DEVELOPMENT OF A LOW-COST VISION SYSTEM FOR FINDING CONTOUR AND SURFACE DEFECTS ON CAST IRON ENGINE COMPONENTS

Master of Science Thesis in the Master Degree Programme Production Engineering and Management

Alex Kiviorg

Thesis Supervised by:
Anders Berglund
Lorenzo Daghini
Peter L. Ekberg
Lars Mattsson

Department of Production Engineering and Management
School of Industrial Engineering and Management
Royal Institute of Technology
Stockholm, Sweden

June 2014

Acknowledgements

The author would hereby like to thank all persons involved in this thesis for the exceptional support, passion and sharing of knowledge. Despite the thesis being written by one person it is the result of co-operation and great effort of many persons at KTH and Scania.

Special thanks to (in alphabetical order) Anders Berglund, Lorenzo Daghini, Peter L. Ekberg, Lars Mattsson and Gabriella Svan. Last but certainly not least, thanks to supporting family members Elisabeth and Karl Theodor.

Table of Contents

Abstract

The thesis aims to develop an image processing algorithm for a 2D vision system for quality inspection of cast iron components (size 250x190x120mm). The system is designed to be low cost and be an easily implementable solution for an existing production line to complement or replace part of manual quality inspection.

The defects are mainly caused by gas porosity and molding errors and can have a very different nature starting from small cavities and chipped edges ending with large portions of missing material. Thus, the methods used in the algorithm must guarantee successful detection of very different types of defects while handling natural shape deviations in contours.

Several known image processing methods were tested and further developed to fit this specific application. First, suitable candidate methods were selected, implemented in MATLAB® and tested on a small set of manually taken test images. The best performing methods were then further developed into a fully functional beta algorithm that was then tested on real production line using prototype hardware. To understand the system's capabilities a blind experiment with quality inspectors was carried out.

The algorithm uses *Canny edge detection operator* for obtaining the contours together with several stages of *dynamic masking* and *morphological processing* to eliminate false edges and non-important contours. *Intensity mapping* and *filtering* is then applied to eliminate traces of cutting fluid on the part as well as occasional high-contrast milling marks. Surface defects are then measured and compared with an inspection standard to decide if part is approved or not.

Thereafter the contours of the part are checked for deviations caused by porosity and damaged molds. *Shape signatures* and *segment-wise alignment* with a template signature is the method that proved to be sufficient in most cases. The method has shortcomings in terms of determining the size of deviations in certain segments due to the complex shape and significant natural variations in contours. Ideas are proposed for improvement of the system's accuracy and robustness.

Sammanfattning

Den här examensarbete syftar till att utveckla en bildbehandlingsalgoritm för ett 2D vision systemet för kontroll av gjutjärnskomponenters (med storlek 250x190x120mm) kvalitet. Systemet är utformat för att vara billig och lätt att implementera i en befintlig produktionslinje för att komplettera eller ersätta en del av nuvarande kvalitetskontroll.

Defekter orsakas huvudsakligen av gasporositet och andra gjutningsfel. Dessa yttrar sig som små håligheter inne i materialet eller som att flisor saknas i kanterna på komponenten. Således måste de metoder som används i algoritmen kunna upptäcka flera olika typer av defekter och samtidigt hantera den naturliga avvikelsen av form som förekommer i konturerna, som kommer av att godset är gjutet.

Flera kända bildbehandlingsmetoder testades och vidareutvecklades för att passa denna specifika applikation. Först valdes lämpliga metoder ut och implementerades i MATLAB ® och testas i en liten uppsättning av manuellt tagna testbilder. De bästa metoderna vidareutvecklades sedan till en fullt fungerande beta-algoritm. En prototyp togs sedan fram och metoden testades sedan i en verklig produktionslinje. För att utvärdera hur kapabelt systemet var, i jämförelse med dagens manuella kontroll, genomfördes ett blind experiment.

Algoritmen använder *"Canny edge detection operator"* för att detektera konturerna tillsammans med flera steg av *dynamic masking* och *morphological processing* för att eliminera falska kanter och icke - viktiga konturer. Därefter används *Intensity mapping* och *filtrering* för att eliminera spår av skärvätska på ytan samt enstaka spår som lämnas i biten efter fräsningen som ger hög kontrast. Alla ytdefekter mäts sedan och jämförs med en kontrollstandard för att avgöra om komponenten är godkänd eller inte.

Därefter kontrolleras konturerna för avvikelser orsakade av porositet och skadade formar. En metod som kallas *shape signatures* tillsammans med *segment-wise alignment* visade sig vara tillräckligt för detta i de flesta fallen. Metoden har dock vissa begränsningar när det gäller att bestämma storleken av avvikelserna i vissa segment på grund av den komplexa formen och stora naturliga variationer i konturer. Idéer föreslås för förbättring av systemets noggrannhet och robusthet.

Introduction

Quality issues, as pores, are sometimes not detected before late in the production line, when a lot of value has been added to product. The quality inspection is still often performed by humans by manual inspection, despite large portion of the production having been automated in developed countries. This means that the inspection is dependent on the operator. Furthermore, industry has to cope with increasing production rates and tougher quality standards. This has open the field for machine vision to become an area of great interest as quality inspection could be performed automated and operator independent.

Recent developments in imaging and computer technologies have made it possible to develop advanced high-speed on-line vision systems but on the other hand the technology has become remarkably affordable over the past decade which allows implementing vision systems in manufacturing in a much larger scale.

With that though in mind, this degree project was proposed to investigate the possibilities of using a low-cost 2D vision system for automated quality inspection of cast iron motor components. The scope of the project is to study and develop image processing methods for such vision system and build a prototype system for testing the algorithm.

Thus, the thesis involved literature studies, large amount of programming in MATLAB®, designing and building a prototype system and finally testing and comparing it with current visual inspection. All this was done in a six month period with an aim of finding out if the system would have the potential to be implemented as a permanent part of a production line.

The scope of the thesis was limited to developing and testing a prototype system and the details of the necessary support systems for the final system are not discussed in this degree project. Moreover, the system was not designed to check part's dimensions.

Background Information and Used Methodologies

The thesis is a part of a larger research project involving Scania and KTH investigating various non-destructive testing methods and one of the areas of interest is the methods for detecting surface and contour defects of casted components using a simple automated vision system.

The aim of the project is to study suitable image processing techniques and develop a prototype algorithm and vision system to test the methods and the potential of such approach to finding defects.

During the thesis suitable candidate methods were looked for by studying literature and research articles about image processing and vision systems. Test images were manually taken in a similar environment as the part would be inspected in the real production line to develop an algorithm in MATLAB® to test the potential before investing in a prototype system.

As the initial algorithm showed potential, there was a need for a large database of consistent images for further developing the algorithm as it was clear that the algorithm must be able to find large variety of different defects.

Before the test system was set up the algorithm was developed using the test images which were very inconsistent and did not represent a large variety of the possible defects. Thus, Adobe® Photoshop™ was used to create realistic images with defects of different size and in different locations. The lesson learned is that in order to develop a reliable algorithm a large set of consistent images is required.

Setting up the prototype system enabled just that and after a week of strenuous testing and continuous bug fixing the algorithm and the hardware were considered to be mature enough to carry out a blind experiment with human inspectors to further calibrate the parameters and tune the algorithm to better match human perception of the defects.

State of the Art

Vision systems have been developed for manufacturing environment since digital imaging technologies with a reachable price level were available. That was when CCD sensors were developed in early 1970s [**1**] and enough low-cost computational power was available to perform image processing. Though it wasn't until the 1980s and 1990s when the machine vision industry really took off due to advancements in computer technologies. In 1990s vision systems were becoming wide-spread solutions to various industrial problems and it can even be said that today machine vision is the key enabling technology for some industries such as the manufacturing of semiconductors and advanced computer circuitry. [**2**]

According to Automated Imaging Association, the world's largest machine vision trade association, the machine vision is in in North America alone a 2 billion dollar industry with unconfirmed total market revenue of 4.5B. It has experienced a rather steady 2 digit growth over the past decade, excluding 2009 when global manufacturing industry was suffering from an economic downturn. [**3**] Already in 2004 the estimated number of vision systems sold in North America was 53 000 and at that time the estimate for 2008 was 72 000 systems. [**4**]

The main drivers for growth are considered to be high labor costs in developed countries and new applications supported by advancements in technology and research for image processing algorithms. The applications of machine vision in manufacturing can be divided into six larger categories; code-, object-, position recognition, completeness check, shape and dimension check, surface inspection. Each of them can be further divided into subcategories and mixed with others. [**5**] The resulting vision system is therefore often a specialized system created for a specific task. The application areas investigated in this thesis are qualitative surface inspection together with shape check. This thesis does not focus on part dimension check.

Traditionally, human operators have the task to visually inspect manufactured parts for various quality defects [**6**]. Although, humans may in many cases be more capable than machines it is a known fact that they get tired, distracted and can be slower than machines. Thus, using a machine vision system as full replacement or to complement human inspection can increase productivity, free up resources for other tasks and give a competitive advantage [**7**].

As the preceding discussion showed, machine vision is a relatively young and rapidly growing field. The interest is strong and the amount of available literature is vast with new or improved image processing algorithms being continuously developed and proposed. This means that in most cases there are several if not tens of different methods for solving a single problem and it is not trivial to define which method is best in each case. Though, it can be said that machine vision is highly generic and its basic methods such as edge detection and object location can be rather easily adapted from one application to another [**8**]. Where things might get more complicated is that the inspected component's size, shape, tolerances, required speed of inspection and the environment in which the inspection is carried out can vary significantly. This creates a need for very different methods although the principle task is the same.

The process of developing a vision inspection system is a stepwise procedure that involves several stages of testing, revision and often starting over again when tests fail. Moreover, specification of the defects the vision system is ought to find cannot always be given in a non-fuzzy manner and if specifications are not clear it is hard to derive a solution. Therefore, developing a vision system is a complex task involving both hardware and software development with no guaranteed success. [**8**]

Even though there is plenty of research articles presenting various methods for solving a generalized image processing problem such as image segmentation, shape descriptors or image registration, literature that explaining an existing machine vision system and its image processing algorithm in detail seems to be difficult to find. The used algorithm is rarely described in detail enough that would allow easy replication of the system by another researcher or institution and in most cases, the focus is on a specific novel method or hardware configuration developed [**9**,**10**]. The most likely reasons are both keeping the article short and that most systems are probably built for commercial purposes or involve a strong commercial interest.

Therefore, instead of trying to find an article about a similar vision system, the focus of literature studies during this thesis was on investigating various digital image processing methods, which could potentially solve the problems presented in this thesis, implement them in code and test. There is vast amount of literature for digital image processing in the form of books, guides, research articles and commercial content (blogs, websites and newsletters). Large portion of the methods and image processing knowledge used in this thesis comes from three popular image processing books – Digital Image Processing by Gonzalez and

Woods, Computer and Machine Vision: Theory, Algorithms, Practicalities (4th Edition) by Davies and Digital Image Processing Using MATLAB by Gonzalez et al.

