UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ITALO DE ÁVILA WIECZOREK**

# A POWER LINE DETECTION ALGORITHM TO SUPPORT A FINE GRAIN UAV MOVEMENT GUIDANCE

Porto Alegre
2017

**ITALO DE ÁVILA WIECZOREK**

# A POWER LINE DETECTION ALGORITHM TO SUPPORT A FINE GRAIN UAV MOVEMENT GUIDANCE

Dissertação de mestrado/Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica/Doutor em Engenharia Elétrica.

Área de concentração: Controle e Automação.

ORIENTADOR: Prof. Dr. Edison Pignaton de Freitas

CO-ORIENTADOR : Prof. Dr. Altamiro Amadeu Susin

Porto Alegre
2017

ITALO DE ÁVILA WIECZOREK’

# A POWER LINE DETECTION ALGORITHM TO SUPPORT A FINE GRAIN UAV MOVEMENT GUIDANCE

Esta dissertação/tese foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica/Doutor em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: ____________________________________

Prof. Dr. Edison Pignaton de Freitas, UFRGS

Doutor pela Universidade de Halmstad – Halmstad, Suécia e pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Manuel Menezes de Oliveira Neto, UFRGS

Doutor pela Universidade da Carolina do Norte – Chape Hill, Estados

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS

Doutor pela Universidade Federal de Minas Gerais – Belo Horizonte, Brasil

Prof. Dr. Dionísio Doering, UERGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE: _______________________________

Prof. Dr. Valner João Brusamarello

Porto Alegre, março de 2017.

# DEDICATÓRIA

Dedico este trabalho aos meus pais, Wolney Wieczorek e Ione Wieczorek, que sempre estiveram ao meu lado, me motivando e orientando com muito amor e carinho, sempre pelo melhor caminho. Também dedico este trabalaho a minha namorada, Nicole Correa,  que assim como eu, vivenciou sempre ao meu lado todos os tipos de sentimentos e angustias que o mestrado nos proporcionou durante estes dois anos. Por fim, dedico este trabalho a todos meus amigos, que tornaram minha caminhada até aqui muito tranquila, feliz e divertida. muito obrigado.

# AGRADECIMENTOS

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Ao meu orientador, Edison Pignaton de Freitas, que sem dúvida foi um dos professores mas dedicados, senão o mais dedicado, com quem já tive a oportunidade de trabalhar durante a minha trajetória acadêmica. Acredito que eu não tenha conseguido retribuir a altura tudo o que fizestes por mim, mas gostaria que o senhor soubesse que foi uma honra ter trabalhado junto com o senhor, e que serei eternamente grato por todas as oportunidades que me destes. Estarei sempre à disposição para ouví-lo e ajudá-lo no que for. Muito obrigado.

A SkyDrones, por todo o suporte dado com o fornecimento de equipamentos, auxílio técnico e laboratórios para a realização deste trabalho. Um agradecimento especial aos colegas de trabalho com que tive a sorte de compartilhar conhecimento e muitos momentos bons. São amizades que levarei para o resto da vida.

Aos colegas do PPGEE pelo seu auxílio nas tarefas desenvolvidas durante o curso.

# RESUMO

Detecção de linhas de alta tensão em ambientes complexos é uma das tarefas mais desafiadoras em inspeções que utilizam Veículos Aéreos Não Tripulados (VANTs). Este trabalho foca em dar uma solução para este desafio, através do desenvolvimento de um algoritmo de controle de voo de precisão, que guie o VANT de maneira autônoma sobre as linhas de alta tensão. O algoritmo proposto é baseado em quatro etapas: Captura da Imagem, Filtragem da Imagem, Detecção das Linhas e Controle de Voo. Inicialmente a imagem é redimensionada para um tamanho em que as linhas fiquem em maior evidência, depois uma sequência de filtros é aplicada na imagem para reduzir ruído e evidenciar ainda mais as linhas. Depois deste pré-tratamento, um filtro de duas dimensões com formato similar ao de uma linha de alta tensão é usado para extrair os pixels pertencentes as bordas destas linhas. Após a aplicação do filtro de duas dimensões, a Transformada de Hough é aplicada na imagem resultante para detectar os segmentos de reta. Por fim, todos os dados obtidos no processamento da imagem são utilizados para guiar o VANT de maneira autônoma pelas linhas de transmissão. O algoritmo proposto apresenta um eficiente sistema de detecção de linhas de alta tensão, para auxiliar o controle de voo autônomo de um VANT, apresentando resultados convincentes.

**Palavras-chave: Veículo Aéreo Não Tripulado. Processamento de Imagem. Controle de Movimento. Inspeção Autônoma.**

# ABSTRACT

Power lines detection in complex environments is one of the most important and challenging tasks in Unmanned Aerial Vehicles (UAV)-based inspections. This work focuses on tackling this challenge by developing a control algorithm to support fine grain UAV control to autonomously guide the aerial platform over the power lines. The proposed algorithm is based on four stages: Image Capturing, Image Filtering, Line Detection and Flight Control. Firstly, the image is cropped to a size that fits all the power lines, then a sequence of filters is applied in the image to reduce noise and highlight these lines. After all the image's pretreatment, a 2D filter with similar shape of a power transmission line is used to extract pixels that belongs to the line's edges. Then, the Hough Transform method detects the line segments in the edges result image. Lastly all the obtained data is used to autonomously guide a UAV over the power transmission lines. The proposed algorithm presents an efficient power transmission lines detecting system to support the autonomous UAV guidance, which presents convincing results.

**Keywords: Unmanned Aerial Vehicle. Image Processing. Movement Control. Autonomous Inspection.**

# SUMÁRIO

# LISTA DE ILUSTRAÇÕES

# LISTA DE ABREVIATURAS

2D: Two-dimensional space
BSD: Berkeley Software Distribution
CUDA: Computed Unified Device Architecture
DDR: Double Data Rate
EDLines: Edge Drawing Lines
ELM: Extreme Learning Machine
GPS: Global Positioning System
HD: High Definition
HT: Hough Transform
IP: Internet Protocol
IPM: Inverse Perspective Mapping
LBP: Local Binary Pattern
LS: Level-Sets
LSD: Line Segment Detector
LTS: Long Term Support
PC: Personal Computer
PID: Proportional Integral Derivative
RAM: Random Access Memory
RGB: Red, Green and Blue
ROI: Region of Interest
ROS: Robot Operating System
SIFT: Scale Invariant Feature Transform
TCP: Transmission Control Protocol
UAV: Unmanned Aerial Vehicle

# 1 RESUMO ESTENDIDO

Esta seção apresenta um breve resumo do trabalho descrito nas seções seguintes.

## 1.1 INTRODUÇÃO

Inspeção de redes de alta tensão é uma tarefa perigosa e cara para as empresas responsáveis pela distribuição de energia. O risco que estes procedimentos oferecem à vida humana são muito grandes, pois se trata de uma inspeção onde os técnicos envolvidos precisam ter bastante habilidade. Hoje em dia estas inspeções são feitas com o auxílio de helicópteros, que se aproximam dos fios de alta tensão, e se tornam uma espécie de plataforma para que os técnicos se fixem aos cabos e façam todos os procedimentos necessários.

Com a população dos Veículos Aéreos Não Tripulados (VANTs) muitas universidades e empresas começaram a investir seus recursos em pesquisas onde o foco era explorar o potencial deste equipamento para os mais diversos fins. Sabendo do alto nível de periculosidade, assim como os custos envolvidos, não demorou muito para que estudos começassem a serem desenvolvidos objetivando a inserção dos VANTs na inspeção das redes de alta tensão.

## 1.2 ESCOPO E DEFINIÇÃO DO PROBLEMA

Este trabalho tem como objetivo facilitar as inspeções de redes de alta tensão, com a utilização de VANTs auxiliando no procedimento de inspeção. Para isto um sistema que identifica linhas de alta tensão e fornece dados para o controle de voo é proposto neste trabalho.

Equipado com um computador embarcado capaz executar tarefas em tempo real e câmera de captura de vídeo de alta definição, o VANT deve sobrevoar as linhas de alta tensão, identificá-las através do módulo de processamento de imagem, e a partir da identificação das mesmas, extrair informações de posição para atuar no controle do voo.

Este trabalho, embora tenha apresentado uma certa robustez com relação a variação da luminosidade durante a captura de vídeo, não trata estes problemas em específico, assim como fatores climáticos que possam vir a atrapalhar o voo durante a missão. Também vale ressaltar que o controle de voo proposto neste trabalho se utiliza de controladores proporcionais integrais derivativos (PID) genéricos, e não efetua nenhum estudo aprofundado no intuito de extrair melhores resultados dos mesmos.

## 1.3 MÉTODO PROPOSTO

Para atingir o objetivo deste trabalho, o sistema foi dividido em dois módulos: módulo de processamento de imagem e módulo de controle de voo. Como o próprio nome já sugere, o módulo de processamento de imagem é responsável por todas as etapas que envolvem a captura e processamento das imagens. Já o módulo de controle de voo é responsável por extrair as informações de posicionamento das linhas de alta tensão identificadas nas imagens capturadas, e transformar estas informações em estímulos para o VANT.

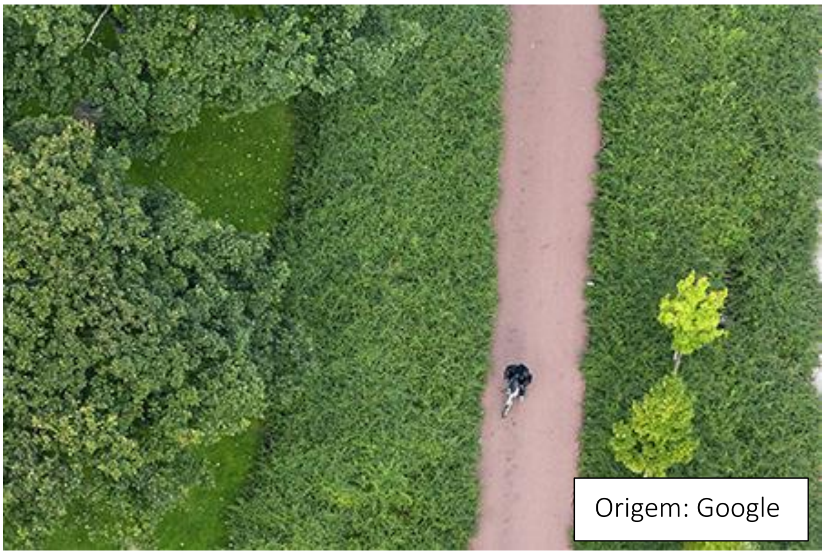
### 1.3.1 MÓDULO DE PROCESSAMENTO DE IMAGEM

O módulo de processamento de imagem apresenta as seguintes etapas: captura da imagem, filtragem da imagem e detecção das linhas de alta tensão na imagem. Estas estapas serão descritas nas seções a seguir.

#### 1.3.1.1 CAPTURA DA IMAGEM

A forma como a imagem é capturada tem papel fundamental no sucesso da execução do sistema. O ângulo de captura deve ser perpendicular ao solo, pois isto facilita a identificação das linhas de alta tensão já que as deixa paralelas umas às outras. A altura de voo acima das linhas também é importante ser definida. Nos testes efetuados para este sistema, alturas entre 6 a 8 metros acima das linhas obtiveram bons resultados.

A câmera usada neste trabalho foi a ZENMUSE X3 fornecida pela empresa DJI. Esta câmera vem acoplada a um gimbal, o qual facilita a obtenção do ângulo desejado e exclui a necessidade de obtê-lo computacionalmente, reduzindo, assim, custos computacionais. A Figura 1 mostra um exemplo de imagem capturada onde o angulo é perpendicular ao solo.



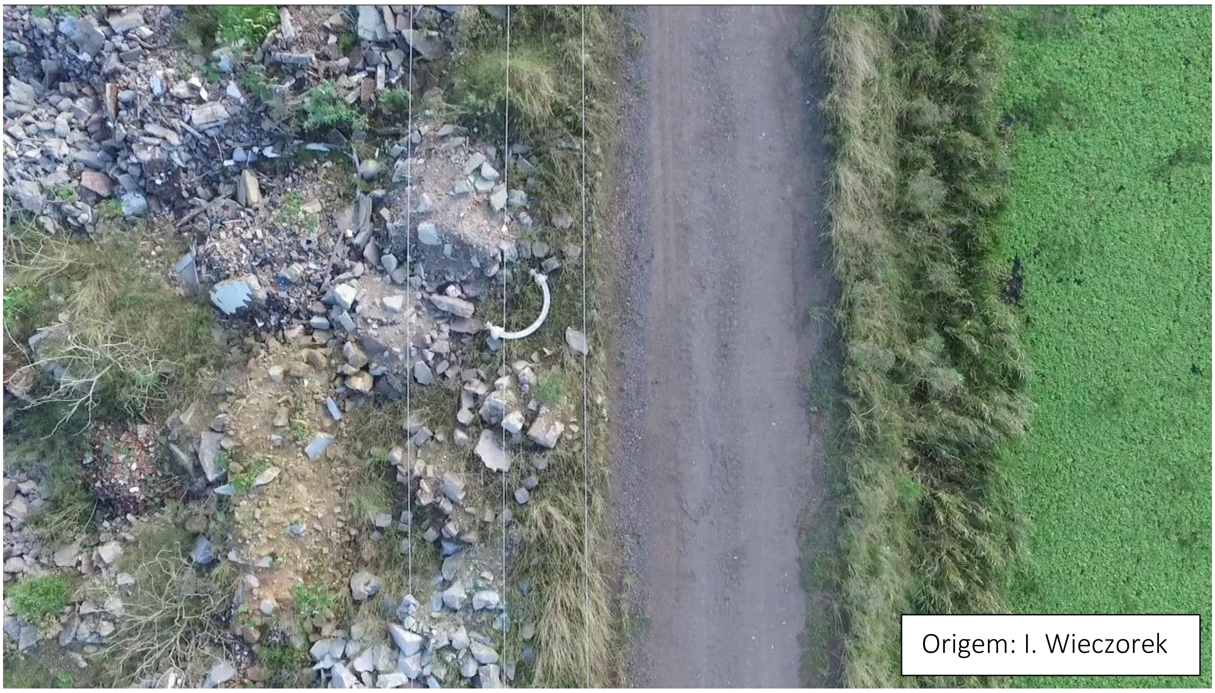
Origem: Google

Figura 1 Ângulo perperdicular ao solo.

### 1.3.1.2 Filtragem da Imagem

Para que a detecção das linhas de alta tensão seja efetuada com sucesso, alguns filtros são aplicados na imagem para segmentar as linhas do resto das informações presentes.

O primeiro passo é reduzir a resolução da imagem capturada. A resolução mínima de captura da ZENMUSE X3 é 1280x720 pixel, o que é uma resolução muito alta para processarmos em um computador embarcado. Porém se reduzirmos muito a resolução da imagem, podemos acabar perdendo informações importantes dos pixels das linhas e isto pode acarretar em uma maior dificuldade na detecção das mesmas. Então, através de uma série de experimentos, chegamos a conclusão que a resolução de 624x352 pixels seria o suficiente. A Figura 2 mostra a imagem redimensionada para a resolução proposta.



Origem: I. Wieczorek

**Figura 2 Imagem redimensionada.**

Após o redimensionamento da imagem, o próximo passo foi remover o máximo de informação que não é de interesse, possível. Para isso, a imagem redimensionada foi convoluída com um filtro 2D de kernel retangular, similar a um segmento de reta, que por sua vez é similar a um segmento de linha de alta tensão. O resultado da convolução deste filtro pode ser visto na Figura 3.

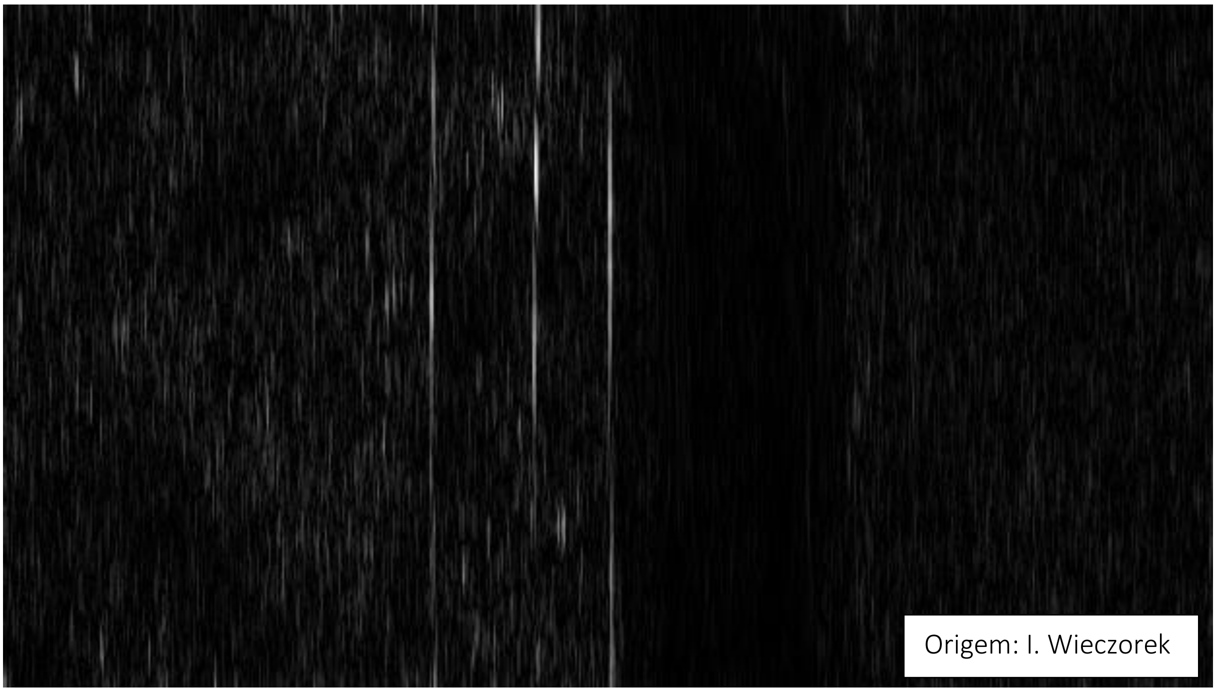
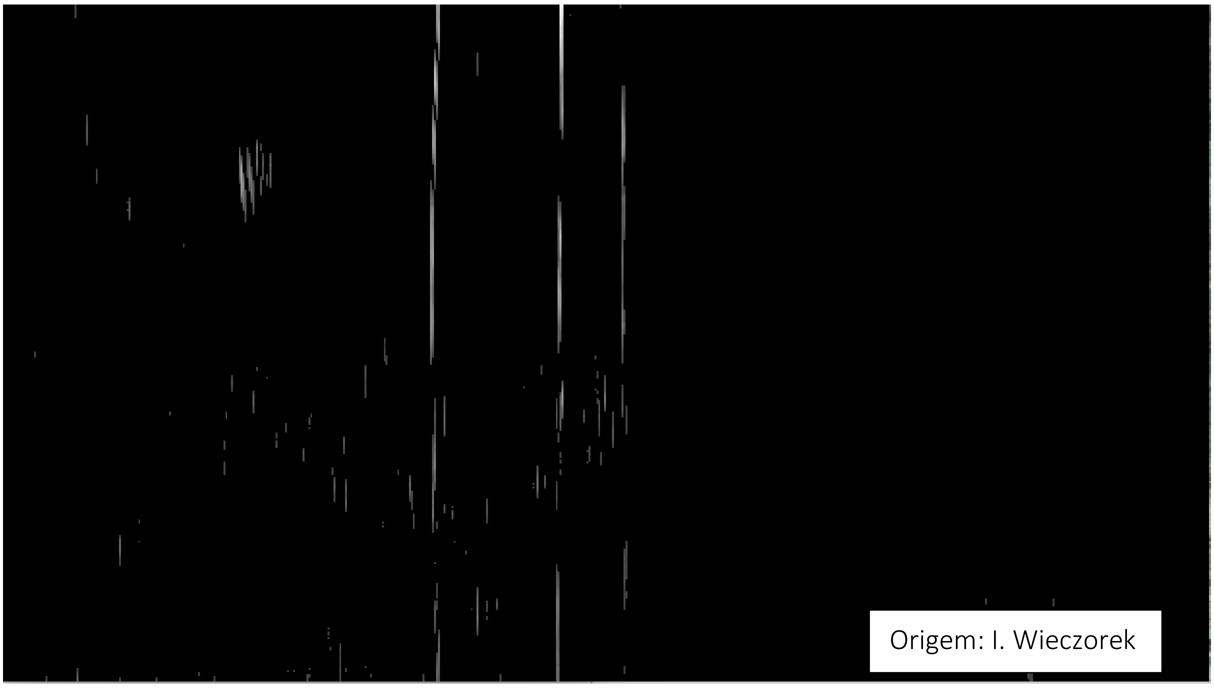


**Figura 3 Convolução com filtro 2D.**

A Figura 3 mostra o resultado da aplicação do filtro. Nela podemos observar que quase a totalidade de informações não relacionadas com as linhas de alta tensão foi removida. Além disso, os pixels pertencentes as linhas de alta tensão foram realçados, facilitando, assim, os próximos passos do método proposto.

