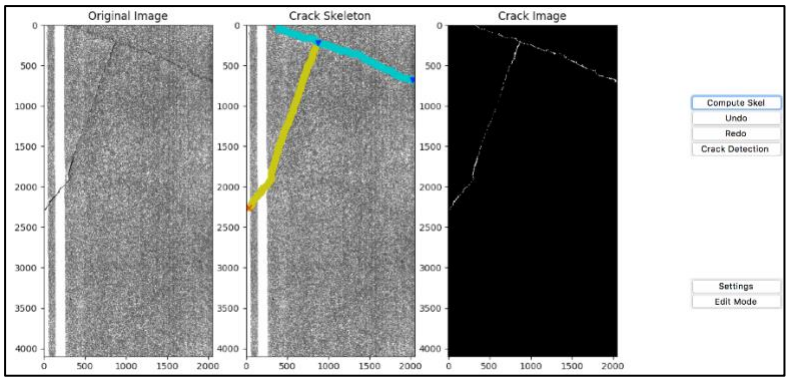
TÉCNICO LISBOA

Image Processing for Detection of Road Pavement Degradation



David Luís Dias Fernandes

Dissertation to obtain a Master Degree in

Electrical and Computer Engineering

Supervisors: Prof. Paulo Luís Serrano Lobato Correia

Prof. Henrique José Monteiro Oliveira

Examination Committee

Chairperson: Prof. José Eduardo Charters Ribeiro da Cunha Sanguino

Supervisor: Prof. Paulo Luís Serrano Lobato Correia

Member of Committee: Prof. José António da Cruz Pinto Gaspar

June 2018

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgment

I would like to thank my Professors Paulo Correia and Henrique Oliveira for all the support that was fundamental for the development and quality of this dissertation. I also would like to thank Joaquim Espada for all his help in many situations occurred during the dissertation. I thank my parents and my grandmother for giving me the opportunity to complete the master course in electrical engineering and my friends as well as my colleagues for always giving me the strength and courage to proceed.

I also would like to thank to Instituto de Telecomunicações for giving me the opportunity to develop my dissertation.

Abstract

To keep a high road surface quality and road safety, an appropriate maintenance policy needs to be enforced as soon as cracks start to appear. Since the traditional way of visually detecting road cracks by a skilled technician is very time consuming, this dissertation presents a semi-automatic solution, therefore increasing the speed and efficiency of road surface pavement analysis and reducing the technician effort and subjectivity of the achieved results.

The proposed system provides a fast and effective unsupervised automatic crack detection. It first applies a fast filtering procedure to identify the most evident crack regions present in the image. The skeleton and its endpoints are identified for each detected crack region. Then, a set of minimal paths among those endpoints are computed. To increase the accuracy of the results a validation procedure is applied, considering the intensities of pixels belonging to the previously computed paths and their likelihood of belonging to the crack. Finally, a GUI was developed, allowing a user to correct the automatic crack detection results, by erasing incorrect detections and adding the missing ones, relying on the same minimal path algorithms to minimize the need for user interactions.

The databased used for testing purposes was acquired by the LRIS (Laser Road Imaging System) and it was divided into two subsets, one that contains easier crack examples and another with more challenging crack images. The automatic results produced by the proposed system were compared with a manually created ground-truth. The results achieved show high recall values for both databases and a high precision value for the first dataset. A comparison with a state of the art shows that the proposed semi-automatic solution can compete with the best results reported in the literature.

To further improve crack detection results a possibility is to consider the usage of new sensors. In this dissertation an experimental automatic crack detection system using a light field imaging sensor, notably the Lytro Illum camera, instead of a conventional 2D camera, is also explored. Light field images capture the light rays originating from different directions, thus providing a richer representation of the observed scene. The developed system explores the disparity information which can be computed from the light field, to obtain information about cracks observable in the pavement images. Only simple image processing techniques were considered, to show the potential of using this type of sensors for crack detection. Encouraging experimental crack detection results are presented based on a set of road pavement light field images captured over different pavement surface textures. A performance comparison with a state-of-the-art 2D image crack detection system is included, confirming the potential of using this type of sensors

Keywords: Crack detection, minimal path, light field, unsupervised, image processing.

Resumo

Para manter a qualidade e a segurança das estradas, é necessário aplicar uma política de manutenção apropriada, assim que fendas começarem a aparecer. Uma solução semi-automática é proposta nesta dissertação, pois o modo tradicional de detectar fendas por um técnico é muito demorado, aumentando assim a rapidez e a eficiência da análise do pavimento rodoviário e reduzindo o esforço feito pelo técnico tal como a subjetividade dos resultados.
O sistema proposto começa por proporcionar uma detecção automática de fendas não supervisionada rápida e eficaz através um processo de filtragem para identificar as regiões de fendas mais evidentes na imagem. O esqueleto e respetivos pontos finais de cada região identificada são obtidos usando um método de esqueleto. Em seguida, um conjunto de caminhos mínimos entre esses pontos finais é calculado, usando um algoritmo de caminho mínimo. Para aumentar a precisão dos resultados, é aplicado um procedimento de validação, considerando as intensidades de pixels pertencentes aos caminhos previamente computados e a probabilidade de pertencerem à fissura. Finalmente, uma GUI desenvolvida permite que um usuário apague segmentos de caminhos incorretos e adicione novos, usando os mesmos algoritmos de caminho mínimo para minimizar a necessidade de interações do usuário.
O dataset usado para fins de teste foi adquirido pelo sistema LRIS e foi dividido em dois conjuntos de imagens, um que contém exemplos de fendas mais fáceis e outro com imagens de fendas mais desafiantes. Os resultados automáticos do sistema proposto foram comparados com o respectivo ground truth. Os resultados obtidos mostram altos valores de recall para os dois conjuntos de imagens e um alto valor de precisão para o primeiro conjunto. Uma comparação com um sistema da literatura mostra que esta solução semi-automática pode competir com os melhores resultados relatados na mesma.

Um sistema experimental de detecção de fendas que utiliza um sensor de imagem de light field, nomeadamente a câmera Lytro Illum, em vez de uma câmera convencional 2D, é também apresentado nesta dissertação. As imagens light field captam raios de luz provenientes de diferentes direções, proporcionando uma representação mais rica da cena observada. Este sistema explora a informação de disparidade que pode ser obtida neste tipo de imagens, para obter a detecção de fendas observáveis no pavimento. Apenas técnicas simples de processamento de imagens foram consideradas, para demonstrar o potencial de utilização deste tipo de sensores para detecção de fendas. Obtiveram-se bons resultados experimentais de detecção usando um conjunto de imagens light field do pavimento rodoviário capturadas sobre diferentes texturas do pavimento. Uma comparação de desempenho com um sistema de detecção de fendas de imagem 2D de última geração está incluída, confirmando o potencial de uso deste tipo de sensores.

Palavras chave: detecção de fendas, caminho mínimos, light field, não supervisionado, processamento de imagem.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BPNN – Back Propagation Neural Network

FFA – Free Form Anisotropy

GaMM – Gauss Markovian modeling

GEV – Generalized Extreme Value

GUI – Graphical User Interface

JAE – Junta Autónoma das Estradas

KOMPS – Kernel Oriented Minimal Path Selection

KOMPS-MR – Kernel Oriented Minimal Path Selection with Manual Refinement

KNN – K-Nearest Neighbor

LFCD – Light Field Crack Detection

MPS – Minimal Path Selection

NN – Neural Network

SVM – Support Vector Machine

TRIMM – Tomorrows Road Infrastructure Monitoring and Management

LRIS – Laser Road Imaging System

1 Introduction

This chapter presents the motivation for addressing the subject of automatic crack detection, the objectives and the contributions of this dissertation. Also the structure of the text is presented.

1.1 Motivation

Roads play an important role in the world development and road networks allow connecting many different places, not just within cities, but also between different cities and countries, becoming the dominant mode of transportation throughout the world nowadays, supporting the mobility of people, goods and merchandises. As a consequence of their constant usage, road pavement surfaces are subjected to a continuous degradation. If an appropriate maintenance policy is not applied, the quality of the pavement surface will degrade, thus compromising road safety.

Road pavement surfaces are often composed of asphalt, although there may be other types, notably based on concrete materials and also presenting different texture characteristics. For instance, the granulation size and grey level can vary drastically from one pavement type to another. Also, the texture of first layer of pavement surface can change, as illustrated in Figure 1.

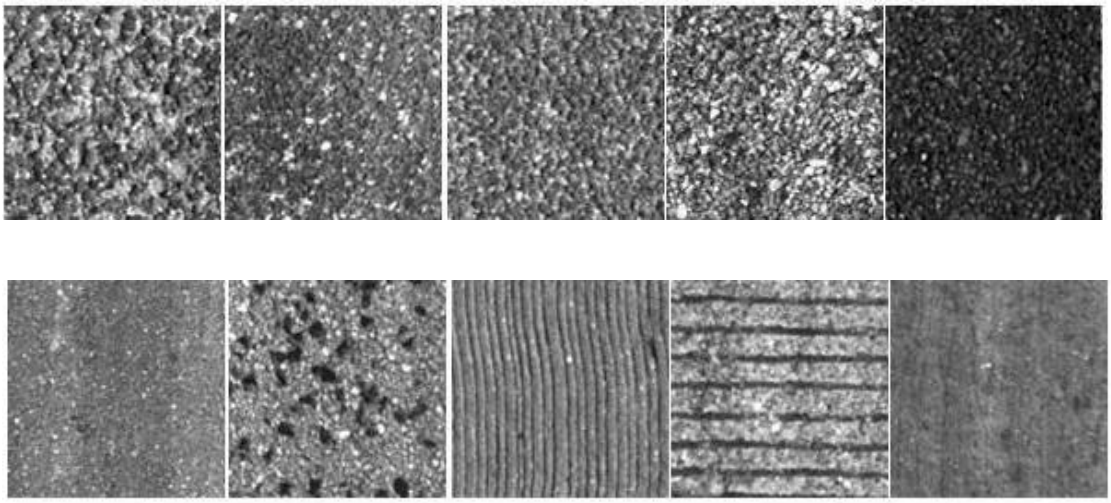


Figure 1 – Example of ten different types of pavement surface made of asphalt (Top image) and concrete materials (Bottom image)[1].

There can be several types of degradations in road pavement surfaces, with the most common corresponding to cracks [2]. A crack in pavement surface is mostly a thin and long road distress, characterized by its dark visual appearance. There are various types of cracks, with longitudinal, transversal and alligator types being the most commonly found in road pavement surfaces, as stated in the National Distress Catalog of Road flexible Pavements (see [3]). Additionally, the interconnections of several cracks, which divide the road pavement surface up into rectangular pieces, may lead to the appearance of block cracking, with these type of cracks exhibiting a rectangular pattern (see [4]).

Figure 2 – Image of a longitudinal road crack.

Whenever cracks start to show on road pavement surfaces, it is an indicator that the quality of the pavement is degrading and maintenance is needed. The solution usually involves a qualified technician traveling along the road, collecting images and recording the conditions of the pavement. After taking images of the pavement surface during manual surveying, qualified technicians must analyze each of them to determine the existence of cracks and classify their type. This process is time-consuming and requires a lot of effort to analyze the complete set of acquired images. In addition, it may happen that two inspectors fail to reach the same decision when classifying a crack.

Several automatic crack detection solutions have already been developed, mainly to overcome the time spent during the manual analysis of the images and to reduce also the subjectivity of the human classification of cracks into types. After the collection of data about the pavement distresses found by these systems, maintenance measures could be taken, thus preventing an even further degradation of the pavement and keeping the road in adequate quality and performance.

Most of the developed solutions for capturing of road pavement surface images use line-scanning CCD cameras to collect pavement images. However, the captured 2D optic data (images) have limitations in representing cracks on the pavement, mainly due to following: (i) the pixel intensities are highly affected by the exposure light direction; (ii) artifacts like shadows and oil stains possess similar intensity as pixels belonging to cracks.

To deal with these limitations, the usage of another imaging sensor type can be helpful. As an example, a 3D laser imaging system [5] or a lenslet camera [6] can give more information that may enhance crack detection.

1.2 Objectives

The main objective of this dissertation is to develop a semi-automatic system capable of detecting cracks from previously captured road pavement images. An objective evaluation methodology with known metrics is adopted, to provide a quantitative evaluation. The proposed solution should consider using minimal path algorithms to detect cracks. To reach this goal several techniques must be developed:

- A fast and effective unsupervised automatic crack detection methodology that quickly provides an initial automatic detection result to be further improved.
- A skeleton computation method to be applied to each detected crack region, and a method to automatically select the skeleton endpoints.
- A method based on a minimal path algorithm to compute the paths between the identified endpoints. To ensure the accuracy of results, a correction based on the intensity of pixels belonging to previously computed paths and their likelihood of belonging to the crack should be considered.
- A GUI to allow the refinement of the automatic results, allowing the user to have the final word. The GUI will allow erasing incorrectly computed paths and adding new ones, relying on the same minimal path algorithms to minimize the need for user interactions. Another usage of this GUI will be to quickly generate a pixel based pseudo ground truth image.

A known dataset should be used to test the system and tune its parameters. The performance of the system should be compared against the state of art, including methods such as MPS [7], to validate the results.

Another objective of this dissertation is to test a new type of sensor for the detection of cracks. An automatic system using a light field sensor, namely the lenslet camera Lytro Illum, will be developed to evaluate the potential of this new sensor for crack detection.
To take advantage of this kind of sensor the following steps are foreseen:

- Create a small light field image dataset.
- Decode the light field images into a multi-view array of 2D images.
- Develop a simple pre-processing technique to enhance crack features.
- Develop a technique that computes the disparity information
- Develop a simple preliminary crack detection technique to provide a first detection.
- Develop a post-processing method to enhance the detection results.

The created light field dataset will be used to test the system and tune its parameters. Comparison against state of art 2D methods, such as Crack-IT [8] will be considered to validate the results.

1.3 Contributions

In this dissertation, a novel unsupervised semi-automatic crack detection system is proposed. The automatic part of the system consists of four steps:

- (i) A fast initial segmentation method using a designed crack detection kernel is first applied to retrieve the most salient regions of the cracks present in the image;
- (ii) The retrieved crack regions are processed using a procedure that identifies the skeleton and respective endpoints;
- (iii) A set of minimal paths among those endpoints are computed, using a bidirectional version of the A^* algorithm, using a novel heuristic that allows reducing the computational time without compromising the quality of the results. To increase the accuracy of the

results a further a validation procedure is applied to each computed path, based on statistical information about intensities of pixels belonging to the previously computed paths;

- (iv) The remaining paths and identified skeleton at step (ii) are merged together forming the complete crack skeleton. The crack width is estimated using an existent method that adds pixels to the formed crack skeleton, depending on their position and photometry information.

Finally, the manual part consists of using a GUI to refine of the automatic results. The GUI allows erasing incorrectly included paths and adding new ones, relying on the same minimal path algorithms to minimize the need for user interactions. All the modules of the proposed system are unsupervised, although some parameters may be manually adjusted, according to the image acquisition system considered.

This dissertation also proposes a new crack detection strategy, by exploring light field images captured with a Lytro Illum, to improve the automatic detection of cracks in images of road pavement surface, notably by exploring the disparity between the different viewpoints available from the captured imagery. Obtained results favorably compare with those results obtained by the analysis of conventional 2D image[9].

1.4 Thesis Outline

This dissertation has the following structure. Chapter 1 motivates the problem and describes the main goal and structure of this dissertation. Chapter 2 presents a review of the most important techniques in the literature. A block diagram illustrating the traditional steps of an automatic solution is presented, to clarify the type of techniques related to each of the automatic crack detection stages. Chapter 3 presents the proposed semi-automatic system, it details its architecture, the assessment protocol that was used and provides the performance assessment results. Chapter 4 introduces an experimental automatic crack detection system and its performance results that relies on light field sensor to collect the pavement images. An introduction to light field is presented at the beginning of the chapter. Conclusions and future work are addressed in Chapter 5.

.

2 Literature Review

This chapter presents a literature review of some of the road pavement surface crack detection techniques based on image processing techniques, as well as an introduction to the light fields topic. The first four sections present an introduction about the road surface crack detection subject and addresses the most important methods in the literature using image processing techniques including a similar approach to one of the proposed systems in this dissertation. The fifth section addresses the existing methods on ground truth creation.

2.1 Introduction

Data collected about pavement surface degradations helps road managers deciding for the best road maintenance policies. The drawbacks of traditional human visual inspection while driving, a task usually considered as time-consuming, labor-intensive, potentially dangerous and prone to subjectivity, have been led to the development of automatic high-speed image capture systems suitable for monitoring extensive road networks at traffic speeds. These high-speed road pavement surveying systems typically employ line scan cameras, such as the Laser Road Imaging Systems (LRIS – see Figure 3) [10], allowing the capture of high-resolution road pavement surface images, while traveling at speeds up to 100 km/h.

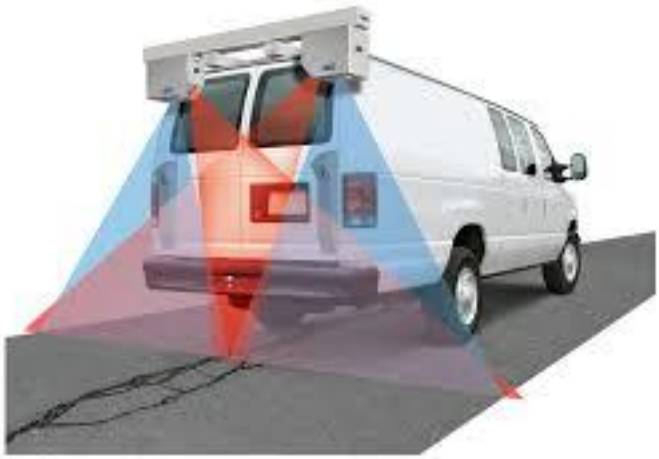


Figure 3 – LRIS system [10].

Once a set of pavement images is available, they need to be analyzed to detect pavement surface degradations. Automatic systems aiming the identification of several types of distresses are being developed by researchers all over the world. Nowadays, the automatic detection of cracks is still far from being considered a solved problem, due to the complexity resulting from having to deal with different types of pavement materials and textures, or the irregular shapes of cracks, whose width typically changes along the crack development, along with the presence of artifacts on the pavement surface with similar visual appearance, such as oil stains, tire marks or even shadows.

An important aspect when developing automatic crack detection systems is how to evaluate the results obtained. In the literature, researchers typically compare detection results against a ground truth. This ground truth is a reference binarized image at a suitable image scale, i.e., the pixel (pixel based evaluation) or any larger gridding (block based evaluation) often created by a human operator (road technician), who manually classifies image pixels, or blocks of pixels, as containing cracks or not. Such manual ground truth creation is a very time-consuming task. Moreover, if a considerable number of images are to be analyzed, different road technicians will create slightly different datasets, because this manual classification is a subjective process by nature.

Figure 4 – Architecture of a general automatic road crack detection system.

The architecture of a general automatic pavement surface crack detection systems is presented in Figure 4 and it usually includes four main steps:

- **Pre-processing:** After image acquisition, the collected images are pre-processed to remove noise and enhance crack features;
- **Feature Extration**: In this step, some chosen crack features are extracted from the image to serve as input for a crack detection method;
- **Crack Detection:** Crack detection will notice if it is present a road crack in the processed image.

Some papers only deal with the detection of cracks in each image [8], while others also consider crack classification [11], which is the process of classifying the detected cracks into types, considering some of their characteristics.

Each step of this system architecture is explained in more detail in the following subsections.

2.2 Pre-processing

Road pavement images are not always captured under the same illumination conditions (day/night), (sun/cloud). In addition to that, the images can contain unwanted objects like random textures, non-uniform illumination and irregularities in the pavement surface. In the acquisition process the image can also be corrupted by random noise, which can affect the detection of distresses in road pavement surface. Other artifacts appearing in road pavement images may interfere with crack detection procedure, like shadows caused by trees or cars, tire marks, oil stains, water, among others.

