Infrared and visible images fusion for user perceptual enhancement

Aurélien Godet *CSI*
Aurelien.Godet@grenoble-inp.fr

June 28, 2024

GIPSA-Lab - DeepRed

First year of thesis

Supervisors:
Gabriel Jobert and Mauro Dalla Mura

October 01, 2023 - September 30, 2026

Abstract

In the last recent years the use of the thermal infrared images came to be more proheminent in a wide scale of application. This thesis will put a special focus on driving data, but the goal of the subject is to propose several fusion pipeline each aiming to reach the best performances on a given application, such as enhance pedestrians perception for the drivers in difficult lighting conditions or reduce the glare of an incoming car during the nighttime. Those are common examples in Advanced driver-assistance system (ADAS) field, but we can imagine others like enhancing night vision for mountainbikers or hunters or even propose a thermal view superposition usable by anyone with a smartphone.
The final goal of this images fusion is yet to debat because in heavily relying on the purpose of the fusion. If we want to get the most natural image for human perception, the fusion process won't be the same as if we try to maximize pedestrians detection for drivers.
The main problem I tackled so far is to get perfectly aligned images with no parallax disparities as the fused image is directly seen by the final user and not a machine or a Convolutionnal Neural Network.

Acknowledgment

Thanks to Mauro Dalla Mura and Gabriel Jobert for guiding and helping me all along this first part of thesis. Thank to the DeepRed chaire to provide the funding and material necessary to work in good conditions and especially to Jocelyn Chanussot for his technical assistance. And a last thank to Alan Chauvin and Olivier Aycard for accepting to be part of my CSI along this thesis.

Contents

1 Introduction

This internship is a final year project for a DHET - Diplôme des Hautes Etudes Technologiques - at the Grenoble INP Phelma for the scholar year 2021-2022. The cursus is SICOM, majoring in signal processing and more specifically in image processing.
The internship took place from the 14th of February to the 29th of July 2022 at the GIPSA-LAB in Saint-Martin-d'Hères. It's an internship supported by the chaire DeepRed, association between Grenoble INP and Lynred a major actor on the market of infrared systems. Lynred is a well-known thermal sensor developer and manufacturer with application in a lot of different fields going from the aerospace to the defense for both industrial and public uses.
Thereafter the planning I followed during those month:

`images/Gantt.png`

1.1 Problematic presentation

The problematic of data fusion in the field of computer vision is a trending subject nowadays due to its numerous industrial applications. It's a wide open subject with a lot of contribution and papers and it may be easy to get lost in such a rich environment. The main purpose of this study was to get the best fusion method between visible and IR images for the human eye's appreciation. This subject has been widely covered through many different methods going from space-scale representation to Deep Neural Network. It stays nonetheless a challenging topic as assessing of the quality of the fusion is really subjective and application dependant.
The main application of this work is the Advanced driver-assistance system. If the safety for drivers in the car is improving years after years, an increasing proportion of the road fatalities or injuries occurring in France are pedestrians or cyclists. The figures of the French Road Safety Observatory [?] presented in Table **??**.

	Fatalities 2010	Fatalities 2021
Car User	53%	48%
Pedestrians	12%	14%
Cyclist	4%	8%

Table 1: French Road Safety Observatory 2021 figures

To provide highlight and additional information and enhanced vision to the driver may prevent a part of some avoidable accidents. In particular during nighttime or with sharp variation of illumination (like a tunnel entrance, a tunnel output or even some blinding lights from a car arriving in the opposite direction).
During this project we worked with a Database created by Lynred using two sets of infrared and visible cameras, one being the master set, the other one the slave set. That will come handy when coming to process the parallax before to fusion the images.
In the field on image processing, the Deep Neural Network (Deep NN) are in most of the situation the best choice considering the performance and the quality of the result. I chose to try out a panel of algorithmic solutions before to select a suited Deep NN in order to understand as much as possible the hidden behavior of the Deep NN.
My work will be split in two main parts: analyze the dataset and try some simple fusion methods to highlight the essential point of focus to get the best fusion and then try to answer those points as best as possible.

1.2 Dataset and Recording setup

The Lynred's dataset has been recorded within multiple sessions of several hours counting about one million frames for each 4 cameras, in daylight or at nighttime, during summer and winter and every possible meteorological conditions. For our study we will use only a portion of it with two extracts of about 2 minutes each, one during the day the other during the night. The recording devices were a couple of pairs IR - RGB cameras all set on the roof of a car on a aluminium structure as shown Figure **??**.

images/dataset/Setup1.png

Figure 1: Cameras setup

The infrared cameras integrate a VGA sensor (640x480 pixels) with a lens allowing an horizontal field of view of 42°and an aperture of f /1.2.
The RGB cameras integrate a Sony IMX273 sensor (1448x1086 pixels) cropped down to the SXGA format (1280x960)for a equivalent field of view of 45°and an aperture of f /1.4 [**?**]
When a dataset is recorded using several cameras synchronized all together, a calibration of the setup is required to get exploitable images. To do so, the usual process is to use a checkerboard as it's easy to detect the corners and match them within the different images. It was a bit more difficult to design a checkerboard visible for both the classical RGB camera and the infrared camera as well. A checkerboard made of squares cut off from foam and aluminium glued on a wood plate has been designed on this purpose. For this dataset the checkerboard has been put at a distance of roughly 5m in front the cameras and 1m bellow.

There is a baseline of 214mm between the two set of cameras and a smaller baseline of 127mm between the Infrared and Visible cameras within a set.
As the recorded Dataset is sampled at 30 Frame Per Second (fps) over several hours with 4 channels (2 Visible, 2 Infrared) the quantity of data is too important and not relevant enough to be processed fully. So I start with a small partition of the data, 2 minutes of Day recording and 2 minutes of Night recording with a total of 14400 images.

Following sections we will use this pair of images showing a scene in daylight with objects close and far to have the best appreciation of the fusion result. The example images are presented in Figure **??**.

images/dataset/setup_schema.png

Figure 2: Extrinsic calibration of the cameras

1.3 First approach: What is image fusion?

The infrared and RGB fusion takes as input two images : one visible with 3 primary colors channels (usually Red Blue and Green) the other with a lesser number of pixels and only containing one channel of heat measurement. As a first and naive approach we can take two resized images and proceed to simple fusion operations where each pixel pixel is summed with the pixel at the same position in the other image. Of course a direct weighted summation is not possible as the RGB present three layers against one for the IR images. For the sake of simplicity we will use a representation for the visible image where one of the channel is representing the luminance of the image and fuse this channel with the infrared grayscale image. To do so we have to pass the RGB image in the LAB space - presented in Annex **??**. We will refer to this technique in the following parts of the document as "luminance fusion", see Figure **??**.

When we take a closer look to the result of the fusion we can see the next problem we have to solve in order to have a good looking result of fusion : the parallax effect.

[1]The faces and license plates have been blurred after processing, in accord with the General Data Protection Regulation, L119, 4 May 2016, p. 1–88 [?] (GDPR)

images/first_approach/vis_ex.png images/first_approach/inf_ex.png

Figure 3: Image color and Infrared of a daylight scene[1]

2 Parallax correction

The first problem we encounter to fusion the infrared information with the visible image is to correlate each pixels from an image to another. It's a problem of stereo vision. The first thing to do is to align every images of the dataset in order to compute the disparities. After a brief study of the stereo vision principle I chose a method to compute the pixel-wise disparity within the stereo pairs.

2.1 Stereo Vision

The principle of stereo vision is well known for a long time as it's exactly how human' sight works. Two input images slightly apart by a distance called "Baseline" allow the brain to evaluate the depth. Of course our brain is using much more information than just the signal from the eyes, like knowledge of object and prior about those (shape, size, color...) to estimate their position and distance in space. That's why we still have a certain appreciation of the depth even using just one eye. The geometrical explanation of the stereo vision is simple: the further is an object, the closer it will appear from one to another in the image, Figure **??**.

The computer - or human brain - try to find related pixels in both images to estimate the distance between them. This distance is called "Disparity". Thus, using those both images we can draw a disparity map for each pixel, which will have the same size as the input image. There are two disparity maps for each image couple as the value of the disparity is linked to a pixel position in the image, Figure **??**.
There are many techniques of depth estimation in the literature. This is still nowadays a burning topic with numerous industrial applications. Lately the Deep Neural Network show performances way above the previous used algorithms.

images/first_approach/schema.png

Figure 4: Luminance fusion process

2.2 Disparity map

A quick overview of the different existing techniques to compute the disparity between a stereo pair of images present me with several possibilities. To try them out I used a couple of classical artificial stereo images known as 'Teddy' and 'Cones'. I also kept the same image as before from the Lynred's database as a real image for validation purpose.

2.2.1 Semi-global matching

The first and most accessible method is the classical Semi-Global Matching (SGM), which is a compilation of matching cost of a sliding window from an image over the other. This requires of course the images to be aligned in prior to apply the algorithm as the sliding window, for efficiency reason, compute matching cost only along the horizontal axis. There are two versions of this algorithm implemented in OpenCv library: Classical SGM and SGBM (for semi-global block matching). The second one is a modified version on the classical SGM algorithm by H. Hirschmuller [?].

The result of the SGBM is always better the classical SGM, but the computation time is about ten times higher(0.2 sec). If the disparity map seems good enough here, let's remember that's a toy example made from artificial images. So the alignment is perfect and the scene well exposed.

2.2.2 Deep Neural Network : ACVNet

There are plenty of deepNeural network design for this task. It's even easy to find some with pre-trained weights, which is easier to test and assess their performances. I chose a recent images/first_approach/luminance_para.png

Figure 5: Parallax effect

images/parallax_correction/stereo_cones.png

images/parallax_correction/stereo.jpg

Figure 6: Principle of stereo vision, Illustration from [?]

network design, submitted to Kitti Stereo Benchmark. It's one of the first Network with code and weights available in the leaderboard and it's inference time should be one of the lowest aswell, which is perfect for our application. [?].