Após a convolução da imagem com o kernel retangular, embora a quantidade de informação não relevante tenha sido reduzida, um filtro de threshold binário é aplicado. Este filtro segmenta os pixels das imagens de acordo com um valor dado, transformandos para 0 ou 1. O valor utilizado para este filtro foi encontrado empiricamente, através de tentativa e erro. O resultado da aplicação deste filtro pode ser visto na Figura 4.



Origem: I. Wieczorek

Figura 4 Threshold binário.

### 1.3.1.3 DETECÇÃO DAS LINHAS DE ALTA TENSÃO

A etapa de identificação das linhas de alta tensão utiliza a imagem resultante da filtragem e aplica a Transformada de Hough para a segmentação das linhas. Os primeiros frames processados são utilizados como uma espécie de calibração do processo. Nestes frames, após a declaração manual da quantidade de linhas que serão identificadas, o sistema efetua uma varredura em toda a imagem em busca das linhas. Após identificadas, o sistema calcula a posição das mesmas e cria regiões de interesse ao redor destas linhas. Estas regiões de interesse são utilizadas como máscaras para a imagem, ou seja, a Transformada de Hough passa a ser aplicada apenas no interior destas regiões, reduzindo assim custo computacional. Estas regiões de interesse são dinâmicas, ou seja, elas se adaptam conforme a variação da posição das linhas a cada frame processado.

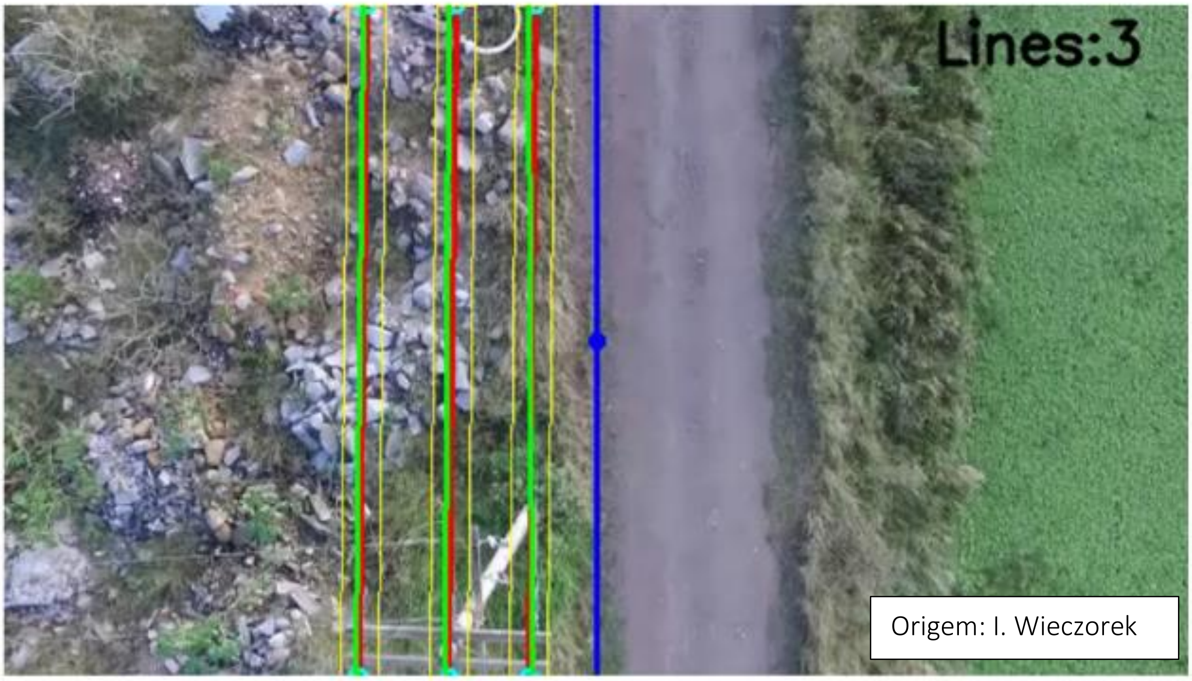


Figura 5 Detecção das linhas de alta tensão.

### 1.3.2 MÓDULO DE CONTROLE DE VOO

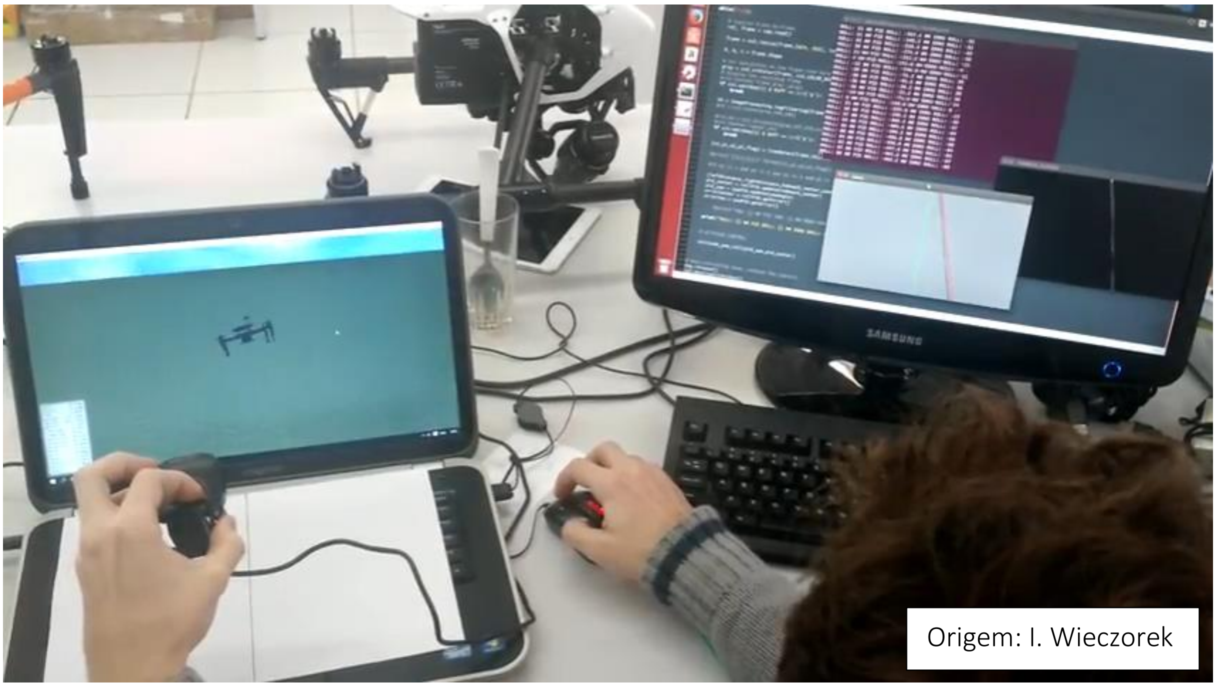
No módulo de controle de voo as informações de posição das linhas identificadas no módulo de processamento de imagem são extraídas e utilizadas como entrada de dados para os controladores PID, que controlarão os eixos Yaw e Pitch do VANT.

O VANT possui três eixos de voo, PITCH, ROLL e YAW. O eixo PITCH é o que define a velocidade do VANT e este foi definido como uma constante, neste caso 10m/s. O eixo ROLL é o que desloca o VANT para a esquerda e para a direita. O eixo YAW é o eixo que permite o VANT girar no seu próprio eixo. Como a velocidade é constante, os controladores atuarão apenas em dois eixos.

Conforme o VANT se desloca a referência das linhas de alta tensão varia, e de acordo com esta variação os controladores geram saídas que serão transformadas em estímulos na navegação do VANT. Toda esta comunicação é feita através de um sistema operacional robótico, chamado ROS. Este sistema operacional é responsável por toda a comunicação entre os periféricos envolvidos, como a câmera, a placa controladora de voo e os módulos de processamento de imagem e controle de voo. Isto tudo é possível devido a facilidade que o ROS nos traz, através da abstração de hardware fornecida pelo mesmo. Assim não é necessário perder tempo desenvolvendo drivers para os periféricos inseridos no sistema, basta apenas desenvolver uma arquitetura de troca de mensagens.

## 1.4 Experimentos e Resultados

A validação do sistema proposto se deu em laboratório. Os equipamentos utilizados foram: VANT Matrice 100 da DJI, computador embarcado Manifold com 2.3GHz de processamento e 2GB DDR3 de RAM também fornecido pela DJI, uma webcam para simular a variação de posicionamento das linhas em uma folha de papel e um computador com o simulador da DJI para experimentos com a utilização de VANTS.



Origem: I. Wieczorek

Figura 6 Cenário de teste em laboratório.

O experimento consistiu em analisar os tempos de resposta do VANT de acordo com a variação de posição da linha de referência. O tempo de resposta de um laço do systema alcançou 150ms, o que foi muito satisfatório devido a complexidade do sistema. O sistema foi capaz de gerar estímulos nos dois eixos controlados, ROLL e YAW.

## 1.5 CONCLUSÕES

O sistema alcançou o objetivo esperado, que era identificar as linhas e gerar estímulos de controle de navegação. O sistema de processamento de imagem se mostrou bastante eficaz na identificação e segmentação de linhas em estudos de caso reais, com ambientes bastante ruidosos e de difícil segmentação. O fluxo sistêmico de geração se estímulos para o controle de voo se mostrou viável, embora o controle de voo não tenha sido explorado e estudado afundo.

## 2 INTRODUCTION

High voltage transmission lines inspection is a dangerous and expensive task done periodically by companies in charge of power distribution. These inspections have the objective of search for anything that could potentially cause a problem in the power distribution, as cracked insulators and frayed wires, for example. Very often, the transmission lines crosses areas of difficult access, demanding high skilled professionals and costly equipment, which makes the inspection slow and expensive. Figure 7 presents an example of a cracked insulator as an example of a problem that is searched in these inspections.

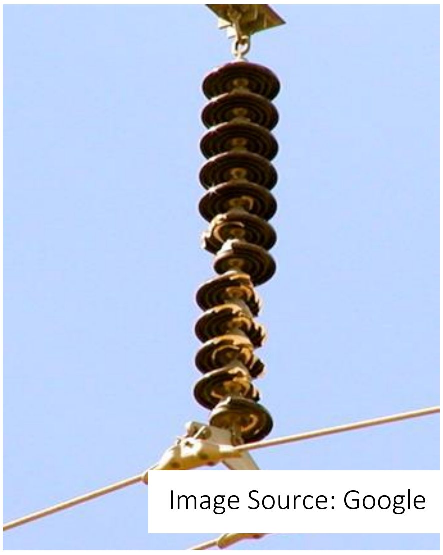


Image Source: Google

Figure 7 Cracked insulator.

With the popularization of the UAVs (Unmanned Aerial Vehicles), many applications that take advantage of their facilities, such as low cost, easy access to remote areas, remote sensing and easy maintenance, began to emerge. Following this trend, researches aiming the optimization of the high voltage transmission lines inspection using UAVs started. The main benefit of the usage of UAVs to do this task, besides the cost reduction, is the human life risk reduction, since this profession is considered a profession of high risk. Besides the inherent risk of fly with manned helicopters close to power transmission lines, some types of problems may increase the risk of accidents, such as frayed wires that are about to rip what may occur during the inspection. Figure 8 presents an example of frayed wire. Figure 9 presents an example of a conventional way to perform transmission power line inspection with a manned helicopter and a human inspector close to the lines.

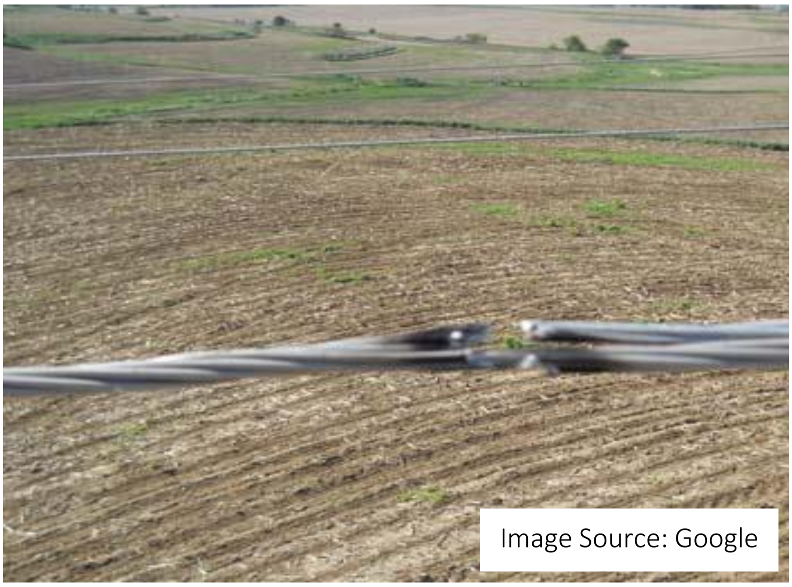


Image Source: Google

**Figure 8 Frayed wire.**

The challenges on this kind of application are basically the detection and segmentation of the transmission power lines and the autonomous control of the UAV. Pattern recognition in images has been a live research topic in computer vision, and there are still many challenges in real applications that applies this technique. This kind of application requires a robust and reliable algorithm to detect the transmission lines and towers, and at the same time avoid colliding with these structures, ensuring a safe flight. However, such application needs to fit in embedded hardware carried by the UAVs that although nowadays possess a substantial processing power, they are still considered as a limiting factor (HULENS, 2015).

Pattern recognition in images has been an important research topic in computer vision, and there are still many challenges in real applications that applies this technique (HUANG, 2014) of the type of application considered in this work requires a robust and reliable algorithm to detect the transmission lines and towers, and at the same time, avoid colliding with these structures, ensuring a safe flight.

Figure 9 Power line inspection being doing by a man.

Knowing that the aerial images captured by the UAV's camera have complex backgrounds, the illumination can vary according to UAV's movement and the camera's quality determines how good the captured image is, a solution that considers all these factors is a must. The work presented in (ZHANG, 2010) points out that the traditional edge detection algorithms cannot achieve ideal effect in the extraction of power lines, and the method of simple randomized Hough Transform to extract the power lines presents missed detections and wrong detections. The work presented in (SUN, 2010) reports a conducted a study on the Sobel and Canny edge detection operator for image detection of transmission lines and it shows that, using edge operator for images in the complex background is still insufficient due to the difficulty of parameterization. In (LI, 2010) it is reported the development of an image processing system that automates conductor localization and spacer detection in order to reduce the work required in visual inspection. In (KOSHELEV, 2015) it is reported the proposal of a method that allows detecting transmission lines and estimating their filter parameters without any human intervention, but their method uses a priory data as power line towers position and UAV's position. By the restrictions and limitations reported in these works found in the literature review, it is clear that a complete solution for automated UAV movement control assisted by embedded image processing of power transmission lines detection is still an open research topic.

## 2.1 Scope and Contribution

The objective of this work is to develop a system that can identify transmission power lines in captured images by a UAV during its flight and, based on the power lines position, generate data to assist its flight while in mission. This system intends to assist the inspections that are done nowadays with a safer and cheaper solution.

The first alternative that could be considered to trace a route to a UAV inspect power lines would be using the GPS coordinates of the poles that support the transmission power lines. These coordinates would guide the UAV through pole to pole, while it flies over the transmission power lines, until it finishes its mission. The problem is the precision of the GPS coordinates reference. It is known that the GPS coordinates have a margin of error that can reach meters away, and this lack of precision may make the UAV not fly over the transmission power lines at all. Thus, observing this problem, the necessity of a fine grain algorithm that does these flight adjustments was identified.