The greatest challenge of this thesis was to select the methods that have the highest potential and could be implemented and tested with reasonable time and effort. There are image processing algorithms such as SURF [**11**] or SIFT [**12**] which are mathematically very complex, have taken long time to develop and have been patented (SIFT). While some complex algorithms are already built into MATLAB® and its Image Processing Toolbox™ or Computer Vison Toolbox™, many potential methods are not easily available in form of code and must be implemented by the author which can be very time-consuming and has no guaranteed outcome.

There are several feature-rich commercial image processing/machine vision software available from market leaders such as Cognex and Keyence although they might come with a high price level (no price quotation was received in time from either company) and do not have as high customizability when compared to native programming or MATLAB which is a good development tool that is available (including Image Processing Toolbox™) free for KTH students. One can also find low-cost commercial software such as RoboRealm® [**13**] which comes with a surprisingly comprehensive feature list but is more targeted for robot vision.

There is also OpenCV, a highly capable open source image processing library for various programming languages (C++,C, Java™, Python) that also has many powerful functions (also 3D) and could be used for machine vision systems [**14**]. MATLAB® was the chosen software for developing the algorithm in this thesis due to its great documentation, good support and availability. On the other hand, OpenCV has a similar set of functions [**15**] and uses a native programming language which makes it the preferred choice for final implementation of the algorithm.

1. Case Study

A case study was carried out as part of the thesis to better understand the problem that the vision system should solve. Quality inspectors were interviewed and worked together with at the quality check looking into various defects that the parts could have. Moreover, the production line was studied to find the optimal place for the prototype vision system. The main outcomes of this case study are presented in the following paragraphs but to honor the confidentiality agreement not all info and details can be included.

1.1. The Inspected Part and Production Line

The part is a casted iron motor component with approximate dimensions of 250x200x120mm. It enters the production line in unprocessed state and is then machined and components are added to it throughout the automated line. Note that in this thesis the part is discussed and shown as it is at the place of inspection by the vision system under development.

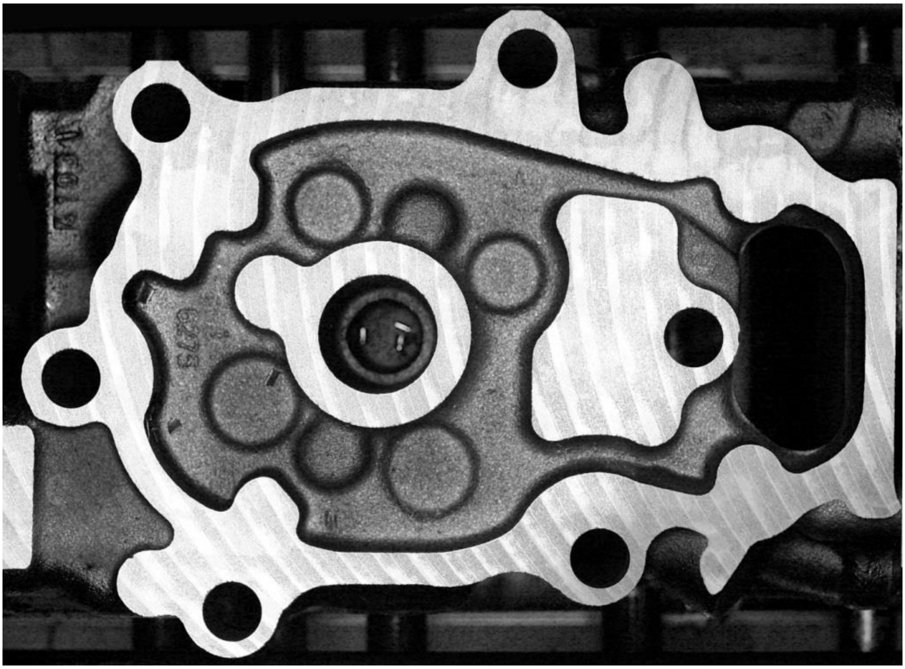

Figure 1 The inspected part on conveyor

The part has a fairly complex shape with three distinct surfaces, six drilled holes and one large non-machined hole in the middle. In the center of the part is a pool that will be machined later down the line. The part comes in two different versions which have slightly different contour shapes and the other one has two surfaces instead of three. The overall dimensions of both versions are approximately the same. Only one of those two types is discussed in the thesis but the methods of inspection are developed with keeping both types in mind.

Machining is done in various steps, thus the appearance and feature set of the part changes significantly throughout the line as it gets further processed. The parts are available for visual inspection at several locations on the line and also in different conditions. At some locations the parts are highly contaminated with metal chips and cutting fluid while at other places the parts have been cleaned and dried and are in perfect condition for inspection.

Naturally, the first place that comes to mind when searching for a location for the vision system is the earliest place in the line where inspection is physically possible. This kind of approach would minimize waste in the production line and ensure the highest ROI for the system. Following this idea, a suitable place for the vision system was found to be after the first stage of milling when all sides of the part have been face milled. Due to spatial constraints the first possible location for the test system was not feasible and thus a place a little bit further down the line is used for the test system.

In that location, only one side of the part is available for inspection without additional measures. Fortunately, that side is the one which has statistically most defects which is good for developing and testing the algorithm. If all four sides need to be expected the part should be turned on its side and then rotated 4x90degrees on a table. This could be done with a relatively simple actuator or by placing the system in a nearby robot cell. Though, designing support systems is not part of this thesis.

1.2. Contamination

At this location the parts have either come out directly from the milling machine or from a small buffer. In the first case the part can be highly contaminated by chips and cutting fluid or in the other case the parts might by dry but have small bits of plastic from the pallet on them. The contamination looks very similar to how a defect would look on the image thus making it nearly impossible for a vision system that uses a single image as input to differentiate the metal chips and pieces of plastic from the defects. Moreover, the cutting fluid is in large spills all over the surface of the part and it can be highly reflective depending on the angle of light.

There can also be strong contrast between the cutting fluid and clean surface which causes difficulties for edge extraction and detection of false edges.

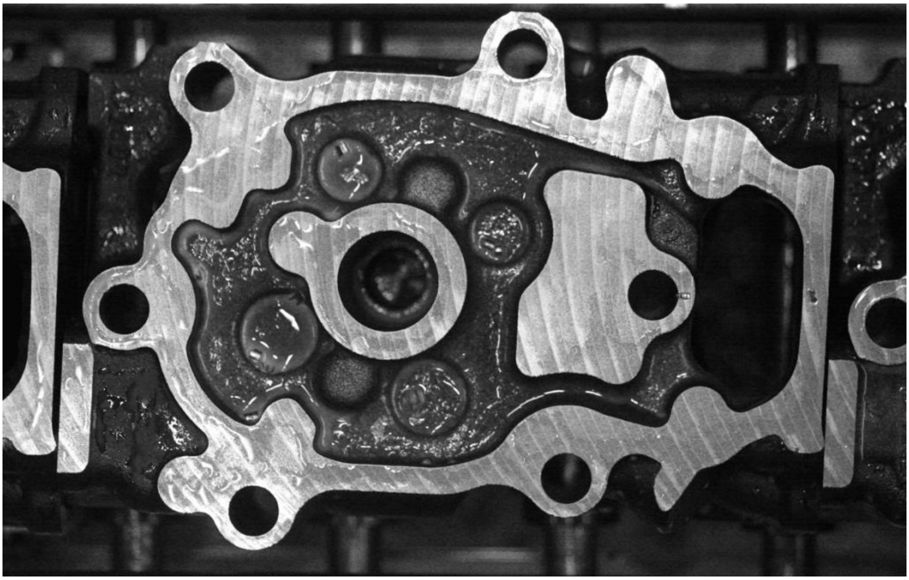

Figure 2 Part with contamination of metal and plastic chips (from a pallet) as well as traces of cutting fluid

The contamination issue was further investigated and with varying degrees of success the cutting liquid and chips can be removed by at least the following three ways:

1) **Industrial washing station**. Washing stations are used as a part of the line but at a fairly late stage. They are able to automatically remove all solid contamination and remove traces of cutting fluid and oil. At the end of the cycle the parts are dried. Adding an additional washing machine upstream the planned location of the vision system is an effective but expensive solution. Additionally, there is not much room for adding another machine to the line.
2) **Pressurized air**. Another option is to use a so called air knife which creates a thin wide stream of air which would blow off the cutting fluid and chips. This method is fast and requires little space but its reliability is questionable as the part has a large pool of cutting liquid and chips in the middle of it and the pressurized air might actually cause some of those chips to end up on the surface instead. Another

issue is that the chips and cutting fluid might get splashed on nearby equipment and sensors if proper shielding is not in place

3) **Mechanical removal**. The third option is to use a simple set of brushes that mechanically remove all solid contamination as well as most of the cutting fluid. The part will not end up completely dry but the amount of fluid on the surface is kept to minimum so the reliability of the image processing algorithm would be largely unaffected. Due to its low cost, simplicity and no major drawbacks this method is considered to be the best out of these three.

1.3. Type of Defects

The vision system is supposed to find various casting defects on the parts. The nature of those defects varies significantly starting from very small errors in the contours ending with several cm^3 of missing material or large cavities on the surfaces. From the perspective of the vision system there are two main types of defects; contour and surface defects. The reason for this type of classification is that the image processing methods for detecting these two types of pores are completely different with the latter ones being significantly easier to detect.

Surface defects

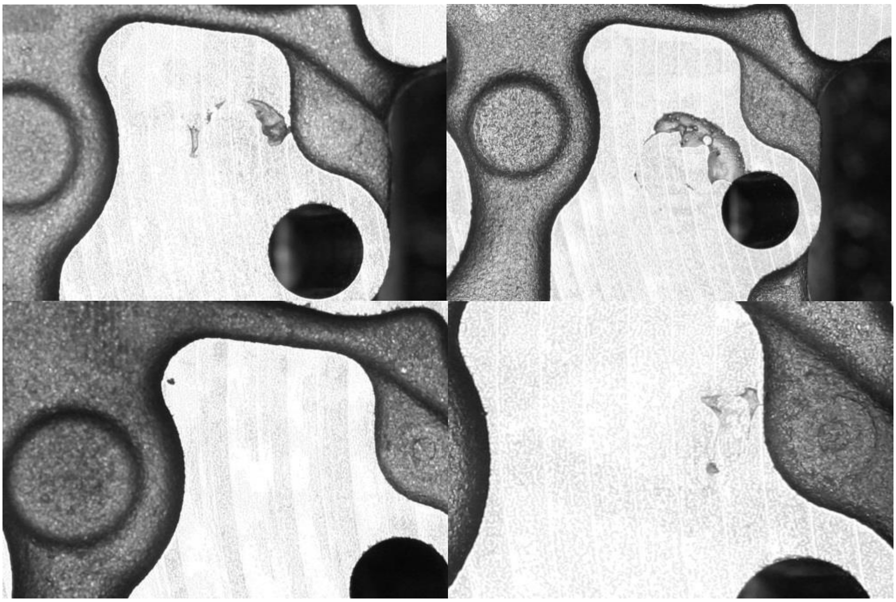

Figure 3 [a,b;c,d] Contour defects with different size and contrast

As can be seen from the images the defects can be virtually any size and shape. While in some areas are more prone for defects, pores can be situated almost anywhere. Therefore, the vision system must be able to check all surfaces of the part and cannot be focused on one specific area. Additionally, the vision system must be equally good at finding large defects as well as small ones which means different strategies might be necessary.

