The image processing algorithm for a 360-degree camera

Bc. Trong Nghia Dao

Master's thesis
2022

Tomas Bata University in Zlín
Faculty of Applied Informatics

Tomas Bata University in Zlín
Faculty of Applied Informatics
Department of Informatics and Artificial Intelligence

Academic year: 2021/2022

ASSIGNMENT OF DIPLOMA THESIS

(project, art work, art performance)

Name and surname:	**Trong Nghia Dao**
Personal number:	**A19890**
Study programme:	**N3902 Engineering Informatics**
Branch:	**Information Technologies**
Type of Study:	**Full-time**
Work topic:	**Algoritmus zpracování obrazu 360 stupňové kamery**
Work topic in English:	**The Image Processing Algorithm for a 360 Degrees Camera**

Theses guidelines

Get acquainted with the essential topics of image processing in computer vision and summarize its basic information.
Focus on 360 degrees cameras with an emphasis on their technical specifications.
Analyze the data obtained in the real environment of a specific hydroponic greenhouse in the form of video recordings from a 360 degrees camera.
Based on the previous analysis, propose a suitable algorithm for preparing images for use in subsequent machine vision algorithms, most essentially CNN.
Code several versions of the proposed algorithm and try their suitability on the real data.
Evaluate the obtained results, identify problems, and suggest best practices for image capturing, outline directions for future image processing developments in this area.

Form processing of diploma thesis: **printed/electronic**

Recommended resources:

Forsyth, David, and Jean Ponce. *Computer Vision: A Modern Approach*. 2nd ed, Pearson, 2012.
Cameron, Joshua, et al. *360 Essentials: A Beginner’s Guide to Immersive Video Storytelling*. Ryerson University Library, https://pressbooks.library.ryerson.ca/360essentials/. Accessed 7 Oct. 2021.
Greene, Ned. 'Environment Mapping and Other Applications of World Projections.” *IEEE Computer Graphics and Applications*, vol. 6, no. 11, 1986, pp. 21–29, doi:10.1109/MCG.1986.276658.
Kweon, Gyeong-Il, and Choi, Young-Ho. 'Fisheye Lens for Image Processing Applications.” *Journal of the Optical Society of Korea*, vol. 12, no. 2, June 2008, pp. 79–87, doi:10.3807/JOSK.2008.12.2.079.
Kopf, Johannes, et al. 'Locally Adapted Projections to Reduce Panorama Distortions.” *Computer Graphics Forum*, vol. 28, no. 4, 2009, pp. 1083–89, doi:10.1111/j.1467-8659.2009.01485.x.

Supervisors of diploma thesis: **Ing. Bc. Pavel Vařacha, Ph.D.**
Department of Informatics and Artificial Intelligence

Date of assignment of diploma thesis: **December 3, 2021**
Submission deadline of diploma thesis: **May 23, 2022**

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

doc. Mgr. Milan Adámek, Ph.D. v.r.
Dean

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
Head of Department

In Zlín January 24, 2022

I hereby declare that:

- I understand that by submitting my Master's thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defence of the thesis.
- I understand that my Master's Thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Master's Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlín.
- I am aware of the fact that my Master's Thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, Tomas Bata University in Zlín has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work – Master's Thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlín with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Master's Thesis include the use of software provided by Tomas Bata University in Zlín or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Master's Thesis cannot be used for commercial purposes.
- I understand that, if the output of my Master's Thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defence of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- The submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlín; dated: .....

Student's Signature

ABSTRACT

The goal of the thesis is to make an image of an entire row of fruits (in this case, tomatoes) in a greenhouse farm, which will be captured using a 360-degree camera. The picture produced will be utilized for various reasons, such as counting and monitoring. To begin, this thesis will review the basics of computer vision and introduce essential issues. The features of the 360-degree video, as well as their technological specs, will be discussed next. The OpenCV library may now be used to evaluate the data collected in the greenhouse in the form of 360-degree video. It is possible to begin video processing and picture stitching to get the desired outcome with all of that knowledge. However, the fisheye lens causes significant distortion, necessitating extra procedures to undistort the image using methods such as cube mapping. There are also flaws in determining the video's speed, which will result in an undesired outcome. This issue may be solved by using dynamic stitching, which calculates the movement speed in real-time. All of the above techniques have resulted in a few different algorithm implementations. An assessment utilizing a generated video with all the controlled parameters is used to quantify the mistakes caused by several variations of the algorithm in order to choose the optimum technique. The pixel differences technique delivers the best result with a decent speed after a lengthy testing period. Furthermore, future enhancements for best practices in picture capture and processing for this project will be offered.

Keywords: Panorama, Image processing, 360-degree, Stitching, Undistort, Greenhouse

ACKNOWLEDGEMENT

During my time at Tomas Bata University in Zlin, the passionate support and instruction of the school's professors, particularly the professor of the faculty of Applied Informatics, provided me with the knowledge and essential experience that will serve me well in the future.

First and foremost, I want to express my heartfelt gratitude to Ing. Pavel Vařacha, Ph.D. He led and showed me how to complete my Master's thesis. Thank you for directly educating me, providing necessary knowledge, assisting me in choosing the appropriate path, and energetically answering my questions throughout the topic.

Not only that, with deep respect and gratitude, I sincerely thank all the professors for creating the conditions for me to join and be a part of this project. Besides, I also sincerely thank Ing. Peter Janků, Ph.D., who, with his experience, has guided me in detail to solve complex problems during my progress.

I hereby declare that the print version of my Bachelor's/Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

TABLE OF CONTENTS

INTRODUCTION

The thesis aims to create a picture of a whole row of fruits (i.e., tomatoes) in a greenhouse farm which is recorded initially using a 360-degree camera. The resulting image will be used for other purposes like counting or monitoring. Six key points can be listed here:

- Get acquainted with the essential topics of image processing in computer vision and summarize its basic information.
- Focus on 360° cameras, emphasizing their technical specifications.
- Analyze the data obtained in the natural environment of a specific hydroponic greenhouse in the form of video recordings from a 360° camera.
- Based on the previous analysis, propose a suitable algorithm for preparing images for use in subsequent machine vision algorithms, most essentially CNN.
- Code several versions of the proposed algorithm and try their suitability on the actual data.
- Evaluate the obtained results, identify problems, and suggest best practices for image capturing, outline directions for future image processing developments in this area.

I. THEORETICAL PART

1 Computer vision

1.1 Overview

When mentioning computer vision, we can think of it as a field within Artificial Intelligence and Computer Science that aims to recreate complex parts of the human visual system in order to give computers the ability to identify objects that appear in images and videos like a human being. Needless to say, it is not an easy task to create such machines that can behave like a person, not only because it is incredibly complicated on so many levels, but even the brightest minds among us do not comprehend how the information is processed within our brain. Furthermore, because brain research touched on everything from understanding the principles of cell function to how societies are constructed, we needed experts who could bridge this gap between these fields (1).

Despite all the difficulties mentioned above, scientists and researchers have finally figured out how to simulate and mimic our eyes and brains' activities with countless experiments. These topics can be traced back to the 20th century. It has been getting a lot more popular in the modern world than it used to be throughout the years. We live in an era that generates an immeasurable amount of digital data, which has been the main factor attracting more attention to this field. We are talking about more than hundreds of thousands of images that are shared online every day from more than a billion users (2) throughout activities like taking selfies or sharing beautiful moments on social media.

Nevertheless, what does it mean to have a considerable amount of digital images, precisely what can we do with it, and how could it guarantee the revolution of image processing? The answer lies in the way we handle them. These images can be used for the training process, making an algorithm better and more reliable. This is because, in the machine learning field, the quantity and quality of the data play a crucial role in improving the results of algorithms. It is like how we learn as a child; the more information we collect, the wiser we become. Unfortunately, however, collecting data is usually the part that requires lots of time and causes the progress to slow down (3).

Along with the massive amounts of visual data, the field of computer vision evolves with better hardware. The closest example is the evolution of GPU or CPU throughout the years. Since these processors have become faster with an increased number of cores and speed, they help accelerate the time spent on experimenting and training with

a more extensive set of data through parallel computing (4). Back in the 90s, when most hardware could not keep up with the task at hand, there was not a lot of data processing that scientists and researchers could execute in the same amount of time compared to modern-day achievement.

Last but not least, optimization algorithms are also a crucial part of this whole development to help researchers choose the correct solution for that particular problem (5). Without them, we can potentially face the loss of a considerable amount of time and resources. Furthermore, an optimized algorithm can help boost the processing time by a massive margin. Finally, these factors also increase the accuracy rates for object recognition as a whole. Therefore, the combination of these elements is the reason for such fast development in computer vision.

Since the early of the century, technology has come a long way. Today's systems have reached 99% accuracy from the original 50% while performing object detection tasks in just over a decade (6). This is, in many cases, more accurate than humans at responding quickly to visual input. Of course, this is far from the future in which computers will replace humans anytime soon. However, it is undeniable that machines have already had some advantages for long and repetitive tasks. Furthermore, because of that, it is the right decision to focus on developing new technologies related to this area.

1.2 Computer Vision Process

Human vision is not easy to explain for most people because it is such a complicated process that contains many pieces of knowledge from many different fields. That is not even mentioning that we have not fully understood how it works yet (7) . Nevertheless, it does not mean that scientists cannot try and simplify it. Generally, it can be divided into three sequential stages, each of which has a specific goal.

- Input acquisition - Eye simulation
- Input processing - Visual cortex simulation
- Input analysis - Brain simulation

1.2.1 Image acquisition

The very first step is the acquisition of the input. Just as we, as humans, see the world, what we are trying to accomplish here is to simulate the functionality of the eyes. However, since we are working with a computer, the input should be in the form of photos and videos. So in practice, at this phrase, the task at hand is to collect as much as possible data from the real world by filming lots of videos or taking photos of a specific topic. With the help of modern cameras that come in many sizes and brands, this can be achievable. The most common types of cameras are either digital or analog, mechanical or mirrorless, and they also come with countless different lenses.

Image acquisition is the field that has had the most success so far compared to the others. With the evolution of new technology, modern sensors and processors have been created to be used in a camera that resembles the human eye's ability to see. To some extent, it is even better. Larger, optically perfect lenses and high-resolution sensors make today's cameras incredibly precise and sensitive. The camera can capture thousands of images per second and accurately detect objects from a far distance. Nevertheless, despite all of that, the large image resolution only plays a small role in the field of Computer vision.

Despite their high fidelity, these devices are not much better than the pinhole cameras of the 19th century in the case of processing and analyzing the images. They merely record the distribution of photons in a predetermined direction, and that is it. Even the best camera sensors cannot detect a ball at all, let alone try to predict which direction it is flying toward. In other words, the hardware is limited by the advancement of the software - by far the most challenging problem to solve.

1.2.2 Image Processing

Because of the reasons above, the procedure of handling an image has such a big impact on the development of computer vision. Imagine an imperfect picture that can be modified to create the desired result. Noise, for example, is a random change in picture intensity that appears in the image as grains. It might appear in the image as a result of fundamental physics, such as the photon nature of light or the thermal energy of heat within the image sensors. It may appear during the image capture or transmission process. When pixels in a picture exhibit various intensity levels instead of real pixel values, this is referred to as noise (8).

The presence of such noise in an image might have a negative impact on its future use. However, if the data is to be further processed, noise can make it become a complicated task. For instance, some basic tasks like edge recognition and segmentation can become much more challenging (9). That is why the images should be processed before being used for many different purposes. However, whatever is being done to the image after capturing can be considered to belong in the processing step and not just about noise reduction. Therefore, there is no one specific definition or rule to follow for this section. In this thesis, the images will be processed in many different ways, like unwrapping, stitching, and resizing. Nevertheless, before that, it is better to go through some of the most basic image processing examples.

Firstly, edge detection is the technique of locating spots in a digital image where the image brightness varies dramatically using a number of methods. Basically, this method's purpose is to find the edge that separates different objects that are in the same frame. The locations in a picture where the brightness changes dramatically are usually organized into a collection of curved line segments called edges (10). Moreover, edge detection is a critical technique, especially in the fields of feature recognition and extraction.

Secondly, segmentation is the process of dividing a digital image into several segments or sections in computer vision. Segmentation is generally used to simplify or transform an image's representation into something that might create more meaning or simplify analysis. In short, this process will make the shape of an object stands out compared to the others. Therefore, it is necessary to locate the boundaries of these objects like lines or curves in pictures via image segmentation. We can then process the image after it has been broken down into segments.

1.2.3 Image analysis

The final problem is how to simulate the function of the brain when it comes to recognizing objects that appear in photos. Many scientists have been trying to build a neural network based on the human brain. The brain was built from scratch, and the data is gradually filled into it. Billions of cells work together to pick up patterns. Take recognizing an object as an example. One group of neurons will notify another when there is a difference between objects like the shapes of a circle, a triangle, or a square. These artificial neurons must be trained with lots of data in order to differentiate the shapes. Without the training process, these neurons cannot do anything rather than make a lucky guess.

A bottom-up approach that mimics how the brain works seem more promising. The computer can apply a transformation sequence to the image and find out the contours, objects it refers to, angles of view, movement, and many other aspects. A lot of calculations and statistics are required in this process. However, there is only a finite number of images that it can be taught. The same thing also applies to our human brain. We only have a small amount of time to learn about new things around us, but we are much better and faster than machines. Creating and operating artificial neural networks was difficult due to the sheer amount of computation until recent years. Advances in parallel computing have alleviated this difficulty. The past few years have seen an explosion of research and the use of this system in mimicking the human brain. The pattern recognition process is accelerating, and we are still making progress.

As the researchers pointed out, the nervous system comprises all of your body's nerve cells. We communicate with the outside world through the nervous system, and numerous internal systems are controlled simultaneously. Information from our senses is received by the nervous system, which then analyzes it and initiates reactions, such as moving muscles or making people experience pain. Nevertheless, a computer cannot tell what an apple is, what it tastes like, whether it is edible or not, how big or small it is, or what it is used for. That is because a program lacks a lot of features compared to a human brain like short-term and long-term memory, data from the other senses like your nose and your mouth, attention, perception, and lessons learned from interacting with the world. They are written and stored on a network of connected neurons that is more complicated than anything we have ever seen, in a way that we still cannot fully comprehend.