The power lines detection in images acquired by UAVs is a complex problem. Due to the large number of variables in the considered image analysis problem, this number needs to be delimited to restrict the bounders of a feasible solution. The first characteristic to be considered is the fact that the aerial image have to show the power lines from the top, collinear with the motion shown in the acquired video. To begin the process of line detection, the first image frame has to contain the number of power lines to be detected, for example, let's consider segments with three parallel lines. Then, the algorithm detects these lines and follow through the next frames. The illumination can interfere, because this kind of image always use natural environmental light, so this problem need to be treated by the algorithm.

The focus of this work is to propose an approach for power line detection, and then, using the obtained data, do an autonomous flight with a rotary wing UAV over the detected lines. In this case, features like distances between the lines and other objects, detection of defects in the power transmission lines structures, like towers, transformers, and the power lines themselves are not treated by the method. As can be seen on the work presented in (SAMPEDRO, 2014), changes in the background are another common source of problems that the vision systems have to deal with. In cases in which these background changes occur, there is a big change of the image processing system parametrization, so the algorithm needs to be robust against environment changes.

Another factor that interferes in the execution of the method is the image's quality. Considering quality of the images, there are many factors like focus, stabilization and their size. The last cited factor has a direct impact in the processing cost, in view of implementation on an embedded system, this parameter must to be seriously taken into account. The other parameters are treated by the hardware to stabilize the image and with the software which controls the camera system to maintain the focus.

Currently on the area of automated power line detection using UAVs, there are no solution that fully addresses the problem, thus there is a wide area for research. In this work, the focus is to propose a solution to detect the power lines and to do an autonomous flight using an efficient and reliable algorithm considering the boundary conditions and the scope delimited in this section. Thus, the contribution provided by this work is a solution that composes the embedded image processing for the power lines detection and its integration in the UAV movement control to support its autonomous flight over the power transmission lines.

### 2.2 WORK ORGANIZATION

This work is organized as follows: Section 2 presents the background concepts and Section 3 discusses the related works. Section 4 presents the problem statement, presenting the details that delimit the scope of this work, while Section 5 describes and provides details about the proposed method to the addressed problem. Section 6 presents the performed experiments and discusses the obtained results, while Section 7 concludes the work providing directions for future work.

# 3 BACKGROUND CONCEPTS REVIEW

## 3.1 DIGITAL LINE DETECTION

Power lines detection constitutes an important part of this work. Digital camera based methods consider these power lines as digital lines and edges which are desired to be detected using line and edge detectors. EDLines, LSD and Hough Transform are the best line detection methods that are known in the literature. (YETGIN, 2015).

EDLines, or Edge Drawing Lines, is an edge detection algorithm that instead of give out a binary edge image as output, as Canny method does, where the detected edge pixels are usually independent, disjoint, discontinuous entities, it produces a set of edge segments, which are clean, contiguous, i.e., connected, chains of edge pixels. Thus, while the outputs of other edge detectors require further processing to generate potential object boundaries, which may not even be possible or result in inaccuracies; EDLines not only produces perfectly connected object boundaries by default, but it also achieves this in blazing speed compared to other edge detectors (AKINLAR, 2011).

Given a grayscale image, ED performs edge detection in four steps:

(a) The image is first passed through a filter, e.g., Gauss, to suppress noise and smooth out the image. A 5 × 5 Gaussian kernel with $\sigma = 1$ is used by default.

(b) The next step is to compute the gradient magnitude and direction at each pixel of the smoothed image. Any of the known gradient operators, e.g., Prewitt, Sobel, Scharr, etc., can be used at this step.

(c) In the third step, a set of pixels is computed, called the anchors, which are pixels with a very high probability of being edge elements. The anchors correspond to pixels where the gradient operator produces maximal values, i.e., the peaks of the gradient map.

(d) Finally, the anchors computed in the third step are connected by drawing edges between them; hence the name Edge Drawing (ED). The whole process is similar to children's boundary completion puzzles, where a child is given a dotted boundary of an object, and s/he is asked to complete the boundary by connecting the dots. Starting from an anchor (dot), ED makes use of the neighboring pixels' gradient magnitudes and directions, and walks to the next anchor by going over the gradient maximas. If you visualize the gradient map as a mountain in 3D, this is very much like walking over the mountain top from one peak to the other.

Figure 10 presents the four steps described.

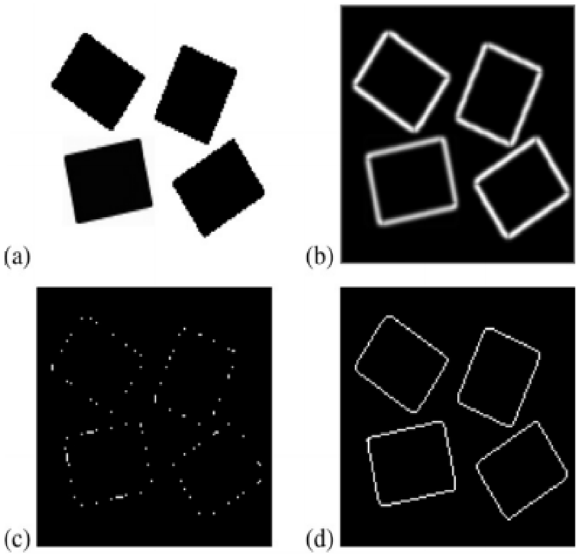


Image Source: AKINLAR

**Figure 10 (a) A grayscale image containing four rectangles, (b) gradient map, (c) anchor points, (d) final edge map.**

Given an edge segment comprised of a contiguous chain of edge pixels, the goal of this step is to split this chain into one or more straight line segments. The basic idea is to walk over the pixels in sequence, and fit lines to the pixels using the Least Squares Line Fitting Method until the error exceeds a certain threshold, e.g., 1-pixel error. When the error exceeds this threshold, we generate a new line segment. The algorithm then recursively processes the remaining pixels of the chain until all pixels are processed.

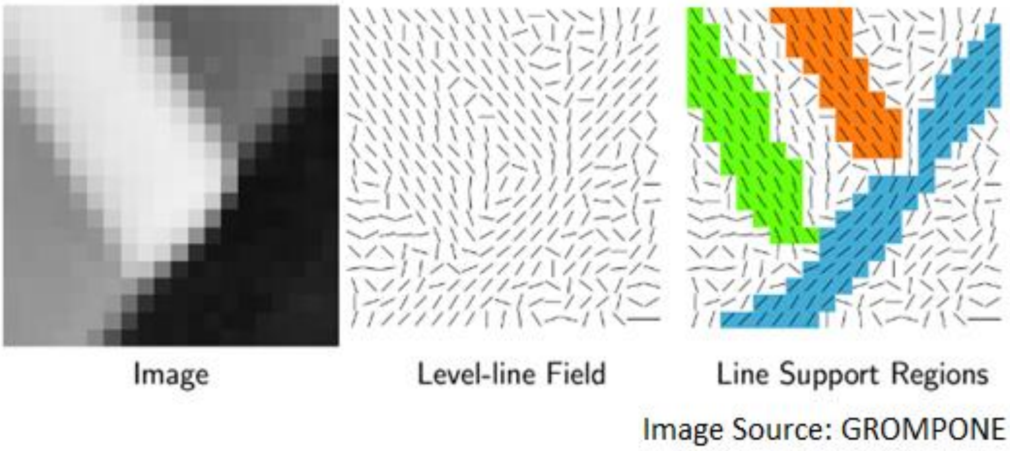


Image Source: GROMPONE

Figure 11 Line support regions.

The LSD method is aimed at detecting locally straight contours on images which are called line segments. Contours are zones of the image where the gray level is changing fast enough from dark to light or the opposite. Thus, the gradient and level-lines of the image are key concepts and are illustrated in Figure 11.

Each line support region (a set of pixels) is a candidate for a line segment. The corresponding geometrical object (a rectangle in this case) must be associated with it. The principal inertial axis of the line support region is used as main rectangle direction; the size of the rectangle is chosen to cover the full region, as in Figure 12.

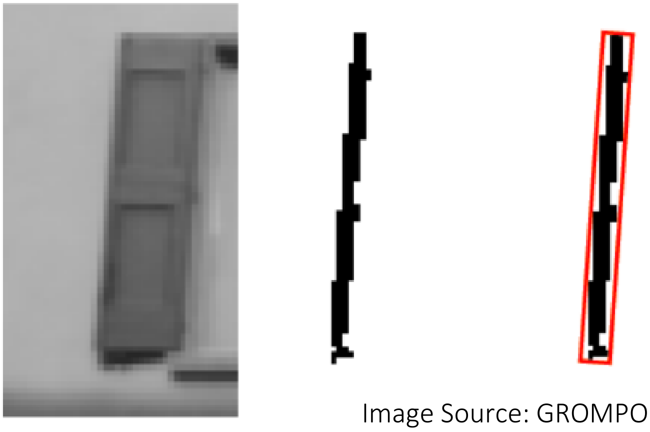


Image Source: GROMPONE

**Figure 12 Rectangle approximation of line support region.**

Each rectangle is subject to a validation procedure. The pixels in the rectangle whose level-line angle corresponds to the angle of the rectangle up to a tolerance τ are called aligned points, see Figure 13.

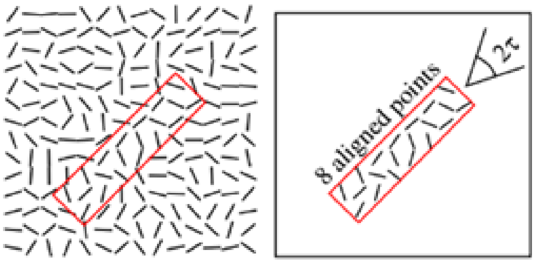


Image Source: GROMPONE

**Figure 13 Aligned points.**

The total number of pixels in the rectangle, n, and its number of aligned points, k, are counted and used to validate or not the rectangle as a detected line segment.

The validation step is based on the *a contrario* approach and the Helmholtz principle proposed by (DEOLNEUX, 2015). The so-called Helmholtz principle states that no perception (or detection) should be produced on an image of noise. Accordingly, the *a contrario* approach proposes to define a *noise* or *a contrario* model $H_0$ where the desired structure is not present. Then, an event is validated if the expected number of events as good as the observed one is small on the *a contrario* model. In other words, structured events are defined as being rare in the *a contrario* model.

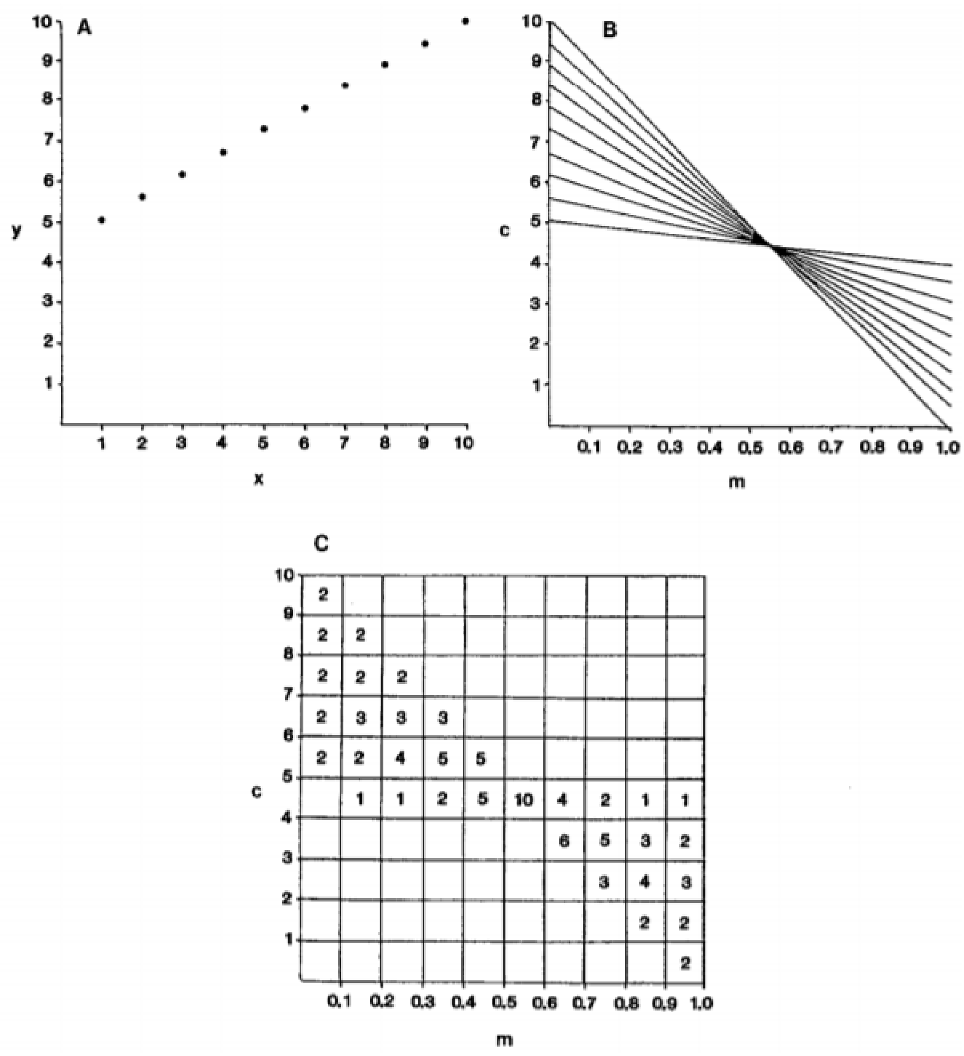
In the case of *line segments*, the number of aligned points is the main concern, so the event that a *line segment* in the *a contrario* model has as many or more aligned points is considered as in the observed *line segment* (GROMPONE, 2010).

The Hough Transform (HT), was originally introduced as a method of detecting complex patterns of points in binary image data by determining specific values of parameters which characterize these patterns. Spatially extended patterns are transformed so that they produce spatially compact features in a space of possible parameter values. This method converts a complex global detection problem in image space into a more easily solved local peak detection problem in a parameter space. The method can be illustrated by considering identifying sets of collinear points in an image. A set of image points *(x, y)* which lie on a straight line can be defined by a relation, *f*, such that

$$f\big((\hat{m},\hat{c}),(x,y)\big) = y - \hat{m}x - \hat{c} = 0 \quad (1)$$

where *m* and *c* are two parameters, the slope and intercept, which characterize the line. Equation (1) maps each value of the parameter combination $(\hat{m},\hat{c})$ to a set of image points. The hat symbol denotes quantities in the domain of the mapping. The mapping is one to many from the space of possible parameter values to the space of image points. The HT uses the idea that Eq. (1) can be viewed as a mutual constraint between image points and parameter points and therefore it can be interpreted as defining a one to many mapping from an image point to a set of possible parameter values. This corresponds to calculating the parameters of all straight lines which belong to the set that pass through a given image point $(\hat{x},\hat{y})$. This operation is called backprojection of the image point and the defining relation which achieves this is given by:

$$\boldsymbol{g}\big((\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}),(\boldsymbol{m},\boldsymbol{c})\big) = \hat{\boldsymbol{y}} - \hat{\boldsymbol{x}}\boldsymbol{m} - \ \boldsymbol{c} \ = \ \mathbf{0} \ (2)$$



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | | | | | | | | | |
| 2 | 2 | | | | | | | | |
| 2 | 2 | 2 | | | | | | | |
| 2 | 3 | 3 | 3 | | | | | | |
| 2 | 2 | 4 | 5 | 5 | | | | | |
| | 1 | 1 | 2 | 5 | 10 | 4 | 2 | 1 | 1 |
| | | | | | | 6 | 5 | 3 | 2 |
| | | | | | | | 3 | 4 | 3 |
| | | | | | | | | 2 | 2 |
| | | | | | | | | | 2 |

Image Source: ILLINGWORTH

**Figure 14 The basis of the Hough transform for line detection: (A) *(x, y)* point image space; (B) *(m, c)* parameter space; (C) accumulator space corresponding to (B).**

In the case of a straight line each image point $(\hat{x}, \hat{y})$ backprojects or defines a straight line in *(m, c)* parameter space. Figure 14a is a typical point image and Figure 14b shows the parameter lines produced by backprojecting image points into parameter space using Equation 2. Points which are collinear in the image space all intersect at a common point in the parameter space and the coordinates of this parameter point characterizes the straight line connecting the image points. The HT identifies these points of intersection in parameter space. Determination of the point of intersection in parameter space is a local operation and should be considerably easier than detecting extended point patterns in image space (ILLINGWORTH, 1988).

## 3.2 ROBOT OPERATING SYSTEM

ROS (Robot Operating System) is a BSD-licensed system for controlling robotic components from a PC. ROS includes libraries and tools that helps in development of applications in robotic area, including abstraction of hardware, drivers, specific libraries, viewers, message transmission, package management and many others functionalities. The ROS system was developed with the objective of promote collaboration between robotic developers, allowing, for example, that different research groups could collaborate with each other exchanging experiences and reusing code.

ROS is one of the most utilized operating systems available to robots (KEER, 2012) with works being developed both in industrial environment (ROBOT OPERATING SYSTEM, 2017) and academic environment like courses that teaches basic robotics concepts using ROS as one of the components of a virtual laboratory (CORRELL, 2013).

This section will describe the ROS Packages, ROS Stacks, ROS Nodes and the communication mechanism between the ROS Nodes, as shown in Figure 15. These are the basic concepts of the ROS architecture and will helps the contextualization of the ROS system in the proposed system in this work.

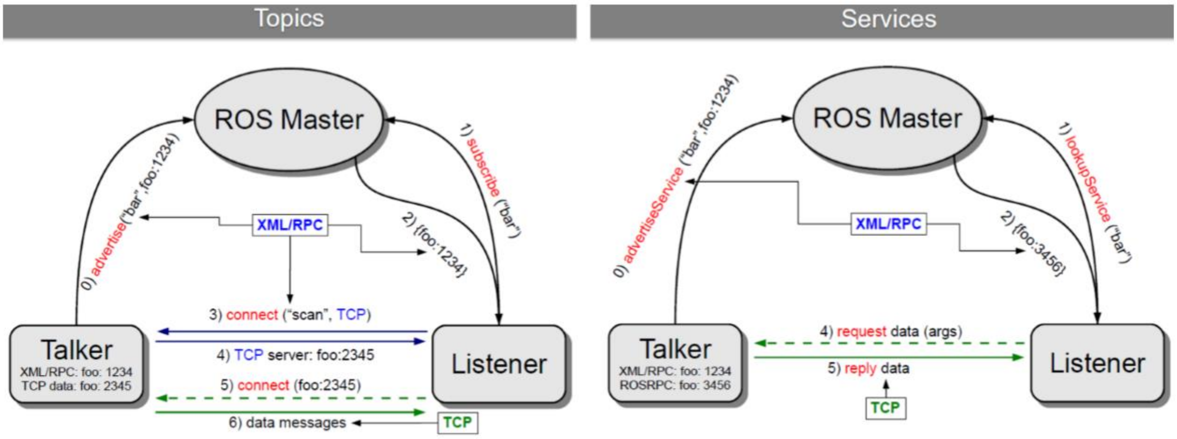


Image Source: Robot Operating System

Figure 15 Topics and Services in ROS.