Using this network the result is way better, with sharper edges and homogeneous areas. This will be our choice for the best pipeline possible. However the inference time is about 0.3 second which is not fit a real-time application. For the moment the goal is to produce the best looking result possible without the real-time constraint, thus this will be our choice of solution

Figure 7: Left Teddy and Cones artificial images and Lynred's database example

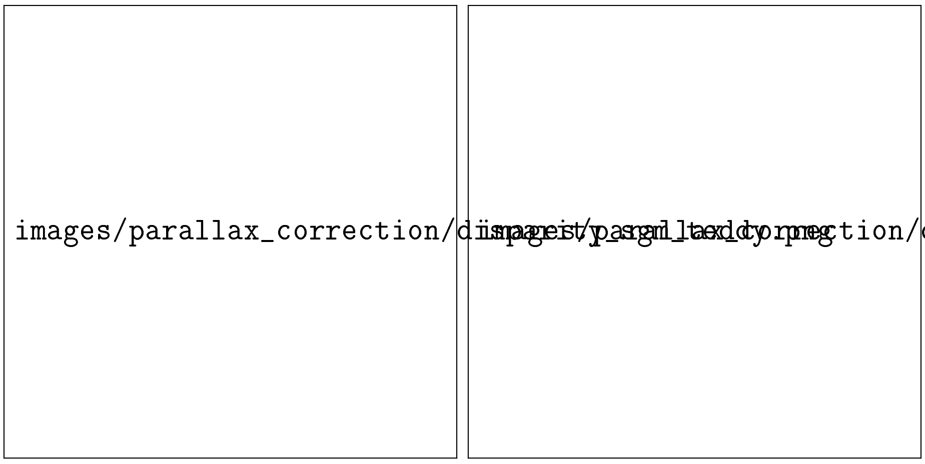


Figure 8: Teddy SGM & SGBM disparity

for the next part.

2.2.3 Monocular depth estimation : monodepth2

The Deep Neural Network try to mimic the human brain comprehension of the world for a particular task. As our understanding of the Deep Neural Network grows, some people try to work on the depth estimation problem but using a single input image. As I said earlier, our brain is able to have a good estimation of the depth using one eye because it is use to see the same object and remember their shape, texture, color, etc. Using those contextual pieces of information and some other like the illumination of the scene for example we are able to have a spatial representation of our surroundings. The Deep Neural Network I used [?] here is train to estimate the depth using a single input image. As the training set was outdoor images of urban scene and taken from a car, the Network is efficient mostly for this kind of input in the daylight. As we could expect, the result is less sharp and accurate but we obtain nonetheless a rough estimation of the disparity. The inference time is also much lower for this Network than ACVNet, that's also why this solution stays interesting.

2.3 Images alignment

Before to apply the Neural Network and draw the disparity maps we will need to align as accurately as possible our images. The convention for stereo matching in application such as driving is to aim all the cameras to the infinite. The closer an object, the more disparity we get in the image. Here our Dataset has been recorded with a focus point or 'towed-in' at about images/parallax_correction/teddy.png images/parallax_correction/cones.png

Figure 9: Teddy & Cones ACVNet disparity

images/first_approach/vis_ex.png images/parallax_correction/monocular.png

Figure 10: Monocular disparity estimation by monodepth2

5m. So we need to rectify that in order to get good result with the Neural Network.

2.3.1 Automatic registration : SIFT

To align automatically a bunch of images could seem like a trivial task, but when the exact position and orientation of each camera is not known it's become a bit more complex. The images needs to be rotated and translated over the 3 axis in order to have a perfect superimposition of the furthest objects and a horizontal alignment over the whole image. Of course the transformation is never perfect as each lens has its distortion and each optical systems its field of view. The easiest way but letting close to zeros control over the alignment distance within the image is to use automatic keypoints matching technique, as SIFT (Figure **??**).

But as we want to align the images only on the furthest plan possible, we need to add a constraint to keep only points from the background. For the towed in images, the further the objects the maximum the disparity. So we could add as criterion to sort the keypoints a minimum disparity threshold to have between a matched pair of points. We optionally can add another threshold discarding points matched with a too large Z disparity, as the image should be horizontally aligned (at +- 5 pixels...). With these constraints the keypoints finding becomes Figure **??**. Now we can use those points to estimate the homography matrix to align images/parallax_correction/SIFT.png

Figure 11: SIFT Keypoints finding

one image on the other (Figure ??).

If this technique is efficient to match points within well resolved clear image, the algorithm is not performing well for a hybrid RGB - IR couple or a image taken during nighttime where the visibility is much lesser. The keypoints descriptor use by the SIFT algorithm is not designed to overcome the structural difference between a infrared and a RGB image.

2.3.2 Manual method

The method I chose to use is to set manually each parameters on well chosen example images (one for the day, one for the night and for each cameras). To evaluate the quality of the alignment I display the result of the disparity estimated with the newly aligned pair using ACVNet presented above. This method is a bit subtle to set properly but it needs to be done just once per image couple. We need to do this calibration for the couple RGB master - RGB slave and for the couple IR Master - RGB slave. The main limitation here is that the disparity shown for the hybrid alignment is really noisy and difficult to optimize to get the right parameters, see Figure ??. That makes sens because that exactly why we are constrained to use the visible stereo couple to estimate the disparity.

Of those 6 values of rotation, and translation we deduce the transformation matrix we have to apply to all the image of the dataset. The Figure ?? shows separately the matrices deduces from this parameters. Once the projection is done the images are cropped to show only the areas where both images carry information.

images/parallax_correction/SIFTplus.png

Figure 12: SIFT Keypoints finding with minimum threshold at 105px

2.4 Disparity estimation using the stereo visible channels

Considering the setup shown in Chapter **??**, we can use the aligned visible channels to compute a disparity map. The ACVNet Neural Network return the disparity between the left and the right image projected on the left image.

This is perfect for our application because we want to project this computed disparity on the infrared left image located between the two visible cameras. As a result we get the disparity map of the left input image with respect to the right input image. Of course as the disparity is expressed in pixels, the scale of the disparity map is directly proportional to its value. If we need to downscale the map by a factor 2, the values will be downscaled by a factor 2 as well.

2.5 Disparity projection on the infrared master input

The next step of our pipeline is to project the disparity map of the master visible channel onto the master infrared channel. We will use the proportionality between Baseline and disparity to do so :

$$disparity = \frac{Baseline * focal}{Depth} \tag{1}$$

The schema thereafter **??** shows a close view of the disparity value within an image. This

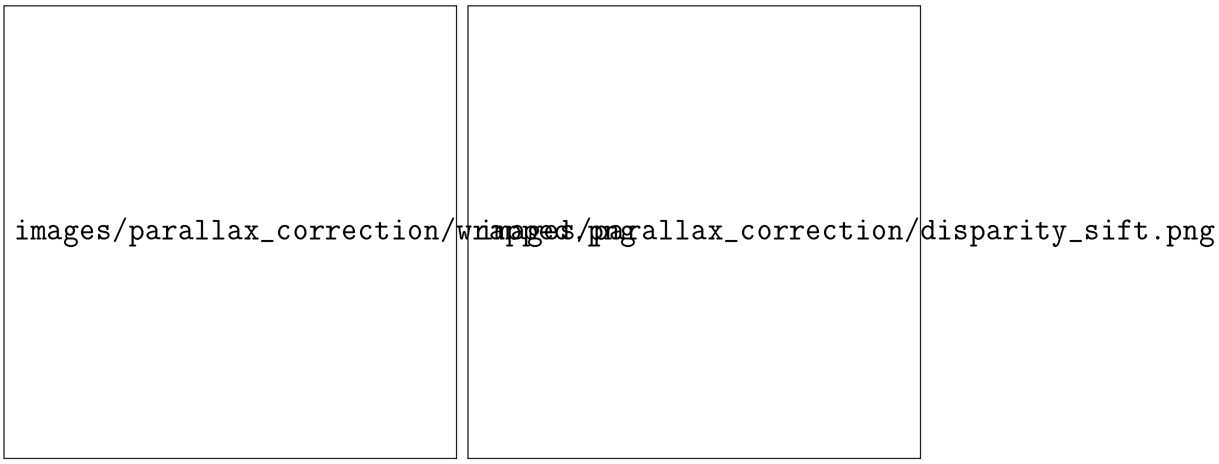

images/parallax_correction/wrapped.png
images/parallax_correction/disparity_sift.png

Figure 13: Superimposition of the left aligned and right images

Figure 14: Window of manual image alignment Visible - Visible

disparity is estimated as float number by the Neural Network but as the reconstruction method use integer values of pixel translation the value are rounded.

For the projection of the disparity map, we consider all the cameras to be aligned. The baseline between the RGB master camera and IR master camera is about $127mm$ while the disparity has been estimated with a baseline of $127 + 214 = 341mm$. The disparity being proportional to the baseline with respect to the formula **??**, we can compute the new value of the projected disparity map as a fraction of the original disparity value.

$$Disparity\ image = \frac{Disparity * 214}{341} \tag{2}$$

$$New\ disparity = \frac{Disparity * 127}{341} \tag{3}$$

Now that we know the value of disparity of the master IR image we need to project those images/parallax_correction/image_alignment.png

Figure 15: Window of image alignment Infrared - Visible

images/parallax_correction/matrices_diagram.png

Figure 16: Different matrix of geometrical transformation

values onto their related pixel. To do so we re-build the image layer after layer starting with the lowest disparity. The algorithm used is the algorithm **??**.

The Input used for this algorithm in this case is the *Disparity image* computed in equation(**??**) and the *New disparity* values from equation(**??**) as Disparity. The Figure **??** shows the output at different step of the algorithm on the disparity map.
The last step to obtain the final disparity map projected on the infrared image is to crop the output disparity map by the range between the $min(Disparity)$ and $max(Disparity)$ in pixels. Once we obtain the disparity map projected and we crop it to fit pixel-wise to the infrared image, we can finally make the last step and truly correct the parallax in the infrared image.

Figure 17: Disparity estimation from visible stereo pair

Algorithm 1: Reconstruction from disparity

```
Data: $Input$ and $Disparity$
Result: $NewImage$
for $i = min(Disparity)$ to $max(Disparity)$ do
    $Mask = (Disparity\_map == i)$;
    $NewImage[Mask\ translated\ of\ i\ pixel] = Input[Mask]$;
end
```

2.6 Final projection: parallax correction

The last step is easy once we have the tool ready. We just have to apply the algorithm **??** with as "Input" the infrared master image and as "Disparity" the projected disparity obtain just before. The occluded pixels are filled with in-painting.

Once we get the newly corrected infrared image, we can apply the fusion method that we want. The quality of the reconstruction depends heavily on the quality of the initial disparity map. For that reason, the parallax correction is a bit more noisy during the night, as the input images are more blurred and less contrasted.