That is where computer science and artificial intelligence meet. Unfortunately, there is still no agreement on how the mind works (11) among computer scientists, engineers, psychologists, neuroscientists, and philosophers, let alone trying to replicate an exact model. However, the lack of understanding has never stopped humans from exploring and experimenting with new ideas and concepts. Even though it is currently in its infancy stage, computer vision is still beneficial. Many applications can be presented on a mobile phone, like recognizing a user's face to unlock the device. It also helps self-driving cars recognize signs and pedestrians in order to avoid any accidents. We can also find it resides in the robots in the factory, which usually identify products that increase the profit for the company by speeding up the entire process.

Going back to two of the most basic methods for image processing in the section above, it is also good to know how they are used in an actual application when we try to analyze an image. The first example is object detection. The algorithm for object detection

Fig. 1.1 Object detection

creates a box for each class or type of item that appears in the image. This can be considered the most well-known computer vision application (12) with lots of examples and tutorials on the Internet. Nevertheless, because this approach only draws boxes around the object, it has the drawback of not containing any helpful information about the shape of that object. Therefore, this method is quite unclear if we want to collect additional data. On the other hand, a different method called instance segmentation can build a mask that covers the shape of each item in the image. Compared to object detection, this approach gives us a lot more information about the object.

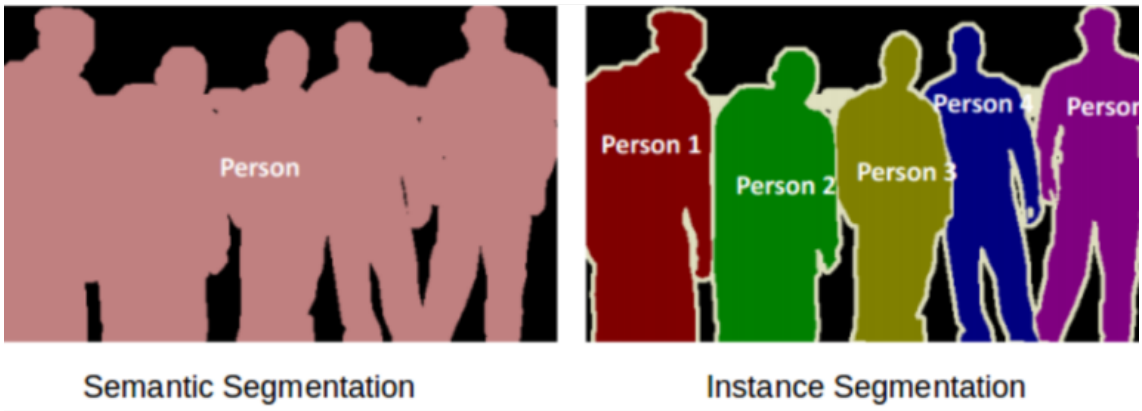

Fig. 1.2 Segmentation

Image segmentation techniques can be divided into two different types. Both strategies, however, are utilized in distinct contexts. We can see that each pixel in the first image

corresponds to a specific section thanks to semantic segmentation (either a person or the background). Pixels in the same class are assigned the same color using semantic segmentation. In this example, the humans are represented by pink pixels, whereas the background is represented by black pixels.

On the other hand, instance segmentation varies from semantic segmentation in the sense that it assigns distinct colors to each item of the same class. For example, the first person should be red, and the second person should be green, the backdrop should be black, etc. In summary, if an image contains several objects, semantic segmentation will prioritize identifying all of them as a single instance. Instance segmentation, on the other hand, will recognize each item separately.

Last but not least, classification is a computer vision method that can categorize an image based on its visual information. For example, by feeding a large number of photos of an object, in this case, a car, to an artificial neural network, the algorithm should be able to determine whether or not an image contains a car or not. Convolutional Neural Networks are the most extensively used image categorization architecture (13). The most common use case for this type of network is when the network is given photos, it will classify the data based on that data.

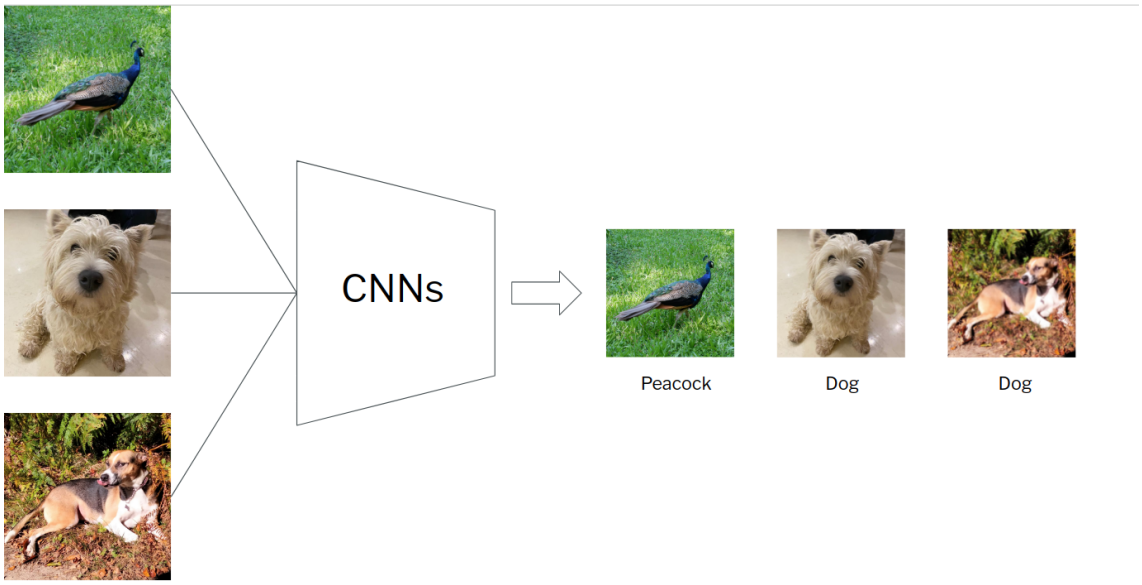

Fig. 1.3 Classification task

2 360-degree videos and photos

The definition of 360-degree video can be summarized as a video that records all the surroundings of the camera at the same time (14). Even though panorama image has existed for such a long time, 360-degree video is a relatively new term. Nevertheless, it is not unfamiliar to a vast majority of people and is becoming the new direction of future image processing.

An interactive 360-degree photo is a type of image that allows viewers to interact with the image and explore the entire scene around the captured point. When viewing photos, viewers only need to manipulate the 360-degree icon on the image, and the image will rotate so they can see the panorama of surrounding areas. Many prominent social platforms like Google have implemented this feature for viewers to see the entire surrounding of a specific location on Google Map. This is apparently more useful than a still image because of the rotation feature that can give the user the awareness they need. Many smartphones and cameras nowadays also implement this feature in order for the users to capture panorama pictures.

Fig. 2.1 An interactive layout of a dormitory room

It has become a hot trend that encourages more and more people to participate and create videos in this format in recent years. The most famous application has to be virtual reality (15). Many big and famous tech companies have already joined the race to develop the product and invested more into the research and development process. The most recent example is the Metaverse, which is currently in the development

process by Meta, the owner of the biggest social platforms like Instagram, Facebook, or Whatsapp. However, it is easier encounter a much simplier example in the form of virtual tour. Viewers can drag the mouse to move slowly or rotate the direction based on gestures to see different angles in that space and go to different directions. These type of virtual tour are great because it eliminate the physical distance between the user and the virtual space (16).

360-degree video stands out more than conventional videos because it creates a natural feeling about the object and the surrounding scene. Not only that, with rich viewing methods, users can also freely choose the direction to view. This makes the whole experience feels more immersive and that people are actually visiting new places without even going there yourself. The applications for this new technology are numerous, and our imagination only limits the possibilities.

2.1 Experience a 360-degree video

There are many ways that users can experience a 360-degree video. Nevertheless, augmented reality is considered to be the most cutting-edge technology possible to experience it. To do that, people can use 3D scanning virtual reality glasses like Oculus or just a simple product like Google Cardboard that is accompanied by your smartphone. Through this prism, people can watch the videos the way they are intended. This lets us see the surrounding whenever we move our head around. And because of that, it makes the whole experience more engaging to the user. That is the reason why well-established companies like Facebook, Samsung, Apple, and many more companies want to become the ones that will dominate this field in the near future.

As briefly mentioned in the above section, users can also watch the panoramic video via the YouTube app on Android or watch it directly on the website. Instead of wearing a heavy device on their head, people can explore videos in all directions with a few simple gestures. For example, when watching a 360-degree video on a computer, the viewing angles are changed by dragging and dropping the mouse over the video. On mobile devices, changing the viewing angle when watching a 360-degree video instead of using the mouse, users can manipulate their fingers more straightforwardly and conveniently. Either way, it is now effortless to enjoy this technology just with a small device that fits inside the palm of a human hand.

However, despite the exciting characteristics of this new type of video, there will be at least some noticeable distortion because of how those videos and images are made.

It can only be fixed by the software and therefore depends on how the developers of those videos and applications handle this inconvenience.

2.2 Fisheye lens

A crucial part of making a 360-degree video is a wide-angle lens, and for some specific reasons, the fisheye lens tends to be the most popular one. They are also known as super wide-angle lenses. Accordingly, it is inherently produced to pursue a panoramic view or a unique spherical image. In particular, this super wide-angle lens can produce a full-frame image with a field of view that can reach up to 180 degrees (17) and thus, cannot be replicated on any conventional standard lenses. Furthermore, with two fisheye lenses combined, it is now possible to take a 360-degree image.

Fig. 2.2 A normal 35mm lens

Fig. 2.3 Fisheye lenses

2.2.1 Focal length and focus distance

The distance from the lens to the camera sensor is called the focal length (18). However, the sensor can also be considered the focus point in other cases. Focal length is measured in millimeters. An essential characteristic of a lens is its focal length for the camera. The lens's focal length will depend on the type of camera and the location where the camera sensor is installed.

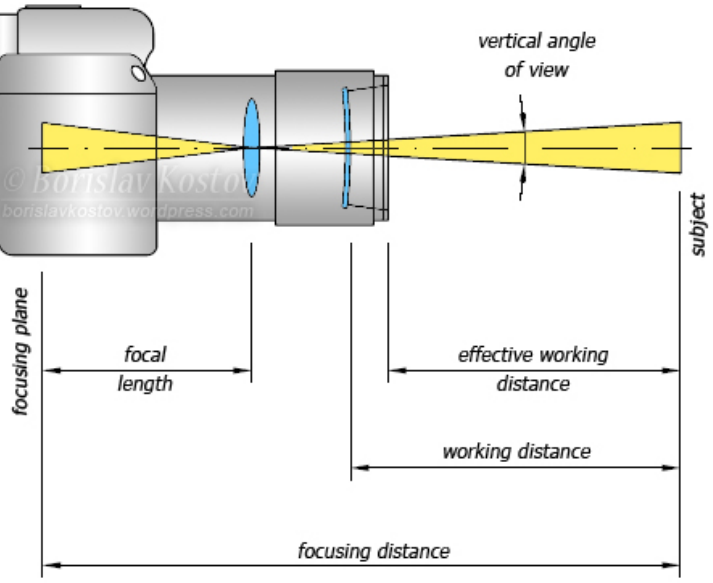

Fig. 2.4 Focal length

It is also essential to mention the magnification and the viewing angle in relation to focal length. The relationship between them is straightforward. If the value of the focal is high, the magnification will also be better, but the viewing angle will be extremely narrow. On the contrary, it will be hard to see what is in the distance, but the image will have a much more comprehensive range of what is in front of a camera if the focal length is short. That being said, there is another factor that contributes to the final focus distance, which is the aperture.

Aperture refers to how wide a lens is opened to let the light into the camera. The higher or larger the aperture, the more light the sensor can be exposed to. It is calibrated with f/stops. Lower f/stops give greater exposure because they reflect larger apertures, whereas higher f/stops give less exposure. After all, smaller apertures are represented in this case. It will become more apparent when people take pictures at different f/stops.

There are some conditions where users do not adjust their ISO and shutter speed.

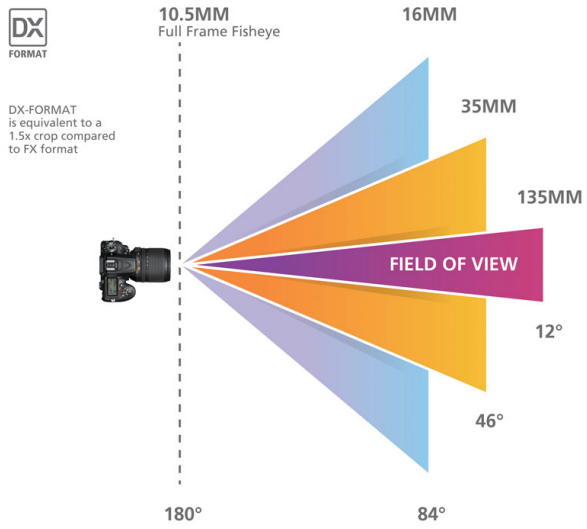

Fig. 2.5 Field of view of several focal lengths

Depending on what they want to capture, the aperture will be the key to proper exposure. Additionally, field depth is the sharpness of the objects in front of the lens (19). The aperture also decides this element in photography. The wider the lens opening, the lower the f/stop, the less field depth and the background blurrier. The greater the f/stop, the narrower the lens opening, the greater the field depth, and the sharper the context.

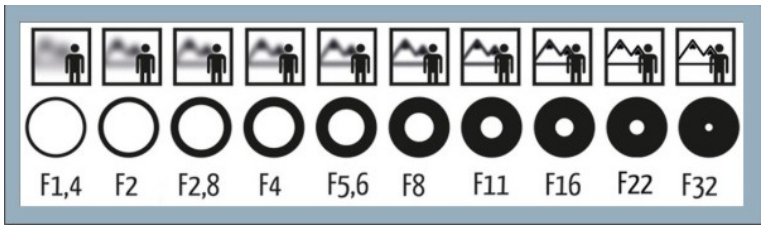

Fig. 2.6 F stop depth of field

However, the focal length is also affected by the size of the camera sensor, which is called the crop factor. There are many different types of sensors with unique sizes. Therefore, the captured image on each sensor will also have a different dimension even though the same lens is used throughout the process. This information should factor in the final decision when looking for a suitable camera. That is why, in order to make a consistent and meaningful comparison, it is best to use the same camera with one type of sensor and different lenses. The bigger the sensor, the higher the pixel density and the bigger the size of the image. It also has a broader view range compared to other smaller sensors. Nevertheless, it is not always an economical solution since full-frame, or even medium format cameras are often expensive, extensive, and challenging to handle or control.