### 3.2.1 ROS Packages

Software in ROS is organized in packages. A package might contain ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module. The goal of these packages it to provide this useful functionality in an easy-to-consume manner so that software can be easily reused. In general, ROS packages follow a "Goldilocks" principle: enough functionality to be useful, but not too much that the package is heavyweight and difficult to use from other software (ROBOT OPERATING SYSTEM, 2017). The structure of a ROS package is presented in Figure 16.

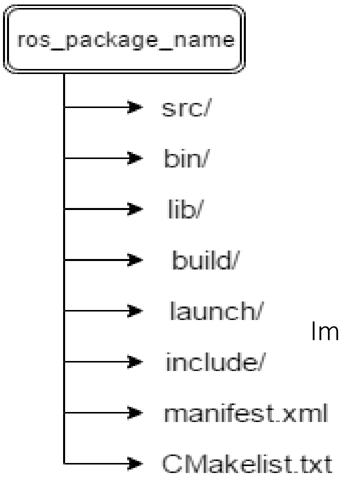


Image Source: I. Wieczorek

Figure 16 Structure of a ROS package.

Figure 16 describes the ROS package structure. Each package has a particular name that usually makes reference to its functionality. The *src/* folder stores all the source codes of the application. The *bin/* folder stores the generated executable files. The *lib/* folder stores all the shared libraries. The *build/* folder stores all the compiled files. The *launch/* folder stores all the package initialization files. The *include/* stores all the header files. The *manifest.xml* file contains basics information about the package. The *CMakelist.xml* file contains definitions about the package compilation.

### 3.2.2 ROS STACKS

Packages in ROS are organized into ROS stacks. Whereas the goal of packages is to create minimal collections of code for easy reuse, the goal of stacks is to simplify the process of code sharing. Stacks are the primary mechanism in ROS for distributing software. Each stack has an associated version and can declare dependencies on other stacks. These dependencies also declare a version number, which provides greater stability in development (ROBOT OPERATING SYSTEM, 2017).

Stacks are the basic unit of releasing ROS code. They usually collect together thematically similar Packages. Ultimately Stacks are meant to bundle together code that is developed together and is mutually interdependent. For example, the navigation stack consists of several planner packages, a high-level ROS node, a localization package, and obstacle data structures.

Stacks collect packages that collectively provide functionality, such as a navigation stack or a manipulation stack. Unlike a traditional software library that the developer can link against at compile time, these stacks can also provide this functionality at runtime via ROS topics and services.

### 3.2.3 ROS NODES

A node is a process that performs computation. Nodes are combined together into a graph and communicate with one another using streaming topics, RPC services, and the Parameter Server. These nodes are meant to operate at a fine-grained scale; a robot control system will usually comprise many nodes. For example, one node controls a laser range-finder, one Node controls the robot's wheel motors, one node performs localization, one node performs path planning, one node provides a graphical view of the system, and so on (ROBOT OPERATING SYSTEM, 2017).

The use of nodes in ROS provides several benefits to the overall system since is possible to isolate errors to individual nodes. Another benefit is the code complexity reduction when compared to monolithic systems. The implementation details of the nodes are also well hidden, since they provide an API (Application Programming Interface) simplified that allows alternative implementations even in other programming languages.

Figure 17 shows an example of two nodes sending (talker) and receiving (listener) data. This node connection is made through a topic, which publishes messages to all subscribed nodes

Image Source: I. Wieczorek

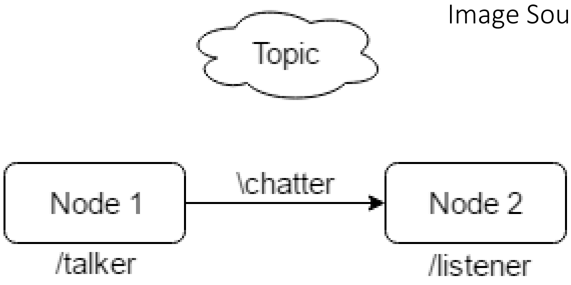


**Figure 17 ROS node example.**

### 3.2.4 COMMUNICATION MECHANISMS BETWEEN ROS NODES

To provide communication between the nodes, ROS make use of three mechanisms:

(a) *Topics:* Topics are named buses over which nodes exchange messages. Topics have anonymous publish/subscribe semantics, which decouples the production of information from its consumption. In general, nodes are not aware of who they are communicating with. Instead, nodes that are interested in data *subscribe* to the relevant topic; nodes that generate data *publish* to the relevant topic. There can be multiple publishers and subscribers to a topic (ROBOT OPERATING SYSTEM, 2017).

Topics are intended for unidirectional, streaming communication. Nodes that need to perform remote procedure calls, i.e. receive a response to a request, should use services instead. Figure 18 represents a ROS topic with its publishers and subscribers.

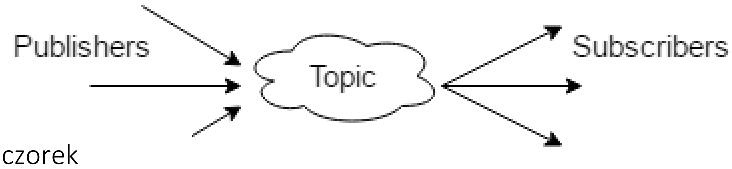


Image Source: I. Wieczorek

**Figure 18 ROS topic example.**

(b) *Services:* The publish/subscribe model is a very flexible communication paradigm, but its many-to-many one-way transport is not appropriate for RPC request/reply interactions, which are often required in a distributed system. Request/reply is done via a Service, which is defined by a pair of messages: one for the request and one for the reply. A providing ROS node offers a service under a string name, and a client calls the service by sending the request message and awaiting the reply. Client libraries usually present this interaction to the programmer as if it were a remote procedure call (ROBOT OPERATING SYSTEM, 2017). Figure 19 represents a ROS service.

ROS Services also can make *persistent connection* to a service, allowing high performance at the cost of less robustness to service provider change.

Image Source: I. Wieczorek

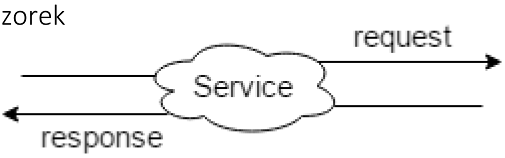


**Figure 19 ROS service example.**

(c) *Action Servers:* In any large ROS bases system, there are situations in which someone would like to send requests to a node to perform some task and also receive a reply to the request. This scenario is very common and can be achieved via ROS Services described above. But in some cases, if the service takes a long time to execute, the user might want the ability to cancel the request during execution or get periodic feedback about how the request is progressing. The *Action Server* mechanism provides tools to create servers that execute long-running goals that can be preempted (ROBOT OPERATING SYSTEM, 2017). It also provides a client interface in order to send requests to the server.

# 4 RELATED WORKS

(YUAN, 2015) have proposed a new application to UAVs for environment surveillance. They worked on a UAV based forest fire detection and tracking method. The basic idea of the proposed method is to adopt the channel "a" in Lab color model to extract fire-pixels by making use of chromatic features of fire.

In order to achieve the main objective of their work, the paper conducts a preliminary research on developing a set of image processing algorithms that are capable of effectively detecting and tracking forest fire. Through laboratory experiments analyzing fire segmentation results of using different color spaces, they found that the "channel a" of Lab color model segmentation, which usually displays the reddish color, has the best results of fire segmentation.

The test scenarios were basically two: real forest fire images and real-time fire images. The experimental results proved that the proposed algorithm can not only test with real forest fire images, but also perform well with the images collected by a UAV in lab environment.

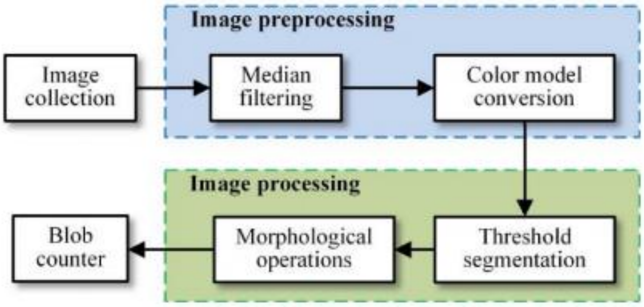


Image Source: YUAN

**Figure 20 Flowchart of fire detection and tracking algorithms.**

(YANG, 2016) proposed an edge-based moving object tracking algorithm for an embedded platform. Their work has the objective of tracking objects in real time, processing images sequences of 1280 x 720 resolution. In the proposed method, an adaptive local edge detection method is employed to extract the feature pixels of a tracked object. To improve the effectiveness of the proposed method, a region-based local binary pattern feature was employed to describe the edge pixels of the tracked object. Then, the method was implemented in an embedded platform aiming real time execution for experimental testing in complex environments.

The experimental results of this work demonstrated that the proposed system can handle various complex situations, including scene illumination changes, object deformation and partial occlusion reaching at least 30 frames per second for a HD resolution on an embedded platform.

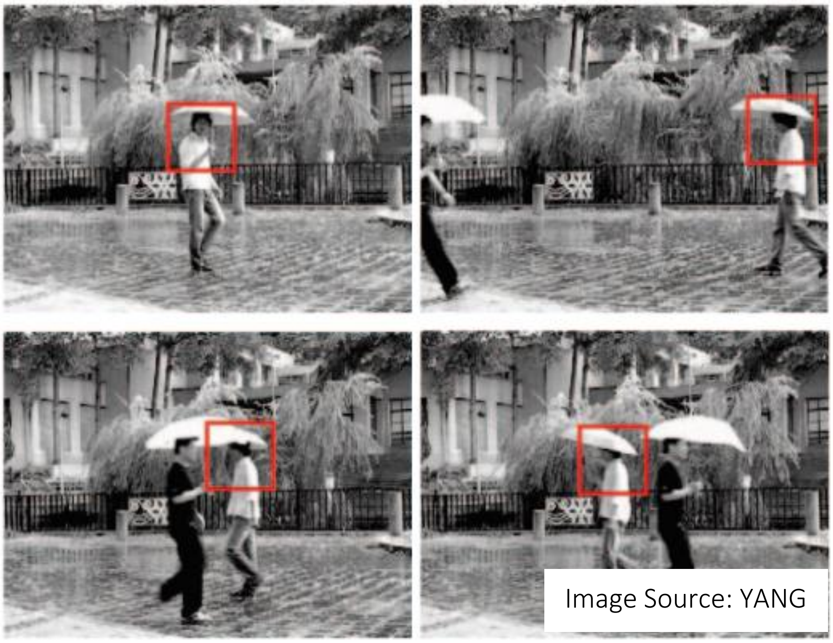


Image Source: YANG

Figure 21 Tracking result.

(LEIRA, 2015) proposed a system that makes automatic detection, classification and tracking of objects of interest in the ocean surface from UAVs using thermal cameras. Knowing that UAVs can operate autonomously in dynamic and dangerous operational environments, object detection, classification and tracking can often be one of the main goals. This paper discusses the development of a machine vision system for a low-cost fixed-wing UAV with an embedded computer processing the images in real-time.

The proposed system incorporates the use of thermal imaging camera and on-board processing power to perform real-time object detection, classification and tracking in the ocean surface. The system was tested on thermal video data captured in a test flight, and was able to detect 99,6% of the objects of interest located in the ocean surface. Only 5% of the objects were false positives. Furthermore, it classified 93,3% of the object types it was trained to classify. Finally, the system was able to track 85% of the object types it was actively searching for in real-time simulation test.

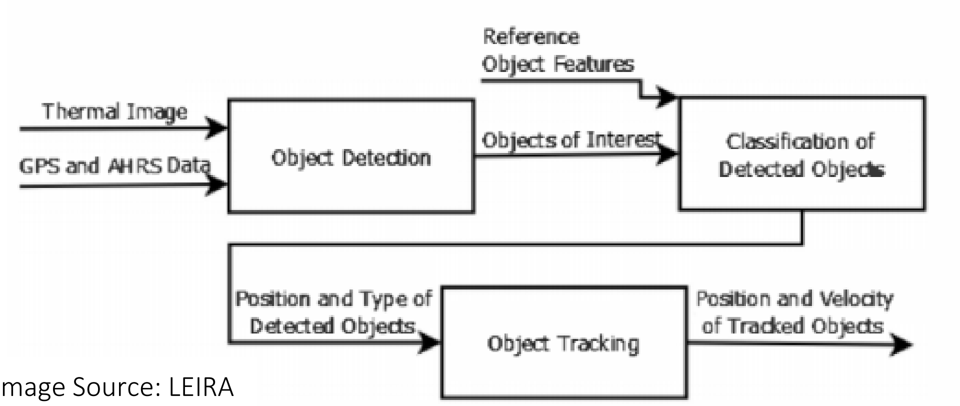


Image Source: LEIRA

**Figure 22 Data flow in the object detection, classification and tracking module.**

(BAZI, 2014) proposed an automatic approach for palm tree counting in images captured by UAVs. First a set of keypoints are extracted using the Scale Invariant Feature Transform (SIFT), then these key points are analyzed with a kernel-based classification method called Extreme Learning Machine (ELM) that is a priori trained on a set of palm and no-palm key points extracted by SIFT from a set of palm and no-palm training templates. To capture the shape of each tree, they proposed to merge these key points with an active contour method based on Level-Sets (LS). Due to the key points incorrectly classified by ELM as palm, the end of the grouping process could be effected by issues like merging very close trees or creation of regions related to other green vegetation. Focusing on this problem, they introduced processing operations using mathematical morphology to separate large connected components that may contain more than one palm tree. Lastly the local spatial structure of the obtained regions with local binary patterns (LBPs) is described. This method is responsible to distinguish palm trees from other type of vegetation.

Image Source: BAZI

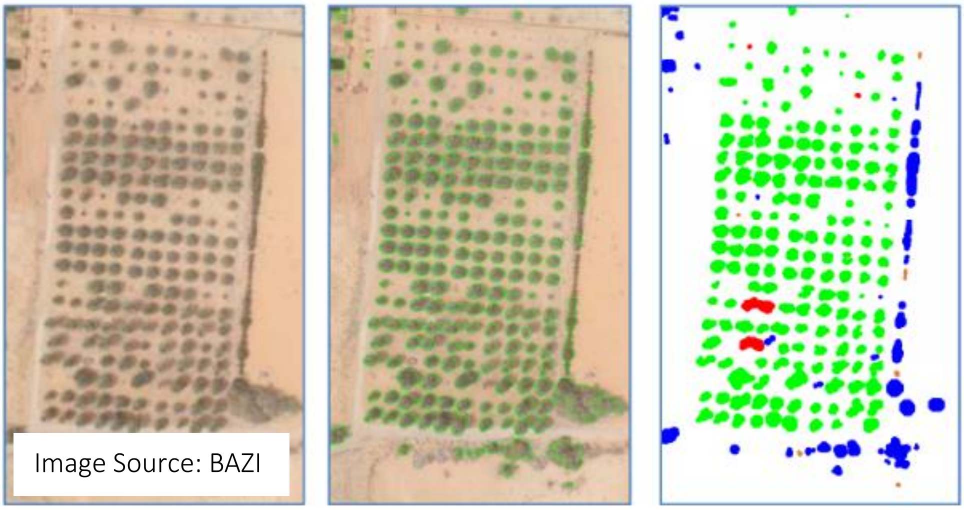


**Figure 23 Detection Results: UAV image, grouping of SIFT key points with LS and final classification map respectively.**

(ZHOU, 2016) proposed a novel method that can effectively tune the thresholds in detection algorithms. In this work was observed that the high level spatial information of the power lines is very helpful to filter out noise for edge detection. Accordingly, a parameter mapping method was introduced to associate the optimal detection parameters with the corresponding backgrounds. In each frame, a power line model is built and saved. For the unvisited background, the UAV uses the power line model detected in previous frames to select the best parameters. For the visited background, the optimal parameters are reused. By doing so, the accuracy of the edge detector is improved since the parameters used for the corresponding background is always optimal. For the second problem, the camera is tilted to face the power line at an angle so that a larger region of the power line can be shown in the frame. By doing so, each input frame provides a globalized view of where the power line goes and, hence, benefits power line tracking. Figure 24 shows the overall flow of the detection algorithm.

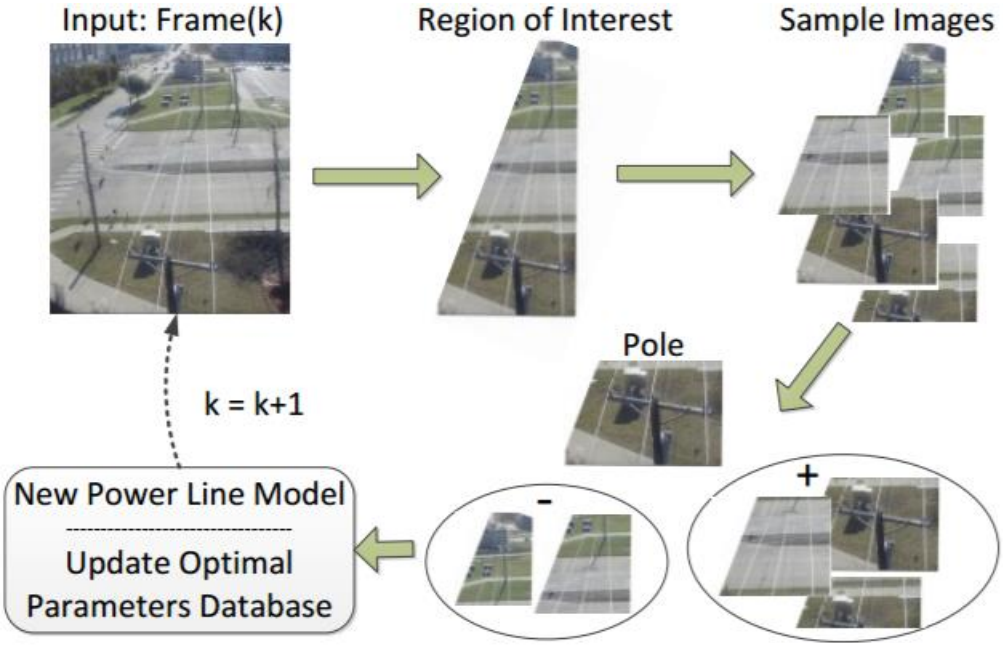


Image Source: ZHOU

**Figure 24 Overall flow of the detection algorithm.**

(TIAN, 2015) implements a multiple and part line image recognition method of power lines, to detect the high voltage overhead transmission line rapidly and accurately. The method provides a simple and intuitive measurement method for power line recognition of UAVs inspection, especially the recognition for some complicated geographical conditions and the line segment artificial difficult to reach, which makes up the shortage of traditional power line recognition method. The algorithm shows a good application in the field of electric automatic detection and fault identification. Figure 25 shows the flow chart of the power recognition method.

