

# RS: Epipolar Plane Image Analysis

## Application to SkySat videos

Quentin CHAN-WAI-NAM

March 23, 2018

### Abstract

This document describes and implements a depth estimation method using “light fields”, i.e. dense sets of images captured along a linear path, based on a method originally presented by Kim et al. in [1]. We test the proposed approach in the context of satellite optical imaging.

## Contents

<b>1 Description of the probem and setup</b>	<b>2</b>
<b>2 Description of the method</b>	<b>3</b>
2.1 Disparity estimation from EPIs on confident points	3
2.2 Depth propagation along temporal axis	5
2.3 Selective median filtering	5
2.4 Fine-to-coarse approach	6
<b>3 Results</b>	<b>9</b>
3.1 Datasets and parameters	9
3.2 Validation on the original dataset and qualitative results	10
3.3 Influence of the candidate disparities	12
3.4 Influence of the resolution	12
<b>A Implementation details</b>	<b>18</b>

# 1 Description of the problem and setup

**Light fields** In [1], Kim et al. describe a method for computing precise and exhaustive depth maps using “light fields”, i.e. dense sets of images captured 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 which slope depends directly on its distance to the camera. Thus, by estimating these slopes, one can reconstruct the disparity of each point of the scene. Using camera parameters, one can then reconstruct the depth of each point from the measured disparity. This is summarized in Figure 1.

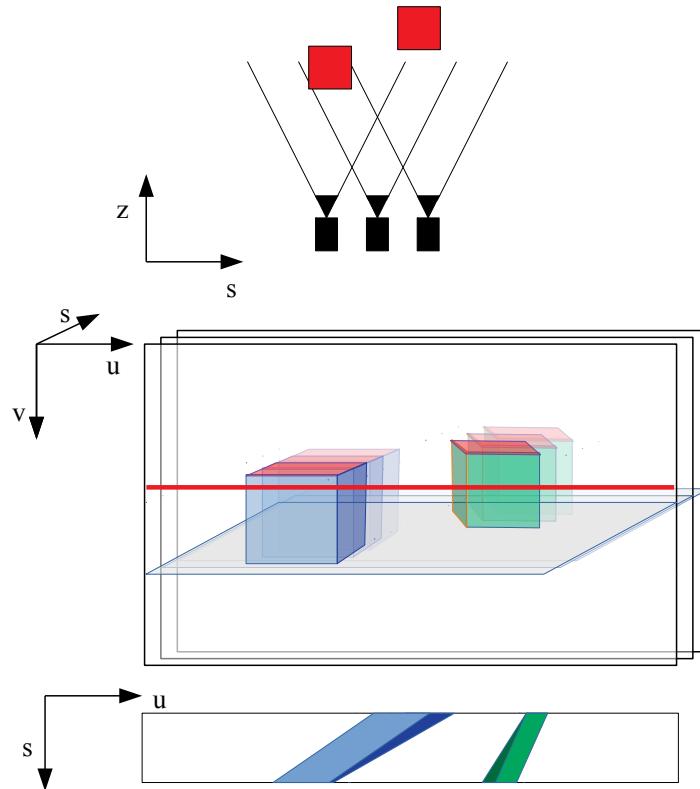


Figure 1: Light field capture process. Top: a camera captures the scene along a linear path. Middle: by concatenating the images, one obtains a pile of images with spatial dimensions  $(v, u)$  and temporal dimension  $s$ . Bottom: by slicing along a single row  $v$  (in red), one obtains an EPI (epipolar-plane image, along dimensions  $(s, u)$ ) in which single points in the scene create a linear trace.

Some real examples of EPIs are presented in Figure 2. Compared to using only pairs of images, proceeding with light fields for stereo reconstruction could enable more precise depth estimations and natural ways to handle occlusions (since an object might be occluded only on some frames and not others). Compared to direct 3D point acquisition using 3D scanners, the method should prove faster (one only has to capture the images and may leave the computation of the actual depth map for later) and more resilient to occlusions for instance (since the camera viewpoint changes, the occlusions may also change). A downside of this method is that it proves computationally expensive. This initially limits its application to low resolution images. The method presented by [1] tackle this particular issue.



Figure 2: Sample EPIs from SkysatLR18 (top) and MansionLR (bottom) datasets. We see that individual points leave a linear trace in the temporal dimension and that occlusions change over time.

The authors use very high definition images – the article thus describe 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.

**Goal** The goal of this project is to adapt the method proposed by Kim and al., that they tested on urban landscapes with mainstream cameras, to the context of satellite images. It should noted that the case of rectified satellite images correspond to case of a set of images taken along a linear path such that the ground is pushed to the infinity (thus its disparity should be zero). We chose not to focus on GPU implementation and memory management and choose to proceed with mildly high resolution images to ensure computational feasibility.

## 2 Description of the method

### 2.1 Disparity estimation from EPIs on confident points

**Principle** The main idea is, given a list of candidate disparities, to select the best slope for each point at which a confident measure can be done. Given a single line of an EPI, one first computes a confidence score. Then, for each confident point, one select the slope such that pixel radiances on the resulting line are the most densely distributed around a given mean radiance. This process is robust to occlusions.

**Details** Let us consider 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 at a time (beginning with the center line at  $\hat{s} = s_{\max}/2$ ).



Figure 3: Disparity estimation process. Left: we select a row  $\hat{s}$ . Middle: given a column  $u$ , we test several candidate  $d$ 's and select the one with the highest score  $S(u, d)$ . Right: we propagate the disparity along the temporal dimension for points belonging to a confident edge, then repeat the process along each column  $u$ , and for every value  $\hat{s}$  if  $d(\hat{s}, u)$  was not computed yet.

To each point, we define an edge confidence measure  $C_e$  such that

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

This measures diversity in the radiances of the image along a small linear window  $\mathcal{N}_{C_e}(\hat{s}, u)$  around the point  $(\hat{s}, u)$ . It should be high on edges that show great color transitions and low on flat surfaces. The process is illustrated in Figure 4.