2.7 Summary on parallax correction

To conclude this part about parallax correction, let summaries the process pipeline:

- Image alignment (automatic via SIFT or manual via the interface, the calibration is valid as long as the cameras doesn't move)
- Disparity estimation of the master RGB image (via ACVNet or monodepth2, or using a SGBM algorithm)
- Projection of the disparity map from master RGB to master IR.

Figure 18: Zoomed view of a disparity image

- Projection of the master IR image using the new disparity map onto the slave RGB image

In order to skip most of these reconstruction steps we may train a Neural Network base on the same architecture as ACVNet but with as entry the RGB and the IR image directly. As the images are really different during the night and the day, two separate trainings may be done to specialize the network for each condition.
To apply the method presented here with only one RGB and one infrared camera, we need to rely on the monocular depth estimation of the scene either using the infrared channel - which required to re-train the neural network using infrared inputs - or the visible channel and transpose this depth estimation onto the infrared image. It's important to transform the infrared image and not the visible as the RGB channel carries more information, has 4 times more pixels and will support most of the fusion final rendering, it's for the best to let it untouched.

2.8 Metric to assess the reconstruction quality

If the final purpose of this fusion is the visual rendering for the users, we have to find a way to assess the quality of the output. This estimation metric is really important because it can be use to train a future network design in order to maximise the output information quality.
There are plenty of indexes to compare two images, but we have to know what is relevant to compare in our case. As the intensity of the infrared pixel is only linked to the body heat radiance, we can't compare intensity between the two images as we would do with regular RGB images. The only common features in between those two type of images is the edges of the objects. The idea then is to compare the contour of the object in the two images and compute the correlation between the two.

2.8.1 Construction of the comparison images

From the infrared and the visible image we extract the edges information using a simple Sobel filter convolved in the 2D with the images (Matrices of the filters **??**)

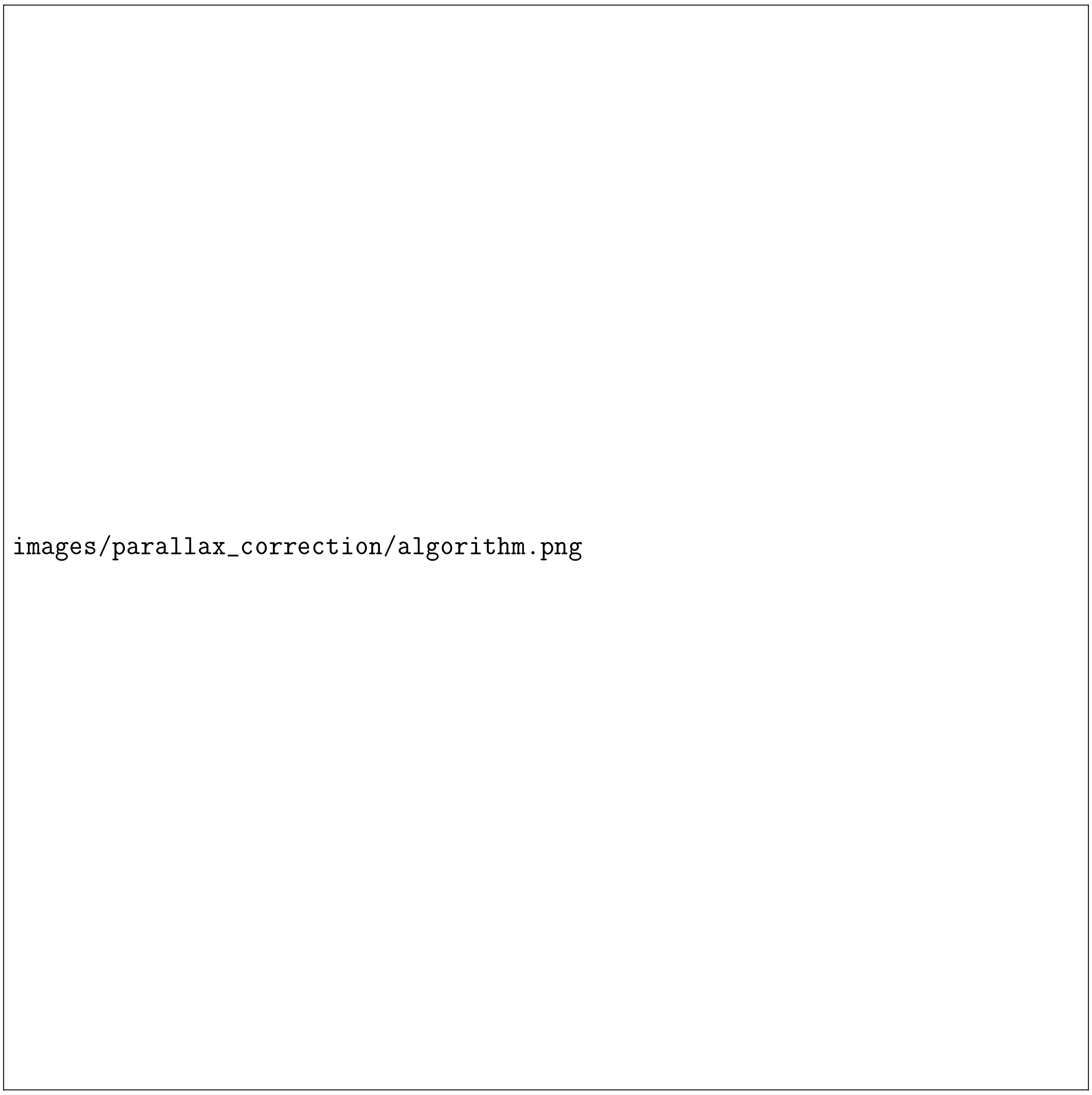


Figure 19: Step by step projection of the disparity map

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix},\ G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4}$$

And we can compute two images, one with the norm of the gradient, the other with the orientation of the gradient.

$$G = \sqrt{G_x^2 + G_y^2} \tag{5}$$

$$\Theta = \arctan 2(G_y, G_x) \tag{6}$$

Finally we constraint Θ between 0°and 90°to avoid as much as possible sign error in the gradient images/parallax_correction/schema4.png

Figure 20: Disparity projection onto infrared image

orientation. We put thresholds on the gradient intensity to remove the noise created by the texture of the image and to avoid loss of dynamic due to high value outliers:

$$\Theta = abs(abs(\Theta - 180) - 90) \tag{7}$$

$$G[G < mean(G)] = 0 \tag{8}$$

$$G[G > 5 * mean(G)] = 5 * mean(G) \tag{9}$$

Using the HSV representation (or HLS colorspace as any cylindrical colorspace should work) we can create a new image where:

- The Hue is the orientation of the gradient Θ computed above
- The Saturation is a uniform layer with all the pixel to the maximum
- The Value (or Lightness) s the norm of the gradient G computed above, normalized between 0 and 255

The following Figure (Figure **??**) shows the result with the example images.

2.8.2 Evaluation indexes

The classical indexes used to compare two images can be used:

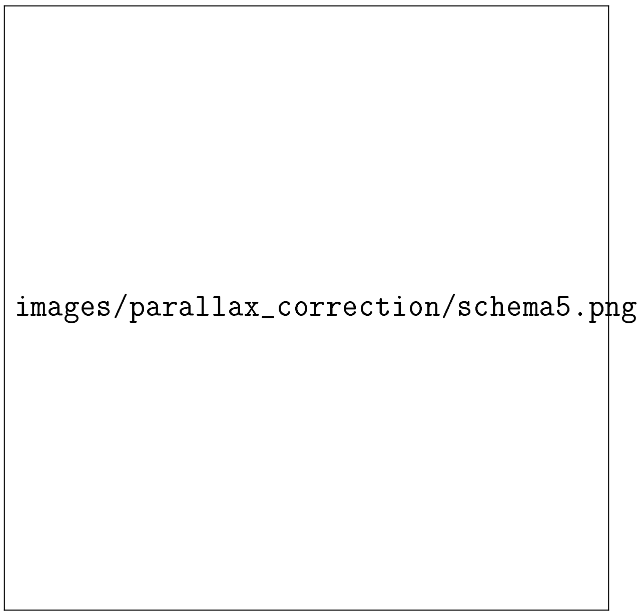


Figure 21: Correction of the infrared parallax

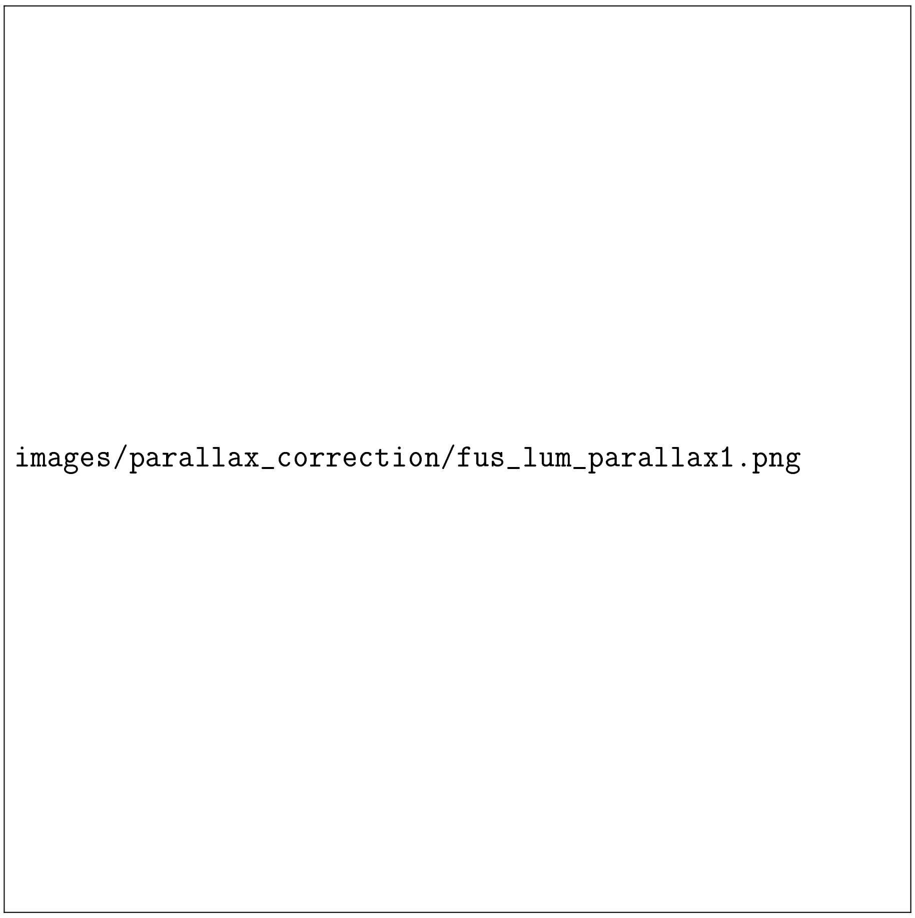
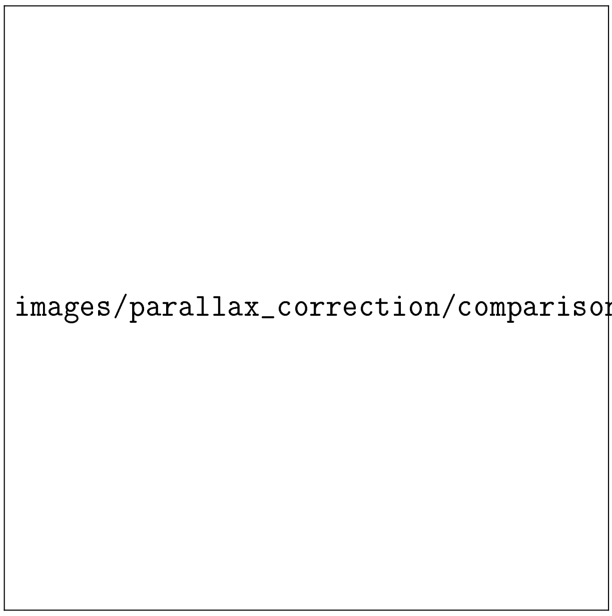


