

# RS: Epipolar Plane Image Analysis

## Application to SkySat videos

Quentin CHAN-WAI-NAM

March 12, 2018

### 1 Choice of the article and objective

We first investigated two different articles that describe two different methods for estimating depths maps from videos.

- [2] describes an algorithm that estimates the optical flow given two images using a global optimization scheme minimizing a data attachment term and a regularization using the total variation of the flow. The idea is that the brightness  $I$  of single points along their trajectories should be constant in time, leading to the optical flow constraint equation

$$\nabla I \cdot \mathbf{u} + \frac{\partial I}{\partial t} = 0$$

where  $\mathbf{u}$  is the optical flow (the velocity vector field). The article then introduces a regularization on  $\mathbf{u}$  (its total variation) and reformulates the problem in order to adapt to discrete sequences of images so that the attachment term consists in minimizing some  $L^1$  term. In order to solve the subsequent global optimization problem, the authors propose a numerical scheme based on alternate optimization scheme. Finally, the authors investigate the influence of the several parameters of the algorithm – noticeably the weight  $\lambda$  of the data attachment term – on the precision of the estimation of the optical flow and the sensitivity to noise.

In our case, the optical flow computed with this algorithm could be interpreted as some disparity measurement. An interesting point is that the computation of the optical flow can be done on any pair of images, even if not rectified.

- [1] describes a method for computing precise and exhaustive depth maps using “light fields”, that is a dense set of images captures along a linear path. By concatenating one line of the rectified images together, one obtains an “epipolar-plane image” (EPI), in which a single scene point appears as a linear trace whose slope is related to its distance to the camera. Thus, by estimating these slopes, one can reconstruct the depth of each point of the scene.

The authors use very high definition images, so that the article includes several implementation details in order to ensure computational feasibility, both in terms of space (sparse representation of light fields) and computational power. For instance, they prefer local optimization near object boundaries and propagation to nearby areas in a fine-to-coarse approach to global optimization on

the whole image. In the end, the method seems relatively fast, precise and robust to inconsistencies and outliers like noise or temporary occlusions.

We chose to work on the latter article. The objective of the project is then to implement the method presented by Kim et al. in [1] and investigate how it performs with videos taken from SkySat. We will first investigate the case in which the images from the video are pre-rectified. Then we will investigate more complicated situations, for instance increasing the density of the images and considering non-pre-rectified sequences.

## 2 Sample EPIs from Skysat (step 18)

**Test** `test_build_row_epi_skysat_rectified_18.cpp`.

We first write codes to compute and display EPIs. We investigate possible artefacts that appear in the EPIs extracted from the rectified images of Skysat, with step 18.

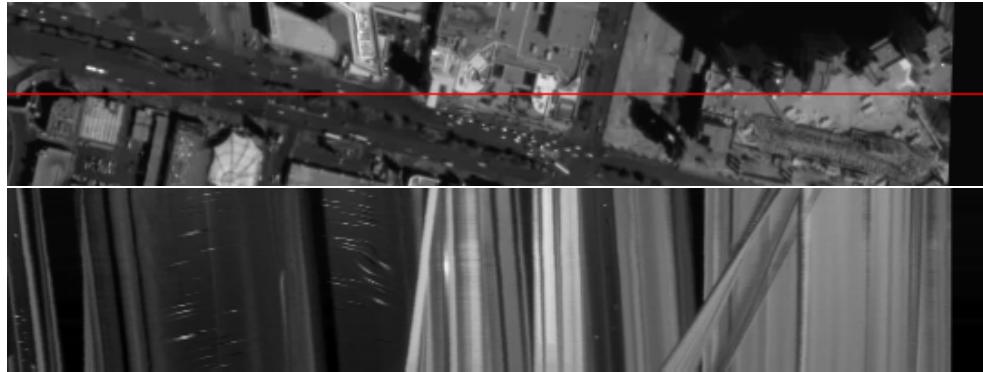


Figure 1: Row 600 – We can see that the cars produce some artefacts.

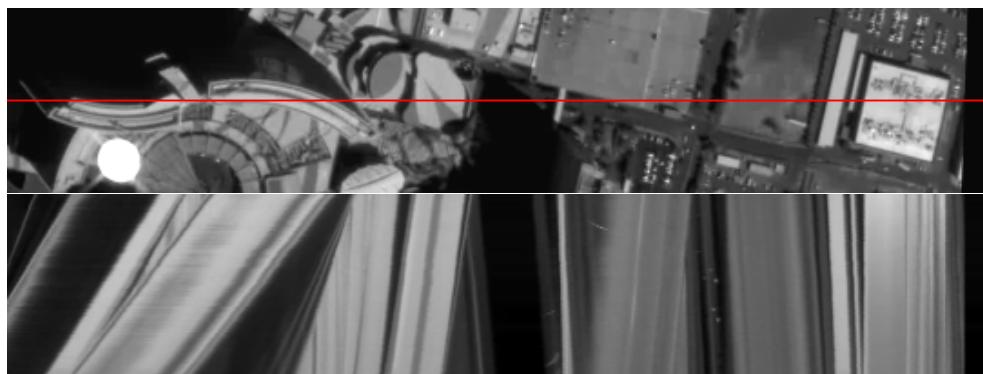


Figure 2: Row 380 – The high building has a very steep slope. The color of some points seem to vary in time (due to changes in illumination and shading). We also note some strong occlusions.