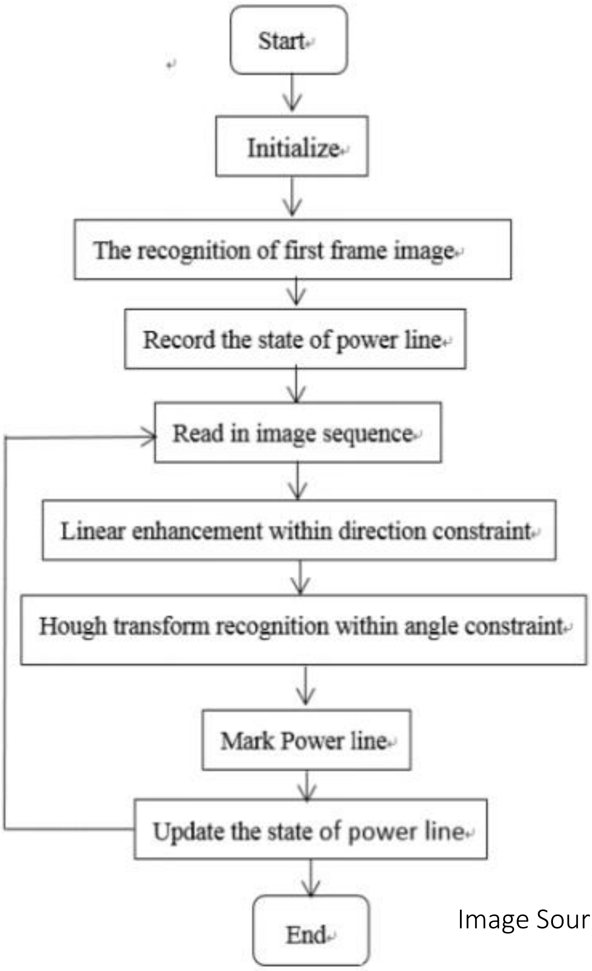

Start
Initialize
The recognition of first frame image
Record the state of power line
Read in image sequence
Linear enhancement within direction constraint
Hough transform recognition within angle constraint
Mark Power line
Update the state of power line
End

Image Source: TIAN

**Figure 25 The flow chart of power line recognition.**

# 5 PROBLEM STATEMENT

The problem to be tackled by this work is the use of embedded image processing to support autonomous flight of UAVs in electrical power lines inspections. This major problem has two main parts: the first is the image feature extraction to detect the power lines and the second part is how to use the image extraction information about the power lines to drive the UAV movement on top of the lines.

The first alternative that could be considered to autonomously trace the route to a UAV inspect power lines would be using the GPS coordinates of the poles that support the transmission power lines as waypoints of a preprogrammed fly mission. The major problem in this approach is the lack of precision of the GPS coordinates used as reference. It is known that the GPS coordinates have a margin of error that can reach meters away, and this lack of precision may make the UAV not fly over the transmission power lines at all. Figure 26a shows the way the UAV should fly to perform the inspection mission while Figure 26b shows the possible problem in using only GPS-based waypoints to setup the mission. Observing this problem, the necessity of a fine grain algorithm that does these flight adjustments was identified.

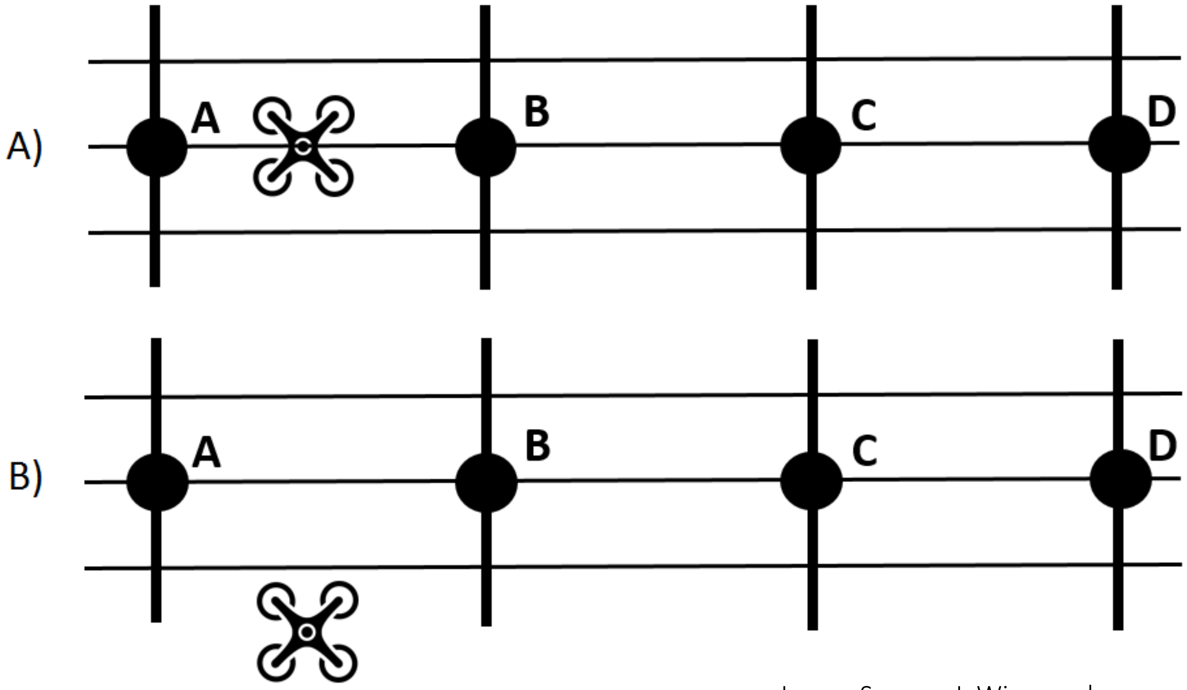


Image Source: I. Wieczorek

**Figure 26 The problem of using GPS waypoints to setup the power lines inspection mission: a) how the UAV should fly over the power lines; b) a possible deviation due to the lack of precision the GPS.**

The power lines detection in images acquired by UAVs is a complex image processing problem. There are several variables that play a role in the considered image analysis problem, however, this number needs to be delimited to restrict the bounders of a feasible solution. The first characteristic to be taken into account is the fact that the aerial image has to show the power lines from the top, collinear with the motion shown in the acquired video, as shown in Figure 26a. Another important part of the problem to be considered is that to begin the process of line detection, the first image frame has to contain the number of power lines to be detected. In the example shown in Figure 26, there are three parallel lines, which have to be detected by the algorithm and then follow these lines through the next frames. The illumination is an important factor that can interfere, because this kind of image always use natural environmental light, so this problem need to be treated by the algorithm, for example, considering the incidence of sunlight in different periods of the day.

Changes in the background (or very non-uniform background) represent another common source of problems that the vision systems have to deal with. An example of this operation scenario is presented in Figure 27. In cases in which these background changes occur, there is a big change of the image processing system parametrization, so the algorithm needs to be robust against environment changes.

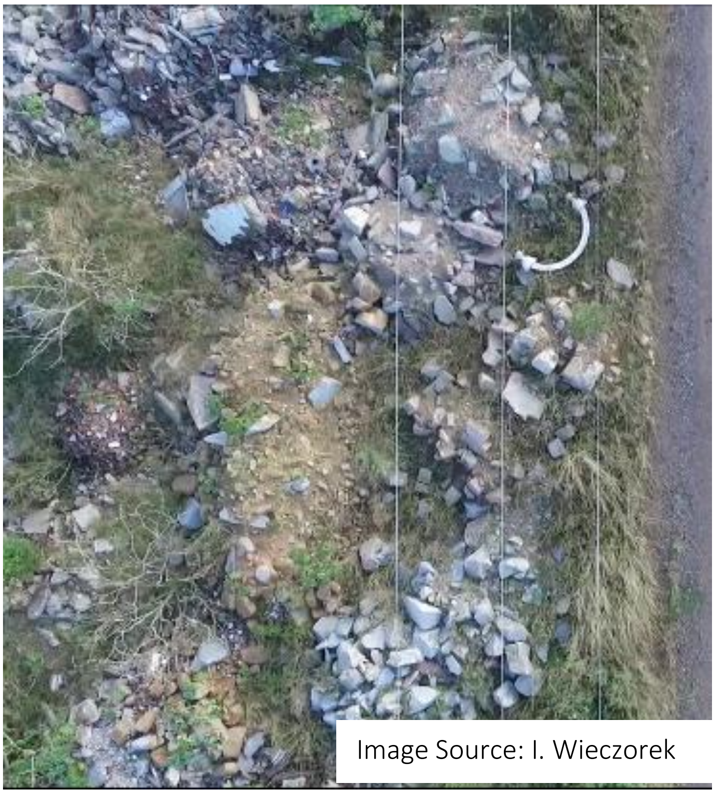


Image Source: I. Wieczorek

**Figure 27 Example of noise background that makes harder the detection process.**

Another factor that interferes in the execution of the method is the image's quality. Considering the quality of the images, there are many factors like focus, stabilization and their size. The last cited factor has a direct impact in the processing cost, in view of implementation on an embedded system, this parameter must to be seriously taken into account. The other parameters are treated by the hardware to stabilize the image and with the software which controls the camera system to maintain the focus.

Currently on the area of automated power line detection using UAVs, there are no solution that fully addresses the problem, thus there is a wide area for research. In this work, the focus is to detect the power lines and to use this detection to perform an autonomous flight using an efficient and reliable algorithm considering the boundary conditions and the scope delimited in this problem statement.

# 6 PROPOSED METHOD

In order to minimize all the possible image noises, thus facilitating the line detection, the proposed method does a series of pretreatments and calibrations in the image before the line detection stage. Figure 28 presents a block diagram of the proposed solution. In this figure, blocks 1 and 2 represents the image capturing and noise filtering, then block 3 represents the line detection, and block 4 the control module. This section describes step by step all the stages in the line detection procedure and the resulting automated fine grain flight control.

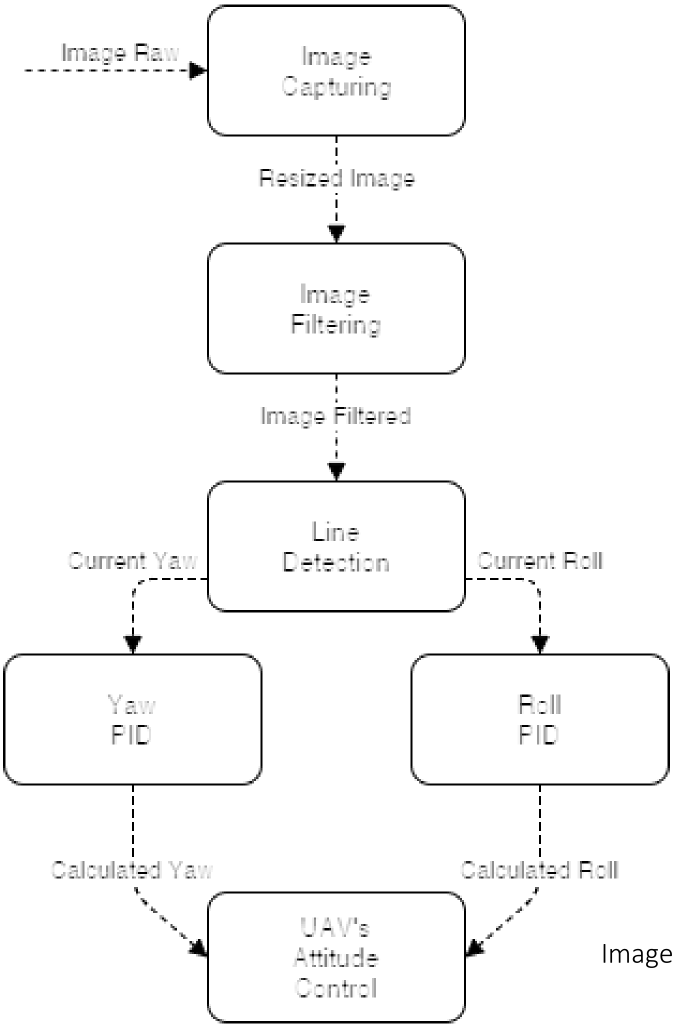


Image Source: I. Wieczorek

Figure 28 Proposed solution.

### 6.1 Image Capturing

The way that the image is captured has a crucial importance in the efficiency of the method. A perspective angle of capture may insert a certain level of noise causing effects such as line cracking, which can difficult the detection and segmentation of line segments.

To avoid this kind of situation and to facilitate the line extraction from the acquired images, the most appropriate angle is the one provided by a top view, or bird's eye view as shown in Figure 29. This image view can be computationally obtained, but that transformation has a considerable computational cost and may add some image distortion. The camera used in this work was the Zenmuse X3. This camera is coupled to a gimbal, which enables its adjustment to the desirable angle. If this feature was not available, the Inverse Perspective Mapping (IPM) method would be necessary, increasing the computational cost. With the bird's eye view angle, the power lines become parallel among them, making the process of delimitation the Regions of Interest (ROIs) easier. ROIs are defined areas around the power lines that delimits the area where the system will process data. The usage of ROIs contributes to the optimization of the process since it reduces the algorithm’s data load.

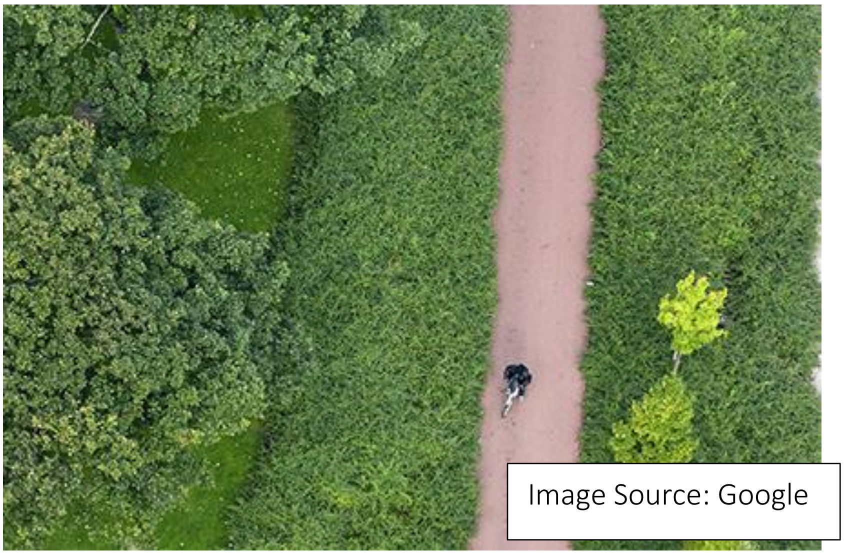


Image Source: Google

**Figure 29 Bird’s eye view angle.**

The bird’s eye view angle shown in Figure 29 must be adjusted in the applications’ start by the operator, and once set, it never changes.

### 6.2 IMAGE FILTERING

As mentioned in the introduction of this section, the image captured by the camera needs to receive some pretreatment to improve the efficiency of the line detection method. So, a couple of filters are applied on the input image to make the line detection easier. All the image filtering steps will be described in this section.

#### 6.2.1 IMAGE RESIZE

The minimal capture resolution of the Zenmuse X3 is HD, which means 1280x720 pixels. So firstly, the original image is resized to 624x352 pixels to reduce the processing load. When the image's resolution is reduced, some important data may be affected, such as line pixels for example. So, it is important to find a threshold between efficiency and quality. Figure 30 shows the original image resized in the new dimensions.

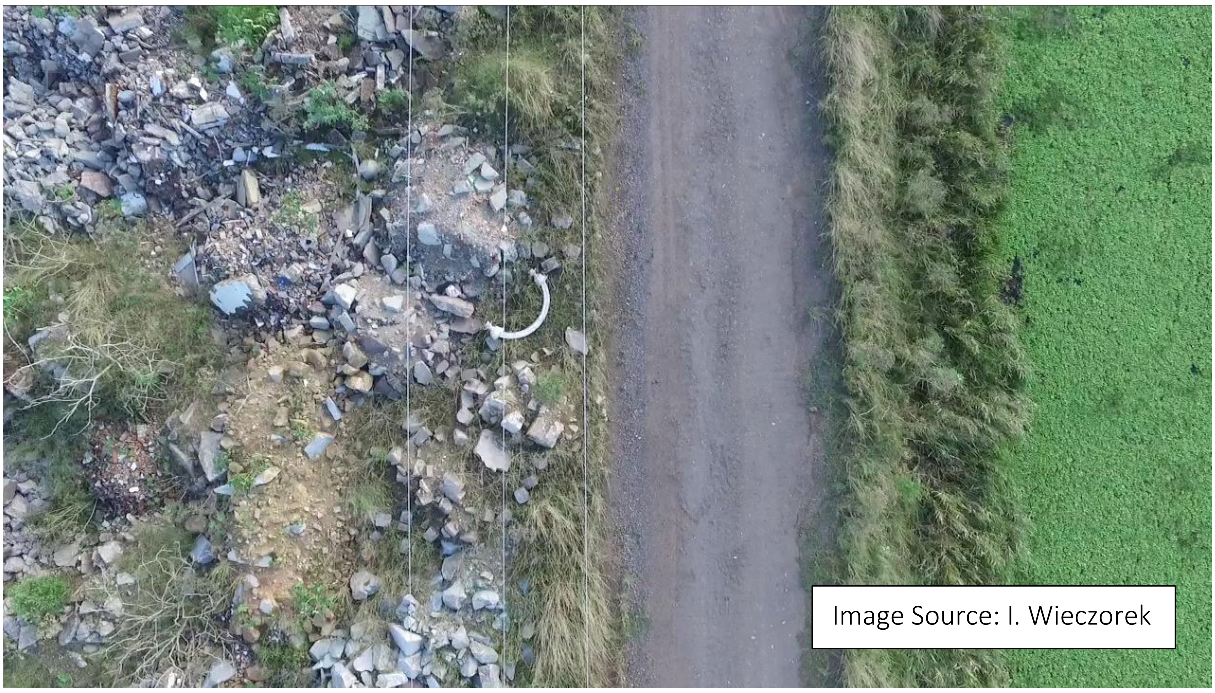


Image Source: I. Wieczorek

**Figure 30 Resized image**

The image resizing has a fundamental importance to the effectiveness of the image processing module, since this system is intended to be executed in an embedded computer that in most of cases does not have a powerful processing capacity, which can make inviable its execution in real time.

