اَلجُمهُورِيَّة الجَزَائِرِيَّة الدِيمُقرَتِيَّة الشَعبِيَّة
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
–T L E M C E N–

المدرسة العليا في العلوم التطبيقية
École Supérieure en Sciences Appliquées

وِزَارَة التَعلِيم العَالِي وَالبَحث العِلمِي
اَلمَدرَسَة العُلِيَا فِي العُلُوم التَطبِقِيَة
–تِلِمسَان–

Graduation Thesis

For the attainment of the master's degree

Field : Automatic Control
Speciality : Automatic Control

Presented by : KORICHI Mohamed Yasser
LAZREG Abdellah

Title

Application of Intelligent Vision Systems for the Automation CNC Engraving

Publicly defended, on 07/01/2024 , before the jury composed of :

Mr.CHERKI Ibrahim	Professeur	ESSA-Tlemcen	President
Mr. ABDELLAOUI Ghouti	MCA	ESSA-Tlemcen	Director
Mrs. OUHOUD Amina	MCB	ESSA-Tlemcen	Co-Director
Mr. MEGNAFI Hichem	MCA	ESSA-Tlemcen	Reviewer 1
Mrs. DIDI Ibtissem	MCA	ESSA-Tlemcen	Reviewer 2
Mr. CHIALI Anis	MCA	ESSA-Tlemcen	Incubator representative
Mr. KANOUN Ahmed Ali	MRA	CDS-Oran	Socio-economic Partner
Mrs. Merad Faiza	MRA	CDS-Oran	Guest

Academic year : 2023-2024


بسم الله الرحمن الرحيم

Dedications

I dedicate this modest work to my entire family
in particular to my dear father and mother and to all those who helped me

Yasser

I express my deepest gratitude to my parents for their unwavering support throughout our school years, and to my sisters and extended family.

I extend my sincere appreciation to my professors and instructors, whose guidance and wisdom have been instrumental in inspiring and shaping my academic journey.

I am thankful to my friends and colleagues for their camaraderie and shared experiences, which have made this journey unforgettable and profoundly enriched it.

Finally, to all who share a belief in the transformative potential of knowledge and innovation, I trust that this work will contribute to our collective understanding and progress.

Abdellah

THANKS

The first and last thing is for Allah, who gave us the ability to complete this work.

We would like to express our sincere thanks to my supervisors:

- *Mr. **ABDELLAOUI Ghouti**, professor and head of the electronics department at the Higher School of Applied Sciences in TLEMCEN, who played a major role in our progress with his constant advice and guidance.*
- *Mrs. **OUHOUD Amina**, a professor at the Higher School of Applied Sciences in TLEMCEN, for the advice, encouragement, and constant attention she gave us in difficult moments.*

Our thanks also go to:

- *Mr. **CHERKI Ibrahim**, professor at the Higher School of Applied Sciences in TLEMCEN, who honored us by chairing the defense jury.*
- *Mr. **MEGNAFI Hichem**, Head of the department at the Higher School of Applied Sciences in TLEMCEN, for providing us with advice and suggestions in developing our project.*
- *Mr. **CHIALI Anis**, a lecturer at the Higher School of Applied Sciences in TLEMCEN, who honored us as a member of the jury.*
- *Mrs. **DIDI Ibtissem**, a lecturer at the Higher School of Applied Sciences in TLEMCEN, who honored us as a member of the jury.*
- *Mr. **KANOUN Ahmed Ali**, is an associate research scientist and member of the Satellite Development Center, who honored us as a member of the jury.*

We would also like to thank:

- *Mr. **ADJIM Ramz-Eddine Abderrazak**, FabLab Engineer, for his patience and availability and provide permanent assistance.*
- *Mr. **KAID Abdelatif**, the laboratory engineer, for his good understanding with us throughout this period.*

ملخص

في عالم يعد فيه الابتكار والتحسين المستمر أمرًا ضروريًا، تحتل ماكينات التحكم الرقمي(CNC) مكانة بارزة في تكنولوجيا التصنيع. ولتحقيق أقصى استفادة من هذه الماكينات، من الضروري تطوير أنظمة متكاملة تسهل استخدام هذه الماكينات، مما يتيح لنا إمكانيات جديدة لإنشاء منتجات عالية الجودة. كانت هذه المسألة هي دافعنا خلال هذا العمل. لذلك، تم تناول ثلاثة جوانب رئيسية.

لمحة عامة عن تقنيات معالجة الصور، ومقدمة عن البرمجة الرقمية مع التركيز على لغات البرمجة باستخدام الحاسب الآلي مثل G-code و M-code، وتطوير واجهة رسومية في لغة Python باستخدام Qt Designer. تدمج هذه الواجهة تقنيات معالجة الصور وتوليد ملفات G-code، مما يتيح تحويل الصور الرقمية إلى أوامر G-code للتحكم المباشر في ماكينات التحكم الرقمي باستخدام الحاسب الآلي في عمليات التصنيع.

الكلمات المفتاحية : آلة التحكم العددي بالكمبيوتر ، معالجة الصور، الصور الرقمية، البرمجة الرقمية، الكود G، الكود M، واجهات المستخدم الرسومية، Qt Designer, PyQt5

Abstract

In a world where innovation and continuous improvement are essential, CNC machines are in the background of manufacturing technology. To get the most out of these machines, it is necessary to develop integrated systems that facilitate the use of these machines, giving us new possibilities to create high-quality products. This issue has been our motivation during this work. Therefore, three main aspects are addressed.

An overview of image processing techniques, an introduction to numerical programming with a focus on CNC programming languages such as G-code and M-code, and the development of a graphical interface in Python using Qt Designer. This interface integrates image processing techniques and G-code file generation, enabling the conversion of digital images into G-code commands for direct control of CNC machines in manufacturing processes.

Keywords : CNC, image processing, digital images, numerical programming, G-code, M-code, graphical user interfaces, Qt Designer, PyQt5

Résumé

Dans un monde où l'innovation et l'amélioration continue sont essentielles, les machines CNC sont au cœur de la technologie de fabrication. Pour tirer le meilleur parti de ces machines, il est nécessaire de développer des systèmes intégrés qui facilitent l'utilisation de ces machines, en nous donnant de nouvelles possibilités de créer des produits de haute qualité. C'est cette question qui nous a motivés tout au long de ce travail. C'est pourquoi trois aspects principaux sont abordés.

Une vue d'ensemble des techniques de traitement d'images, une introduction à la programmation numérique en mettant l'accent sur les langages de programmation CNC tels que le G-code et le M-code, et le développement d'une interface graphique en Python à l'aide de Qt Designer. Cette interface intègre les techniques de traitement d'images et la génération de fichiers G-code, permettant la conversion d'images numériques en commandes G-code pour le contrôle direct des machines CNC dans les processus de fabrication.

Mots-clés : CNC, traitement d'images, images numériques, programmation numérique, G-code, M-code, interfaces graphiques, Qt Designer, PyQt5

Contents

List of Figures

List of Tables

List of Acronyms

CCD Charge Coupled Device

CMOS Complementary Metal Oxide Semiconductor

RGB Red Green Blue

JPEG Joint Photographic Experts Group

PNG Portable Network Graphics

TIFF Tagged Image File Format

GIF Graphics Interchange Format

BMP Bitmap

SVG Scalable Vector Graphics

FFT fast Fourier Transform

CNC Computer Numerical Control

USA United States of America

MIT Massachusetts Institute of Technology

NC Numerical Control

MOCN Machines Outils a Commande Numerique in franch

CAD Computer Aided Design

CAM Computer Aided Manufacturing

GUI Graphic User Interface

SOR Statement Of Requirements

CSS Cascading Style Sheets

OS Operating System

Open CV Open Source Computer Vision Library

IDE Integrated Development Environment

PPM Portable Pix-map

PGM Portable Gray-map

PBM Portable Bit-map

OOP Object Oriented Programming

SADT Structured Analysis and Design Technique

ICAM Integrated Computer Aided Manufacturing

PCB Printed Circuit Board

GENERAL INTRODUCTION

In the modern industrial context, automation and accuracy of manufacturing processes are essential to improve product quality. Technological progress has led to the emergence of computer numerical control machines (CNC) and their widespread spread in factories, representing a major innovation in modern manufacturing. These machines have brought about a major transformation in production processes and their ability to manufacture parts accurately and quickly without direct human intervention. In the modern era, quality and efficiency are essential to maintaining competitiveness. CNC machines allow production processes to run with less human intervention, which reduces costs and provides greater accuracy and efficiency. The exceptional precision of these machines is essential for industries where quality is critical, such as automotive and medicine. Moreover, the flexibility of digital control machines plays a major role in saving time, meaning that production can be changed quickly without the need to make major mechanical adjustments. CNC machines include advanced technologies represented in simulation and real-time control, which allows errors to be predicted and corrected before they occur. This not only improves the quality of the product, but also improves the overall efficiency of the manufacturing process.

Programming numerical control machines is an essential element in manufacturing, as it plays a crucial role in converting digital designs into physical products using software instructions, often in the form of the G code, to control the movement and operation of manufacturing tools such as milling machines, laser cutting machines, and even 3D printers. These instructions are generated from digital models created by computer-aided design (CAD) and computer-aided manufacturing (CAM) programs, allowing the transition from concept to finished product. This process not only saves time but also reduces the risk of human error.

Although CNC machines play a pivotal role in achieving high levels of accuracy and efficiency, this always requires the presence of skilled and trained operators so that they can operate these machines effectively and configure the numerical control system. Their role begins with understanding the drawings and technical specifications of the parts to be manufactured by using several programs to create accurate digital models and then converting them into G code instructions. Although CNC machines are largely automated, they require human expertise to improve the programs and ensure quality. Operators must have a strong understanding of the programming languages of the numerical control system, especially the G code and the M code. This matter can be considered a problem because understanding these languages and dealing with these machines is somewhat complicated. In order to take full advantage of the capabilities of these machines, it is necessary to develop an integrated system that facilitates the conversion of visual data into code capable of controlling CNC machines. Thus, dealing with these machines becomes simple and easy for operators, even if they do not have an extensive background in the field of numerical control or programming.

The master's thesis addresses this problem by developing a graphical interface capable of converting images to G code and facilitating communication between the operator and the machine. We will learn about it by presenting three main chapters.

- Chapter One: Image processing, which is an essential step in converting designs and images into code by improving and increasing image quality to facilitate to extract information from them.
- Chapter Two: Numerical control programming, which plays a crucial role in manufacturing processes by determining the paths of machine tools and their procedures, so that the higher the quality of numerical control programming, the greater the accuracy of the machine and the quality of the product.
- Chapter Three: Converting images to G code through a graphical interface programmed in Python with the aim of facilitating dealing with this type of machine and increasing productivity.

In a world where innovation and continuous improvement are essential, CNC machines are at the forefront of manufacturing technology, opening up new possibilities for creating complex, high-quality products.

Chapter 1

IMAGE PROCESSING

In this chapter we will learn about the most important points related to the field of image processing, starting from digital images, their types and characteristics, up to analyzing and processing images.

Introduction

Image analysis currently affects many fields, with objectives as varied as diagnostic assistance for medical images, artificial vision in robotics or the analysis of earth resources from images taken by satellite. The goal of processing these images is both simple in concept and difficult in execution. Simple indeed, since it involves recognizing objects that our visual system perceives quickly, at least for the majority of them. Difficult, however, because from the large quantity of information contained in the image, it is necessary to extract elements relevant to the intended application, independently of the quality of the image. Image analysis has therefore acquired powerful tools and methods from fields as varied as mathematics, signal processing, and computer science. Therefore, image processing is the culmination of all methods and techniques applied to images with the aim of making this operation feasible, more straightforward, efficient, and pleasant, enhancing the image's visual aspect, and extracting relevant information from it.[7]

1.1 What is an image ?

An image is a representation of a person or object created by painting, sculpture, drawing, photography, video, etc. It's also a structured collection of data that, once displayed on the screen, has meaning for human eyes. The shape of this function can be described as a continuous analog brillance function I(x,y), defined in a born domain. I is a function of color and light intensity, while x and y are the spatial coordinates of a point in the image. Due to this feature, the image cannot be used by the machine, necessitating its numericization.[33]

1.1.1 Image acquisition

To acquire digital images, the acquisition device must have lighting, an optical element, a processing unit or processor and a backup memory or display screen. The acquisition device captures light information and converts it into an analog electrical signal, thanks to photodetectors organized in arrays (linear cameras) or matrices (matrix cameras). Among the best-known sensors are CCD (Charge-Coupled Device) and CMOS (Complementary Metal-Oxide Semiconductor). Figure 1.1 shows chain of acquire digital images [9]

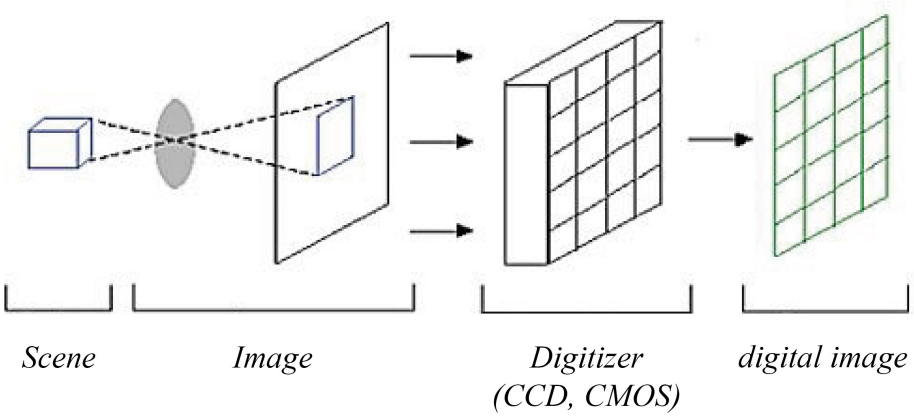


Figure 1.1 *The image acquisition chain[9].*

1.1.2 Representation of images

Figure 1.4 shows the representation of a digital image in terms of lenghth and width and grayscale

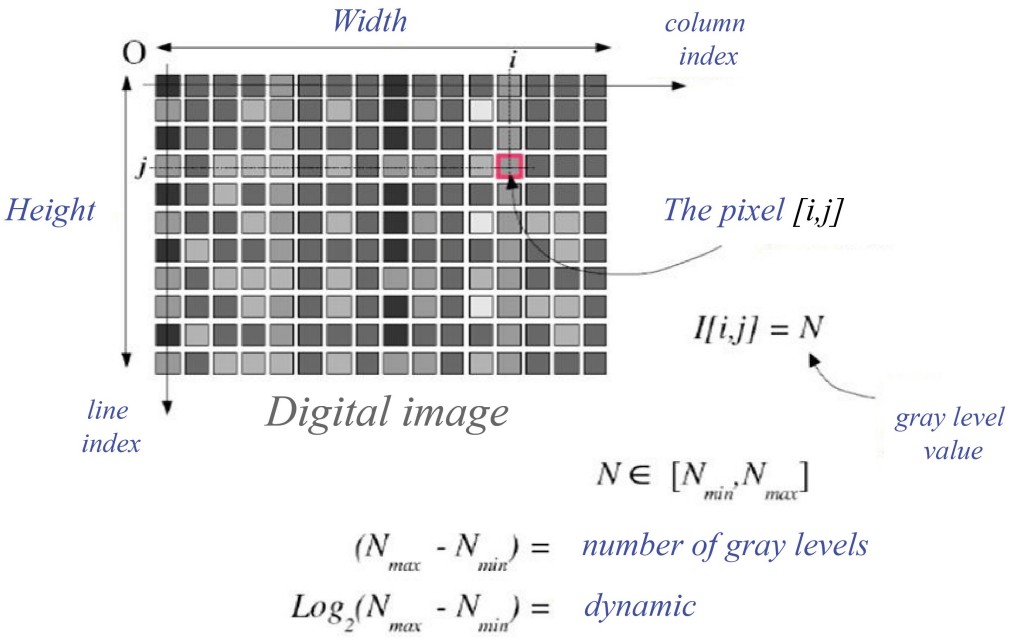
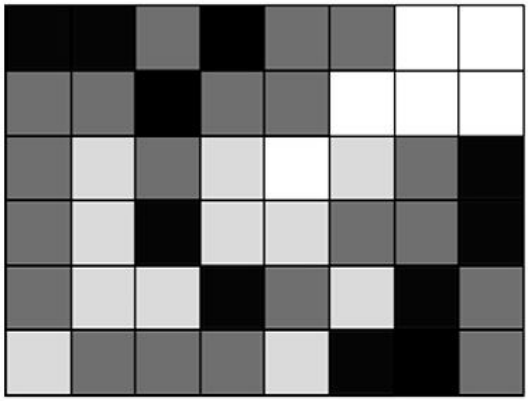


Figure 1.2 *representation of a digital image[23].*

1.1.2.1 Grayscale images

The gray level is the value of light intensity at a point. The color of a pixel can take on values ranging from black to white, passing through a finite number of intermediate levels. To represent grayscale images, each pixel in the image can be assigned a value corresponding to the amount of light reflected. This value can be between 0 and 255, for example. This means that each pixel is no longer represented by a bit, but by a byte. To achieve this, the hardware used to display the image must be capable of producing the corresponding gray levels.[33]



89	89	120	20	120	120	255	255
120	120	20	120	120	255	255	255
120	181	120	181	255	181	120	89
120	181	89	181	181	120	120	89
120	181	181	89	120	181	89	181
181	120	120	120	181	89	29	181

(a) *Grayscale image.* (b) *Grayscale pixel.*

Figure 1.3 *Grayscale representation[9].*

The number of gray levels depends on the number of bits used to describe the "color" of each pixel in the image. The greater this number, the more levels possible levels (as shown in the figure 1.3).