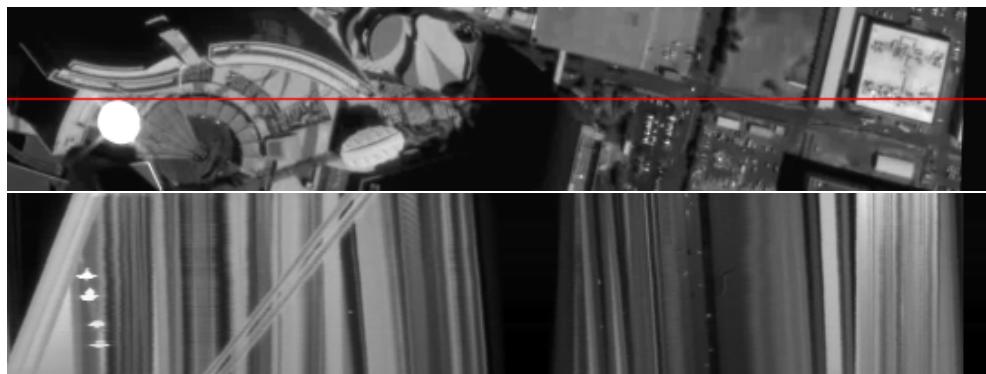


Figure 3: Row 400 – The specular reflexions produce artefacts.

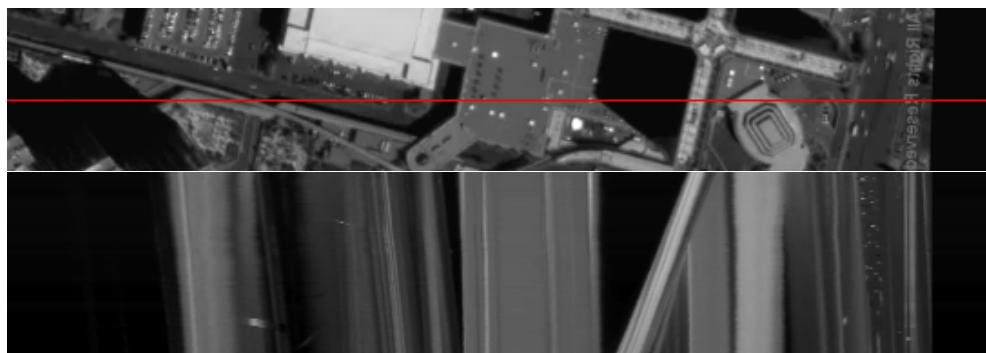


Figure 4: Row 920 – The shadows produce empty black zones.

### 3 First rough estimation of the disparities

**Principle** We apply a very simplified version of the first step of the “fine to coarse” approach of [1]: we take some EPI  $E$  corresponding to a fixed  $v$ , which dimensions are  $S$  along the rows and  $U$  along the columns. We work only on one line of this EPI (the center line, at  $\hat{s} = S/2$ . Let  $d_{\text{list}} = \{d_1, \dots, d_D\}$  be a search set of disparities. We compute the set of radiances  $\mathcal{R}$  for all couples  $(u, d)$ :

$$\mathcal{R}(u, d) = \{E(u + (\hat{s} - s)d, s) \mid s = 0 \dots S - 1\}.$$

We then compute a score  $S(u, d)$  defined as:

$$S(u, d) = \frac{1}{|\mathcal{R}(u, d)|} \sum_{\mathbf{r} \in \mathcal{R}(u, d)} K(\mathbf{r} - \bar{\mathbf{r}})$$

where  $\mathbf{r}$  can be a float, or in the case of the images of the mansion, a tricolor vector, and  $K$  is a “kernel”. We choose the kernel presented in the article, i.e.

$$K(\mathbf{r}) = \begin{cases} 1 - \|\mathbf{r}/h\|^2 & \text{if } \|\mathbf{r}/h\| < 1 \\ 0 & \text{else} \end{cases}.$$

We name this kernel the “bandwidth kernel” (and we take  $h = 0.2$  arbitrarily for now).