### 6.2.2 GRAY SCALE

The second step is transforming the image from RGB to gray scale to reduce unnecessary information and consequently computational cost. The main reason why grayscale representations are often used for extracting descriptors instead of operating on color images directly is that grayscale simplifies the algorithm and reduces computational requirements. Indeed, color may be of limited benefit in many applications and introducing unnecessary information could increase the amount of training data required to achieve good performance (KANAN, 2012). Figure 31 shows the image in grayscale.

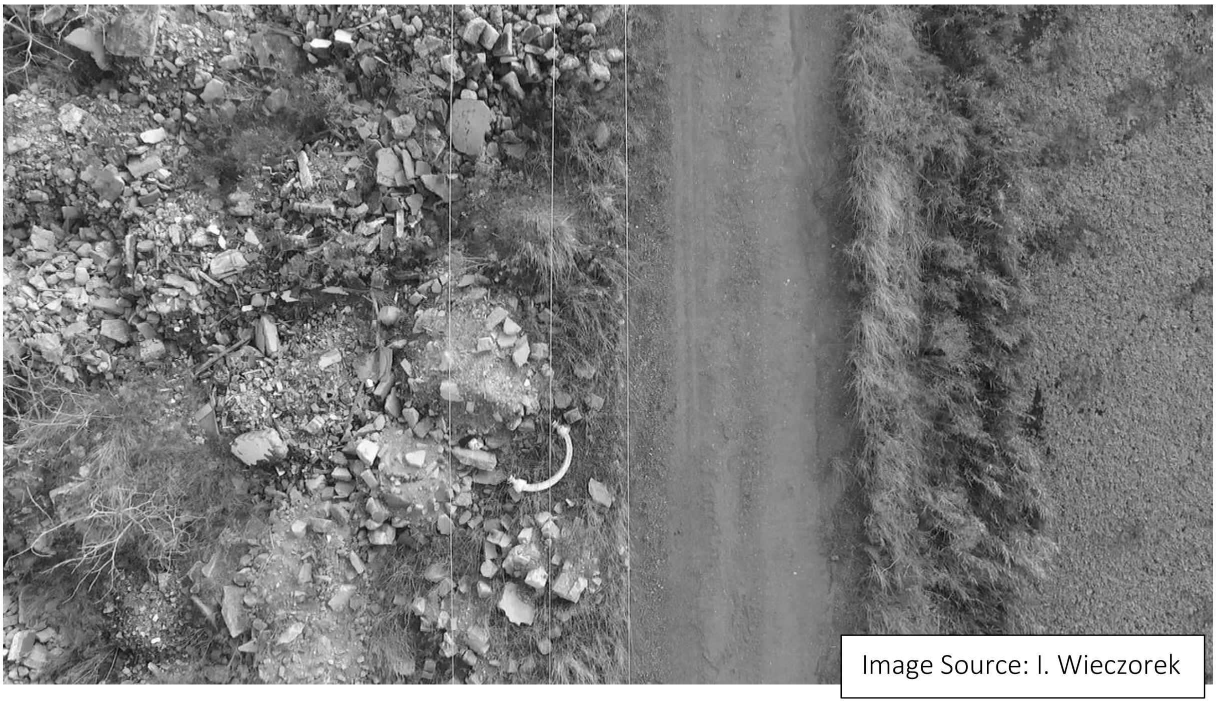


Image Source: I. Wieczorek

**Figure 31 Gray scale.**

### 6.2.3 Linear Filter

After the image has been converted to gray scale, a 2D linear filter with rectangular shape is convolved with the resultant image from the gray scale transformation. That filter helps the extraction of the pixels that are similar to the kernel’s shape, consequently to the transmission power lines.

2D filtering basically does a convolution between every pixel of the input image and a predefined kernel. The kernel's shape was defined as a rectangle with 20 pixels of height and 2 pixels of width. This rectangular shape matches the shape of the transmission power lines.

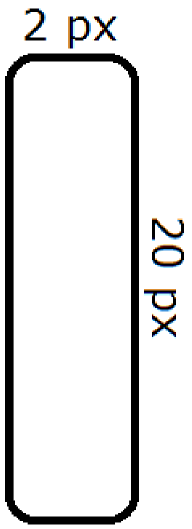


Image Source: I. Wieczorek

**Figure 32 2D filter kernel shape.**

The image resultant after the 2D filtering is shown in Figure 33. As can be seen, almost all the pixels belonging to the transmission power lines still remain in the image, while most others that no belong to them do not are shown. This filter helps to eliminate almost all data from the background of the image that is not of interest.

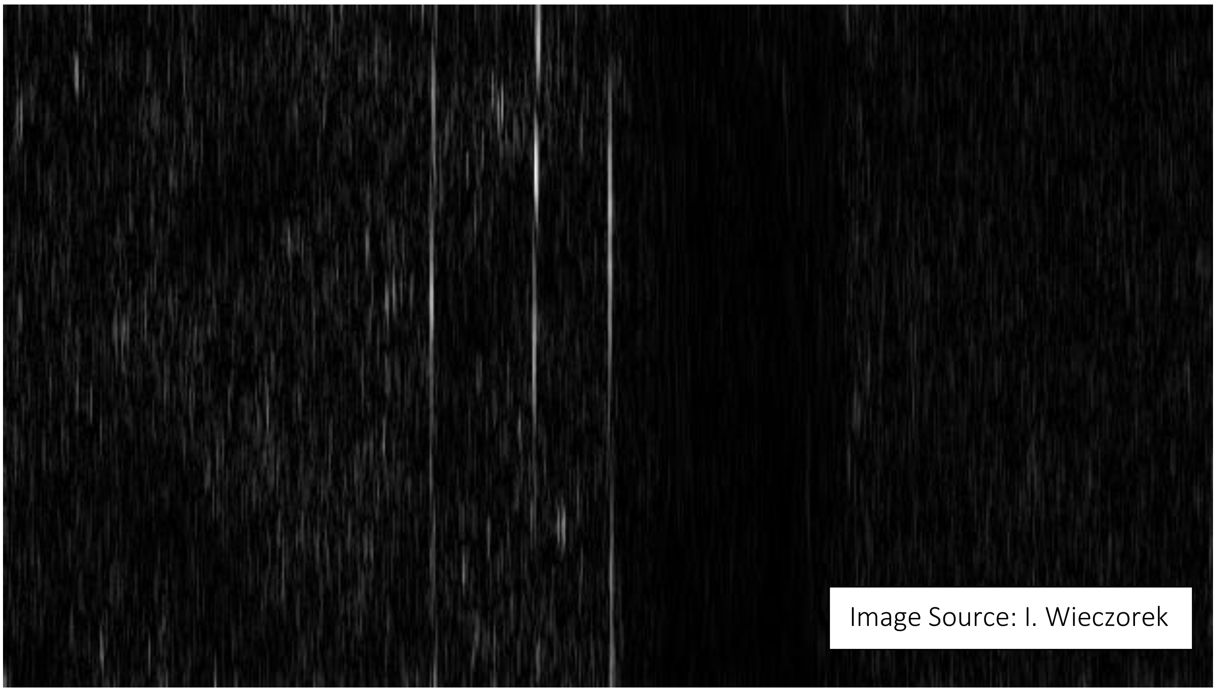


Image Source: I. Wieczorek

**Figure 33 2D filter.**

After the 2D filtering, a binary threshold is applied to improve the segmentation of the pixels that belongs to the power line transmission. The result is shown in Figure 34.

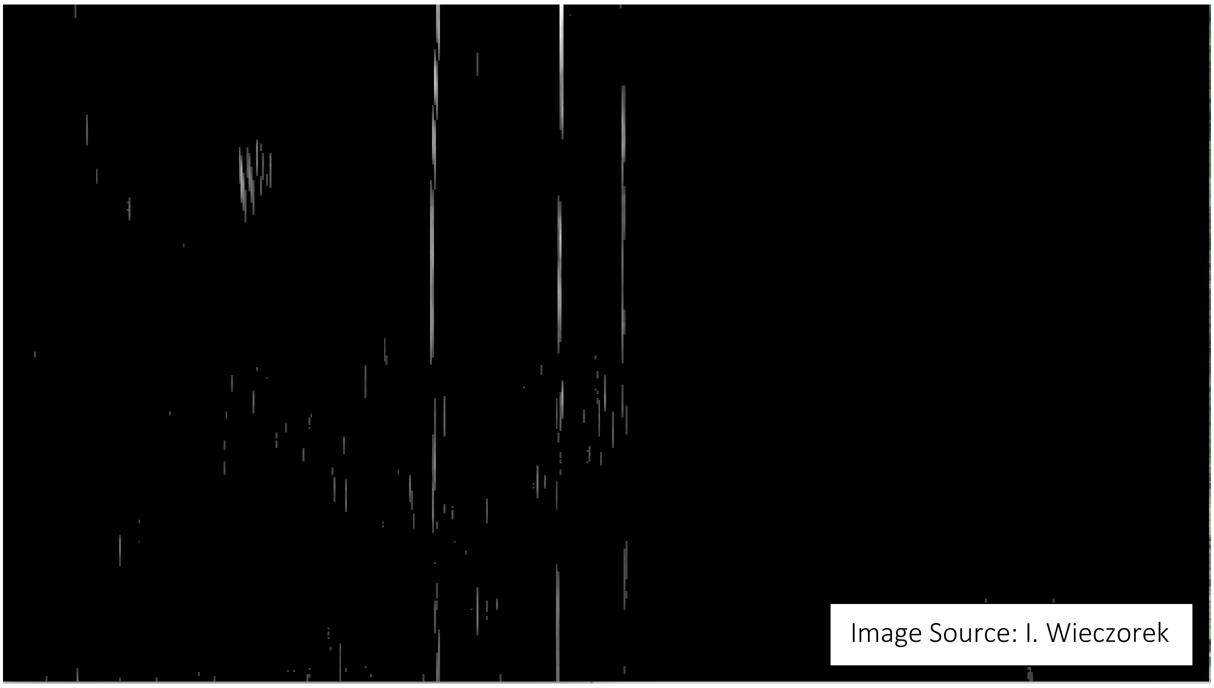


Figure 34 Binary Threshold.

### 6.3 LINE DETECTION

The line detection procedure is the most important stage of the proposed technique in this work. All the pretreatment performed before this stage have the goal to create an environment which makes the detection of the power lines easier.

In this stage, the camera must be in the birds eye view angle and the UAV must be flying between 8 to 10 meters above the transmission lines and focusing all the lines that will be identified. That is the initial position of the UAV and camera to start the application. Other important defined parameter is the number of lines that the system will recognize while in mission This number is important because it will help to discard false positive lines identification during the mission.

After all these requirements have been met, the module start detecting the lines and analyzing data about the position of the transmission power lines in the image to feed the

Proportional Integral Derivative (PID) controllers, which are responsible to the flight navigation.

### 6.3.1 HOUGH TRANSFORM APPLICATION

To identify the transmission power lines, the Hough transform (DUDA, 1972) is used. The Hough transform is an image processing technique that is used to detect lines (or other parametric shapes) in digital images. In a typical use case scenario, the image is preprocessed first (e.g. with an edge detector) to obtain pixels that are on the lines (or curves) in image space. Unfortunately, because of noise in the image, there may be missing pixels, and the extracted pixels often do not align well with the lines (curves) someone looks for. For this reason, it is often nontrivial to group the extracted pixels to an appropriate set of lines (curves). The purpose of the Hough transform is to assign these pixels to line (curve) instances by performing a voting procedure in a parameter space where the lines (curves) can be described with a small number of parameters (LUO, 2016).

When the system starts the Hough Transform is applied in all the image. Then all the identified lines are analyzed and based on its similarity, parallelism and angle. Lines that no match those requirements are excluded. So, if the number of lines that will be identified were defined as three, for example, the three lines that met the characteristics cited above will be classified as transmission power lines. The application of Hough Transform in all image only occurs on the firsts frames to not overload the hardware capacity, due the cost of the application of this method in all the image. This approach is used only to initially identify the position of the transmission power lines that the UAV will follow, and then create regions of interest around them.

### 6.3.2 Dynamic Regions of Interest

After the initial frames were processed, the position of the transmission power lines that UAV will follow are now known. To not overload the system, regions of interest that follow the identified lines are created. These regions are rectangles areas that are created around the identified lines and are used as masks to remove all the image information outside of the rectangle's area. This approach reduces the context where the Hough Transform is applied and consequently reduce computational cost and false positive detection.

The position of the regions of interest are dynamic and their locations vary according to the position of the lines. The position of the region of interest is always a delayed iteration with respect to the input image. This approach takes into account that the positioning of the transmission power lines does not vary so sharply to the point of leaving the region of interest. Figure 35 shows the lines being detected.

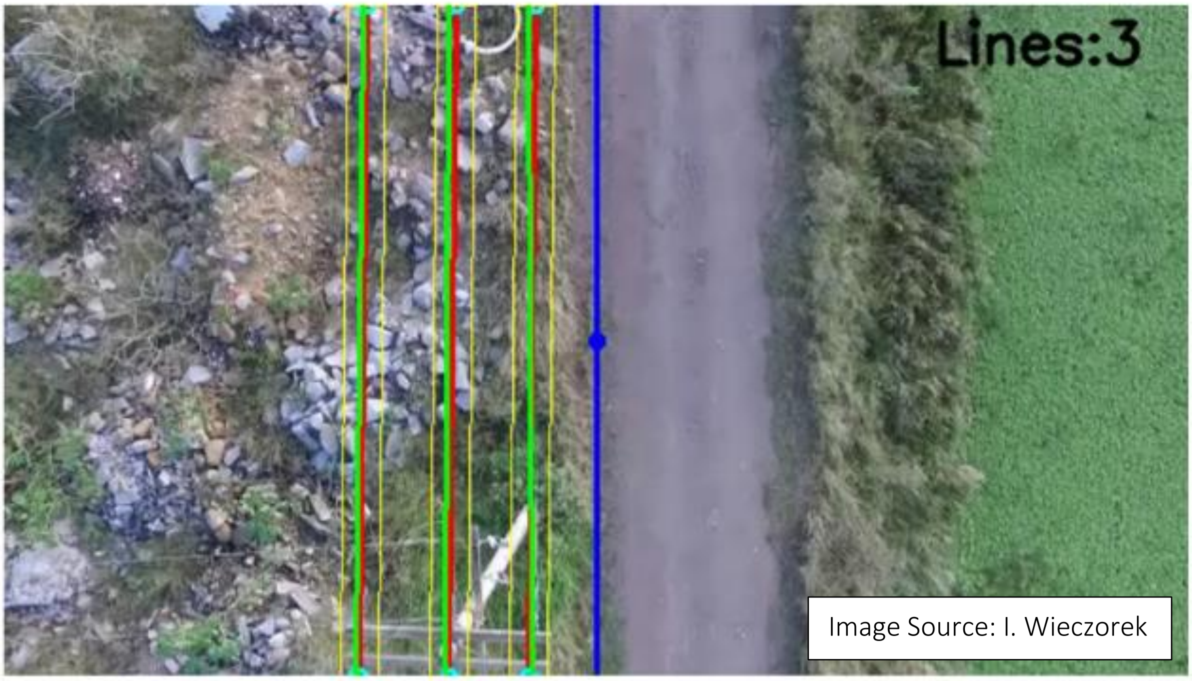


Image Source: I. Wieczorek

**Figure 35 Regions of Interest.**

In Figure 35 the region of interest is represented by the area inside the yellow boundaries. In green, the lines successfully detected. In red, the position of the detected lines in the last iteration. This position is used to draw a line in situations when no lines were identified. And finally, in blue, a reference line which its position is known and fixed.

### 6.3.3 Yaw and Roll data

During the mission, the UAV may change course due to climate effects, or the lines may change its course and the UAV must adjust its flight route. All these scenarios impact in route correction. To identify variations in the position of the transmission power line, two variables are always analyzed. The distance of the lines from the reference line (the blue line in Figure 35), and the inclination variation of the identified lines.

The distance between the reference line is the Roll data, and it is simply the distance between the transmission power line and the reference line. The angle variation of the lines defines the Yaw data. This variation is measured by the arc tangent between the first and last pixel of a detected line. Figure 36 illustrates how Roll and Yaw angles are estimated.

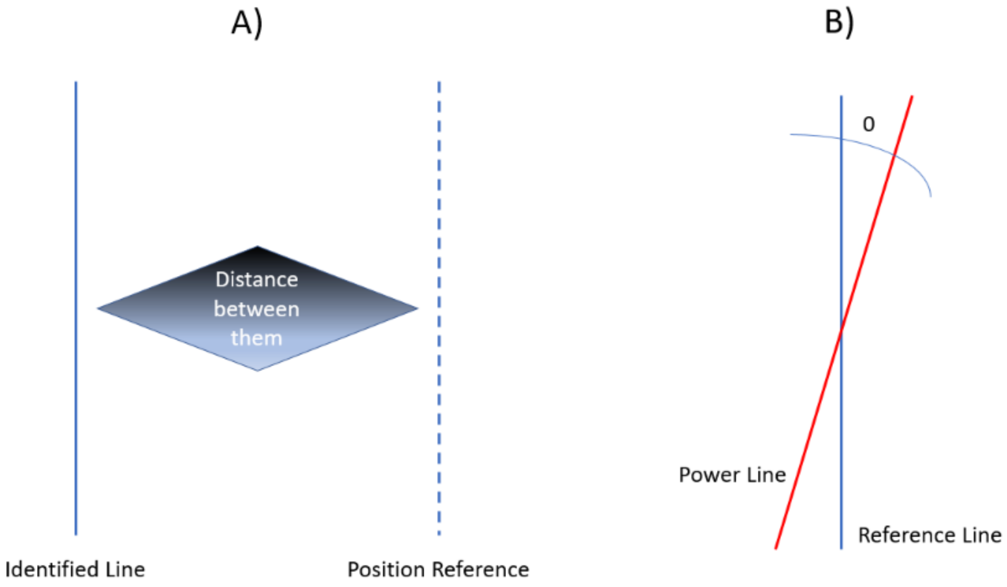


**Figure 36 Roll and Yaw estimation, respectively.**

### 6.4 Flight Control

The flight control module has the objective to autonomously guide the UAV over the transmission power lines. While the raw image is processed and the lines are being detected, the distance between the detected lines and the center of the captured image is measured in each frame. Figure 37 shows the three flight angles of a multi-rotary UAV.