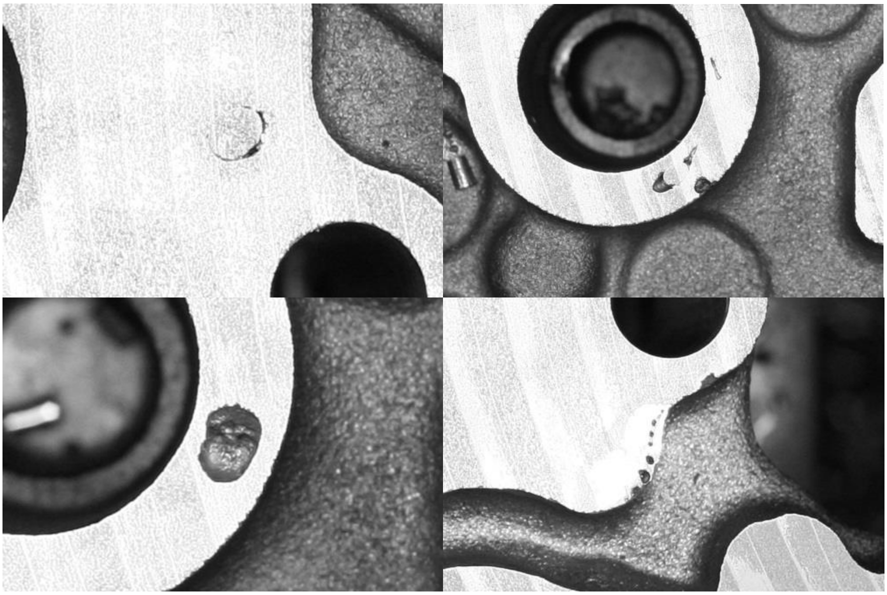

Figure 4 [a,b;c,d] Images illustrating possible defects

Observations have shown that the contrast of the pores varies and it can be generalized up to some extent that the shallower the defect the less contrast it has but this is not always the case. The reason is that surface roughness and profile of the pores varies and thus each pore reflects light differently. If a pore has a rectangular profile - it goes sharply down to its maximum depth and is almost flat in the bottom – and when it also has smooth surface it will have very little contrast and can be hard to detect despite its significant depth. To be able to detect these kind of pores using a single 2D image the lighting setup must be very well designed and possible multiple images with different light setups should be captured. A system that can measure the depth of pores has a significant advantage in this case.

The surface defects can be very close to the contours of the part and can sometimes even be connected to them. In such cases the algorithm might detect them as a part of a contour and the surface defects will essentially become deviations in contour (Figure 3 a,b). This doesn't change much for a human but for an image processing algorithm this might make the pore much harder to detect. Moreover, depending on location a pore can occasionally 'bridge together' two contours which means there would be one large contour instead of two separate ones as normally. The algorithm must be smart enough to recognize and handle such cases without crashing or accidentally approving the part.

The large variety of possible defects and locations indicates that the algorithm must be thoroughly tested before a specific pore detection method can be approved or rejected.

Contour defects

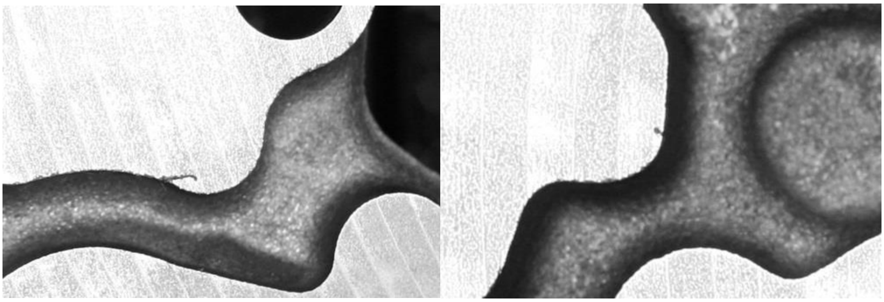

Figure 5 Narrow scratch-like defects

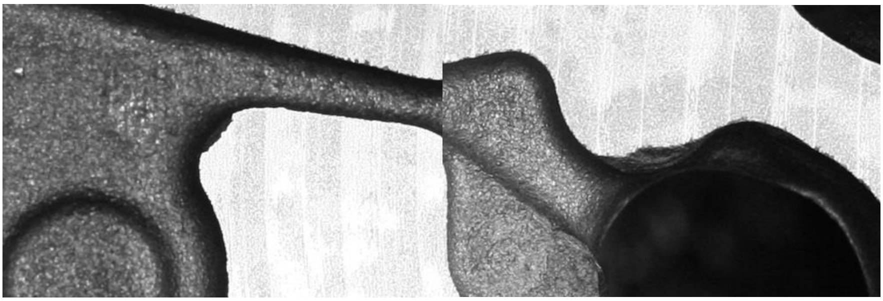

Figure 6 Few millimeters of missing material

As with the surface defects, contour errors come in all shapes and sizes. The defects can be couple of millimeters of missing material making them shallow and often hard to detect visually or they can be sharp ‘bites’ or deep scratches which are visually much easier to notice.

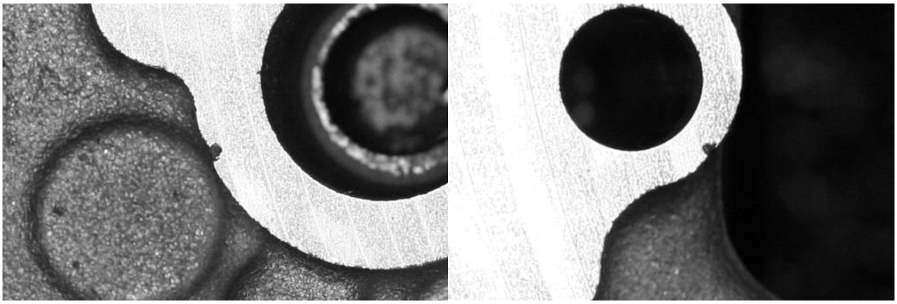

Figure 7 Typical contour defects most commonly present

Important thing to notice here is that sometimes an edge of a part can be slightly chipped and thus have a fairly rough appearance. The algorithm should be able to separate edge roughness from real defects which might be difficult depending on how small defects the system must be able to find. A possible way of differentiation could be that the chips are normally light gray and are usually less than half a millimeter in size.

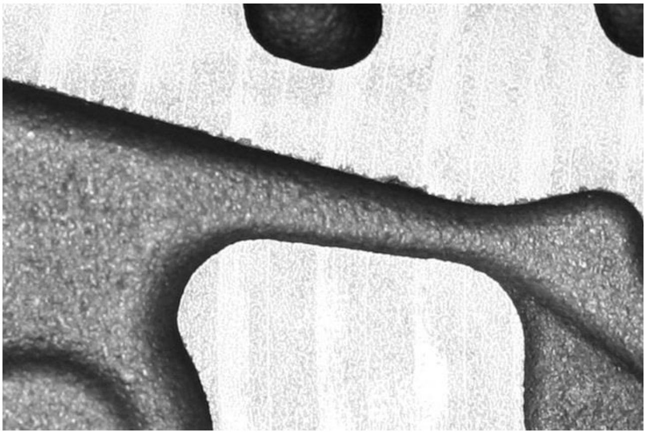

Figure 8 Rough edges that should not to be confused with defects

1.4. Shape Variations

Since this is a cast component it has natural variations in shape and size which are considered normal and do not affect the performance of the part. Therefore the vision system must not consider these deviations as defects and it must be able to handle local variations of contours up to a rather significant extent.

The part has a distinctive section that can sometimes be connected or partially connected to the large surface while on others it might be completely separated in case it appears as a fourth surface. Therefore, the algorithm must be able to adapt to the part or strictly separate the two surfaces so all parts could be checked the same way.

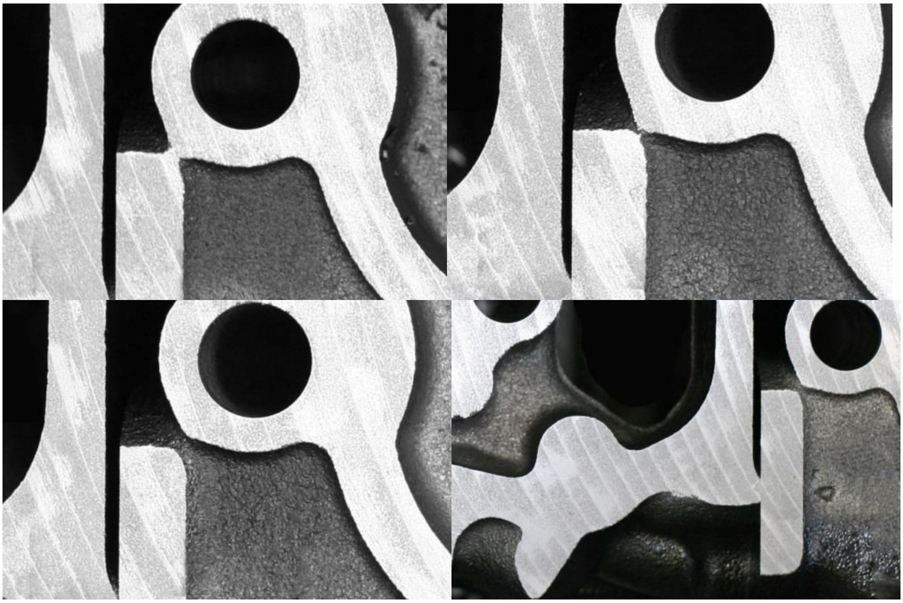

Figure 9 [a,b;c,d] Variations in contour (a,b,c). Shape variations causing separation problems between consecutive parts (d)

Similarly, shape variations can cause an interesting phenomenon where the contours of two consecutive parts on the conveyor are virtually inseparable from one another (Figure 9 d) making it difficult to define where one part ends and the next one starts. In such a case it can happen that a part of the next component on the line will be accidentally inspected together with the current one.

This problem can be solved by physical separation of the parts so that each component on the line gets inspected in isolation. This might require a two sets of flow controlling actuators to be installed making the supporting systems more complicated and expensive. Another solution would be to isolate the parts using software. Depending on exact solution this might cause some loss of detail or inability to check the area where the parts are connected.