Like other industry and scientific fields, camera photography or cinemaphotography also has its format and conversion between them. The standard for this conversion is the full-frame format. As its name suggests, it is a standard frame for every other sensor image to be compared against and converted to. A sensor can have a larger field of view than the full-frame one, called the medium format. It is currently the most prominent camera sensor in the consumer market, which is also a lot more expensive than other options. Other types of sensors are often smaller than the standard one, and there are plenty of them. The sizes also depend on the manufacturers' decisions. Take the APS-C sensor, for example. With the Nikon sensor, the APS-C is smaller than the sensors made by Canon or Fujifilm.

The size of the sensors can lead to something called crop factor, which is the decisive element of the field of view of the final image. Naturally, the larger the sensor size, the bigger and more detailed the image will be. Because of this simple reason, we have the crop factor based on the sensor size. To simply put it, the percentage between the size of the sensor is also the percentage that the image will be bigger or smaller compared with others. It is demonstrated in the image below how the same scene will look when captured by different sensors of various sizes.

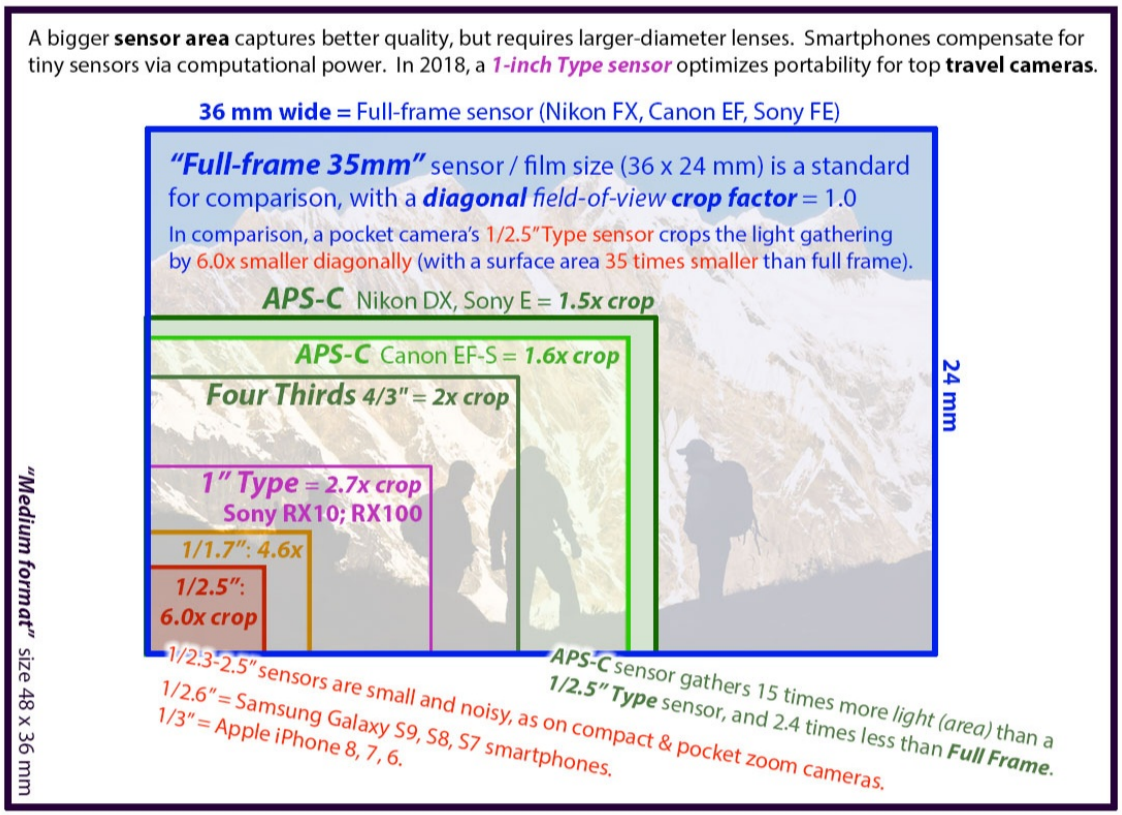

Fig. 2.7 Field of view of several sensors with different sizes

Usually, the 360-degree camera used in the industrial farm is relatively compact. Therefore, the size of the sensor has to be relatively small. In fact, as the specification of the product points out, it is only one inch in diameter. Because of this, we can calculate the sensor's crop factor by a simple calculation. As shown in the image above, the result is 2.7, which means that, for whatever the lenses are used on this sensor, the focal length of that sensor is 2.7 times longer than what is stated on the package.

For example, a 35mm lens will give the resulting image as if a 95mm lens is placed on the camera. The actual result is 94.5, but usually, manufacturers do not have any products with that specification. Furthermore, as the explanation about the focal length, a 95mm lens will give a much narrower view compared to a 35mm one. Fortunately, with a fisheye lens on this camera, the original focal length is 2.6mm, making the actual focal length around 7mm. This is good enough to capture a wide image, which is one of the strengths of these types of lenses.

2.2.2 Optical distortion of fisheye lenses

With increasing distance from the image center, radial distortion can significantly affect the image. In addition, short focal lengths have barrel distortion, which causes straight lines to bend outwards, whereas long focal lengths, or as many people call it, telephotos, have pincushion distortion, which causes straight lines to bend inwards (20). These two types of distortions are shown in the image below.

Zoom lenses have more of these effects than fixed-focus lenses. However, they are also rarely as optically crisp as prime lenses. Even with modern technology, the gap between these two lenses persists, especially when mentioning extreme lenses like the 18-200mm. Another downside is that they also have limited apertures compared to prime lenses, which is a considerable disadvantage. This also means that prime lenses may be more desirable in a low-light environment. Furthermore, therefore, many photographers highly recommend a prime lens, as professionals call a fixed lens, over a zoom lens. Nevertheless, this radial distortion can be fixed in the post-processing step with image processing.

The focal length is usually less than or equal to 24mm for these types of lenses. Fisheye lenses are often used when all we need is the center of a frame because it is less affected by distortion or when capturing a close-up of the subject. Many researchers have looked for a way to extract visually more stunning photos or computationally more convenient images from raw fisheye photographs due to their vast fields of vision, and

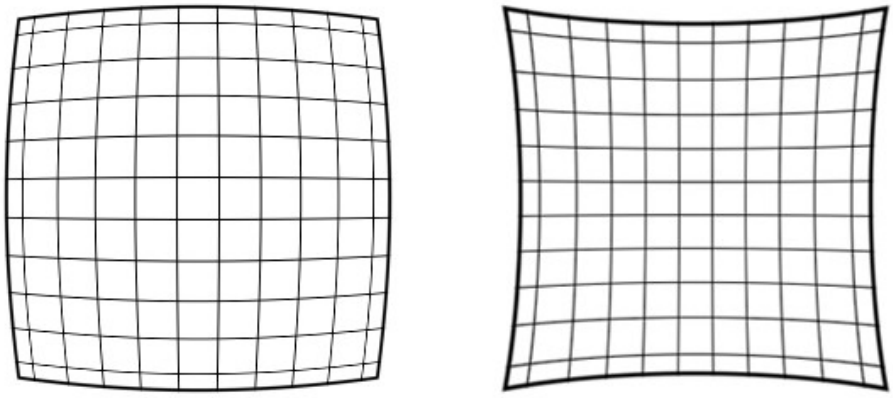

Fig. 2.8 Barrel and pincushion distortion

unavoidable significant distortion (21).

Many users dislike the fisheye effect because they simply do not like the distortion on their subjects, such as the barrel and pincushion distortion. However, when it comes to this kind of ultra-wide-angle lens, people still pay special attention to the image distortion it offers. Furthermore, as an inevitable result, the less distortion, the better the quality of an image, even though it will also come with a higher price.

Besides the noticeable distortions, the fisheye lenses have limited use-cases. Although using a fisheye lens gives a fresh and dynamic look to the subject and can make the photos unique, unlike any traditional images, that result depends on the human factors in a significant way. In short, if the operator has no idea how to control the camera entirely, the resulting image will often end up worst than intended. Because of that, the popularity of these lenses is still not as high as in others.

2.3 Popular 360 Degree and virtual reality applications

360-degree video has been used in many different fields and has achieved many unexpected successes. Suppose there is anything that can be considered the essential thing about documentaries. In that case, it should be to remind viewers of the natural world with all the precious moments, And there is no better way to achieve it than utilizing the advantages of the 360-degree format. This technology can let the user immerse in

the world and follow the stories in a way that the directors intended to.

The same can be said for the field of sports. Imagine experiencing a game of football or ice hockey with the same atmosphere without being in the stadium. That would sound impossible a few decades ago. Nevertheless, now, especially in this day and age when everything starts moving in the direction of remote working and online environments, it has become more popular than ever.

When creators want to convey their creative ideas, styles, and thoughts to the viewers, they often choose 360-degree videos because it offers limitless potential. Music video is one of the creative fields involved in this category. One of the biggest video platforms on the Internet nowadays, Youtube, has recently implemented new video formats called VR-180 or 360 degrees. They are such a fresh air to the viewing experience that they have attracted much attention from the users in a short amount of time.

Needless to say, the gaming industry nowadays has become so big that it cannot be ignored anymore. Moreover, one of the newest gaming experiences this decade is Virtual Reality Gaming. To show how significant it is, the market size of VR gaming in 2020 was more than 6 billion USD (22). This is an unbelievable number. However, recent events have changed the world in one way or another. Furthermore, Virtual Reality has come out on top and become a trend that a lot of big companies are trying to catch up with.

That being said, this new technology can also be used in other fields that not many people will think of, like farming. In this industry, daily monitoring is considered to be extremely important and also time-consuming. Nevertheless, with the right tool, this process can be less time-demanding, thanks to the help of artificial intelligence and computer vision.

3 Image processing in farming

For the last few decades, many aspects of our lives have been replaced by machines. This modern lifestyle has been bringing quite a lot of benefits for both users and the manufacturers and, therefore, will undoubtedly become more and more popular in the near future. Therefore, it is crucial for companies to implement new technology in their production chain in order to improve the quality as well as cut down the cost and errors during the process.

3.1 Monitoring in farming

In a specific case of fruit farms that operate on a large area of land, it becomes essential to count, check, and predict the quantities of the fruits. This is because, in order to reduce loss of time and money, the managers need to plan ahead and order in advance several tractors, trucks, and ships, predict cash-flow budgets, and schedule delivery estimates (23). Besides, the number of laborers required for the job is significantly less than what is currently being deployed in traditional methods, which can be used in a different stage of the whole process. This brings benefits not only to the manufacturer but also to the customers since the product's price will go down because of the reduction in the production cost.

Using the computer vision method for monitoring nowadays is considered to be one of the best approaches because it helps cut down the need for more laborers as well as reduce the amount of time needed. Nevertheless, for each type of fruit, some different statistics and characteristics need to be collected and analyzed before they can be applied to the algorithm. Therefore, the work for detecting and counting fruits is not close to being done. However, to apply the algorithm, a clear image is necessary. The process of collecting the data should also be automatic. It is impractical and time-consuming if there is a person who brings the camera around every meter and takes a picture of the whole row of fruits on a giant farm.

There is no practical way to take a whole picture of a line that long and not have any distortion. That is why a means to stitch the image together is an essential requirement for this job. Additionally, there are many types of cameras out there, and some of them can take a whole 360-degree image, which is helpful since it only needs to run once and capture the images of two opposite sides. This can save up much time for operating the camera and make the process much faster. However, the accumulated amount of time can be pretty significant if it is applied to a large-scale farm.

3.2 Data from tomato farms

The data used for this project are taken from a tomato farm by a 360-degree camera that is moved between the two rows of tomato plants. The device is put on a moving platform that goes from one end to the other, usually in a straight line. Some situations might require it to turn left or right. Nevertheless, in the scope of the thesis, it will not be discussed here. While moving, a 360-degree video will be recorded and used for further analysis.

Fig. 3.1 Ricoh Theta Z1 - a 360-degree camera

The camera used in this project is the Ricoh Theta Z1 with effective megapixels of 22.6, which will produce an image with a resolution of 6720 x 3360. The resolution is excellent in this case, but in general photography, the number of pixels only plays a part in whether the image is considered to be great or not. The camera processing also has a significant role to play, but it is out of scope for this project. The video resolution is around 3840x1920 and 30p, which is lower than the still image but still acceptable.

As we can see, the row of tomatoes is quite close to the camera itself. Therefore, a small aperture or a short focal length is required, which is the main feature of a camera like this. It is also essential to know that a short focal length plays an important role here. If a standard lens with a focal length above 15mm is used in this situation, most parts of the image will be cut out of the final image, including the top and bottom sections. Additionally, with a normal lens, it is hard or sometimes even impossible to focus on a subject that is close to the camera. Using 360 in farming has some positive effects, like covering both up and down angles, but it also creates a few problems like the distortion of the image, which will be the main point of the thesis.

Fig. 3.2 Camera on the field

4 Open CV

As far as the information on the Internet provided, this project was started in 1999 by Gary Bradsky while he was working at Intel with the hope of speeding up the process of Artificial Intelligent and computer vision by providing the standard infrastructure for everyone who works and shares the same interests on the field.(24)

This computer vision library is open-source and is one of, if not the most popular open-source libraries. Scientists and researchers use it for many purposes, including Computer Vision and Machine Learning. The most significant aspect of the OpenCV library is that it is released under a BSD license, which means that everyone can use it for their purposes with minimum restrictions. This also means that the library can be used for commercial purposes without paying any fee. Furthermore, this is, in fact, a massive benefit for many enthusiasts out there who want to contribute to the project and develop their own applications at the same time without worrying about any other legal problems. This helps expand the library and detect and fix errors that will inevitably appear in the project. The community around it also proliferates, which is a positive sign for an open-source product.