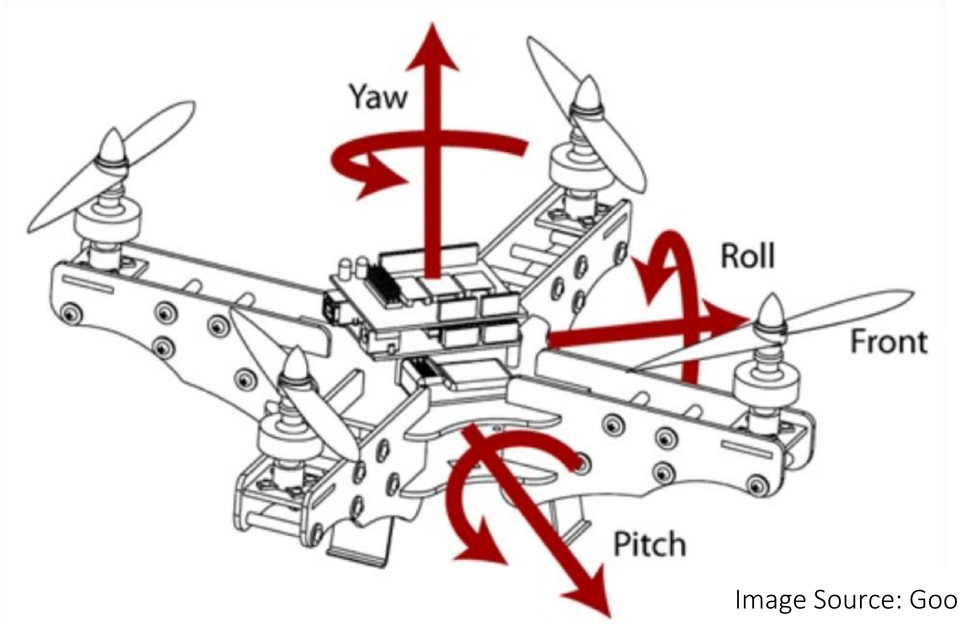


Image Source: Google

**Figure 37 Multi-rotary UAV’s flight angles.**

All the collected data from the measure between the detected lines and the center of the captured image feeds a Proportional Integral Derivative (PID) controller that acts in the UAV's roll flight angle. Figure 38 shows the PID diagram. At the same time that roll is estimated, another PID controller estimates the yaw flight angle. This measure is done by comparing the angle variation between the detected line and a set point defined as a perpendicular line from the image's bottom margin. The last flight angle is the pitch. The pitch angle defines the UAV's forward velocity, and in this work, it is considered as a constant defined before the mission starts.

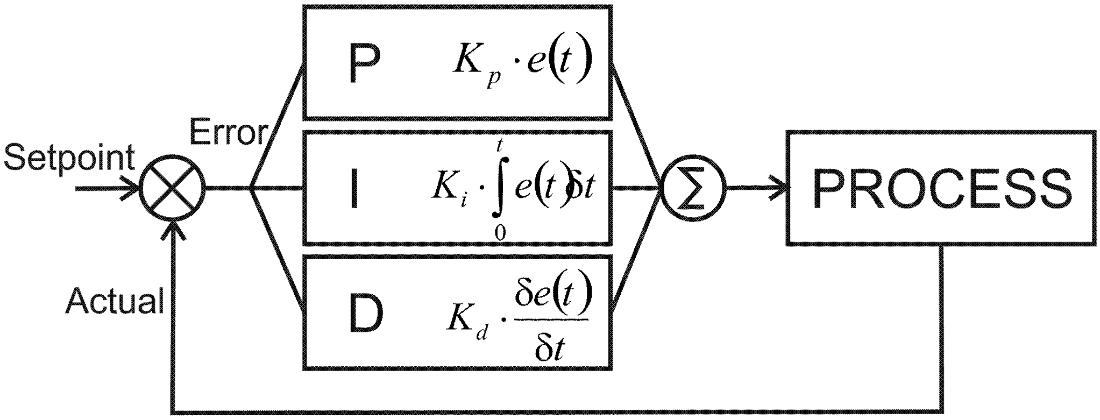
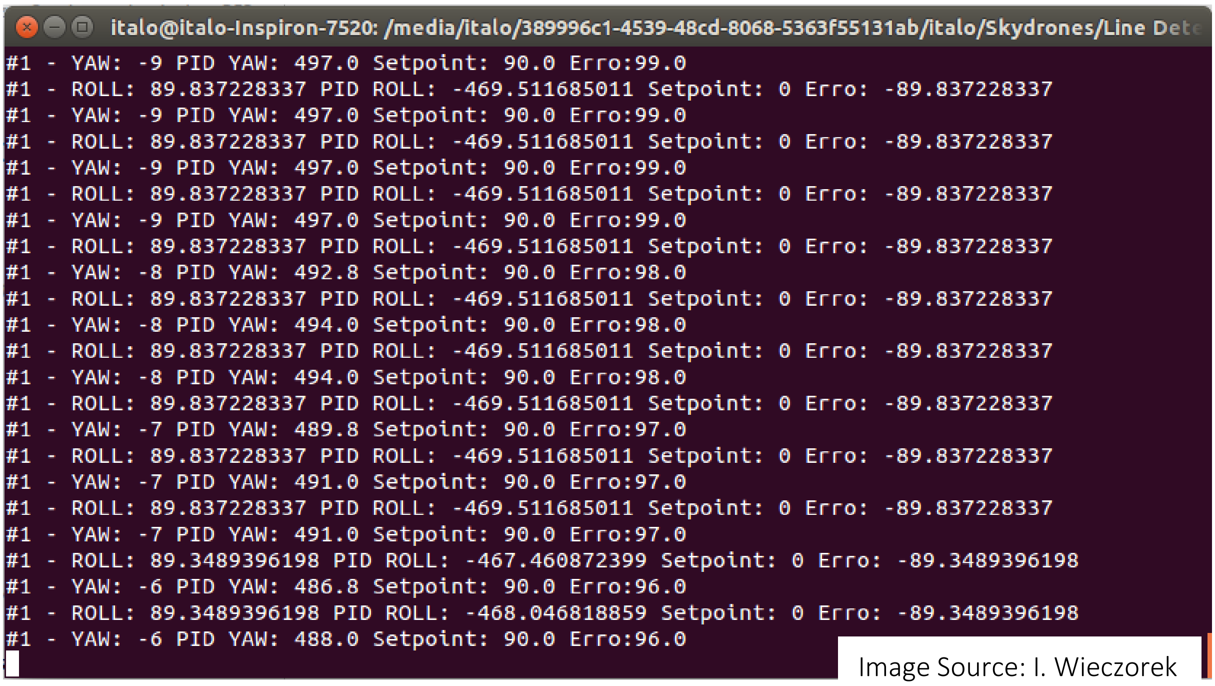

Setpoint
Error
Actual
P $K_p \cdot e(t)$
I $K_i \cdot \int_0^t e(t)\delta t$
D $K_d \cdot \frac{\delta e(t)}{\delta t}$
$\Sigma$
PROCESS

Image Source: Google

**Figure 38 PID Controller.**

Figure 39 shows the data output screen of the PIDs controllers estimating the flight angle positions.


italo@italo-Inspiron-7520: /media/italo/389996c1-4539-48cd-8068-5363f55131ab/italo/Skydrones/Line Dete

```
#1 - YAW: -9 PID YAW: 497.0 Setpoint: 90.0 Erro:99.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -9 PID YAW: 497.0 Setpoint: 90.0 Erro:99.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -9 PID YAW: 497.0 Setpoint: 90.0 Erro:99.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -9 PID YAW: 497.0 Setpoint: 90.0 Erro:99.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -8 PID YAW: 492.8 Setpoint: 90.0 Erro:98.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -8 PID YAW: 494.0 Setpoint: 90.0 Erro:98.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -8 PID YAW: 494.0 Setpoint: 90.0 Erro:98.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -7 PID YAW: 489.8 Setpoint: 90.0 Erro:97.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -7 PID YAW: 491.0 Setpoint: 90.0 Erro:97.0
#1 - ROLL: 89.837228337 PID ROLL: -469.511685011 Setpoint: 0 Erro: -89.837228337
#1 - YAW: -7 PID YAW: 491.0 Setpoint: 90.0 Erro:97.0
#1 - ROLL: 89.3489396198 PID ROLL: -467.460872399 Setpoint: 0 Erro: -89.3489396198
#1 - YAW: -6 PID YAW: 486.8 Setpoint: 90.0 Erro:96.0
#1 - ROLL: 89.3489396198 PID ROLL: -468.046818859 Setpoint: 0 Erro: -89.3489396198
#1 - YAW: -6 PID YAW: 488.0 Setpoint: 90.0 Erro:96.0
```

Image Source: I. Wieczorek

**Figure 39 Flight control data.**

In Figure 39 is possible to see the roll and yaw measured, the output response of the PID, the defined set point and the error between the set point and the measured angle, respectively. For each video frame, the algorithm measures a roll and yaw estimation, so these values change based on the difference between the define set point and the measured error, as the position of the transmission power lines varies.

It is important to highlight that all the PIDs controllers developed to this work are generic ones, and they are not parametrized and neither studied in this work. However, its parameterization could improve significantly the time response of the route correction, although it still works fine as it is. The improvement of the parameterization of the controllers will be considered in future works.

It is known that a completely autonomous flight system that treats every unexpected event would be ideal (CHEN, 2009), but to reach this goal, more sensors would be necessary in addition to the imagery system.

## 6.5 IMPLEMENTATION DETAILS

After a frame has been processed according to proposed method, data about navigation is available to be used in the attitude control of the UAV. The interface between processed data and the UAV's attitude control is provided by the Robot Operating System (ROS) (QUIGLEY, 2009).

ROS is a software framework collection for robot software development, providing operating system-like features. It provides standard operating system services such as hardware abstraction, low-level device control, and package management.

The main idea of ROS utilization is to provide hardware abstraction to control the UAV. To make this possible, two main packages were concurrently executed. The first one is responsible for collecting the raw image captured by the camera and making this data available through a topic which could be subscribed by any module. This package is also responsible to provide all the UAV's flight functionality to any other package that need it. The second package has all the proposed image processing method described in this section.

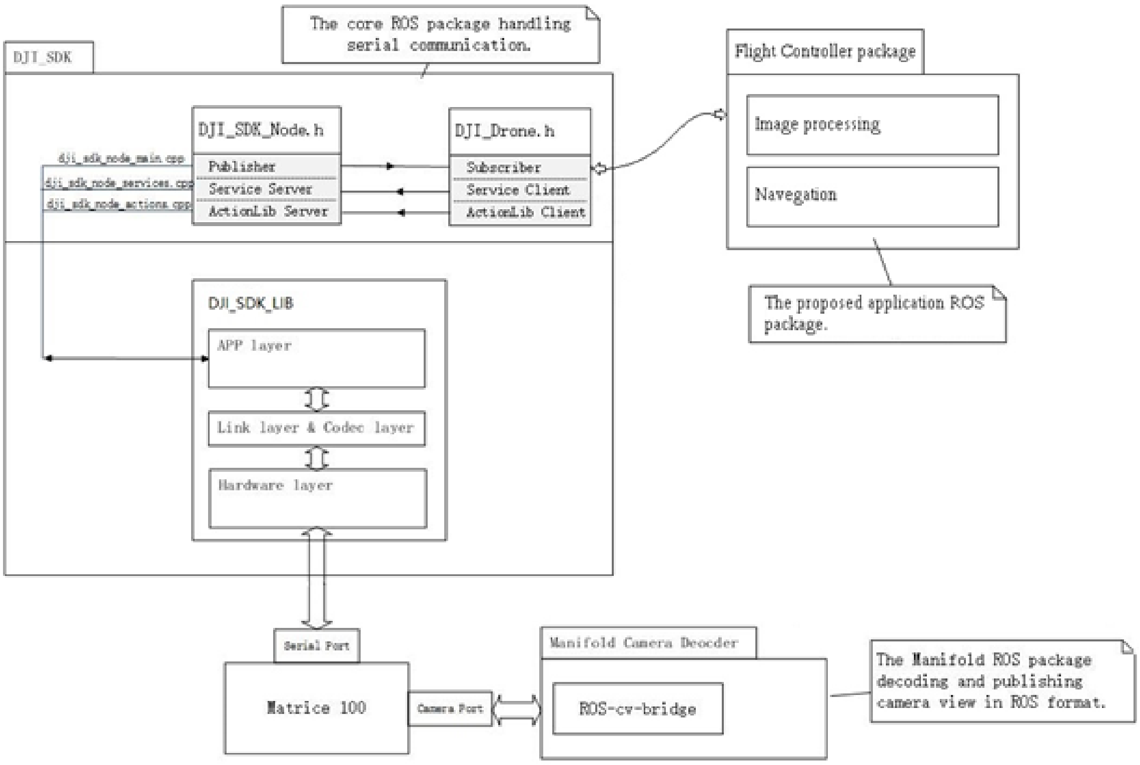


Image Source: I. Wieczorek

**Figure 40 ROS architecture.**

Figure 40 shows the proposed ROS architecture. The architecture consists of a core package that is responsible to handle all the serial communications between peripherals like the camera, the embedded computer and the UAV. This package has a library composed by three layers that communicates with the hardware, manipulates the collected data and provides all the hardware's features as services. These provided services are then used in the Flight Controller package, which contains all the proposed method described in last subsections.

The data flow in the architecture starts with the UAV`s camera. This camera is attached in the UAV`s frame through a gimbal and all the captured images are sent by serial connections to the UAV's internals embedded computer and to the mission control embedded computer. The objective of sending the image data to the UAV's embedded computer, is to provide the images in capturing time to the tablet attached in the remote control. The image data provided to the mission control embedded computer are used to execute the proposed method in the ROS. As the image date is provided as a service in ROS, all the UAV`s navigation sensors and actuators are too. Lastly, the package containing the proposed image processing method uses the image service to acquire the images in flight time, and the UAV`s attitude control service to act in the UAV`s navigation.

# 7 EXPERIMENTS AND RESULTS

To analyze in more detail the effectiveness of the algorithm's response time, a test scenario was built in laboratory. This scenario consists in the algorithm identifying a drawn line in a paper with a commercial web-cam. The experiment starts with the web-cam focused and centralized in the drawn line, then the web-cam's position is changed both in X and Z axis, to simulate roll's and yaw's variation.

The scenario built in laboratory showed in Figure 41 had the objective of to validate the algorithm's response to the line's position variation both in roll and yaw angles, as well as to assess its response time. The performed tests showed a positive reaction of the UAV with an average response time of 150 milliseconds between capturing the raw image, processing it and sending the attitude control data to the simulator.

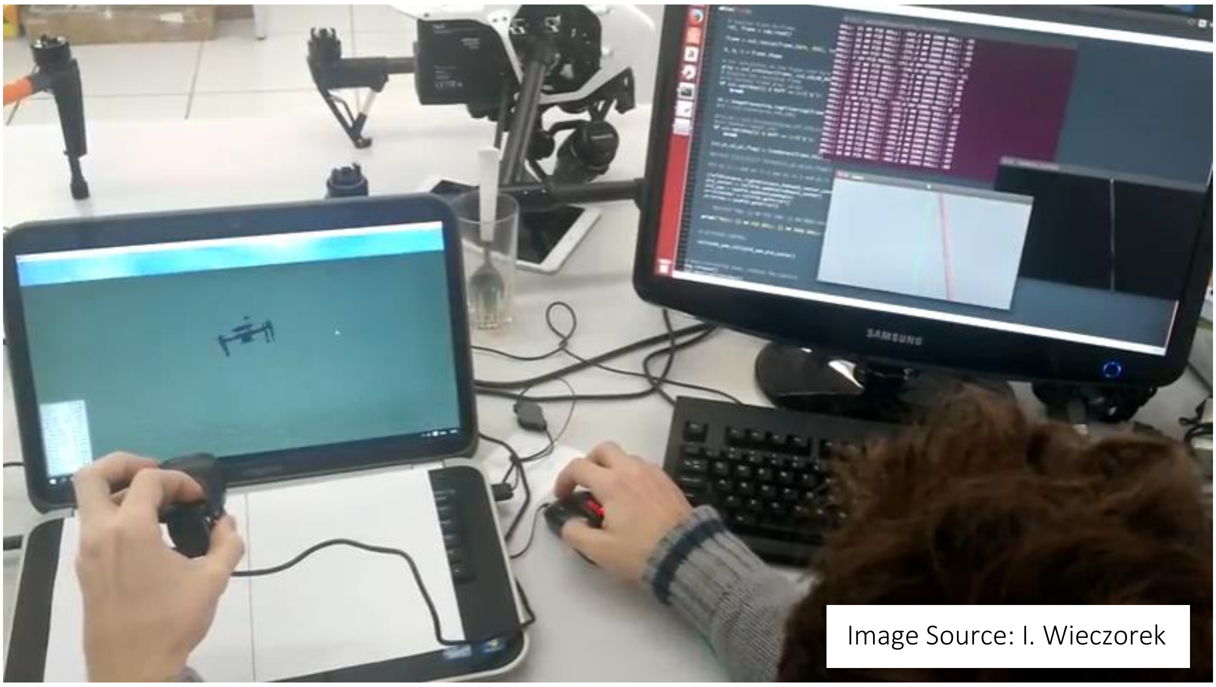
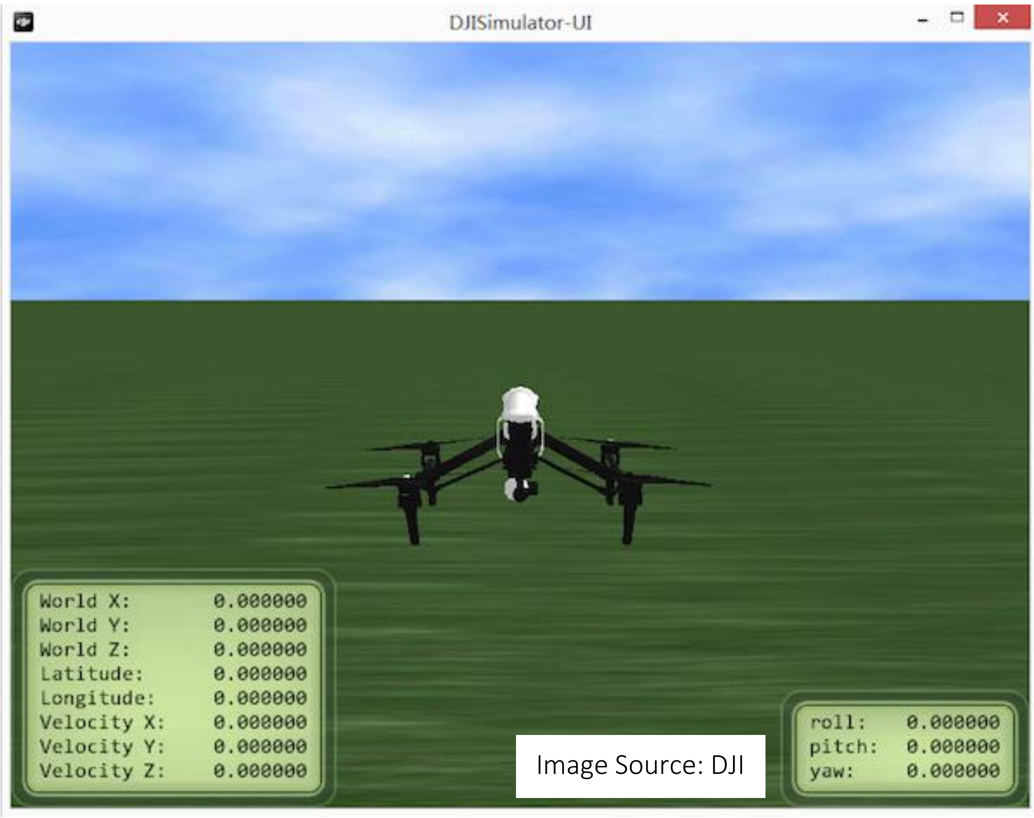


Image Source: I. Wieczorek

Figure 41 Laboratory test scenario.