Another observation was that there is a specific contour section that has part to part deviations up to 5mm. It is bottom right corner of the part where there is a smooth transition from the surface to the side of the component. Depending on the initial height of the cast body the face milling operation can sometimes remove more material making the milled surface area slightly bigger. Occasionally small spots are left in that area that can appear exactly as surface pores or deviations in the contour which can trick the algorithm to trigger false alarms. Moreover, this type of gradient transition of machined surface to non-machine one can result in a very low contrast area which might cause errors in the extraction of the part's outer contour.

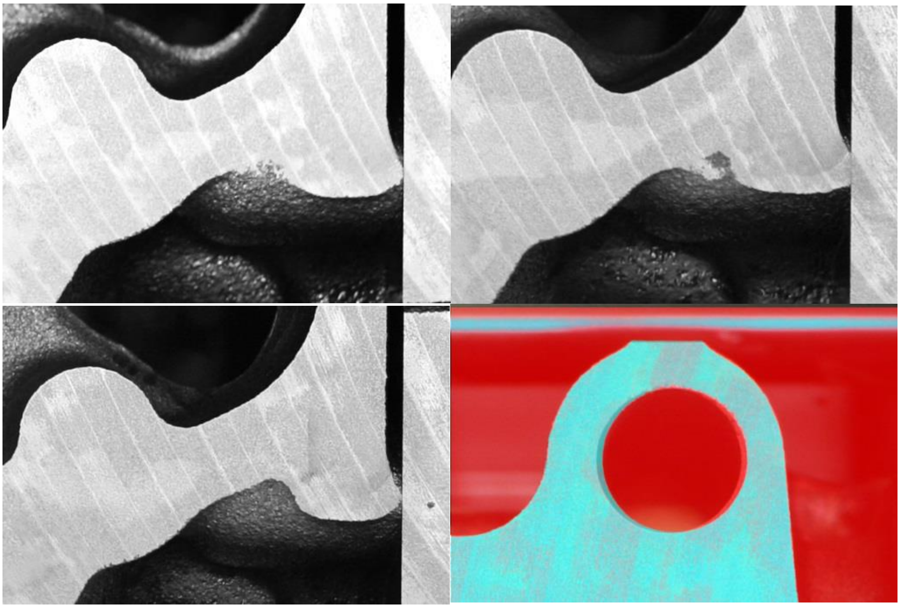

Figure 10 [a,b;c,d] Same corner of three different parts (a,b,c). Hole is misaligned due to fixturing error (d, result of superimposing images in Figure 11 on each other)

Another type of variations can be illustrated with an example where the amount of material around the machined hole of two different parts is not the same as if the hole was slightly misplaced on one of them (Figure 10 d). As the contours of the parts align very well it's likely that this is not a casting error. The parts are machined in a CNC machine with automatic fixtures and this error has likely been caused by work holding in the CNC-mill.

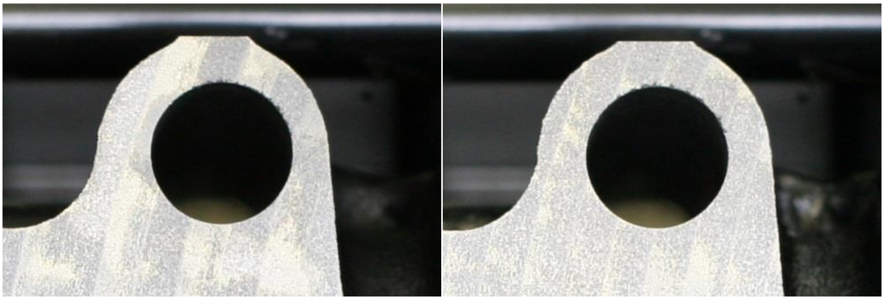

Figure 11 Same area of two different parts illustrating the shift of the center of the hole due to fixturing error. See also Figure 10d.

In most cases the local shape deviations are relatively small and usually have no effect on the performance of the component but they can make the inspection of the part much more difficult for the algorithm and specific safeguards must be coded into it to avoid false alarms. On the other hand, in some areas the part can have extreme deviations from the drawing such as illustrated in Figure 12. Depending on the image processing method chosen to check the contours, this can easily trigger false alarms or the algorithm might miss a real defect because of this large deviation affecting the overall appearance of the contour.

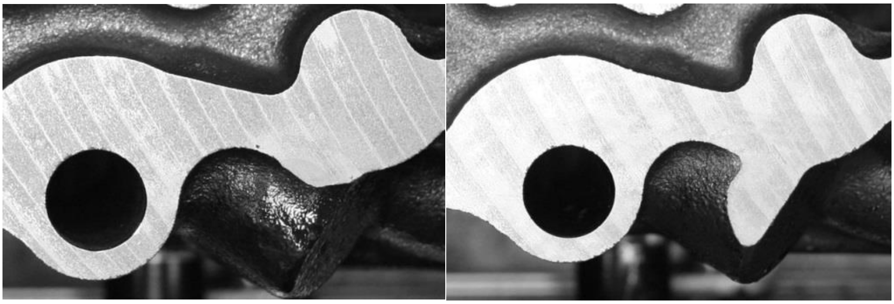

Figure 12 Large deviation from drawing

1.5. Inspection Standard and Other Requirements

The parts are inspected by quality inspectors following a standard which describes where defects are allowed together with their maximum size and depth. A representation of the standard is shown in Figure 13 and it indicates the areas with higher importance. The algorithm must be able to inspect the parts according to the same standard and it must be flexible enough to allow future changes in the standard without significant re-programming.

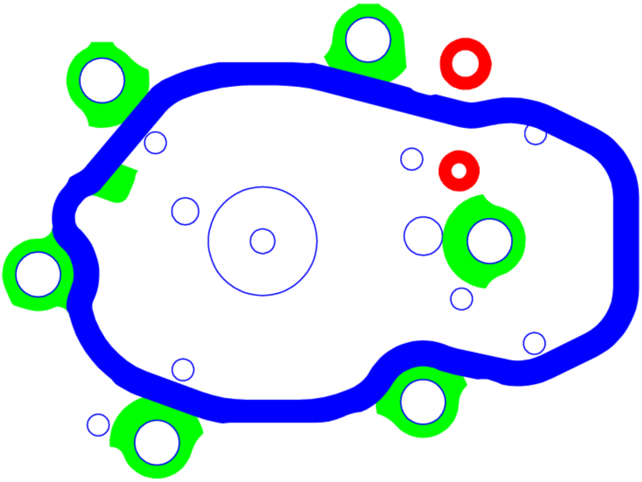

Figure 13 Simplified representation of the critical areas in the inspection standard.

Speed requirements

The vision system must be able to inspect the part without becoming the bottleneck of the line. The components do not move continuously on the conveyor and stop for long enough time to capture an image and process it without needing high-speed image processing or the use of parallel systems. Moreover, the prototype system does not have be able to control the part flow on the conveyor as there is sufficient time available for removing the defective parts manually.

2. The Prototype Hardware

Depending on the specific task and requirements a machine vision can be a very complex one-off system or in other cases smart cameras could be used making the vision system consist of just one major component. The vision system developed in this thesis consists of four main components which include a camera, camera stand, lights and finally a PC. Naturally, the system of the latter type cannot independently control the flow of parts, eject bad parts or visually notify operators of the result. The process flow of the system is described in Appendix 1.

The aim of this thesis was to develop a low cost machine vision system and that limits the hardware possibilities. The general approach when developing a machine vision system is to leave as little work as possible for the image processing algorithm. This means that hardware and the overall setup should support the algorithm as much as possible with appropriate camera/lighting setup and by controlling all the environmental factors such as image background, ambient lighting (amount, angle and color temperature), contamination on part etc. From the image processing point of view an ideal vision inspection would take place in a 100% controlled environment but from the cost, space and sometimes inspection time perspective this type of approach is not the preferred one.

Ideally, the part would be washed and dried before inspection to remove metal and plastic chips as well as traces of cutting fluid. Thereafter, the part would be placed into a closed cabinet where there is no natural light and the background and position of the part would be controlled to remove unnecessary rotations and translations as well as reflections from the background.

To reduce the hardware requirements and cost, the vision system in this thesis leaves much of the environmental factors for the camera and image processing algorithm to handle. This increases the complexity of the algorithm and might decrease system's robustness but lowers the cost on hardware and system set up time. Naturally, more effort is required for programming which also has costs which is why a good balance must always be aimed for.

2.1. Camera

This is obviously the key component of any vision system that relies on images. Over the years many different types of imaging devices have been developed and each of them have their own advantages and disadvantages which makes them more or less suitable for a specific application. There are three main camera types that could be considered for solving the task in this thesis.

Camera types

The first one is a line scan camera which has an imaging sensor that has just one line of pixels and the image is composed by moving the camera or the part under camera and combining all the pixels rows to produce the final image. This way, the image can theoretically be with infinite size in one of the two dimensions Line scan cameras are usually used for high speed conveyors or when the parts is long and capturing an image of the entire part at once is not practical. Line scan cameras allow continuous processing of the image which in some applications gives a significant speed advantage. Although, in this application the line scan camera has no significant advantage as the maximum part dimension is just 250mm and the conveyor does not move continuously. Moreover, line scan cameras create additional complexities such as controlling the speed of motion, the exposure for each line on the image can be different and parts of the image might be missing.

The second type of imaging system would be a structured light system which in principle creates a 3D image of the part. The obvious advantage is that it could measure the depth of the defects to better estimate its severity and whether a part is defective or not. Like line scan cameras, structured light systems require relative movement between the part and the camera which creates a need for using an industrial robot or a linear axis system. Alternatively one could use the movement of the parts on the conveyor but based on the observation the speed of movements is not precision controlled. Another disadvantage is the higher price of such imaging systems compared to other imaging devices.

Third option would be to use a regular array sensor camera which can capture the whole part at once. This type of system is the easiest to implement and it has the lowest price point. The downsides compared with other mentioned options are that it requires the largest working distance, is more prone for distortions and lack of image sharpness and it cannot measure the depth of defects when just a single image is used. Despite the disadvantages this is the type of camera chosen for this system.

Choosing the specific camera

It must be mentioned hereby that the planned inspection system will not require camera capable of high frames per second (FPS) which is often what industrial cameras are designed for. In fact a regular still camera can be used which means there is a vast choice of suitable industrial and consumer cameras to choose from.

Other than usually high fps, industrial cameras have a wide selection of specialty lenses, interfaces for control and data transmission, environmental sealing (for dust and moisture), high vibration tolerance and they can be well customized for a specific vision inspection task to achieve the best possible result. Moreover, they usually have a global electronic shutter which means no moving parts in the camera. The disadvantages of industrial cameras are generally higher price or for a similar price they usually have a lower sensor resolution and the sensor is physically smaller. The latter generally means worse image quality and requires the use of wide angle lenses to avoid large working distances. This problem is further discussed in the lens paragraph.