1.1.2.2 Colored images

Although it's sometimes useful to be able to represent images in black and white, multimedia applications most often use color images. Color is represented in the same way as monochrome images, but with a few special features. First, a representation model must be chosen. Colors can be represented using their primary components. Light-emitting systems (computer screens, etc.) are based on the principle of additive synthesis: colors are composed of a mixture of red, green and blue (RGB model) shown figure 1.4.[33]

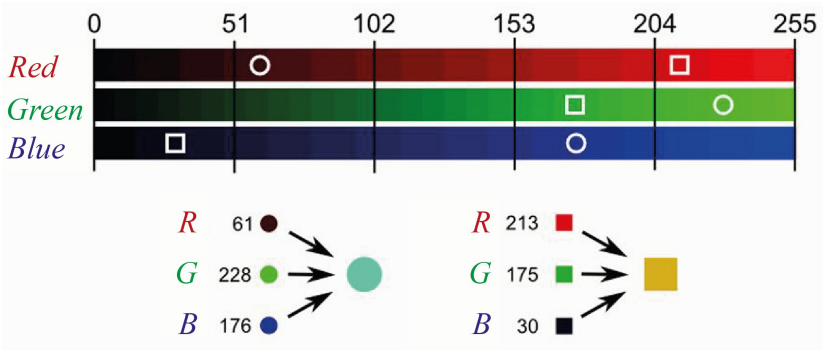


Figure 1.4 *RGB representation [9].*

1.1.3 Characteristics of a digital image

The image is a structured set of information characterized by the following parameters:

a) **Pixel** : A contraction of "Picture Element", the pixel is the smallest point in the image. It is a calculable entity that can be structured and quantized. If the bit is the smallest unit of information that a computer can process, the pixel is the smallest element that can manipulate display or printing hardware and software. The letter A, for example, can be displayed as a group of pixels, as shown in the figure 1.5.[33]

Figure 1.5 *The letter A displayed as a group of pixels.[33].*

b) **Size** : This is the image size. It takes the form of a matrix whose elements are numerical values representing light intensities (pixels). The number of rows in this matrix multiplied by the number of columns gives us the total number of pixels in an image.[33]

c) **Resolution** : Resolution is defined by the number of pixels per unit length of the image to be scanned, in dpi (dots per inch). We speak of definition for a screen and resolution for an image. The greater the number of pixels per unit length of the image to be scanned, the greater the amount of information describing the image, and the higher the resolution (and the greater the weight of the image).
The resolution of an image corresponds to the level of detail that will be represented on it. For digitization, consider the following 2 equations[33]

- (X*resolution) = x pixels
- (Y*resolution) = y pixels

Such as :

- X and Y represent the size (inch or cm, 1 inch=2.54 centimeters) of the structure to be digitized.
- resolution represents the scanning resolution.
- x et y represent the size (in pixels) of the image.[33]

d) **Noise** : Noise in an image is considered to be a phenomenon of sudden variation in the intensity of a pixel in relation to its neighbors as shown the figure 1.6, and is caused by the illumination of the sensor's optical and electronic devices.[33]

Figure 1.6 *The difference between an image with noise and an image without noise[24].*

e) **Histogram** : The grayscale or color histogram of an image is a function that gives the frequency of appearance of each grayscale (color) in the image. To reduce quantization error, to compare two images obtained under different lighting conditions, or to measure certain properties of an image, the corresponding histogram is often modified. It provides a wealth of information on the distribution of gray levels (color) and shows where the majority of gray levels (color) are distributed in the case of an image that is too light or too dark. It can be used to improve the quality of an image (image enhancement) by introducing a few modifications, in order to extract useful information from it. Figure 1.7 shows an image with its histogram.[10]

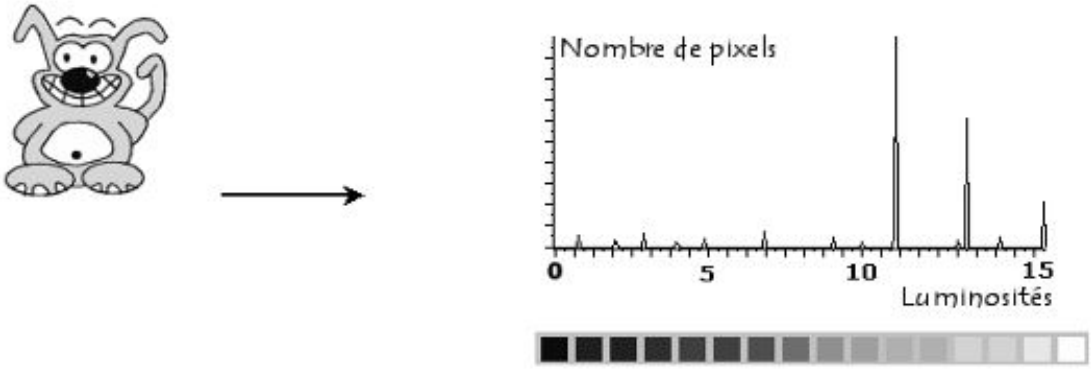


Figure 1.7 *Image with histogram[10].*

f) **Contours and textures** : Contours represent the boundary between image objects, or the limit between two pixels as shown figure 1.8 ,whose gray levels represent a significant difference.
Textures describe their structure. Contour extraction consists in identifying the points in the image that separate two different textures.[17]

Figure 1.8 *Edges detection of image.*

g) **Luminance** : This is the degree of brightness of the image points. It is also defined as the quotient of the luminous intensity of a surface by the apparent area of this surface, for a distant observer, the word luminance is substituted for the word brilliance, which corresponds to the brightness of an object. Good luminance is characterized by

- Brilliant images.
- Good contrast: avoid images where the contrast range tends towards white or black; these images lead to loss of detail in dark or bright areas.
- The absence of parasites.[12]

1.1.4 Image types

Images belong to two main families: bitmap (image-bit) or raster and vector. Whereas a vector image is described using mathematical equations, a bitmap image is made up of pixels, and is therefore reduced to a matrix of points. While vector images can be manipulated with great ease, changes of size, for example, to a size, for example, to a bitmap image. Figure 1.9 shows the difference between the two families.[33]

(a) *vector image.* (b) *matrix image.*

Figure 1.9 *Difference between vector and matrix images[33].*

1.1.5 Image compression formats

Image compression formats are necessary to reduce the size of image files and maintain acceptable quality. Below are some of the most commonly used image compression formats.

1.1.5.1 Joint photographic experts group (JPG ou JPEG):

JPEG is an excellent way to store 24-bit photographic images, such as those used for imaging and multimedia applications. JPEG 24 bit (16 million color) images are superior in appearance to 8-bit (256 color) images on a Video Graphics Array (VGA) display and are at their most spectacular, when using 24-bit display hardware (which is now quite inexpensive) . JPEG was designed to compress, color or gray-scale continuous-tone images of real-world subjects, photographs, video stills, or any complex graphics, that resemble natural subjects. Animations, ray tracing, line art, black-and-white documents, and typical vector graphics don't compress very well under JPEG and shouldn't be expected to. And, although JPEG is now used to provide motion video compression, the standard makes no special provision for such an application. [11]

1.1.5.2 Joint photographic experts group 2000 (JPEG 2000)

JPEG 2000 is a wavelet-based image compression standard. It was formed by the Joint Photographic Experts Group committee with the intention of overriding their original discrete cosine transform based JPEG standard.
JPEG 2000 has superior compression ratios than JPEG. It does not undergo the uniform

blocks, so distinctiveness of JPEG images with very high compression rates. But it generally makes the image more blurred that JPEG. [18]

1.1.5.3 Portable network graphics (PNG)

PNG is a file format for lossless image compression.Typically, an image in a PNG file can be 10% to 30% more compressed than in a GIF format . It allows making a trade-off between file size and image quality when the image is compressed. It produces smaller files and allows more colors. PNG also supports partial transparency. Partial transparency can be used for many useful purposes, such as fades and antialiasing for text.[11]

1.1.5.4 Tagged image file format (TIFF)

The TIFF is a flexible format that can be lossless or lossy compression. It normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively. The details of the image storage algorithm are included as part of the file. In practice, TIFF is used almost exclusively as a lossless image storage format that uses no compression at all. TIFF files are not used in web images. They produce big files, and more importantly, most web browsers will not display TIFFs.[11]

1.1.5.5 Graphics interchange format (GIF)

GIF is useful for images that have less than $256 - 2^8$ colors, grayscale images and black and white text. The primary limitation of a GIF is that it only works on images with 8-bits per pixel or less, which means 256 or fewer colors. Most color images are 24 bits per pixel. To store these in GIF format that must first convert the image from 24 bits to 8 bits. GIF is a lossless image file format. Thus, GIF is “lossless” only for images with 256 colors or less. For a rich, true color image, GIF may “lose” 99.998% of the colors. It is not suitable for photographic images, since it can contain only 256 colors per image.[11]

1.1.5.6 Bitmap (BMP)

The Bitmap (BMP) file format handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large; advantage is that their simplicity, wide acceptance, and use in Windows program.[11]

1.1.5.7 Scalable vector graphics (SVG)

The SVG (Scalable Vector Graphics) standard allowsrepresenting complex graphical scenes by a collection of graphic vectorial-based primitives, offering several advantages with respect to classical raster images such as: scalability, resolution independence, etc. In this paper we present a full comparison between some advanced raster to SVG algorithms: SWaterG, SVGenie, SVGWave and some commercial tools. SWaterG works by a watershed decomposition coupled with some ad-hoc heuristics, SVGenie and SVGWave use a polygonalization based respectively on Data Dependent and Wavelet triangulation. The results obtained by SWaterG, SVGenie and SVGWave are satisfactory both in terms of perceptual measured quality and compression ratio.[6]

Table 1.1 shows the difference between the various image formats

Table 1.1 *Image compression formats.*

	Type (matrix/ vector)	Compression of data	Number of colors supported	Display progressive	Animation	Transparence
JPEG	matrix	yes, adjustable (with loss)	16 millions	Yes	No	No
JPEG2000	matrix	yes, with or without loss	32 millions	Yes	Yes	Yes
GIF	matrix	Yes, No loss	256 maxi (pallet)	Yes	Yes	Yes
PNG	matrix	Yes, without loss	Palletized (256 colors or less) or 16 million	Yes	No	yes (Alpha layer)
TIFF	matrix	Compression or not with or without losses	from monochrome to 16 million	No	No	Yes (Alpha layer)
SVG	vector	compression possible	16 million	* does not apply *	Yes	Yes (by nature)

1.2 What is digital image processing ?

Image processing is the discipline of analyzing, enhancing, extracting or summarizing information from an image through the application of algorithms.[24]

1.2.1 Image processing history

Image processing began to be studied in the 1920s for the transmission of images by the submarine cable from New York to London. Harry G. Bartholomew and Maynard D. McFarlane perform the first image scanning with data compression to send faxes from London to New York. The transfer time thus goes from more than a week to less than three hours. There was no real development until the post-war period.

Signal processing gained importance towards the end of World War II with the arrival of radar. Oil prospecting also contributes greatly to the development of signal processing techniques.

The real boom in image processing did not occur until the 1960s when computers began to be powerful enough to work on images. Shortly after, the rediscovery of the fast Fourier transform (FFT) revolutionized the field, making it possible to manipulate the frequency content of signals on a computer. However, most of the research at this time still focused on improving images and their compression.

In 1980, David Marr was the first to formalize edge detection in a precise manner (D. Marr and E. Hildreth: Theory of Edge Detection, Proc. R. Soc. London, B 207, 187-217, 1980). During the 1980s, a real craze emerged for image processing and especially for image understanding by expert systems. The ambitions were much too great, the failure was all the more bitter.

The 1990s witnessed the constant improvement of operators. Medical research is becoming a very large demand for image processing to improve diagnoses made using numerous medical imaging techniques, the main technique being MRI. Advertisers, then the general public, became familiar with image editing using Photoshop software, and image processing for

aesthetic purposes became more widespread with the appearance of other dedicated software (The Gimp, Paint Shop Pro). Finally, the decade ends with the craze for wavelets and multimodal images.[3]

1.2.2 Image processing fields

1.2.2.1 Image enhancement

adjusting images to make them more suitable for display or analysis. Image enhancement techniques include morphological operator filtering, histogram equalization, Wiener filter noise removal and contrast adjustment.[24]

1.2.2.2 Image registration

aligning images to build a panoramic image, or merging images from different sources or devices. This takes into account common problems such as rotation and change of scale when superimposing images.[24]

1.2.2.3 Image analysis

isolating objects of interest in order to extract information and statistics. Some examples of image analysis techniques are edge detection, extraction of objects of a certain shape such as circles, and measurement of properties in regions of interest.[24], and improve lighting as shown in image 1.10

Figure 1.10 *Enhancing low-light images through haze removal.[24].*

1.2.3 Image pre-processing

The main goal of pre-processing is to improve the quality of the image to make it ready for further processing by removing or reducing irrelevant and redundant parts.[32]
Table 1.2 lists common image pre-processing operations, with examples from each of the four descriptor families, illustrating both differences and commonality among these image pre-processing steps, which can be applied prior to feature description.[19]

Table 1.2 *Possible Image Pre-Processing Enhancements and Corrections[19].*

Image Pre-Processing	Local Binary (LBP, ORB)	Spectra (SIFT, SURF)	Basis Space (FFT, Code books)	Polygon Shape (Blob Metrics)
Illumination corrections	X	X	X	X
Blur and focus corrections	X	X	X	X
Filtering and noise removal	X	X	X	X
Thresholding				X
Edge enhancements		X		X
Morphology				X
Segmentation				X
Region processing and filters		X	X	X
Point processing		X		X
Math and statistical processing		X		X
Color space conversions		X	X	X

1.2.3.1 Corrections

During image pre-processing, there may be artifacts in the images that should be corrected prior to feature measurement and analysis. Here are various candidates for correction[19]

- **Sensor corrections** : these include dead pixel correction, geometric lens distortion, and vignetting.

- **Lighting corrections** : Lighting can introduce deep shadows that obscure local texture and structure; also, uneven lighting across the scene might skew results. Candidate correction methods include rank filtering, histogram equalization, and remap.

- **Noise** : This comes in many forms, and may need special image pre-processing. There are many methods to choose from.

- **Geometric corrections** : If the entire scene is rotated or taken from the wrong perspective, it may be valuable to correct the geometry prior to feature description. Some features are more robust to geometric variation than others.

- **Color corrections** : It can be helpful to redistribute color saturation or correct for illumination artifacts in the intensity channel. Typically color hue is one of the more difficult attributes to correct, and it may not be possible to correct using simple gamma curves and the sRGB color space. [19]

1.2.3.2 Enhancements

Enhancements are used to optimize for specific feature measurement methods, rather than fix problems. Familiar image processing enhancements include sharpening and color balancing. [19]

- **Scale-space pyramids** : When a pyramid is constructed using an octave scale and pixel decimation to sub-sample images to create the pyramid, sub-sampling artifacts and jagged pixel transitions are introduced. Part of the scale-space pyramid building process involves applying a Gaussian blur filter to the sub-sampled images, which removes the jagged artifacts.

- **Illumination** : In general, illumination can always be enhanced. Global illumination can be enhanced using simple LUT remapping and pixel point operations and histogram equalizations, and pixel remapping. Local illumination can be enhanced using gradient filters, local histogram equalization, and rank filters.

- **Blur and focus enhancements** : Many well-known filtering methods for sharpening and blurring may be employed at the pre-processing stage. For example, to compensate for pixel aliasing artifacts introduced by rotation that may manifest as blurred pixels which obscure fine detail, sharpen filters can be used to enhance the edge features prior to gradient computations. Or, conversely, the rotation artifacts may be too strong and can be removed by blurring.[19]

1.2.4 Filtering

Image filtering consists in eliminating noise to facilitate image processing and interpretation. Noise generally arises during the image acquisition and digitization phase. Image filtering is the convolution of the image function, **I(x, y)**, with a function, **f (x, y)**, called the impulse response of the filter. It simply involves replacing each gray level with a linear combination of the gray levels of neighboring points. The coefficients of this combination are given by the impulse response function. In the continuous case, the filtered image is given by[29]:

$$I(x, y) = (f * I)(x, y) \tag{1.1}$$

$$I(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') I(x - x', y - y')\, dx'\, dy' \tag{1.2}$$

In the discrete case, the domains of 1 and f are bounded. If the domain of I is [-N/2, +N/2] and the domain of f is [-k/2, +k/2], we necessarily have $k \leq N$. The convolution is therefore written:

$$I(x, y) = (f * I)(i, j) \tag{1.3}$$

$$I(x, y) = \sum_{i'=-\frac{k}{2}}^{i'=\frac{k}{2}} \sum_{j'=-\frac{k}{2}}^{j'=\frac{k}{2}} f(i', j') I(i - i', j - j') \tag{1.4}$$

Equation 1.4 translates into the calculation of the $matrix^2$ scalar product between a window of k x k around the pixel and a k x k matrix containing the given coefficients

1. Convolution is defined by the operator : *
2. The matrix scalar product is defined by the operator : $\bigoplus$

by the impulse response function. This last matrix is called the mask, and the filtering operation is called masking. parameter k is called filter parameter. If all mask values are equal to 1/k filtering consists in replacing the value of each pixel by an average taken over a k x k window around that pixel. This is referred to as the mean filter. Figure 1.11 shows the filtering process.[29]

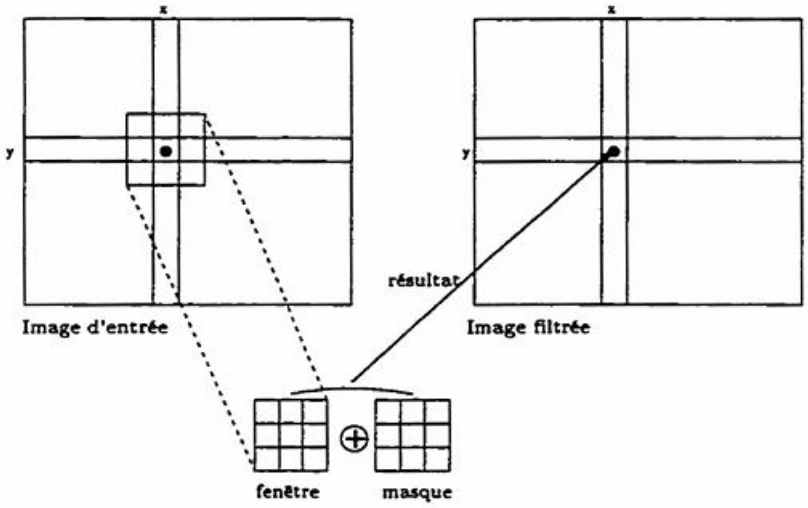


Figure 1.11 *Filtering process[29].*

1.2.5 Image segmentation

Image segmentation is a process of partitioning an image into a multiple number segments, so it is a method to classify the pixels of an image correctly in a decision oriented application. Therefore, we can state that the objective of image segmentation is to simplify or change the representation of an image or convert the information of an image into a more meaningful form so that it make it easier for further analysis. It divides an image into a number of specific regions such that the pixels exhibits high similarity in each regions and between the regions they have high contrast. It is a valuable tool in many fields and applicable in many application like health care, medical image processing, traffic image, pattern recognition etc. There are different techniques and approaches for image segmentation like threshold based, graph based, and morphological base, edged based, clustering based, neural network based etc. All these methods have their own advantages and disadvantages and therefore, one have to choose the algorithm based on the needs from their own perspective. From these techniques, one of the most used efficient method is clustering method. Again, clustering techniques can be divided into different types like K-means clustering, Fuzzy C-means clustering, Subtractive clustering methods etc.[13]

1.2.5.1 Segmentation techniques

There are many methods for segmenting an image that have been recognized by scientists and researchers. Therefore, there are several such techniques that are quite popular, important and are regularly used for image segmentation. These are classified as follows: [13]

1. Thresholding based segmentation
2. Region based segmentation
    - Region growing
    - Region merging and splitting

3. Edge based segmentation
4. Clustering based segmentation
5. Bayesian based segmentation
6. Classification based segmentation

1.2.5.1.1 Threshold

Thresholding is the simplest image segmentation method. It tries to differentiate between the image background and image foreground. A thresholding procedure uses the intensity histogram and attempts to determine the intensity values called threshold value and these threshold values differentiate the desired classes. So it segment the image based on the threshold value and hence it is fast and computationally efficient method. Although it is simple and fast, it does not take into account the spatial characteristic of an image. So thresholding method is sensitive to noise and intensity inhomogeneity.