The objective of the pre-processing step usually includes the removal, as much as possible, the noise present on pavement images, while keeping the ability to detect the existing distresses along the next steps. An effective pre-processing process is vital for obtaining good results and speeding up the following steps on the crack detection procedure. The challenging part in this

first step is to quantify the changes or to enhance the images. Almost all the methods are empirical and need an simulation procedure to know how to get the best results. Some of the most common approaches to this pre-process step in the field of automatic crack detection are point operations, which is based on the transformation $v = f(u)$ by mapping any point in the image into a grey level value. Thresholding, noise clipping, contrast stretching and bit removal are several examples of these transformations [12].

Other methods are based on spatial operations that use local neighborhoods of the input pixels. On these operations, a spatial operator (spatial mask or kernel) is usually applied to perform convolution with the image captured by the imaging system device. Spatial averaging, median filtering, directional smoothing, un-sharp mask and crisping, low pass, band pass and high pass filtering, magnification and interpolation are some examples of spatial operators.

Several pre-processing methods have been proposed in the literature. Oliveira and Correia in [13] used a normalization technique to reduce non-uniform background illumination. For that purpose, a mean value matrix (where each element is the average value of each image block) along with a preliminary classification of crack pixels based on their grey level is applied to equalize the average of the regions preliminarily labeled as non-crack, maintaining the average intensity of the regions labeled as cracks. A region saturation algorithm (top-hat) is also applied in [13] to reduce the influence of white pixels that can lead to standard deviation values like what is observed in blocks of pixels containing cracks, therefore hampering the system performance when using such feature.

Gavilán in [1] proposed an adaptive road crack detection system, where in the pre-processing step an histogram technique together with a sliding window with a determined size was used to smooth the texture, reducing noise and enhancing the crack features. Chambon and Moliard [14] presented the “morph” method which uses image processing techniques like erosion, conditional median filter, conditional mean filter and histogram equalization to remove noise and increase the contrast between the crack and the image background (road pavement free from cracking distress). Anisotropic diffusion filtering to smooth the random image texture is used by Oliveira and Correia in [8].

A shadow-removal technique without affecting the crack pixels is presented in [15]. It consists of four steps. It begins with a grayscale morphological close operation to remove thin cracks, facilitating an accurate shadow region identification without the influence of the cracks. A 2D Gaussian filter is applied to smooth the texture and increase the identification of the shadow area. The third step consists on the creation of N geodesic levels, where each geodetic level contains all the pixels between two gray level values so that each one present a similar number of pixels when compared to the others. After that, the first L low intensity levels will be part of the shadow region, while the remaining levels will be part of the non-shadow region. After identifying the shadow, the last step is to apply the following equation to remove the shadow and get a more uniform image.

$$\lambda = I_B + \alpha I_s \quad (1)$$

$$I'_{ij} = \begin{cases} \alpha I_{ij} + \lambda & if\ (i,j) \in S \\ I_{ij} & if\ (i,j) \in B \end{cases} \quad (2)$$

In equation (1), $\alpha = \frac{\sigma_B}{\sigma_S}$ is the ratio between the intensity standard deviation of the non-shadowed area B (σ_B), and the shadow region S (σ_s) respectively, In equation (2) I_B is the average intensity of area B and I_S is the intensity of the area S.

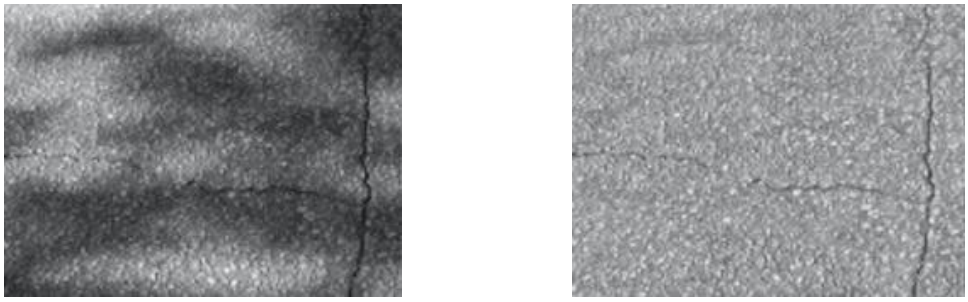


Figure 5 – (Left) Original image. (Right) Pre-Processed image with the method proposed in [15]

An sample result based on the shadow removal technique proposed in [15] is shown in Figure 5. The non-uniform illumination due to shadows is removed, thus obtaining a more uniform image without affecting significantly the crack pixels intensities.

There are several artifacts that can difficult the process of identifying cracks in a road pavement image such as the random texture of the top pavement layer, illumination, image resolution noise, etc. Each of them can influence the choice of the most appropriated technique to be adopted in pre-processing stage of automatic crack detection systems. Reducing the texture, together with the non-uniform illumination, may improve significantly the crack detection performance. From the techniques presented above, mean and median filtering are two simple and fast strategies that can be selected to make the image more uniform. However, sometimes these techniques are not robust enough to deal with the non-uniform illumination and noise presented in images of road pavement surface. Therefore, methods like the one proposed by Zou in [15] can be applied to deal with those artifacts.

2.3 Feature Extraction

The next step after pre-processing the road pavement image is to perform feature extraction using image processing techniques. The objective of the previous step was to enhance the crack region of the image from the background.

Depending on the features quality, the ability to distinguish crack features from non-crack features, the overall system performance can change drastically,

Just like in the pre-processing step there are several image processing techniques in the literature to extract features from the pre-processed image. Some interesting features and those most commonly reported for crack detection are described in this section.

Road pavement images with cracks tend to have several characteristics that can be used to discriminate them from the non-crack ones. Their photometric characteristics (darker pixels in the crack zone), geometric characteristics (continuous structures) and frequency properties (sudden transitions in the image at the frontier of the crack zone) are exploited in crack detection algorithms.
To exploit the above properties, two main approaches to extract features, and therefore to detect cracks can be distinguished in the literature: i) **pixel-based** and ii) **block-based**.

2.3.1 Block Based

Block-based methods split the image into squared blocks, extracting a set of features from each block. Several features have been adopted for this purpose.
Oliveira and Correia in [11] and [16] proposed a block-based approach using non-overlapping $75x75$ image blocks that can be labeled 'crack' or 'non-crack' depending on the mean and standard deviation of the image intensity in each block. A two-dimensional feature space is obtained. A binary classifier labels each block as containing cracks or not in [17], using a density feature, followed by a proximity feature and a fractal dimension feature. Rosa and Correia applied in [16] the features dynamic range, minimum intensity pixel and standard deviation. Average value and standard deviation are the features used in [11].
Mean and standard deviation are two traditional features used in a block-based approach, leading to good results. In [13] the best results are achieved for a parametric learning algorithm with a f-measure of 94.7%. In [16] the best recall presented using the minimum intensity pixel is 95.44% and 95.02% using standard deviation together with the minimum intensity pixel. These three features (mean and standard deviation and minimum intensity pixel) show good results and provide one of the highest results in the literature.
A block-based type approach is less sensitive to high textured pavement surfaces than a pixel base, this approach performs a less coarse analysis of pavement image, but is also less effective against small cracks because they can be classified as texture artifacts.

2.3.2 Pixel Based

The pixel-based methods focus on several road surface crack properties, e.g. photometric and geometric properties
A use of the photometric and geometric characteristics is presented in [18], they propose the usage of the pixel gray level and its neighborhood photometric property mean gray level and local variance.
Photometric and geometric features can detect part of the crack region but they are not very robust to unwanted noise. More complex techniques like the one in [19] use the high frequency

properties of a crack in a pavement image to extract features like the wavelet coefficients. The image is decomposed into different frequency sub-bands by applying a wavelet transform. The distress is transformed into the high amplitude wavelet coefficients representing the details in high frequency sub-bands then the wavelet modulus is calculated. Since cracks have dominant orientations in space and wavelet domain, a radon transform is applied to the modulus of the wavelet to obtain a representation of the cracks in the Radon domain as peaks. The patterns and parameters of the peaks are used in the crack detection and parameter estimation steps. Another usage of the frequency characteristics is presented in [14], it is proposed a second technique called *GaMM (*Gauss Markovian modeling*)* besides *Morph* that uses wavelet coefficients as features.

Oliveira and Correia in [11] is also proposed a pixel-based approach that takes the pre-processed images and performs segmentation based on a dual intensity threshold automatically computed for each image to distinguish crack pixels (*foreground*) and those belonging to the image background. A connected components algorithm is applied to group the crack pixels and identifying a crack candidate.

2.4 Crack Detection

Depending on the level of detail of the image analysis performed, two main classes of methods are identified: **block-based** and **pixel-based**.

2.4.1 Block Based

Block-based methods partition the image into non-overlapping blocks, typically squared blocks, extracting a set of features from each block. Each block is then classified as containing crack pixels or not. Often supervised learning techniques are adopted, considering a training set containing the features extracted from a previously created ground-truth set.

Several machine learning algorithms have been reported in the literature to perform block based crack detection, notably: neural networks (NN) [20], K-Nearest Neighbor (KNN) classifiers [13], Adaboost [16] or Support Vector Machines (SVM) [17].

Neural Networks (NN) is one of the machine learning algorithms used in the literature. The relationship between the input and the output is typically non-linear and depends on many coefficients (weights) which must be learned from the data. Neural Networks (NN) are usually composed of three layers, each one can be composed of several nodes, see Figure 6. The first layer is the input layer and has as many nodes as the number of features being used for crack detection. The second layer is a hidden layer and the third layer is the output layer, typically representing the class or a value attributed to the input. Another characteristic of NN is the ability of the system output converging to the desired output, through a training step with a learning parameter.

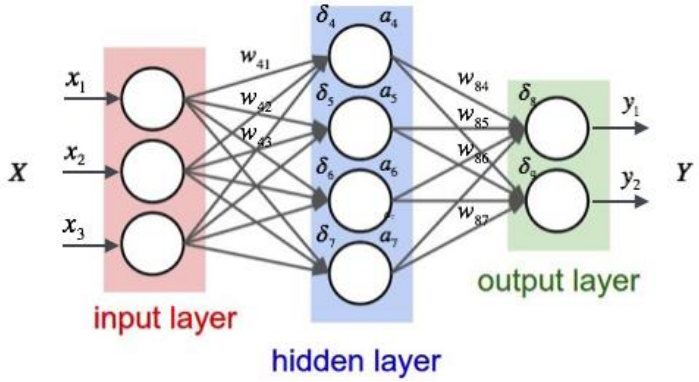


Figure 6 – Neural net architecture

Despite the ability to classify correctly noisy data, this technique also presents some limitations, namely, slow speed of convergence [17] during the learning phase, the number of nodes in the hidden layer and the need of a large amount of good samples to train the system properly [4].
An SVM classifier, just like other machine learning algorithms like NN, is composed by training and testing stages. In the training stage, the selected features are extracted and typically mapped into a higher dimensional space to efficiently separate crack features from non-crack features. Since the ground-truth of the training set is supplied, the features that correspond to cracks and to non-cracks can be determined.

Li et al in [17], compare two machine learning algorithms, namely BPNN (Back Propagation Neural Network) and SVC (Support Vector Classification) based on SVM, to label each image block with one of five possible crack types, respectively longitudinal crack, transversal crack, alligator crack, block crack or no crack.
The training set is composed of 450 images and the testing set of 305 images. The SVM parameters were tuned using a genetic algorithm while the BPNN is composed of 15 nodes in the hidden layer with a learning rate of 0.01. The final results have shown that SVM was more accurate than BPNN for all the training sizes considered and were always faster than BPNN as well. The best results of the SVM and BPNN were for the training set of 450 images, achieving a classification rate of the correct crack type of 78.4% and 69.6%, respectively

The K-nearest neighbor classifier is a non-parametric machine learning algorithm [13] which labels each test sample, taking into account the class of the closest training samples. The most voted class among the classes of the k nearest neighbor dictates the class attributed to the test sample.
Oliveira and Correia apply a 1-KNN (one nearest neighbor) in [6] for crack detection together with an estimated posterior probability density functions, achieving a recall of 94.6%. The database used has images with several crack types namely longitudinal, transversal and miscellaneous but also images without cracks. Two different resolutions are referred, namely 2048x1536 and 1858x1384, being the block size chosen $75x75$ pixels.

Adaboost is a learning algorithm based on boosting that combines weak classifier algorithms to form strong classifier. A single algorithm may classify the objects poorly. But if we combine multiple classifiers with selection of training set at every iteration and assigning right amount of weight in final voting, we can have good accuracy score for overall classifier.

Three types of Adaboost classifiers were compared in [16], namely, Modest Adaboost, Gentle Adaboost and Real Adaboost, being the Modest Adaboost the algorithm chosen since it converges faster and provides a better system overall performance than the other two. To achieve the minimum error, the number of iterations for each training set was 100. Two different image databases were used, being divided each image in blocks of $64x64$ pixels. The best result achieved for the first database was a recall of 95.44% for a training set of 25% of the respective database images. The best results of the second database, that has harder images to analyze, achieved a recall of 85.44% with a crack classification correction of 100% also for a training set of 25% of the respective database images.

2.4.2 Pixel Based

Pixel based methods select which image pixels correspond to cracks. As crack pixels are often the darkest ones, a simple approach would be to select pixels with value below some pre-defined threshold as crack pixels. In practice, such a simple strategy leads to too many false positives. Some solutions additionally consider using post-processing morphological techniques [18].

Alternatively, the use of Markov random fields was introduced in [21], [14] and [22] , to connect local crack regions with their respective neighbors, based on the comparison of their orientation and distance. Another drawback of directly thresholding a pavement surface image, for instance using the Otsu method [23], comes from crack images not presenting a bimodal distribution. Additionally, thresholding methods cannot differentiate between dark image artifacts, like shadows, tire marks or oil spills, and true pavement cracks.

Minimal Path Approach

Considering an image as a graph, each pixel corresponds to a node. In a graph, finding shorter paths is to estimate a path between two nodes, so that it minimizes a cost that is usually the sum of the cost of each edge included in the path. In this dissertation, the term minimal path was used instead of the shortest path because it corresponds better to what is done. The objective is to minimize a cost function that does not consider a (only) geometric distance. For the rest of this dissertation, the term minimal path is the one used.

There are several applications where minimum path algorithms are used such as transport networks, high performance communication, fault tolerant routing [24]. When considering images as a connected pixel graph, minimal path computation has been used for various applications in image processing, such as segmentation or medical imaging [25], extraction of roads in satellite images [26], or extraction of boundaries of objects [27].

There are two important elements that must be chosen when using minimal path algorithms:

- The **cost function** used depends on the weight of each edge. It should be designed so that good solutions have a low cost, while bad solutions have a high cost. In the image processing, most cost functions are based on gray levels or colors. There are other constraints that can be used in cost function such as path regular shape and path length [28]. An example is the detection of blood vessels in medical images because the pattern to be detected has a regular shape and a restriction length [27]. In the case of crack detection, these constraints are not appropriate because cracks have arbitrary shapes and lengths.
- The **optimization** method to find the minimal path. The most popular methods are the Dijkstra algorithm [29], A* [30] (an informed search algorithm based on the Dijkstra algorithm) and the fast walking approach [26].

In the context of road crack detection, the main difficulty for applying minimal path methods is choosing in advance the endpoints of the desired paths. Potentially, a minimal path could be estimated between each pair of pixels of the image. Then the best ones could be selected in a validation step. However, the computational time to solve the amount minimal paths problems using a Dijkstra type algorithm is unfeasible. Therefore, a strategy must be considered to reduce the computational time. Two types of solutions have been proposed in the recent literature such as [7] and [31]. These methods find the best path between two nodes in a weighted graph, using a cost function for estimating the cumulative cost along the path. An example of crack detection using a minimal path approach is shown in Figure 7.

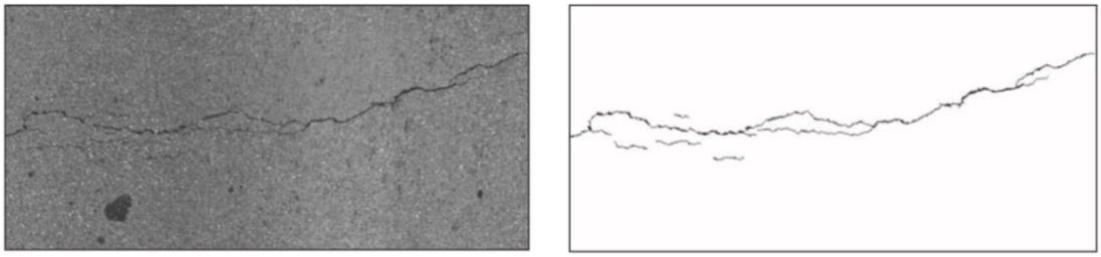


Figure 7 – (Left) Original Image. (Right) Segmented crack detection image using a minimal path method [7].

Cracks are not well-behaved lines and they can appear as complex networks of unknown geometry. To solve these difficulties, minimal path methods typically adopt the following strategy:

- In a first step, some of the darkest image pixels are selected (manually or automatically) as seeds for the crack detection procedure.
- In a second step, pairs of seeds are considered as endpoints and minimal paths are computed between them. The goal is to be able to detect all relevant cracks in the image by applying the procedure to the set of identified seed points.

Two methods following the above approach are the Free Form Anisotropy [31] and the Minimal Path Selection [7]. The first method uses a minimal path algorithm using a 4-connected oriented neighborhood and, with help of 4-oriented graphs, the minimal path is defined from the central pixel towards two unknown extremities at a fixed distance d. The biggest shortcoming of this approach is its dependence on distance d, because there is no automatic way to compute this parameter, but its value must be higher than granulate size although not too high to avoid burdening the calculations. The second method is described with more detail below.

Minimal Path Selection (MPS)

Rabih Amhaz et al [7] proposed a crack detection algorithm called Minimal Path Selection (MPS) that is only based on the photometry and on a minimal assumption that pixels belonging to the cracks form continuous paths of arbitrary shape. The architecture of the MPS algorithm is illustrated in Figure 8.

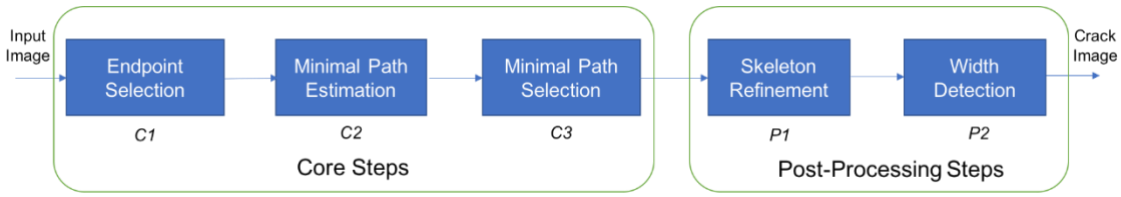


Figure 8 – MPS System architecture

The steps of the MPS algorithm are detailed below:

C1) *Endpoint Selection:* The goal is to select a significant proportion of the endpoints e_i inside the cracks. A first simple step is to partition the image into small square sub-images and to retain the one darkest pixel mi in each of them. Some of the candidates are then removed if their intensity is above an adaptive threshold value.