The advantages of consumer cameras, namely DSLRs, are their lower price, higher resolution and larger sensor size (at the same price point). Additionally, DSLRs are usually excellent at exposure metering and autofocus should it be necessary. The biggest drawback of consumer cameras is their (usually) mechanical shutter and lack of propriety interfaces for industrial applications. Moreover, there are very limited selection of specialty lenses such as Telecentric lenses available for DSLRs. Lastly, consumer cameras tend to be replaced by newer models much more often which might create availability or backwards compatibility problems when the camera needs to be replaced.

For the test system a rather old 8MP Canon 20D DSLR was used with no other specific reason than its suitable resolution, environmental sealing and that it was available for use in this system. The resolution of 8MP means a ratio of approximately $0.0875 \cong 0.1$mm per pixel which is just enough resolution for ensure the system's capability for finding the smallest pores detected by operators.

Automatic triggering of the camera

Various solutions are available for both consumer and industrial cameras. Triggering the latter ones is much easier as they often have a 24V digital input for this and in general they are designed to be remotely triggered.

Things are a little bit more complicated with consumer cameras and it depends on the camera of choice. Remote triggering is easier with DSLRs as they can usually be triggered in at least four ways; IR remote control, wired remote with a physical button/sensor (i.e. laser or proximity sensor), over WIFI with a dedicated device (*TriggerTrap® Ada™, Cactus® LV5™)*, or USB cable/WiFi with PC (using sensor input or software based motion detection [**16**]. Compact system or compact cameras might have the same capabilities but they usually lack dedicated remote triggering ports and sometimes they lack proper PC remote control software and can instead be controlled using smartphones or tablets. Finally, custom software/hardware solutions are always a possibility.

The basic principle of triggering for example with a laser is that the laser beam interrupted by the arriving part which triggers the camera. Then the sensor should be reset. The latter might not be straightforward as the part flow is continuous and there is no gap between the parts. One solution is to place the laser sensor vertically as the part has sections where the beam can shine through it. Although, the placement must be so that the sensor wouldn't get contaminated by dripping cutting fluid and falling chips.

Figure 14 Cactus® LV5™ laser camera trigger

2.2. Lens

Choosing a lens for a machine vision system can be a tricky task as it is a vital component affecting the design and the performance of the entire system. For more complex vision systems such as vision based measuring system with gauge capability, the parameters and quality of the lens has a critical role in the systems capability. Especially, if a special telecentric lens is used.

Although, in this vision system, the lens does not have that high importance but it is still a vital component of the system. The lens together with the camera's imaging sensor size define the working distance, minimum distance from the surface of the part to the camera so that the whole part fits in the image with some extra space for possible translation and rotation of the part. Other key properties of a lens are its sharpness and distortion. It is worth mentioning that since camera lenses require precision engineering and manufacturing, high quality lenses tend to be expensive pieces of hardware and they can quickly drive up the overall cost of the system.

The wider the lens, the shorter the working distance can be. Short working distance in turn allows compact system design which is a desired property. On the other hand, wider lenses must have optically higher quality to avoid heavy distortion and loss of sharpness. While distortions can be reduced or virtually eliminated with software it is still important for the optics to be of very high quality. Another possible drawback is that wide angle lenses capture, in addition to the inspected part, a wide area of background which one might not be able to easily mask or shield. This background can cause low contrast between the part's surfaces and the background which creates additional difficulties for the algorithm to robustly extract the edges.

Based on the factors described above a standard Canon 50mm F/1.8 II lens was chosen for the vision system. It is a very low cost lens which *optically* compares surprisingly well to high end lenses such as Carl Zeiss Planar T 50mm f/1.4 ZE or Canon 50mm f/1.4 USM which cost up to 7 times more. Although, the casing of 50mm f/1.8 is made of basic plastic and it has relatively poor build quality. It also has a smaller maximum aperture than its peers.

A lens is typically sharpest around its center and sharpness decreases when moving from the center towards the edges. The same principle is true for distortions which increase with distance from the lens center [**5**]. The tests done by DxOMark with the chosen lens attached to Canon EOS 20D show that the lens is sharpest when stopped down to f/4 which was considered when setting camera exposure parameters for the vision system. Although, stopping the lens down from f/1.8 to F/4 requires slower shutter speeds which makes the system more prone to vibrations which is discussed later.

The sensor in Canon EOS 20D is of APS-C type which has a crop factor of 1.6. This means that the 50mm lens, when attached to the EOS 20D, has approximately the same field of view as an 80mm lens attached to a camera with a full frame sensor. This increases the working distance to a fairly large 0.7m but on the other hand having a smaller sensor allows to benefit from the so called center sweet-spot of the lens which usually has better optical properties than the edges of the lens as was discussed before.

Figure 15 Comparison of 50mm lens on Canon EOS 20D. Source: www.DxOMark.com

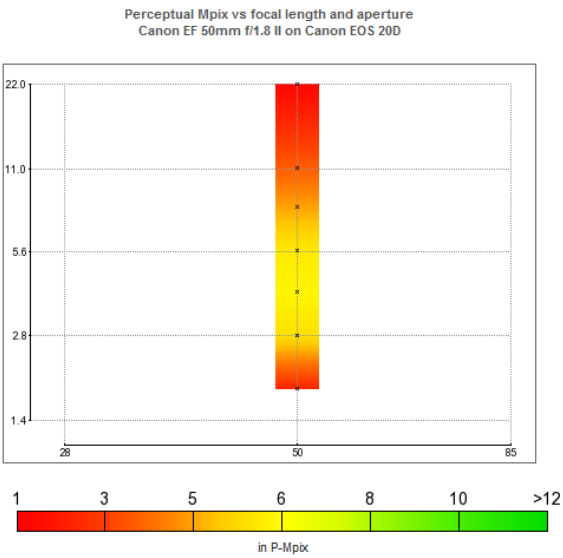

Figure 16 Sharpness vs Aperture. Source: www.DxOMark.com

2.3. Lighting Setup

It is said that finding the correct way of illumination is the key for a successful machine vision system which is why careful considerations are required in this area [**5**]. Thoughtful illumination can make a night and day difference in terms of the complexity of image processing and system's robustness.

There are many different possible illumination schemes including the selection of suitable spectrum, type of light source (type, shape, intensity), the technology (LED, halogen, xenon, fluorescent etc.), the angle and dimensions of the light source and one can even attempt to match the imaging sensor sensitivity with the spectrum of illumination source. Therefore, there are very many aspects to consider and testing of different setups might be necessary.

The lighting setup of a vision system should aim to achieve these three main goals below [**17**]

1. Maximize contrast of features of interest
2. Reduce contrast in other areas
3. Increase system's robustness

To illustrate the effect of different lighting methods, here is a selection of examples images from other vision system applications.

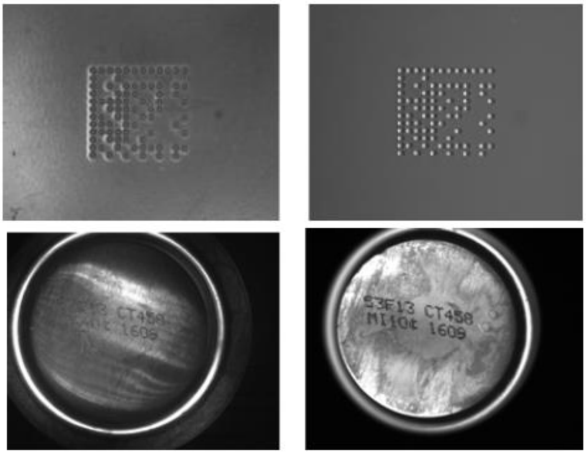

Figure 17 Same objects under different light setups. Source: [18]

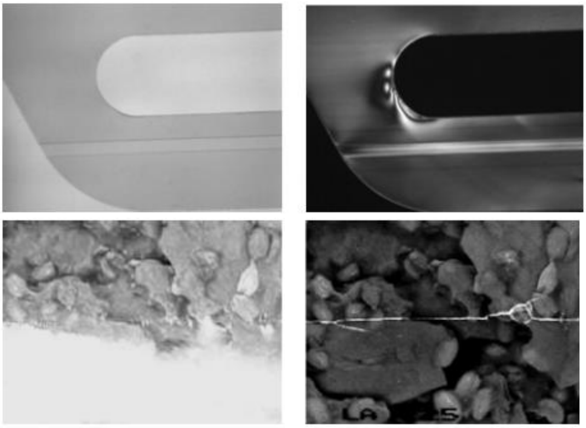

Figure 18 Same objects under different light setups. Source: [18]

Due to lack of availability of different light sources, filters and the general decision of not blocking ambient light (for reasons of cost, space and simplicity) there was limited investigation and testing of different setups done during the development of this system. Thus, it cannot be claimed that the final chosen illumination method is the best for this case.

The principle of the illumination scheme chosen for this vision system is to use bright field light lighting (Figure 19) with a light source that evenly covers the whole part with near-perpendicular light in relation to the parts surface. When the perpendicular light ray falls on the flat surface on the part it reflects right back to the camera but the light that hits a defect on the surface or a non-perpendicular surface gets reflected away from the camera. The result is that the flat surface appears bright, almost white, while the defects appear gray, dark -gray or black depending on the type of defect. This difference in intensity levels (contrast) means that defects can be detected using thresholding or edge detection.

Theoretically a better option would be to use on-axis lighting using a half-mirror (Figure 20) as this way the light would be truly perpendicular and would thus further increase the contrast of defects. Due to the size of the part this lighting setup would be much more expensive and the idea was therefore discarded. Another interesting option would be to use dark field lightning (Figure 21) which would illuminate the defects while the rest of the surface is dark. Though, this method would require filtering or shielding of the ambient light. Moreover, it is only suitable for finding surface defects not deviations in contours, but this could be solved by capturing two images with different lighting setups.