$\bar{\mathbf{r}}$  is a parameter that depends on  $(u, d)$ : following the article, we perform a truncated mean-shift algorithm (10 iterations), with  $\bar{\mathbf{r}}_0 = E(u, \hat{s})$  and

$$\bar{\mathbf{r}} \leftarrow \frac{\sum_{\mathbf{r} \in \mathcal{R}(u, d)} K(\mathbf{r} - \bar{\mathbf{r}})\mathbf{r}}{\sum_{\mathbf{r} \in \mathcal{R}(u, d)} K(\mathbf{r} - \bar{\mathbf{r}})}$$

prior to the computation of  $S(u, d)$ .

Finally, for each  $u$ , we select the  $d$  with the best score. We discard scores  $< 0.01$ ; if no score is available, then we take  $d = 0$ . We apply a linear median filter (along the  $u$  dimension only for now) on the resulting values  $d$  (we choose a filter size of 5).

We propagate the disparities from  $\hat{s}$  to the other  $s$  simply by drawing the depths along the lines  $u + (\hat{s} - s)d$  and respecting the occlusions (a point with higher disparity will be “occluding” a point with a lesser disparity).

**Technical difficulties** A line defined by  $(u + (\hat{s} - s)d, s)$  will very certainly end up going out of the image. Moreover, these coordinates are not integer coordinates. We thus take the following decisions:

- Values outside of the image are considered *nan* and not taken into account in  $\mathcal{R}(u, d)$ .
- We have yet to decide on some proper interpolation method for computing non-integer values of  $E$ . For now, we choose a simple linear interpolation along the line  $E(u, \cdot)$ .
- The complexity of the algorithm increases quite a bit with the dimensions of the image. We might implement the algorithms using GPUs, but now we use tricks like OpenMP.

We also created a template code so as to be able to adapt to different data types for  $E$  (e.g. `float`, `uint`, `cv::Vec3b`...).

**Results** We choose  $d_{\text{list}} = \{-2.0, -2.05, \dots, 3.95, 4.0\}$ . We end up with the following results. We expect to improve these using confidence scores  $C_e$  as in the article for instance, and computing depths for more  $s$  than only  $\hat{s}$ .

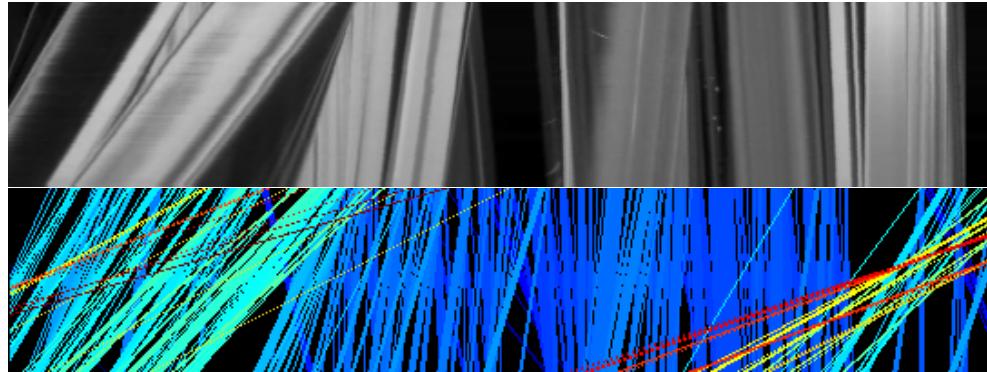


Figure 5: Skysat rectified, row 380. We note some parasite extreme lines. We interpret these as parasite low-confidence slopes.

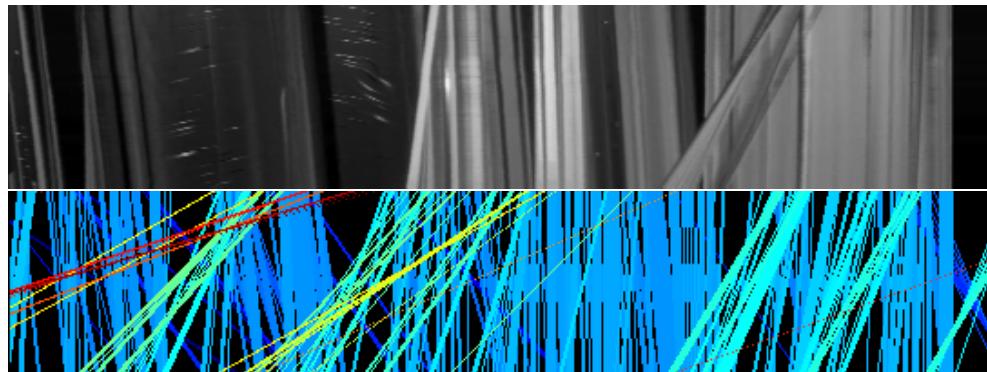


Figure 6: Skysat rectified, row 600. Cars do not seem to make a huge difference, but homogeneous zones are not well handled.