1.2.5.1.2 Region growing

Region growing is a method for extracting a connected regions of the image which consists of group of pixels with similar intensities. In this method, a point is initially defined which is known as seed point. Then all the points which are connected to seed point having same intensity as that of seed point are selected and are added to the growing regions. This procedure is repeated until no more pixel can been added to the region.

1.2.5.1.3 Region splitting

Rather than choosing initial seed as in case of region growing, image can be divided into unconnected regions and then merge again based on some condition. That means it is consists of two steps- splitting and merging step. Quad tree method is generally used in splitting.

1.2.5.1.4 Clustering

Clustering method is an unsupervised image segmentation method. It classify the image into a finite number of cluster, where the number of cluster can be user defined or can be find using an algorithm. So in this process there is no training stages, but train themselves using the available data. Based on some criteria, the pixels are grouped together and form the cluster. Initialization of values are required and these initializations plays an important role in determining the performance of the segmentation. So initialization should be done very carefully

1.2.5.1.5 Edge detection

Edge detection method detects the edge or pixels between different regions. The condition for different regions may be rapid transition of intensity. So those pixels are extracted and linked together to form a closed boundary.

1.2.5.1.6 Bayesian

Bayesian method is used for the classification purpose and it is works by considering probability in the image to construct models based on the probability that is further utilized for the class assignment of pixels in the image. There are different approaches in Bayesian method like Markov Random Field (MRF), Expectation Maximization (EM).

1.2.5.1.7 Classification

Classification method use data with known labels to partition the image feature space. In other word, classification of image done by deriving a feature space from the image. Then this feature space is further divided into different regions depending upon the function being defined in the feature space. This classification method can be both supervised and unsupervised. In supervised, the image is trained and it is manually segmented and it is used further for the automatic segmentation of new images.[13]

1.2.5.1.8 Segmentation example

Real example of image segmentation usig python language (openCV).

```
import cv2

# Loading a grayscale image
image = cv2.imread('sans bruit.JPG', 0)

# Apply thresholding to segment the image
ret, segmented_image = cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)

# Display original image and segmented image
cv2.imshow('Original Image', image)
cv2.imshow('Segmented Image', segmented_image)
cv2.waitKey(0)

# Save the results
cv2.imwrite('segmented_image.jpg', segmented_image)
cv2.imwrite('original_image.jpg', image)

cv2.destroyAllWindows()
```

Program result

Figure 1.12 *Original image.*

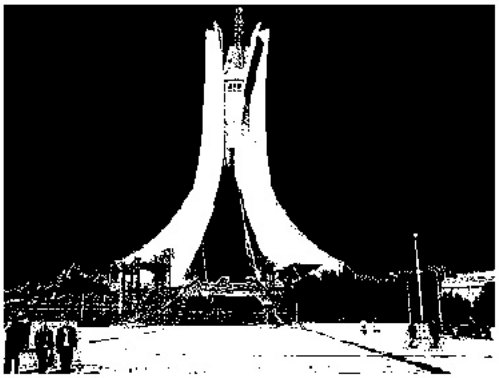


Figure 1.13 *Segmented image.*

In this exemple, we applied a segmentation method which is thresholding, so that the background of the image and the foreground of the image were differentiated (figure 1.12 and figure 1.13) through the intensity histogram which is called the threshold value

Conclusion

in conclusion, in this chapter we have provided a comprehensive overview of the field of image processing, covering various aspects from images, their types, characteristics, pre-processing to advanced processing, and then discussing some techniques such as image segmentation, object detection and edge detection.

In short, image processing is a vast and constantly evolving field that plays a crucial role in many areas of science and technology. By understanding the principles and techniques presented in this chapter, researchers and practitioners will be better equipped to meet the challenges and exploit the opportunities offered by image processing in the modern world.

After completing the field of image processing in the first chapter, we will discuss in the next chapter another field numerical programming also known as numerical control programming, which is regarded as the foundation of contemporary manufacturing.

Chapter 2

Numerical control programming

In this chapter, we will discuss how to program the computer numerical control machine using G code (setup code), which determines the movement of the machine, M code (add-on code) and some other codes.

Introduction

Numerical control, also referred to as Computer Numerical Control (CNC), is a key technology in modern manufacturing that involves using computers to control machines such as lathe machines, laser cutters and others.
CNC are much more than a simple machine they are architects of innovation and productivity engines with point technologies and unparalleled intelligence. CNC transform conceptual drawings into tangible reality with remarkable efficiency and high precision.

2.1 Numerical control

Numerical control is a technique whose applications seem destined for significant development, due to the great advantages it brings, not only in machining proper, but also in many other fields such as metrology, welding, oxy-fuel cutting, laser beam guidance, electro-erosion, waterjet cutting, etc. [14]

2.1.1 Definition

Numerical control is an automation process that allows machine tool components to be moved, based on coded alphanumeric character information. [4]

2.1.2 History

Numerical control first appeared in the USA during the last world war (the first NC machine dates back to 1942) to machine reactor parts so complex that they had to be machined by hand. The real launch of this technology took place at the Chicago Exhibition (USA) in 1952, by Cincinnati Milling Corporation, MIT, etc., but it was around 1960 that CNC machining began to enter common practice in this country. A few years later, European countries and Japan began to develop this technique in their industries. Today, CNC is still in the process of evolving, and is making a significant in manufacturing workshops. In France, around 60,000 NC machines were installed between 1990 and 1998 to modernize production facilities in small and medium-sized businesses. Japan, in turn, holds over 50% of the world market for CNC machines. In Canada, CNC machines have penetrated the vast majority of manufacturing companies, especially in Quebec. From this brief history, we can say that NC was primarily a response to a technical problem, not an economic one. [14]

2.2 Numerical control machine tools(MOCN)

"MOCN (Machines Outils a Commande Numerique in franch) that the same name chose from CNC or CNC machine tools".

The CNC (Computer Numerical Control) machine tool it is a fully or partially automatic machine to which orders are communicated by means of codes carried on a physical medium. The primary role of a CNC machine is to generate movements. It receives positioning, speed and acceleration values and, following processing, generates numerical output instructions.[20]

2.2.1 Comparing CNC machine tools with conventional machines

Figure 2.1 shows a practical example comparing production times for a part on a CNC machine and on a conventional machine.
So that the two machines were tested on the same piece, the conventional machine took about 6 hours and 55 minutes from the start of operation, while the CNC machine took 5 hours and 40 minutes because most of its time was in programming the machine, but after the second experiment it took less time, about 1 hour and 30 minutes only, so the two machines continued to work on 10 pieces and the result was that the CNC machine saves 39 hours and 30 minutes compared to the conventional machine.

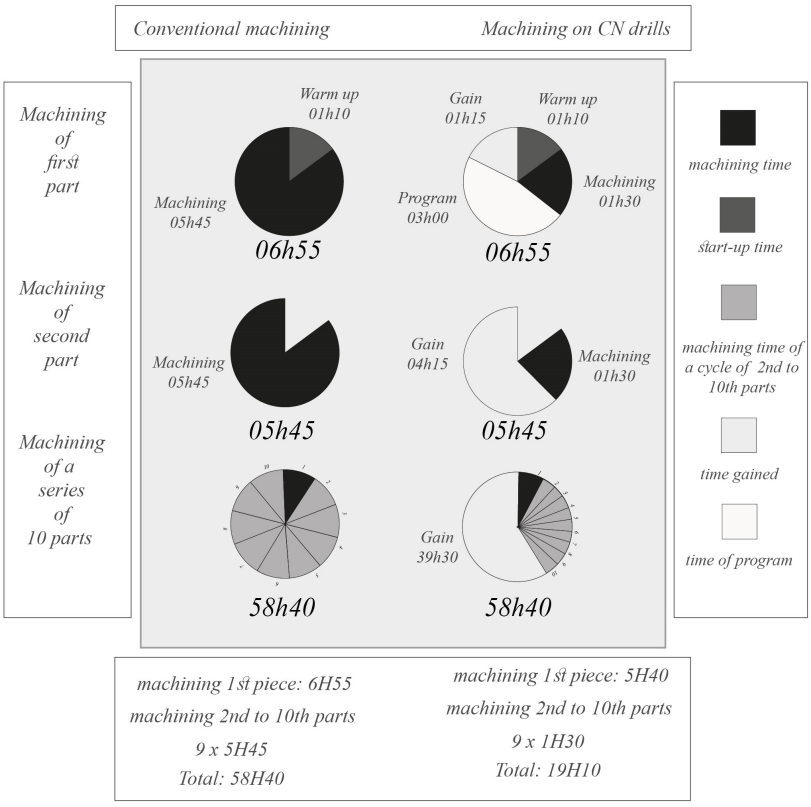


Figure 2.1 *Comparing CNC machine tools with conventional machines[26] .*

2.2.1.1 CNC advantages

These advantages are due to the technical contributions of numerical control, and are all the more perceptible as assisted programming has reduced programming time and cost constraints. It enables :

- transfer tasks that used to be carried out on site, such as taking into account tool geometry using a tangential approach, to an external workstation, thus enabling them to be carried out in hidden time.

- reduce idle times through automatic sequencing of machining sequences by positioning tools at high feed rates, automatic tool changes and continuously variable speeds.

- easily define optimal operating conditions, since speeds can be varied continuously, thus increasing tool life.

- create complex surfaces by managing movements on several axes simultaneously, thus enabling the design of parts with geometries closer to functional requirements[26].

Another way to look at the economic benefits of numerical control is to evaluate the time it takes for a machine tool to do the job it was designed to do Analyse the effective production times as shown in figure 2.2 .

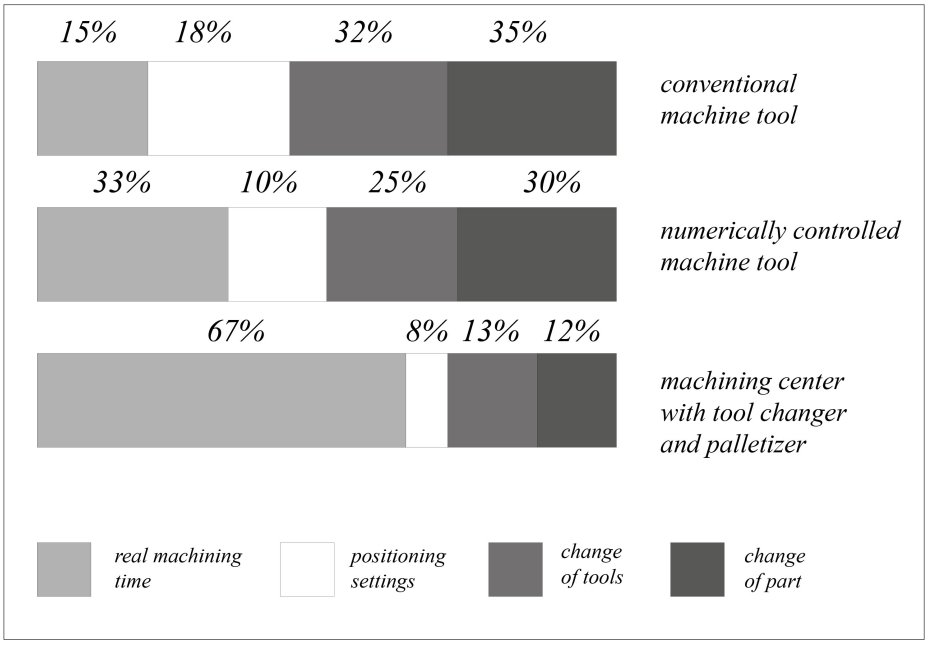


Figure 2.2 *Comparing machine productivity [26].*

2.2.2 Types of CNC programming

To manufacture a part on a CNC machine tool, you need to write a program called an NC program. This program can be written directly by the programmer (manual programming). It can also be created using a computer (assisted programming and automatic programming).[25]
There are three main types of NC programming used to drive numerically controlled machine tools (MOCN):

1. Manual Programming.

2. Computer-assisted programming.

3. Automatic programming.

2.2.2.1 Manual Programming

The first step in programming is to determine and organize the data required for the NC program, based on the machining sequence. This data is either technological or geometric. Technological data concerns the choice of machine, machining processes, tools and cutting conditions. Geometrical data relates to the choice of parameters defining the tool trajectory to obtain the desired shape. The programmer must present all elementary operations in chronological order. Then, for each sequence of operations, he must specify the corresponding technological data (cutting speed, coolant, etc.) as well as the coordinates of the characteristic points of the tool path. Once this work is complete, the programmer can write the NC program. [25]

2.2.2.2 Computer-assisted programming

Some manual programming steps require a great deal of effort before the NC program can be written. One of the most delicate steps is the calculation of the characteristic points of the tool path. Indeed, for part geometries featuring shapes such as connections, circular arcs, or even complex surfaces, calculating characteristic points becomes a tedious task. On the other hand, it's difficult to check an NC program for syntax or calculation errors. This task becomes absurd when the NC program is large. In this case, it is inevitable to correct the program at the machine. This will bring the machine to a standstill and waste valuable time that could be production. What's more, a production shop may be equipped with several machines with different controllers. The programmer must therefore master all the functions of each NC machine. Clearly, computer assistance can make an enormous contribution to saving manufacturing preparation time and getting the most out of the NC machine. Numerous NC programming aids (computer-aided NC programming) have been developed. [25]

2.2.2.3 Automatic programming

In an automatic programming system, the operator can directly exploit the possibilities offered by the computer through the graphic interface as shown in figure 2.3 . He can describe the geometry in the form of points, lines, arcs, etc., as in a definition drawing, rather than translating it into a textual representation. Use of the graphic interface also enables tool trajectories to be visualized, enabling rapid program verification and avoiding costly corrections at the machine.[25]

```
G00 S1; endstops
G00 E0; no extrusion
G01 S1; endstops
G01 E0; no extrusion
G21; millimeters
G90; absolute
G28 X; home
G28 Y; home
G28 Z; home
G00 F300.0 Z10.0;
G00 F2400.0 Y15.0;
G00 F2400.0 X10.0;
G00 F2400.0 X10.0 Y15.0;
G00 F300.0 Z5.0;

 (-----------------Layer 1 -----------------)
G00 F1500 X10.0 Y15.0;
G00 F300.0 Z5.0;
G00 F1500 X10.0 Y49.986144101346;
G00 F1500 X46.565716547901815 Y49.986144101346;
G00 F1500 X46.565716547901815 Y15.0;
G00 F1500 X10.013855898653999 Y15.0;
G00 F300.0 Z6.0;
G00 F1500 X38.584718923198736 Y15.845209817893902;
```

Figure 2.3 *Example of an automatically generated G-CODE .*

2.2.3 MOCN manufacturing process

The figure 2.4 shows the main stages involved in producing a component on a CNC system.

1. A part program is written, using G and M codes.This describes the sequence of operations that the machine must perform in order to manufacture the component.This program can be produced off-line, ie, away from the machine, either manually or with the aid of a CAD/CAM system.

2. The part program is loaded into the machines computer, called the controller. At this stage, the program can still be edited or simulated using the machine controller keypad/input device.

3. The machine controller processes the part program and sends signals to the machine components directing the machine through the required sequence of operations necessary to manufacture the component.[15]

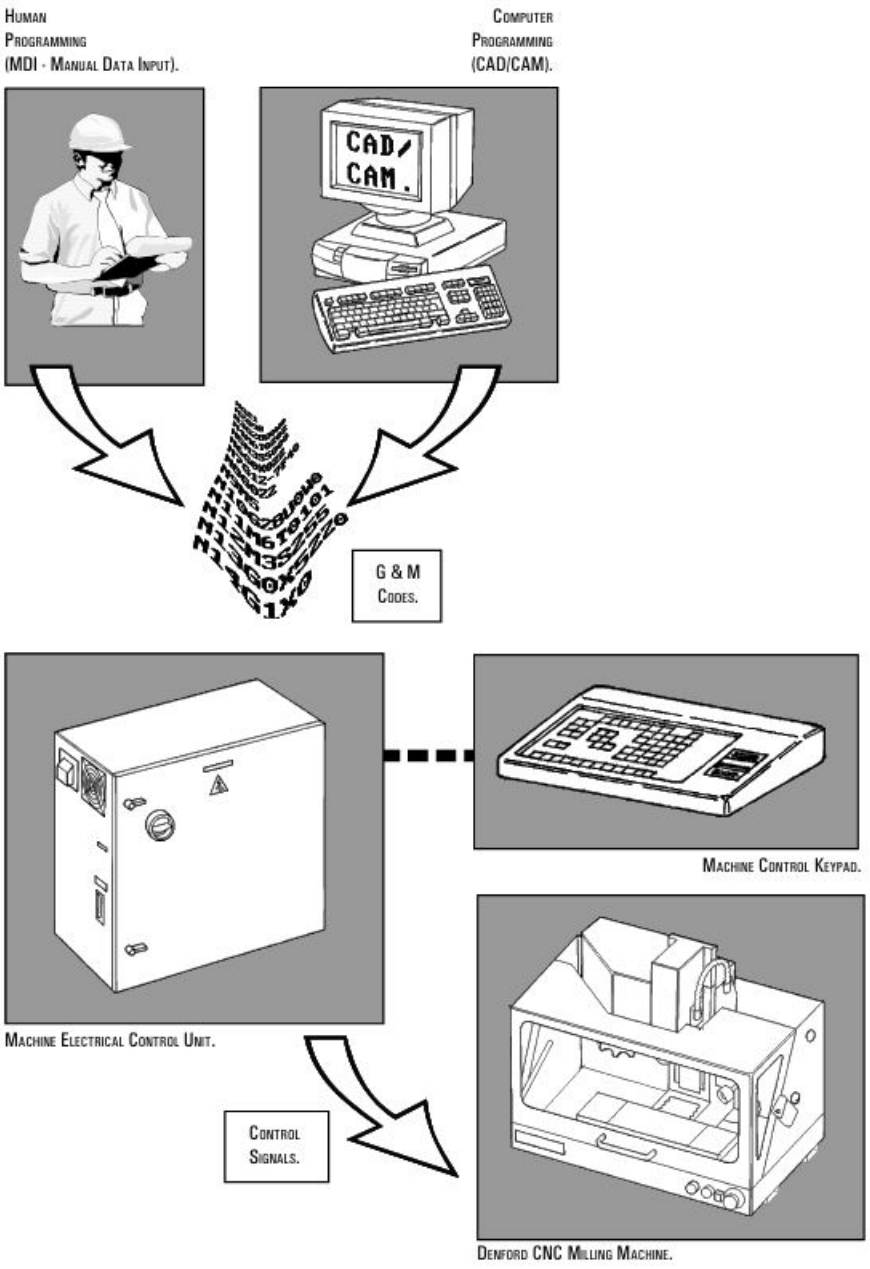


Figure 2.4 *Example MOCN manufacturing process[15].*

2.3 G-Code programming language

G-code is the programming language for digital machines and is based on lines of code. Each line (also called a block) can include commands to make the machine produce various actions. Several lines of code can be grouped together in a file to create a G-code program. [30]
A G-code is defined using the G address letter and a two digit number as follows (figure 2.5)[15].

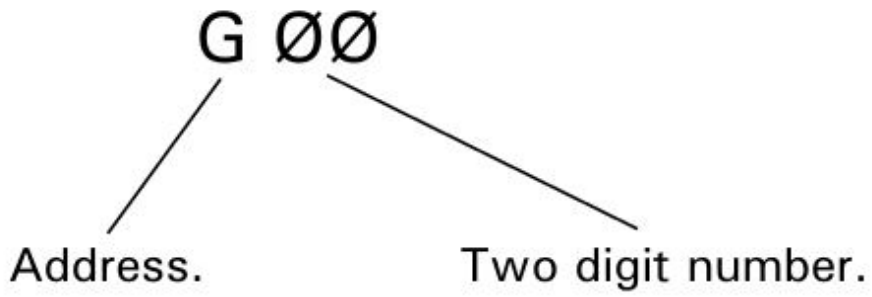


Figure 2.5 *Example of G-code definition[15].*

2.3.1 Implementation of G-code

The majority of G code commands are in alphanumeric format and begin with the letter G, which stands for geometry. G-code programming, on the other hand, can be quite difficult for machinists due to the fact that different machines interpret G codes in different formats. The presence or lack of spaces between commands, as well as the number of zeros between the letter and number in the commands, are the main differences between most computers. A machine may, for example, utilize G3 while another uses G03. Machinists must always be aware of the machine they're working on. Otherwise, command errors can cause major problems with part production.[27]
Although G is the most commonly used letter in CNC machine programming, it is not the only one. Different commands are often represented by other characters. Table 2.1 shows lists of typical G-code commands and their interpretations for milling and turning operations[27].

Table 2.1 *Function of G-codes[27].*

Variable	Description
A	It controls the tool's movement rotating the x-axis.
B	It controls the tool's movement rotating the y-axis.
C	It controls the tool's movement rotating the z-axis.
D	Cutter compensation diameter or radial offset. On lathes, D stands for depth of cut. It's used in photoplotters for angle selection and commands.
E	Precision feedrate for threading on lathes
F	Feedrate
G	Preparatory commands which tell the control what kind of motion is wanted
H	Tool length offset which corresponds to the z-axis
I	Arc center in x-axis for G02 or G03 arc commands. In some fixed cycles, it's also utilized as a parameter.
J	Arc center in y-axis for G02 or G03 arc commands. In some fixed cycles, it's also utilized as a parameter.
K	Arc center in z-axis for G02 or G03 arc commands. In some fixed cycles, it's also utilized as a parameter.
L	Fixed cycle loop counts which specification of what register to edit using G10
M	Miscellaneous function
N	Line or block number in program which the system parameters to change using G10
O	Program name
P	Serves as parameter address for various G and M codes
Q	Peck increment in canned cycles
R	Size of arc radius, or retract height in milling canned cycles
S	Speed, either spindle speed or surface speed depends on mode
T	Tool selection
U	Progressive axis relating to the x-axis (usually just lathe group A controls) also specifies dwell time (instead of "P" or "X").
V	Progressive axis relating to y-axis
W	Progressive axis relating to z-axis (usually just lathe group A)
X	Absolute or incremental position of x-axis. Also defines dwell time on some machines (instead of "P" or "U").
Y	Absolute or incremental position of y-axis
Z	Absolute or incremental position of z-axis

2.3.2 Classification of preparatory functions G

There are two types of G function, modal G-functions and non-modal G-functions.

2.3.2.1 Modal G-function

A function is said to be "modal" when it remains active (memorized) after the block in which it is written, until it is revoked. until it is revoked.
These functions belong to a family of mutually revoking G functions.
Some G-function families include a function initialized when the system is powered up. The

validity of these functions is maintained until a function of the same family revokes their validity.[5]

2.3.2.2 Non-modal G functions

Functions only valid in the block in which they are programmed (revoked at end of block). [5]

2.3.3 Composition of a part program

A part program is a list of coded instructions which describes how the designed component, or part, will be manufactured.
These coded instructions are called data a series of letters and numbers. The part program includes all the geometrical and technological data to perform the required machine functions and movements to manufacture the part.
The part program can be further broken down into separate lines of data, each line describing a particular set of machining operations. These lines, which run in sequence, are called blocks[15].

2.3.3.1 Main program structure

The Main Program is the controlling program, ie, the program first read, or accessed, when the entire part program sequence is run. This controlling program can then call a number of smaller programs into operation. These smaller programs, called Sub Programs, are generally used to perform repeat tasks, before returning control back to the main program.[15]

2.3.3.2 Sub program structure

A program which contains fixed sequences or frequently repeated patterns may be entered into memory as a Sub Program, in order to simplify the main program.
A sub program is entered into the machine controller memory in Edit Mode, in the same manner as the main program.

1. A sub program does not have a billet size definition at the top of the program listing.

2. A sub program is ended by the M99 code.

The sub program can be called into operation when the machine is set to run in Auto Mode. Sub programs can also call other sub programs into operation.[15]
Figure 2.6 shows the example of Composition of a part program (main program and sub program) structure.

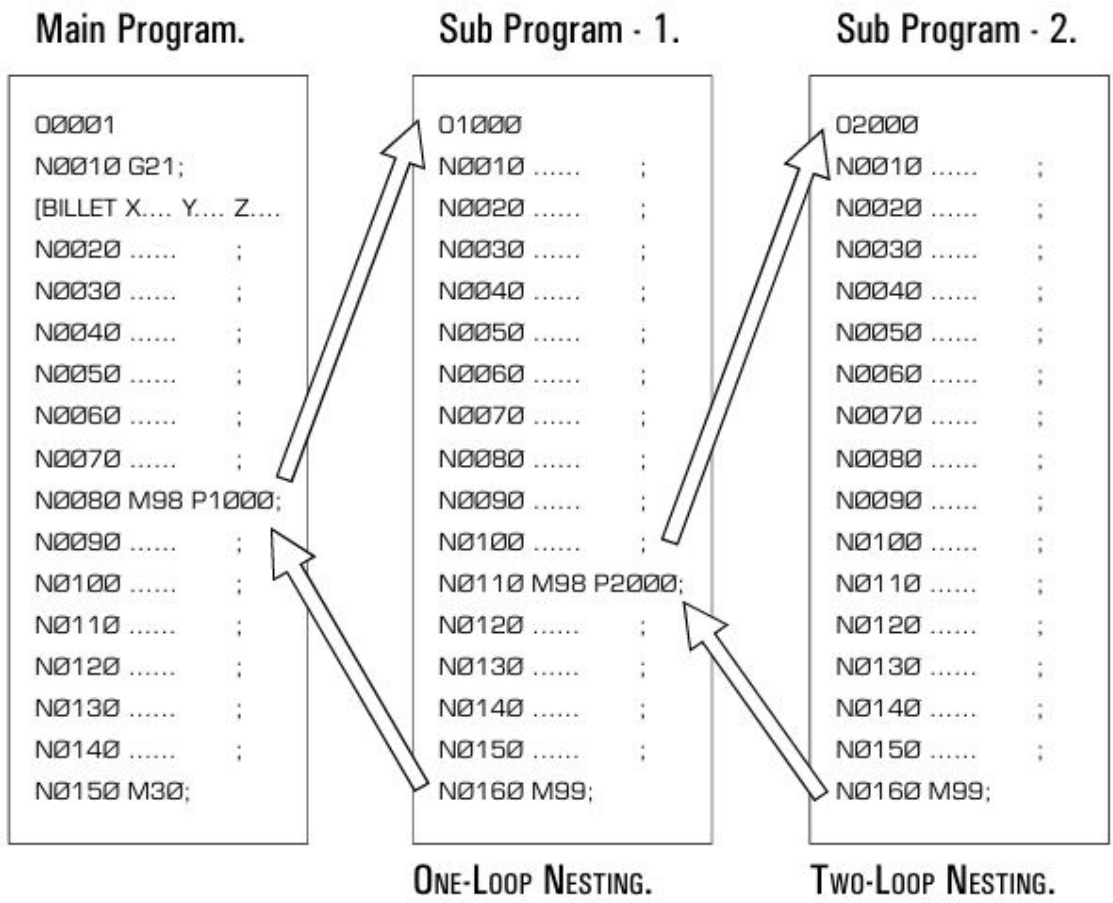


Figure 2.6 *Composition of a part program[15].*

2.3.4 Commonly used G-code

G-Code or preparatory code is the language used to control CNC machines. It's one type of CNC .
G-code includes many commands that we will display in table 2.3.

Table 2.3 *Commonly used G-code[22].*

Code	Function
G00	Rapid traverse motion: this is used for non-cutting rapid moves of the machine axis or rapid retract moves after cuts have been completed
G01	Linear interpolation motion : used for cutting in a straight line under a controlled feedrate
G02	Circular interpolation (Clockwise)
G03	Circular interpolation (Counterclockwise)
G04	Dwell
G17	Circular motion XY plan selection
G18	Circular motion XZ plan selection
G19	Circular motion YZ plan selection
G20	Verify inch coordinate positions
G21	Verify metric coordinate positions
G28	Machine home (rapid traverse) G91 is required for rapid move to the G28 reference point
G40	Cutter compensation CANCEL
G41	Cutter compensation LEFT of the programmed path
G42	Cutter compensation RIGHT of the programmed path
G43	Tool length compensation
G49	Tool length compensation CANCEL
G53	Positions the machine axis relative to machine home, It is non modal
G54	Work coordinate #1 (part zero offset location)
G80	Canned cycle CANCEL
G81	Drill canned cycle
G82	Spot drill canned cycle
G83	Peck drill canned cycle
G84	Tapping canned cycle
G90	Absolute programming positioning
G91	Incremental programming positioning
G98	Canned cycle initial point return
G99	Canned cycle rapid (R) plane return

2.3.5 Displacement programming

Displacement G-code are instructions used in CNC machines to control the movement and operation of the cutting tool.

2.3.5.1 G00 rapid positioning / traverse

The G00 code executes a non cutting movement, at a rapid feedrate (shown figure 2.7), to a specific co-ordinate position in the working area (operating under absolute coordinate movement) or when a certain distance from a previously stated position (under incremental coordinate movement) is programmed[15].

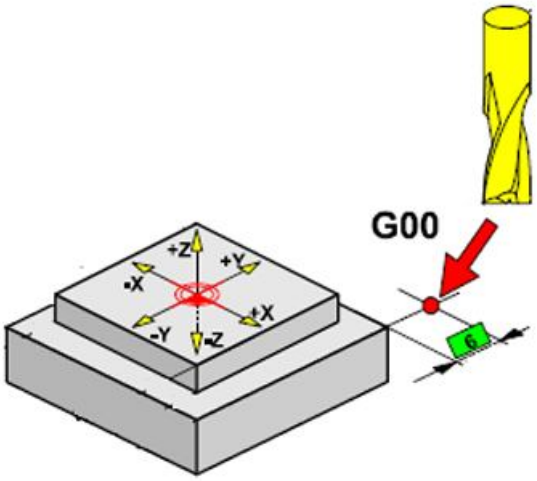


Figure 2.7 *G00 rapid positioning[5].*

Syntax : figure 2.8 shows how to write the G00 command .

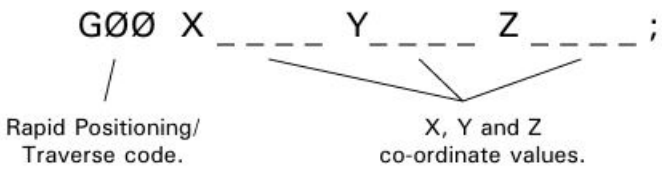


Figure 2.8 *G00 format[15].*

2.3.5.2 G01 linear interpolation

The programmed point is reached by performing a linear trajectory at a programmed feed rate as shown in figure 2.9.[5]

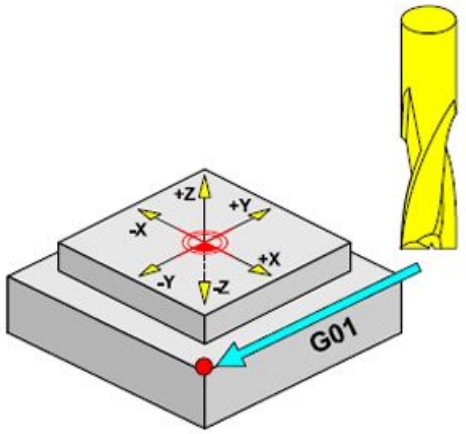


Figure 2.9 *linear interpolation[5].*

Syntax : figure 2.10 shows how to write the G00 command.

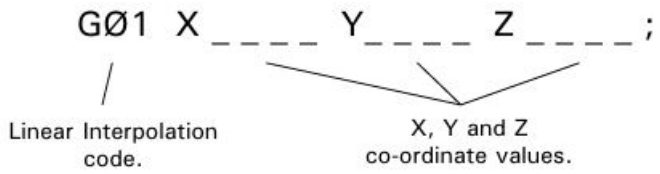


Figure 2.10 *G01 format[15].*

2.3.5.3 G02/G03 circular interpolation

The G02 code executes a cutting movement following a clockwise circular path, at a set feedrate.
The G03 code executes a cutting movement following a counterclockwise circular path, at a set feedrate.[15]

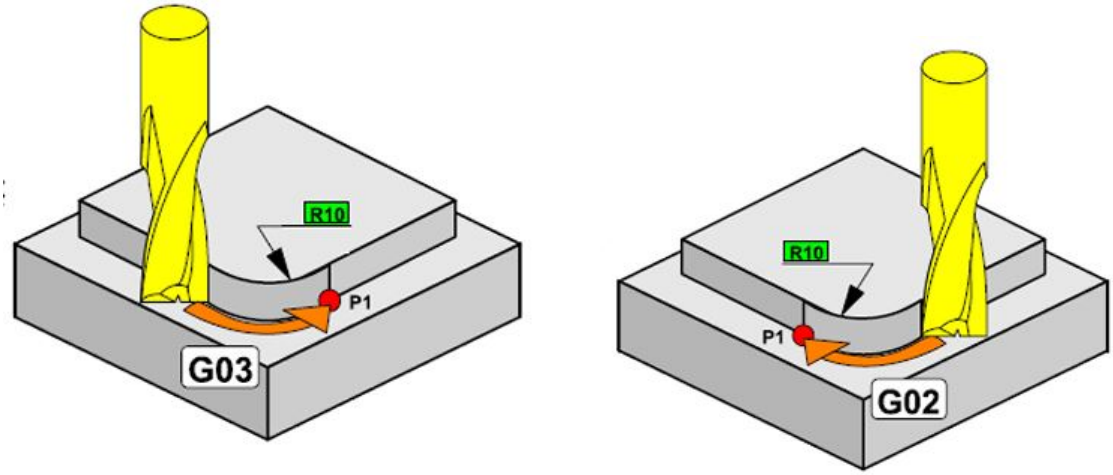


Figure 2.11 *circular interpolation [5].*

The definitions of clockwise (G02) and counterclockwise (G03) are fixed according to the system of co-ordinates in the figutre 2.12 below[15].

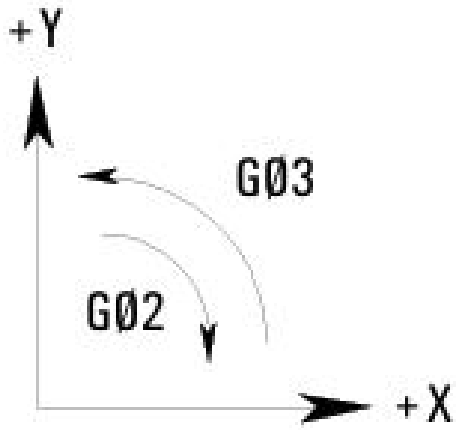


Figure 2.12 *clockwise (G02) and counterclockwise (G03)[15] .*

2.3.6 Interpolation plane

- G17 : Plane XY
- G18 : Plane XZ
- G19 : Plane YZ

Two controlled linear axes depending on the choice of interpolation plane (Milling only, shown figure 2.13 below). [5]

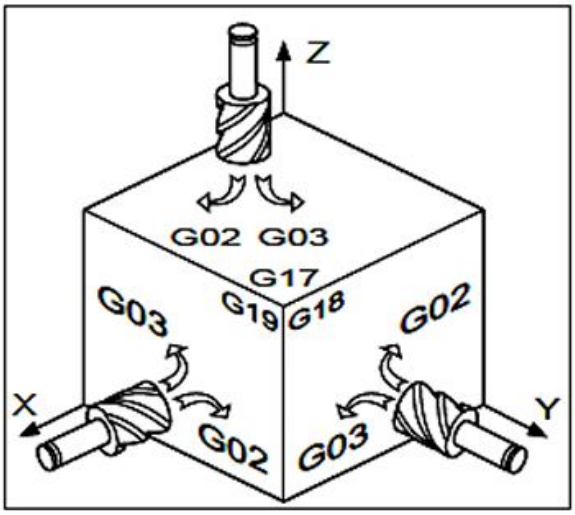


Figure 2.13 *working plane for circular motion operations[5]* .

These codes are used to define the working plane for circular motion operations (G02, G03).

2.3.7 G20/G21 coordinate positions

The machine controller can be programmed in either imperial (inch) unit input (G20) or metric (millimetre) unit input (G21). The standard format for a CNC part program is to write the G20 or G21 code in the first block of the program.[15]

Table 2.4 *coordinate positions[15]* .

G-code	Type	Units	Lowest inpute value
G20	imperial	inch	0,0001 inch
G21	metric	milimetre	0,001 mm

2.3.8 Programming mode G90 / G91

There are two types of tool movement commands: absolute and incremental (relative).
G90 : Absolute programming relative to the program origin. The value programmed on an axis is referenced to the program origin .
G91 : Programming relative to block start point. The value programmed on an axis is referenced to the last programmed position.
To ensure the machining of a part on a numerically controlled machine tool, the programmer can receive the dimensioned finished product drawing in two modes. [5]

2.3.9 Other commands

2.3.9.1 Rotation speed (S)

With this address, we can program the speed of rotation of the tool on a milling machine, drilling machine, lathe, etc. This speed is programmed in revolutions per minute or meters per minute.
Example: N5 G97 S1250 (rpm) / N5 G96 S80 (m/min) .[14]

2.3.9.2 Work advance (F)

This address is used to program the work feed speed on a milling machine, drilling machine, lathe, etc. This feed speed is programmed in millimeters per minute or millimeters per revolution of the rotating element. This speed is programmed in millimeters per minute or millimeters per revolution of the rotating element.
Example: N50 G94 F300 (mm/min) / N50 G95 F0.3 (mm/rev) .[14]

2.4 M-code

Miscellaneous functions, called M codes, are used by the CNC to command on/off signals to the machine functions. ie, M03 - spindle forward (CW), MØ5 spindle stop, etc...
The functions allocated to lower M code numbers are constant in most CNC controls, although the higher M code number functions can vary from one make of controller to the next.
An M code is defined using the M address letter and a two digit number as follows (figure 2.14). [15]

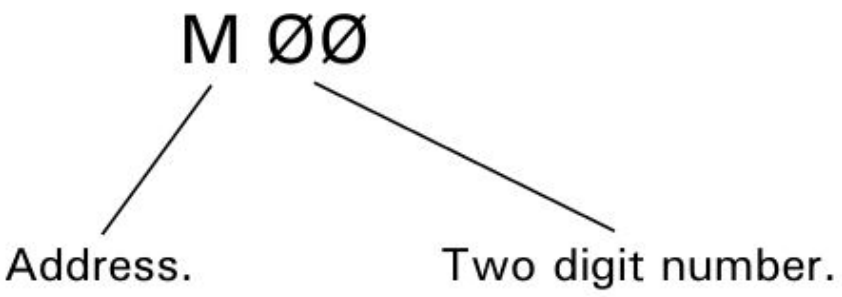


Figure 2.14 *Example of M-code definition[15].*

2.4.1 Commonly used M-code

Table 2.5 shows the various commands used in the M-code or the auxiliary function code for numerical control machines.

Table 2.5 *Functions of M-Code [31].*

Code	Function
M00	Program stop
M01	Optional program stop
M02	End of program
M03	Spindle rotation in anti-trigonometric direction
M04	Spindle rotation in trigonometric direction
M05	Spindle stop
M06	Tool call (tool change)
M08	Start watering
M09	Watering stop
M13	Anti-trigonometric spindle start with coolant
M14	Trigonometric spindle start with coolant
M30	Program end and reset to the beginning of program
M46	Feed speed modulation not enabled
M47	Feed speed modulation enable
M48	Spindle speed modulation not enabled
M49	Spindle speed modulation validation
M68	Counterpoint sleeve feed
M69	Tailstock sleeve retraction
M99	End of sub-program

2.4.1.1 End of program (M02)

The M02 auxiliary function is perhaps the most important of all. It indicates the end of the program and also cancels spindle rotation and coolant. Its primary purpose is to restore the starting position (first block to be executed). [14]

2.4.1.2 Spindle rotation (M03-M04)

These two auxiliary functions give the command to rotate the spindle clockwise or counter-clockwise as shown in the figure 2.15. They are normally found in the first or second block of a program. [14]

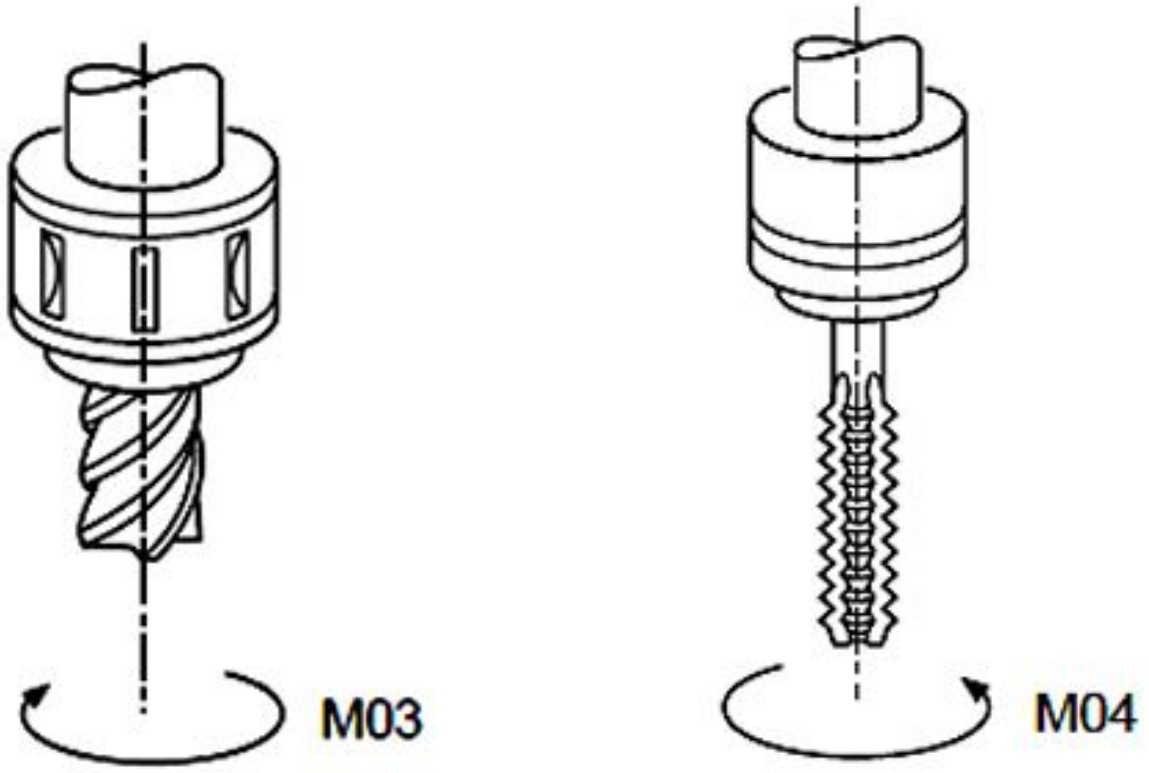


Figure 2.15 *circular interpolation [5].*

2.4.1.3 Spindle stop (M05)

The M05 code, to stop the spindle rotating, is activated at the end of the block in which it is programmed, ie , after any axis movement.[15]

2.4.1.4 Tool change (M06)

This auxiliary function prescribes manual or automatic tool change. Tool selection is not governed by this function, but by tool numbering. This function may or may not automatically stop the coolant and spindle. The choice of NC manufacturer must be specified in the NC operating instructions. [14]

2.5 Real example of G-code

In this part, we will see an applied example of the G-code, as shown in figure 2.16, which expresses an image of an electronic circuit.

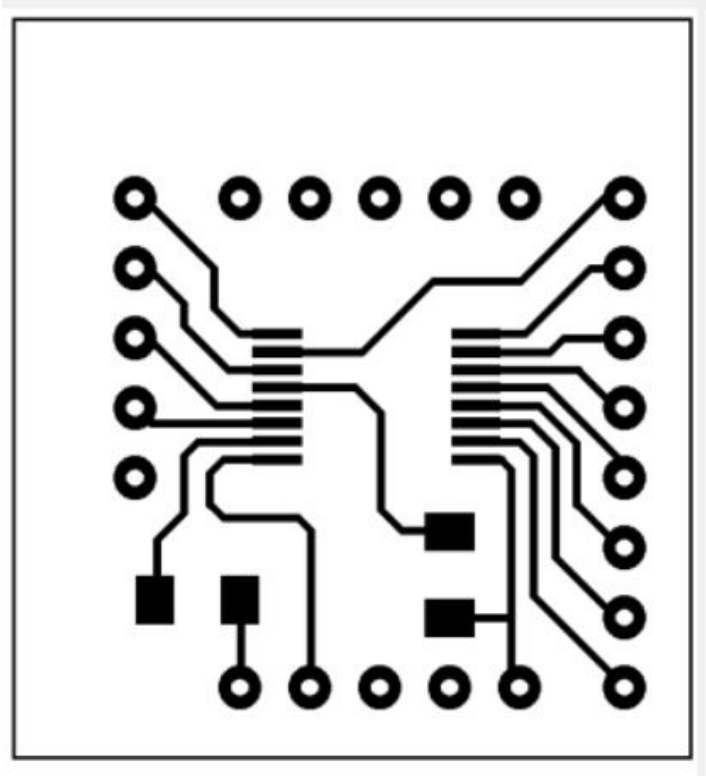


Figure 2.16 *Real image of electronic circuit.*

After loading the image, edges are extracted from it as shown in figure 2.17.

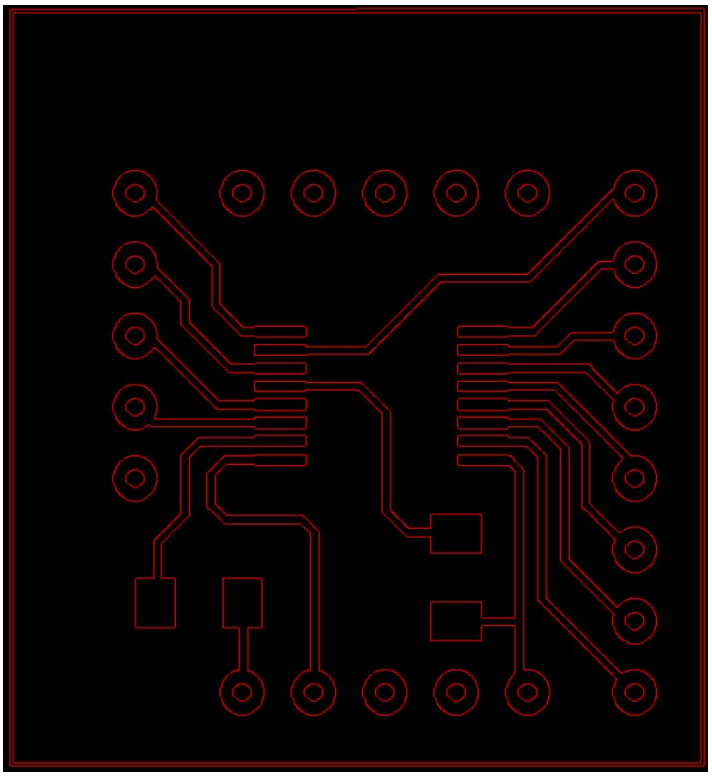


Figure 2.17 *Extract image edges.*

Finally, the edges of the image are converted into G-code (figure 2.18).

```
G00 S1; endstops
G00 E0; no extrusion
G01 S1; endstops
G01 E0; no extrusion
G21; millimeters
G90; absolute
G28 X; home
G28 Y; home
G28 Z; home
G00 F300.0 Z10.0;
G00 F2400.0 Y15.0;
G00 F2400.0 X10.0;
G00 F2400.0 X10.0 Y15.0;
G00 F300.0 Z5.0;

 (-----------------Layer 1 -----------------)
G00 F1500 X10.0 Y15.0;
G00 F300.0 Z5.0;
G00 F1500 X10.0 Y49.987947658402206;
G00 F1500 X41.80612947658402 Y49.987947658402206;
G00 F1500 X41.80612947658402 Y15.0;
G00 F1500 X10.012052341597796 Y15.0;
G00 F300.0 Z5.0;
G00 F1500 X10.638774104683195 Y15.807506887052341;
```

Figure 2.18 *The G-code of the electronic circuit.*

2.6 Conclusion

In this chapter, we provided a general and comprehensive overview of the field of numerical control programming using a computer, with a focus on the basics and programming languages (G-code & M-code).
We started with an overview of numerical control and numerically controlled machine tools (MOCN) and how to program them. The we explored both the G-code and the M-code in detail, mentioning the various commands used in programming. The G-code is the basic language for programming machine movements, and the M-code which in turn completes G-code by managing the auxiliary functions of numerical control machines.
Finally, everything we discussed in this chapter is what caused the industrial revolution in the world , which reduced human errors and increased the efficiency and accuracy of manufacturing. Therefore, understanding and mastering the concepts of G code and M-code is essential for engineers and those wishing to excel in manufacturing. Modernization and increasing competitiveness in the market.

In the next chapter, we will rely on image processing concepts and numerical programming concepts to convert images to G-code by developing a graphical interface in the python language.

Chapter 3

Image to G-code conversion

in this chapter, we will discuss how to convert image to G-code , in addition to how to create a graphical user interface using python language.

Introduction

After we learned about both image processing in the first chapter and numerical control programming in the second chapter, the question posed was how can we combine these two different fields? and this is what we will answer in this chapter.

This chapter focuses on the procedure for converting images to G-code using a graphical user interface (GUI) developed entirely in python and decorated with Cascading Style Sheets (CSS) language. Computer numerical control machines will be under our control thanks to this interface. We will also learn about some python principles and how to create a graphical user interface that can control CNC machines without using a computer. In addition, we will discover how to convert any image to G-code using this interface.

3.1 Statement of requirements

The use of a statement of requirements (SOR) is essential for projects management, as it details the requirements and specifications of the project. Below we will apply the SOR to a graphical interface creation project.

3.1.1 Requirements

It is necessary to familiarize yourself with some basic concepts before starting this project.

3.1.1.1 Image processing

Image processing is a field that specializes in analyzing and extracting information from images with the aim of improving or summarizing them by applying different algorithms. It is divided into several branches :

- image enhancement.
- image registration.
- and image analysis.

3.1.1.2 G-Code programming language

G-code is a programming language for numerical machines, based on lines of instruction such that each line includes a set of commands to make the machine produce different actions. This language is not limited to the functions of the G-code only, but rather it requires what is called the M-code, which is complement to the G-code so that it allows us to control the laser or the Spindle and other parts.

3.1.1.3 Python programming language

Python is a very powerful high-level, dynamic object-oriented programming language created by Guido van Rossum in 1991. It is implemented in C, and relies on the extensive portable C libraries. It is a cross-platform language and runs on all major hardware platforms and operating systems, including Windows, Linux/UNIX, and Macintosh. Python has an easy-to-use syntax and is quite easy to learn, making it suitable for those who are still learning to program. Python has a rich set of supporting libraries, and many third-party modules are available for it. Python is a programming language that also supports scripting, making it suitable for rapid application development. Python comes with a powerful and easy to-use graphical user interface (GUI) toolkit that makes the task of developing GUI applications in Python quite easy. It is freely available.[16]

3.1.2 Presentation of the project

This project is developing a graphical user interface using the python language.

3.1.3 Objectives of the project

- Developing a graphical interface that can be communicate with CNC machines.
- Convert digital images to G-code.
- Integrating the interface on raspberry pi with touchscreen.

3.1.4 Technical specifications

3.1.4.1 Language and Libraries

- **Programming language:** Python, CSS.
- **Libraries:** PyQt5, Pillow, openCV, potrace, serial, numpy, time, sys.

3.1.4.2 Compatibility

Operating system: Raspberry pi OS (Legacy, 64-bit) Full.

3.1.5 Project steps

- Choice of programming language.
- Drawing the interface using Qt designer.
- interface programming and development.
- Testing.

3.2 Software tools

Creating a GUI is an essential element of modern software development, making functionality available in a visual and interactive way. For GUI development, many software tools, libraries and programming languages are available, each of which offers specific advantages in terms of performance.

3.2.1 Graphical User Interface design

In the realm of Python application development, crafting visually appealing and functionally robust graphical user interfaces (GUI) stands as a pivotal task. PyQt5 emerges as a cornerstone in this domain, seamlessly integrating the powerful Qt framework with Python, thus offering developers an unparalleled toolkit for GUI design.[28]

3.2.1.1 PyQt5

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.[21]

3.2.1.2 Qt Designer

It is a Qt tool for designing and building graphical user interfaces, it uses drag and drop technology, which greatly reduces the time and effort required for manual programming.[28]

3.2.1.3 Integration of PyQt5 and Qt Designer

Integrate PyQt5 with Qt Designer to leverage the visual design capabilities of Qt Designer while harnessing the power of PyQt5 for backend functionality. This integration allows developers to design UI layouts visuallyand seamlessly convert them into PyQt5 code for implementati.[28]

3.2.2 Choice of programming language

By choosing python as a programming language to develop graphical user interfaces, we benefit from a unique combination of simplicity, flexibility and efficiency.

3.2.2.0.1 Advantages of Python

This language offers many advantages in the field of graphical interface development. Including:

- Open Source and Free.
- Richness of image processing and G-code generation libraries.
- Ease of peogramming and integration with other tools and software used in the digital manufacturing chain.
- Portability and operation of GUI developed in python on different platforms (Windows, MacOS, Linux).

3.2.3 Choice a code editor (IDE)

We chose pycharm as our code editor because pycharm is an integrated development environment (IDE) used to program in python.

3.2.4 Necessary libraries

- **Pillow**: This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.[1]

- **OpenCV**: short for Open Source Computer Vision Library, is an open-source computer vision and machine learning software library.[1]

- **Potrace**: is a tool for tracing a bitmap, which means, transforming a bitmap into a smooth, scalable image. The input is a bitmap (PBM, PGM, PPM, or BMP format), and the output is one of several vector file formats. A typical use is to create SVG or PDF files from scanned data, such as company or university logos, handwritten notes, etc. The resulting image is not "jaggy" like a bitmap, but smooth. It can then be rendered at any resolution.[1]

- **Numpy**: NumPy is the fundamental package for scientific computing with Python.[1]

- **PyQt5** : Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development.[1]

- **Serial**: pyserial is a module for Python that allows access to serial ports on various platforms. It supports Win32, OSX, Linux, BSD, Jython, IronPython and more.[1]

3.3 Convert image to G-code

Converting images to G-code requires many stages, including image processing, edge extraction and converting them into understandable codes, and thid is what we will learn about here.

3.3.1 Importance of Converting Images to G-code for CNC Machining

The ability to convert images or text directly into G-code brings numerous benefits to CNC machining. Firstly, it simplifies the design process by allowing users to leverage their existing digital assets, such as images or text, and transform them into physical objects with minimal effort. This opens up possibilities for personalization, replication, and customization in CNC-based production. Additionally, it facilitates the integration of CNC machines into digital workflows, enabling seamless communication between design software and CNC equipment. [36]
The diagram in figure 3.1 shows the stages of converting an image to G-code with an example of each stage using the python language.

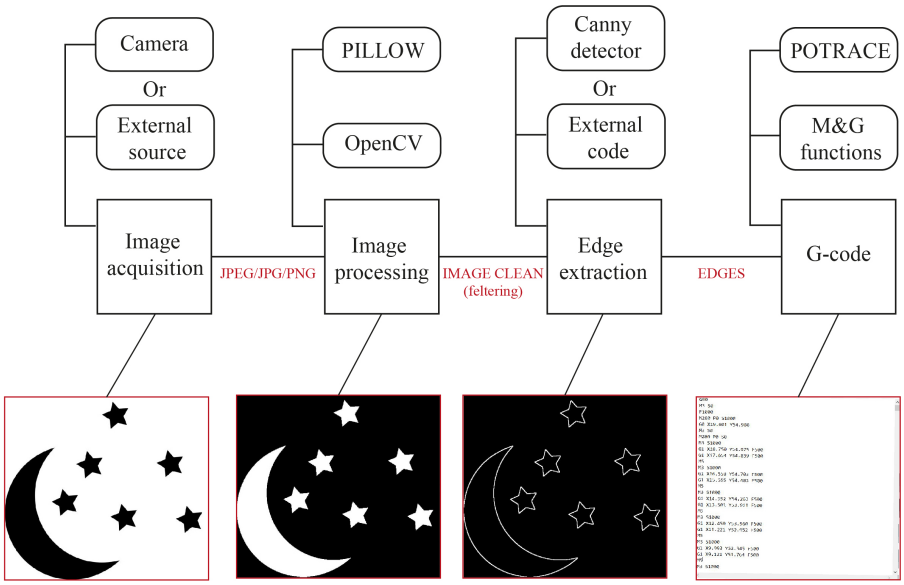


Figure 3.1 *Stages of converting image to G-code.*

3.3.2 Image processing

Image processing is the most important stage before converting images to G-code.

3.3.2.1 Convolution

Convolution is a tool for using linear filters or linear displacement filters through a mathematical process that consists of multiplying two functions by superimposing them. Equation 3.1 shows the general convolution equation.[8]

$$g(x) = f(x) * h(x) = \sum_{\forall k} h(x-k)f(k) \tag{3.1}$$

- g(x) is the convolved function (result of convolution).
- f(x) is the original function.
- h(x) is called convolution mask, convolution kernel, filter, window, kernel, ...

Figure 3.2 shows how convolution works. When any filter is applied to the image, the size of the resulting image remains the same as the original image.

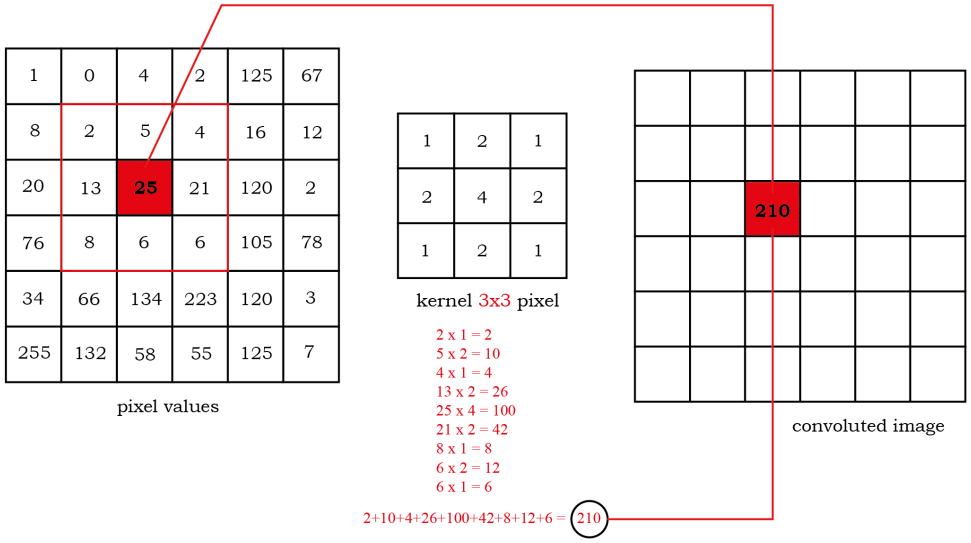


Figure 3.2 *Convolution work.*

The kernel is a small matrix (often 3 x 3, 5 x 5, 5 x 5, 5 x 5, etc.) relative to the size of the image that is applied to each pixel of the image and its neighbors to produce a new pixel, in the kernel the largest pixel value is at the center of the matrix, then the value gradually decreases (symmetric).

The convolution process is done until the entire image is covered, as in figure 3.3.

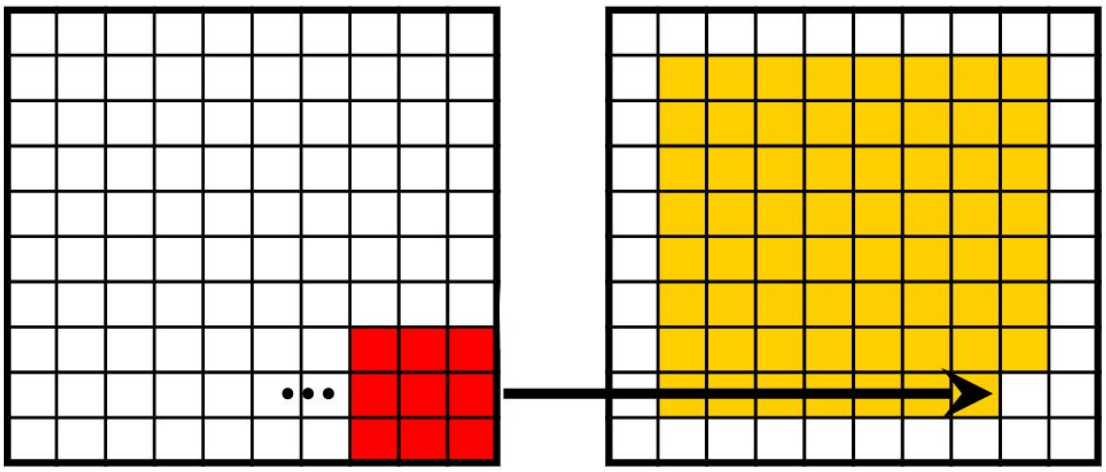


Figure 3.3 *Convolution utilizing the all pixel [8].*

But the problem occursat at the edges of the image (look at figure 3.4) because the pixels at the edges do not have neighbors for the convolution to be applied to them.

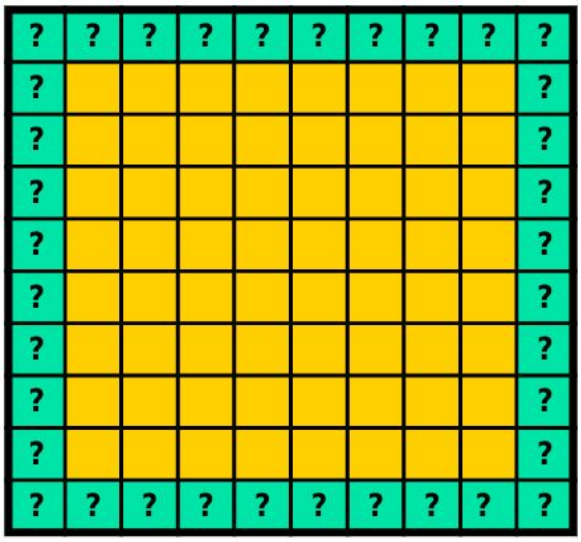


Figure 3.4 *Border Problem [8].*

To solve this problem, we add pixels to the edges with a value of zero (figure 3.5) and then apply convolution to them (this is one of the most commonly used methods).

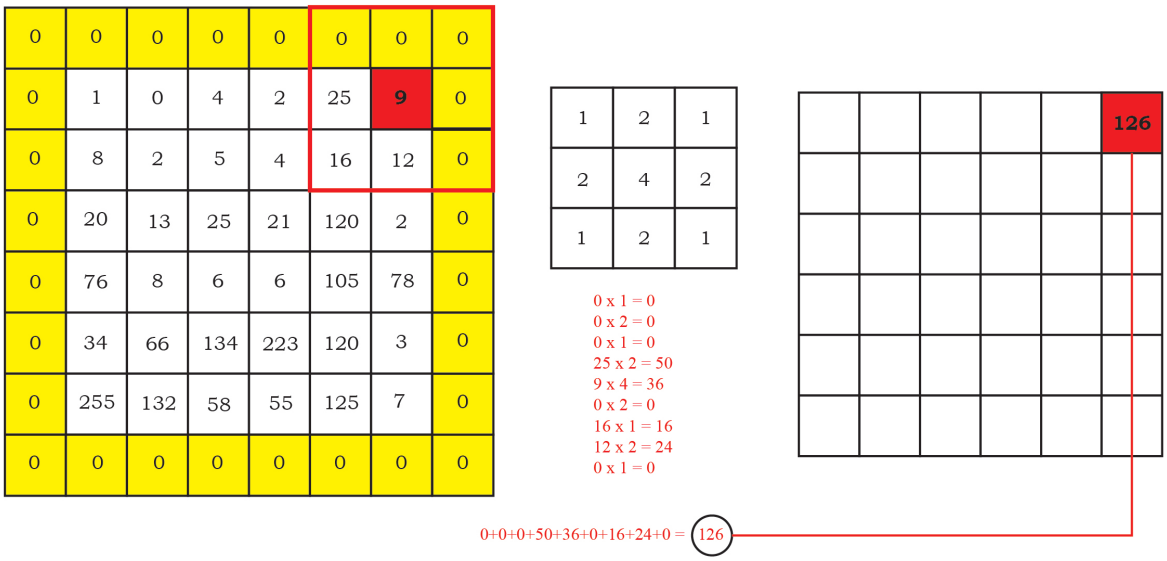


Figure 3.5 *Border constent method.*

3.3.2.2 Image filtering

Image filtering is a stage of image precessing whose goal is to remove noise from images that generally arise during the image acquisition and digitization stage.
In this project, we experimented with many image filters using the python language or specifically the opencv library.

Implementing filters in this python code can improve the quality and precision of image to G-code conversion. to obtain quality G-code, it's important to test different filters and find the optimum combination.

Diagram 3.6 it shows some of the filters we tried based on the opencv library.

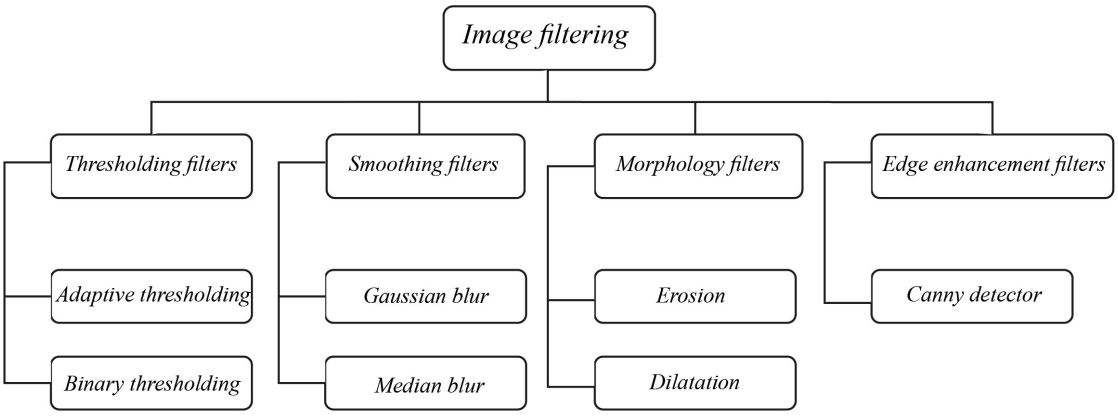


Figure 3.6 *Diagram of some image filtring.*

3.3.2.2.1 Gaussian blur

the Gaussian filter is a type of filter that uses a Gaussian function to calculate the transformation to be applied to each pixel in the image.
Equation 3.2 gives the formula for a Gaussian function in two dimensions.

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.2}$$

- x: original distance on the horizontal axis.
- y: distance from origin on vertical axis.
- σ: standard deviation of Gaussian distribution.

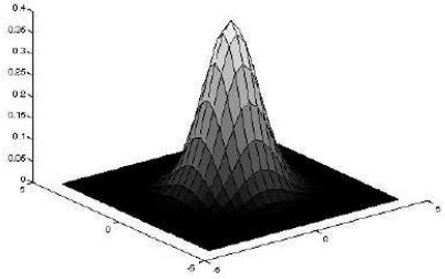


Figure 3.7 *3D Gaussian function [8].*

The Gaussian filter provides better smoothing and noise reduction than other filters (this is what i found when applying various filters).

Example of a Gaussian filter using the opencv library.

```
import matplotlib.pyplot as plt
import cv2

image = cv2.imread('original.jpg')
image2 = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
gaussian1 = cv2.GaussianBlur(image2, (3,3), 0 ,
                        borderType=cv2.BORDER_CONSTANT)

gaussian2 = cv2.GaussianBlur(image2, (15,15), 0 ,
                        borderType=cv2.BORDER_CONSTANT)

gaussian3 = cv2.GaussianBlur(image2, (25,25), 0 ,
                        borderType=cv2.BORDER_CONSTANT)

plt.figure(figsize=(8, 8))

plt.subplot(2,2,1)
plt.imshow(image2)
plt.title('original image')

plt.subplot(2,2,2)
plt.imshow(gaussian1)
plt.title('kernal = 3x3')

plt.subplot(2,2,3)
plt.imshow(gaussian2)
plt.title('kernal = 15x15')

plt.subplot(2,2,4)
plt.imshow(gaussian3)
plt.title('kernal = 25x25')

plt.tight_layout()
plt.show()
```

Figure 3.8 *GaussianBlur with varying kernal values.*

In this example, we applied a Gaussian filter using the opencv package on an image with varying kernal values, such that the image's blur percentage increased with the size of the kernal 3.8.
Everything we talked about previously about convolution and the Gaussian filteris automatically present in the opencv library and is called GaussianBlur(). We just enter some settings, which are

- image to which the filter is applied.
- kernal size.
- pixel value added to the border along with its type.

3.3.2.2.2 Grayscale images

Gray level is the value of the light intensity at a point such that each pixel of the image has a color value ranging between white and black. But we fixed each pixel to one color, either white or black, by extracting all the colors of the image that represent a number between 0 and 255, so that the number 0 represents the black color and the number 255 represents the white color. After we extracted all the colors in the image, we divided them by 255 so that they were between 0 and 1. Then we made every pixel that contained a number less than 0.5 take the number 1, meaning white, and every pixel that contained a number between 0.5 and 1 automatically took the number 0, meaning black color, Thus, the entire image becomes black and white only, as shown in image 3.9.

(a) *Original image .* (b) *Black and white image.*

Figure 3.9 *Convert image colors to black and white.*

3.3.2.2.3 Edge detection

The traditional Canny algorithm adopts nite difference of 2x2 neighbouring area to calculate gradient magnitude and gradient direction to obtain the corresponding gradient magnitude image G and gradient direction image . The rst order partial derivatives in the directions of x and y can be got from following formulas respectively [35]:

$$G_x(i,j) = \frac{(I(i+1,j) - I(i,j)) + I(i+1,j+1) - I(i,j+1))}{2} \tag{3.3}$$

$$G_y(i,j) = \frac{(I(i,j+1) - I(i,j)) + I(i+1,j+1) - I(i+1,j))}{2} \tag{3.4}$$

$$G(x,y) = \sqrt{G_x^2(i,j) + G_y^2(i,j)} \tag{3.5}$$

Example of a Canny detection using the opencv library.

```
import cv2
import matplotlib.pyplot as plt

image = cv2.imread('original.jpg')
RGB_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

imagegray = cv2.imread('original.jpg', cv2.IMREAD_GRAYSCALE)

blurred_image = cv2.GaussianBlur(imagegray, (7, 7), 1.4)

edges = cv2.Canny(blurred_image, threshold1=20, threshold2=200)

plt.figure(figsize=(8,8))
plt.subplot(221)
plt.imshow(RGB_image, cmap='gray')
plt.title('Image originale'), plt.axis('off')
```

```
plt.subplot(222)
plt.imshow(imagegray, cmap='gray')
plt.title('Gray image'), plt.axis('off')

plt.subplot(223)
plt.imshow(blurred_image, cmap='gray')
plt.title('Blur image'), plt.axis('off')

plt.subplot(224)
plt.imshow(edges, cmap='gray')
plt.title('Bord detection (Canny)'), plt.axis('off')

plt.show()
```

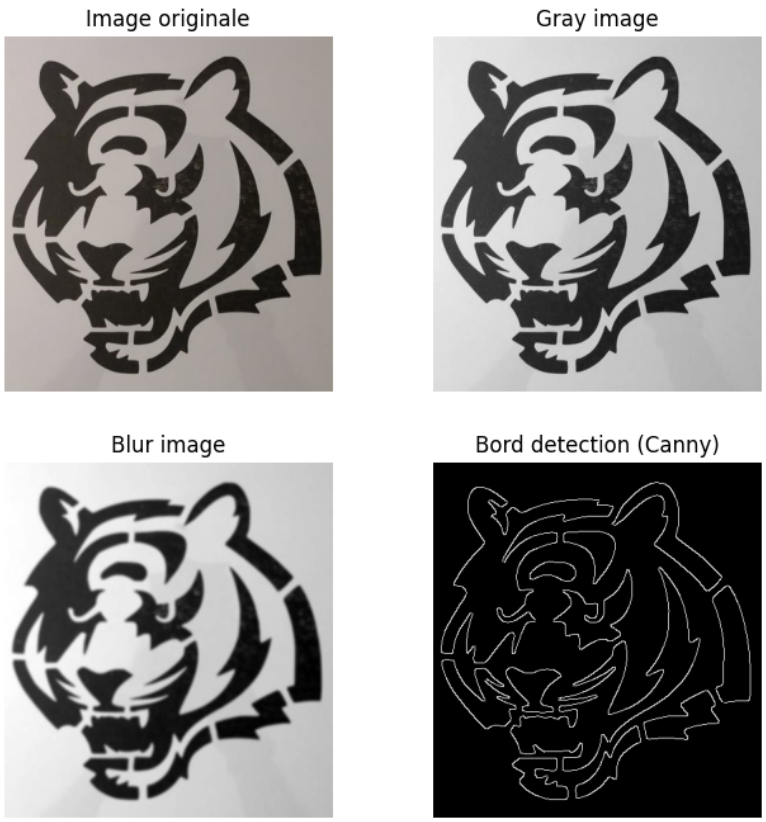


Figure 3.10 *Bord detection (Canny).*

After learning about both the Gaussian filter and the Grayscale image, in this example we applied Canny detection to extract edges (figure 3.10), as Canny is one of the best methodes used for edge extraction. Also Canny is a ready-made package inside the opencv library and has some settings.

- original image or blur image
- threshold1 : lower limit of detection
- threshold2: maximum detection limit

3.3.3 G-code generation

After completing the image processing and edge extraction stage, we will move to the last stage, which is converting the edges to G-code.

3.3.3.1 G-Code initialization

At the beginning of preparing the G-code file it must first be configured to be compatible with the CNC machine by specifying the machines's movement coordinates (G20 inch or G21 milimeter), then programming mode (G90 absolute programming or G91 relative programing), and some other settings sush as activating the homing G28, and specify the laser power (if it is a laser cutting machine) or the spindle speed (if it is a spindle engraving machine) by commanding M3 S.. to turn on and M5 to stop.

3.3.3.2 Coordinate system transformation

Before generating G-code instructions, a coordinate system transformation is often necessary to align the image or object's coordinates with the CNC machine's coordinate system. This transformation involves scaling, rotation, and translation to ensure accurate positioning of the tool relative to the object.[36]

3.3.3.3 Path generation from edges

Based on the image edges extracted from the images, trajectories for tool movement can be generated so that these trajectories follow the structure of the edges. Path generation algorithms can follow lines, whether curve or segment, and extract the coordinates of each pixel so that these coordinates are replaced within the G-code function, as shown in Figure 3.11.