Figure 22: Simple luminance fusion with corrected parallax

- Mean Squared Error (MSE) or Root Mean Squared Error (RMSE):

  $RMSE(A,B) = \sqrt{\frac{1}{N_{pixels}} * \sum(A-B)}$, hold between 0 and 255, the lower the better



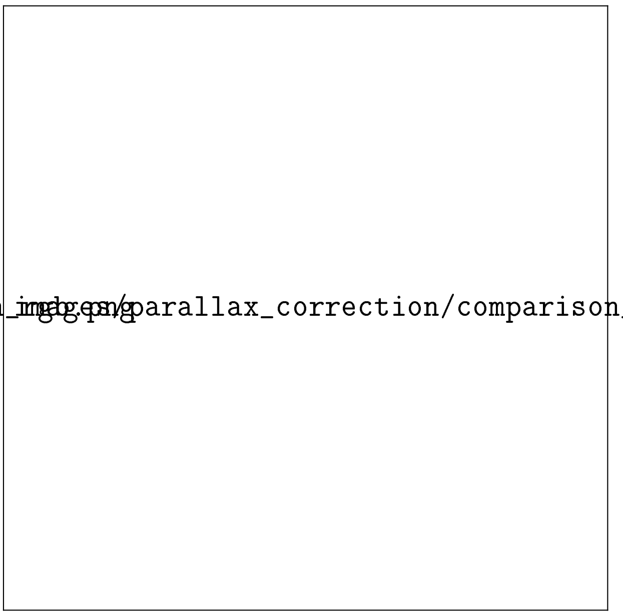


Figure 23: Left: RGB oriented edges, Right: Infrared oriented edges

- Normalized Mutual Information (NMI):

  $NMI(A,B) = \frac{H(A)+H(B)}{H(A,B)}$, Where $H(X) = -\sum xlog(x)$ is the entropy, value between 1 and 2, the higher the better

- Peak Signal Noise Ratio (PSNR):

  $PSNR(A,B) = 20 * \log_{10}(MaxA) - 10 * \log_{10}(MSE(A,B))$, Where MaxA is the maximum value of image A (up to 255 for an 8 bit image)

  Each of the previous indexes compare the two images pixel per pixel, but the next index is much more sensitive to the structure of the input images.

- Structural SIMilarity (SSIM):
  The SSIM Metric is computed all over the image within a sliding window. For each pixel the centered window create two little images A and B from the pixel's surroundings in the RGB image and IR image.

  $SSIM(A,B) = \frac{(2\mu_A\mu_B+c_1)(2\sigma_A\sigma_B+c_2)(cov_{AB}+c_3)}{(\mu_A+\mu_B+c_1)(\sigma_A+\sigma_B+c_2)(\sigma_A\sigma_B+c_3)}$, With:

  - μ_A the mean of A
  - μ_B the mean of B
  - σ_A the variance of A
  - σ_B the variance of B
  - cov_{xy} the covariance of A and B
  - $c_1 = (k_1L)$, $c_2 = (k_2L)$ and $c_3 = \frac{c_2}{2}$
  - L the maximum value that can take a pixel (255 for a 8 bits image)
  - $k_1 = 0.01$ and $k_2 = 0.03$ as default values

- To evaluate the quality of the parallax correction I created a new index based on the edges correlation between the two images. Here A and B are two edges images computed using a Sobel filter or the newly created Gradient image (See chapter **??**).

  Normalized Edges Correlation (NEC):

  $NEC(A,B) = \frac{\sum_{x,y} A(x,y).B(x,y)}{\sqrt{\sum_{x,y} A(x,y).\sum_{x,y} B(x,y)}}$, with values between 0 and 1, the higher the better.

We can use those indexes to assess of the quality of the correction. The most relevant ones shall be the SSIM and the NEC, the other are given for information purpose only.

2.9 Reconstruction quality assessment

In order to evaluate the quality of the parallax correction, I selected randomly 100 images on each daylight and nighttime.
The evaluation procedure is the following:

- Each sample is a group of 3 images of the same frame: the original infrared aligned, the parallax corrected infrared and the RGB image. Those 3 images are transformed using the method described in chapter **??**.
- The indexes are computed twice with as input: Original infrared / RGB and modified infrared / RGB, giving respectively the indexes "ref" and "new".
- The two results are compared computing:
  - The mean : $\mu = \frac{1}{N} * \sum(new - ref)/ref$
  - The minimum and maximum : min = $\min(new - ref)/ref$ or min = $\max(new - ref)/ref$, Depending if the index has to be higher or lower to be better.

Day	Mean	Variance	Minimum	Maximum	Best
SSIM	+0.019 | +9.13%	0.008	+0.005	+0.042	Higher
NEC	+0.058 | +17.6%	0.016	+0.016	+0.104	Higher
NMI	+0.006 | +0.60%	0.002	+0.001	+0.010	Higher
PNSR	+0.411 | +3.51%	0.111	+0.095	+0.767	Higher
RMSE	-3.057 | -4.61%	0.839	-0.717	-5.805	Lower

Table 2: Indexes evolution over 100 Day images

Night	Mean	Variance	Minimum	Maximum	Best
SSIM	+0.023 | +35.3%	0.009	+0.010	+0.063	Higher
NEC	+0.027 | +12.0%	0.010	+0.005	+0.048	Higher
NMI	+0.002 | +0.15%	0.001	+0.000	+0.003	Higher
PNSR	+0.169 | +1.51%	0.065	+0.027	+0.303	Higher
RMSE	-1.361 | -1.93%	0.519	-0.215	-2.484	Lower

Table 3: Indexes evolution over 100 Night images

The two tables presented above (Table **??**, **??**) shows the evolution of the different indexes between before and after mitigating the parallax effect.

As expected the indexes SSIM and NEC are the ones improving the most as they focus on the images structure. Nonetheless the other indexes show an improvement as well as we apply the computation on a gradient based image.
If the indexes show an improvement for every single image samples we test, we have to keep in mind that those corrections are not perfect. Some of the pixels are not well estimated in the disparity and as a result we can see some artefacts after the reconstruction (see Figure **??**). These examples are from nighttime images where the disparity is much more difficult to estimate. More example of reconstruction can be found in the Annexes.

`images/parallax_correction/artefact.png`

Figure 24: Pixel miss-reconstructed due to a wrong disparity

3 Fusion Mask

In the first part we presented a fusion technique base on the luminance. we can develop this method but not using only a homogeneous weighting between the images, but to put the focus on the areas of interest. In the first part the weight mask between the two input images was a constant value of $\alpha = 1/2$. In this section we will try to adjust those mask weights according to the perceptual relevance that the images carry for each pixel. When both source provide little to no information, the RGB image will be prioritized in the balance to keep the most natural looking fused image possible.
Once again we will need some indexes to assess the quality of the fusion. That's a bit more tricky question than before, because the "classical" indexes presented in chapter **??** do not take into account what value is relevant to the final user, but only the similarity between two inputs. Nonetheless they may give a first criterion to know if the information from both sources has been transferred to the new one.

3.1 Example image

For this part about fusion I selected an image of a night scene because most of the daylight images in my 2 minutes run are well defined and objectively do not need additional information. During nighttime there are plenty of thing that the infrared can bring to complement the RGB images: Background buildings or mountains shapes, object in a saturated area due to car's lights or even public lighting.

`images/mask_fusion/example_rgb.png` `images/mask_fusion/example_ir.png`

Figure 25: Images example selection

There are several interesting elements in this example:

- The saturated are due to the car arriving in front of us, the infrared can show the shape of the car where the RGB is just overwhelmed (1).
- The shape of the buidings skyline on the right side of the road is much more detailed and go a lot further on the infrared image (2).

- The shape of the mountain and the electrical wire just above it in the middle of the image is invisible with only the RGB image (3).

images/mask_fusion/example.png

Figure 26: Areas of interest

3.2 Saliency Mask

The idea here is to find in the input images the features that the human's eye will focus on. The design of the mask is split in the three following parts: Saliency of intensity, color and orientation. Those three values will be computed through a Gaussian Scale pyramid (image from "wikipedia", see Figure **??**

For each computed maps in the next part, I tried out two models: one using a mono-scale pyramid but several intervals with larger and larger gaussian kernels, another more classical using several octave of pyramid with only one blurring level. The method I used is described in [?].