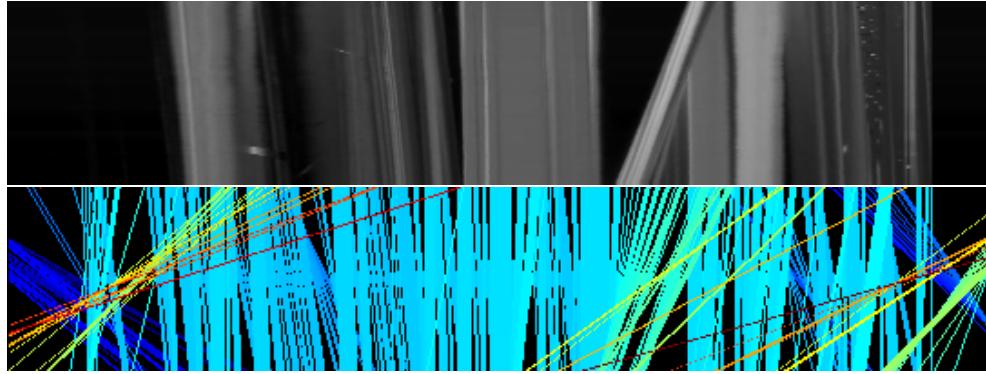


Figure 7: Skysat rectified, row 920. As expected, the algorithm does not work well in the shadows.

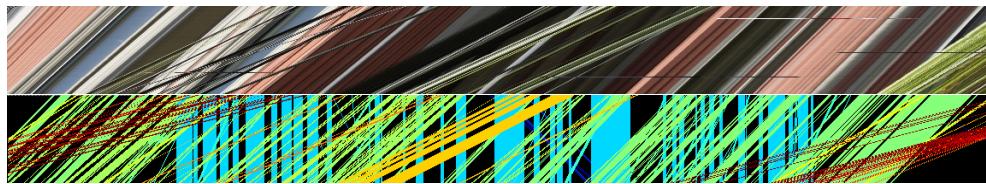


Figure 8: Mansion sample (undersized to  $1146 \times 720$ ), row 380.

## 4 After scaling, confidence threshold

In fact, it is better to scale the images between 0.0 and 1.0 before processing. We do the following:

- If the provided image is of type uchar, i.e. values are in  $0 \dots 255$ , then we scale by 255.
- Else, we scale by the max over all the channels.

Results are much cleaner (Figures 9, 10).

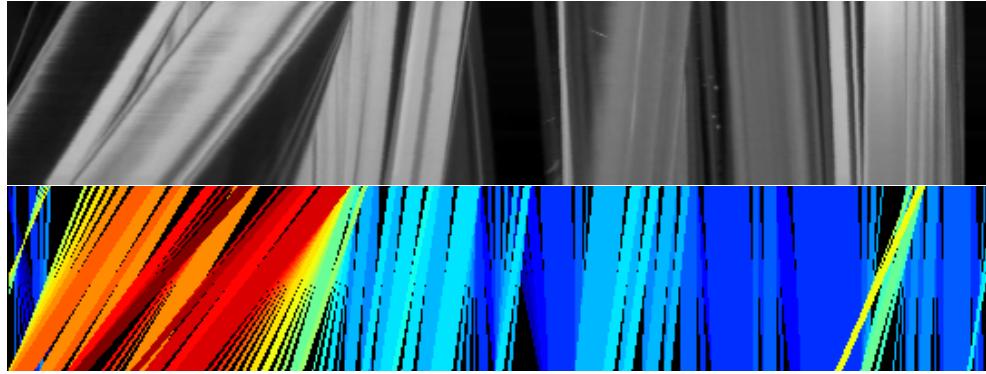


Figure 9: Skysat rectified, with scaling, row 380.



Figure 10: Mansion sample, with scaling, (undersized to  $1146 \times 720$ ), row 380.

If we add the confidence measures: confidence edge  $C_e$  and final confidence  $C_d$  such that:

$$C_e(u, \hat{s}) = \sum_{u' \in \mathcal{N}(u, \hat{s})} \|E(u, \hat{s}) - E(u', \hat{s})\|^2 \quad (1)$$

$$C_d(u, \hat{s}) = C_e(u, \hat{s}) \left\| \max_d S(u, d) - \text{mean}_d S(u, d) \right\| \quad (2)$$

we obtain the results presented in Figures 11, 12. We disable the median filter along d's for these tests.