C2) Minimal Path Estimation p_{ij}*:* Dijkstra algorithm [29] is used to compute the minimal paths p_{ij} between each pair (e_i, e_j) of endpoints using (3) for the minimization. An important aspect is that by using this algorithm and the cost defined in (3), there is no constraint on the shape and the length of the paths.

$$\text{cost}(p_{ij}) = \sum_{i=m}^{j} I(m) \qquad (3)$$

C3) Minimal Path Selection: Among the many computed paths at the previous step, only a small proportion of them are within (or partially within) a crack. Such paths are expected to be made of pixels of darker intensities than the others, so we use again a threshold on the cost path to select the best candidate paths. Since the goal of this proposed step is to select paths with the lowest mean intensities and not the shortest ones, it is important to threshold the following normalized version of cost (4):

$$c(p_{i,j}) = \frac{1}{card(p_{i,j})} \sum_{m=i}^{j} I(m) \quad (4)$$

where $card(p_{i,j})$ is the length of the path (in pixels). The threshold value is chosen using $T_c = \mu_c - k_c\sigma_c$, where μ_c is the mean and σ_c is the standard deviation of the costs. This step makes the result converges towards the skeleton of the crack, a one pixel-wide estimation of the crack with some artifacts.

After applying the core steps some improvements can still be performed. Some obtained paths or some parts of the paths correspond to false detections and the width of the crack must be evaluated. So, the authors propose additional post-processing steps, described below:

P1) Elimination of Artifacts: Some small isolated paths may be present in the estimated crack, due to the texture of the road. A simple thresholding operation is applied on the minimal size of a path.

Some of the paths that constitute the crack skeleton, a frequent situation is that only one endpoint belongs to the crack. The resulting path is then partly inside the crack, and partly outside. The skeleton is divided into linear segments, such that both extremities of each segment are either a junction point, or an extremity of the crack itself. The interest of such a partitioning step is to isolate the spikes and the spurious parts of loops into segments of higher average intensity than that of the "good" segments. A new thresholding operation is then performed on each segment, using the same threshold as in step *C3.*

P2) Width Detection: The proposed width detection procedure consists in absorbing dark pixels neighboring the currently detected crack, according to a threshold test. The adaptive threshold is $= \mu_w + k_w \sigma_w$, where μ_w and σ_w are the mean and the standard deviation of the grey levels within the currently detected crack skeleton. This process is performed iteratively, so that several "layers" of dark pixels may be incorporated.

The results of each above detailed step is illustrated in Figure 9.

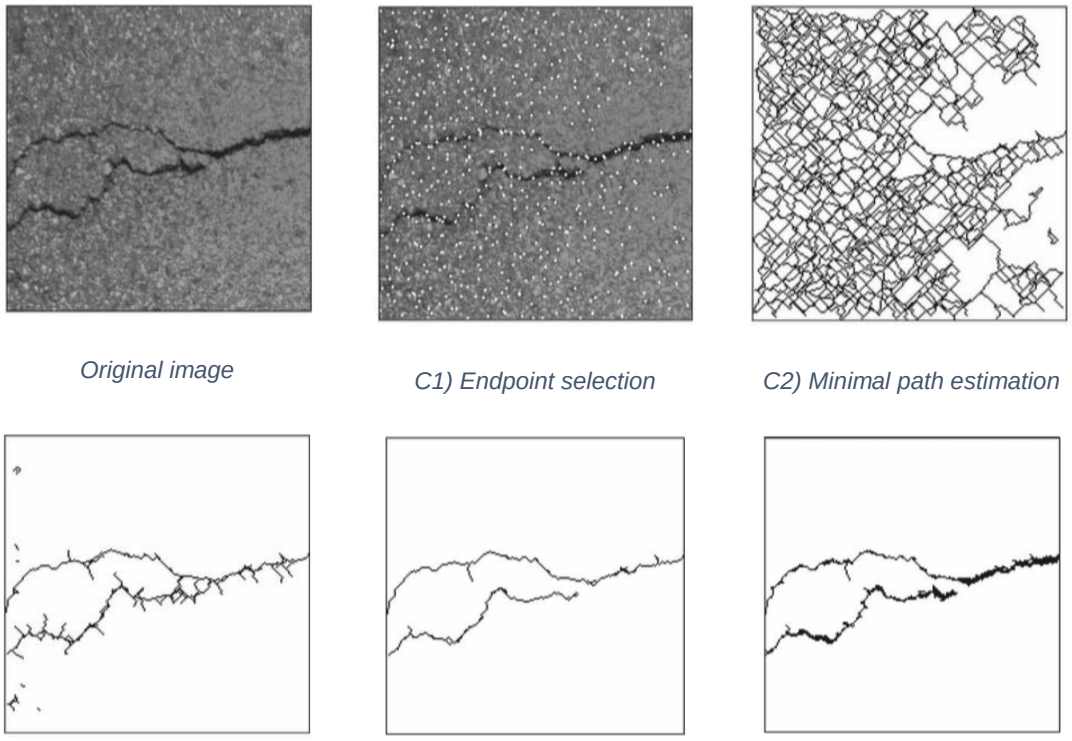


Figure 9 – Illustration of the five steps of the MPS method.

The MPS system is evaluated considering a dataset composed of one simulated image and 269 real images, which can be decomposed into two categories: 68 images with ground truth (details about these reference segmentations are given in the next section) and 201 images without reference segmentation. Initially, 62 images from one sensor in the *Aigle-RN* [32] system were used, and the authors computed a reference segmentation for 38 images from this database, called TRIMM (Tomorrows Road Infrastructure Monitoring and Management) [33], 207 additional images were acquired with four other sensors. From these 207 images, only 30 have a reference segmentation. The complete dataset, containing the 269 images, was divided in 3 different sets:

- **Dataset 1:** containing the 38 images with reference segmentation acquired by Aigle-RN system.
- **Dataset 2:** containing 30 images acquired by the other four systems, with reference segmentations.
- **Dataset 3:** corresponds to the remaining 201 images without reference segmentations

Six different methods including MPS were evaluated using the first two sets of images. Two Markov methods M1 [21] and M2[14] and four minimal path methods GC (Geodesic Contour) [22], FFA (Free Form Anisotropy) [31], MPS0 (early version of MPS) [34].

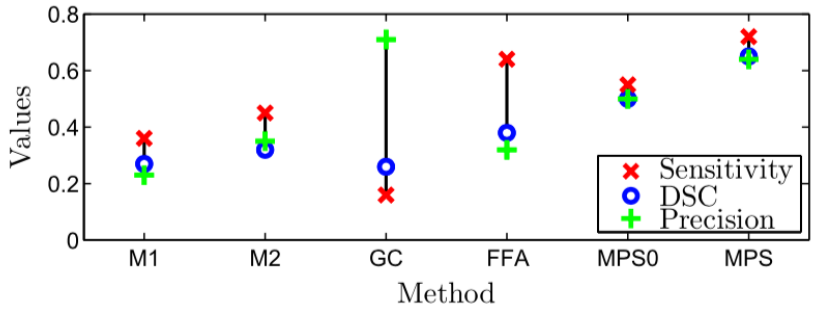


Figure 10 – Averaged values of precision, sensitivity, and DSC for the 38 real images of data set 1.

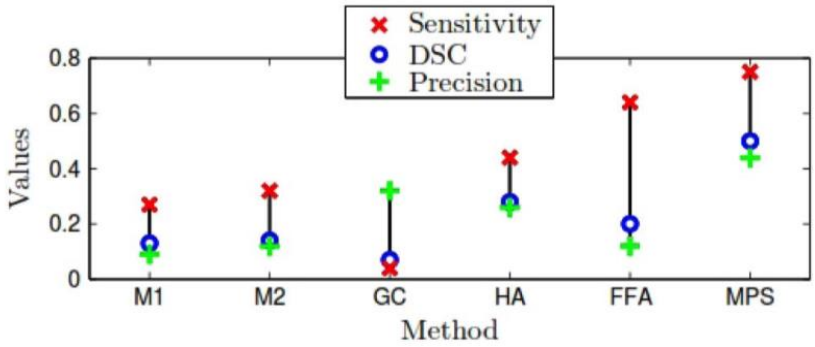
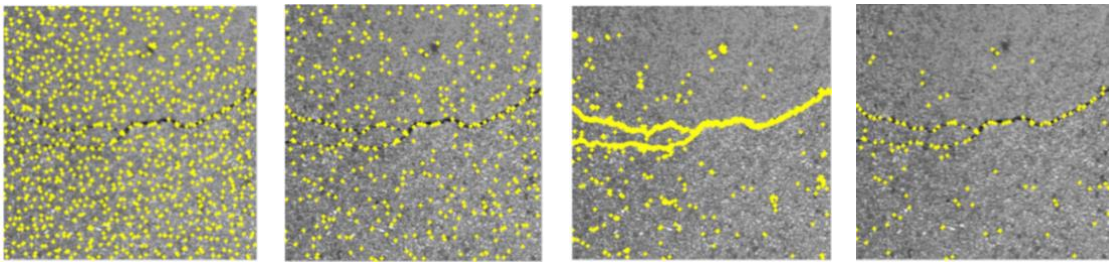


Figure 11 – Averaged values of precision, sensitivity, and DSC for the 30 real images of data set 2.

For dataset 1 the two MPS methods yield by far the best average performance in terms of sensitivity, precision and DSC. Once again, MPS provides a significant improvement over MPS0, meaning that the estimation of thickness has significantly improved the segmentation result.
The overall performance on dataset 2 are under the overall performance of the first dataset. However, the same remarks can be done and the MPS approach still gives the best results.
As a conclusion, the average results obtained on the real image datasets confirm the conclusion drawn on the simulated case: MPS is the most accurate method, followed by MPS0, FFA, M1, M2, and GC in descending order of performance.

Kaddah et al in [35], focused on the improvement of the MPS [7] algorithm (MPSV1 and MPSV2) to reach a robust and efficient crack segmentation within gray level pavement images. Their contribution has focused on the core steps of the MPS algorithm to afford a faster and more reliable crack skeleton. For step C1, two processing techniques have been proposed to further reduce the number of endpoints. The first one is based on the local anisotropy analysis method (FFA). The second processing is defined as the combination of the latter with a global gray level thresholding NSCOT (non-sensitive crack orientation threshold), resulting in a sharper endpoints selection. The second one has been found to greatly reduce the FP (false positive) rate and to make the raw segmentation result getting closer to the ground truth.



a) b) c) d)

Figure 12 – Comparison between the three methods for the endpoints selection at step C1 of MPS: ***a)*** *MPS by local minima selection (LM) within* $P \times P$ *image blocks,* ***b)*** *MPSV1 by local anisotropy analysis (FFA) within P × P image blocks,* ***c*** *after the global thresholding (NSCOT),* ***d)*** *final endpoints selection in MPSV2: common endpoints between (****b****,* ***c****) within P × P image blocks*

For step C2, the computation of the shortest paths has been improved by a new strategy to browse the pavement image. The two latter modifications have greatly reduced the amount of the shortest paths to compute. The parameter setting at step C3 has been then adapted to the new path costs distribution. Finally, the three versions of the MPS algorithm are tested and compared on two image datasets. The average DSC is computed on each dataset as shown in. For the first and second image dataset, the average DSC value gradually increased from 64% (respectively, 70%) for the MPS [7] to 74% (respectively 71%) for the MPSV1 and 75% (respectively 72%) for the MPSV2. The improvement seems to be minored on the second data subset, which suffers from calibration problem and shows stronger image texture [35]. In details, the two improved MPS versions increase the true positive rate (good detection) and reduce the false positive rate (false alarm).

The results shown that both updated MPS versions improve the DSC value compared to the original MPS version. Among the two improved versions, MPSV2 achieves the largest reduction in the computing time without loss of performance.

2.5 Ground Truth Creation

The default solution for the performance evaluation of a crack detection algorithms is comparing the segmented results with a ground truth image. This evaluation can be pixel or block based depending on the scale of the ground truth image. Oliveira and Correia in [36] provide a tool in the Crack-IT image processing toolbox that can be used to manually generate binarized images at the desired image scale as illustrated in Figure 13.

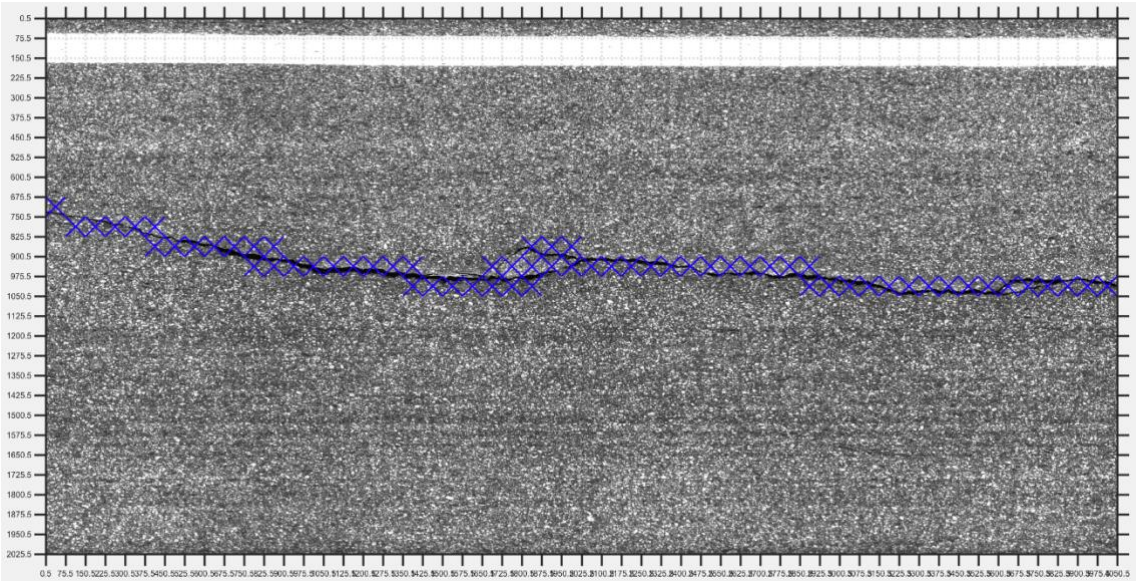


Figure 13 – Example of manually generated block based reference image/ground-truth with [36]

This approach allows a faster creation of ground truth images, but does not replicate exactly the crack shape. To create a perfect ground truth the block size must be at the pixel level. Only this way it is possible to obtain the real representation of the crack.

A solution to quickly create a pixel based level ground truth image is presented in [37], using a minimal path algorithm to find the crack skeleton and a method to find its width. It operates within a user interface that allows manual endpoint selection to then compute the skeleton segments and estimate its width, see Figure 14. A limitation of this solution is that it does not give the user the possibility of changing the results. For instance, if the width estimation fails and overestimate the width, there is no way to adjust it to its real shape.

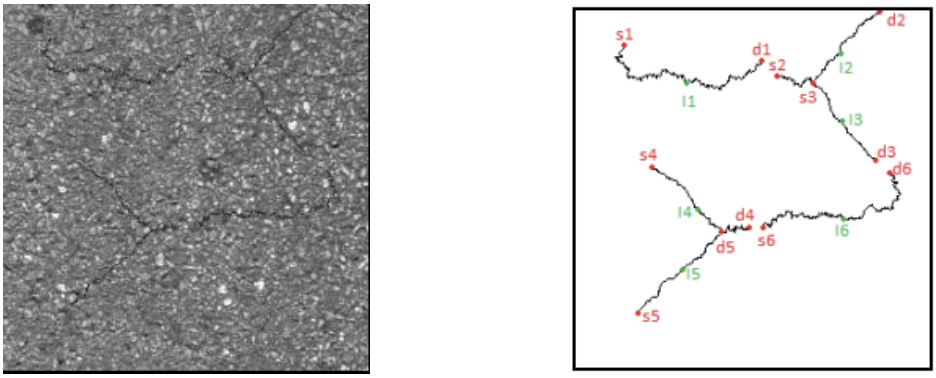


Figure 14 – Example of pseudo ground truth creation from [37]*; (Left) Original image (synthetic image in [7]); (Right) Generated ground truth; in red, the 6 pairs of manually selected sources and destination seed points ,using the developed GUI, for the minimal path search algorithm; in green, the 6 seed points used for initializing the path search by ST Fast-Marching.*

3 Proposed System

In this chapter, a novel semi-automatic crack detection system named **K**ernel **O**riented **M**inimal **P**ath **S**election with **M**anual **R**efinement (KOMPS-MR) is proposed. The first section presents the proposed crack detection algorithm architecture and details its modules. The assessment protocol and datasets used are introduced in the second section. The performance of the used minimal path algorithm and a study about the tuning of the system parameters is presented in the third section as well as an assessment of the system performance including a comparison with the MPS (Minimal Path Selection) proposed in [7].

3.1 System Architecture

The main goal of the KOMPS-MR system, proposed in this dissertation, is to perform crack detection on road surface images. The system relies on a novel automatic and unsupervised crack detection system (KOMPS), whose results can be refined using a graphical user interface (MR). The architecture of the proposed system is shown in Fig. 15.

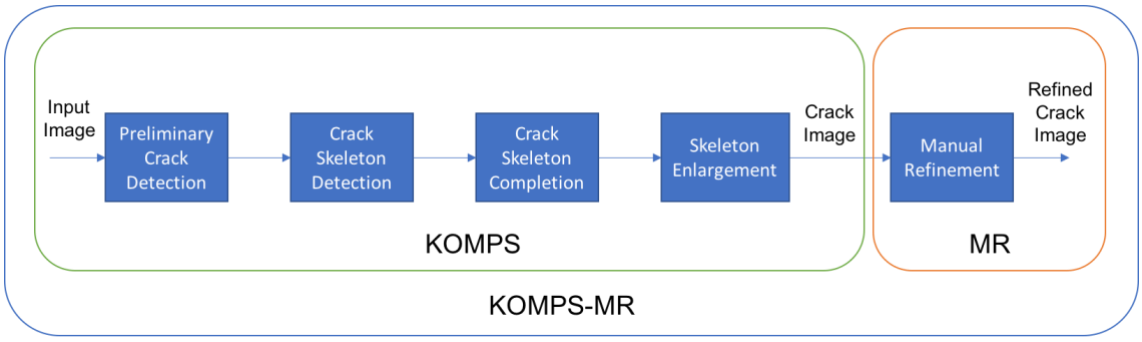


Figure 15 – Architecture of the proposed crack detection system

The proposed system is composed of five main modules. The first module performs a fast identification of crack pixels on the input image, using a new set of filters designed for road crack detection.

The second module uses the preliminary crack detection result to compute a first crack skeleton and identify a set of endpoint seeds. For this purpose, a thinning algorithm is applied, followed by a removal of unneeded spurs and branches, retaining only the longest path between any terminal points in each connected component.

The third module establishes any relevant connections between seeds belonging to different connected components, using minimal path search with a bidirectional A* algorithm. A validation step is included to ensure that relevant crack segments are added.

The final automatic processing module enlarges the estimated crack skeleton identified, to provide a good estimate of the crack width.

Finally, the user is given the opportunity to refine the automatically generated results, using a GUI.

Details on each of the modules are provided in the following sub-sections.

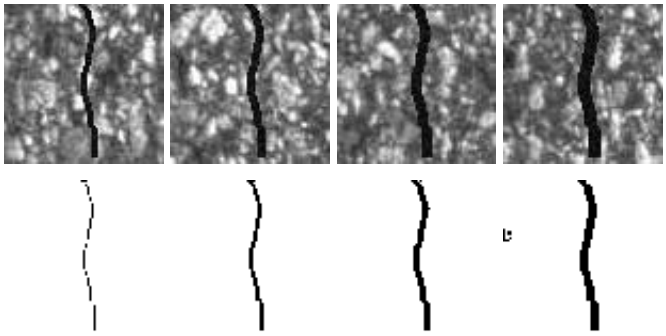
3.1.1 Preliminary Crack Detection

The goal of the preliminary crack detection module is to provide a fast detection of the most relevant cracks present in the image. For this purpose, a new 3×3 kernel was developed, with kernel values: $k_{1:3,1}$=1; $k_{1:2,2}$=-2; $k_{3,2}$=3; $k_{1:3,3}$=1. Four rotations of the kernel are considered at: 0, 90, 180 and 270 degrees, as illustrated in Figure 16.