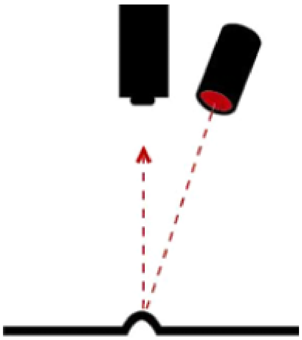

Figure 19 Bright field lighting with single light source. Source: [19]

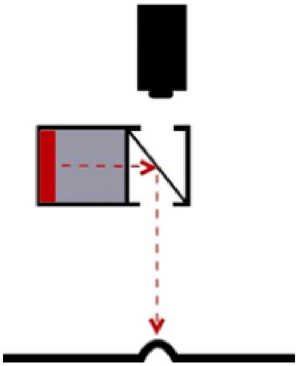

Figure 20 On-axis lighting using half-mirror. Source: [19]

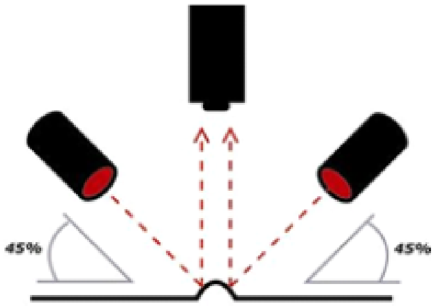

Figure 21 Dark field lighting. Source: [19]

Some combinations of light sources were investigated for the system starting from combination of spotlights, array lights and ring lights ending with various camera flashes. Though, only the ring light and a standard flash were tested. The latter ones are an interesting option as their intensities could be easily controlled by the camera (depending on which camera is used) and theoretically a powerful flash could fire a high intensity light pulse (strobing) that would wash out all ambient light. A standard Canon 430EX external flash mounted on a DSLR was tested and in principle the results were fairly good but the illumination was uneven with the furthermost areas having poor illumination while the surface directly below flash was overexposed. Also, the light was not perpendicular to the entire surface of the part thus the contrast of the defects in some areas could have been better.

Another disadvantage of using a flash is that the lighting is not continuous and can thus be disturbing for the nearby operators given the relatively high working distance of 0.7m with the chosen 50mm lens. Furthermore, most affordable flashes meant for using on a DSLR cannot be fully battery independent. While the flash bulb itself can be AC-powered the control circuitry of flash would still require batteries thus leaving the need to occasionally change batteries in the flash which is inconvenient and not a particularly robust solution.

Therefore, an AC powered LED ring light (F&V HRD-300) was chosen as the single light source for this vision system. The diameter of the ring light is not ideal as it is slightly smaller than the maximum dimension of the part but using a larger ring light would have resulted in substantially higher cost. There are high quality ring lights for video shooting and special purpose lights for vision systems available that can even be controlled over USB or Ethernet but the price point of those is significantly higher.

Ambient light

An important factor in illumination is the ambient light which changes a lot in the factory as there are a lot of windows. Ambient light illuminates the part from all possible directions with uncontrolled intensity. Therefore, it reduces the contrast of the defects, making them harder to detect. On the other hand, this can be seen as an advantage as shallow defects appear lighter than deeper, more severe defects, allowing rough classification of the pores by depth. Mainly this can be used to separate cases when an edge is just slightly chipped from cases when a contour contains a defect.

The level, angle and color temperature of the light depend on the time of day and weather, thus being a subject of change. There is a big difference in the illumination of the factory on a cloudy midday and during a clear sunrise or sunset.

Therefore, if no light shielding (or other methods such as filtering or strobing) is used in the vision system, the camera must be able to accurately meter the exposure in all conditions and preferably the LED illumination should be adjustable or self-adjusting. The LED ring light used in the system must be adjusted manually but during the test period of the system there wasn’t much need for it as camera exposure metering was mostly correct.

If further testing shows the light can in certain conditions be a largely disturbing factor for the vision system then light shielding with curtains could be used.

Reflections

Another illumination related factor is various reflections from the part itself and the conveyor components. In fact, the reflections from the rollers of conveyors were in some occasions so intense and at such a bad location that the image segmentation couldn’t successfully find the contours of the cylinder head. The problem was solved by adjusting the parameters in the algorithm and by masking some parts of the rollers which are not in contact with the product with a simple black masking tape.

Furthermore, the some bright surfaces of the conveyor reflected some of the light from the LED ring light onto the sides of the part again creating occasional difficulties in the edge detection. The solution was again to use the masking tape in some specific locations. Although, the problem could have been solved with software, the masking tape was a very simple and robust solution.

Diffuser

The direct light from the ring light is by default undiffused and thus it can be very harsh. This tends to create images with very high contrast and strong reflections. Therefore, a self-made diffuser was attached for the ring light which softens the emitted lite to reduce undesired reflections and areas with high contrast. The result is a much more even illumination over the entire surface of the cylinder head. Diffusing the light also creates an effect where shallow defects are lighter than steep ones. The difference exits even with undiffused light but the diffuser increases the difference and thus makes it possible for the algorithm to use this information to separate shallow pores from deep ones.

2.4. Camera Stand

When the camera, lens and lighting were selected the requirements for the camera stand can be specified. The required working distance is around 700mm and the weight of the camera, lens and ring light together is less than 2kg. The stand must allow adjustments of the position of the camera in all six degrees of freedom so that the camera could be place directly above the part on the conveyor.

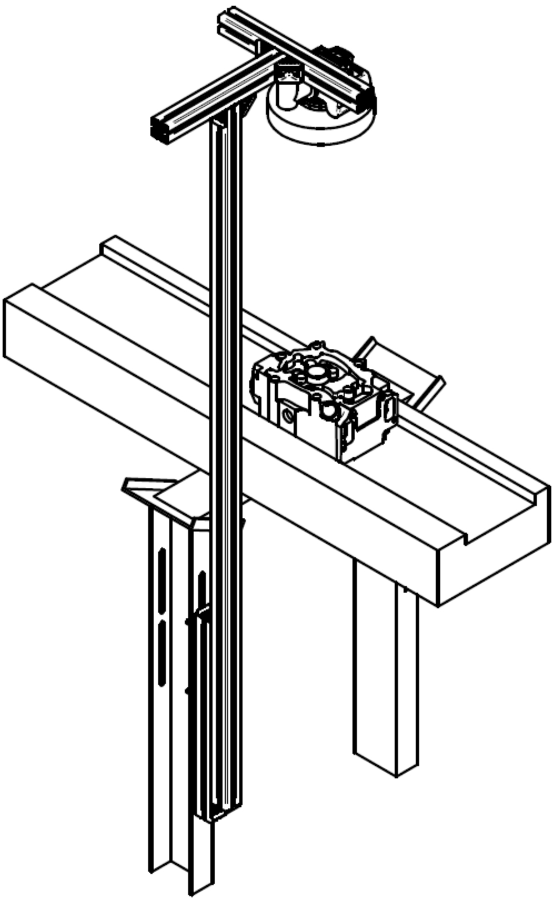

Figure 22 Illustrative drawing of the prototype vision system without the PC, monitor and brush system.

The camera stand is built using 45mm aluminum profiles and a three-way tripod head for possible rotations of the camera around all three axes. The stand is bolted to one of the conveyor legs and 45degree gussets allow easy translational adjustments while keeping the system rigid.

After first testing it was clear that the design was flawed as the system was prone to oscillation. The reason is that the parts on conveyor have a significant moment of inertia and when the stoppers that control the flow on the line open or close there is a strong mechanical shock which makes the camera and the stand to oscillate. This causes blurry images and it tends to shake the lens so its focus ring loses its position causing it to gradually lose focus over a certain period of usage.

Though, the problem was easily solved by making the top part of the stand more rigid by attaching it to a nearby post. Naturally, the design would be much better if the camera stand was not fastened to the conveyor entirely and would stand on its own supporting leg.

2.5. PC

The computing power of this system comes from a laptop with a 27inch monitor coupled with it. The laptop has a second generation Intel® Core™ i5-2410M CPU at 2.3GHz, integrated Intel® graphics and 4GB of RAM. It is a fairly mediocre processor by today's standards but it is more than capable for running this specific algorithm (without thorough optimization) below the time limit set by the production speed.

The camera is remote controlled from the PC by using Canon's own EOS Utility™ software. Although, more advanced remote controlling software is available [**16**] the EOS Utility software comes free with the camera and since the camera does not have live view capability the more advanced software does not justify itself in this specific case. The used EOS Utility software allows triggering the camera and changing all the main camera parameters such as aperture, shutter speed, ISO, exposure compensation, metering mode etc.

An unnecessary difficulty was created by the fact that EOS 20D can be remote controlled using EOS Utility only on Windows XP. For this reason a Virtual Machine was created that handles the PC to camera communication and images are then sent over to Windows 7 where MATLAB automatically reads the image file and processes it. MATLAB® itself has Image Acquisition Toolbox™ for direct communication with many industrial and web cameras but unfortunately it does not support communication with most DSLRs including this one.

2.6. Brush System

As was described in the pre-study the cylinder heads can be highly contaminated with cutting fluid and metal or plastic chips. To increase the reliability of the system and reduce possible false alarms the condition of the parts needs to be improved before inspection.

For this purpose, a brush system was developed that uses a combination of three synthetic brushes fixed at a slight angle to ensure effective removal of chips without damaging the surface of the parts. The height of the brush system is adjustable for finding the optimum pressure required to remove all contamination while not causing too much friction.

Tests have shown that the brush system works well with little or no contamination left. One observed problem is that a single piece on the conveyor might get stuck under the brushes. Although, the part is freed when the next one arrives and pushes it through. This is not a desired outcome and it can be a problem if there is a switch of product type on the line. Fortunately, the brush is situated right near the workplace of an operator so it hasn't been a serious issue. Potential solution for this problem is to separate the three brushes from each other so that less friction occurs simultaneously. Furthermore, single parts are known to occasionally get stuck on other parts of the conveyor as well so this issue is known for the operators and attention is paid whenever production is stopped or product type is changed.

Figure 23 Uncovered brush system

2.7. System Cost

The total cost of the vision system depends on many factors starting from the choice of software, quality of the hardware and the amount of support systems required. The latter consists of the automation system that controls the flow on conveyor, system for ejecting defective parts, getting necessary certifications etc. The conveyor has existing pneumatic actuators nearby which could be repositioned and programmed to work with the vision system. Hence it is difficult to estimate these costs and therefore all those support systems have been excluded from hereby price estimation.

The biggest single source of cost is the software. The algorithm in the test system was implemented in MATLAB® with Image Processing Toolbox™ v 8.4 which consists of various functions used for digital image processing. MATLAB® was chosen as it was readily available for use and it is a great development tool with great documentation and support community. Although, due to high price it is not a suitable software option for a low cost system which is why free MATLAB® alternatives or an open source image processing library, OpenCV, is to be used for the final algorithm. Based on preliminary observation OpenCV has similar or occasionally superior functionality and it is free. Additional benefit of OpenCV is that it is based on native programming languages such as C++, Java or Python and can thus have a lower algorithm execution time than MATLAB®.

Table 1Approximate budget of the prototype vision system. All prices are estimated.

IPC	10000
Camera	7500
Monitor	3000
Camera stand	2000
LED light	1500
Tripod head	1500
Lens	900
Accessories	750
Brush system	150
MATLAB® with IPT™	35000*
Total	**27300**
Including MATLAB®	62300*

All prices listed here are approximate prices based on preliminary observations. It is worth mentioning that the prototype system is based on an existing laptop PC, 27” monitor and a camera. Therefore, the costs shown in the budget table do not reflect the funds actually spent.

Despite being an industrial environment it does not set high requirements for the hardware but as the vision system must be able to operate 24/7 it cannot be based on low cost consumer hardware and components must be carefully chosen.