4.1 Advantages of OpenCV

Convenience can be considered to be the first reason that so many people choose to use this library. Because it is already so popular, there are already a lot of resources and documents on the internet that can be used in the developing and debugging process. Furthermore, since its first development at the end of the 20th century, productivity and reliability have been considered by many users to be relatively stable.

The library now has over 2500 optimized algorithms, including a wide range of traditional and cutting-edge computer vision and machine learning techniques. Such algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, and produce 3D point clouds from stereo cameras. Moreover, they can also stitch images together to produce a high-resolution image of an entire scene, remove red eyes from images taken with flash, follow eye movements, recognize scenery, and establish markers to overland.

GPU acceleration has been added to the library for real-time applications in recent years. Moreover, it is also available in many programming languages such as C, C++,

Python, and Java. It also supports Windows, Linux, Mac OS, iOS, and Android, which are almost every popular operating system nowadays. Therefore, the library is on a trajectory to keep developing, and its popularity will be increasing. That means this library is considered future-proof and will have a high chance of becoming more well-known in the future.

OpenCV is designed to support computation efficiency. If written in optimal C or C++, this library can utilize multi-core processing. Moreover, this is a massive deal for a few reasons. At first, it is necessary to mention the speed of the C++ programming language compared to other languages, for example, Matlab. As researched by Tyler, C++ is over 500 times quicker than Matlab code in terms of processing performance (25). Furthermore, because the libraries are written in C or C++ programming language, they can also be optimized in the future to be more efficient and take advantage of multi-core processing.

4.2 Other alternatives

Especially if the project is academic-related, Matlab and Mathematica are excellent and suitable software systems for sketching or validating ideas in a short amount of time. Mathematica 8 comes with a comprehensive feature set for image processing, linear algebra, numbers, GPUs. Meanwhile, with Matlab, users will have access to a large amount of code from other researchers. Prototyping, visualizing and evaluating will be quick and easy (26). Nevertheless, whatever developers have developed in this environment can be be challenging to put into production because of its speed (27). This is a significant drawback for different reasons.

Let us say that what if a programmer spent a considerable amount of time on a prototype project and finally made it work. However, for on-field deployment, he or she will need to program everything from scratch again but in a different environment, language and library. This is undoubtedly an extremely time-consuming process that comes with too many potential risks. On the topic of converting into a different programming language, depending on the purpose of the code, users may experience memory or performance issues because of the lack of understanding of how the new environment could be optimized. Other tasks like interacting with databases and web servers will not be easy, and sometimes, it is not even an option. The last disadvantage is that software like Matlab will cost more money than other available options.

4.3 Programming languages

Programming languages are also important because of their efficiency. C++, for example, is what is used for many production-level computer vision systems, and it is for a good reason. For instance, we can think of something at the scale of the image search engine, street view from Google, or any other commercial application. In order to have such a smooth experience with that software, there is no doubt that they are optimized from the lowest level with C++. However, it is not nimble for prototyping and quite terrible when trying new ideas because the development time is slower. Moreover, in the hands of inexperienced programmers, it can be challenging to keep track of performances with many instability issues.

Python is, as many consider, an easy-to-use solution when it comes to prototyping. Users can use it for both Matlab type numeric computing library like NumPy, or that has constraints on libraries like OpenCV. Moreover, people can implement systems and data structures with it and get acceptable performance. So, Python is widely used nowadays as a compromised solution between the academic and industrial world because of its compatibility.

4.4 Final conclusion

Due to the nature of OpenCV, which is an open-source library, it is comparatively harder to learn. The lack of documentation and error handling codes are often the main reasons. Although there are many tutorials about this topic, most of them are just basic, and some might even require a deep understanding of programming itself to accomplish the task. More often than not, this disadvantage is what makes novice computer vision users lean towards MATLAB. Nevertheless, once a person gets used to OpenCV, some professionals suggest sticking with it as it is the most comprehensive open-source library for computer vision and has a large user community.

Some professionals also suggest MATLAB as it is helpful for rapid prototyping, and its code is straightforward to debug. Moreover, it has good documentation and support. However, the disadvantage surrounding MATLAB is that it is not open source, the license is a little bit costly, and its programs are not portable. However, MATLAB is an entirely scientific suite that consists of a massive IDE with its language.

Therefore, it is pretty clear that MATLAB is suitable for exploring computer vision concepts as researchers and students at universities that can afford the software. However,

OpenCV is relatively more convenient while building production-ready and real-world computer vision projects. Therefore, it is a better choice to use OpenCV in order to develop it into a complete product in the future. Python will also be used as the programming language of choice since it is easy to implement as a proof of concept.

II. PRACTICAL PART

In this practical section, we will discuss the various algorithms that are used to achieve our goal of this thesis, which is creating a stitched image from a 360-degree video. In practice, many approaches can be used to obtain the result. Nevertheless, this thesis will focus on a few key aspects.

At first, we will discuss how the video is processed since this is the core procedure that will be used in all of the algorithms. Then, to simplify the process, we will try creating an image from a small section of each frame. The width of the section will be fixed in this step with a pre-defined number of pixels. This is far from the best method, but it is the easiest and fastest way to demonstrate that the overall procedure is working correctly as expected.

Secondly, it is crucial to have a method to undistort fisheye-type of images. This process can be executed before or after generating the resulting image. While the OpenCV has already implemented such a function, it is not explicitly made for an equirectangular image that is extracted from a 360-degree video. Therefore, the thesis presents the cube map conversion method and some other undistortion approaches for specific parts of the image like the top and bottom sections.

Furthermore, during the processing of recording the video, errors in movement might occur and affect the speed as well as the image quality. Therefore, the result might appear disjointed. To tackle this problem, an algorithm to detect the movement speed and vertical movement from the video will be implemented. An additional step is needed to ensure the quality of the results. It will evaluate all the algorithms mentioned above with one test video.

5 Video processing

5.1 Extracting frame

As a video is the result of a series of images or frames, it is necessary to extract each and every single one of them and process them separately. However, for a program where speed has a high priority, it is recommended to shorten the process as much as possible by not saving all the images on the disk and reloading them when necessary. Those processes will take up a significant amount of time and storage if the frames retain all of their details. Therefore, a frame should be processed immediately after being extracted from the video by being saved in a temporary buffer.

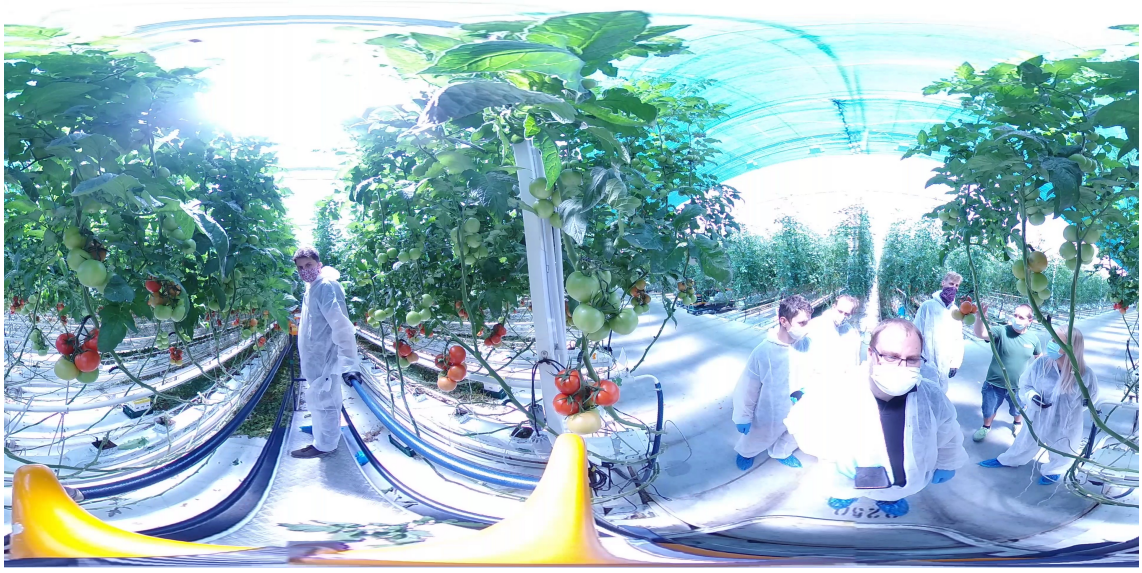

Fig. 5.1 Original frame from the video

5.2 Modifying

The images from the video that arrived from the farmhouse are shifted from the center due to the software limitation. This means that the two rows of tomatoes are not appropriately separated into the left and right sides. Therefore, an additional step to modify them is required. However, this can be considered to be an easy step since the video provided by the camera, which has been modified once, has a particular fixed setup parameter. Therefore, it is as simple as cutting the 1/4 part of the image from the right side and concatenating it to the left side of the rest of the image. The result is shown in the image below. As we can see, two rows of fruits are now in their correct position, and we can continue to the next step.

Fig. 5.2 Modified frame

5.3 Image compression

Resolution is the standard for the quality of an image. The higher the resolution is, the more pixels that image contains. However, with modern equipment, photos are often taken with a massive resolution. Additionally, they also take up much storage space in the hard drives. Therefore, image compression was invented to handle this problem. This method will save images to a hard drive using as little data as possible.

Nevertheless, it is not a miracle. Using image compression often comes with some quality risks. As we increase the image compression, the quality of the photo will decrease. High compression means that pixels of roughly the same color are made into one color. This will eventually result in a less detailed image when we save it in a compressed format. Furthermore, those lost pixels cannot be retrieved again.

There are two types of compression that are widely used, lossy and lossless. Lossy is a filter that will remove some of the data. Of course, this will downgrade the image, so it is necessary to carefully check the documentation about how much of the quality will be reduced during the process. But, on the bright side, the file size can be relatively small. Lossless is a data compression filter. What it means is that the image will not lose any quality. However, in order to be displayed, it has to be decompressed first.

5.4 Image formats

Those two types of compression come with many image formats. In this project, the following format is considered to be used: BMP, PNG, and JPEG. However, each type of format has its pros and cons. For example, JPEG has two significant advantages in both the form of its size and loading speed. Nevertheless, since it is a lossy compression format, the decrease in quality makes it unsuitable to be used for the project that requires as much information as possible in the image.

Here is the result of the time take to load and save the same image of each format 10 times for all 3 formats using Python:

- PNG: 13.52s
- BMP: 12.27s
- JPG: 06.26s

In this project, the image is saved as a .bmp file which stands for bitmap file. It is chosen because the file can retain most of the details and colors from the original image, which is definitely crucial for the recognition process after this. Another reason is that the quality of the BMP format is better than the JPG format (28). That is no surprise since there are no compression methods are used that can cause the loss of information in an image while storing it.

One downside of the BMP format that should be mentioned is that the size of the file will end up becoming quite large and, therefore, takes up a lot of memory space both while processing and after processing. This can be problematic and will be addressed later. Although the PNG format has the same quality as the BMP format, since the program runs continuously and the image will be loaded from time to time or immediately after saving, it is just a slight advantage not to compress it.

6 Image stitching

6.1 Moving direction of the camera

As the camera moves between two rows of fruits, we will receive the images from both sides of the camera, resulting in a panorama video. Nevertheless, handling the data from each side is different from each other since the row on the left side, for example, can seem to be moving forward or to the right from our perspective, while the other side will look like it is moving backward or to the left. In this thesis, this direction will be considered as moving outward. The other direction will be called moving inward, and both will be shown in the images below.

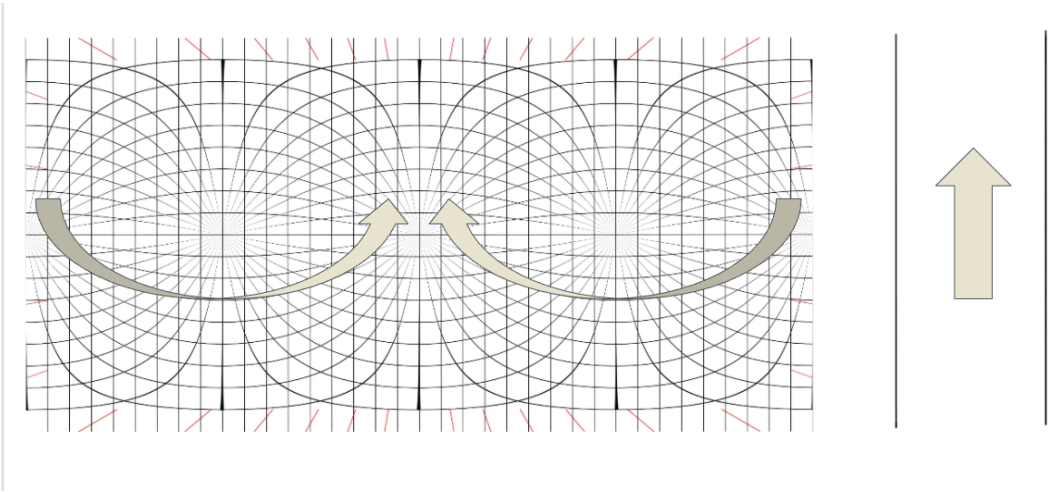

Fig. 6.1 Direction outward

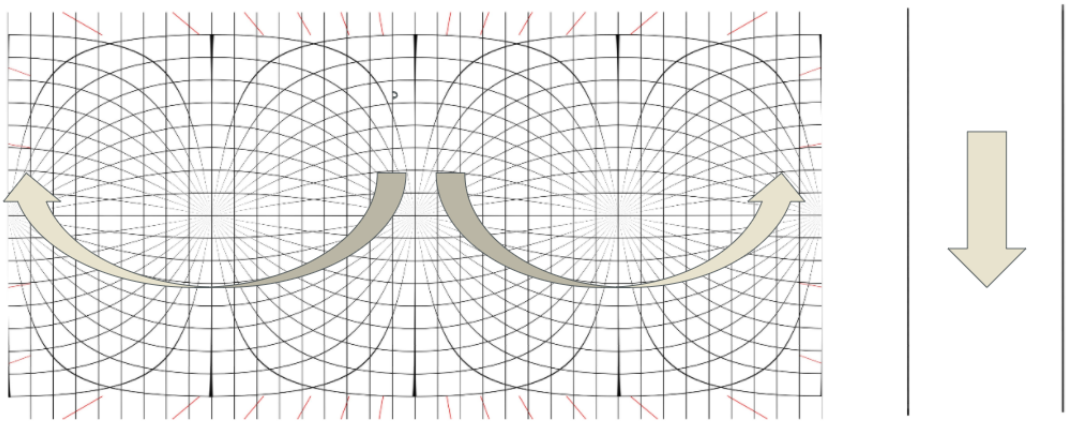

Fig. 6.2 Direction inward

These different directions are causing problems while stitching the images together. If the same stitching process is applied for both situations, one of the result images

will not appear to be as expected. In order to solve this problem, for the side that has the opposite direction of the stitching process, it is needed to flip the part of the image before stitching to the others. By applying this method, we can use the same stitching procedure without concerning the direction at this step and therefore make it separated and stable during the development process.