The disparity estimation process is detailed in Figure 3. Let  $d_{\text{list}} = \{d_1, \dots, d_D\}$  be a set of candidate disparities. We compute the set of radiances  $\mathcal{R}$  of values of  $E$  along the line centered on  $u$  of slope  $d$ , for all confident couples  $(u, d)$ , i.e. at points  $(\hat{s}, u)$  such that  $C_e(\hat{s}, u) > \varepsilon_{C_e}$ :

$$\mathcal{R}(u, d) = \{E(u + (\hat{s} - s)d, s) \mid s = 0 \dots S - 1\}. \quad (2)$$

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

$$S(u, d) = \frac{1}{|\mathcal{R}(u, d)|} \sum_{\mathbf{r} \in \mathcal{R}(u, d)} K(\mathbf{r} - \bar{\mathbf{r}}) \quad (3)$$

where  $\mathbf{r}$  is a value  $E(\cdot, \cdot)$ ,  $\bar{\mathbf{r}}$  is some mean radiance value described hereafter, and  $K$  is a “kernel”. We choose the kernel presented in the article, i.e.

$$K(\mathbf{r}) = \begin{cases} 1 - \|\mathbf{r}/h\|^2 & \text{if } \|\mathbf{r}/h\| < 1 \\ 0 & \text{else} \end{cases}. \quad (4)$$

We name this kernel the “bandwidth kernel” (where  $h$  is a parameter). Its value is 0 when  $\|\mathbf{r}\| \rightarrow +\infty$  and 1 for  $\|\mathbf{r}\| \rightarrow 0$ . Its role is to select  $\mathbf{r}$  values that are reasonably near zero.

$\bar{\mathbf{r}}$  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

$$\bar{\mathbf{r}} \leftarrow \frac{\sum_{\mathbf{r} \in \mathcal{R}(u, d)} K(\mathbf{r} - \bar{\mathbf{r}}) \mathbf{r}}{\sum_{\mathbf{r} \in \mathcal{R}(u, d)} K(\mathbf{r} - \bar{\mathbf{r}})} \quad (5)$$

prior to the computation of  $S(u, d)$ , initializing with  $\bar{\mathbf{r}} = E(\hat{s}, u)$ . The expected result is that  $\bar{\mathbf{r}}$  will shift to the mean of the closest blob of radiances near  $E(\hat{s}, u)$ , ignoring the values that are too far from this value (that are expected to be occlusions).

**Technical difficulties** A line defined by  $(u + (\hat{s} - s)d, s)$  will very certainly have non-integer coordinates or go out of the image boundaries. 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 instance linear interpolation along the line  $E(u, \cdot)$ .

## 2.2 Depth propagation along temporal axis

For each  $u$ , we select the  $d$  with the best score. We then propagate the disparities from  $\hat{s}$  to the other  $(s', u')$  simply by drawing the depths along the lines  $u' = u + (\hat{s} - s)d$  if

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

These points  $(s', u')$  are marked as computed and not considered anymore. The algorithm then proceeds to new values  $s$  that are close to  $\hat{s}$  (i.e.  $s_{\max}/2 + 1, s_{\max}/2 - 1, \dots$ ) and repeat the process.

After these steps, we obtain disparity values for each point such that  $C_e$  was high enough, the other points having been masked.

**Differences with the article** It should be noted that in [1], Kim et al. handle the situation in which a point with high edge confidence  $C_e$  could have a bad disparity estimation  $d$ . This could for instance happen if the point is in the intersection of two lines of the same color with different slopes in the EPI. In order to do so, the authors define a disparity confidence score

$$C_d(\hat{s}, u) = C_e(\hat{s}, u) \left\| \max_d S(u, d) - \text{mean}_d S(u, d) \right\| \quad (6)$$

and only retain and propagate  $d$ 's corresponding to points with a high  $C_d$ . We chose not to use this conservative criterion since it decreases significantly the number of valid estimations and tend to erase mildly sharp borders on buildings in our satellite images.

## 2.3 Selective median filtering

We repeat the process for all the lines  $v$  – since the process is independant for each  $v$ , this operation can be simultaneous.

When all lines  $v$  have been computed, we apply a median filter on the resulting disparity map along dimensions  $(v, u)$  such that in the end, for a point  $(v, u)$  such that  $C_e(v, u) > 0.02$ , the selected value  $d_{v,u}$  is the median value of the set:

$$\{d(v', u') \mid (v', u') \in \mathcal{N}_{\text{sel. median}}(v, u), C_e(v', u') > \varepsilon_{C_e}, \|E_v(\hat{s}, u) - E_{v'}(\hat{s}, u')\| < \varepsilon_E\}.$$

$\mathcal{N}_{\text{sel. median}}(v, u)$  is a neighbourhood of  $(v, u)$ . The expected result is to filter speckles among points of the same color that are also close one to the other. After this step, all the disparities have been computed and filtered in points  $(s, v, u)$  such that  $C_e(s, v, u)$  was high enough.

## 2.4 Fine-to-coarse approach

The approach presented in [1] is based on a “fine to coarse” approach. It consists in making confident disparity estimations in the images for different spatial scales ( $v, u$ ) and recombining these disparities in order to reconstruct a complete disparity map at the finest scale.

In order to proceed to a multi-scale analysis, the idea is to use the regularizing effect of downsampling. We thus create a pyramid of downsampled images, with coarser and coarser disparity estimations. The confident estimations at the finest scale are likely to be made on small objects or contours, whereas the estimations at the coarser scales will be related to plain surfaces for instance. The algorithm will then merge the estimations computed at the different scales by sequentially filling the blank estimations back from coarse to fine by upsampling disparity maps at a given scale to the preceding finer scale. This process is presented in Figure 4.

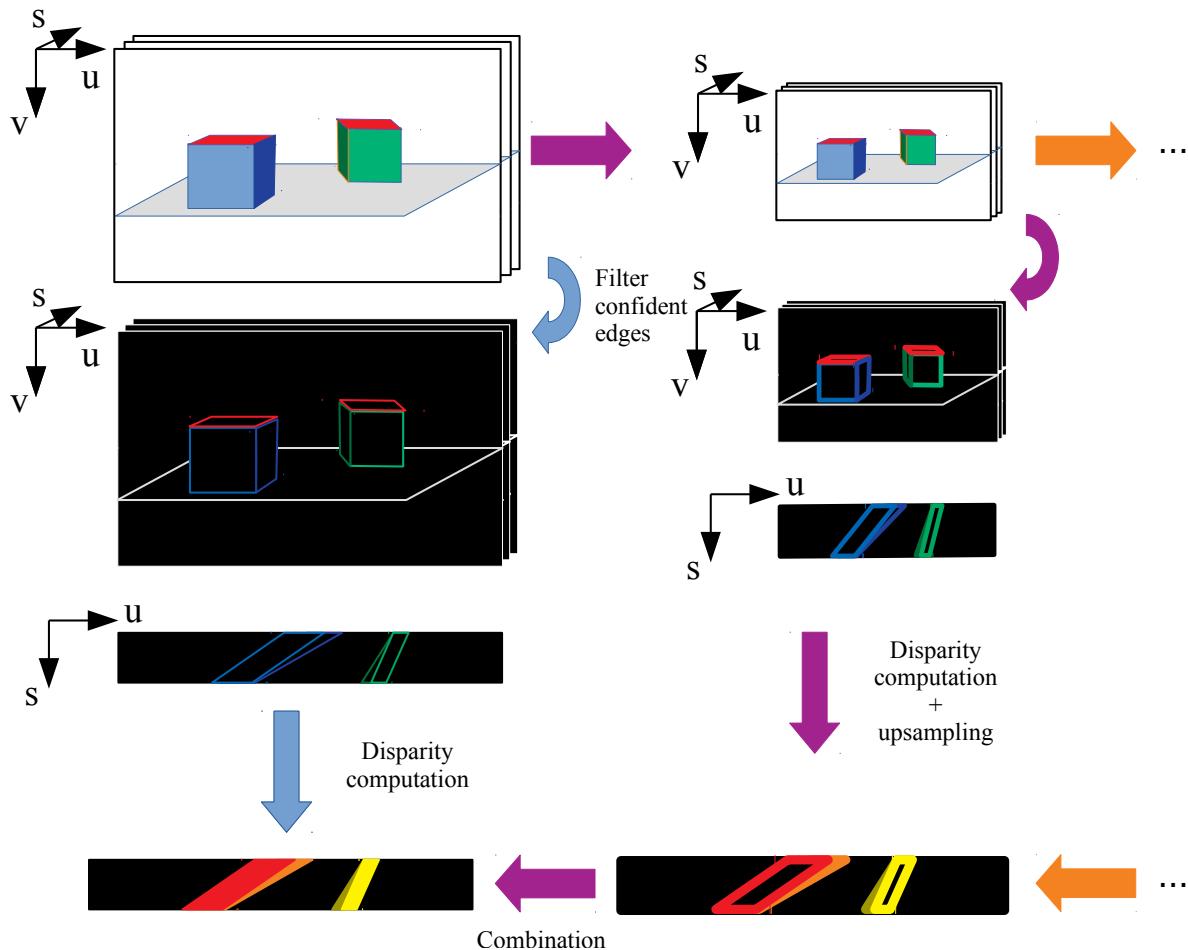


Figure 4: Fine to coarse approach.

**Downsampling** We progressively fill the zones left blank by our confidence criterion (we choose here  $C_e > \varepsilon_{C_e}$  ; one could apply  $C_d > \varepsilon_{C_d}$  like in the article but this is significantly more conservative). This approach is similar to building a pyramid of images, and writes as follow:

1. Let us consider a pile of images at the scale  $p$  (the “stair  $p$ ” of the pyramid). One first applies a Gaussian filter in the spatial dimensions  $(v, u)$  with window  $\mathcal{N}_{\text{downsampling}}$ .
2. We then downsample the images by taking one line over 2 in the spatial dimensions  $(v, u)$  ; the resolution along the temporal dimension  $s$  is left unchanged. This is the stair  $p + 1$  of the pyramid of images.
3. Then, we use the confident disparities computed at the scale  $p$  in order to derive disparity bounds for the scale  $p + 1$ . This step is not described precisely in the article ; we chose, given a point  $(s, v', u')$  in  $p + 1$ , to select the 2 pairs of points  $(s, v, [u_1^1, u_2^1])$   $(s, v + 1, [u_1^2, u_2^2])$  where  $v = 2v'$  and  $u_1^1, u_1^2$  are the first confident points at the left and right of  $u = 2u'$ . The bounds for  $(s, v', u')$  will be the min and the max disparities among these points.
4. We then perform a disparity computation along all axis as described in the preceding section.

The pyramid process is represented in Figure 5. The confidence score will be less and less restrictive due to the regularization effect of downsampling.

**Going back up to the finest scale** In order to retrieve the disparity map to the finest scale  $p = 0$ , one then proceeds the other way round:

1. Starting at  $p = p_{\max}$ , one upscales the disparity estimates (with bilinear interpolation) and the validity mask  $M_e = \{C_e > \varepsilon_{C_e}\}$  (with nearest neighbour interpolation).
2. Consider the disparity map at scale  $p - 1$ . Fill the blank zones of this disparity map using the upscaled disparity map from  $p$  at confident points from  $M_e$ .

In the end, following the paper, we also make the following refinements:

- We apply a median filter on the final disparity map estimates with window  $\mathcal{N}_{\text{fin. median}}$ .
- When establishing the map of valid points  $M_e = \{C_e > \varepsilon_{C_e}\}$ , one can perform a mathematical opening in order to remove falsely confident points. We find out that this degrades the performance of the algorithm on non-sharp images such as the Skysat sequence, so we chose to deactivate this feature.

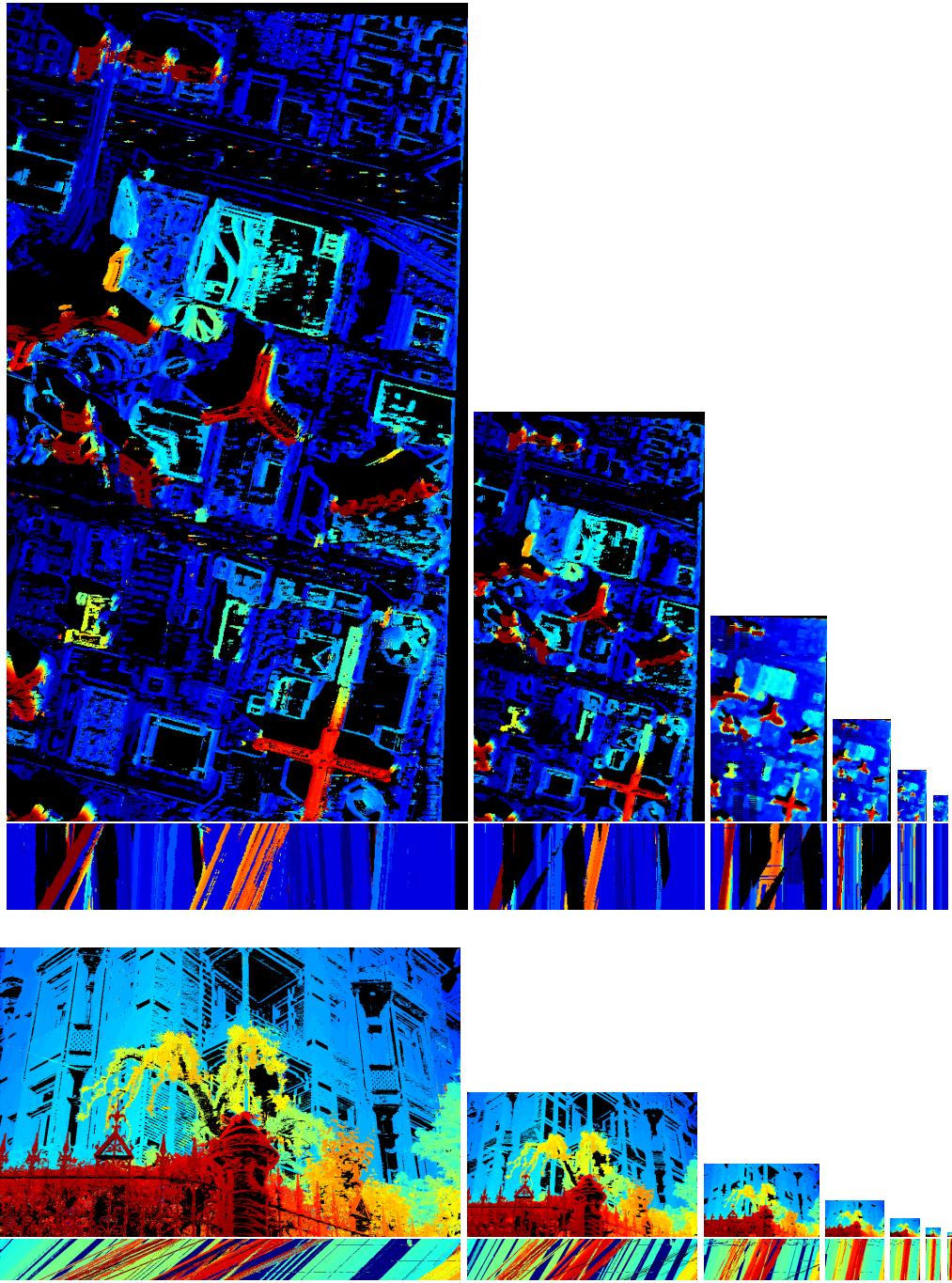


Figure 5: Pyramid of downsampled disparity maps and sample EPIs (after disparity computation) for the Skysat sequence with step 18 (SkysatLR18, top) and mansion sequence (MansionLR, bottom).

## 3 Results

### 3.1 Datasets and parameters

The parameters we used for the following results are presented in Table 1. We chose to test our algorithm on the original data of [1] along with satellite images taken from a public Skysat video. The datasets are described in Table 2.

**Parameters and number of channels** It should be noted that MansionLR is a tri-channel dataset and Skysat is a mono-channel dataset. Our kernels, confidence measure and color thresholds use the Euclidean norm of the values of images – in order to use the same parameters for all the experiments and maintain coherence between the experiments, we decide to define the parameters for tri-channel images and redefine the Euclidean norm for the mono-channel images as follows:

$$\forall x \in \mathbb{R}, \quad \|x\| = \sqrt{3} \times |x|. \quad (7)$$

We could also redefine the norm for quad-channel images for instance with a factor  $\sqrt{3/4}$ .

**Preprocessing normalization** Images can be given in the tricolor format  $(0 \dots 255)^3$  or in `float` format. We decide to convert all the EPIS in `float` as a preprocessing and scale the values by a factor 255 if the image was in `uchar`, or so that the maximum is 1.0 if the image was given in `float`.

**Shading** Satellite images show numerous dark zones due to shading. These zones are homogeneous and non-textured, so that no clear disparity estimation can be done – we decide to mask these zones both during the disparity estimation process and in the final disparity maps. We thus ignore zones such that  $\|E\| < 0.05\sqrt{3}$ .

**Colors and plots** We use the colormaps of OpenCV<sup>1</sup> in our plots. In the following, we mainly use the JET colormap.

Parameter	Description	Value
$\mathcal{N}_{C_e}$	$C_e$ computation window.	Horizontal window of size 9.
$h$	Bandwidth kernel parameter.	0.2
$\varepsilon_{C_e}$	Confidence edge threshold.	0.02
$\varepsilon_E$	Color difference propagation threshold.	0.1
$\mathcal{N}_{\text{sel. median}}$	Selective median filter window.	Square centered neighbourhood of size 11.
$\mathcal{N}_{\text{downsampling}}$	Downsampling gaussian filter window.	Square centered neighbourhood of size 7.
$\mathcal{N}_{\text{fin. median}}$	Final median filter window.	Square centered neighbourhood of size 3.

Table 1: Values of the parameters used

<sup>1</sup>[https://docs.opencv.org/3.4.1/d3/d50/group\\_\\_imgproc\\_\\_colormap.html](https://docs.opencv.org/3.4.1/d3/d50/group__imgproc__colormap.html)

<sup>2</sup><http://people.csail.mit.edu/changil/publications/scene-reconstruction-from-high-spatio-angular-resolution-light-fields-siggraph-2013-datasets.html>

<sup>3</sup><https://www.youtube.com/watch?v=1KNAY5ELUZY>

Dataset	Description
MansionLR	Mansion preprocessed dataset <sup>2</sup> , that we resize to a resolution of $1146 \times 720$
SkysatLR18	Satellite images taken from a public Skysat video <sup>3</sup> that we rectify. We proceeded first with a version downsampled to $540 \times 960$ using one frame every 18 frames.
SkysatLR01	Same as SkysatLR18 but taking every frame.
SkysatHR18	Same as SkysatLR18 but using the original $1080 \times 1920$ resolution.

Table 2: Datasets and runtimes. Each dataset sequence is of length 100 (i.e.  $s = 0 \dots 99$ ).

Dataset	Candidate $d$ 's	Runtime (s)
MansionLR	$0 : 4 [120]$	7409
	$0 : 4 [240]$	16666
SkysatLR18	$-1 : 4 [120]$	448
	$-1 : 4 [240]$	804
SkysatLR01	$-0.5 : 0.5 [120]$	143
	$-0.5 : 0.5 [240]$	254
SkysatHR18	$-2 : 8 [120]$	1714
	$-2 : 8 [240]$	2888

Table 3: Tests and runtimes. Candidate  $d$ 's are indicated by range[number of values]. Computations were run on a desktop Intel Core i3-6100 (2 cores, 4 threads @3.70 Ghz) with 16 Gb of RAM.

### 3.2 Validation on the original dataset and qualitative results

**Validation** We perform the depth estimation on one of the original datasets of Kim et al. (MansionLR) and compare our disparity map to theirs. The results are presented in Figure 6. As expected, our results are less precise, noticeably behind the tree (that is partially merged with the wall behind it). This can be explained by the fact that we used images of 8 MP (compared to 19 MP) and did not implement the conservative criterion  $C_d$  on disparity confidence (that result in more artefacts in the final estimation). The results are quite similar on other zones: the main structures seem correctly detected and estimated, especially the foliage at the right, the tree branch silhouettes in the center, the barrier at the front and the flat surfaces on the mansion wall. This validates the pertinence and efficiency of the fine-to-coarse approach.

**Limitations** The fact that we define new disparity bounds at each step ensures the coherence of the depth estimations when the depth at the boundaries of an object have been computed. It should be noted that the best estimates are made for frames near  $\hat{s} = \dim s/2$  as the propagation process might end up perturbing the estimates. This is due to changes in the image such as:

- New objects appearing, objects disappearing.
- Objects changing shape (for instance, the walls of a building are likely to change shape due to

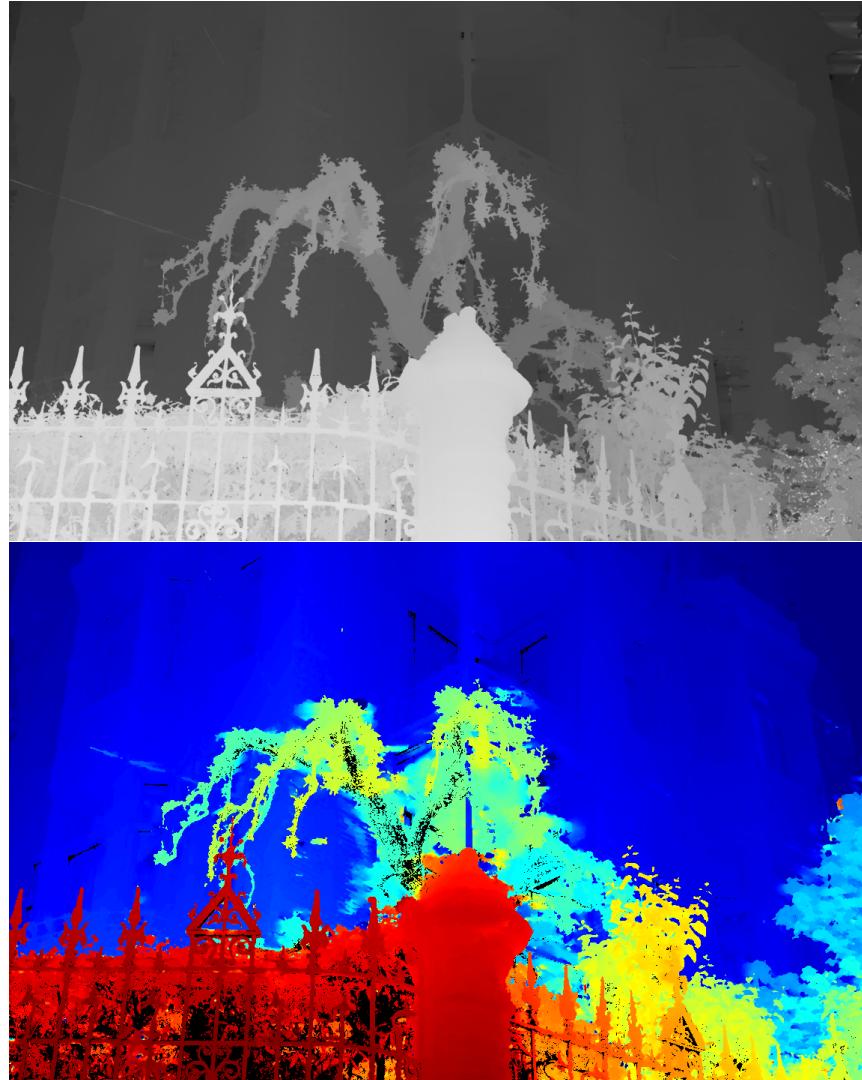


Figure 6: Comparison of the results of Kim et al. in [1] (top, computed using high definition images) and ours (bottom, computed with MansionLS, with  $d = 0 : 4$  [240]).

perspective effects).

- Global changes in illumination conditions. This is notable in the Skysat dataset.
- Shading. This appears to be a great source of errors in the Skysat dataset, as the shadows will be interpreted as rough confident edges, but are not textured, so their depth estimates will be flawed. We decided to remove the shadow areas by setting the pixels with  $\|E(s, v, u) < 0.05\|$  to  $C_e = 0$ .
- Tricky surfaces such as reflective surfaces (the glass surfaces on buildings) will be misinterpreted since they reflect the ground... and thus will have a disparity dependant on the viewpoint of the camera, with a negative sign.

**On a sequence with small slopes** We investigate the performance of the algorithm with very small slopes with a sequence of Skysat with slower variations. A sample EPI is presented in Figure 7. This example is harder since the variations are mostly under the scale of a pixel. We ran the algorithm with a choice of  $d_{\min} = -0.5$  to  $d_{\max} = 0.5$  and 120 steps. Some results are presented in Figure 9. These results are quite satisfactory given the small slopes to be measured, but show globally some underestimated slopes on the top of the buildings. We assume these are due to the fact that the disparities are harder to be estimated when undersampling - these may be improved by using better interpolation formulae.

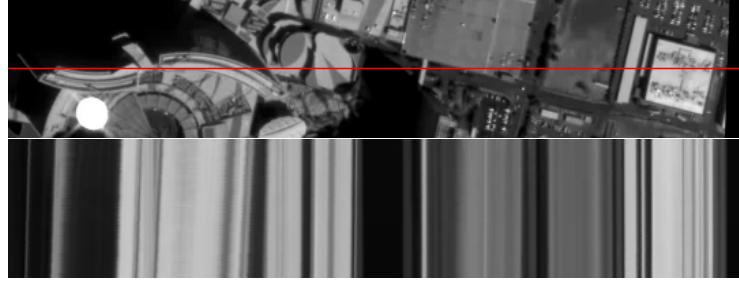


Figure 7: Sample EPI from the Skysat sequence with slow variations.

### 3.3 Influence of the candidate disparities

We compare the results for the same frames of the same datasets with a number of 120 and 240 candidate  $d$ 's between the same bounds to evaluate the influence of the candidate  $d$  set on the results. The results are presented in Figures 8, 9 and xxx.

### 3.4 Influence of the resolution

We compute the disparity estimations on the same frame Figure 11

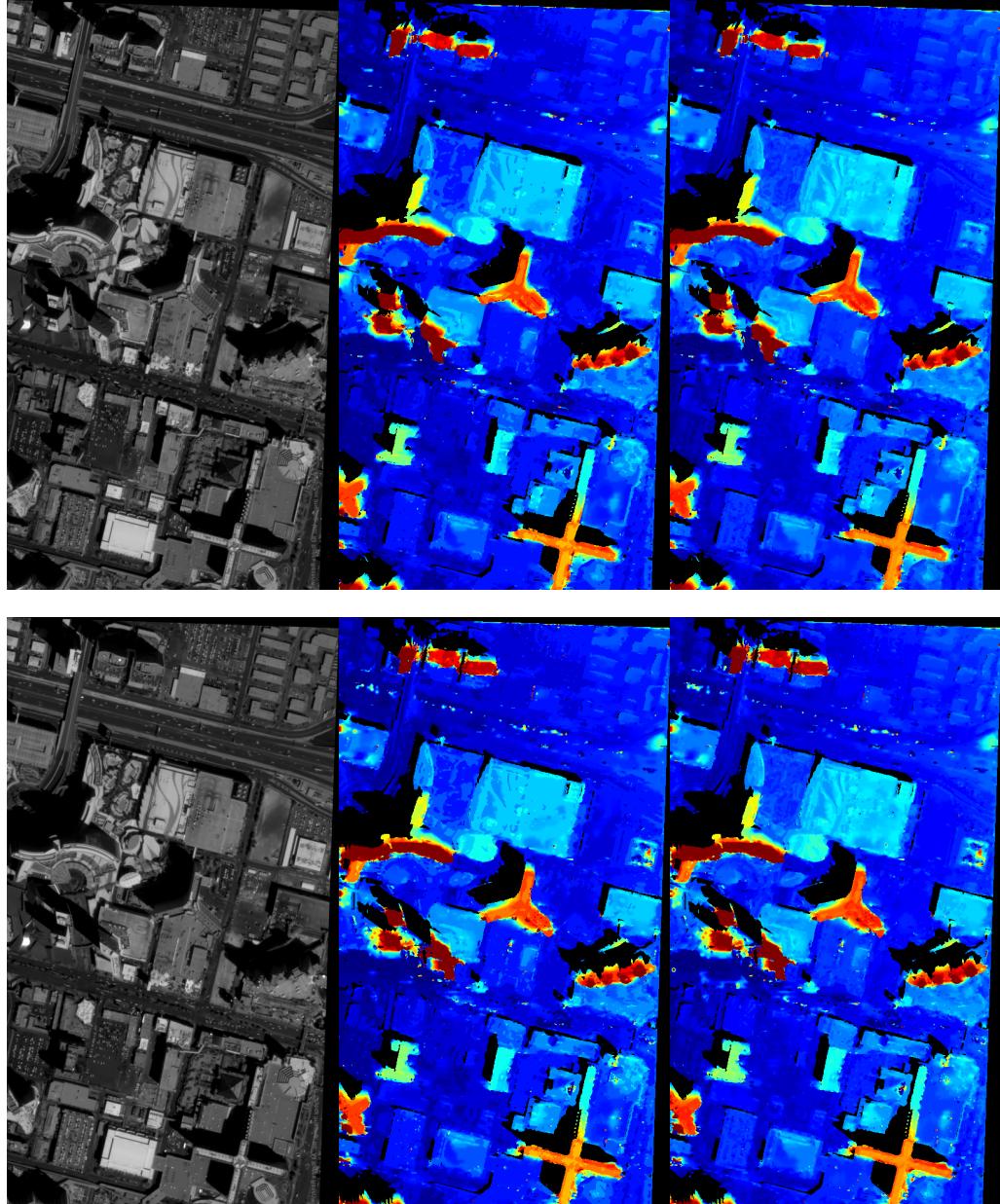


Figure 8: Influence of the candidate disparities. SkysatLR18, original frame (left) –1 : 4 [120] (middle) and –1 : 4 [240] (right). Frame 50 (top) and 25 (bottom).

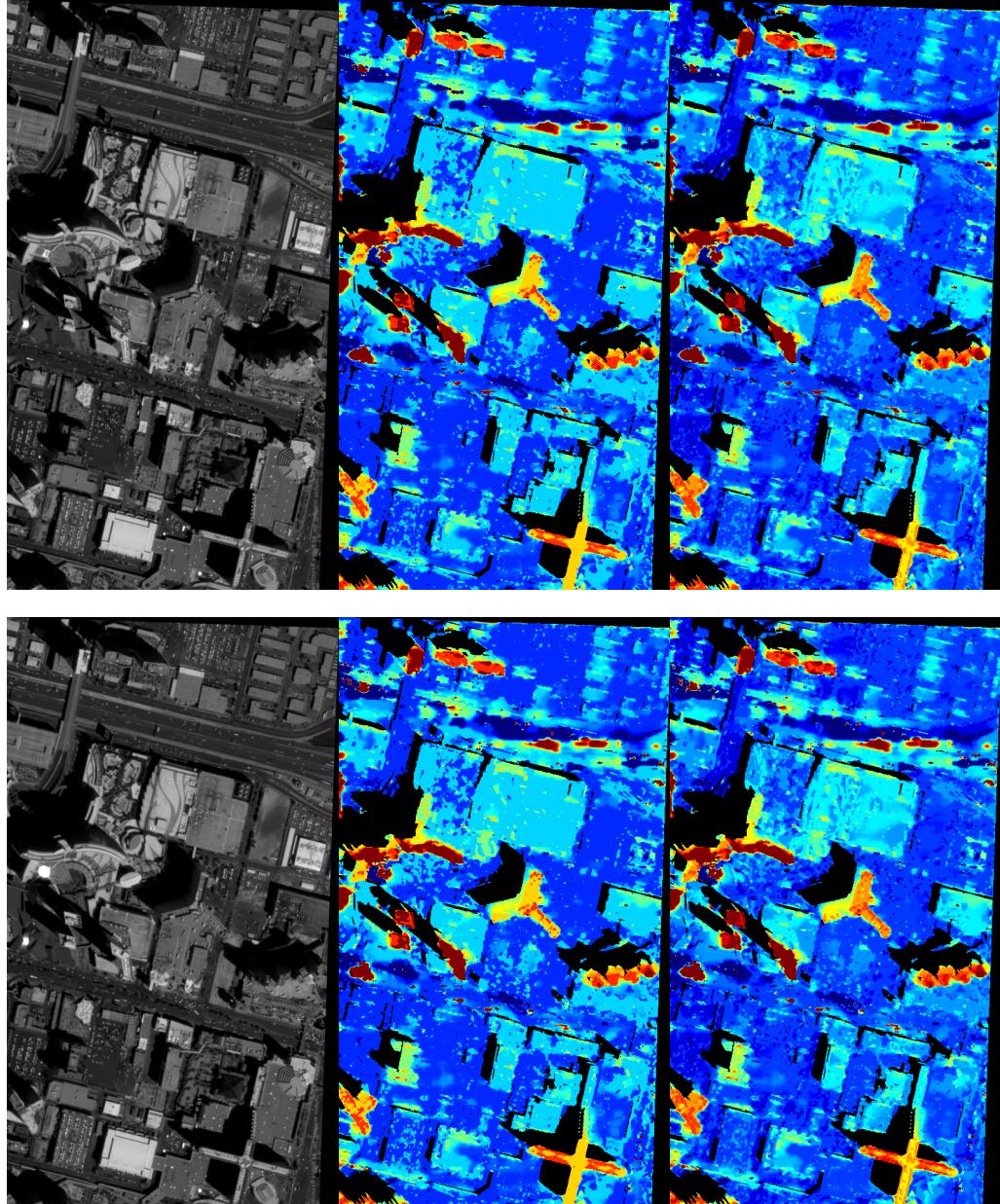


Figure 9: Influence of the candidate disparities. SkysatLR01, original frame (left)  $-1 : 4$  [120] (middle) and  $-1 : 4$  [240] (right). Frame 50 (top) and 25 (bottom).