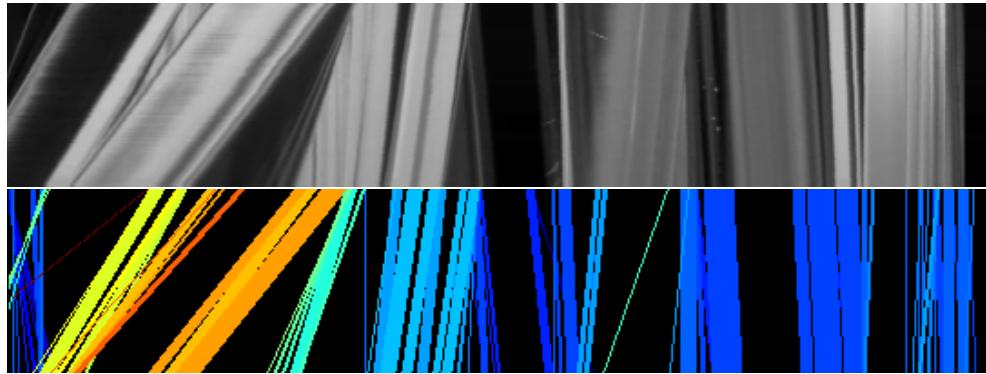


Figure 11: Skysat rectified, with scaling and score threshold, no median filter, row 380.

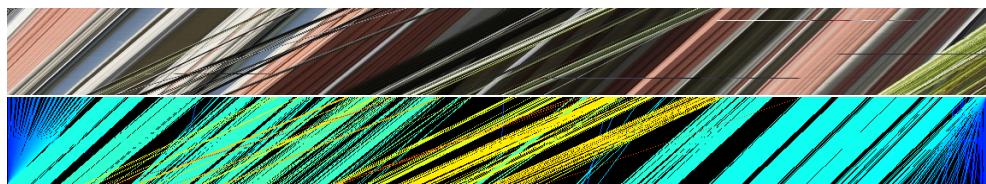


Figure 12: Mansion sample, with scaling and score threshold, no median filter, (undersized to  $1146 \times 720$ ), row 380.