3.2.1 Normalization function

Before to fusion the pyramid's different levels a normalization operation will be applied on each levels. The normalization procedure aims to decrease the outlier intensity to keep as much contrast as possible. That can be applied with one or two images in input. The steps of normalization are the following:

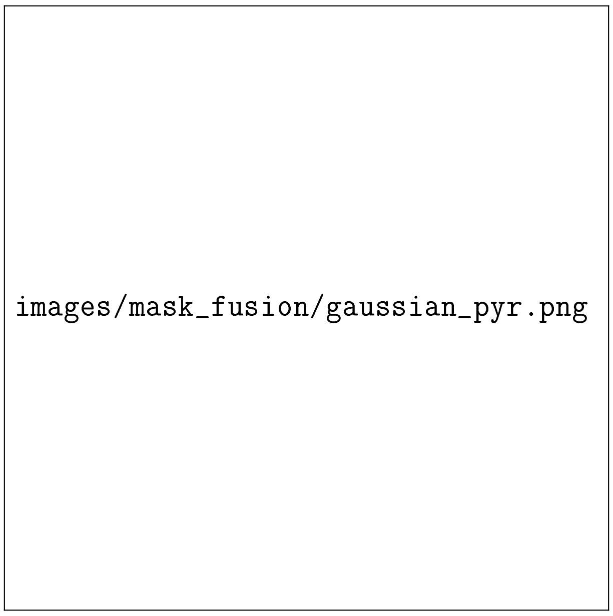


Figure 27: Scale Pyramid construction

Algorithm 2: Normalization procedure

```
Data: $image_1$ and $image_2$(optional)
Result: $image_1$ $normalized$ and $image_2$ $normalized$
$M = Max(image_1,\ image_2)$;
$m = Min(image_1,\ image_2)$;
$\mu = Mean(image_1,\ image_2)$;
for $k \in 1, 2$ do
    $Mask_{weight} = 1 - \frac{|image_k - m|}{M}$;
    $Output_k = Normalize(image_k * Mask_{weight},\ 0,\ 255)$;
end
```

3.2.2 Intensity Saliency

The high luminosity tends to attract our sight much more than the dark area within an image. The first part of the saliency map will put the focus on the luminance variation. Two method but the same idea behind: go down the scale to separate the most prominent features from the details.
For this step we work with the intensity only. For the IR image we can use the grayscale directly, for the RGB we will use the Luminance layer of a LAB decomposition. Then we create 6 maps for each images:

- $I_{RGB}(c, s) = |I_{RGB}(c) \ominus I_{RGB}(s)|$, where $\ominus$ is the pixel-wise difference
- $I_{IR}(c, s) = |I_{IR}(c) \ominus I_{IR}(s)|$
- $I_{RGB}(c, s), I_{IR}(c, s) = Normalize(I_{RGB}(c, s), I_{IR}(c, s))$ using algorithm (**??**)
- c $\in \{2, 3, 4\}, s = c + \delta, \delta \in \{3, 4\}$

For the scale pyramid c and s design the octaves of the pyramid, for the Gaussian interval pyramid, c and s design the σ of the Gaussian kernel. The result of the pyramid inter-scale fusion is presented Figure **??**. Two masks are presented for the Gaussian version and the scale version : RGB and IR.

Figure 28: Saliency - intensity mask

3.2.3 Color Saliency

The saliency of the colors is often very informative as the color can exist on a CMOS sensor only if there is a difference of measurement between the three channel R-G-B. That means there is enough light but not too much on each channels. The saturated areas and the dark areas will produce a very balanced mix of Red-Green-Blue.
This part is of course, only applicable for RGB as it requires 3 layers to be differentiable. The saliency will be maximum using the color pair: green/red and blue/yellow. We will then For each pyramid we create 4 channels from the r-g-b decomposition of the RGB image:

- R = r - (g + b) /2
- G = g - (r + b) /2
- B = b - (r + g) /2
- Y = (r + g)/2 - |r - g|/2 - b

Then using those new channels we can compute 6 maps:

- RG(c,s) = $|(R(c)\text{-}G(c))\ominus(G(s) - R(s))|$

- BY(c,s) = |(B(c)-Y(c))⊖(B(s) - Y(s))|
- RGBY(c,s) = Maximum(RG(c,s), BY(c,s))
- c $\in \{2,3,4\}, s = c + \delta, \delta \in \{3,4\}$

Once we compute the inter-scale fusion of the RGBY images, we get the color saliency mask presented on Figure **??** for both methods.

images/mask_fusion/saliency_mask/saliency_color.png

Figure 29: Saliency - color mask

3.2.4 Orientation Saliency

Once we remove the color and the variation of intensity, the last element catching our attention is the structure of the image. Vertical and horizontal lines, long and clear edges or even patterns as a metal fence or a road sign contour. We will use a selection of 4 Gabor's filters, covering every direction at 45° in absolute value. The filtering will be applied on the luminance layer as in the first part of the algorithm.
Then we compute :

- $O(\Theta)$ with $\Theta \in \{0, 45, 90, 135\}$, for each octave
- $O(c, s, \Theta) = Normalize(|O(c, \Theta) \ominus O(s, \Theta)|)$, using algorithm (**??**)
- c $\in \{2,3,4\}, s = c + \delta, \delta \in \{3,4\}$

After the inter-scale fusion of the pyramid and a last normalization with the algorithm (**??**), using both IR and RGB maps, we get Figure **??**.

images/mask_fusion/saliency_mask/saliency_orientation.png

Figure 30: Saliency - orientation mask

3.2.5 Mask concatenation

Once each of the three masks are computed for the infrared image as well as the RGB image we can at least compute the final saliency mask, using all the pieces of information.
First for each image the three mask are linearly combined using weights provided by the user (usually 1, 1, 1):

- $Wi_{RGB} = Weight_{intensityRGB}$, $Wi_{IR} = Weight_{intensityIR}$
- $Wc = Weight_{color}$
- $Wo_{RGB} = Weight_{orientationRGB}$, $Wo_{IR} = Weight_{orientationIR}$
- $I_{RGB} = IntensityMap_{RGB}$, $I_{IR} = IntensityMap_{IR}$
- $C = ColorMap$
- $O_{RGB} = OrientationMap_{RGB}$, $O_{IR} = OrientationMap_{IR}$

$$Mask_{RGB} = [Wi_{RGB} * I_{RGB} + Wc * C + Wo_{RGB} * O_{RGB}] \times \frac{15}{16} + 16 \tag{10}$$

$$Mask_{IR} = [Wi_{IR} * I_{IR} + Wo_{IR} * O_{IR}] \times \frac{15}{16} + 16 \tag{11}$$

images/mask_fusion/saliency_mask/masks_saliency.png

Figure 31: Saliency - final masks before information concatenation

The results are images with value hold within 16-255 (see Figure **??**).

Some tests could be ran to find the best linear combination of those masks in most of the situations..
In a second time we can compute the weights mask for the RGB image (which is the opposite of the mask for the IR image), following this formula:

$$Mask\ Saliency\ RGB = \frac{Mask_{RGB}}{Mask_{RGB} + Mask_{IR}} \tag{12}$$

The values of the image obtained here are between 0 and 1. To get the mask for the IR image, we just have to take the complementary image to 1. The final saliency masks for this example are shown Figure **??**.

The result of fusion that we get using these masks will be presented at the end of the chapter along with some metrics to evaluate it.

3.3 SSIM Mask

The idea here was to create masks from the SSIM images. The SSIM index usually computed as in Tab(**??**) is the mean of the SSIM values computed for each pixel using the formula presented

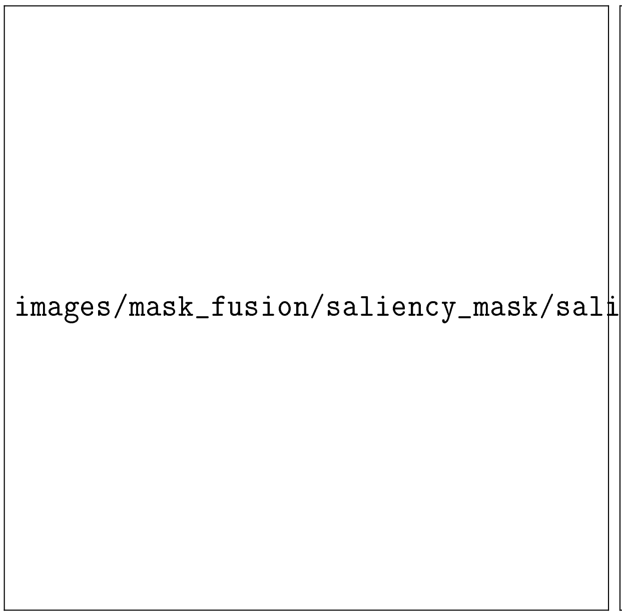


Figure 32: Scale pyramid saliency | Gaussian interval saliency

in chapter **??**. But using *OpenCv*, it's possible to get the full image in return and not only the mean value. The function called "$SSIM_{image}$" in the algorithm thereafter compute the SSIM full image between the two inputs. If only one input is given, the algorithm will compute the SSIM image between the input and an constant image equal to the max pixel value of the input.

3.3.1 SSIM image Algorithm

The algorithm I used to create a mask from the SSIM images of the IR and RGB images is the following:
Now that we have a method to compute a SSIM image, we can apply the algorithm to a scale pyramid.

Algorithm 3: SSIM image using both input channel

```
**Data:** $image_{RGB}$ and $image_{IR}$ and $Weight_{RBG}$ and $WindowSize$
**Result:** $mask_{SSIM}$, between 0 and 1
$WindowSize \leftarrow 2 \times WindowSize + 1$;
$maskDiff \leftarrow SSIM_{image}(image_{RGB}, image_{IR}, WindowSize)$;
$maskRGB \leftarrow SSIM_{image}(image_{RGB}, WindowSize)$;
$maskIR \leftarrow SSIM_{image}(image_{IR}, WindowSize)$;
$maskDiff \leftarrow$ Normalization between 0 and 1, mean shift to 0.5;
$maskRGB \leftarrow$ Normalization between 0 and 1, mean shift to $\frac{Weight_{RBG}}{Weight_{RBG}+1}$;
$maskIR \leftarrow$ Normalization between 0 and 1, mean shift to $\frac{1}{Weight_{RBG}+1}$;
$mask_{SSIM} \leftarrow (maskRGB - maskIR) \times maskDiff$;
$mask_{SSIM} \leftarrow$ Normalization between 0 and 1, mean shift to $\frac{Weight_{RBG}}{Weight_{RBG}+1}$;
**return** $mask_{SSIM}$;
```

3.3.2 SSIM scale pyramid

Algorithm 4: SSIM pyramid

```
**Data:** $image_{RGB}$ and $image_{IR}$ and $Weight_{RBG}$ and $WindowSizeMax$
**Result:** $mask_{SSIM}$, between 0 and 1
$pyrRGB \leftarrow ScalePyramid(image_{RGB}, Octave = [0, 1, 2])$;
$pyrIR \leftarrow ScalePyramid(image_{IR}, Octave = [0, 1, 2])$;
**for** $c \in [0, 1, 2, 3]$ **do**
    **for** $i \in range(WindowSizeMax)$ **do**
        $WinSize \leftarrow i + 1$;
        $pyrSSIM[c][i] \leftarrow SSIM_{image}(pyrRGB[c], pyrIR[c], Weight_{RBG}, WinSize)$;
    **end**
**end**
$mask_{SSIM} \leftarrow$ inter-scale fusion of $pyrSSIM$;
$mask_{SSIM} \leftarrow$ Gaussian filtering;
**return** $mask_{SSIM}$;
```

We can compute a scale pyramid as chapter **??**. We design a pyramid with 3 octaves and as intervals we increase the window size of the $SSIM_{image}$ algorithm within an octave. The $Weight_{RBG}$ variable set the importance of the RGB input in regard to the IR input. The obtain mask trying out this algorithm with our example gives us the map presented Figure **??** on the left. The map shown on the right is the simple application of the algorithm $SSIM_{image}$ without the scale pyramid construction.