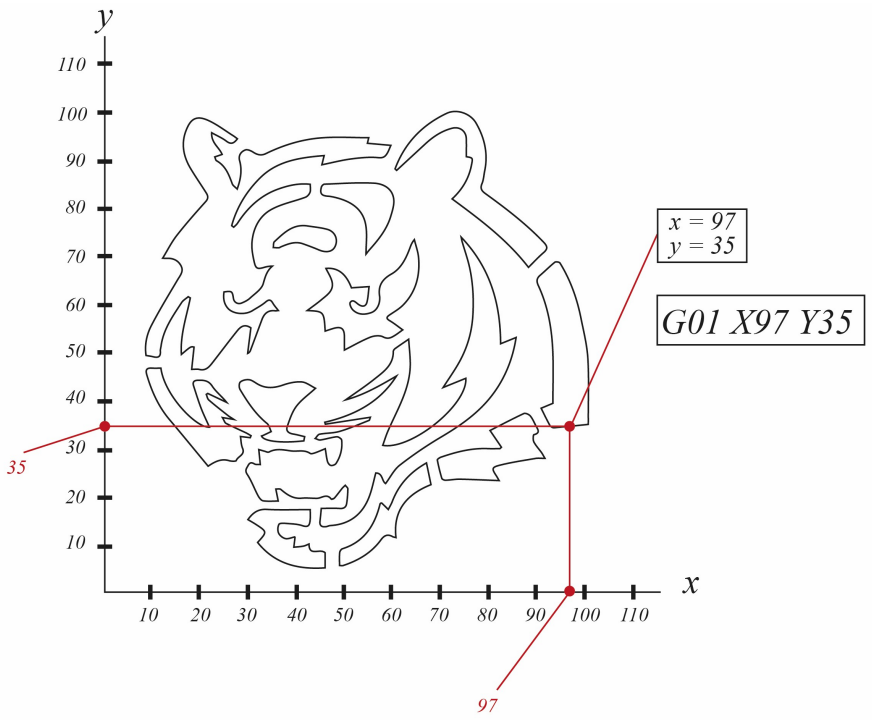


Figure 3.11 *Generate G-code from edges .*

3.3.3.4 Save G-code file

A G-code file is often saved with extension gcode or txt, depending on the software and the type of file it recognizes. For example, if the CNC is a spindle machine the file must have the extension nnc or txt, and if it is a laser cutting machine the file must have the extension gcode or nc.

3.4 Creating a graphical user interface

After completing the stage of converting images to g-code, it was necessary to create a graphical interface that would perform this conversion. In addition to making it able to controlling the CNC machine by adding buttons to control the movement of the motors, as well as sending the generated G-code file to the machine's controller.

3.4.1 Frontend GUI

On the front end, we used Qt Designer (Figure 3.12), a drag-and-drop program that enables us to create a high-quality graphical interface. This application contains many of the tools we needed, such as ready-made buttons, a text browser, line editing, and other tools, and you can modify everything inside it, such as font size and button colors, and give beautiful and modern effects to the tools used by writing some CSS code inside the interface, as if you were programming in a website or web page.

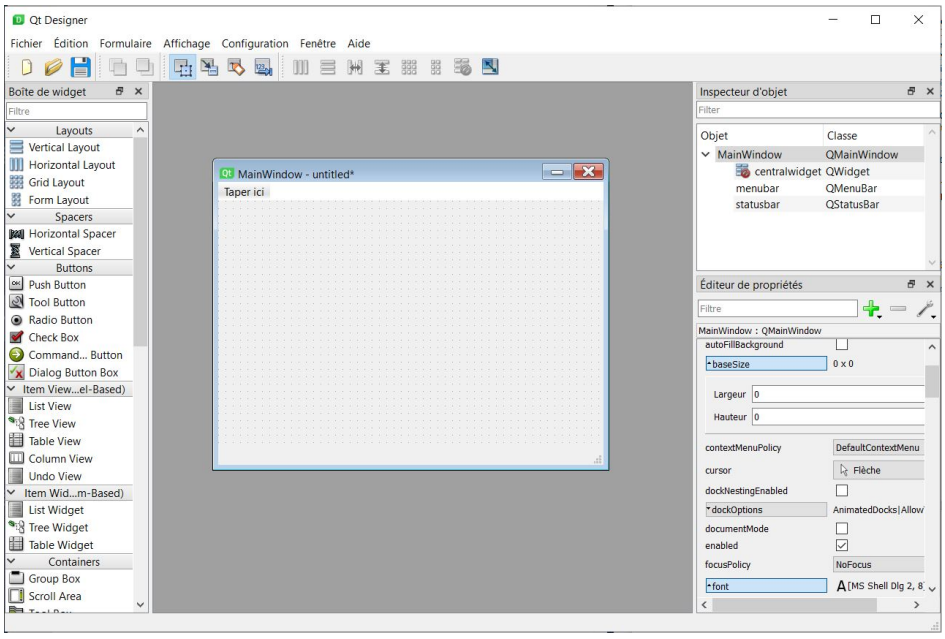


Figure 3.12 *Qt Designer interface.*

At the beginning of designing the interface, we added all the tools necessary for this project. On the other hand, we can say that this interface is divided into 3 sections.

1. The first section contains tools for connecting and controlling the machine, so that the controller name and baud rate are entered and then the connection button is pressed, so that if the connection fails, an error message is displayed to the user. This section also contains two buttons to create a G-code file and send it to the controller, and other buttons that allow the user to control the movement of the machine.

2. The second section is for opening the camera, taking images (image acquisition), displaying them, and saving them. There is also a line edit for adjusting the dimensions of the image to be converted to a G-code file, as well as a combo box for adjusting the speed of the machine, and special buttons for turning on and off the laser, as well as a slider for control. In laser power.

3. The third section is for displaying G-code commands within a text browser, in addition to line edit, which enables the user to enter the G-code command (this is called manual programming).

All of this is shown in Figure 3.13.

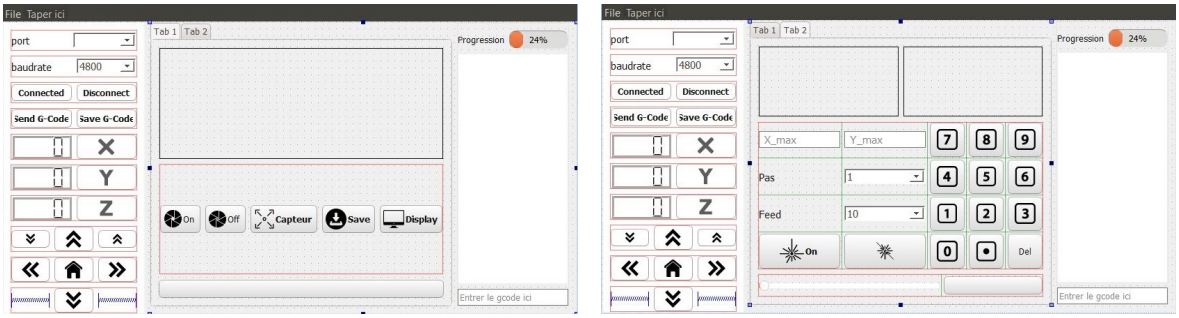


Figure 3.13 *Custom interface.*

Once the interface design was finalized, we added an aesthetic touch using CSS code to make it a modern interface with beautiful effects. Figure 3.14 shows the difference between the default interface and the interface after adding the aesthetic modifications.

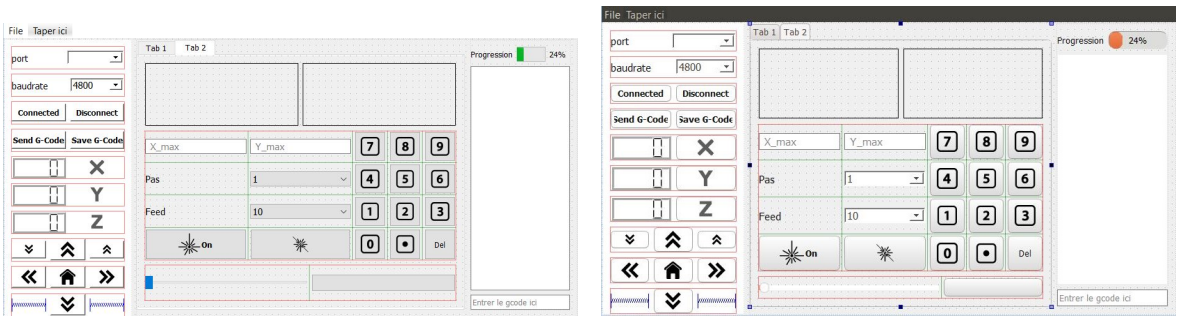


Figure 3.14 *The difference between the default interface and the modified interface..*

3.4.2 Backend GUI

Qt Designer allows us to easily link the design with the code. Once all the design is completed, we move to the programming stage using the Python language, specifically the PyQt5 library. When starting to program the interface, it is enough to save the interface file (ui extension) and then call it inside the code using a function called loadUiType() specific to pyqt5. In programming, we relied on the concepts of object-oriented programming (OOP) so that the code becomes easy and understandable. As for the programming stages, they were as follows.

1. Call the necessary libraries and create a class called Main that contains the entire program.

2. Create functions for each element of the interface, starting with opening the camera to sending the G code file to the controller.

3. Calling the tools used in the interface and linking them with their functions. For example, calling the open camera button and linking it with the open camera function by writing a line of code (pushButton_13.clicked.connect(open_camera)) so that the name of the button is pushButton_13 and the name of the function is open_camera().

4. Integrate the previously created image to G-code conversion program with this program, as well as link it to the conversion button in the interface.

Thus, the entire interface was coded tool by tool, so that the size of the program exceeded 600 lines of code , the result was as follows (figure 3.15).

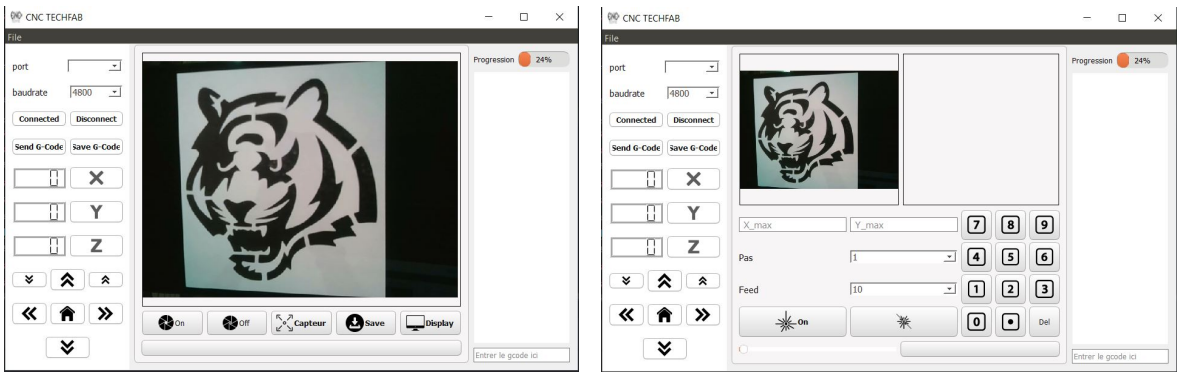


Figure 3.15 *Final interface result.*

3.4.3 Function diagram SADT

Structured analysis and design technique (SADT) is a graphical language and was used extensively for describing complex systems in communicative designs (shown figure 3.16), military planning and computer-aided manufacturing (Dickover, Mcgowan, and Ross 1977). SADT was successfully applied in problem analysis and functional specifications; however, it has been used most effectively in the requirements definition phase for software design (Ross and Schoman 1977; Ross 1985). SADT was adopted as definition for Function Modeling by US Air Force Integrated Computer Aided Manufacturing (ICAM) in 1980s [2].

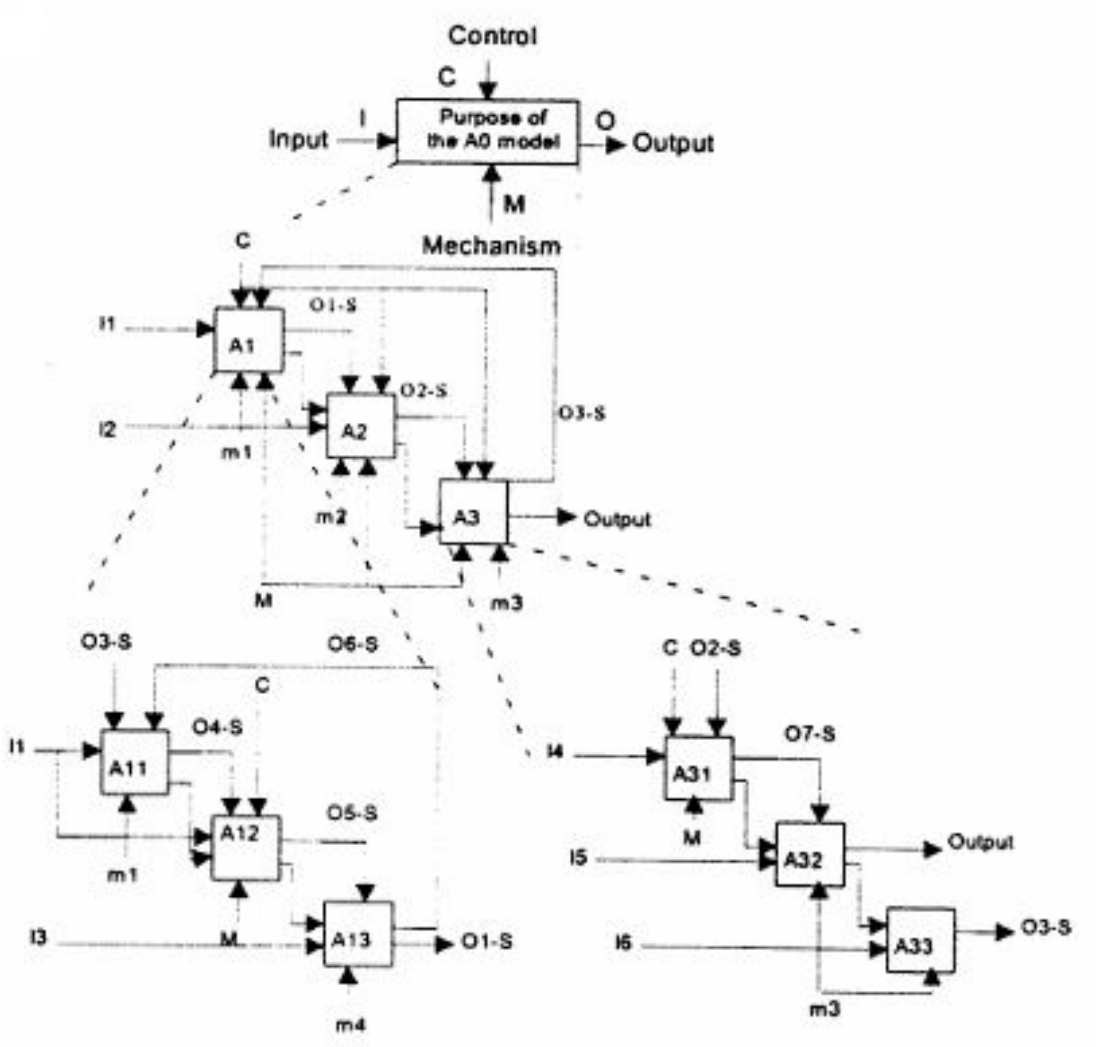


Figure 3.16 *Decomposition of SADT diagrams [34]..*

SADT is a modeling method used to describe and control system function. When we apply these conceots in the field of image processing project and converting to G-code, we obtain the SADT shown in figure 3.17

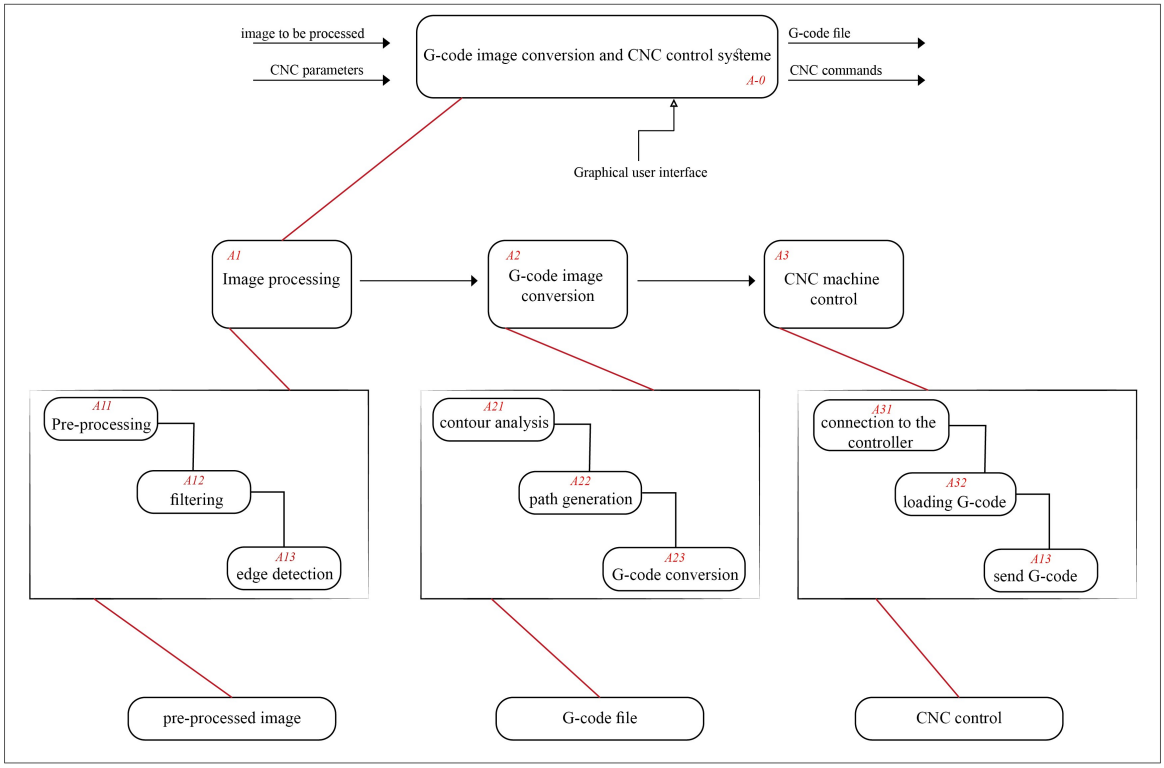


Figure 3.17 *SADT diagram for our project..*

3.4.3.1 Context diagram (A-0)

This stage contains the overall project system with its main inputs and outputs.

1. Inputs:
   - Image to be processed.
   - CNC parameters.
2. Outputs:
   - G-code file.
   - CNC commands.

3.4.3.2 Decomposition into main functions (A0)

We can say that this system is divided into three main function (A1, A2 and A3).

- Image processing.
- G-code image conversion.
- CNC machine control.

3.4.3.3 Functions in detail

Here each one is divided into main functions and sub-functions.

- A1 function : this function includes thress sub-functions (A11, A12 and A13) which are image. Preprocessing, image feltering and edge detection.

– A2 function : the function of converting image to G-code can include three sub-function (A21, A22 and A23) which are. Contour analysis, path generation and G-code conversion.