The PC should be and industrial PC with an SSD hard drive for high reliability but it does not necessarily have to be a fanless system as the environment is not particularly demanding. The PC should be able to communicate with the automation system that controls the ejection mechanism and flow on conveyor. This could be implemented with a dedicated 24 digital I/O or over local area network using, for example, TCP socket communication. Suitable PCs are for example Intel® Core™ processor based passive NUCs such as Logic Supply ML-320 or Habey BIS-6922 which both cost around 1100$ depending on configuration [**20**,**21**]. Alternatively, a lower cost non-passive Intel D54250WYKH Haswell NUC Kit could be used which in suitable Intel® Core™ i5 configuration with 8 GB RAM and an SSD drive costs around 700$ [**22**].

Figure 24 [a,b,c] Habey BIS-6922 passive industrial PC (a,b). Intel D54259 NUC Kit (c)

Figure 25 [a,b,c] Baumer LX series high resolution industrial camera (a). Nikon D7100 DSLR (b). Nikon 1 J2 compact system camera (c).

The camera can be a consumer DSLR (i.e. Nikon D7100 or Canon 70D) or a compact system camera preferably with environmental sealing and preferably electronic shutter (i.e. Nikon 1 series, Panasonic G series) that would eliminate the problem of having to replace the shutter mechanism or the entire camera approximately once a year or two years depending on production level. Alternatively, a dedicated machine vision camera could be used. Depending on final choice of camera the lens cost can be significantly higher as a lens with matching price and performance are not available for all lens mounts. Fortunately, various lens adapters can be used to mount the existing lens to different cameras with no real downsides as the system uses manual focus and a fixed aperture value.

3. The Algorithm

As was mentioned earlier, the image processing algorithm for the system was programmed in MATLAB® R2013a with Image Processing Toolbox™ 8.4. The code is built on a mixture of native and custom functions developed specifically for this system. Some of the custom functions are or are based on third party code from [**23**], Mathworks File Exchange (under BSD license) and by Peter L. Ekberg. The algorithm has approximately 4000-5000 lines of code (see Appendix 1 for a sample) but it has several sections that are rather similar to each other and that could be further developed into custom functions which would make the code more compact and easier to follow. As this is a beta version of the algorithm this has not yet been done for easier and faster debugging. The code is a property of Scania and cannot be included in the thesis although most of the important methods are discussed in the thesis.

The algorithm is divided into four main parts.

1) Image segmentation, masking and filtering
2) Verification of contours and finding contour defects
3) Finding surface defects
4) Analysis and presentation of results

It is ran as an infinite loop that processes an image, displays and saves the results and then waits for the next image. All main parts are further divided into sections and it can skip each section mid-way if no defects are found. It can also happen that the first part of the algorithm fails and in this case it displays an error message with a possible reason, saves the results and waits for the next image. In the following paragraphs the most important sections of the algorithm are described.

3.1. Image Segmentation and Noise Removal

This is the first part and can be called the most critical one in the algorithm. Here the contours of the part and the possible pores are found and extracted from the input image. The algorithm builds upon solid detection of edges and removal of all irrelevant objects in the image. If this procedure fails, the rest of the algorithm cannot be executed successfully and the inspection has failed at an early stage.

Edge detection

Image segmentation or edge detection is one of the most basic image processing techniques and a great deal of research effort has been dedicated to this starting from the early days of image processing up till today. MATLAB® IPT has several native functions for edge detection, all of which were tested for this algorithm. The most advanced edge detector or operator currently in this toolbox (v8.4) is the *Canny operator* first proposed by Canny in 1986. It is a four step process starting with Gaussian filtering for noise reduction. Then local gradients and edge directions are calculated at each image point and a point is considered to be an edge point if its strength is locally maximum in the gradient direction. Thereafter *non-maximal suppression* is done and remaining edge pixels are *thresholded* with two different thresholds. Points which exceed the upper threshold are strong edge points and points between the two threshold levels are weak edge points. The last step is to connect the weak edge points to strong ones using 8-connectivity. [**23**]

The result of using the *Canny operator* is and image with thin contour lines which indicate pixels in the image where there is strong contrast between neighbor pixels. Canny detector was the method of choice for this algorithm due to its good accuracy and high robustness. There are a lot of false edges detected together with the real contours of the inspected parts but most importantly the contour of the part is mostly continuous with only occasional discontinuities in the areas of low contrast.

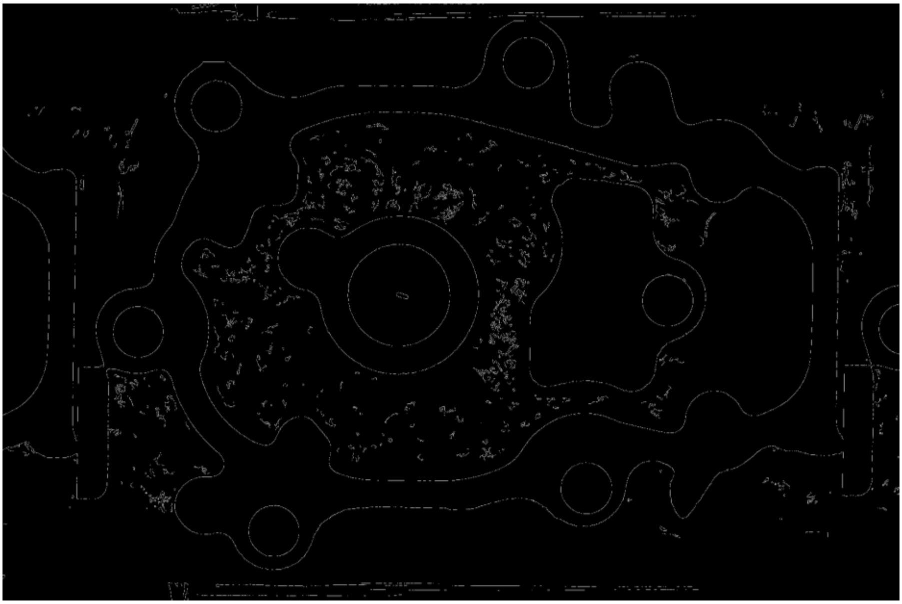

Figure 26 All edges detected by Canny edge operator. Note how much additional edges are found in addition to the contours of the part.

An important thing here is to set the two thresholds right. When set too low, then even objects with very weak contrast are detected, thus the resulting image has a lot of irrelevant contours that could be connected to the main contours of the part, thus making it very difficult to further process the image. On the other hand, when thresholds are set too high then very little noise will be detected but the contours of the part might have significant gaps in it.

There are newer alternative edge detection algorithms available that have sub-pixel accuracy. One of these methods is Sobel-Zernike moments operators which is a two-stage algorithm first finding candidate edges with Sobel operator and then uses Zernike moments for finding final contours [**9**]. Though, no sub-pixel edge detection algorithms were tested for this system and their advantages and reliability in this application are yet to be studied.

Furthermore, a very simple yet sometimes effective technique for image segmentation is thresholding where the pixels of the grayscale image are set to either 0 or 1 based on their intensity level. If the intensity is above certain threshold it is set to one and zero otherwise. Alternatively, more advanced adaptive local thresholding can be used that checks the intensities of neighboring pixels and then makes a decision if the pixels should be set to 1 or 0 in the output image. Main advantage of thresholding is its speed.

Morphological processing

Since there might be discontinuities in the contour, especially in the lower right corner where non-machined surface gradually becomes machined surface, the edges detected in the previous operation must be further processed. There are several methods for filling the gaps. One option is to use *morphological processing*, namely *dilation* which is a process where the image is convolved with a structuring element (i.e. 3x3 square). The element is incrementally moved over the entire image so that the center of the structuring element visits each pixel. Then if at that location any of the pixels in the 3x3 square overlap with an image pixel that is 1 (not 0), then that pixel currently at the center of the square is set to 1 in the output image.

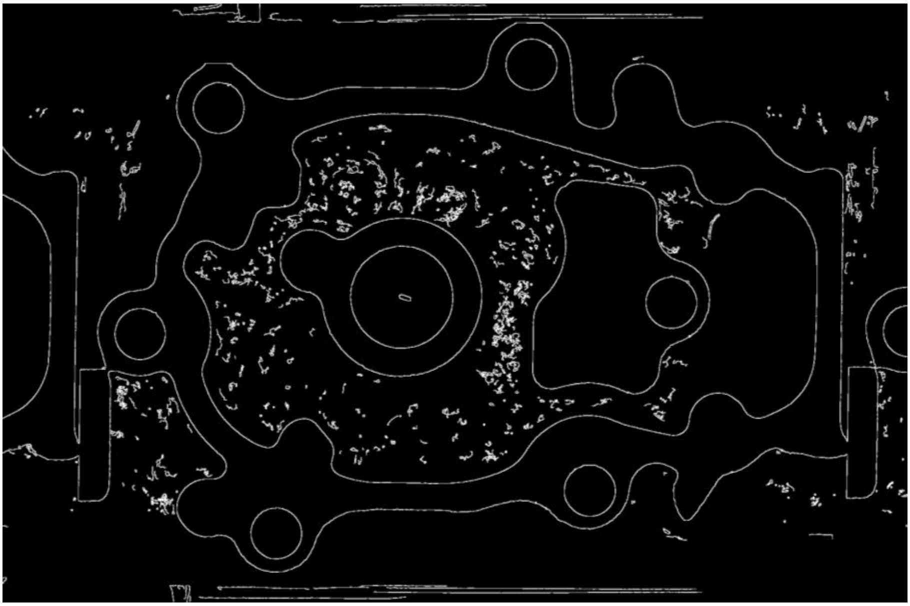

Figure 27 Edges after dilation with 3x3 square structuring element.

As a result all objects in the image get thicker and if there is a one or two pixel length gap in the contour, those gap pixels will be set to 1 during dilation. Dilation will not only fill gaps it will also make all contours lines wider which depending on situation can have a positive or negative effect. Positive, as it connects gaps in all directions but negative as fine detail is lost.

Naturally, the structuring element can be any kind of element with any size. Although, computationally, square structuring elements ought to be the faster and due to risk of losing detail, the structuring element should be as small as possible.

Noise removal

Once the contours have been closed, the removal of noise and irrelevant objects begins. There are several ways of doing this with varying degree of robustness. If little noise is present and it is of high certainty that all edges of interest have been properly found, then in an ideal case one could simply select out contours based on their length or pixel count. This could work well for large contours namely the outer perimeter of the part but it would be robust only when the image is taken with a controlled background and if there is no liquid or contamination in the center pool. The latter can be detected as objects with very long contours and there is high image-to-image variation due to them.

Therefore other methods must be used such as masking or further morphological processing to make the noise removal as robust as possible. As it was mentioned before without successful image segmentation the rest of the algorithm has no real value as it will most likely result in a false negative or positive decision.