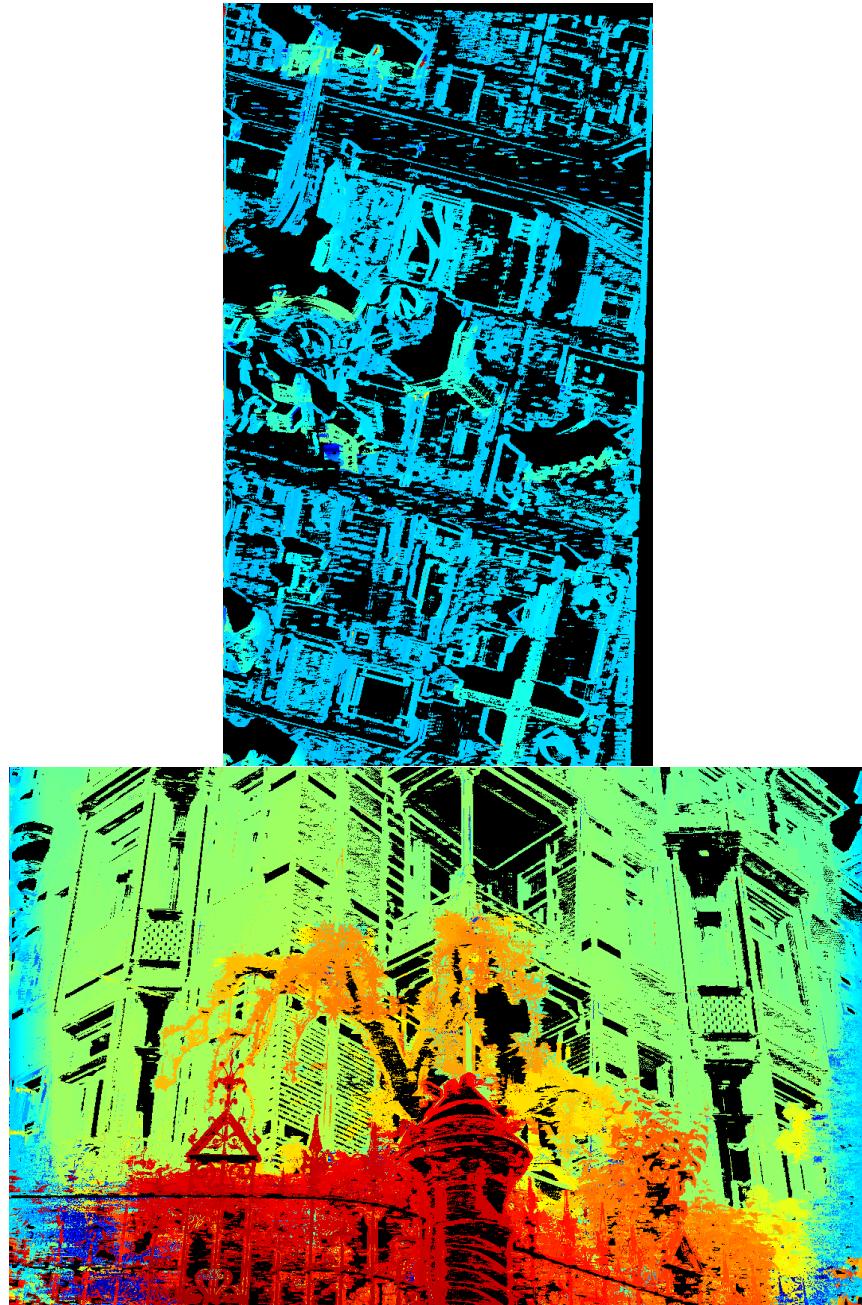


Figure 13: Sample outputs depth maps where  $C_e > 0.02$  (mansion [626 seconds] and Skysat [51 seconds] with 120 tested  $d$  values between  $-2.0$  and  $4.0$ , no median filter)

## 5 Selective median filter

We implement and apply a median filter on the resulting disparity map  $D$  along dimensions  $u, v$  such that in the end, for a point  $(u, v)$  such that  $C_e(u, v) > 0.02$ , the selected value  $d_{u,v}$  is the median value of the set:

$$\{d_{u',v'} \mid (u', v') \in \mathcal{N}_{u,v}, C_e(u', v') > 0.02, \|E_v(\hat{s}, u) - E_{v'}(\hat{s}, u')\| < 0.1\}.$$

$\mathcal{N}_{u,v}$  is a neighbourhood of  $(u, v)$  – we select a square neighbourhood centered on  $(u, v)$ , of overall size  $k$ . The results are presented in Figures x, x.

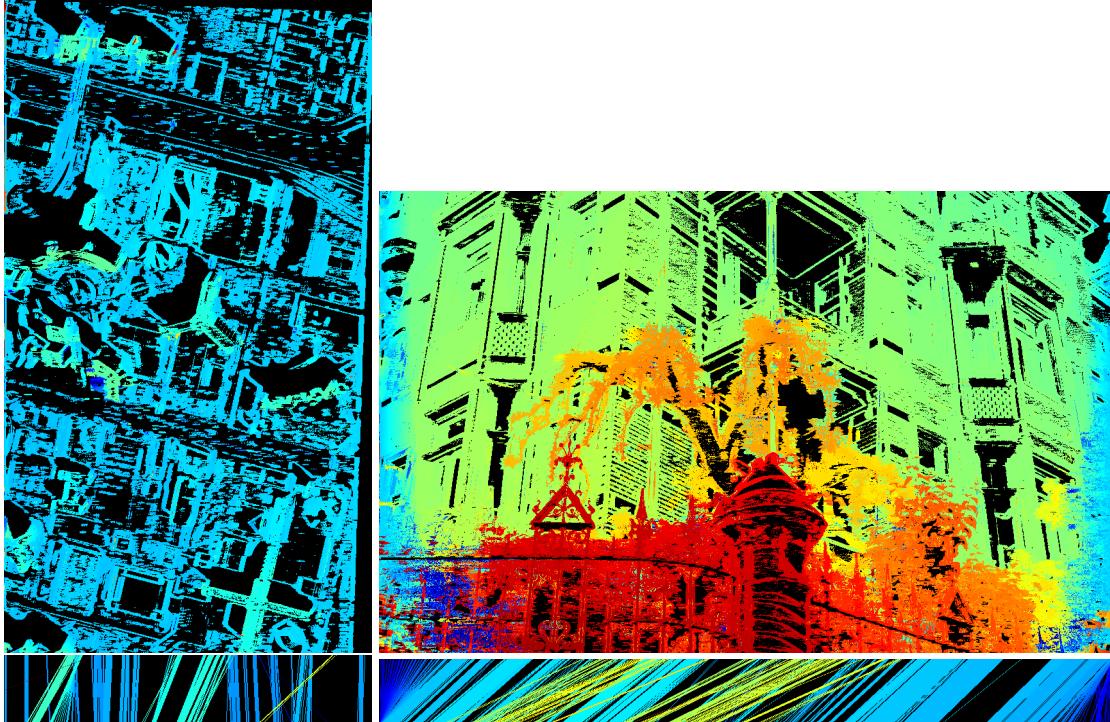


Figure 14: Disparity map and sample epi after median filter,  $k = 1$  (no median)