Fig. 6.3 Moving inward - image from the farm

In the example below, the stitching direction is from left to right, but the moving direction in the video is the opposite. Therefore, the result is not as expected and cannot be used for different purposes. On the other hand, if the image is flipped horizontally before being stitched together, we get a clean image of the tomatoes.

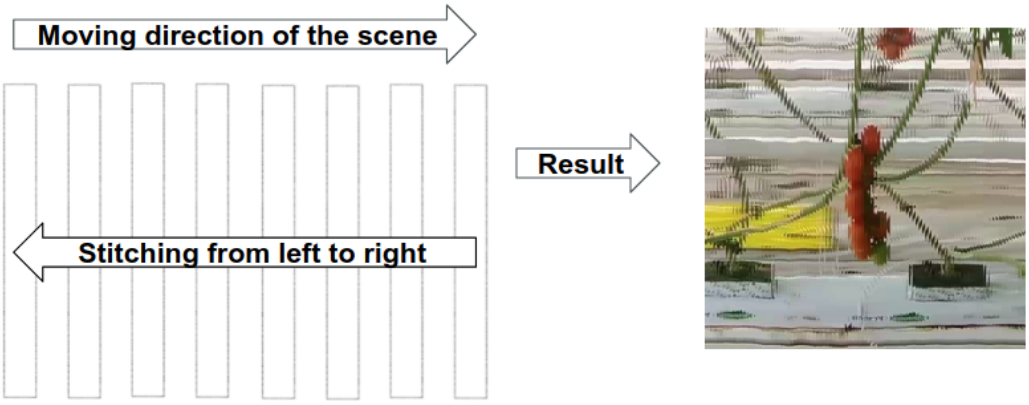

Fig. 6.4 Without flipping the frames

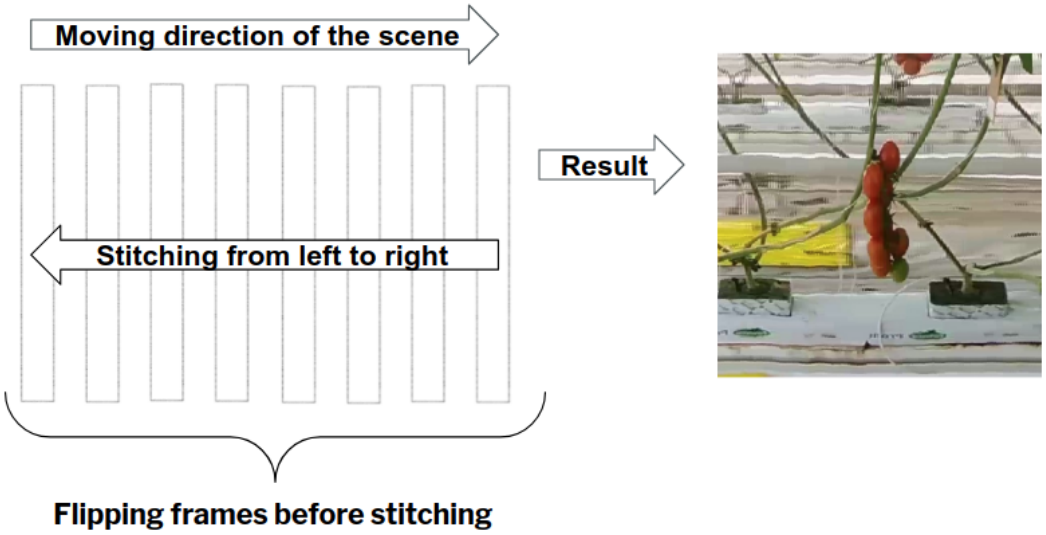

Fig. 6.5 Flipping the frames

6.2 The result after stitching

Fig. 6.6 Stitched image of the left row

Note that, in this chapter, the stitching process only used a fixed length for the added slice. As shown in the images above, the result after stitching is considered to be good, but there is one problem with it. The parts in the top and bottom of the image are compressed. This is a known problem when using a 360-degree camera. The solution will be discussed in the next section. The images can also be looked at from a left or right angle to see the whole row of fruits from different perspectives. This will also be discussed later in the thesis.

Fig. 6.7 Stitched image of the right row

7 Final undistortion

7.1 Leftover distortion

However, the final image after the stitching process still has some distortion toward the top and bottom sections. Therefore, our task has not yet been concluded. There is one more step that is needed in order to receive the best possible image for the tomato recognition process later.

As a 360-degree camera, it is built with a glass dom around the camera in order for the view from all directions to come into the lens at once. Therefore, using this lens makes it possible to achieve the whole 360-degree image without turning the camera around continuously. Nevertheless, the disadvantage of this is that, unlike the view directly in front of the camera, the light from narrow angles on the top or bottom of the lens will be squished, and thus, the distortion will be created. That is why an additional step to correct the image after stitching it together is essential.

7.2 In depth explanation

Imagine the camera as a spherical object that will take the view directly in front of it. Nevertheless, it is not just that plant simple. If the object is at the same level as the camera, it is unlikely to be distorted. However, the further and higher the object is, the more distorted it will get since the image of the whole area is shrinking down when it goes through the lens to fit in the final image.

It would be fine if the camera is put in a spherical environment like an observatory that is also a dom. The distance between the camera and the wall is the same at every angle. But, in reality, the greenhouse is usually a rectangular cuboid. In the illustration below, the blue line is supposed to be the wall or the row of tomatoes that we want to take a picture of. The red line resembles the final image that would have been reflected on the sensor by using the spherical dom class.

The length of AB and BC is the width of the view that will come into the lens and be reflected on the sensor as A'B' and B'C' accordingly. In this example, AB and BC are the same lengths as each other. Without applying any mathematical elements, A'B' appears to be shorter compared to B'C'.

In the context of $\triangle AOC$, the line OB is a median line that splits the triangle into the

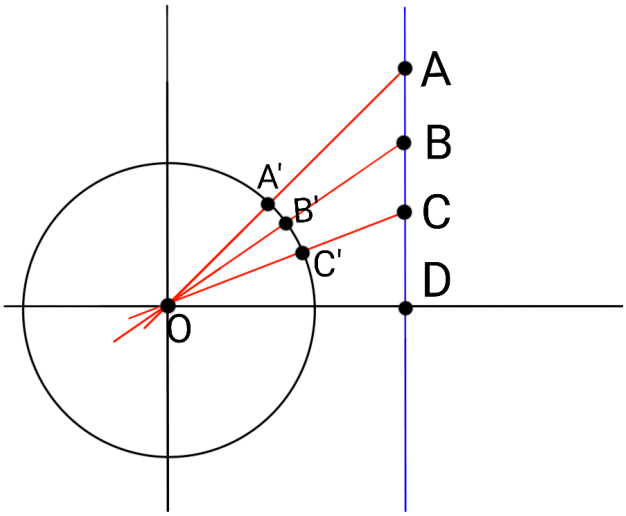

Fig. 7.1 Projection of the scene on fisheye lens

two separated ones with the same area. This can be proven easily by calculating the area of each of them. $\triangle AOB$, the area is equal to a half of the multiplication between OD and AB. The same can be applied in the context of triangle $\triangle BOC$ but BC will be used instead of AB. Nevertheless, since OB is a median line, AB and BC is as the same length as each other. Therefore, the two triangles have the same areas.

Speaking of calculating the area of a triangle, we can also use a different formula. The area of $\triangle AOB$ is calculated as follow: $\frac{1}{2}.OA.OB.sin(\angle AOB)$. And for $\triangle BOC$, it is $\frac{1}{2}.OB.OC.sin(\angle BOC)$. Therefore we have:

$$\begin{aligned} \frac{1}{2}.OA.OB.sin(\angle AOB) &= \frac{1}{2}.OB.OC.sin(\angle BOC) \\ \Leftrightarrow OA.sin(\angle AOB) &= OC.sin(\angle BOC) \\ \Leftrightarrow \frac{OA}{OC} &= \frac{sin(\angle BOC)}{sin(\angle AOB)} \end{aligned}$$

From the range of 0 to $\frac{\pi}{2}$, the larger the value of the angle, the larger the sin value of that angle. Moreover, since OA is longer than OC, $\angle AOB$ is smaller than $\angle BOC$ which also means that A'B' is shorter compared to B'C' even though they are the reflections of the views with the same size. Therefore, the top and the bottom part of the panorama image usually appear to be compressed.

7.3 Equation for resizing image

In the illustration below, O is supposed to be in the position of the camera sensor and the circle around it is the lens that we use in the farm. The blue line starting from N is the row of tomato that we want to capture.

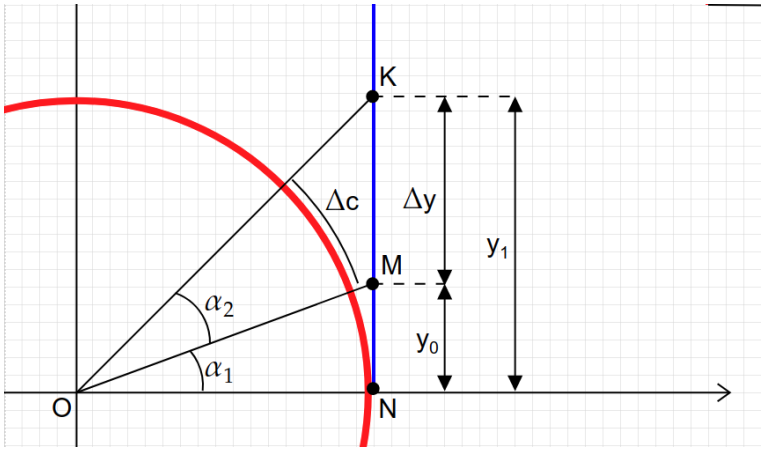

Fig. 7.2 Illustration of the camera in the environment

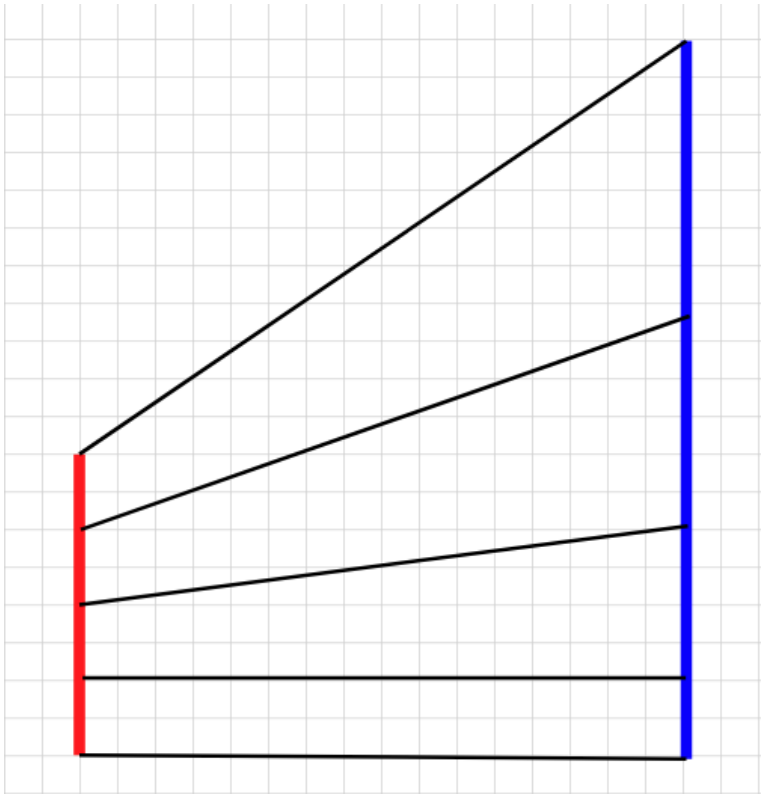

Fig. 7.3 Illustration of a 360-degree camera

We have

$$y_1 = y_0 + \delta y$$

with

$$y_0 = ON.tan(\alpha) \qquad\qquad y_1 = ON.tan(\alpha + \delta\alpha)$$

Since the angles increases with the same value, we can call the increased angle as $\triangle\alpha$. We will have:

$$\begin{aligned} &\triangle y = y_1 - y_0 \\ \Leftrightarrow &\triangle y = ON.tan(\alpha + \triangle\alpha) - ON.tan(\alpha) \\ \Leftrightarrow &\triangle y = ON.(tan(\alpha + \triangle\alpha) - tan(\alpha))(1) \end{aligned}$$

At the same time, from the circle circumference formula, we can calculate $\triangle c$:

$$\triangle c = \frac{2.\pi.r}{\frac{360}{\triangle\alpha}} \qquad\qquad \triangle c = \triangle\alpha.\frac{2.\pi.r}{360}(2)$$

From (1) and (2), we have:

$$\frac{\triangle y}{c} = \frac{ON.(tan(\alpha + \triangle\alpha) - tan(\alpha))}{\triangle\alpha.\frac{2.\pi.r}{360}}$$

Here, it is necessary to consider that the rest of the image should be scaled accordingly to the middle part of the same image. Furthermore, as the red line is supposed to be the lens on the camera, therefore, ON and r should be equal to each other. Another point is that c is 1 pixel of the source image, and the 2.pi/360 is the transformation to radian.

$$\triangle y_n = \frac{(tan((n+1)\triangle\alpha) - tan(n.\triangle\alpha))}{\triangle\alpha}$$

7.4 Experiment with real data

Here, it is possible to see the difference before and after applying the algorithm in the real-world data from a tomato farm. On the right side is the modified image. The shape of the tomatoes is not as tight as the ones on the left side.