To test the flight control module, the DJI's UAV flight simulator was used. The DJI's flight simulator is used to test applications under development that utilizes its UAVs, like the

Matrice 100 that was the one used in this project. Figure 42 shows the simulator screen, in which it is possible to observe in the bottom left part the positioning and velocity data of the UAV, while in the right bottom part the values of the flight angles. The embedded computer utilized to process all data during the simulation was the Manifold, provide by DJI. The Manifold has a Nvidia Tegra K1 quad core 2.3GHz processor, 2GB DDR3L RAM memory and 16GB of storage memory running Ubuntu 14.04 LTS with CUDA, OpenCV (BRADSKI, 2000) and ROS support.



```
World X:      0.000000
World Y:      0.000000
World Z:      0.000000
Latitude:     0.000000
Longitude:    0.000000
Velocity X:   0.000000
Velocity Y:   0.000000
Velocity Z:   0.000000
```

```
roll:    0.000000
pitch:   0.000000
yaw:     0.000000
```

Image Source: DJI

**Figure 42 DJI's simulator.**

Since the image input was not acquired in running time -a recorded video was used as study case-, the PIDs used to control UAV's roll and yaw did not work properly, because the data used to feed the controller in each processed frame came from the video, which does not reflect the controller acting in the UAV's navigation. However, the main objective of this simulation was to validate the image processing module while extracting navigation data, and the flight control module notwithstanding its behavior was not like a real flight. The trajectory flew by the UAV in the case study video was 350 meters at a speed of 10 meters per second.

During the video processing, not all frames were processed. For each three frames captured by the camera, only one was processed. This technique aims to reduce the computation cost, since the flight distance travelled is less than 30 centimeters between one frame and other.

The video utilized in the system’s validation has 36 seconds of duration, totalizing 864 frames. Figures 30 and 35 represents frame samples of the video.

Considering this scenario, the results obtained in the previously described performed testes and without taking into account the algorithm's route correction, in a segment of 50 meters, if the navigation suffers a variation of 5 degrees in its flight route, the UAV could vary its route around 4.5 meters away from the transmission power lines that it was following. This means that it would start flying completely away from the lines, which is completely undesirable.

Considering the assessed algorithm's response time of 150 milliseconds, the route variation over the transmission power lines could reach at most 40 centimeters, which makes an eventual route correction entirely viable in flight time. Figure 43 illustrates this eventual route deviation, in which the vertical lines represent the power transmission lines. In this figure, part A represent the moment in which the UAV start to deviate from its correct flight direction, part B represent the instant in time just after a small displacement (within the 40 centimeters that it is possible to fly in 150 milliseconds), and then part C represents the actuation due to the proposed fine grain guidance system bringing the UAV back to the correct flight direction.

The measure of 150 milliseconds takes into account the moment that the image is captured by the camera until a frame has been processed and the flight control stimulus has been generated. So, all the image capturing, image filtering, flight control data generation and UAV's attitude control steps take around 150 milliseconds to happen.

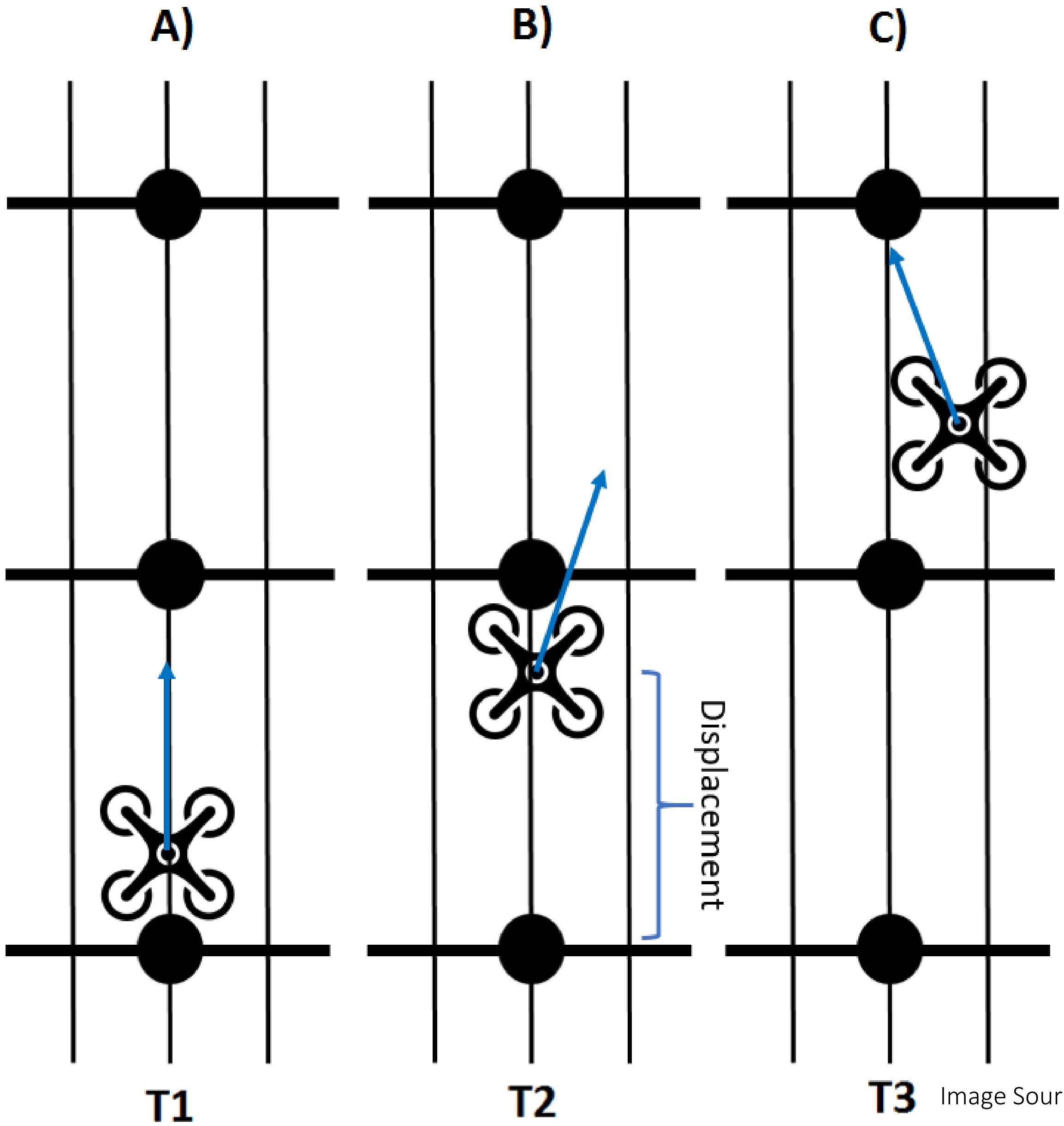


Image Source: I. Wieczorek

**Figure 43 Route deviation in time T2 and the correction of movement in time T3.**

Evaluating a little more deeply the 150 milliseconds of the system response, almost of 70% of this time come from the image processing module due to the robustness of the method. The rest of the time is generated by the image capturing node on ROS, that provides all frames to the application.

# 8 CONCLUSIONS

This work presented an image processing algorithm aiming at supporting a fine grain UAV guidance system for transmission power lines inspection applications. The frames computed by the algorithm firstly are filtered focusing on noise removal and shapes of interest highlighting. Than the Hough transform method is applied to detect all the transmission power lines inside of regions of interest that covers the lines. Finally, a flight control module extracts data from the captured images, using them to autonomously control the flight.

The work described in this text has a huge importance in the inspection field, because it automates a process that nowadays is performed by two or three persons with the assistance of a helicopter, incurring in high risk to the involved professionals. Besides the cost reduction of making this kind of inspection autonomously, another aspect that highlights the importance of this work is the life risk reduction of who is doing the inspection, since only one person monitoring the UAV is necessary, which is not in direct and close contact with the electrical power lines.

The utilization of UAVs to do professional tasks like the inspections detailed in this work is relatively new. The main motivation of this work was the exploration of the UAV's resources to do tasks that before were performed by people, whereas UAVs are gaining more space in the market.

The proposed method has presented a good performance detecting lines with illumination and background texture variation, which is one of the biggest challenges in this field. The main technical contribution of this work is the effectiveness of the image processing module while removing noises from the raw image without any other dedicated subsystem. Other relevant aspect of this work is the navigation response time that makes viable the utilization of this system in real inspection applications.

Observing the acquired results in terms of computational cost of each part of the implemented solution, future works aiming at improving the method of line extraction without losing it robustness, would be beneficial for the entire solution performance.

The next steps of this work are improvements on treatment of unexpected events during the flight and improvements in the robustness of image processing module. Another direction for future work is to incorporate other feature extraction in the algorithm, for example, to inspect the poles that support the lines. In the same direction of improving robustness, another possible future work is preparing the algorithm to identify transitions or bifurcations.

# REFERENCES

AKINLAR, Cuneyt; TOPAL, Cihan. Edlines: real-time line segment detection by Edge Drawing (ed). In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 18, 2011, Brussels. **Proceedings...** Brussels: IEEE, 2011. p.2837-2840. Disponível em: <http://dx.doi.org/10.1109/icip.2011.6116138>. Acesso em: 5 jan. 2017.
BAZI, Yakoub et al. An automatic approach for palm tree counting in UAV images. In: IEEE GEOSCIENCE AND REMOTE SENSING SYMPOSIUM, 2014, Quebec. **Proceedings…** Quebec> IEEE, 2014. p.537-540. Disponível em: <http://dx.doi.org/10.1109/igarss.2014.6946478>. Acesso em: 6 jan. 2017.

BRADSKI, Gary. The OpenCV Library. **Dr. Dobb's Journal of Software Tools,** New York, v. 25, p.120-126, Jan. 2000.

CERON, Alexander; MONDRAGON, Ivan; PRIETO, Flavio. Power line detection using a circle based search with UAV images. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2014, Orlando. **Proceedings…** Orlando: IEEE, 2014. p.632-639. Disponível em: <http://dx.doi.org/10.1109/icuas.2014.6842307>. Acesso em: 8 jan. 2017.

CHEN, Hai; WANG, Xin-min; LI, Yan. A survey of autonomous control for UAV. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND COMPUTATIONAL INTELLIGENCE, 2009, Shanghai. **Proceedings…** Shangai: IEEE, 2009. p.267-271. Disponível em: <http://dx.doi.org/10.1109/aici.2009.147>. Acesso em: 9 jan. 2017.

CORRELL, Nikolaus; WING, Rowan; COLEMAN, David. A one-year introductory robotics curriculum for Computer Science upperclassmen. **IEEE Transactions on Education,** New York, v. 56, n. 1, p.54-60, 2013. Disponível em: <http://dx.doi.org/10.1109/te.2012.2220774>. Acesso em: 10 jan. 2017.

DESOLNEUX, Agnès. When the a contrario approach becomes generative. **International Journal of Computer Vision,** Dordrecht, v. 116, n. 1, p.46-65, 2013. Disponível em: <http://dx.doi.org/10.1007/s11263-015-0825-x>. Acesso em: 11 jan. 2017.

DUDA, Richard; HART, Peter. Use of the Hough transformation to detect lines and curves in pictures. **Communications of the ACM,** New York, v. 15, n. 1, p.11-15, 1972. Disponível em: <http://dx.doi.org/10.1145/361237.361242>. Acesso em: 12 jan. 2017.

FERNANDES, Leandro; OLIVEIRA, Manuel. Real-time line detection through an improved Hough transform voting scheme. **Pattern Recognition,** Porto Alegre, v. 41, n. 1, p.299-314, 2007. Disponível em: <http://dx.doi.org/10.1016/j.patcog.2007.04.003>. Acesso em: 13 jan 2017.

FREEMAN, William; ADELSON, Edward. The design and use of steerable filters. **IEEE Transactions on Pattern Analysis and Machine Intelligence, [**New York], v. 13, n. 9, p.891-906, 1991. Disponível em: <http://dx.doi.org/10.1109/34.93808>. Acesso em: 14 jan. 2017.

GROMPONE, Rafael et al. LSD: A fast line segment detector with a false detection control. **IEEE Transactions on Pattern Analysis and Machine Intelligence,** Illinois, v. 32, n. 4, p.722-732, 2010. Disponível em: <http://dx.doi.org/10.1109/tpami.2008.300>. Acesso em: 24 jan. 2017.

HUANG, Yongzhen; WU, Zifeng; WANG, Liang. Feature coding in image classification: a comprehensive study. **IEEE Transactions on Pattern Analysis and Machine Intelligence,** [New York], v. 36, n. 3, p.493-506, 2014. Disponível em: <http://dx.doi.org/10.1109/tpami.2013.113>. Acesso em: 15 jan. 2017.

HULENS, Dries; VERBEKE, Jon; GOEDEMÉ, Toon. How to choose the best embedded processing platform for on-board UAV image processing?.In: INTERNATIONAL CONFERENCE ON COMPUTER VISION THEORY AND APPLICATIONS, 10., 2015, Berlin. **Proceedings…** Berlin [s.n.], 2015. p.377-386. Disponível em: <http://dx.doi.org/10.5220/0005359403770386>. Acesso em: 16 jan. 2017.

ILLINGWORTH, John; KITTLER, Josef. A survey of the hough transform. **Computer Vision, Graphics, and Image Processing,** New York, v. 44, n. 1, 1988. p.87-116. Disponível em: <http://dx.doi.org/10.1016/s0734-189x(88)80033-1>. Acesso em: 17 jan. 2017.

KANAN, Christopher; COTTRELL, Garrison. Color-to-grayscale: does the method matter in image recognition?. **PLOS ONE,** San Francisco, v. 7, n. 1, p.1-6, 2012. Disponível em: <http://dx.doi.org/10.1371/journal.pone.0029740>. Acesso em: 17 jan. 2017.

KERR, John; NICKELS, Kevin. Robot operating systems: bridging the gap between human and robot. In: SOUTHEASTERN SYMPOSIUM ON SYSTEM THEORY (SSST), 44., 2012, Jacksonville. Proceedings... Jacksonville: IEEE, 2012. p.99-104. Disponível em: <http://dx.doi.org/10.1109/ssst.2012.6195127>. Acesso em: 18 jan. 2017.

KOSHELEV, Valery; KOZLOV, Dmitry. Wire recognition in image within aerial inspection application. In: MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO), 4., 2015**,** Budva. **Proceedings...** Budva: IEEE, 2015. p.159-162. Disponível em: <http://dx.doi.org/10.1109/meco.2015.7181891>. Acesso em: 19 jan. 2017.

LEIRA, Frederik; JOHANSEN, Tor; FOSSEN, Thor. Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera. In: IEEE AEROSPACE CONFERENCE, 2015, Big Sky. **Proceedings...** Big Sky: IEEE, 2015. p.1-10. Disponível em: <http://dx.doi.org/10.1109/aero.2015.7119238>. Acesso em: 20 jan. 2017.

LI, Wai Ho et al. Image processing to automate condition assessment of overhead line components. In: INTERNATIONAL CONFERENCE ON APPLIED ROBOTICS FOR THE POWER INDUSTRY (CARPI 2010), 1., 2010, Montreal. **Proceedings…** Montreal: IEEE,, 2010. p.1-6. Disponível em: <http://dx.doi.org/10.1109/carpi.2010.5624447>. Acesso em: 21 jan. 2017.

LIU, Yuee; MEJIAS, Luis. Real-time power line extraction from Unmanned Aerial System video images. In: INTERNATIONAL CONFERENCE ON APPLIED ROBOTICS FOR THE POWER INDUSTRY (CARPI), 2., 2012, Zurich. **Proceedings…** Zurich: IEEE, 2012. p.52-57. Disponível em: <http://dx.doi.org/10.1109/carpi.2012.6473348>. Acesso em: 22 jan. 2017.

LIU, Yuee; MEJIAS, Luis; LI, Zhengrong. Fast power line detection and localization using steerable filter for active UAV guidance. In: INTERNATIONAL SOCIETY FOR

PHOTOGRAMMETRY & REMOTE SENSING (ISPRS2012), 12., 2012, Melbourne. **Proceedings…** Melbourne: ISPRS, p.1-5, jan. 2012.

LUO, Yifu. A hough transform based approach to acoustic shot localization. In: INTERNATIONAL CONFERENCE ON INFORMATION FUSION (FUSION), 19., 2015, Heidelberg. **Proceedings…** Heindelberg: ISIF, 2015. p.1420-1426.

QUIGLEY, Morgan; CONLEY, Ken; GERKEY, Brian. ROS: an open-source Robot Operating System. **Icra Workshop on Open Source Software,** New York, v. 3, p.1-5, 2009.