1	-2	1	1	1	1	1	3	1	1	1	1
1	-2	1	3	-2	-2	1	-2	1	-2	-2	3
1	3	1	1	1	1	1	-2	1	1	1	1

Figure 16 – Kernel values for 0, 90, 180 and 270 degrees of rotation.

The kernel's suitability for crack detection was verified on a set of test images created for this purpose using the following procedure: (i) 150 road surface images of size 75×75 pixels were obtained by randomly cropping pavement surface images without cracks, where each pixel corresponds to 1mm^2; (ii) four artificial longitudinal crack masks were defined, with widths between 3 and 6 mm; (iii) the image areas identified by the crack masks were replaced by crack pixels intensities computed according to a Generalized Extreme Value (GEV) distribution, resulting in a total of 150x4=600 test images, as illustrated in the top row of Figure 17.



re=90.2%	*re*=91.3%	*re*=90.5%	*re*=91.1%
pr=85.3%	*pr*=91.2%	*pr*=91.5%	*pr*=91.3%
Fm=87.7%	*Fm*=91.2%	*Fm*=91.0%	*Fm*=91.2%

Figure 17 – (Top-row) samples of the 600 test images containing cracks of widths between 3 and 6mm (from left to right); (Middle row): preliminary crack detection results; (Bottom-row): corresponding evaluation metrics.

Preliminary crack detection results obtained by applying the proposed kernel to the test images, followed by a thresholding operation with a threshold value T_k discussed in Section 3.3.2 are illustrated in the middle-row Figure 17, and the corresponding recall (*re*), precision (*pr*) and F-measure (*Fm*) metrics, see section 3.2.3, are shown at the bottom-row of Figure 17. By analyzing the sample results shown, the kernel reduces the width of a detected crack by 2 pixels (or 2 mm,

for this image resolution) in comparison to the original crack width. This is not a limitation in the context of the proposed automatic crack detection system, as preliminary crack detection results are only used for identifying each crack connected component's skeleton, as discussed next. Sample preliminary crack detection results for a real pavement image are shown in Figure 18.

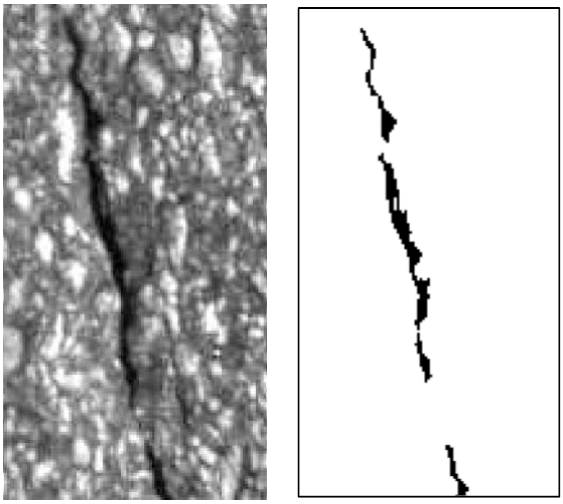


Figure 18: – (Left) Crack image. (Right) Preliminary Crack Detection result.

3.1.2 Crack Skeleton Computation

The goal of second module of the proposed crack detection system is to compute a first crack skeleton and identify a set of crack endpoint seeds. This information will be used by the subsequent minimal path selection method to complete the crack skeleton. This module architecture is illustrated in Figure 19 – Preliminary Crack Skeleton Computation module architecture.

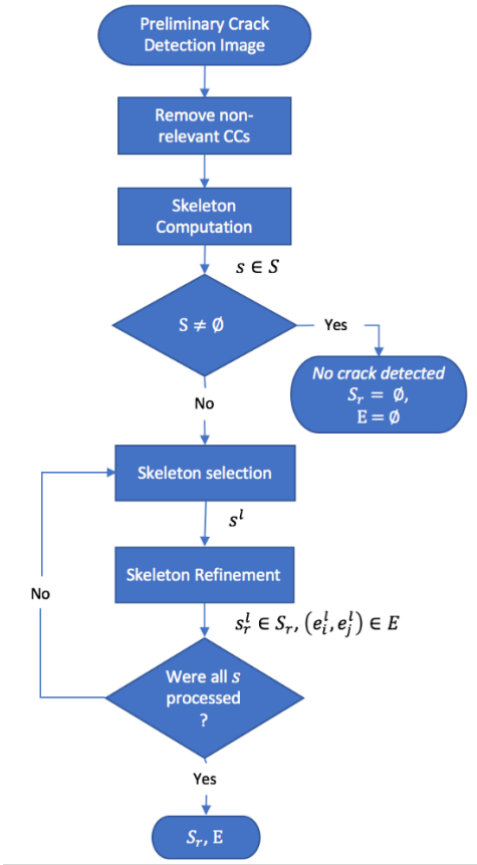


Figure 19 – Preliminary Crack Skeleton Computation module architecture.

Taking as input the preliminary crack detection results, this module includes three main steps: (i) removing non-relevant connect components from the preliminary crack detection; (ii) computing a skeleton for each connected component of the input; (iii) refine the obtained skeletons, ignoring small branches and loops, and selecting as crack seeds the two extremities points furthest apart from each other.

Removing non-relevant CCs (connected components) is done by applying a threshold value A_{ccs} to the area of each connected component present in the preliminary crack detection results. This step removes small non-relevant connected components that do not belong to the crack area, allowing an increase of efficiency for the following steps. The value of A_{ccs} is discussed in Section 3.3.1.

Skeleton computation, for each connected component is done using a thinning algorithm [38]. The result is a skeleton, s, for each connected component in the input, forming a set of skeletons, S.

Skeleton refinement removes unwanted branches. This refinement starts by moving a 3x3 sliding window along each connected component's skeleton s, and checking whether there is a single neighbor that also belongs to s, thus identifying all its extremity points. Then, a breath first search is applied to each extremity point, identifying the shortest path between all possible extremity pairs. The longest path is retained, s_r, corresponding to the refined skeleton, without branches

and loops, as illustrated in Figure 20. Endpoint seeds for the next step are identified as the extremities (e_i, e_j) of the refined skeleton paths s_r, forming a set of endpoint seeds, E.

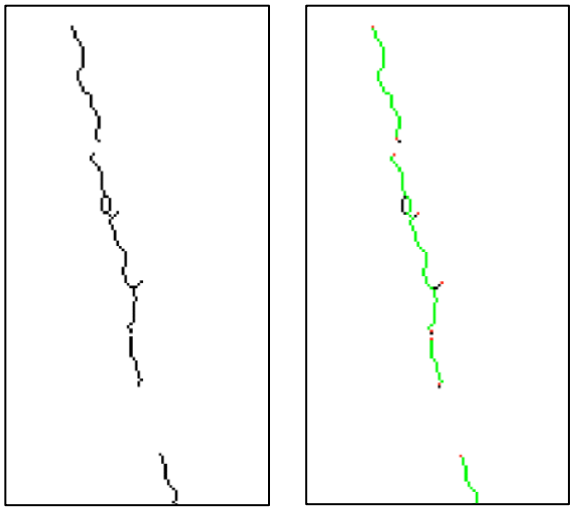


Figure 20 – (Right) Skeleton computation result for the preliminary crack detection result shown in Figure 18. (Left) Skeleton refinement result (green), with the identified extremity points marked in red.

3.1.3 Crack Skeleton Completion

As illustrated in Figure 20 the obtained set of refined skeletons, S_r, may not cover the entire crack. Therefore, the goal of the crack skeleton completion module is to expand the set of refined skeletons, providing a more complete coverage of the crack areas.

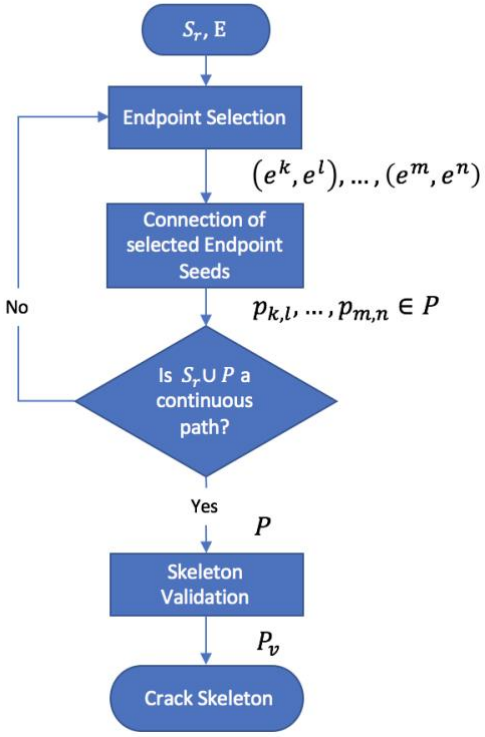


Figure 21 – Crack Skeleton Completion module architecture.

This module contains three main steps: (i) select relevant endpoint seeds from different skeleton fragments, s_r; (ii) create new skeleton branches connecting the selected endpoint seeds and (iii) validate new skeleton fragments. The first searches for additional paths, not yet considered, that may correspond to crack areas; while the second evaluates the new path candidates, to avoid including paths not corresponding to cracks.

Endpoint selection: searches through the endpoint set E to select the endpoint seeds of different s_r skeleton branches, to select those that when connected are more likely to belong to the crack area. For each s_r^k, its closest neighbor, s_r^l, is found and their two closest endpoints are selected, (e^k, e^l), so that a minimal path algorithm can be applied for their connection. This selection procedure is very efficient because it uses the information from preliminary crack detection results to identify the most relevant endpoints seeds in the crack skeleton. Therefore, a more exhaustive approach like the one proposed in [7] (each selected endpoint can have eight connections tested) is not necessary for computing the complete crack skeleton.

Connection of selected endpoint seeds: uses a minimal path search algorithm to test the connection between the previously obtained refined crack skeletons belonging to the S_r set.

The paths $(p_{k,l}, \ldots, p_{m,n}) \in P$ are computed using a known minimal path search algorithm described below in this section. These paths are marked on the image and this procedure is repeated until $P \cup S_r$ form a continuous path see Fig. 6 b). This connected component represents the complete skeleton of crack, but may contain some false positives. Some of the paths obtained may not belong to the crack skeleton.

Skeleton Validation, Among the many paths obtained in the previous step, there are a small proportion of them that do not belong to the crack skeleton. Such paths are expected to be made of pixels of darker intensities than the others, so a threshold is used on the cost path to validate candidate paths. Since the objective of this proposed step is to retain paths with the lowest mean intensities, it is important to threshold the following normalized version of cost:

$$c_v(p_{k,l}) = \frac{1}{len(p_{k,l})} \sum_{m=k}^{l} I^2(m) \qquad (5)$$

where $len(p_{k,l})$ is the length of the path (in pixels). The chosen threshold was $T_v = \mu_v + k_v\sigma_v$, where μ_v is the mean and σ_v is the standard deviation of the costs (see Section VI-A for the choice for k_v). Instead of using the normal intensity value as in [7], the squared value was used to help differentiating the candidate paths. As shown in Figure 6 c), this step makes the result converges towards the complete skeleton of the crack P_v, i.e., a one pixel-wide estimation of the crack with some artifacts. Some small isolated skeleton fragments can be left in P_v after applying T_v. These fragments may be present in the estimated crack skeleton, due to the texture of the road. A threshold operation with a value of T_s is applied on the minimal size of a skeleton fragment.

3.1.4 Skeleton Enlargement

Finally, the width of the crack is estimated resulting in the segmented image. The Skeleton Enlargement algorithm relies on absorbing the neighboring pixels of the currently detected crack, according to a thresholding operation, it was proposed in [7]. This step is repeated until no more pixels pass the thresholding test, given by $T_w = \mu_w + k_w\,\sigma_w$, where μ_w is the mean and σ_w is the standard deviation of grey levels of the detected crack.

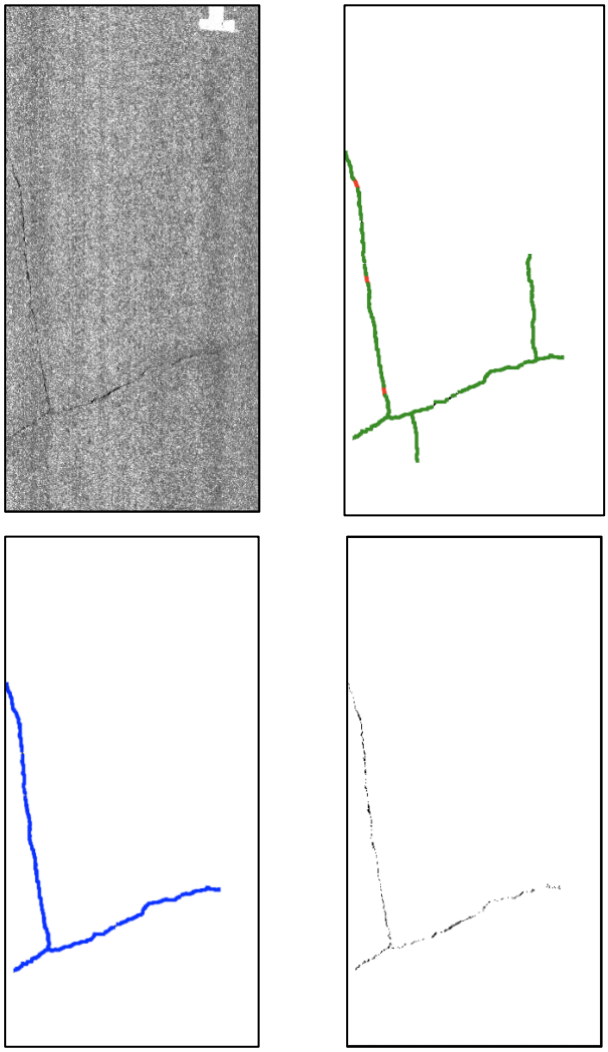


Figure 22 – (Top left) Original image. (Top Right) Preliminary Crack Skeleton Computation (Red), Crack Skeleton Completion (Green). (Bottom Left) Skeleton Validation. (Bottom right) Skeleton Enlargement.

3.1.5 Manual Refinement

After applying the automatic crack detection algorithm, the obtained result may not be exactly what an expert would produce. In that case, it is desirable that an expert can improve the existing result with minimal interactions, with the help of an appropriate graphical usage interface. An application was developed in Python that provides a simple GUI to help the user, allowing to view the results of the automatic crack detection, and refining them if needed.

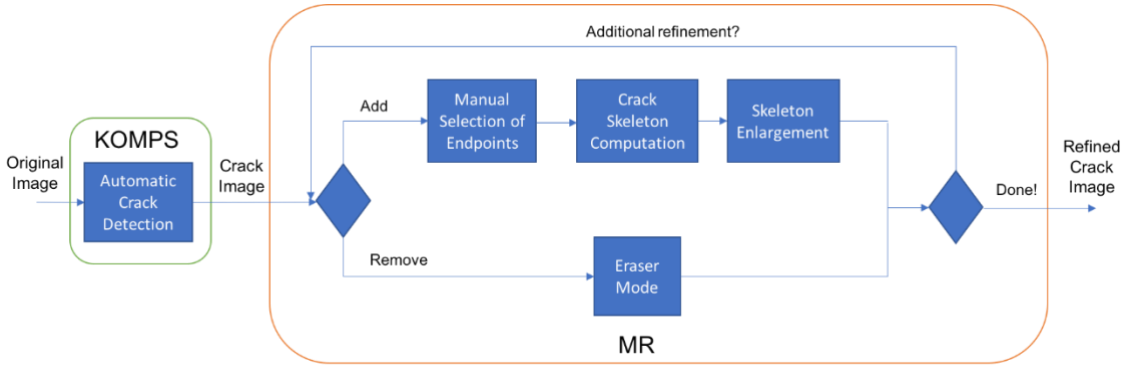


Figure 23 – Architecture of the Manual Refinement (MR) module.

The Manual Refinement module, whose architecture is presented in Figure 23, incorporates the following four components:

- *Manual Selection of Endpoints*: This module allows the user to manually select several pairs of endpoints (source and destination) within the crack image.
- *Crack Skeleton Computation*: Takes the previous selected endpoint pairs and computes minimal paths between each pair using the same minimal path algorithm used in Crack in the Skeleton Completion module to compute the crack skeleton.
- *Skeleton Enlargement*: estimates the width of computed crack skeleton, as discussed in Section 3.14.
- *Eraser Mode:* With a selected size rectangle, the user can erase any area in the detection results

There are two ways to use the GUI:

- *Manual Refinement* – The user starts by applying the automatic crack detection algorithm *(KOMPS)* and then performs a manual refinement (MR) to remove portions of incorrectly detected cracks or to add missing ones, following the architecture represented in Figure 23.
- *Pseudo Ground Truth Generator* – The GUI can be used to identify cracks in an image starting with a set of crack endpoints manually drawn by the user. In this case the architecture is similar to that of Figure 23, but not considering the initial KOMPS module.

In both operation modes, the user is able to remove or add missing crack portions:

- *Eraser Mode* – By selecting this option it is possible to remove parts of incorrectly detected cracks. The user only needs to draw a rectangle, that acts like a rubber, and overlap the crack areas to be removed from the result.
- *Addition Mode* – To add any missing crack part, the user only needs to mark two points in the image *(Manual Selection of Endpoints)*, and the *Crack Skeleton Computation* automatically connects those points relying on the computation of the minimal paths, followed by the path enlargement algorithm previously described *(Skeleton Enlargement)*. In this mode the user is allowed to:

- Apply the KOMPS algorithm in a selected image by pressing the *Crack Detection* button.
- Select several pairs of endpoints in the Crack Image, using a *right mouse click*.
- Compute minimal paths between those selected endpoints pairs and estimate its width by pressing the *Compute Skel* button.
- Undo the last action taken by pressing the *Undo* button. For instance if the last selected endpoint is miss placed from the crack, the user can unselect it by pressing that button.
- Redo all the actions taken by pressing the *Redo* button.

Examples of using the GUI for manual refinement and as a pseudo ground truth generator are included below.

Example usage of the GUI as a Manual Refinement tool:

First an image is loaded by selecting the option *Load Image* available from the menu *File*, on the top of the user interface, see Figure 24. The loaded image, along with the already detected crack areas, and the corresponding crack mask are displayed, as illustrated in Figure 25.

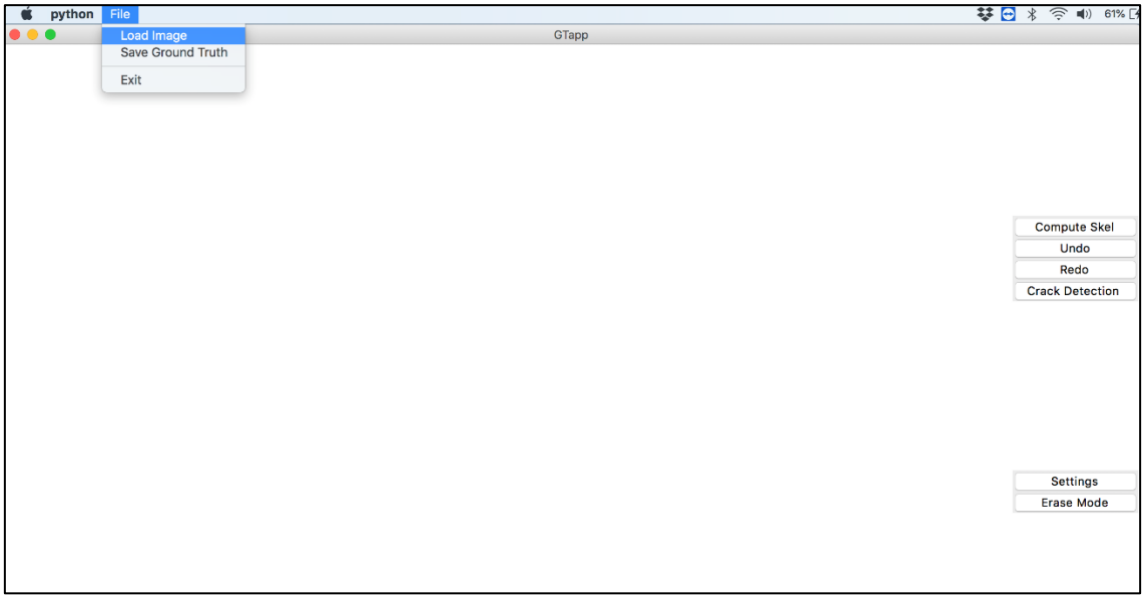


Figure 24 – Loading an image

After selecting an image, if the *Crack Detection* button is pressed, the proposed automatic crack detection algorithm is applied to the original image, obtaining a set of results as illustrated in Figure 25.

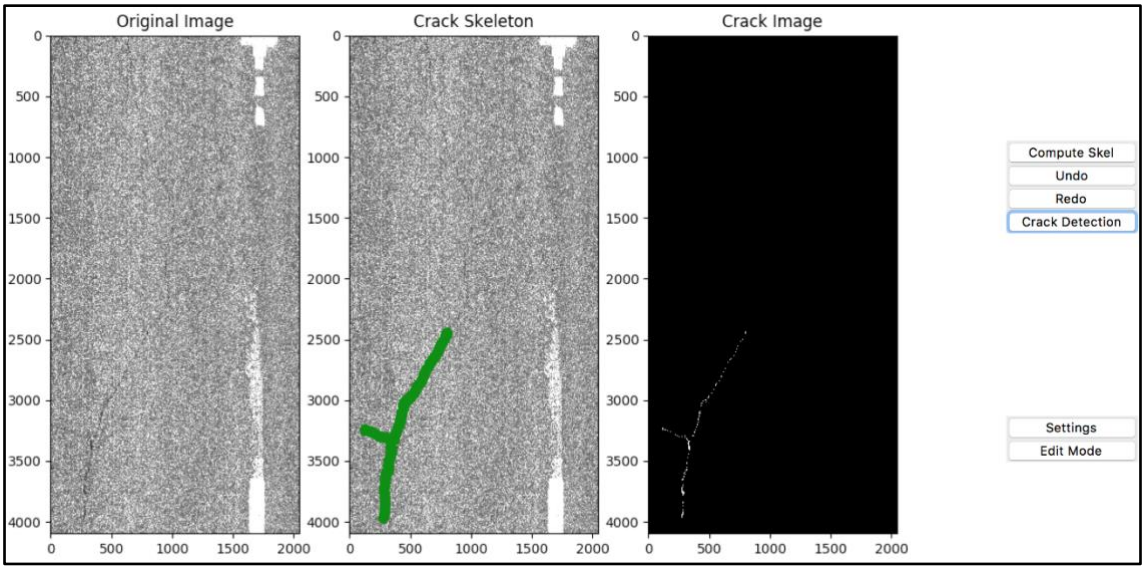


Figure 25 – Crack Detection results

In this example the crack detection algorithm missed the top part of the crack, as the preliminary crack detection module (section 3.1.1), that part of the crack is too thin for the chosen kernel operators. Consequently, no connected components were found for the next modules of the algorithm be able to detect that part of the crack. A manual refinement is therefore needed, and the provided GUI lets the user select additional endpoints with the right click of a computer mouse in the Crack Skeleton image (see Figure 26) and by pressing the *Compute Skel* button, they will serve as seeds for the minimal path algorithm used in Crack Skeleton Completion, see section 3.1.4 to find the remaining skeleton and then perform Skeleton Enlargement to obtain the Crack Image (see Figure 26).

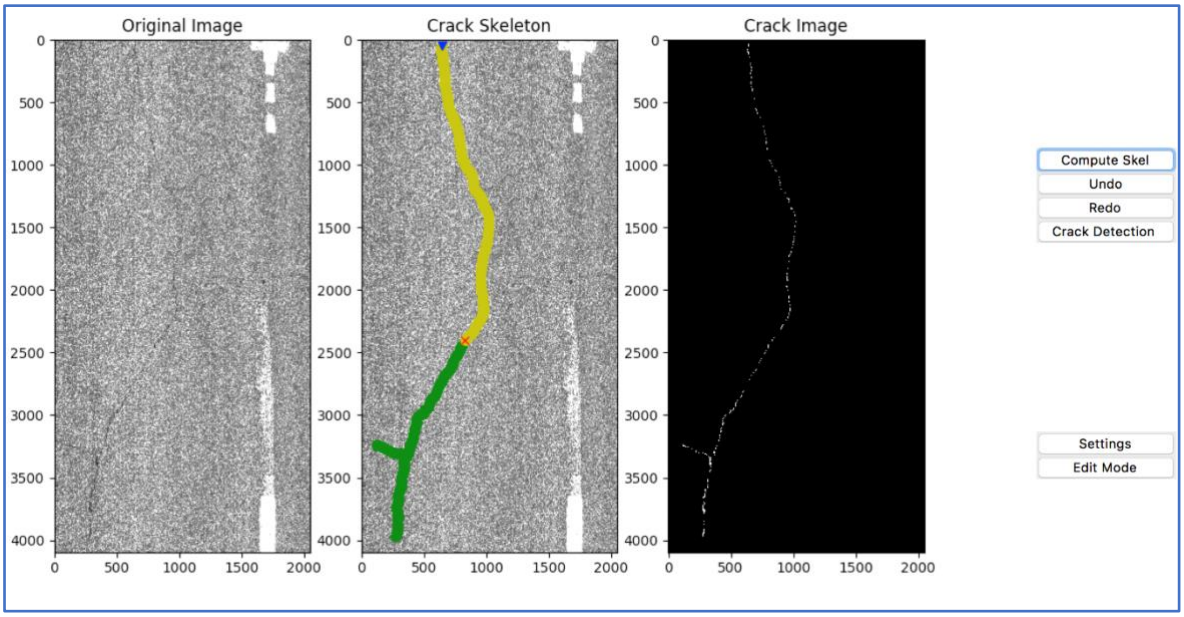


Figure 26 – Manual Refinement. (Green) Crack Detection Algorithm results; (Yellow) Manual Refinement results; (Red cross and Blue triangle) Manually marked endpoints.

In the Manual Refinement module, it is also possible to erase some parts of the crack detection results. To do so, just press the *Eraser Mode* button. In this mode, the user can draw a rectangular area and then press the *Erase* button to erase everything that is covered by that shape, as illustrated in Figure 27.

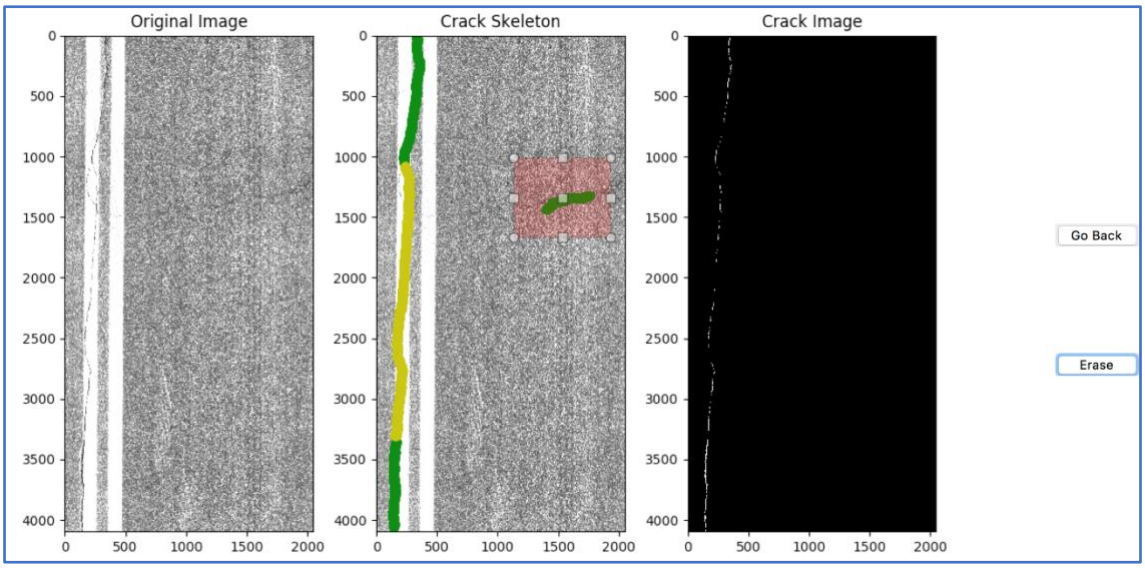


Figure 27 – Edit Mode, erasing a false positive detection. (Green) Crack Detection results; (Yellow) Manual Refinement results.

This GUI can be used to easily improve the results of any other crack detection algorithm.

Example usage of the GUI as a Pseudo Ground Truth Generator

Removing the KOMPS from the configuration illustrated in Figure 23, leaving just the MR module, the GUI can work as a pixel based ground truth creator tool, see Figure 28.

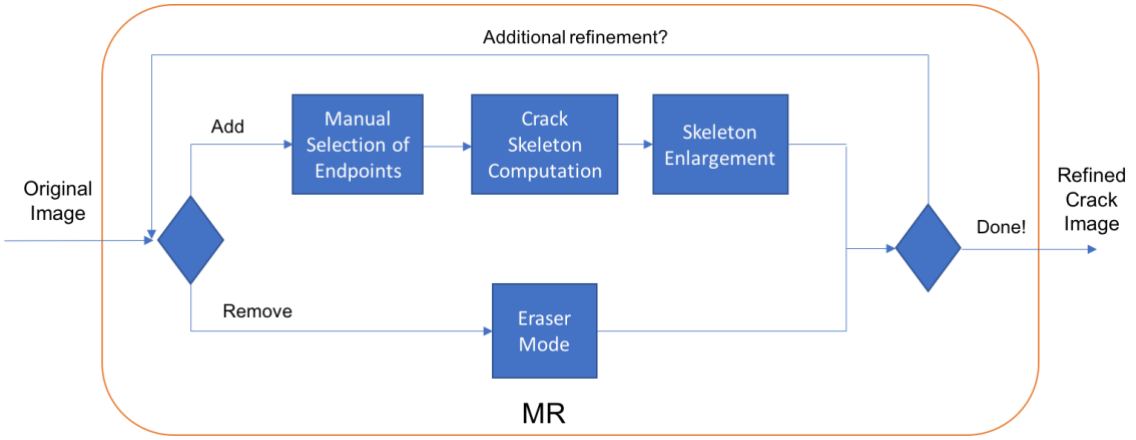


Figure 28 – Using the Manual Refinement module as a pixel based ground truth creator.

It is possible to manually mark skeleton endpoints in the *Crack Skeleton* image and then press the *Compute Skel* button to obtain the crack skeleton using the marked endpoints as seeds for the minimal path algorithm, see Section 3.1.6. Also, the *Skeleton Enlargement* module, see section 3.14, is used to estimate crack skeleton width. This process is illustrated in Figure 29

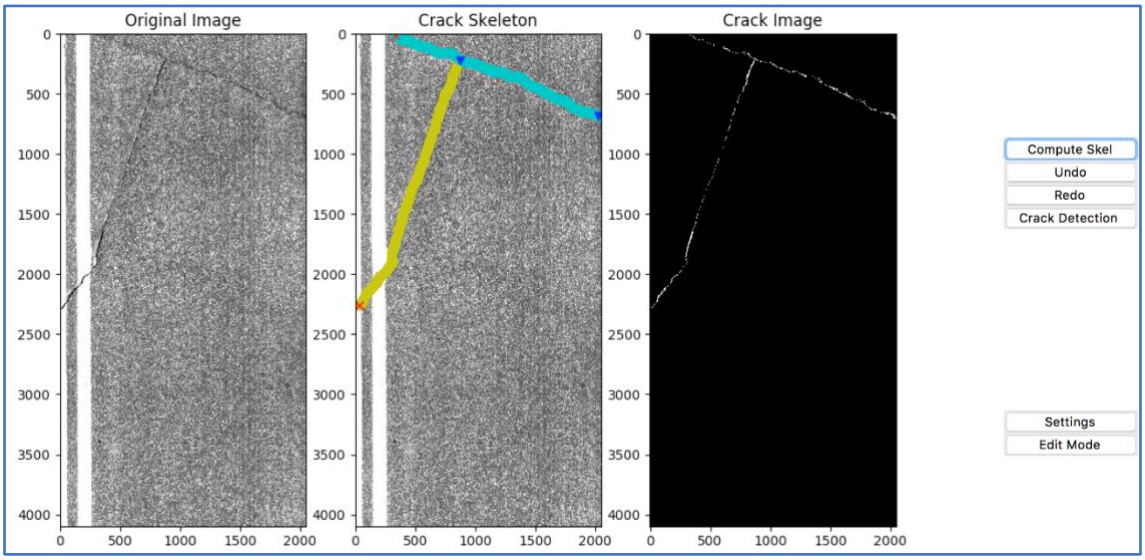


Figure 29 – Pseudo Ground Truth generation by manually select the endpoint seeds (red crosses and blue triangles)

3.2 Assessment Protocol

This section presents relevant information about used dataset to test the proposed system in the first sub-section. The process of creating the ground truth is addressed in second sub-section and the third section introduces the metrics used in the performance assessment section.

3.2.1 Dataset

The tested dataset is composed of 166 images with resolution 2048x4096 taken by the Laser Road Imaging System (LRIS) [10] in Canadian roads at 70 km/h.

The dataset was divided in two. The first part contains the first 83 images taken by the right camera, the second part of dataset contains the remaining images and were taken with the left camera. The second part showed to be $\approx 6\%$ darker than the first one.

3.2.2 Ground Truth Generation

The ground truth was generated by a human operator using a tool from Crack IT toolbox [36] that allowed to block based ground truth generation for all 183 from the two used datasets. The used block dimension was 75x75 pixels. An example of a generated ground truth is illustrated in Figure 30. The created ground truth images are subjected to the subjectivity of operator, a different operator can select different blocks for the same crack image.

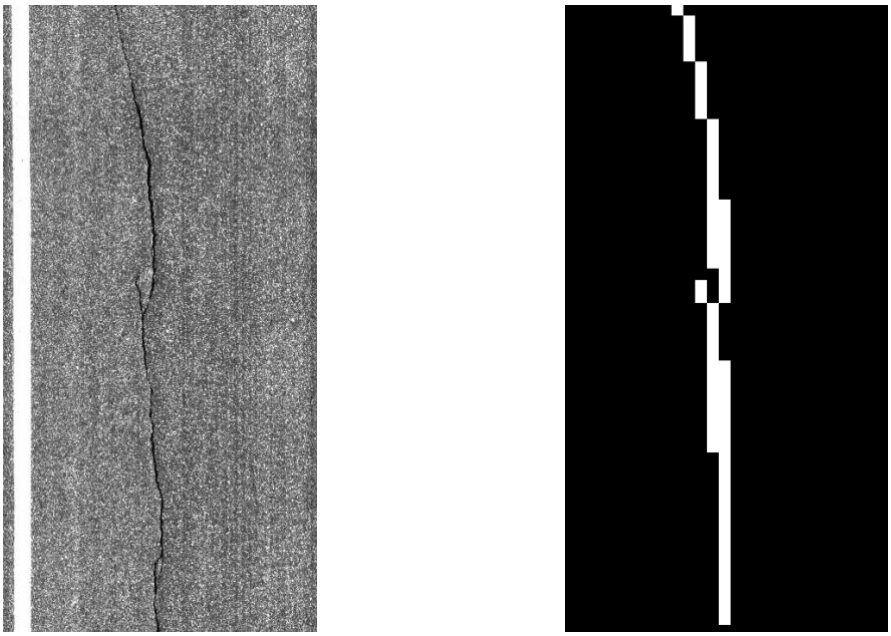


Figure 30 – (Left) Original image. (Right) Generated Ground Truth

3.2.3 Evaluation Metrics

While comparing the segmentation method with the ground truth the following pixel classification was taken in to account:

- True positives (TP) – Pixels that belong to the crack;
- False positives (FP) – Badly detected blocks;
- False negatives (FN) – Missed pixels that belong to the crack.

To evaluate the performance of the segmentation method the several indicators were used:
The Precision index (P) highlights the proportion of badly detected pixels:

$$P = \frac{TP}{TP + FP} \tag{6}$$

The Recall index (R) calculates the proportion of non-detected pixels:

$$R = \frac{TP}{TP + FN} \tag{7}$$

The F-measure (FM) is the harmonic mean between the Precision and Recall:

$$FM = \frac{2 \times TP}{(2 \times TP) + FN + FP} \tag{8}$$

The F-measure, see (6) indicator is used as a global quality measure. When $FM = 1$ a perfect detection is achieved, while $FM = 0$ equals a total inaccurate detection. A FM value above 0.7 is considered a good detection.

Due to the complexity and subjectivity of ground truth generation, one block in each of the 8 adjacent directions is tolerated for the calculus of true positives.

3.3 Performance Assessment

This section addresses the performance of the proposed system. The first sub-section introduces the minimal path algorithm used in the Crack Skeleton Completion module and addresses its performance against an alternative algorithm. The second sub-section shows how system parameters were tuned to obtain the best results. The third sub-section addresses the proposed system performance including a comparison with MPS [7] (Minimal Path Selection).

3.3.1 Minimal Path Algorithm

Minimal path search algorithms usually have a cost function related to the computed path. In the context of crack detection, the goal is to find a curve that fits the crack skeleton by iteratively minimizing a cost given by a defined cost function.

Assuming darker pixels correspond to a crack the following cost function is used:

$$\mathrm{cost}(p_{ij}) = \sum_{i=m}^{j} I(m) \tag{9}$$

where I is the image represented by gray level intensity values, i is the source, j the destination and m is a pixel of the path between i and j. This function tries to minimize the sum of pixel intensities in the path.

The chosen optimization method was A* algorithm that is an informed version of Dijkstra algorithm. The heuristic function $\mathrm{h(n)}$ allows an estimate of the minimum cost from any $vertex_n$ to the destination $vertex_m$.

It is important to choose a good function $\mathrm{h(n)}$ because the behavior of A* is controlled by it. If $\mathrm{h(n)} \leq \mathrm{h*(n)}$, where $\mathrm{h*(n)}$ is the heuristic that provides perfect information, A* is guaranteed to find the minimal path.

The develop heuristic function is given by the following formula:

$$\mathrm{h(n)} = \mathrm{d(n, goal)} \times \mu(\mathrm{K}) \tag{10}$$

where $d(n, goal)$ is the Manhattan distance between a given node n and the goal node, and K is a square window centered in the goal pixel with 1 cm of side.
This heuristic function is highly adaptable, it varies accordingly the radiometric information present in the image between a source and a destination pixel.

To validate the develop heuristic several paths were computed in different images with the same sources and destinations using A* and Dijkstra algorithm. For both algorithms was considered their bidirectional versions. The path costs were compared between the two using three different images, see Table 1.

Table 1 – Averaged path cost comparison between A and Dijkstra.*

Image	A* cost	Dijkstra cost	Error %
Crack003	13534	13272	1.97
Crack010	5737	5640	1.72
Crack138	9685	9576	1.14

The error between the average costs of A* and Dijkstra is rather low, less than 2%. This shows that the used heuristic function is not totally admissible, but the tradeoff between the speedup and the error pays off in the end by obtaining a quite faster crack detection, see Table 2.

Table 2 – Time comparison between A and Dijkstra.*

Image	A* time(s)	Dijkstra time(s)	Speedup
Crack003	18.87	36.36	1.93
Crack010	11.13	28.48	2.56
Crack138	15.97	35.20	2.20

This increase in performance is shown in Figure 31 by illustrating the node expansion for both algorithms. The A* (bidirectional) needs less node expansions to find minimal paths than Dijkstra (Bidirectional), the heuristic function helps the algorithm to expand in the wright direction by increasing the cost of other paths that grow in other directions.

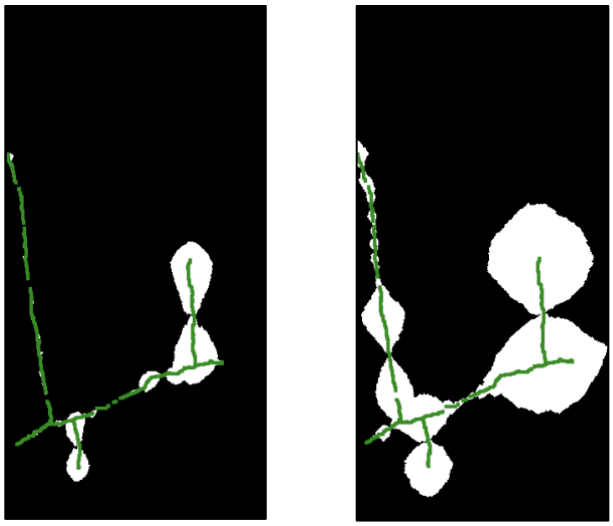


Figure 31 – Crack 138 node expansion. (Left) A. (Right) Dijkstra*

3.3.2 Parameter Tuning

In practice, the proposed algorithm requires the tuning of five parameters: the kernel threshold T_k and connected component size A_{ccs} for the preliminary detection, the constant k_v and threshold T_s for skeleton validation and the constant k_w for the skeleton enlargement.

The value of T_k must not be too large to limit the detection of false positives in the preliminary detection stage. However, in the skeleton validation step of the Crack Skeleton Completion module some of previous detected false positives can be rejected in the final detection. A suited value for T_k is 265, see Figure 32. A small value of T_k influences by decreasing the system performance, as illustrated in Figure 32.

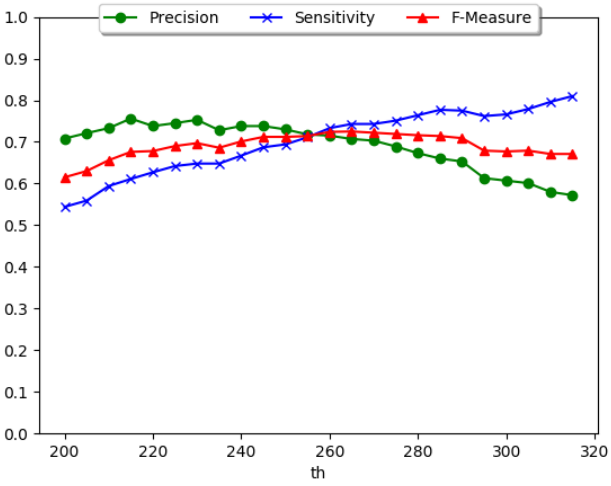


Figure 32 – Precision, sensitivity, and FM value variations with respect to T_k parameter for both datasets.

The choice of A_{ccs} determines the threshold value that allows to retain the most relevant connected components and discard texture artifacts in the preliminary detection stage. Smaller values of A_{ccs} lead to discard only few connected components, so the following steps become more computationally demanding. Larger A_{ccs} values eliminate larger connected components, with the risk to remove crack pixels. In practice, it has been found that $A_{ccs} = 25$ corresponds to a suited value to a resolution of $1\ mm/px$, see Figure 33

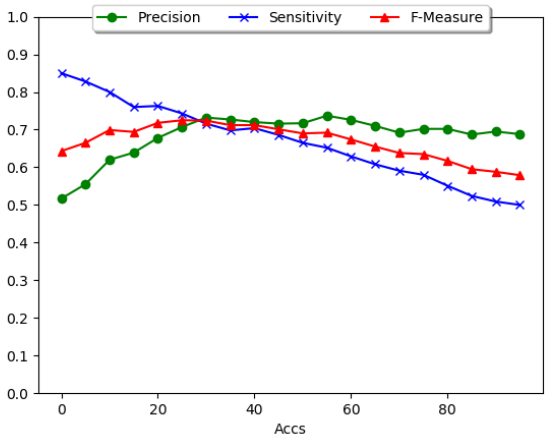


Figure 33 – Precision, sensitivity, and FM value variations with respect to A_{ccs} parameter for both datasets.

The choice of k_v determines the threshold value $T_v = \mu_v + k_v\sigma_v$ that allows to retain only the paths of lowest costs among the ones found at the crack path computation step. This parameter is a crucial one but its value is quite easy to choose. In fact, the results presented in Fig. 8 highlight that the value of k_v corresponds to a trade-off between precision and sensitivity. Beyond k_v = 3, the precision becomes too low, a lot false positives start to appear. Based on Figure 34 an intermediate value k_v = 1.5 was retained.

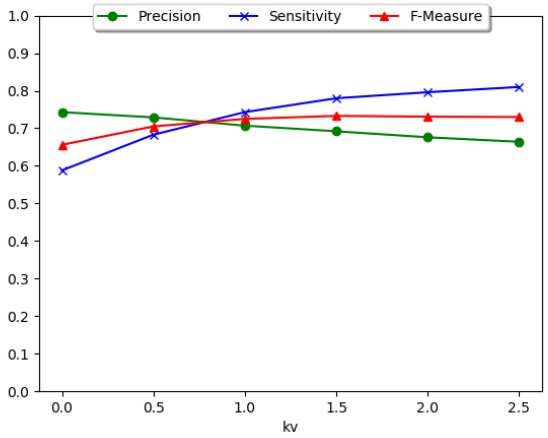


Figure 34 – Precision, sensitivity, and FM value variations with respect to k_v parameter for both datasets.

As concerns the threshold T_S, Figure 35 indicates that if the value $T_S \geq 30$ the system will perform well. $T_S = 100$ was the value used in the simulations and in the system performance. Given that road cracks that are lower than $10\ cm$ are usually not considered as significant this threshold value seems reasonable. Such a choice is coherent for an image resolution of $1\ mm/px$ per pixel and should be fitted to a different resolution.

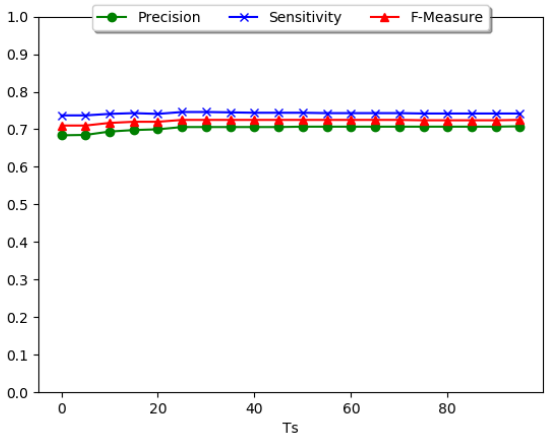


Figure 35 – Precision, sensitivity and FM value variations with respect to T_s parameter for both datasets.

Finally, the choice of k_w determines the threshold value to merge some dark pixels neighboring the currently detected crack structure. An appropriate value for k_v should increase the sensitivity with a low impact on the precision. According to [7], k_w = 0.6 is the best choice. Given that this dissertation uses block detection as the evaluation metric this value does not have a big impact *f-measure*.

3.3.3 System Performance

In this section, the performance of the proposed system without using manual refinement is evaluated on the images of the datasets 1 and 2 using a block based approach. Also a performance comparison (pixel based) with the MPS algorithm [7] using three images from the

Aigle-RN [32] system is presented, here manual refinement is considered. All the performance results were obtained using a computer with an AMD Ryzen 7 CPU (8 cores), 32GB of DDR4 ram and running on Linux 64 bit operating system.

LRIS Dataset Performance

Here, the performance of the proposed system without manual refinement is evaluated on the images of the real datasets 1 and 2 introduced in Section 3.2.1. Manual refinement is not considered here because the objective is the evaluation of the automatic and unsupervised modules of the proposed system. With MR (Manual Refinement), it is possible to correct any errors coming from previous modules, so a performance evaluation considering this last module wouldn´t give much information about validity of the results given by the previous modules. Table 3 shows the results of the automatic modules of the system (KOMPS). For this results the value used parameters were: $T_k = 265,\ A_{ccs} = 25,\ k_v = 1.5,\ T_s = 100,\ k_w = 0.6.$

Table 3 – Averaged values of Precision, Recall and F-measure for images in the Dataset 1 and Dataset 2.

	Precision (P)	Recall (R)	F-measure (FM)
Dataset 1	0.77	0.83	0.80
Dataset 2	0.61	0.73	0.67

The performance is significantly better for Dataset 1, with an F-Measure of 0.80 against 0.67 for Dataset 2. For the first Dataset, the system could detect more parts of the crack (0.83 against 0.73 in the Recall measure) while having less false positives (FP) detections (0.77 against 0.61 in the Precision measure).

The overall performance on dataset 2 is under the overall performance of the first dataset. There are several reasons that can create that difference. Such as, the average pixel intensity of all images from Dataset 2 is 6% lower than the one from Dataset 1. The images from the second dataset seem to present much more artifacts, like tire marks or dark zones, than the ones from the first dataset. Those artifacts and the difference in the average intensities could affect parameter choice, especially the kernel threshold T_k that directly depends on pixel intensities.

However, considering the speed and efficiency of this system, without manual refinement, these results show that it can be used as fully automatic and unsupervised crack detection algorithm. Obtaining these results for both datasets (186 4096x2048 images) took 808.6 seconds using the previously mention computer.

The manual refinement could be used to quickly correct the detection results.

Performance Comparison with MPS algorithm [7]

For this performance comparison with MPS (Minimal Path Selection) algorithm, three images acquired with the Aigle-RN system were chosen [32] – see Figure 36. In this performance evaluation the *Manual Refinement* module was also considered. The comparison of results was done using a pixel based pseudo ground truth created by an expert technician using the system proposed in [37]. Both the MPS results and the pseudo ground truth are available at [39]. The

resolution of these three images is $1.67\ mm\ /px$. Since these images have a different resolution than the $1\ mm/px$ available in the LRIS dataset [10] considered for the development of the proposed KOMPS system, some parameters must be adapted, notably A_{ccs}. The new A_{ccs} value must be $25/1{,}672$, where 25 was the value considered for a resolution of $1mm/px$, so the new value for this dataset is $A_{ccs} = 9$, since A_{ccs} represents an area not a length. The same principle was used to define the new $T_s = \frac{100}{1.67} \approx 60$, but in this case this threshold represents a length. The other parameters were not changed.

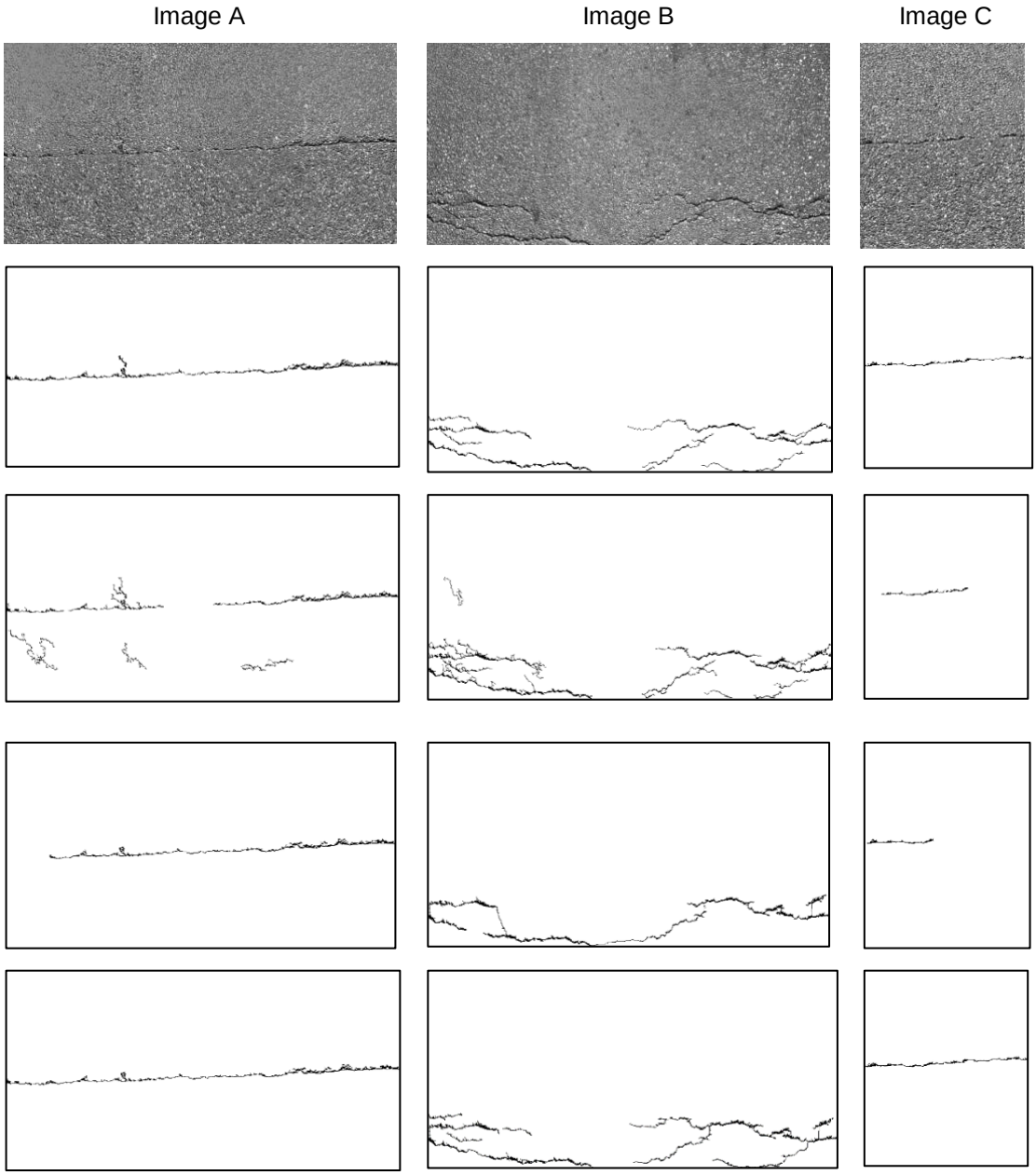


Figure 36 – Visual performance comparison against MPS algorithm. (First row) Original Images. (Second row) Pixel based Ground Truth from [ref]. (Third row) MPS results. (Fourth row) Proposed system results without Manual Refinement (KOMPS). (Fifth row) Proposed system results with Manual Refinement (KOMPS-MR).

Here, three different images acquired with the Aigle-RN system have been selected, with growing crack detection difficulty from Image A to Image C, the latter being the most difficult to analyze. The MPS system performs well on Images A and B, which is confirmed by a high F-measure value, see Table 4. In the case of Image C, MPS produces a coherent result, but fails to detect the complete length of crack. This explains why the obtained F-measure value is low, see Table 4. Looking at the KOMPS results without manual refinement, illustrated in the fourth row of Figure 36, the performance is quite good, except for some missing parts of the crack and some false positive detections in case of image B. This is confirmed by the high value of precision and recall for the KOMPS system for the first two images – see Table 4. Image C gave the worst results for KOMPS, as the system could not detect a larger part of the crack present in the image. One of the reasons that justifies this behavior is because the crack is very thin, leading to the system not being able to detect parts of the crack, as the kernel used in the *Preliminary Crack Detection* module is not well adapted to the resolution of this dataset. For instance, this module only detects well cracks wider than 2 pixels, which with such low resolution images ($1.67\ mm/px$), corresponds to a minimum detectable crack width of $3.33\ mm$. The other reason is the usage by MPS of a pre-processing technique to deal with non-uniform illumination. This can influence the choice of some system parameters specially the threshold T_k because it directly depends on the image pixel intensities that can be affected by pre-processing.

When including *Manual Refinement* (KOMPS-MR), the results are very good for all the images, all the used metric values are over 0.9. This module allows the correction of the automatic results produced by KOMPS. A user with a help of a GUI can easily add missing crack segments or erase some badly detected ones to improve the system performance, see Table 4 and Figure 36.

Table 4 – Metrics values for three different methods applied to the three images in Figure 36. (Top) F-measure; (Middle) Precision; (Bottom) Recall

F-measure	MPS	KOMPS	KOMPS-MR
Image A	0.83	0.91	0.94
Image B	0.9	0.85	0.9
Image C	0.66	0.75	0.95

Precision	MPS	KOMPS	KOMPS-MR
Image A	0.81	0.98	0.98
Image B	0.84	0.86	0.85
Image C	0.50	0.96	0.93

Recall	MPS	KOMPS	KOMPS-MR
Image A	0.87	0.85	0.90
Image B	0.98	0.84	0.96

Image C	0.96	0.61	0.98

Table 5 shows a time comparison between the automatic results of the proposed system (KOMPS) and a partially implemented MPS. This implementation does not have the partitioning technique in the *P1) step (Elimination of artifacts), see section 2.4.2*. However, for this time comparison is not relevant because the main processing time is in the minimal path estimation step *C2)*, where it computes all possible minimal paths between each selected endpoint from the previous step *C1)* (Endpoint Selection). The results show that KOMPS is much faster than MPS while providing better results for image A and C and a similar result for C., see Table 4.

Table 5 – Time comparison between KOMPS and MPS (partially implemented).

	MPS (s)	KOMPS (s)	Speedup
Image A	125.37	3.01	41.7
Image B	121.53	3.14	38.7
Image C	35.87	0.6	59.8

Table 6 – Number of mouse clicks in the MR module. shows the number of right mouse clicks in the Manual Refinement module for each image. For the images A and C only 2 two clicks were needed, because both of them only have a transversal crack and KOMPS could detect great part of it. Adding one crack segment to each of the KOMPS result completed the detection, see Figure 36. Image C needed 12 mouse clicks to complete the detection, because the crack that is present in the image has a very irregular shape.

Table 6 – Number of mouse clicks in the MR module.

	Number of Mouse clicks
Image A	2
Image B	12
Image C	2

4 Crack Detection using a Light Field Camera

This chapter proposes a new crack detection strategy, by exploring light field images captured with a Lytro Illum [6], to ascertain about its suitability as input data used by an automatic pavement surface crack detection system, notably by exploring the disparity between the different viewpoints available from the captured imagery. Obtained results favorably compare with those achieved by the analysis of conventional 2D images.

The chapter is organized as follows. In the first section an introduction to the basic concepts about light fields is presented. The second section describes the proposed automatic crack detection system. The assessment protocol and the performance assessment results are presented in the third and fourth sections, respectively. The Performance Assessment section includes a comparison with a system that processes conventional 2D images of road pavement surface.

4.1 Introduction to Light Field

This section discusses light field based visual representations, namely the one obtained from a lenslet light field camera [6]. This is a very interesting topic since those representations provide much more information than standard visual formats. The extra information can be exploited to improve the conventional methods used in a crack detection procedure. There is not much work available in the literature on the use of light-field cameras in a crack detection problem. The basic concepts about light field visual representations are presented, followed by the description of a simple system that shows the suitability of this type of imaging for road surface crack detection.

4.1.1 Basic Concepts

The starting point on any computer vision related topic is understanding the plenoptic function and then derive the light field concepts from there. This function describes all visual information in the world and it is arguably one of the most important concepts to grasp in this Chapter.

All the visual information in the world is ultimately represented by a description of the electromagnetic field in the visible spectrum for every point in space [40]. The plenoptic function evaluates light not only in terms of objects, geometry and texture, but rather in terms of electromagnetic waves themselves. A scene is no longer a set of surfaces, but rather a volume through which light waves flow, and the act of seeing is no longer one of reaching out to objects with rays, but of measuring the light passing through two openings, the pupils of our eyes.

Assuming only non-coherent light sources, a propagating wave can be described as an infinite sum of random-phase sinusoids, each with different energy, which is possible to represent mathematically as a power density for each wavelength λ. Following this reasoning, an electromagnetic wave at any point (x, y, z) in space can be represented as a sum of wave fronts coming from all directions, each direction described by an azimuth, orientation pair (θ, Φ). If a wave's energy varies in time, this can be quantified by adding in a temporal dimension. Putting

all this information together it is possible to obtain the plenoptic function as $(x, y, z, \Theta, \Phi, t, \lambda)$ .In Figure 37 this function is conceptually visualized.

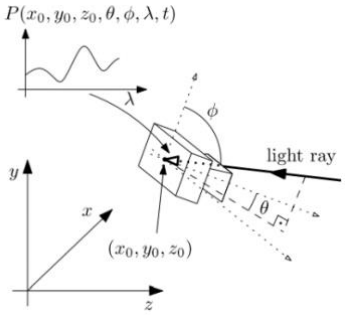


Figure 37 – Visual representation of the Plenoptic Function [40].

This 7-dimensional function contains a lot of information so it is computationally heavy and also difficult to represent it visually, so it is necessary to reduce its dimensionality. Since 19th century that technology has been finding ways to sample a plenoptic function of lower dimensionality, from black and white photography, to 3D cinema, and everything in between. In this sense, every visual representation of the real world that has ever been conceived is a reduction of the plenoptic function. Some of them are simpler and easier to understand, such as: photography, where the plenoptic data is just a 2D (x, y) plane; or video, where the temporal dimension is introduced, creating a (x, y, t) representation. Others are even more complex and richer in information such as the light field representations. These will be listed and presented in this section, as they are essential to understand the rest of the report, especially the lenslet light field.

4.1.2 Light-Fields

Light-fields are a 4D parametrization of the plenoptic function. In current literature, light-fields are divided into several types, the most relevant of which are Multiview light-fields, and lenslet light-fields, this last one being the important ones for this report. The main difference between these two formats is the way how they are captured. The two will now be explained as follows:

- **Multiview Light-Fields:** Captured using an array of cameras and acquiring several views from different viewpoints. The array can be sorted in a multitude of arrangements, the simpler being 1D linear or circular arrays of equally, or irregularly, spaced cameras. The different views can be high-resolution, and this kind of setup can acquire dynamic scenes. For example, in a 2D camera array (see Figure 38) the Plenoptic function is represented in four dimensions $P(k, l, u, v)$, where variables (k, l) are the position of the camera in the 2D array, while (u, v) is the position where a light ray strikes the sensor..

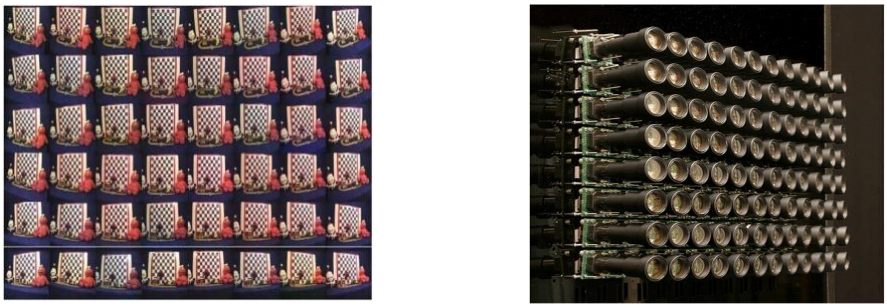


Figure 38 – (Left) Multiview Image, (Right) 2D camera array.

- **Light field gantries: Light field gantries:** For static scenes, a cheaper way to capture them is by using a single moving camera instead of using a multi-camera array. An example of a setup that captures a 4D light field of a scene is shown in Figure 39. It acquires lots of samples of a small object from hemispherical viewpoints under controlled lighting.

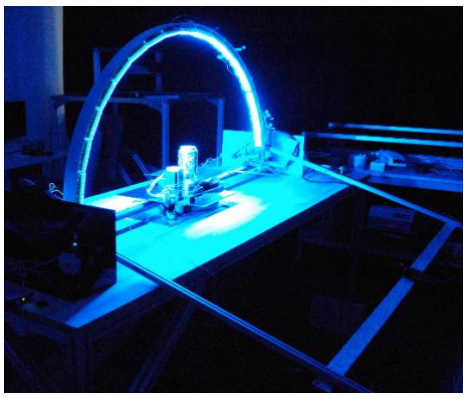


Figure 39 – HCI Light Field Dome

- **Lenslet Light-Fields:** Lenslet light-fields follow a very similar process to Multiview and Gantry light-fields. However, unlike Multiview and Gantry, a lenslet light-field is captured using a single, wide aperture camera. In this way the plenoptic function is reduced to a two plane parametrization, where plane (u, v) is the plane of the camera lens, while plane (s, t) is the plane of the sensor itself. Using this setting, any light ray captured by the camera is simply represented by two sets of coordinates, one pointing to where the ray intersects the lens, and the other to where it hits the sensor Figure 40 (Left) shows this two plane parameterization and Figure 40 (Right) shows a lenslet camera structure camera structure.

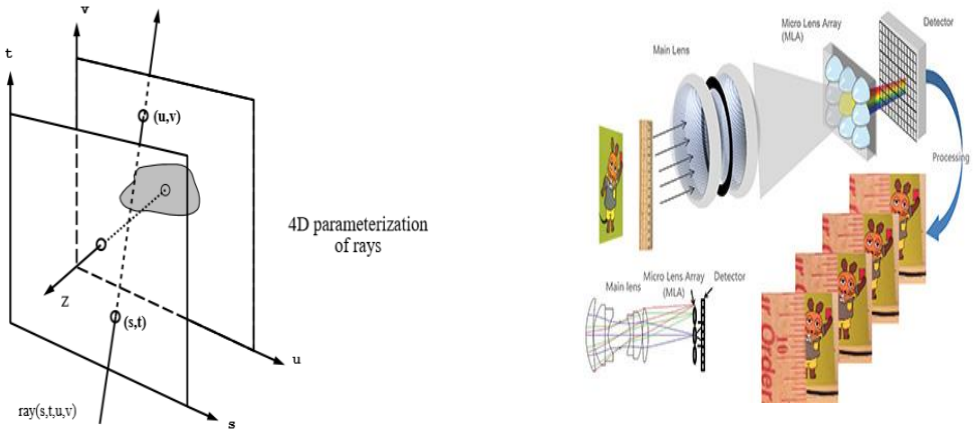


Figure 40 – (Left) Two plane Parametrization of light rays [41]*. (Right) Lenslet camera structure.*

Capturing Light Field Data

As stated before, several technologies allow sampling plenoptic function with lower dimensionality. The conventional photographic and video cameras are the most traditional ways to sample the plenoptic function. The dimensionality reductions of the plenoptic function that were presented above are quite recent and more complex. Thus, two alternatives for capturing this type of information are listed next:

- **Arrays of Traditional cameras:** These arrays can be used to capture multiview content, where the arrays can be setup in several ways, including 1D linear or circular arrays, or 2D arrays of cameras. These can either be equally or not equally spaced, depending on the application. An example is showed in Figure 38 (Right).

- **Arrays of Micro lensesArrays of Micro lenses:** These arrays are in fact a single photographic camera, commonly known as light-field cameras. The Lytro company has been developed several models of micro lens cameras in recent years and *Lytro Illum v2* was the one used in this experiment [6] (see Figure 41). These cameras capture the direction, color, and intensity of millions of individual rays of light, which allows for *a posteriori* photograph refocusing (see Figure 41) and slight perspective shifts, along with the computation of an approximate depth map of the image. for this experiment the depth map wasn´t very helpful.

Figure 41 – (Left) Lytro Illum v2 camera. (Right) Example of posteriori photograph refocusing.

4.2 Proposed System

The architecture of the proposed road pavement surface crack detection system using a light field camera (here designated by the ***L**ight **F**ield **C**rack **D**etection* system – LFCD) is presented in Figure 42.

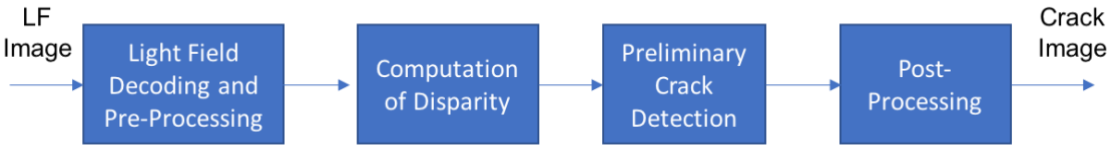


Figure 42 – Light Field Crack Detection system architecture.

Since the light field (LF) camera captures light rays travelling in different directions, a richer visual scene representation is obtained, which provides additional cues, useful for the automatic detection of pavement surface cracks. More specifically, the proposed LFCD system allows recovering a multi-view array of 2D sub-aperture images, from which the disparity can be exploited for improving crack detection. The main components of the proposed system are described in the following.

The Decoding step, takes the raw light field and creates a 15x15 matrix $M(i,j)$ of 2D sub-aperture images, each with spatial resolution 435x625 pixels and representing a slightly different perspective, see. This step shows how richer is the light field data when compared to a conventional 2D traditional image. Note: in the image, it´s only represented a 7x7 matrix of images for demonstration purposes.

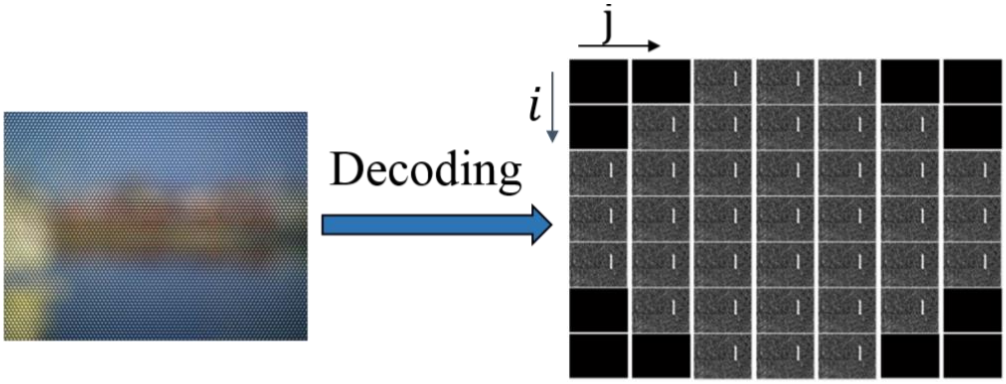


Figure 43 – Light field decoding process. Note: in the image, it´s only represented a 7x7 matrix of images for demonstration purposes.

In the Pre-Processing**,** a gray scale transformation is applied to the images. Since it is assumed that crack pixels are darker than non-crack pixels, those pixels with high intensities can be replaced with an intensity value that is clearly above the crack intensities, but allowing to reduce the intensity variance. All of the images are saturated with a chosen value of 50, see equation (11), chosen by some experimentation. The results are illustrated in Figure 44.

$$I \in M \qquad S = \begin{cases} 40 & if \;\; I(x,y) > 50 \\ I(x,y) & if \;\; I(x,y) \leq 50 \end{cases} \tag{11}$$

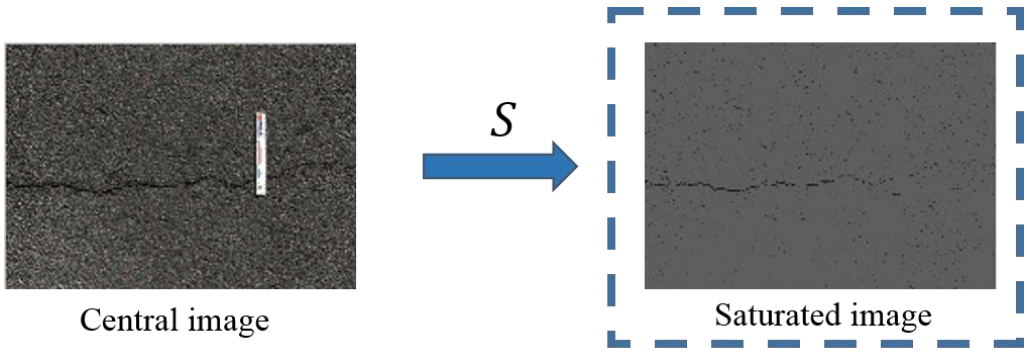


Figure 44 – Saturation.

To exploit the disparity present in the light field images, a selection of sub-aperture images is considered after some experimentation, and according to [42] considering the difference between images at position symmetrical to the central sub-aperture image.

Figure 45 illustrates the creation of the **Vertical (**$dimv$**) and Horizontal (**$dimh$**) Disparity Image**, when subtracting the images of column (resp. row) 3 with those of column (resp. row) 13 of the matrix, and summing the obtained differences as it also showed in the following equations.

$$dimh = \sum_{i=3}^{13} [M\ (i,3)\ - M(i,13)] \quad (12)$$

$$dimv = \sum_{j=3}^{13} [M\ (3,j)\ - M(13,j)] \quad (13)$$

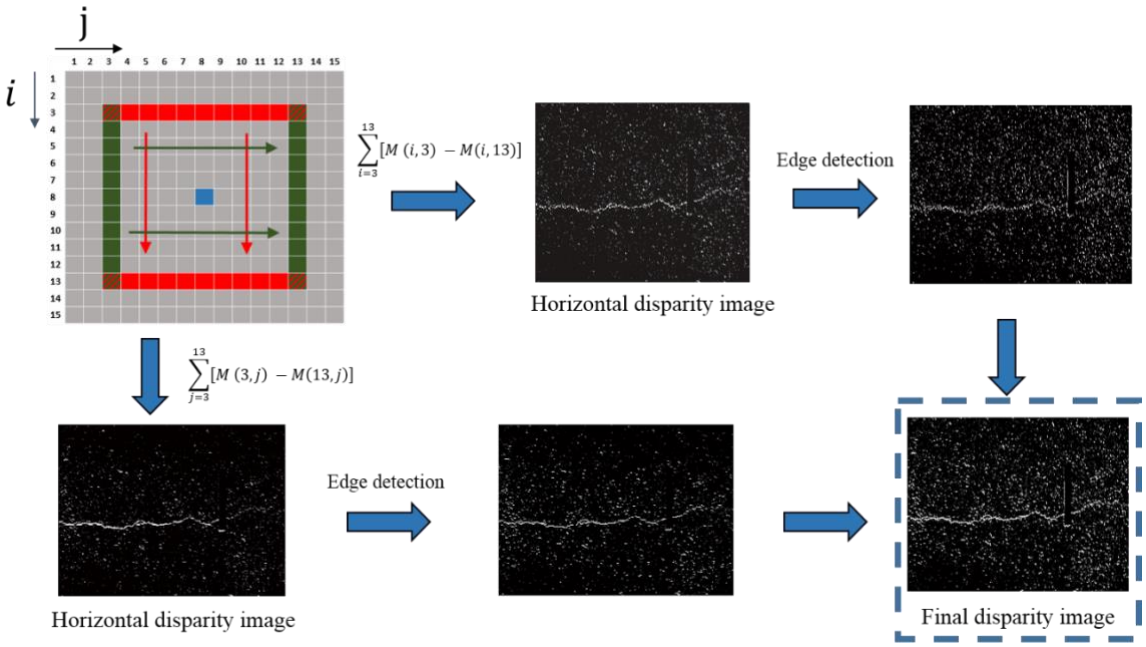


Figure 45 – From the multi-view array of 2D rendered sub-aperture images to the final disparity image.

In the preliminary crack detection module, a simple and fast algorithm is applied to enhance the visibility of crack areas in images. For this purpose, an edge detector is applied to the horizontal and vertical disparity images, $dimh$ and $dimv$ respectively, and the results are added to include the edges with both orientations. Along with edge information due to presence of road cracks, also other small pavement irregularities (typically corresponding to false detections) are enhanced and appear as undesired detection results in the edge image. To reduce this effect, a 5×5 median filter is applied. A sample result is illustrated bellow:

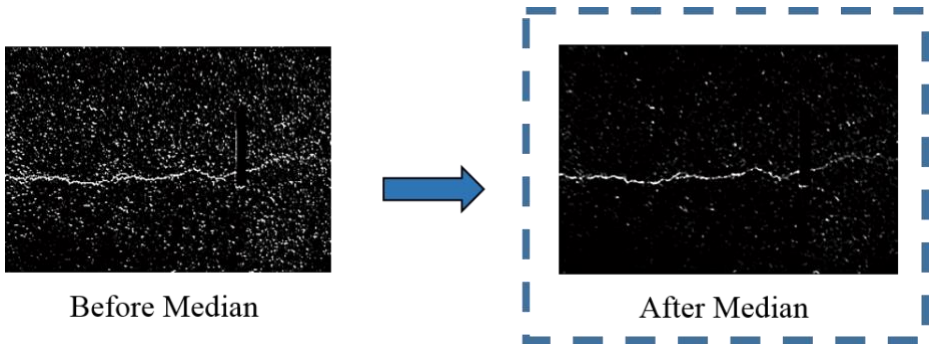


Figure 46 – Median filter results.

The goal of the post-processing module is to reduce possible false detections resulting from the preliminary crack detection, by applying a region filter, which identifies image areas more likely to correspond to cracks. After that, a connected component analysis is performed, aiming the reduction of those possible false detections, as represented in the block diagram of Figure 47.

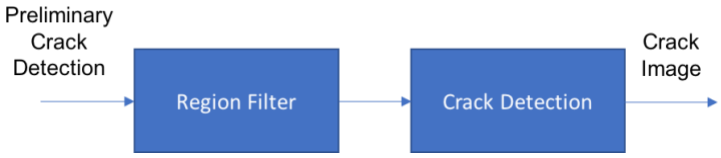


Figure 47 – Detail of the post-processing module.

The region filter considers a set of small circular regions with 16-pixel diameter (empirically chosen after exhaustive testing), covering the edge image in such a way to ensure a 25% overlap with neighbors. It then assigns label '1' to circles with more than 50% of edge pixels, and label '0' otherwise. The overlapping is useful to maintain the continuity of existing crack regions.

Figure 48 – Image with the marked circles.

Connected components are identified in the region filter output image, and very short connected components are excluded as they are very likely to correspond to false crack detections.
These identified connected components form a mask that covers the areas where relevant cracks are located, as illustrated in Figure 49(left). That mask can then be used to exclude false crack detections obtained in the preliminary crack detection, leading to better detection results at pixel level, as illustrated in Figure 49 (right).

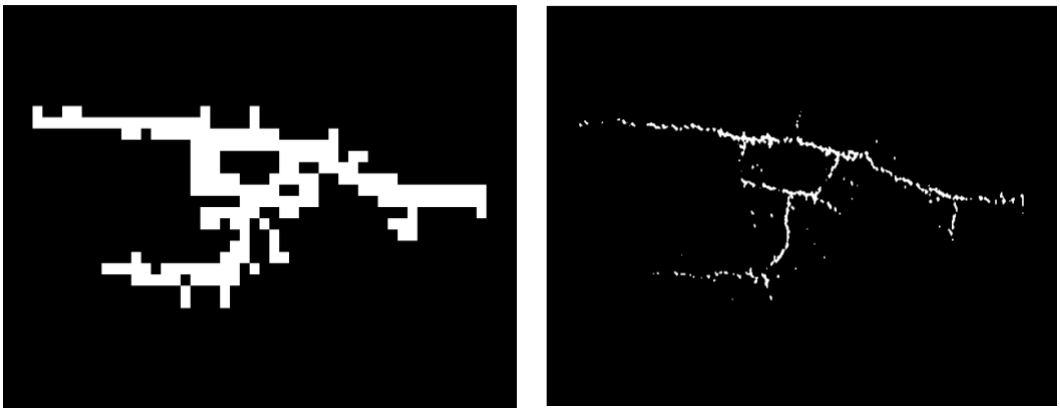


Figure 49 – Region filter: (Left) mask used to select crack areas, (Right) pixel-based crack detection results after applying the mask.

Then, candidate crack regions are found by grouping crack pixels using a region linking algorithm [8]. Relevant connected components are identified by fulfilling a set of three geometric requirements: (i) more than 70% of eccentricity for an ellipse fitted to it; (ii) ellipse major axis longer than 15 pixels; (iii) width of at least 1 mm. The remaining connected components (the non-relevant candidates) are considered for removal, unless they are spatially linked to relevant components, in which case the region linking algorithm keeps them labeled as cracks, thus improving the crack detection result. An example of the region linking module output is illustrated in Figure 50.

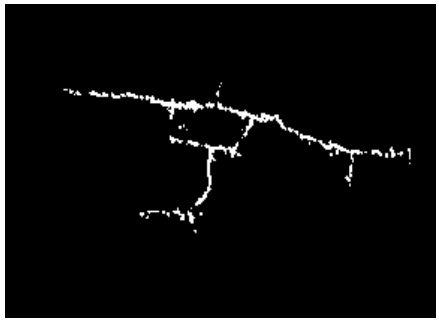


Figure 50 – Region linking results for the image of Figure 49 (Right).

4.3 Assessment Protocol

This section presents the strategy used to evaluate the LFCD system performance on detecting cracks, starting by briefly introducing a benchmarking system that explores the usage of 2D images. After that, a description of the dataset used, as well as the metrics considered to perform the evaluation procedure, are presented.

4.3.1 2D Benchmark Crack Detection – The Crack IT System

As a benchmark for comparing the performance of the proposed LFCD system, an existing 2D crack detection system, known as Crack IT, which fully incorporates the methodology described

in [8], was considered. This system was selected as it is a good representative of the state-of-the-art, and it has an implementation that is publicly available at http://amalia.img.lx.it.pt/CrackIT [36].

The architecture of the considered benchmarking solution is illustrated in Figure 51, and briefly described in the following.

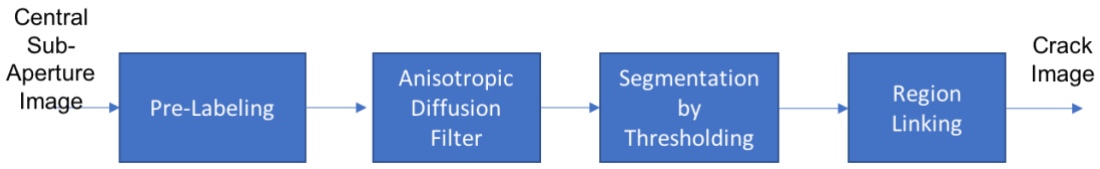


Figure 51 – CrackIT system architecture.

The Crack IT system uses a 2D pavement surface image as input and the processing starts by assigning a 'crack' or 'non-crack' label to each non-overlapping image block, based on the mean and standard deviation of pixel intensity values in each block – the pre-labelling, as detailed in [11]. After that, an anisotropic diffusion filter [8] is applied to deal with non-uniform background illumination and random textures present in pavement surface images, as typically observed in images captured by traditional 2D cameras. Then, a segmentation by thresholding [8] is applied to obtain the initial crack detection result. As a last step, the same region linking procedure mentioned in section 4.2 is applied to obtain the final crack detection result.

4.3.2 Evaluation Dataset

As mentioned before, the *Lytro Illum* lenslet light field camera was used to acquire the image dataset considered in this work. The acquisition setup considered the camera positioned at 1 meter from the pavement surface, with its optical axis perpendicular to the road pavement surface. The dataset for testing is composed of 13 light field images, captured from pavements with different surface texture characteristics. For each image, a multi-view array was computed and each of the recovered 2D sub-aperture images has a spatial resolution of 1 mm^2 per pixel.

To be able to compare the proposed system against the 2D benchmarking, the 2D central sub-aperture image (corresponding to the blue square in) is used as input to the CrackIT system, as it corresponds to what a traditional 2D camera would capture.

4.3.3 Evaluation Metrics

To evaluate crack detection results a block-based ground truth has been created with the help of an expert, who manually labeled each 25×25 pixel block of each image, with the help of a graphical user interface from Crack IT toolbox [36] developed for that purpose, as containing cracks or not.

Since for some blocks only a very small number of crack pixels were included, the expert sometimes labeled those blocks as not containing cracks. As a consequence, and in order to

have a fair evaluation of the automatic results produced by the algorithms, the block-based evaluation done in this dissertation assumes that a block will only be classified as containing a crack if the number of detected crack pixels in that block is at least 4% of the block size.

The block-based evaluation is summarized using the same well-known metrics used in the proposed system in previous Chapter, see section 3.2.3.

4.4 Performance Assessment

The evaluation of the proposed LFCD system is presented in the following, in comparison with the 2D system considered for benchmarking (Crack IT).

Table 1 and Figure 52 summarize the results achieved using the considered performance metrics for all the test images. From these results, it is possible to observe that the usage of light field imaging, even when only applying simple image processing techniques, seems to provide a better crack detection result than using the more conventional 2D crack detection system Crack IT – the *FM* metric results increase, in average, from 79% to 85%. Additionally, when doing a visual evaluation of the results (a subjective analysis), by looking at the identified cracks, it seems that by exploring the light field disparity information, it was possible to obtain a better definition of the cracks present in the pavement surface test images, as can be seen by the quality of the results shown in Figure 53.

Table 7 – Precision, recall and f-measure results for the LFCD and Crack IT systems.

Image	***re*** **(LFCD)**	***re*** **(CrackIT)**	***pr*** **(LFCD)**	***pr*** **(CrackIT)**	***Fm*** **(LFCD)**	***Fm*** **(CrackIT)**
1	100%	98%	100%	100%	100%	99%
2	100%	100%	100%	100%	100%	100%
3	93%	100%	97%	73%	95%	85%
4	86%	86%	93%	42%	89%	56%
5	95%	100%	75%	73%	84%	85%
6	96%	96%	74%	61%	84%	74%
7	91%	69%	77%	94%	83%	80%
8	75%	90%	94%	47%	83%	62%
9	71%	97%	91%	90%	80%	93%
10	70%	100%	93%	69%	80%	82%
11	82%	90%	74%	67%	78%	77%
12	81%	72%	72%	80%	77%	76%
13	80%	76%	61%	46%	69%	58%
Average	85%	90%	85%	72%	**85%**	**79%**

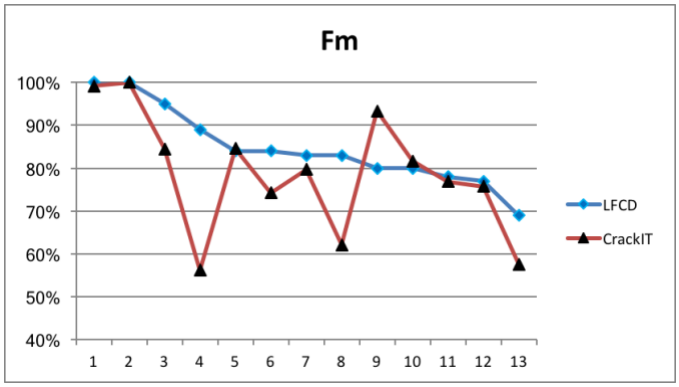


Figure 52 – Comparison of the LFCD and CrackIT systems' performance using the f-measure.

A comparison of the block detection sample results obtained from the proposed light field system and the CrackIT system is presented in Figure 53.

The blocks marked in green represent the ones correctly classified as containing cracks, the yellow ones representing the blocks where cracks were detected but are not marked in the ground truth, and the red ones corresponding to blocks not detected by the system as containing cracks, but belonging to crack according to the ground truth information.

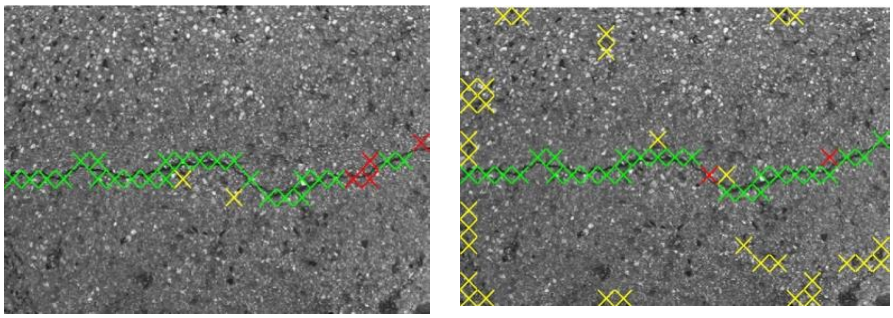


Figure 53 – Block detection results for image 4 of the dataset. (Left) proposed light field system, $re = 86\%, pr = 93\%$ *and* $Fm = 89\%$*. (Right) CrackIT system,* $re = 86\%, pr = 46\%$ *and* $Fm = 56\%$*.*

For the sample result presented in Figure 53, the proposed system showed a greater capacity to deal with the effects of the pavement surface texture, discarding most of the false positive detections present in the CrackIT results, while detecting most of the blocks containing cracks, hence achieving a higher value of the precision metric. In fact, the LFCD system seems to be more robust to deal with the different pavement surface textures found in images, as shown by the smaller variations of the *Fm* metric in Figure 54, thus outperforming the CrackIT system in this respect.

By looking at the precision metric results listed in Table 1, the LFCD system seems to perform significantly better for the entire dataset, meaning that the proposed system produces less false positives. Looking at the recall metric the behavior of the two systems is more similar, both being able to capture the more relevant crack information observable from the images.

Overall, the *Fm* metric (see Table 7 and Figure 52) shows the advantage of the proposed LFCD system.

5 Conclusions and Future work

This dissertation proposed a semi-automatic crack detection methodology. It considers a fast solution to automatically select a set of candidate crack regions, from each of which a skeleton and its endpoints are identified. Then an appropriate selection of minimal paths between endpoints is automatically applied to get a crack detection result. The proposed system then allows the user to use a simple GUI to quickly edit the results, correcting them as desired.
Compared to minimal path selection algorithm presented in [7], the proposal of this dissertation incorporates a new technique to quickly find the most relevant endpoints (using the *Preliminary Crack detection* and *Preliminary Crack Skeleton Computation* modules), saving time by more efficiently computing paths between endpoints and providing an option of manual editing the detection results at end, with the help of a GUI (*Manual Refinement* module). Notice that the algorithm used to compute minimal paths, instead of using Dijkstra's algorithm, uses a faster suboptimal version in the A^* algorithm, optimized for the crack detection problem.

Performance assessment using two datasets, captured using the LRIS system [10], each containing 83 images of size 4096x2048 pixels, has shown that the proposed method without manual refinement (KOMPS) can be used as fully automatic and unsupervised crack detection algorithm.

A comparison with MPS using pixel-based ground truth, using three images from Aigle-RN system dataset has shown that the proposed method KOMPS-MR (proposed method with Manual Refinement) affords the best F-measure for all three images. The comparison was made using a pixel-based ground truth since it is available at [39]. Without Manual Refinement, the proposed system provided similar results to MPS, see Table 4. The automatic detection fails sometimes due to the preliminary crack detection module, where the kernel applied to the image could not detect thin cracks. In those cases considering the *Manual Refinement* module allows correcting the results and with minimal interactions (see Table 6) the obtained results surpass the MPS ones.

In conclusion, the proposed method provides very robust and precise results in a wide range of situations, in a fully unsupervised manner, when considering KOMPS (proposed system without Manual Refinement). It can almost replicate the results of MPS in a much faster and efficient way, see Table 5, and with an advantage of easily allowing to correct the results using a simple GUI.

Using the developed GUI provides a quick way of creating a pixel-based ground truth, something that has been missing in the crack detection research community.

A perspective to a future work will be to develop an adaptive kernel to deal better with different image resolutions and give more reliable endpoints. Another approach is to replace the

Preliminary Crack Detection module with a trained convolutional neural network to identify with more accuracy crack regions in a pavement image.

The usage of novel imaging sensors to improve crack detection results has also been tested.
This dissertation presented a novel crack detection system, exploring for the first time a light field imaging sensor. Although the crack detection system presented has room for improvements, it allowed to show that exploring the disparity information available from the light fields can bring benefits for crack detection. In the experiment, it was noted that this disparity information is more abundant in the regions close to the center of the image.
The proposed LFCD system performed well when compared to the 2D-based CrackIT system, considered for benchmarking purposes. The experimental work has shown that the additional information available in light field images can be favorably explored for crack detection.
Future work on this system involves considering a more sophisticated image processing solution, like the techniques used in the previous presented semi-automatic, as well as the acquisition of a larger image dataset for testing.

6 References

[1] M. Gavilán *et al.*, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, 2011.

[2] G. Moussa and K. Hussain, "A New Technique for Automatic Detection and Parameters Estimation of Pavement Crack," *4th Int. Multi-Conference Eng. Technol. Innov. IMETI*, pp. 1–6, 2011.

[3] "JAE." Catálogo de Degradações dos Pavimentos Rodoviários Flexíveis, 1997.

[4] J. L. Byoung and L. Hosin, "A Robust Position Invariant Artificial Neural Network for Digital Pavement Crack Analysis," *Proc. TRB 2003 Annu. Meet.*, no. 319, pp. 1–41, 2002.

[5] Q. Zou, Q. Li, F. Zhang, Z. Xiong Qian Wang, and Q. Wang, "Path voting based pavement crack detection from laser range images," *Int. Conf. Digit. Signal Process. DSP*, pp. 432–436, 2017.

[6] "Lytro support page." [Online]. Available: https://support.lytro.com/hc/en-us. [Accessed: 14-Feb-2018].

[7] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2718–2729, 2016.

[8] H. Oliveira and P. L. Correia, "Automatic Crack Detectiom On Road Imagery using Anisotropic Diffusion and Region Linkage," *18th Eur. Signal Process. Conf.*, pp. 274–278, 2010.

[9] D. Fernandes, P. L. Correia, and H. Oliveira, "On the Suitability of Light Field Imaging for Road Surface Crack Detection," in *RECPAD 2017*, 2017.

[10] "Laser Road Imaging System." .

[11] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 155–168, 2013.

[12] H. Zakeri, F. M. Nejad, and A. Fahimifar, "Image Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review," *Arch. Comput. Methods Eng.*, no. 424, pp. 1–43, 2016.

[13] H. Oliveira and P. Correia, "Supervised crack detection and classification in images of road pavement flexible surfaces," *Recent Adv. Signal Process.*, 2009.

[14] S. Chambon and J. M. Moliard, "Automatic road pavement assessment with image processing: Review and comparison," *Int. J. Geophys.*, vol. 2011, pp. 1–20, 2011.

[15] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, 2012.

[16] P. Rosa and P. L. Correia, "Automatic Road Pavement Crack Detection using Boosting Classifiers," in *Automatic Road Pavement Crack Detection using Boosting Classifiers, Proc European Signal Processing Conf. - EUSIPCO*.

[17] N. Li, X. Hou, X. Yang, and Y. Dong, "Automation recognition of pavement surface distress based on support vector machine," *ICINIS 2009 - Proc. 2nd Int. Conf. Intell. Networks Intell. Syst.*, pp. 346–349, 2009.

[18] N. Tanaka and U. Kenji, "A Crack Detection Method in Road Surface Images Using Morphology,"

1998.

[19] J. Zhou, P. Huang, and F. Chiang, “Wavelet-Based Pavement,” pp. 89–98.

[20] H. D. Cheng, J.-R. Chen, C. Glazier, and Y. G. Hu, “Novel Approach To Pavement Cracking Detection Based on Fuzzy Set Theory,” *J. Comput. Civ. Eng.*, vol. 13, no. October, pp. 270–280, 1999.

[21] S. Chambon, P. Subirats, and J. Dumoulin, “Introduction of a Wavelet Transform based on 2d Matched Filter in a Markov Random Field for Fine Structure Extraction: Application on Road Crack Detection,” *Image Process. ing Mach. Vis. Appl. II*, 2009.

[22] S. Chambon, “Detection of Points of Interest for Geodesic Contours : Application on Road Images for Crack Detection,” *VISAPP 2011- Proc. Int. Conf. Comput. Vis. Theory Appl.*, pp. 210–213, 2011.

[23] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst. Man. Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[24] C. Sun and B. Appleton, “Multiple paths extraction in images using a constrained expanded trellis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1923–1933, 2005.

[25] T. Deschamps and L. D. Cohen, “Fast Extration of Minimal paths in 3D images and application to virtual endoscopy,” *Med. Image Anal.*, vol. 5, no. 4, pp. 281–99, 2001.

[26] L. D. Cohen and R. Kimmel, “Global minimum for active contour models: a minimal path approach,” *Proc. CVPR IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 24, no. 1, pp. 666–673, 1996.

[27] P. Yan and A. A. Kassim, “Medical image segmentation using minimal path deformable models with implicit shape priors,” *IIEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 4, pp. 677–684, 2006.

[28] V. Kaul, A. Yezzi, and Y. J. Tsai, “Detecting curves with unknown endpoints and arbitrary topology using minimal paths,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1952–1965, 2012.

[29] E. W. Dijkstra, “A Note on Two Probles in Connexion with Graphs,” *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[30] P. Hart, N. Nilsson, and R. Bertram, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths.,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 4, no. 2. pp. 100–107, 1968.

[31] M. Avila, S. Begot, F. Duculty, and T. S. Nguyen, “2D Image Based Road Pavement Crack Detection By calculating Minimal Paths and Dynamic Programming,” *2014 IEEE Int. Conf. Image Process.*, no. 1, pp. 783–787, 2014.

[32] “Results with AigleRN images.” [Online]. Available: https://www.irit.fr/~Sylvie.Chambon/AigleRN.html. [Accessed: 30-Mar-2018].

[33] V. Baltazart, J. Moliard, R. Amhaz, D. Wright, and M. Jethwa, “Some advances / results in monitoring road cracks from 2D pavement images within the scope of the collaborative FP7 TRIMM project,” in *presented at the EGU Meeting*, 2015.

[34] R. Amhaz and S. Chambon, “A new Minimal Path Selection Algorithm for Automatic Crack Detection on Pavement Images,” *Int. Conf. Image Process.*, pp. 788–792, 2014.

[35] W. Kaddah, M. Elbouz, Y. Ouerhani, V. Baltazart, M. Desthieux, and A. Alfalou, “Optimized minimal path selection (OMPS) method for automatic and unsupervised crack segmentation within two-dimensional pavement images,” 2018.

[36] H. Oliveira and P. L. Correia, “CrackIT - An image processing toolbox for crack detection and characterization,” *2014 IEEE Int. Conf. Image Process. ICIP 2014*, pp. 798–802, 2014.

[37] V. Baltazart, L. Yang, P. Nicolle, and J. M. Moliard, “Pseudo-ground truth data collection on pavement images,” *25th Eur. Signal Process. Conf. EUSIPCO 2017*, vol. 2017–Janua, pp. 2021–2025, 2017.

[38] Z. Guo and R. W. Hall, “Parallel thinning with two-subiteration algorithms,” *Comm. ACM*, vol. 32, pp. 359–373, 1989.

[39] “MPS Results with AigleRN images with ground truth.” [Online]. Available: https://www.irit.fr/~Sylvie.Chambon/AigleRN_GT.html. [Accessed: 04-Apr-2018].

[40] F. Pereira, E. A. B. da Silva, and G. Lafruit, “Plenoptic Imaging: Representation and Processing,” pp. 1–45, 2016.

[41] J. Shade, S. Gortler, L. He, and R. Szeliski, “Layered depth images,” *Proc. 25th Annu. Conf. Comput. Graph. Interact. Tech. - SIGGRAPH ’98*, pp. 231–242, 1998.

[42] A. Sepas Moghaddam, P. L. Correia, F. Pereira, I. De Telecomunicações, I. Superior, and T. Universidade, “Light Field Local Binary Patterns Description For Face Recognition.”