Fig. 7.4 Modified vs unmodified result

8 Cube map

In order to handle the distortion of a panorama image, there are a few methods that can be used for this project. Furthermore, this cube map method is the stand-out approach to the problem at first. It is because the method is efficient (29) and the each face of the image after projecting is good with very little to no distortion.

A cube map is a form of environment mapping in computer graphics that contains a set of six square textures that resemble the faces of a cube. The scene in the panorama image is projected onto the cube's sides and saved as six square textures, each representing the views along a specific world axis (up, down, left, right, forward, and back). For better understanding, imagine a globe that is put inside a cube and then projected onto the inside surfaces of the cube. This method can be considered a well-known example of how the map of the earth is made.

8.1 Equirectangular Projection

The projection of a spherical mesh unwrapped on a smooth rectangular plane surface is known as the equirectangular projection of the sphere. That is a simple projection of the latitude and longitude of the spheres on the horizontal and vertical coordinate systems. It is often referred to as "non-projection" or "rectangular projection" as no scaling or transforming is applied.

The resulting frame created by the equirectangular projection often appears to be warped. This is because the artifacts in the middle are spatially flattened and stretched to the top and bottom. The equirectangular frame has a ratio of 2:1, and each half of the image contains a view of an angle that is equivalent to 180-degree. Because of that, it covers 360-degree horizontally and 180-degree vertically.

8.2 Gnomonic projection

The gnomonic projection is a map projection that can be accomplished by projecting a light ray from the middle of the sphere through the sphere's surface. For example, the point where they meet in the figure below will be called point P1. This ray will then goes on and touches the plane at point P. It should be noted that the plane is fixed to the sphere at point S. This point mapping from P1 to P is called a gnomonic map projection.

Fig. 8.1 An equirectangular frame

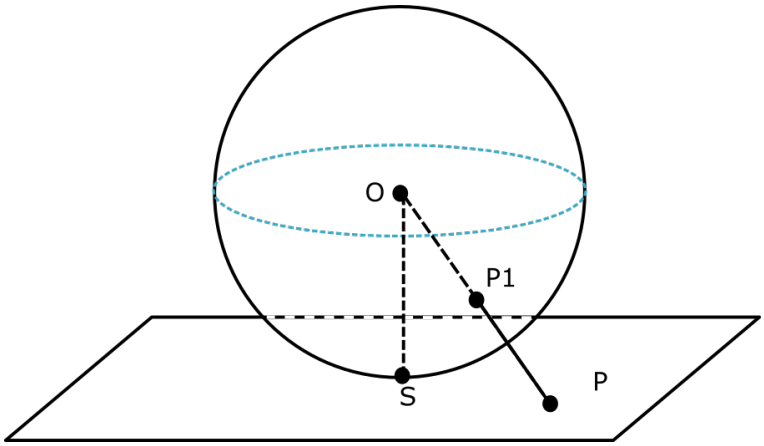

Fig. 8.2 The gnomonic projection

There is zero distortion at the tangent point, but it increases when we step away from it. Gnomonic projections are used to remove the narrow field of view from an equirectangular image. This is because a person's field of view usually has less distortion than a full view of the picture since it is smaller and closer to the fixed point between the sphere and the plane.

8.3 Cube map conversion

With the idea clear and ready for implementation, it is time to convert an equirectangular image into a cube-mapped picture. It starts with the cube and the coordinate. The mission is to project the sphere onto the six faces of the cube, as explained in the previous section. However, first, order needed to be determined between those surfaces for the ease of coding. The order of the six planes can be as in the image below as it is widely used.

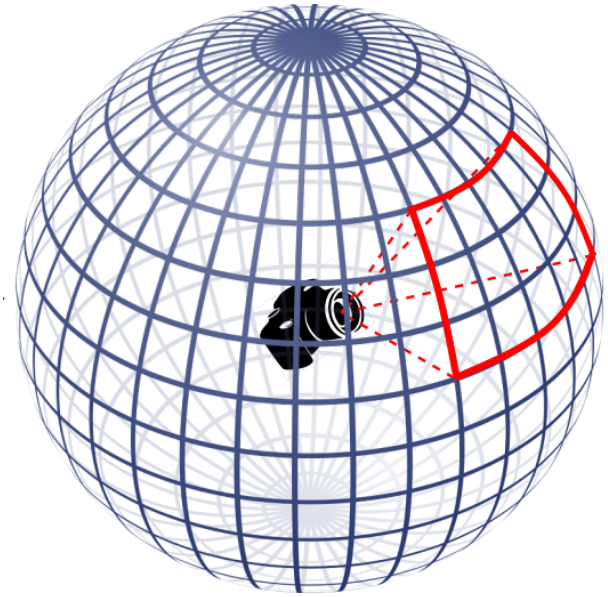

Fig. 8.3 Area of a normal field of view

Nevertheless, an image does not use the longitude and latitude system. It is just simply a matrix of pixels from the different colors that are arranged in that specific order. That is why it is necessary to convert the position of each pixel in the original image to the longitude and latitude system. After finishing this step, it will be possible to map the two images together. As illustrated in the gnomonic projection, it is also possible to set the latitude and longitude system for the cube map as well.

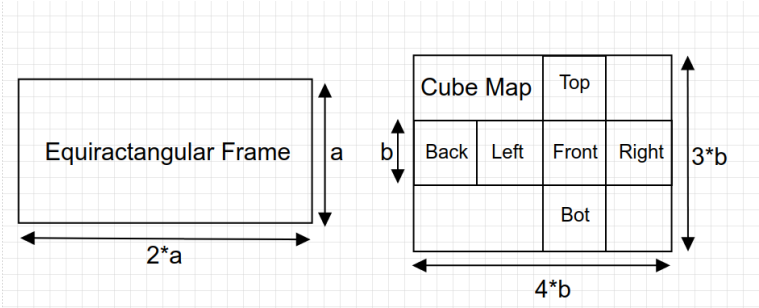

Fig. 8.4 Equirectangular image vs Cube Map

With the two images now can be mapped into one system, it is time to execute the conversion. After being mapped into a cube and then flattened on a surface, the image will contain six separated square areas. The area of the output image will also be double the size of the original one.

8.4 Result of the cube map method

The results are as expected, but it soon comes to the realization that it is not good enough for the final image. As this method only focuses on a small area of the surface, i.e., one face of a cube, the rest of the frame will still be distorted when stitched together. Take the example of the front face when stitched with the top and bottom face. If there is nothing done to the cube map, it is easy to recognize the line between those parts. Nevertheless, with some slide modification, the top and the bottom part still contain some distortions the further it is from the center.

Fig. 8.5 Original image

Fig. 8.6 Cube map

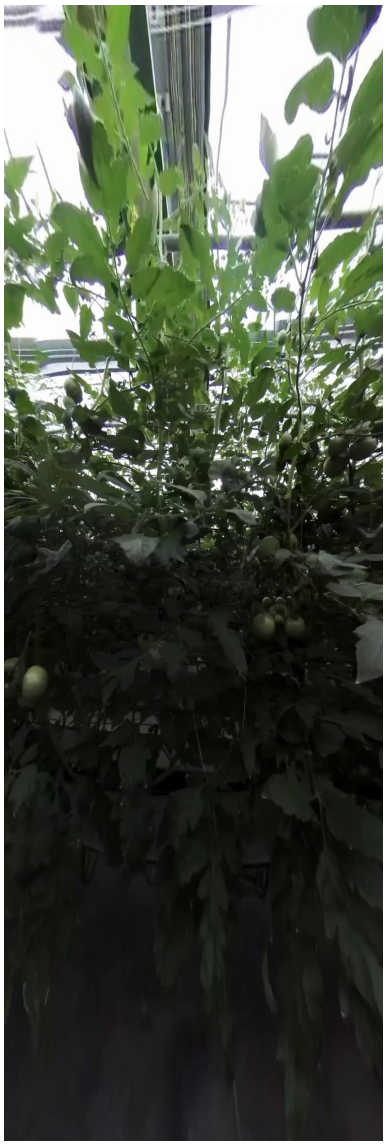

Fig. 8.7 Distortion the further it is from the center

9 Movement errors

Since the number of pixels needed per frame is relatively small, it is a good practice to check if there is any unexpected movement between the frames or how far each frame shifted from the previous one. It is possible to set the speed for the trailer that carries the camera. However, there will always be some impact along the way that may or may not affect the final result. For example, small obstacles like a rock can appear on the track. Upon running over it, the camera can move up and down and thus will create movement errors.

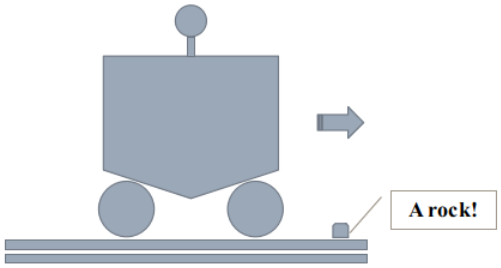

Fig. 9.1 Distortion the further it is from the center

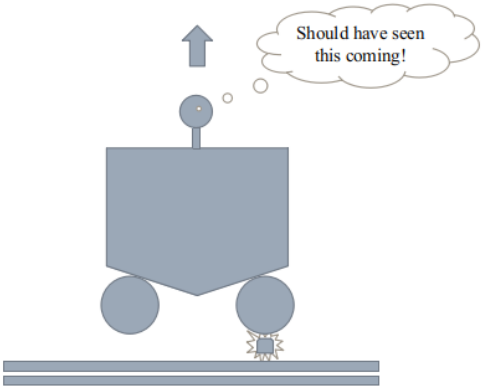

Fig. 9.2 Distortion the further it is from the center

9.1 Key points detection

The solution concept for this step is quite simple in theory but maybe not so much in practice. Basically, the program will compare the two images, in this case, are the two continuous frames, in order to spot if there is any movement and how big it is. In order to achieve that, a number of key points will be detected on both images and

marked with the coordinates. If the coordinates of the same key point on two frames are different, then the camera was moved during the run. This method, in practice, can also calculate the shifting pixels horizontally and stitch the necessary amount of the pixels.

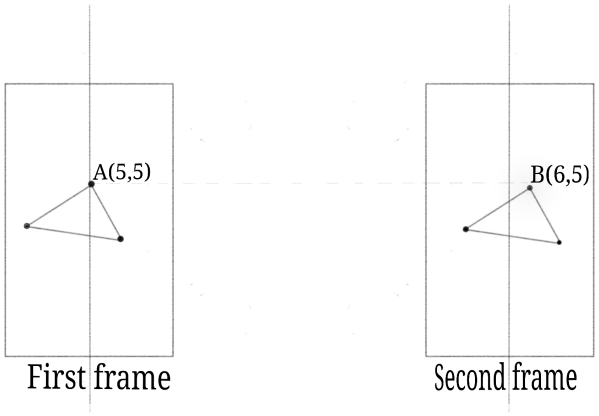

Fig. 9.3 Key point method

The popularity of feature detectors and descriptors in image processing has recently increased significantly. They have been used in the fields of computer vision with a variety of vision applications that rely on a small number of key points to describe pictures, including object identification, classification, matching, and tracking. A feature refers to a piece of information in digital pictures that can be considered as interesting sections of a picture. For example, edges, corners, blobs, and ridges are usually the characteristics that the algorithm should be looking for.

Feature detection is a method for computing and determining whether or not there is a specific image feature at each image location. A low-level image processing procedure is called feature detection if it is frequently the initial operation done on an image. First, it checks each pixel to determine if it contains a feature. After that, additional ways to describe the key points in an image are needed. To do that, the algorithms extract valuable information from the visual data.

There are several techniques encoding the information offered by various feature description techniques. Furthermore, within OpenCV, those methods have already been implemented and are ready for use without any licensing issues. The most popular method is called SIFT - Scale-invariant feature transform. Putting an image through SIFT, what we have as a result will be key points. Each object in the image will produce a lot of different key points. After that, we can distinguish these key points from each other through a descriptor. After applying SIFT transform, we will get keypoint

coordinates and a descriptor for each key point.

Another method is called FREAK - fast retina keypoint, which is a descriptor that tries to mimic human eyes. Similar to SIFT, the steps here also include a sampling pattern that specifies where points in the region surrounding the descriptor should be sampled (30). When constructing the final descriptor, sample pairs are used to choose which pairs to compare. In a nutshell, FREAK employs a machine learning approach to choose the best sample pairings.

Fig. 9.4 Comparing key points in two frames

As in the image above, between the two images, there are a lot of key points that have been detected. Since the two images are the same size as each other, it is possible to use the coordinate of those points in one picture and compare it with the corresponding points in the other picture. For example, the coordinate of one point in the first image is [298:500], but the equivalent of that point in the second image has the coordinate of [300:500]. Therefore, this point has been moved 2 pixels in the right direction.

Nevertheless, one point alone on each image might not be enough in this situation. It is a good idea to consider a large number of points in order to calculate the average movement between the two images. This is not the perfect method, but at least the shifting pixels can now be calculated dynamically and adapted to the real-world situation. However, the result from the test yielded some unwanted outcomes. The execution speed was many times slower than standard stitching and therefore increased the processing time by a significant amount.

9.2 Images differences

Another solution for the movement errors is calculating the pixel differences between the two frames. This can be achieved by taking a fixed part of the previous frame, usually the center part, and using it as a control part. Now we will try to compare the same area on the current frame, but we will move from the same position to the left or right, one pixel at a time. No matter the results, there will always be one position that has the slightest differences from the control part. Moreover, based on the position of that one frame, it is more or less approximately the position of where that control part has been shifted to. Therefore, it is possible to calculate how much of a gap the current frame has been shifted from the previous frame.

Take the example below as a demonstration. First, a section with a width of 100 pixels from the middle is cut from the previous image as a control part or, as we call it, a "slice." Next, on the current frame, we also select a few sections or "slices" with the same width and height as the control part, but this time, each selection will be moving one pixel further from the original position of the control part. Here in the image, that position is marked by the middle line. It will be shifted in both left and right directions, but only the right side is considered in this example. After that, we will calculate the differences between the control part with the newly selected sections.