3.4 Mask application to fusion

Once we get a mask, the fusion process can be a simple luminance fusion as presented in chapter **??**. Another solution giving usually a smoother result is the use of a Laplacian pyramid fusion. The process is explained on Figure **??**. The rendering of such a process is smoother and tends to have a more "natural" look than just a pondered summation at one scale. However the result of the fusion if often quite similar whatever the method used.

Figure 33: SSIM masks with pyramid | SSIM mask without pyramid

4 Fusion quality assessment

The last section of my work will be dedicated to the assessment of the result following the same metrics as before.
I am not yet satisfied with those metrics though, as they are not really describing the improvement or degradation of an image visual perception from a human point of view. There are only a mere representation of the mathematical proximity of some features: edges, intensity of pixels or similarity of small regions. One future axis of development of my work will be on improving those metrics. Maybe allying some mathematical values and some cognitive evaluation metrics that can be used in the social surveys for example or proper neuro-cognitive psychology experiments.
The assessment is done on the same 200 images than the parallax correction assessment presented in chapter **??**. For each image the following mask are computed for several value of

images/mask_fusion/scaled_fusion.png

Figure 34: Image color and Infrared fusion using a Laplacian pyramid

$Weight_{RBG}$:

- $Mask_{\alpha} = \alpha \times ones(image\ shape)$: mask with a constant value α.
- $Mask_{SSIM}$: mask computed directly with the algorithm **??**.
- $Mask_{SSIM\ Scale}$: mask computed with the algorithm **??**.
- $Mask_{Saliency\ Scale}$: mask computed with the scale pyramid presented Figure **??**.
- $Mask_{Saliency\ Gaussian}$: mask computed with the Gaussian intervals pyramid presented Figure **??**.

A higher value of $Weight_{RBG}$ means more focus will be put on keeping the feature from the RGB source than the infrared. The fusion layer is always the luminance layer in grayscale for the RGB. The two other layers defining the color in LAB representation will remain untouched for the fusion. The assessment will be done only on the luminance layer for the RGB image as it's the only layer modified during this fusion operation. Future improvements could extend the color information on the newly fused areas.
For each metrics presented in the chapter **??** the indexes will be computed as follow:

- $index_{ref} = index(RGB, IR)$
- $index_{RGB} = index(RGB, image_{fusion})$
- $index_{IR} = index(IR, image_{fusion})$
- $index_{FUS} = index(RGB \oplus IR, image_{fusion} \oplus image_{fusion})$, Where $\oplus$ is a horizontal concatenation operator.

4.1 Laplacian fusion, index and visual

For the first analyze of the result we gonna see mathematically the effect of the scaled fusion using the Laplacian Pyramid on the indexes evolution. To do so, each index will be compute over the 200 images (night and day together) for each possible mask with AND without the Laplacian fusion algorithm. The aim here is to determine visually and mathematically if the Laplacian fusion is really interesting for our application case. In the table thereafter are contained the results of fusion with the different indexes computed over the 200 images.
For each index and each fusion method presented only the $index_{FUS}$ because the variable we are interested in here is only the Laplacian fusion process.

Fusion Method/Metrics	SSIM	NEC	NMI	PSNR	RMSE
Alpha + L	🟥 0.768	🟥 0.607	🟥 1.105	🟥 16.5	🟥 38.4
Alpha	🟩 0.775	🟩 0.609	🟩 1.106	🟩 16.9	🟩 36.9
SSIM + L	🟥 0.758	🟩 0.610	🟧 1.094	🟥 16.0	🟥 41.0
SSIM	🟩 0.763	🟥 0.609	🟧 1.094	🟩 16.4	🟩 39.0
SSIM_Scale + L	🟥 0.762	🟥 0.613	🟥 1.098	🟥 16.0	🟥 41.1
SSIM_Scale	🟩 0.770	🟩 0.615	🟩 1.099	🟩 16.4	🟩 38.9
Saliency_Scale + L	🟥 0.761	🟧 0.610	🟧 1.089	🟥 16.2	🟥 39.6
Saliency_Scale	🟩 0.768	🟧 0.610	🟧 1.089	🟩 16.6	🟩 37.9
Saliency_Gaussian + L	🟥 0.759	🟩 0.606	🟧 1.089	🟥 16.3	🟥 39.2
Saliency_Gaussian	🟩 0.765	🟥 0.604	🟧 1.089	🟩 16.7	🟩 37.7
Ref Value RGB/IR	0.438	0.323	1.054	10.9	73.7

Table 4: Comparison of indexes w/wo Laplacian fusion

As we can see with those results, the Laplacian fusion is in most of cases not an improvement in a purely mathematical approach. Indeed for the SSIM, PSNR and RMSE/MSE indexes, the straight fusion is in average over the whole 200 images better the the Laplacian fusion. Of course in some cases we can have the opposite result but the trends favor the weighted fusion without Laplacian pyramid. For the NMI the result is the same, but as the value has a logarithmic evolution, the difference is negligible in most of cases. The NEC is the only index going either way and not only in favor of a direct weighted fusion.
As the Laplacian fusion takes more computing time, we can evaluate visually if it's worth to invest in this computing performance with 2 examples. We will compare the difference of fusion using or not the Laplacian pyramid for the basic Alpha mask with $\alpha = 1/2$.

There is a slight difference on intensity between the two images, but it's nigh not perceptible. The following image shows the absolute grayscale difference between those two images.

images/fusion_eval/Lap_on_comp.png

images/fusion_eval/Lap_off_comp.png

Figure 35: Left : with Laplacian Pyramid, Right : without Laplacian Pyramid

There is indeed a slight difference between those two images, but it seems for now not to be significant enough, at least for this mask of fusion.
For the next part we will continue with the simple fusion.

4.2 Influence of the $Weight_{RBG}$ parameter

The question here is to see if every mask strategy gives the same balance of RGB/IR information in the fusion when we increase/decrease the $Weight_{RBG}$ parameter. The best mask here is the one able to keep the relevant information of both images even with a high unbalanced parameter. The focus will be the 3 points of interest highlighted in the chapter **??**. At first, let's have a look to two selected indexes: SSIM and RMSE. The other indexes are correlated with the MSE, so the information is a bit redundant and to keep a bit of readability, the lesser the better.

As we can observe with the graphs Figure **??**, whatever the mask of fusion used and the value of the parameter $Weight_{RBG}$ the RMSE index of the fusion images is in average always closer from both the IR image and the RGB image. It's without surprise that the index is minimum (so the fusion is mathematically the best) when the $Weight_{RBG}$ parameter is equal to 1. We can do of course the same observation with the SSIM index which is maximum for $Weight_{RBG} = 1$.
The reason why the optimal point is at the 50/50 mix is that the indexes measure a kind of L2 norm between the two images pixel-wise. If we have:

$$f(x) = (x-a)^2 + (x-b)^2, x \in [a,b] \tag{13}$$

The optimal solution of such an equation is the middle point:

images/fusion_eval/Lap_diff_comp.png

Figure 36: Absolute difference with/without Laplacian fusion

$$\begin{aligned} \frac{df}{dx}(x) &= 2(x-a)+2(x-b), x \in [a,b] && (14)\\ \frac{df}{dx}(x) &= 0 && (15)\\ \Leftrightarrow 4x &= 2(a+b) && (16)\\ \Leftrightarrow x &= \frac{a+b}{2} && (17) \end{aligned}$$

That's why it's not really interesting to analyze mathematically -Using this kind of indexes at least- the result of our fusion. As the result has to be the best possible for the user comprehension, an visual analyze makes more sense.
Hereafter are presented the images and details for $Weight_{RBG} \in [0.25, 1, 5]$ for each mask method. We will compare the full result and a zoomed view of the detail presented in chapter **??**.

As a first global analyze of those results Figure **??**, a higher $Weight_{RBG}$ gives of course a way more natural rendering for each mask because we getting closer of the original RGB image. the counterpart is that for all the method the IR information is really weak and nearly not visible for most of them. It will be easier to compare those using closer view of the details.

If we have a look to Figure **??**, the difference is thin but if we look at the light bulb on the left side we can see that more details are kept using the SSIM masks than with the other method. The saliency methods tend to highlight the border of the infrared object a bit more than the other.
On Figure **??** we can see that at least 3 of the methods are performing better than the simple Alpha mask to render the car coming in front on the left. With a high $Weight_{RBG}$, when the image looks the more natural, the details from the IR image are nearly impossible to see with the homogeneous weighting. But the incoming car in the saturated area or the license plate of the car in the middle are still visible even with $Weight_{RBG} = 5$ with the mask SSIM method.

images/fusion_eval/rmse_graph.png

Figure 37: $RMSE_{image} = f(Weight_{RGB}, Fusion\ method), Weight_{RGB} \in [0.1, 20]$

4.3 Conclusion on fusion mask

Create a complex fusion mask paying attention to the details of each image is a demanding task for computing resources. Moreover the result of such mask is not as good as we could expect considering the level of complexity in compare with a simple weighted fusion. It stays interesting for areas of interest as shown above. If we consider the full picture, the SSIM fusion mask tends to highlight more the infrared details we don't have in the RGB image but keeping the same natural look than the Alpha mask fusion.

Using computed mask for a simple luminance approach was an efficient to answer the problematic, but there are still plenty of ways to explore: expending the color on the newly fused area to remove this spectral effect. We could highlight on different color the moving objects towards and away from the car. This involve time series and motion field images. I'll try to develop those subjects during a future thesis by the way of the Neural Network.

images/fusion_eval/ssim_graph.png

Figure 38: $SSIM_{image} = f(Weight_{RGB}, Fusion\ method), Weight_{RGB} \in [0.1, 20]$

5 Conclusion

To conclude, I start with a unaligned dataset, rich of 4 channels. I exploited this advantage to compute the disparity maps frame per frame and correct the parallax effect for each of them. We had finally a dataset simple to use to try out some fusion approaches. The results of the fusion obtained are quite satisfying for the SSIM mask method. I keep in mind though that the nowadays and near future trends for image processing is the extensive use of Neural Network. For the future development of this project during the thesis I will have at least those points to answer:

- Design a network able to correct directly the parallax effect using only two sources RGB and IR, and not a RGB stereo pair.
- Work on color propagation in addition to the luminance fusion
- Set a pack of rules to highlight some objects following several parameters: semantic designation, distance, relative speed, etc.
- Create an index or several indexes to associate mathematical analysis and or proper

images/fusion_eval/full.png

Figure 39: Full result for $Weight_{RBG} \in [0.25, 1, 5]$

neuro-cognitive psychology experiments. Maybe find a transverse working relationship with labs dedicated to cognitive sciences.

With such a subjective subject as "getting the best looking fusion possible" it's important to define the goals of this fusion image. If it's to get as much information as possible in a natural looking image, the kind of fusion I proposed here is quite relevant. But if the application aims is to assist a driver for example by adding some important information to drive safely, we have to be sure to not flood the user with to much input information but to give only the most essential one. That's where the highlighting of such potential danger or obstacles for example could be efficient, especially if it doesn't look natural. A lot of different perspectives of evolution are possible...

images/fusion_eval/detail1.png

Figure 40: Detail view of the building background

References

[1] Amine Chadli, Etienne Balit, Pierre Arquier for Neovision, Guillaume Delubac, Emmanuel Bercierand, and Xavier Brenière for Lynred. Gated Multimodal Fusion for Visible-Thermal Pedestrian Detection, 2020.

[2] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. "Digging Into Self-Supervised Monocular Depth Estimation", 2019.

[3] Heiko Hirschmuller. Stereo Processing by Semiglobal Matching and Mutual Information, 2008.

[4] IDS3D. "Données 3D grâce à la stéréovision".

[5] Computation L. Itti C. Koch and Pasadena CA USA E. Niebur Krieger Mind Brain Institute Johns Hopkins University Baltimore MD USA Neural Systems Program, California

images/fusion_eval/detail21.png

images/fusion_eval/detail22.png

Figure 41: Detail view of a highly saturated area

Institute of Technology. "A model of saliency-based visual attention for rapid scene analysis", November 1998.

[6] French Road Safety Observatory. Bilan de l'année 2021, 2021.

[7] GangWei Xu (Huazhong University of Science and Technology). Attention Concatenation Volume for Accurate and Efficient Stereo Matching [ACVNet], 2021.

[8] European Parliament and Council of the European Union. "General Data Protection Regulation", 2016.

A Color space

Color images are represented in 3D in different possible color-space. There are several categories of color-space representation:
- The first one is based on the human perception of light with different representation as the first colorspace CIE 1931 XYZ or the widely use CIELAB. This color space is often chose for image processing operation the distance between the color in this space in close to the human perception of the color. We can use then the "L" layer of the obtained image which derives from the luminance of the surface and fusion it with the infrared layer.

images/first_approach/lab_space.jpeg

Figure 42: LAB color space representation, image from "wikipedia"

- The second category is using the RGB triptych to decompose the light. Each channels R-G-B describes the chromacity component of a given color when excluding the luminance. The main RGB based representation we will use is the sRGB (standart red, blue, green).

images/first_approach/sRGB.jpg

Figure 43: RGB colorspace representation in the space XYZ, image from "wikipedia"

- The third category is a linear transformation of RGB color-space aiming to reproduce the "lightness" + two "chromas" representations. But as the RGB values are gamma-corrected the separation produces a "luma" and two "chrominances". These are the YCbCr and YUV domains respectively. There are mainly use for video encoding as the human eyes is way more sensitive to lightness than color, the bandwith is mainly encoding the luminance information.

- One last category that we can characterize are the cylindrical transformation of the Cartesian RGB primaries. The color here is intuitively describes as hue-saturation-value (HSV) or hue-saturation-lightness (HSL) or even Lightness-Colorfulness-hue (LCh). The issue with both HSV and HLS is that these approaches do not separate colour into their three value according the human perception of color. Even with a 'V' or 'L' fixed, a change in the saturation or hue will be visible as a perceptual change of lightness.

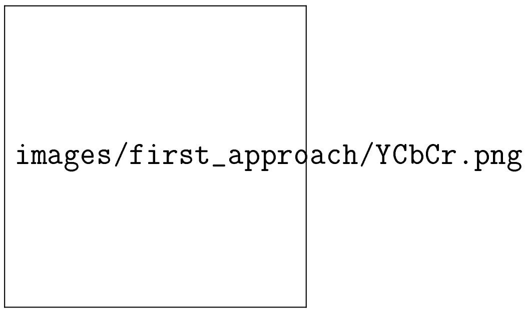


Figure 44: YCbCr colorspace representation for y=50, image from "wikipedia"

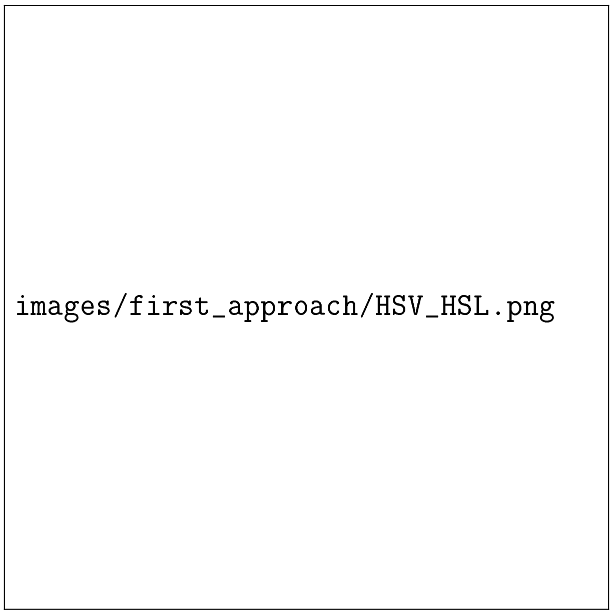


Figure 45: HSV & HSL representation, image from "wikipedia"

A.1 Fusion pixel-wise using a color-map for the infrared

One of the most simple way to fusion two images is to make the pondered sum of it with a constant coefficient all over the images. As the visible image is a 3 channels (Red - Blue - Green) based array, we need to transform the mono-channel IR image in order to obtain those same three channels. An easy way is to apply a color-map defining a color for each level of intensity. This color is coded in a classical three channels array, so we can use the color-mapped IR image to compute the pondered sum over the three channels as in Figure **??**. The applied colormap here is 'inferno' from matplotlib. This kind of colormap is converting a simple luminance in a triptych RGB. The colormap 'inferno' is a common choice to color IR images because it evokes the thermal imagery and it's perceptually uniform colormap aswell, see Figure **??**

This kind of fusion is really simplistic and create a weird looking yellowish image, not the most natural one. Moreover we can see immediately that the images are not aligned one with the other. The closest objects appear quite well fused on the image but the further we go the more the more the parallax duplicate the objects. The first problem that we faced is to realign all those image in order to fusion them.

images/first_approach/fus_cm_ex.png

Figure 46: Inferno colormap fusion

A.2 Fusion pixel-wise on the luminance information of the visible light

This is a second basic approach of fusion. Considering that the luminance information of the infrared is the only information we have, we can use a representation with an explicit luminance channel for the RGB image. Here is a first example of fusion using the L-a-b decomposition with a 50/50 fusion on the layer L.

The result looks a bit more natural to the eye but do not highlight as much as a colormap fusion. It's however one of the best simple approach when the fusion of information is really interesting. The result is nigh the same using the LUV or YCrCb colorspace.

images/first_approach/uniform.png

Figure 47: Inferno colormap uniform perception

B Parallax correction example

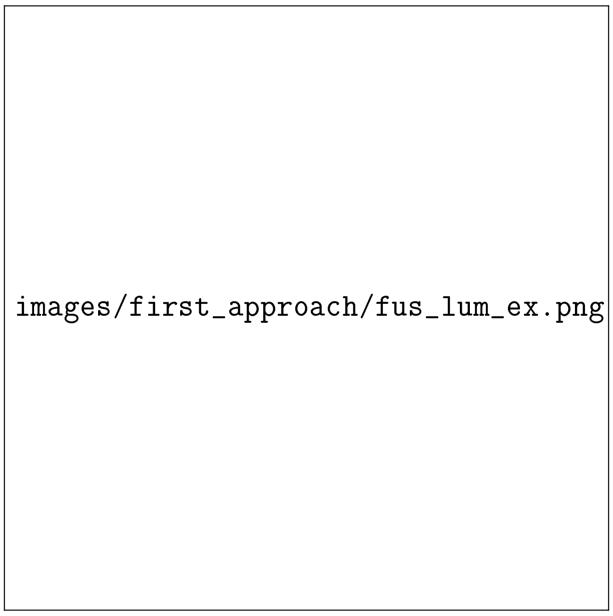
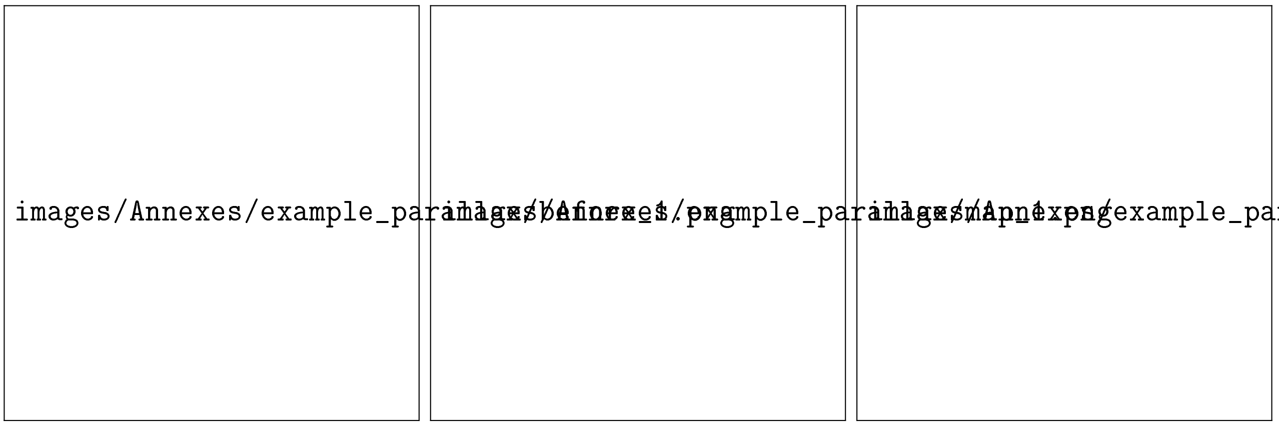