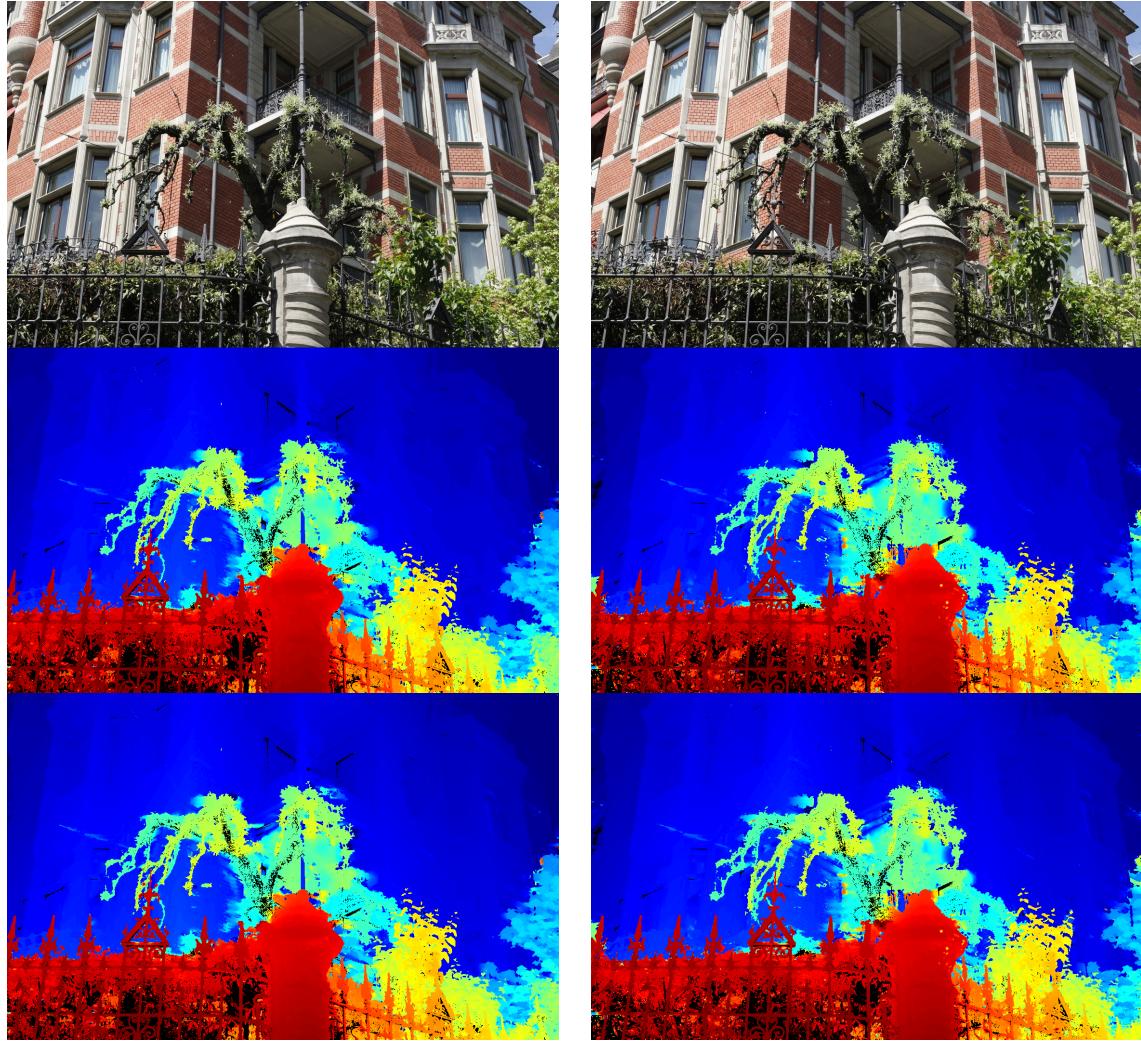


Figure 10: Influence of the candidate disparities. MansionLR, original frame (top) 0 : 4 [120] (middle) and 0 : 4 [240] (bottom). Frame 50 (left) and 25 (right).

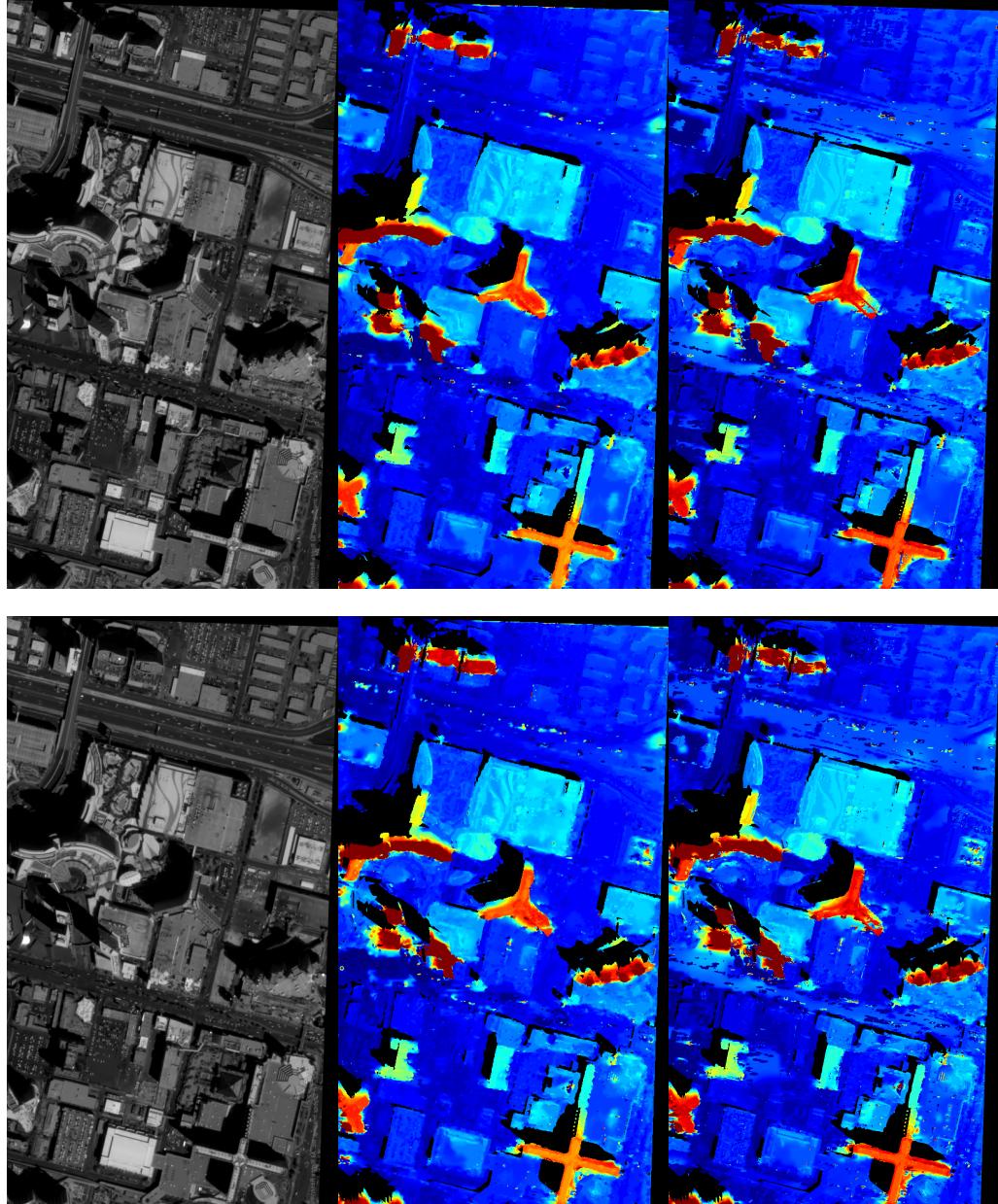


Figure 11: Influence of the resolution of the images. Original frame (left) SkysatLR18 –1 : 4 [120] (middle) and SkysatHR18 –1 : 4 [240] (right). Frame 50 (top) and 25 (bottom).

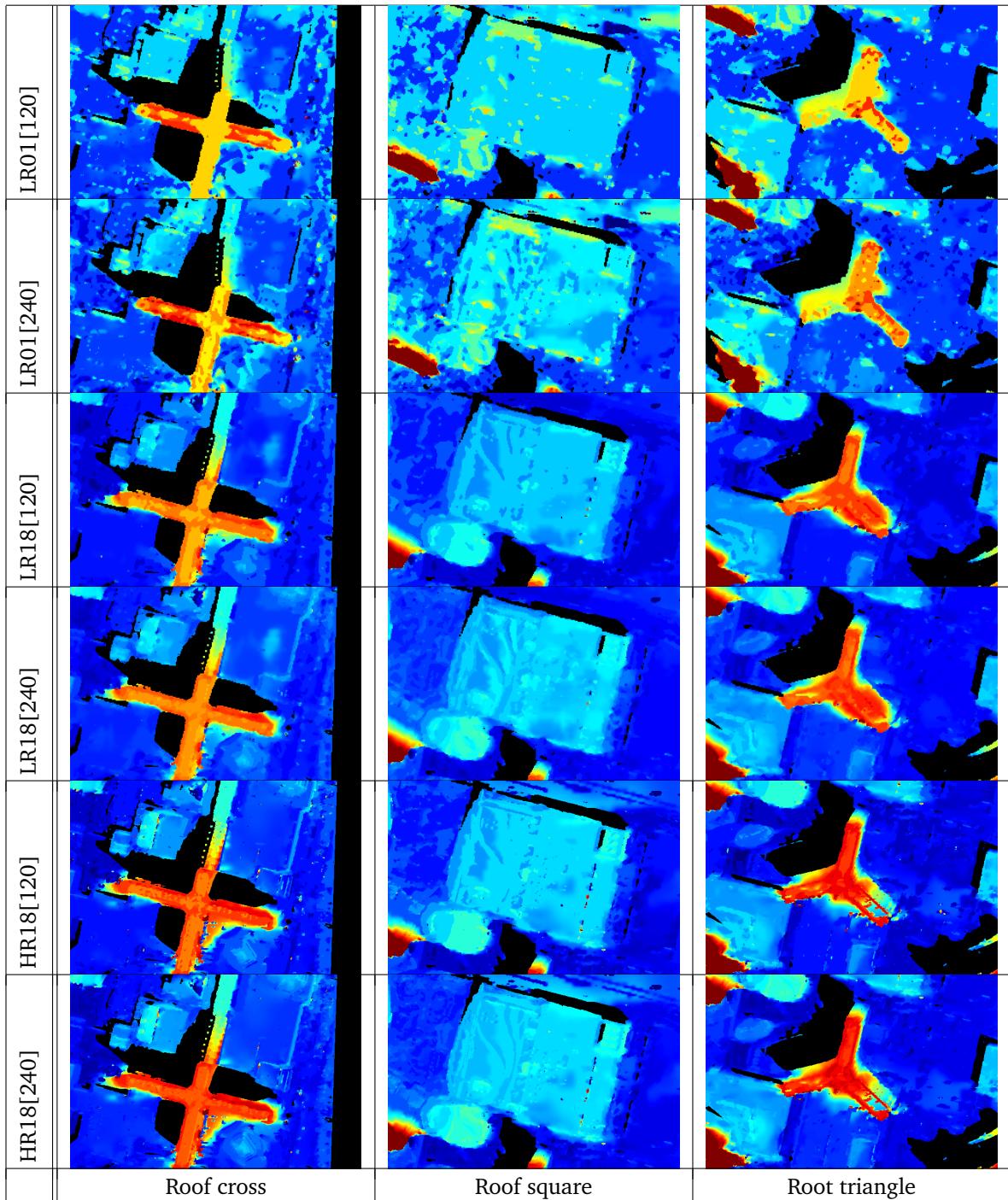


Figure 12: Influence of the resolution of the images. Original frame (left) SkysatLR18 –1 : 4 [120] (middle) and SkysatHR18 –1 : 4 [240] (right). Frame 50 (top) and 25 (bottom).

## 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.

## A Implementation details

TODO (still need to clean code)