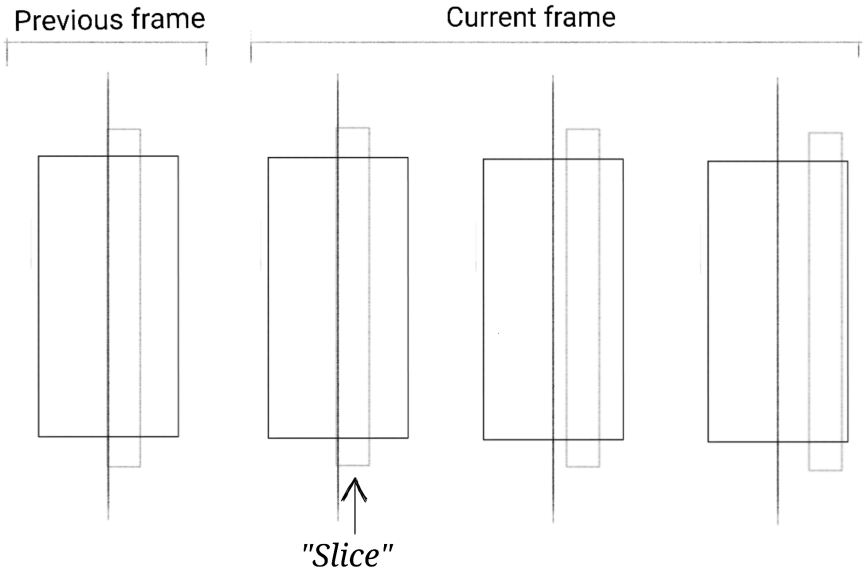

Fig. 9.5 Image Differences Method

In this illustration, let us assume that the "slice" in the third position has the lowest

differences compared to the control part. Therefore, it is more likely to be the most suitable one. Furthermore, because it is in the third position, it has been moved from the middle line to the right by two pixels. Moreover, as a result, we can assume that the current frame has roughly been moved by the same amount of pixels to the right compared to the previous frame.

In the case that the lowest difference belongs to the section that is shifted to the left, it can have several meanings. If we do not already know the direction the camera is moving, we can use this information to solve that problem. On the other hand, if we are sure that the camera is moving to the right side, this means that there is a stutter during the run, and this frame should be skipped from being stitched to the resulting image.

Nevertheless, to firmly confirm the direction, the validation should be done at the beginning while the platform's speed is steady and about to be sped up. This is like the time that not many errors will happen. This extra step will be possible to confirm if the frame is actually shifted in the wrong direction or not before discarding that section entirely from being added to the final image. Last but not least, to improve the result, the process of unwrapping the equirectangular image can be applied using the cube-map method. We can use a small section of the unwrapped image for comparison since there is little to no distortion in that section of the image.

10 Implementation of the algorithm

10.1 Setting up the environment

The programming language of choice for this thesis is Python, which is famous for the purpose of quick implementation and presentation. Python can easily be installed in many different operating systems, and it will be guaranteed to perform as expected. The recommended operating system, as well as the one that is used in this project, is Linux. It is simply because, for many Linux distributions, it usually comes with pre-installed Python. Therefore, the rest of the setup task is to download and install the updated OpenCV library. The specific version of Python in this project is Python 3.8.10.

After installing Python 3 as well as the OpenCV library, the next step should be to configure the project structure. All the configurations can be changed in the "Parameters.py" file. Firstly, it is possible to change the directory to where the source code is actually stored as well as the resources folder where all the panorama videos and the output images can be stored. Secondly, the important thing to keep in mind is that all the codes have to be put in one separate folder. The project can be run as quickly as entering this line "python3 main.py" in the terminal from that folder. Whenever a new image is created, there will be a small notification line on the terminal to let the user know.

10.2 Splitting the resulting image

Here in the project, the final image is divided into smaller images containing a small section of 1000 frames. This is because, for most commercial computers and laptops currently on the market, it is not possible to hold such a large number of pixels in the memory since we are using the BMP format to store it without decreasing the image quality. However, the most important reason is that the larger the image, the longer the program will have to take to process it. Therefore, in order to address these issues, a solution in the form of dividing the final image into smaller images has been proposed.

For example, every time we try to stitch a small middle part of the new frame into a compilation of the previous frames, it will get slower and slower with the higher the total number of frames that we have been processing. In order to illustrate this, an example has been prepared to calculate the time of processing from frames 1 to 100,

Frames	Not-Splitting Image	Splitting Image
1-100	3.699	3.760
101-200	3.789	3.295
201-300	4.192	3.299
301-400	4.969	3.306
401-500	5.425	3.329
501-600	6.277	3.275
601-700	7.122	3.279
701-800	7.629	3.286
801-900	8.129	3.286
901-1000	9.093	3.501

Tab. 10.1 Execution time between non-splitting and splitting method

101 to 200, and 201 to 300. The result will then be saved in a text file and shown as a result.

In order to make it as fair as possible, a specific program has been re-written from the main code, which only contains the process of modification and stitching of the images. Nothing else is added to this test program. There is no saving the image, which also contributes to the increasing time since the more extensive the image, the longer it takes to save it. Nevertheless, since we want to make it as straightforward as possible about how storing more data in the RAM can significantly impact the processing time, that part is unnecessary here. This test program also ignores some time-consuming tasks in other steps, like looking for key points in different images to calculate the shifting distance. The results can now be comparable by using the time library in Python, which can help us calculate the time it takes to run the application. The time will be saved after processing every 100 frames of the videos.

Looking at the numbers from the table, we can see that, without dividing the image into smaller pieces, the time it takes for the whole process to complete will increase with each added section. However, this is just a simple task of stitching the images together. Those numbers will be added up even more if the entire program is used for this test. Moreover, it will always be possible to stitch them afterward. Because of that, there is no need to finish the whole process in one go.

10.3 Different variants

With all of the information above, there are many ways to create an algorithm that will probably achieve the goal of generating a complete image from a 360-degree video.

The easiest way is to stitch a small section from each frame together. This section will have a fixed width that is applied to all the frames in the video. This is the best method to use in the ideal condition where there is no error in movement, and the camera's speed is stable throughout the whole process. This method is a solid one, but there is not much we can do to improve it.

However, there is no such perfect condition in the real world to begin with. Errors will always appear, and the best we can do is to adapt to those changes in the movement of the camera. Therefore, algorithms that can calculate the shifting distance between two continuous frames and use that value for the stitching process have been created. This makes the algorithm more dynamic and resistant to changes. In addition, there are two methods for detecting the camera's speed that stand out compared to the others, calculating differences and using key points.

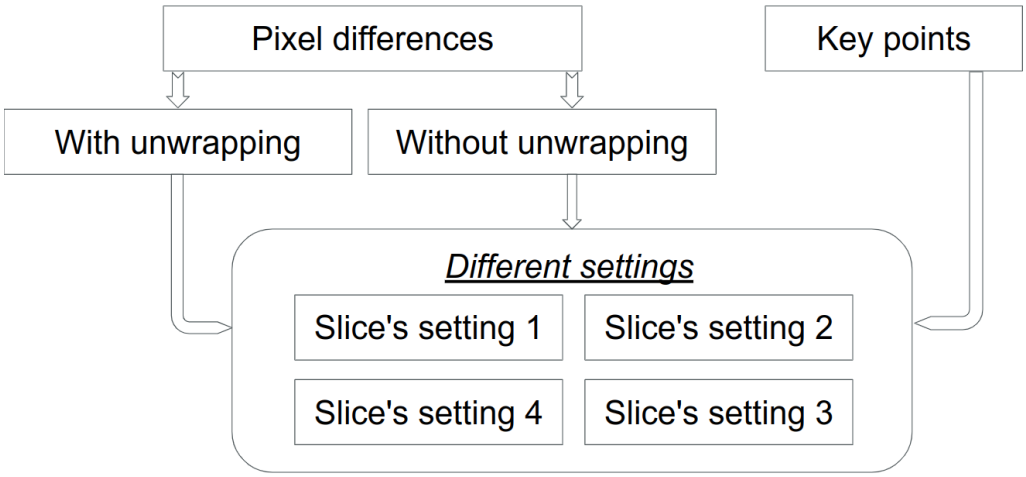

Fig. 10.1 Variations of the algorithm

The first approach can be executed in two different ways. We can either calculate the differences directly from the frames extracted from the camera, or we can unwrap it first and then do the calculation to see if less distortion can make the comparison more precise. Nevertheless, for the method that includes finding key points, it might be a good idea not to keep the two images as close as their original.

11 Evaluation of the algorithms

11.1 Test video

With these variants of the algorithm, a means to evaluate them is necessary in order to select the most capable one and apply it to real-world data. Nevertheless, the problem is that the only videos that have been used for testing are the actual data from the greenhouse. Thus, it is impossible to determine which one of the resulting images is the best since we have no way to compare the results or what to compare them to. Therefore, the idea of creating a simple test video that simulates the camera movement of the actual video is the way to go. However, this time, all the data, like the expected result as well as the speed of the camera, can be controlled.

Fig. 11.1 The first created banner for the test video

Fig. 11.2 The second created banner for the test video

With the Unity program, which is mainly used for game development, creating a simple 3D environment that can simulate the greenhouse is possible. This environment will resemble a tunnel with two walls on the left and right sides decorated with a banner. The camera then starts moving toward the tunnel and records a 360-degree video. After the recording is available, it will be processed with all the algorithm variants. Finally, the result images will be compared directly to the original image that has been put on the wall.

The first created image is quite simple and, therefore, does not have enough noise and

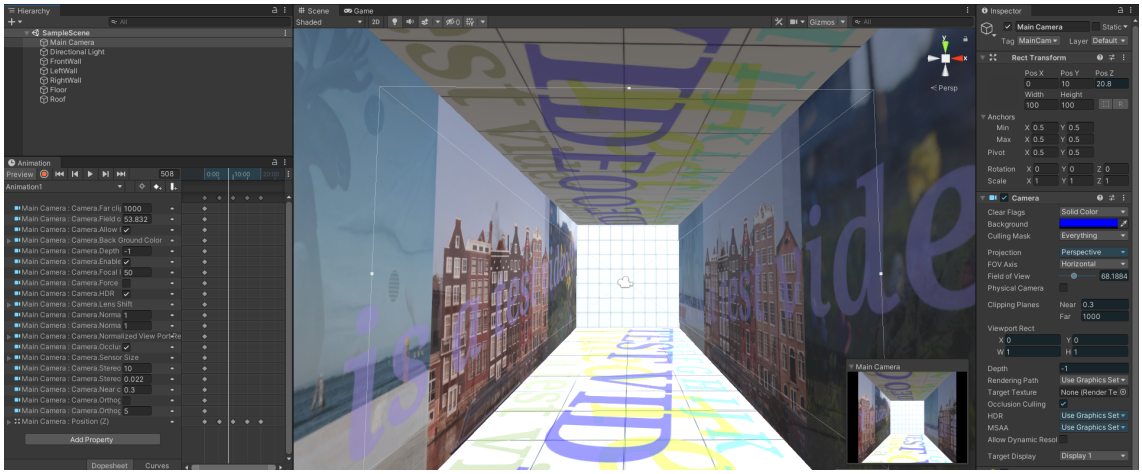

Fig. 11.3 3D model used in the test video

pixel differences for the algorithm to work correctly. Thus, a new and improved image is designed to achieve that goal. In the video generated from the model, the camera is moving at a constant speed and because of that, using a fixed value for the shifting distance is the best way to guarantee the desired result in this situation.

Fig. 11.4 The result from the test video

11.2 Errors quantification

In order to evaluate the algorithms, it is necessary to quantify the number of errors that a method will make during stitching images from the test video. We can achieve that by accumulating the number of pixels that the algorithm gets wrong. The detail for this step will be clearly explained shortly. Then, this result will be used to compare

with the others and find out which one has the minor errors count.

However, we need to figure out how to map this speed with the unit of pixels. Because the camera is moving steadily with an unchanged speed, this speed can be calculated since we know the original dimension of the image on the wall, which is 4000 x 1000. In a 360-degree video, the wall only takes up 50% of the height of the video. Since the video has a resolution of 1920 x 960 or 1080p, the height of the wall can be expected to be around 480 pixels. With that number, we can expect the resulting image in a perfect condition to have a width of 1920 pixels because the image's width is four times larger than the height (4000 x 1000).

The total time for the camera to move from start to finish is 12 seconds, and there are 30 frames per second in the test video. So, we will have around 360 frames in total. 1920 pixels over the course of 360 frames will give us the shifting value of around 5.3 pixels per frame. Therefore, we can expect that there will be a shift of 5 to 6 pixels for every new frame. If the result from an algorithm returns a number that is less or higher than 5 or 6, we will count it as an error. Furthermore, the error will be higher depending on how far that number is from those two values. For example, the algorithm returns 3. Then, 5 minus 3 is 2. Therefore, 2 will be added to the error count.

11.3 Testing and result

The first test is between the three approaches with four different settings for each of them. The first algorithm calculates the differences without any additional steps to undistort the image beforehand. The second method is to convert the frame from an equirectangular image to a cube map and use the least undistort section from the conversion to compute the differences. The last approach is to calculate the shifting distance between key points from the current frame and the previous frame.

Along with these three algorithms are the four settings that will be used to decide the width of the "slice" that will be used in the step of differences calculation in chapter 9. The first setting is defined with the width for the "slice" around 100 pixels, while the second and third settings are set to 150 and 50 pixels, respectively. After running these three settings and receiving the results, the slice with the broadest width has the best result. Therefore, a fourth setting is implemented with a width of 200 pixels to confirm this trend. Moreover, as expected, the latest setting brings minor errors among them. Continuing with this increment, around 200 to 300 is the sweet spot for

Setting	Normal	Cube map	Key point
1	76	182	274
2	75	173	219
3	95	236	-
4	68	114	175

Tab. 11.1 Error count for test video

these two algorithms, and a "slice" with higher width may become unstable and start causing more errors again.

Based on the numbers in the table, we can see that the key point method performs worst and in one specific setting, it does not work at all. One explanation is that the algorithm has difficulty finding the exact key points in two images when there are so many details and layers. On the other hand, the simplest method yields the best result.