Figure 48: Fusion of the presented scene with a ratio 50/50

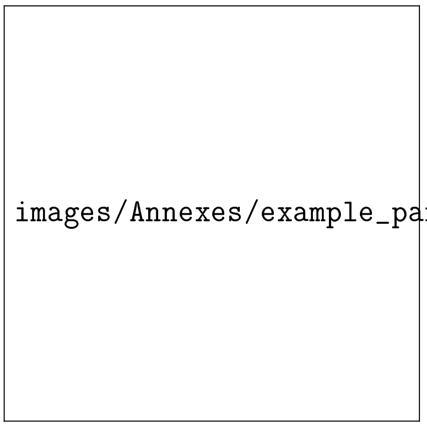


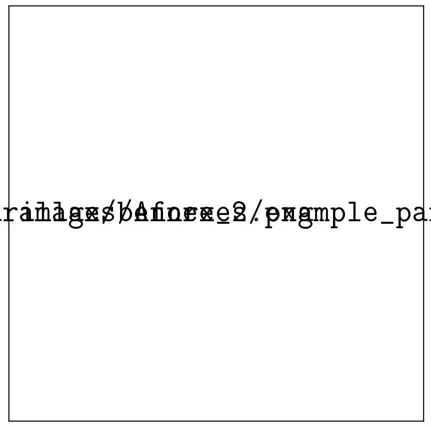
Abstract

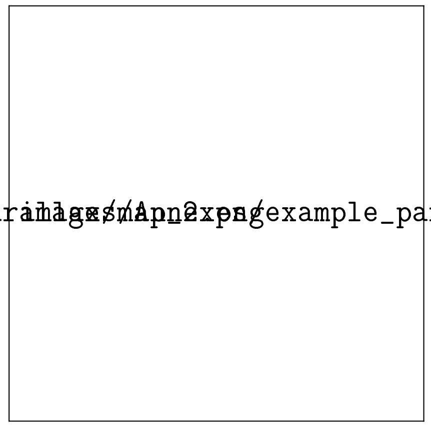
In this paper I propose a complete pipeline to process and fusion a couple of infrared and visible images specialized on driving application.
At first I took advantage of the stereo acquisition of the Lynred's dataset to propose a couple of method to correct the parallax effect. If the cameras are aligned using a checkerboard before acquisition, the spatial baseline between the cameras prevent the images of a stereo pair to be pixel-wise aligned. This step can be avoided using a coaxial optical system with a beam splitter, but this solution is costly.
The next part will be focused more on the quality of the fusion and the composition of the fusion according the objective : give the maximum of information to the user, or highlight the most pressing potential dangers. The main application of this work is the Advanced driver-assistance system (ADAS), but it's opened to other field of course...

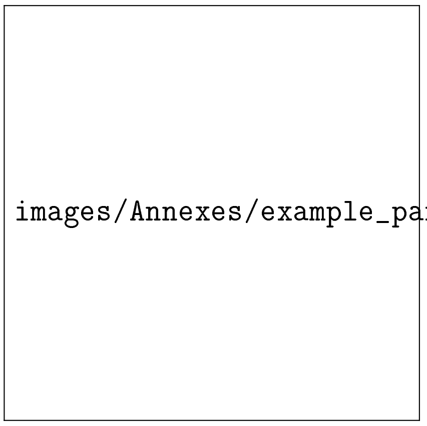
?abstractname?

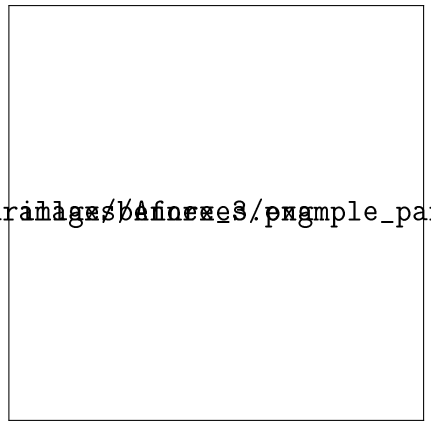
Dans ce rapport je propose un processus de fusion complet de couples d'images visibles - infrarouges avec une attention particulière à l'application automobile.
Tout d'abord j'ai pu utiliser le fait que le dataset enregistré par Lynred comprends deux

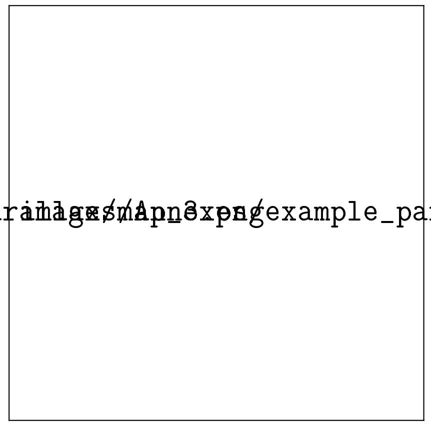


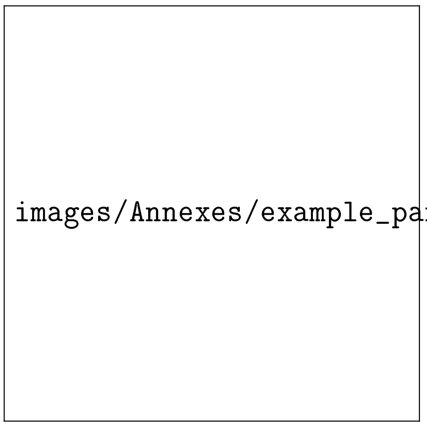


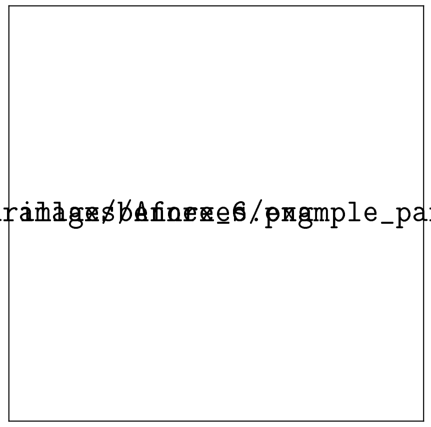


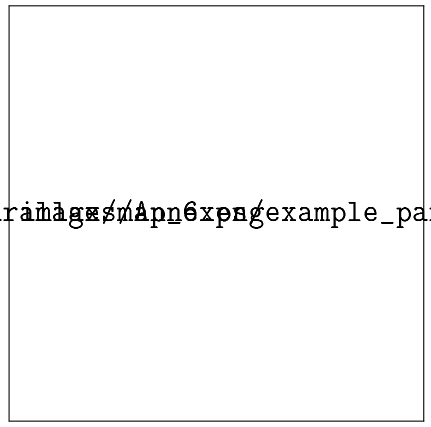


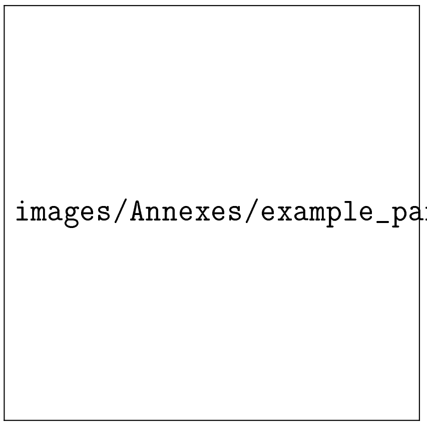


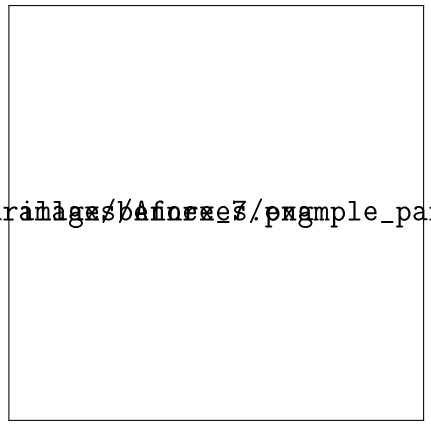


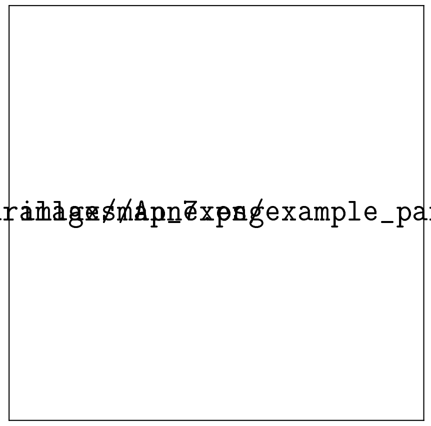












voies visibles et deux voies infrarouges synchronisées, ce qui m'a permis de proposer des

méthodes de correction de la parallaxe de bonne qualité. En effet même si les caméras ont été calibrées en usine grâce à un damier créé pour l'occasion, cela ne change pas le fait que l'écartement des caméras empêche les images d'être fusionnables pixel à pixel. Cette étape pourrait être évitée avec une solution optique telle qu'un séparateur de rayons, mais ce n'était pas le cas ici et le cas échéant le système pourrait être couteux.
La partie suivante se concentre plus sur l'objectif initiale : obtenir la meilleure fusion possible suivant des critères que l'on peut définir en amont : donner le maximum d'information à l'utilisateur ou alors mettre les potentiels dangers en avant. L'application principale de ce travail sont les systèmes avancés d'aide à la conduite (ADAS), mais d'autres champs d'application sont immaginables bien sûr...