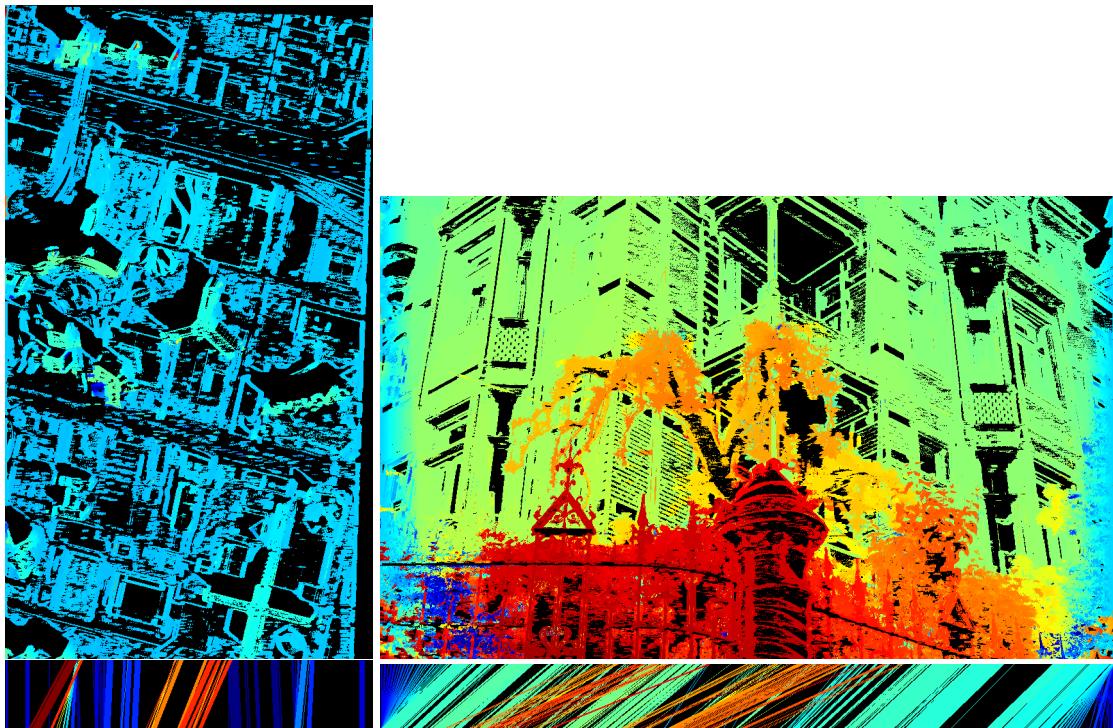


Figure 15: Disparity map and sample epi after median filter,  $k = 5$

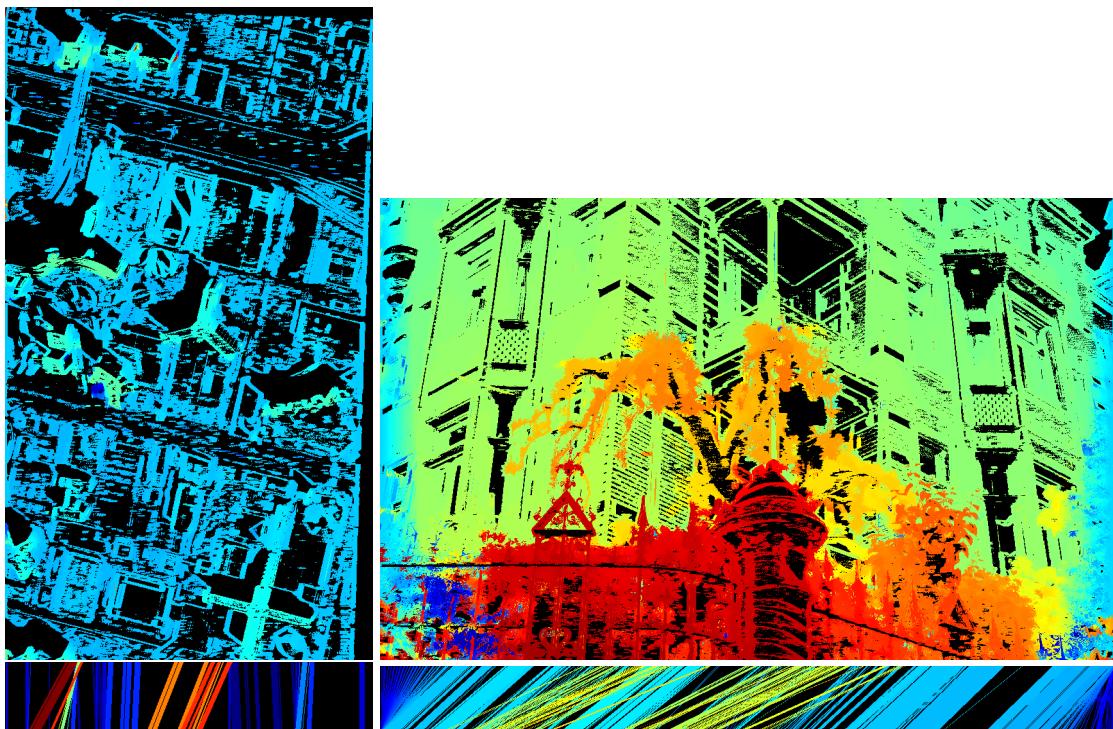


Figure 16: Disparity map and sample epi after median filter,  $k = 11$

## 6 Disparity propagation

After computing the depths for the points  $u$  with some fixed  $(v, \hat{s} = \max s/2)$ , and applying a median filter of size 5 along the  $(u, v)$  axis, we apply the disparity propagation along the  $s$  axis, using the following rules: we propagate the optimal disparity  $d(u, \hat{s})$  computed at  $(u, \hat{s})$  to a point  $(u', s')$  (where  $u' = u + (\hat{s} - s') \times d(u, \hat{s})$ ) if:

1.  $C_e(u, s) > 0.02$
2.  $C_e(u', s') > 0.02$
3.  $\|E(u', s') - E(u, \hat{s})\| < 0.1$

In the article, the first rule would have rather been  $C_d(u, s) > 0.1$ . Yet, we found that in our examples,  $C_d$  is very low for most of the points, resulting in almost no disparity being propagating, even with a low threshold on  $C_d$ . Thus, we accept to propagate depths at points at which the confidence in the value  $d$  is low. This is to be investigated further.

After propagating the depths, following the article, we compute the remaining unknown depths at points  $(u, s)$  where  $C_e(u, s) > 0.02$  for increasing  $|\hat{s} - s|$ , and propagate at each step. This leads to decreased computation times:

- For the Skysat sequence, instead of 4500 seconds (if no propagation is done), the process takes 294 seconds.
- For the downsampled mansion sequence, instead of 51000 seconds, the process takes 7371 seconds.

Some sample frames are presented hereafter. Note that we corrected a bug that perturbated the disparity computation around the borders in the mansion sequence.

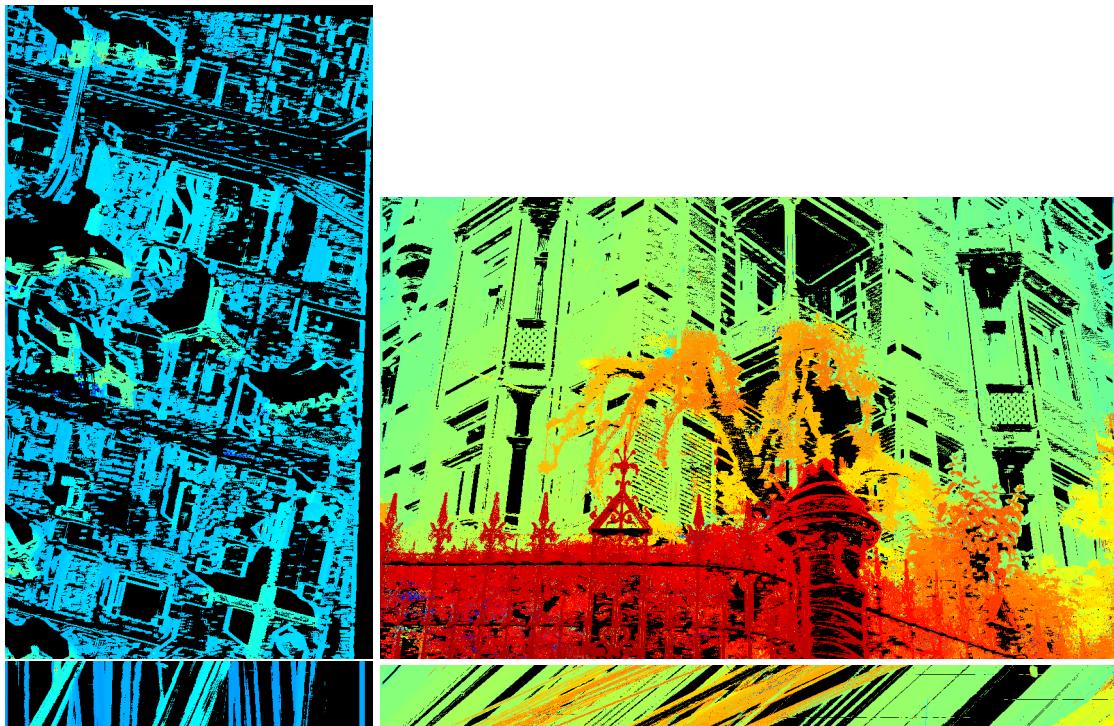


Figure 17: Disparity map and sample epi after median filter,  $k = 5$ , and propagation. The disparity maps correspond to frame 25, while the propagation originated from frame 50. The sample epis correspond to the midline of the image.

## References

- [1] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73:1–73:12, July 2013.
- [2] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. TV-L1 Optical Flow Estimation. *Image Processing On Line*, 3:137–150, 2013.