– A3 function : the control function of the CNC machine is also divided into three sub-function (A31, A32, A33) which are. Connection to the controller, loading G-code file and sending G-code one command after another.

3.4.4 Diagram for using the graphical interface

In the design phase of the graphical interface, we relied on the simplest methods to make it easy and clear for the user so that it does not require significant knowledge in the field of using applications, and this is what the diagram shows in figure 3.18.

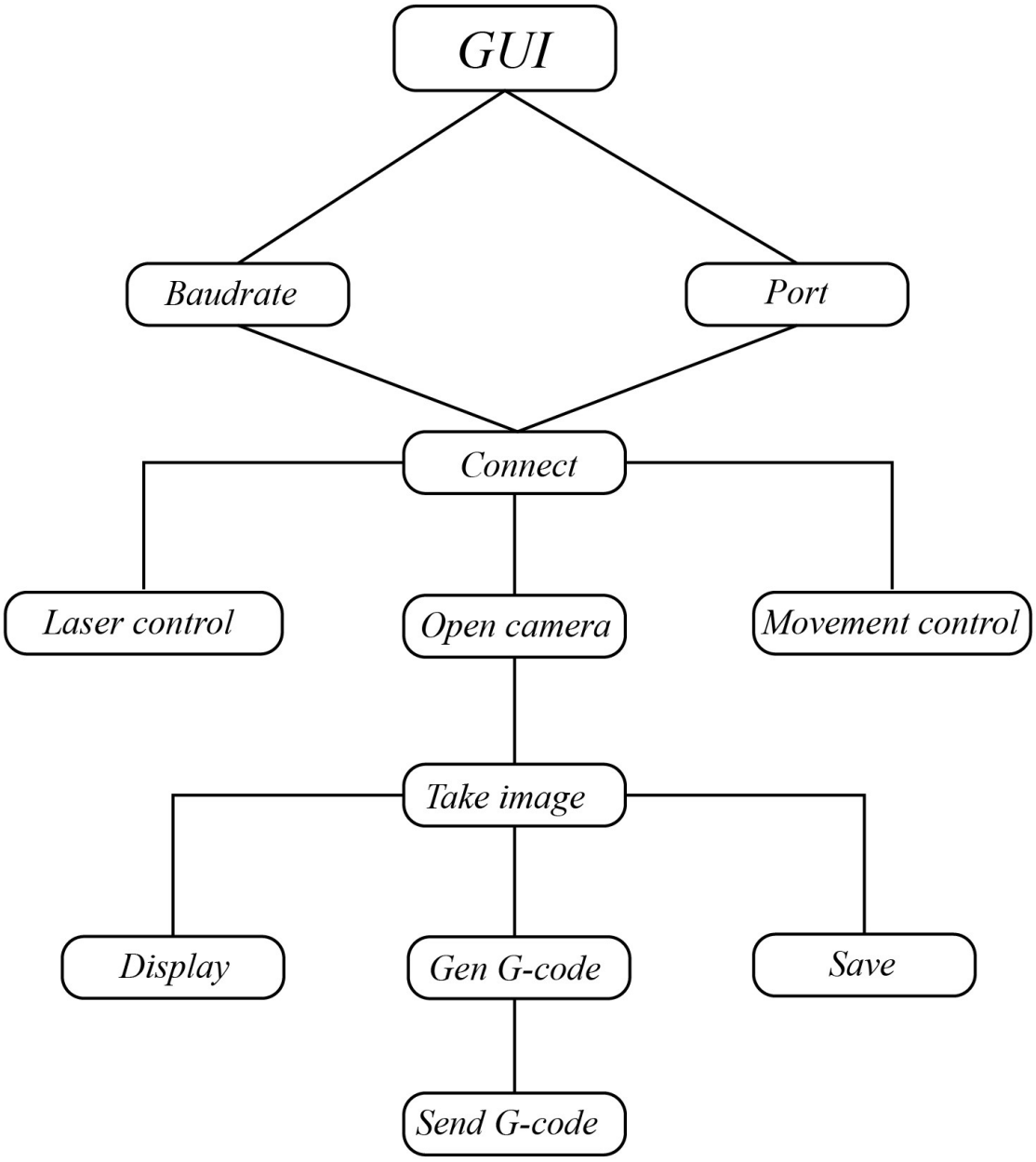


Figure 3.18 *Use diagram.*

3.4.5 Test

In the graphical interface testing phase, to ensure the accuracy and efficiency of the conversion process, we created a drawing of a real electronic circuit using proteus design software with the aim of converting it to the the G-code file and then sending it to the CNC machine to print it on the Printed Circuit Board (PCB). The result was as shown in figure 3.19.

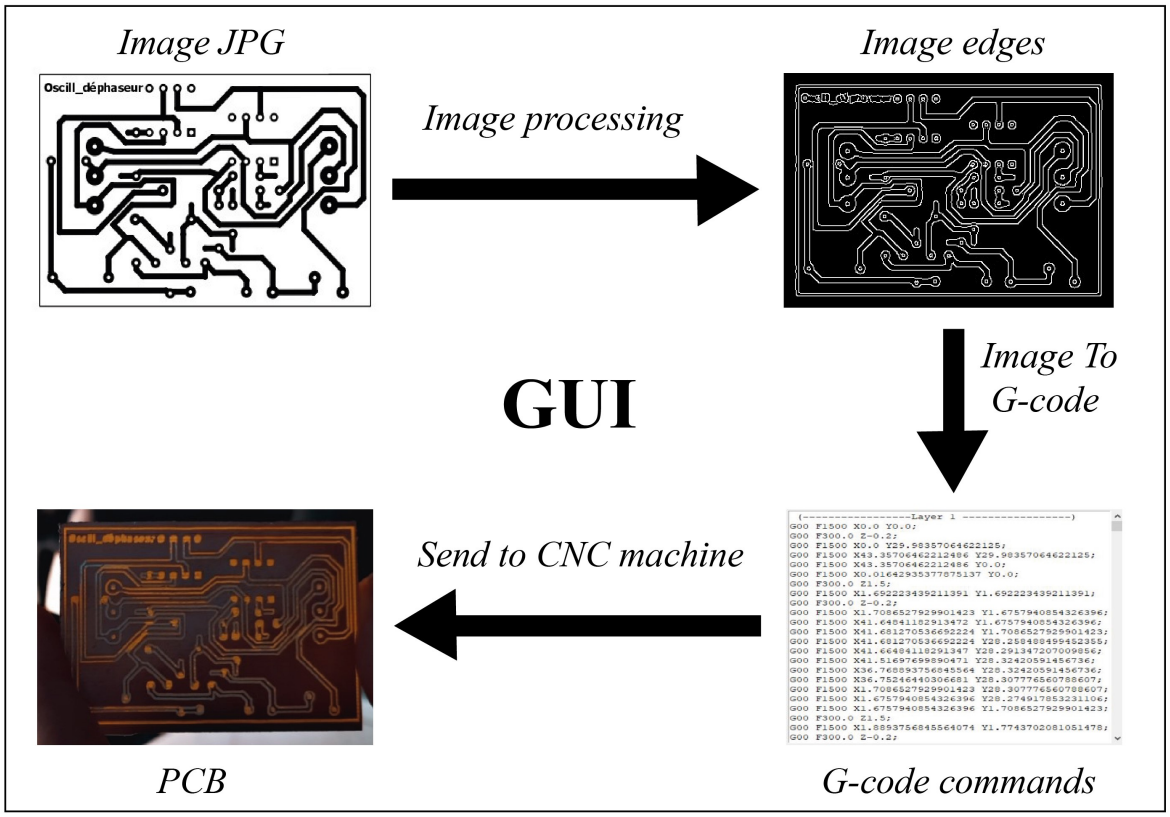


Figure 3.19 *Result of converting to G-code..*

Conclusion

This chapter presented a comprehensive and systematic approach to converting images to G-code via a graphical user interface. This project was built around several basic stages, each of which played a role in achieving the final goal.
First, we talked about image processing, focusing on filtering techniques and edge extraction using the Canny filter, and thus we prepared the image to convert it to G-code commands.
Secondly, the process of converting to G-code was detailed, starting from the edges of the images to the G-code file. This part required a deep understanding of edge analysis and the use of path generation algorithms.
Finally, this stage included developing an easy-to-use and practical graphical interface. All previous concepts, from image processing to converting edges to G-code, have been integrated into this interface. The goal of this design was to immerse ourselves in the field of digital control and increase the efficiency and accuracy of manufacturing.

GENERAL CONCLUSION

In the modern world, numerical control programming is an important element in achieving the highest levels of precision and industrial efficiency. In this thesis, we discovered the techniques necessary to develop a graphical user interface in the Python language with the aim of converting images into G code and the ability to smoothly control numerical control machines using computers to benefit more from the power of these machines. Focusing on three main aspects. Image processing, numerical control programming, and techniques for converting images to G code. Image processing is an essential step in extracting the necessary information from images. In this chapter, we learned about the concept of digital images, their types, and their characteristics. We also examined different image processing techniques, such as grayscale transformation, thresholding, and edge detection, while presenting real examples of each part. These techniques allow complex images to be converted into data usable in the field of numerical control programming. Then we moved to another chapter that contains the most important elements for understanding numerical programming for numerical control machines. We studied the most widely used programming languages in the field of numerical control, which are Gcode and Mcode, which allow controlling the movements and operations necessary to manufacture parts using CNC machines. Understanding these languages plays a major role in ensuring accuracy in manufacturing. Based on the concepts of image processing and numerical programming, we developed a graphical interface with the aim of facilitating the cutting and cutting operations. This interface included various image processing techniques with the aim of extracting edges from images. We also relied on algorithms for converting images to G code by extracting pixel values from the edges and then inserting them into G code functions. The interface also allows connecting to CNC machines and sending G code files directly for execution. This chapter explains the importance of developing an integrated system to take advantage of the capabilities of CNC machines.

In short, this thesis showed how image processing techniques and cnc machine programming can be combined. Each chapter provided a contribution, figure, and guest post that greatly simplifies the process of converting images to G-code.

Bibliography

[1] Python package index (pypi), 2023. Accessed: 2024-06-03.

[2] Fahim Ahmed, Stewart Robinson, and Antuela A Tako. Using the structred analysis and design technique (sadt) in simulation conceptual modeling. In *Proceedings of the Winter Simulation Conference 2014*, pages 1038–1049. IEEE, 2014.

[3] Ahmad El allaoui. *Segmentation Évolutionniste d'Images Numériques*. September 2013.

[4] HAMMAMI Azzeddine. Programmation des machines à outils à commande numérique. 2018.

[5] DAHMANI L. & HADJ BRAHIM B. *Chapitre 3: Programmation des Machines-Outils à Commande Numérique*. PhD thesis, ISET SILIANA / Département GM.

[6] S. Battiato, G. Di Blasi, G. Gallo, G. Messina, and S. Nicotra. Svg rendering for internet imaging. In *Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception*, CAMP '05, page 333–338, USA, 2005. IEEE Computer Society.

[7] Philippe Bolon, Jean-Marc Chassery, Jean-Pierre Cocquerez, Didier Demigny, Christine Graffigne, Annick Montanvert, Sylvie Philipp, Rachid Zéboudj, Josiane Zerubia, and Henri Maître. *Analyse d'images : Filtrage et segmentation*. Enseignement de la physique. MASSON, October 1995.

[8] Alain Boucher-IFI. Traitement d'images. *Introduction à l'image*, 2002.

[9] Abdelnour BOUKAACHE. Traitement d'images et vision. 2016.

[10] Djemaa Boukhlouf. *Résolution de problèmes par écosystèmes: Application au traitement d'images*. PhD thesis, Université Mohamed Khider-Biskra, 2005.

[11] Sonal Chawla, Meenakshi Beri, and Ritu Mudgil. Image compression techniques: a review. *International Journal of Computer Science and Mobile Computing*, 3(8):291–296, 2014.

[12] Kaidi Dalia. *Classification non supervisée de pixels d'images couleur par analyse d'histogrammes tridimensionnels*. PhD thesis, Université Mouloud Mammeri, 2017.

[13] Nameirakpam Dhanachandra and Yambem Jina Chanu. A survey on image segmentation methods using clustering techniques. *European Journal of Engineering and Technology Research*, 2(1):15–20, 2017.

[14] Mohamed Cherif DJAMAA. Machines-outils a commande numerique. 2020.

[15] enford Limited. *G and M Programming for CNC Milling Machines*. 2000.

[16] B. M. Harwani. *Introduction to Python Programming and Developing GUI Applications with PyQT*. 2011.

[17] Mohammed Ali KARFOUF and Ali BEDDIAF. *Reconnaissance des expressions faciales*. PhD thesis, UNIVERSITY OF OUARGLA.

[18] Rajandeep Kaur and Pooja Choudhary. A review of image compression techniques. *Int. J. Comput. Appl*, 142(1):8–11, 2016.

[19] Scott Krig and Scott Krig. Image pre-processing. *Computer Vision Metrics: Textbook Edition*, pages 35–74, 2016.

[20] MOHAMMED LEHZIEL, MOHAMMED Daoudi, et al. *onception et réalisation d'une machine CNC*. PhD thesis, Université Kasdi Merbah Ouargla.

[21] Riverbank Computing Limited. Pyqt5 on pypi, 2024. Accessed: May 31, 2024.

[22] Matthew Manton and Daune Weidinger. *CNC PROGRAMMING WORKBOOK-Mill*. Camlnstructor Incorporated 330 chandos crt. kitchener, ontario N2A 3C2, 2013.

[23] Antoine Manzanera. Traitement d'images et vision artificielle. *Unité d'Électronique et d'Informatique*, 2005.

[24] MathWorks. Image analysis. https://www.mathworks.com/discovery/image-analysis.html. Consulté le: 19 mai 2024.

[25] Hemza MESBAH and Oussama ZERGUINE. *DEVELOPEMENT D'UN PROGRAMME DE VISUALISATION DES TRAJECTOIRES DE LA MOCN A PARTIR D'UN FICHIER DE PIECE 3D ET SON PROGRAMME EN CODE-G*. PhD thesis, Université Ibn Khaldoun, 2017.

[26] Bouabid Mohamed. Projet fin d'étude de master. 2016.

[27] Nur Adriana Auni Mohd Andrie. Generation of g-code on aluminum blocks using vertical cnc machine. 2022.

[28] Soliev Bakhromjon Nabijonovich and Giyosiddinov Najmiddin. Optimizing pyqt5 development with qt designer. *Web of Teachers: Inderscience Research*, 2(4):254–259, 2024.

[29] Maher Najeh. *Filtrage et analyse des images radar*. Universite Laval, 1998.

[30] M ATAMNA Noura, M OUARHLENT Saloua, and M DAHRAOUI Nadia. Réalisation et commande d'une machine cnc à base des moteurs pas à pas.

[31] Tassadit Oulmi and Zina Merzouki. *Conception et étude d'une fraiseuse à commande numérique MOCN*. PhD thesis, Université Mouloud Mammeri Tizi-Ouzou, 2015.

[32] R Ramani, N Suthanthira Vanitha, and S Valarmathy. The pre-processing techniques for breast cancer detection in mammography images. *International Journal of Image, Graphics and Signal Processing*, 5(5):47, 2013.

[33] Mohamed Sandeli. Traitement d'images par des approches bio-inspirées application à la segmentation d'images. *Université Constantine*, 2:s1, 2014.

[34] Krzysztof Santarek and Ibrahim M Buseif. Modelling and design of flexible manufacturing systems using sadt and petri nets tools. *Journal of Materials Processing Technology*, 76(1-3):212–218, 1998.

[35] Renjie Song, Ziqi Zhang, and Haiyang Liu. Edge connection based canny edge detection algorithm. *Pattern Recognition and Image Analysis*, 27:740–747, 2017.

[36] Yan Zhang, Shengju Sang, and Yilin Bei. Image to g-code conversion using javascript for cnc machine control. *Academic Journal of Science and Technology*, 2023.

ملخص

في عالم يعد فيه الابتكار والتحسين المستمر أمرًا ضروريًا، تحتل ماكينات التحكم الرقمي(CNC) مكانة بارزة في تكنولوجيا التصنيع. ولتحقيق أقصى استفادة من هذه الماكينات، من الضروري تطوير أنظمة متكاملة تسهل استخدام هذه الماكينات، مما يتيح لنا إمكانيات جديدة لإنشاء منتجات عالية الجودة. كانت هذه المسألة هي دافعنا خلال هذا العمل. لذلك، تم تناول ثلاثة جوانب رئيسية.

لمحة عامة عن تقنيات معالجة الصور، ومقدمة عن البرمجة الرقمية مع التركيز على لغات البرمجة باستخدام الحاسب الآلي مثل G-code و M-code، وتطوير واجهة رسومية في لغة Python باستخدام Qt Designer. تدمج هذه الواجهة تقنيات معالجة الصور وتوليد ملفات G-code، مما يتيح تحويل الصور الرقمية إلى أوامر G-code للتحكم المباشر في ماكينات التحكم الرقمي باستخدام الحاسب الآلي في عمليات التصنيع.

الكلمات المفتاحية : آلة التحكم العددي بالكمبيوتر ، معالجة الصور، الصور الرقمية، البرمجة الرقمية، الكود G، الكود M، واجهات المستخدم الرسومية، Qt Designer, PyQt5

Abstract

In a world where innovation and continuous improvement are essential, CNC machines are in the background of manufacturing technology. To get the most out of these machines, it is necessary to develop integrated systems that facilitate the use of these machines, giving us new possibilities to create high-quality products. This issue has been our motivation during this work. Therefore, three main aspects are addressed.

An overview of image processing techniques, an introduction to numerical programming with a focus on CNC programming languages such as G-code and M-code, and the development of a graphical interface in Python using Qt Designer. This interface integrates image processing techniques and G-code file generation, enabling the conversion of digital images into G-code commands for direct control of CNC machines in manufacturing processes.

Keywords : CNC, image processing, digital images, numerical programming, G-code, M-code, graphical user interfaces, Qt Designer, PyQt5

Résumé

Dans un monde où l'innovation et l'amélioration continue sont essentielles, les machines CNC sont au cœur de la technologie de fabrication. Pour tirer le meilleur parti de ces machines, il est nécessaire de développer des systèmes intégrés qui facilitent l'utilisation de ces machines, en nous donnant de nouvelles possibilités de créer des produits de haute qualité. C'est cette question qui nous a motivés tout au long de ce travail. C'est pourquoi trois aspects principaux sont abordés.

Une vue d'ensemble des techniques de traitement d'images, une introduction à la programmation numérique en mettant l'accent sur les langages de programmation CNC tels que le G-code et le M-code, et le développement d'une interface graphique en Python à l'aide de Qt Designer. Cette interface intègre les techniques de traitement d'images et la génération de fichiers G-code, permettant la conversion d'images numériques en commandes G-code pour le contrôle direct des machines CNC dans les processus de fabrication.

Mots-clés : CNC, traitement d'images, images numériques, programmation numérique, G-code, M-code, interfaces graphiques, Qt Designer, PyQt5