Fig. 11.5 Best test video result between cubemap (left) and normal (right) method

As we see in the comparison between the two methods of difference calculation, some parts of the building in the image created by the cube map algorithm appear to be missing. This also caused the letter "a" to be distorted. On the contrary, the algorithm, without using any undistortion step at the beginning, generates a better result and is closer to the original image that is painted on the side walls.

With the knowledge from the results of the test video, it is time to apply these algorithms to the video taken from the greenhouse and test it. Although we do not have all the needed information as we have in the test video, after much testing both with a dynamic and fixed width of the "slice," it is safe to assume that for this particular video, the speed is also quite similar to the test video, at least with the first 1000 frames. However, we do not even have to trust the assumed speed. The video itself

Fig. 11.6 Best dynamic result for test video

Setting	Normal	Cube map
1	1848	2059
2	1722	2003
3	2072	2211
4	1668	1922

Tab. 11.2 Error count for real video

has a lot of noise and easy-to-spot errors if they ever occur, so we will be able to see the differences between the algorithms when trying to zoom in on a small section of the picture.

The two algorithms behave as expected, and the method that unwraps the image before calculating has more errors. Moreover, the time difference between these two methods is quite a big gap, with the cube map approach taking approximately 50 times longer than the other one in the case of testing the video from the greenhouse with only the first 1000 frames. We can then compare the best results from the two algorithms by zooming in on one part of the image. The tomatoes on the right, generated by the more straightforward approach, appear to be more rounded and easier to recognize than the ones on the left.

Fig. 11.7 Best real video result between cubemap (left) and normal (right) method

Fig. 11.8 Best dynamic result for real video

12 Additional improvements on the result

12.1 Tomato Recognition

When the resulting image is acquired, it is time to apply a tomato detection algorithm and see how it works. This algorithm is developed and trained by another team in this project. Nevertheless, it can also be used as an additional method to test the quality of the resulting image. As we can see in the figure below, most of the tomatoes are detected even though it is not perfect. However, that is an expected result, and some other improvements can be used to achieve a better outcome.

Fig. 12.1 Tomato detection

12.2 Image illuminating

As the lighting inside the greenhouse was not ideal most of the time, the final image usually appears to be much darker than expected. This is a problem because the fruits can not be detected when there is not enough information about their shape and colors. Therefore, a step to illuminate a specific part of the image is nice to have in the case of situations like this. With the help of so many libraries nowadays, it can be accomplished by changing the HSV color space. Additionally, increasing the contrast of the image can also make the details to become more visible.

Fig. 12.2 Increased brightness vs normal image

Fig. 12.3 Increased contrast vs normal image

12.3 Different angles

On the one hand, with a camera with a standard lens that is not fisheyes, it is impossible not only for an algorithm but also for humans to detect hidden fruits behind the leaf.

It is because most of the image is taken straight in front of the camera, and there is no data that includes the angle needed to detect the fruits.

On the other hand, with the 360 degrees camera, it is possible to detect the fruits that are hidden behind the leaves by moving the angle of the final image slightly to the left or right angle. This can be achieved because, with fisheye lenses, the variety of angles we can choose to see an image is way more than the traditional ones. Therefore, it is now possible to have a final image from a different perspective by moving the part that needed to be stitched by a certain number of pixels to those two directions. This whole process is to mimic the change of view as humans do in real life, as we move to the side to get a better look at what is hidden.

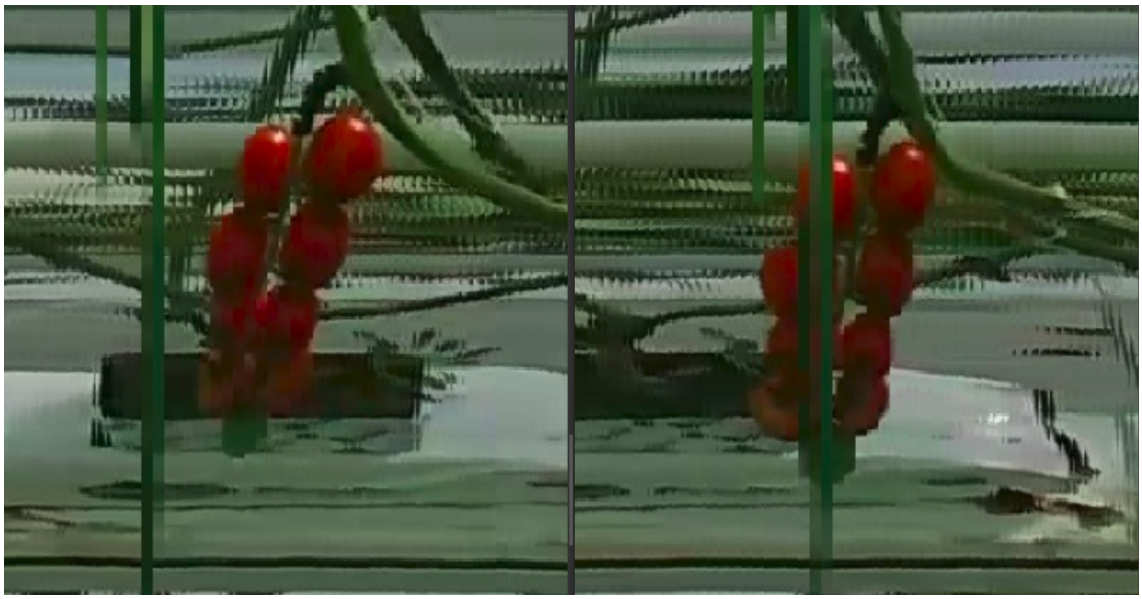

Fig. 12.4 Right vs Center result

CONCLUSION

In this thesis, most of the theory sections are introduced early. In chapter 1, we have the chance to get acquainted with the essential topics of image processing in computer vision and explore how the process was established in practice. All the necessary information has been mentioned in this chapter. Moreover, the technicality of the 360-degree cameras in general and the fisheye lenses have been explained in detail in chapter 2. Along with the core information, some examples and applications of this unique video type are also mentioned there.

Accompany the theory are the real-world practice. The data used for this thesis is obtained in the natural environment of a specific hydroponic greenhouse. This data is recorded in the form of a 360-degree video and discussed in chapter 3 with the help of one of the most popular computer vision libraries mentioned in chapter 4. However, errors can happen during the record session. These problems also come with several solutions discussed throughout chapters 5, 6, 7, 8, and 9. Finally, a few variants of the algorithm have been proposed in chapter 10 for preparing images that can be used in subsequent machine vision algorithms, most essentially CNN, to recognize the fruit.

The last part is also the most important one is to make the idea become a reality. Several versions of the proposed algorithm have been completed and tested on actual data from the farms. In order to evaluate the obtained results like what is written in chapter 11, a test video with a predicted result and controlled parameters has been used to ensure the quality of the final images between the variants of the algorithms. The calculating differences method without unwrapping yields the best result compared to the others. Although the result is not entirely perfect, it can now be used for the process of image recognition. Also mentioned in chapter 12 are the directions for future image processing developments in this area which will improve the quality of the image and another feature to detect hidden fruit.

REFERENCES

[1] Savage, N.: The search for secrets of the human brain. *Nature*, volume 574, no. 7778, October 2019: pp. S49–S51, doi:10.1038/d41586-019-03065-7.

[2] Faiz bin Jeffry, M. A.; Mammi, H. K.: A study on image security in social media using digital watermarking with metadata. In *2017 IEEE Conference on Application, Information and Network Security (AINS)*, Miri: IEEE, November 2017, ISBN 9781538607251, pp. 118–123, doi:10.1109/AINS.2017.8270435.

[3] Roh, Y.; Heo, G.; Whang, S. E.: A Survey on Data Collection for Machine Learning: a Big Data – AI Integration Perspective. *arXiv:1811.03402 [cs, stat]*, August 2019, arXiv: 1811.03402.

[4] Chen, Z.; Wang, J.; He, H.; et al.: A fast deep learning system using GPU. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, Melbourne VIC, Australia: IEEE, june 2014, ISBN 9781479934324 9781479934317, pp. 1552–1555, doi:10.1109/ISCAS.2014.6865444.

[5] Yang, X.-S.: Optimization Algorithms. In *Computational Optimization, Methods and Algorithms*, volume 356, edited by J. Kacprzyk; S. Koziel; X.-S. Yang, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ISBN 9783642208584 9783642208591, pp. 13–31, doi:10.1007/978-3-642-20859-1_2.

[6] Mihajlovic, I.: Everything You Ever Wanted To Know About Computer Vision. Here's A Look Why It's So Awesome. September 2021.

[7] Lam, V.: "We know very little about the brain": Experts outline challenges in neuroscience. November 2016.

[8] Verma, R.; Ali, J.: A Comparative Study of Various Types of Image Noise and Efficient Noise Removal Technique. *International Journal of Advanced Research in Computer Science and Software Engineering*, volume 3, no. 10, October 2013.

[9] Kingravi, H. A.: Nonlinear vector filtering for impulsive noise removal from color images. *Journal of Electronic Imaging*, volume 16, no. 3, July 2007: p. 033008, ISSN 1017-9909, doi:10.1117/1.2772639.

[10] Ansari, M. A.; Kurchaniya, D.; Dixit, M.: A Comprehensive Analysis of Image Edge Detection Techniques. *International Journal of Multimedia and Ubiquitous Engineering*, volume 12, no. 11, November 2017: pp. 1–12, ISSN 19750080, 19750080, doi:10.14257/ijmue.2017.12.11.01.

[11] Einevoll, G. T.; Destexhe, A.; Diesmann, M.; et al.: The Scientific Case for Brain Simulations. *Neuron*, volume 102, no. 4, May 2019: pp. 735–744, ISSN 08966273, doi:10.1016/j.neuron.2019.03.027.

[12] Arulprakash, E.; Aruldoss, M.: A study on generic object detection with emphasis on future research directions. *Journal of King Saud University - Computer and Information Sciences*, August 2021: p. S1319157821002020, ISSN 13191578, doi: 10.1016/j.jksuci.2021.08.001.

[13] Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu, China, 610065; Aamir, M.; Rahman, Z.; et al.: An Optimized Architecture of Image Classification Using Convolutional Neural Network. *International Journal of Image, Graphics and Signal Processing*, volume 11, no. 10, October 2019: pp. 30–39, ISSN 20749074, 20749082, doi:10.5815/ijigsp.2019.10.05.

[14] Cameron, J.; Gould, G.; Ma, A.: *360 Essentials: A Beginner's Guide to Immersive Video Storytelling*. Ryerson University Library.

[15] Layt, S.: AR: The cutting-edge technology slowly changing your reality. May 2021.

[16] Chinu, S.: How 360 degree virtual tours allow companies to expand its reach.

[17] Shalaginov, M. Y.; An, S.; Yang, F.; et al.: Single-Element Diffraction-Limited Fisheye Metalens. *Nano Letters*, volume 20, no. 10, October 2020: pp. 7429–7437, ISSN 1530-6984, 1530-6992, doi:10.1021/acs.nanolett.0c02783.

[18] Miller, N.: Focal Length: Definition, Equation & Examples.

[19] Forsyth, D. A.; Torr, P.; Zisserman, A.: *Computer vision - ECCV 2008: 10th European conference on computer vision Marseille, France, October 12-18, 2008 proceedings*. Number 5305 in Lecture Notes in Computer Science, Berlin Heidelberg New York: Springer, 2008, ISBN 9783540886938.

[20] Aber, J. S.; Marzolff, I.; Ries, J. B.: *Small-format aerial photography: principles, techniques and geoscience applications*. Amsterdam ; Boston: Elsevier, first edition edition, 2010, ISBN 9780444638236.

[21] Kweon, Gyeong-Il; Choi, Young-Ho: Fisheye Lens for Image Processing Applications. *Journal of the Optical Society of Korea*, volume 12, no. 2, june 2008: pp. 79–87, doi:10.3807/JOSK.2008.12.2.079.

[22] Virtual Reality in Gaming Market Size | Global Analysis [2028].

[23] Syal, A.; Garg, D.; Sharma, S.: A Survey of Computer Vision Methods for Counting Fruits and Yield Prediction. *International Journal of Computer Science Engineering*, volume 2, no. 6, November 2013.

[24] Kaehler, A.; Bradski, G. R.: *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, first edition edition, 2016, ISBN 9781491938003 9781491937990.

[25] Lee, H. J.; Goo, J. M.; Kim, N. R.; et al.: Semiquantitative measurement of murine bleomycin-induced lung fibrosis in in vivo and postmortem conditions using microcomputed tomography: correlation with pathologic scores–initial results. *Investigative Radiology*, volume 43, no. 6, june 2008: pp. 453–460, ISSN 0020-9996, doi:10.1097/RLI.0b013e31816900ec.

[26] Toulson, R.: Advanced rapid prototyping in small research projects with Matlab/Simulink. *2008 IEEE International Symposium on Industrial Electronics*, 2008: pp. 1–7.

[27] Elsayed, A. A.; Yousef, W. A.: Matlab vs. OpenCV: A Comparative Study of Different Machine Learning Algorithms. *arXiv:1905.01213 [cs, stat]*, August 2019, arXiv: 1905.01213.

[28] Chebil, J.; Al-Nabulsi, J.; Al-Maitah, M.: A novel method for digitizing standard ECG papers. In *2008 International Conference on Computer and Communication Engineering*, May 2008, pp. 1308–1312, doi:10.1109/ICCCE.2008.4580816.

[29] Chen, Z.; Wang, X.; Zhou, Y.; et al.: Content-Aware Cubemap Projection for Panoramic Image via Deep Q-Learning. In *MultiMedia Modeling*, volume 11962, edited by Y. M. Ro; W.-H. Cheng; J. Kim; W.-T. Chu; P. Cui; J.-W. Choi; M.-C. Hu; W. De Neve, Cham: Springer International Publishing, 2020, ISBN 9783030377335 9783030377342, pp. 304–315, doi:10.1007/978-3-030-37734-2_25.

[30] Alahi, A.; Ortiz, R.; Vandergheynst, P.: FREAK: Fast Retina Keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, june 2012, pp. 510–517, doi:10.1109/CVPR.2012.6247715, iSSN: 1063-6919.

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
FREAK	Fast Retina Keypoint
HSV	Hue Saturation Value
OS	Operating System
RAM	Random-access memory
SIFT	Scale Invariant Feature Transform

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES