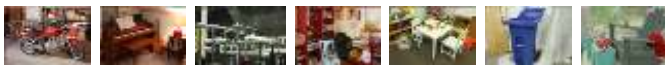
where  $\mathbf{x}'$  is such that  $|\nabla u(\mathbf{x}')|$  is the rank  $\tau \times |\Omega|$  among the image gradients, with  $\tau := 0.94$  a suitable value; see [16].

## Acknowledgements

This work was partly funded by the Centre National d'Études Spatiales (CNES, MISS Project), the European Research Council (advanced grant Twelve Labours), the Office of Naval research (ONR grant N00014-14-1-0023), the DGA Stro project, the ANR-DGA (project ANR-12-ASTR-0035) and the Institut Universitaire de France.

We wish to thank Agustín Salgado, for his help in converting the robust optical flow algorithms to 1D versions.

## Image Credits



Images from the Middlebury 2014 dataset [26]



Images from the CMLA dataset [5]

## References

- [1] S. BAKER AND I. MATTHEWS, *Lucas-Kanade 20 years on: A unifying framework*, International Journal of Computer Vision, 56 (2004), pp. 221–255. <http://dx.doi.org/10.1023/B:VISI.0000011205.11775.fd>.
- [2] J-Y. BOUGUET, *Pyramidal implementation of the Lucas-Kanade feature tracker*, Intel Corporation, Microprocessor Research Labs, (2000). [http://robots.stanford.edu/cs223b04/algo\\_tracking.pdf](http://robots.stanford.edu/cs223b04/algo_tracking.pdf).
- [3] T. BROX, A. BRUHN, N. PAPENBERG, AND J. WEICKERT, *High accuracy optical flow estimation based on a theory for warping*, in Computer Vision - ECCV 2004, Toms Pajdla and Ji Matas, eds., vol. 3024 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 25–36. [http://dx.doi.org/10.1007/978-3-540-24673-2\\_3](http://dx.doi.org/10.1007/978-3-540-24673-2_3).
- [4] I. CABEZAS, V. PADILLA, AND M. TRUJILLO, *A measure for accuracy disparity maps evaluation*, in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Csar San Martin and Sang-Woon Kim, eds., vol. 7042 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 223–231. [http://dx.doi.org/10.1007/978-3-642-25085-9\\_26](http://dx.doi.org/10.1007/978-3-642-25085-9_26).
- [5] T. DAGOBERT, *The Production of Ground Truths for Evaluating Highly Accurate Stereovision Algorithms*, Image Processing On Line, 8 (2018), pp. 1–23. <https://doi.org/10.5201/ipol.2018.187>.
- [6] J. DELON AND B. ROUGÉ, *Small baseline stereovision*, Journal of Mathematical Imaging and Vision, 28 (2007), pp. 209–223. <http://dx.doi.org/10.1007/s10851-007-0001-1>.
- [7] G. FACCILOLO, C. DE FRANCHIS, AND E. MEINHARDT-LLOPIS, *MGM: A Significantly More Global Matching for Stereovision*, in Proceedings of the British Machine Vision Conference (BMVC), 2015. <https://dx.doi.org/10.5244/C.29.90>.
- [8] L. FERNÁNDEZ JULIÀ AND P. MONASSE, *Bilaterally weighted patches for disparity map computation*, Image Processing On Line, 5 (2015), pp. 73–89. <https://doi.org/10.5201/ipol.2015.123>.
- [9] P. FUA, *A parallel stereo algorithm that produces dense depth maps and preserves image features*, Tech. Report 1369, INRIA, 1991.
- [10] G. D. HAGER AND P. N. BELHUMEUR, *Efficient region tracking with parametric models of geometry and illumination*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (1998), pp. 1025–1039. <https://doi.org/10.1109/34.722606>.
- [11] V. KOLMOGOROV, P. MONASSE, AND P. TAN, *Kolmogorov and Zabih’s Graph Cuts Stereo Matching Algorithm*, Image Processing On Line, 4 (2014), pp. 220–251. <https://doi.org/10.5201/ipol.2014.97>.
- [12] J. KOSTLIVÁ, J. ČECH, AND R. ŠÁRA, *ROC based evaluation of stereo algorithms*, Research report ISSN 1213-2365, Department of Cybernetics, Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University, March 2007. <http://cmp.felk.cvut.cz/ftp/articles/kostliva/Kostliva-TR-2007-08.pdf>.

- [13] B.D. LUCAS AND T. KANADE, *An iterative image registration technique with an application to stereo vision*, in Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81, Morgan Kaufmann Publishers Inc., 1981, pp. 674–679. <http://dl.acm.org/citation.cfm?id=1623264.1623280>.
- [14] S. MARTULL, M. PERIS, AND K. FUKUI, *Realistic CG Stereo Image Dataset with Ground Truth Disparity Maps*, Technical report of IEICE. PRMU, 111 (2012), pp. 117–118.
- [15] E. MEINHARDT-LLOPIS, J. SÁNCHEZ, AND D. KONDERMANN, *Horn-Schunck Optical Flow with a Multi-Scale Strategy*, Image Processing On Line, 3 (2013), pp. 151–172. <https://doi.org/10.5201/ipol.2013.20>.
- [16] N. MONZÓN, A. SALGADO, AND J. SÁNCHEZ, *Regularization strategies for discontinuity-preserving optical flow methods*, IEEE Transactions on Image Processing, 25 (2016), pp. 1580–1591. <https://doi.org/10.1109/TIP.2016.2526903>.
- [17] ———, *Robust Discontinuity Preserving Optical Flow Methods*, Image Processing On Line, 6 (2016), pp. 165–182. <https://doi.org/10.5201/ipol.2016.172>.
- [18] M. RAIS, E. MEINHARDT-LLOPIS, AND J-M. MOREL, *Corrélateur et interpolateur à faible complexité*, tech. report, CMLA - ENS Cachan, 2013.
- [19] F. RAUDIES, *Optic flow*, Scholarpedia, 8 (2013), p. 30724. revision #149632, <http://dx.doi.org/10.4249/scholarpedia.30724>.
- [20] N. SABATER, *Reliability and accuracy in stereovision : application to aerial and satellite high resolution images*, PhD thesis, École normale supérieure de Cachan - ENS Cachan, Dec. 2009. <https://tel.archives-ouvertes.fr/tel-00505143>.
- [21] N. SABATER, A. ALMANSA, AND J-M. MOREL, *Meaningful matches in stereovision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 930–942. <https://doi.org/10.1109/TPAMI.2011.207>.
- [22] N. SABATER, J-M. MOREL, AND A. ALMANSA, *How accurate can block matches be in stereo vision?*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 472–500. <http://dx.doi.org/10.1137/100797849>.
- [23] J. SÁNCHEZ, *The Inverse Compositional Algorithm for Parametric Registration*, Image Processing On Line, 6 (2016), pp. 212–232. <https://doi.org/10.5201/ipol.2016.153>.
- [24] J. SÁNCHEZ, E. MEINHARDT-LLOPIS, AND G. FACCIOLO, *TV-L1 Optical Flow Estimation*, Image Processing On Line, 3 (2013), pp. 137–150. <https://doi.org/10.5201/ipol.2013.26>.
- [25] J. SÁNCHEZ, N. MONZÓN, AND A. SALGADO, *Robust optical flow estimation*, Image Processing On Line, 3 (2013), pp. 252–270. <https://doi.org/10.5201/ipol.2013.21>.
- [26] D. SCHARSTEIN, H. HIRSCHMILLER, Y. KITAJIMA, G. KRATHWOHL, N. NEŠIĆ, X. WANG, AND P. WESTLING, *High-resolution stereo datasets with subpixel-accurate ground truth*, in Pattern Recognition, Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, eds., Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 31–42. [http://dx.doi.org/10.1007/978-3-319-11752-2\\_3](http://dx.doi.org/10.1007/978-3-319-11752-2_3).

- [27] D. SCHARSTEIN AND R. SZELISKI, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*, International Journal of Computer Vision, 47 (2001), pp. 7–42. <https://doi.org/10.1109/SMBV.2001.988771>.
- [28] P. TAN AND P. MONASSE, *Stereo disparity through cost aggregation with guided filter*, Image Processing On Line, 4 (2014), pp. 252–275. <https://doi.org/10.5201/ipol.2014.78>.
- [29] KUK-JIN YOON AND IN SO KWEON, *Adaptive support-weight approach for correspondence search*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (2006), pp. 650–656. <https://doi.org/10.1109/TPAMI.2006.70>.