Masking

First all the closed contours in the image are filled, meaning that all pixels inside a closed contour will be set to 1. This will result in an image with filled objects (blobs) instead of the initial contours (Figure 28a). Then *morphological opening* is done on the image.

Morphological opening is *erosion* followed by *dilation*. The former is a somewhat opposite process to dilation where pixels are set to 0 if the structuring element at that pixel location is not entirely inside an object (i.e. blob or contour line). As a result all objects get smaller and objects connected by a thin section or a line get separated (Figure 28b) after which they can be removed or handled separately.

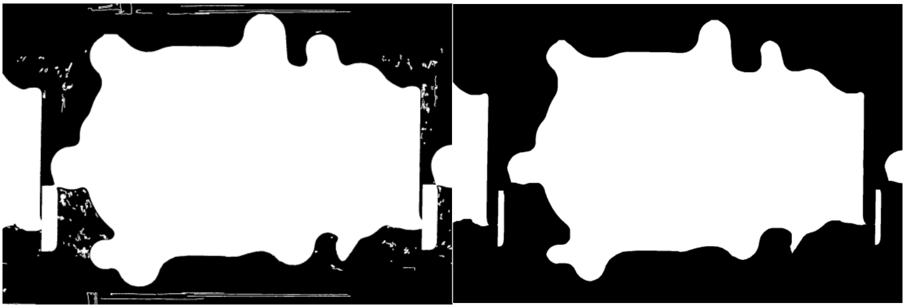

Figure 28 [a,b;c,d;e,f] The sequence of masking. Filling all closed contours (a). Morphological erosion (b). All small components remove (c). Morphological dilation (d). Illustration of the masked area (green) (e). Final result (f).

The actual masking is done by multiplying the dilated contours image with this new binary mask. The result is that all pixels which are not under the mask are set to zero and thus the masked image (Figure 28f) has just the contours of inspected part and its internal objects.

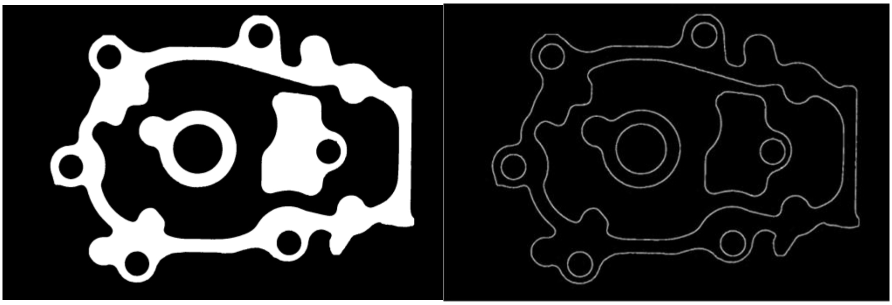

Figure 29 [a,b] The mask (a) and contours after masking (b).

3.2. Contour Defects

When all contours have been extracted from the original image, the algorithm proceeds with analysis of the contours. This is done in two major steps described below.

1) Verification of the presence of all main contours
2) Checking all contours for defects

Step one consists of creating a database of all found objects in the image and finding their perimeter. The part has five distinct main contours and contours of the 6 machined holes which are identical. Therefore, every inspected part should have at least 11 objects in the image at this stage. This info can be used to verify if the algorithm has been able to extract all contours.

Problems can arise if some of the contours become connected to each other making the two contours appear as one. The algorithm has some built in measures to recognize such situations. Though, it must be said that if this scenario happens then the part is 100% defective given that the error was not caused by failing to extract the contours of the part.

Often this type of scenario with connected contours involves one of the six drilled holes (Figure 30b). Another issue involving the six drilled holes is that occasionally some of the surface defects are just as large as or larger than the six holes. So measures are required to reliably detect the contours of the drilled holes and differentiate them from large defects.

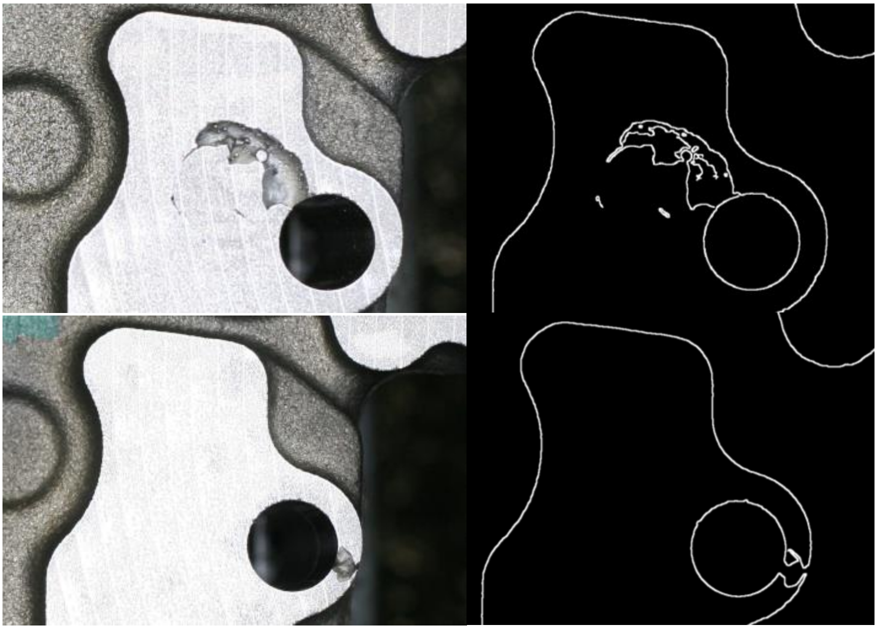

Figure 30 [a,b;c,d] Various defects affecting the appearance and properties of the contours of the drilled holes. The hole can have a significantly larger perimeter if a surface defect is connected to it (a,b). The hole can become connected to another contour (c,d)

Inspecting the contours for defects

Robust inspection of the part's contours was certainly one of the most difficult challenges in the development of this algorithm. Several methods were investigated and tested with varying degrees of success. The difficulties were caused by the complex shape of the contours and/or the large variety of different defects that the parts could have. Often a method was relatively good at finding large defects and missing the small ones or vice versa. Moreover, due to the complex shape a method can have varying degree of accuracy in different contour sections. Furthermore, the task was even more complex due to high part-to-part or part-to-drawing local contour deviations.

Centroidal signature

One group of methods with high potential is shape signatures, namely the centroidal profile. The basic principle of the method is the following. First, the center point of a contour is found either by averaging the pixel coordinates or using the center obtained from a template image. Then, the distance from the center to the contour is measured at a specified angle step from range [0, 2π] or [0,- π;0, π]. The result is a 1D representation of a 2D curve which makes further analysis of the contour faster and simpler.

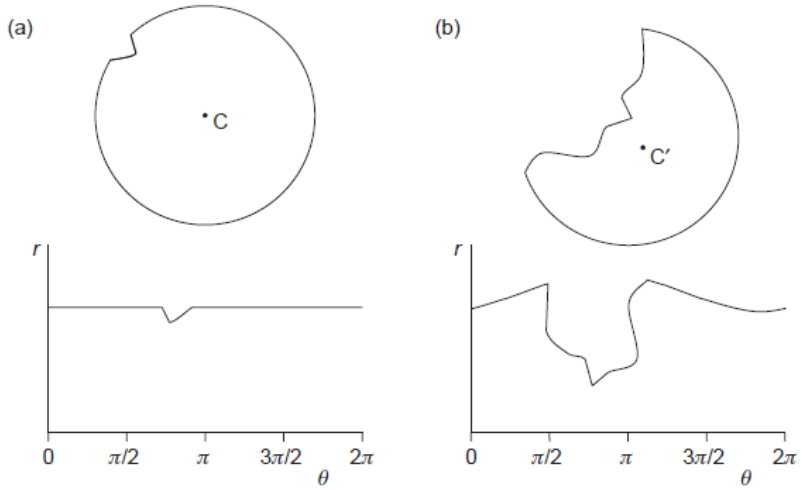

Figure 31 A defective circular part and its centroidal profile. Same part with a large occlusion and its centroidal profile with shifted center point. Source: [8]

The shape signature method is a known basic shape descriptor that is used for template matching and finding shape deviations. Despite its simplicity, robustness and many other advantages, it has a few known flaws that need to be solved before it can be used for this task. The two major ones affecting the current application are the following.

1) The signature changes significantly if the center is shifted due to a large occlusion
2) The signature can become multi-valued making it a 2D plot [**8**]

The first mentioned major problem with the shifting center point mainly affects smaller contours. In this case the problematic contours are the two islands in the center and the six drilled holes. The second major problem is a more difficult one to solve. When the centroidal signature is based on polar coordinates, then the signature will have two or more distances for the same angle value making the signature a 2D plot. One solution is to use the minimum distance value for each angle that has multiple values. This works fairly well for detecting large defects but the downside is that this makes the signature blind for sections that are behind the defect. In other words, if there is another defect behind the first one, it will not be seen. In most cases this won't be a problem as the large defect is sufficient to call the part defective but it might not always be the case.

The issue is more severe if the shape of the contour is self-intersecting such as the outer contour and the second largest contour. In this case the centroidal distance function based on polar coordinates is always a 2D function making this method non-suitable as a single method [**23**]. Thus the signatures method must be adapted for this application to overcome its known shortcomings and preferably used together with other methods.

Finding the defect locations

The principle of finding defects with this method is to find points where the distance from the center to the contour is larger (or smaller, depending on the contour) than it should be according to the signature of an ideal part. Although, this method generally gives robust and fairly accurate defect size estimation it has issues caused by the complex shape and local variations in contours but they can be solved with varying degree of success.

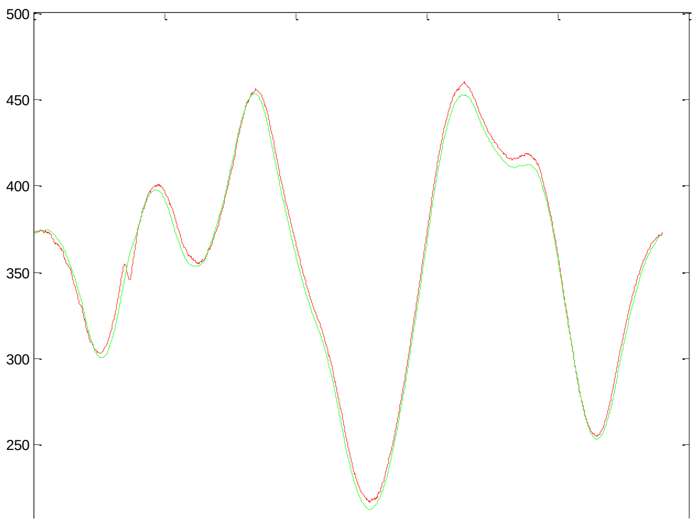

Figure 32 Centroidal signature of one of the contours. Red line = current component, Green line = ideal contour.

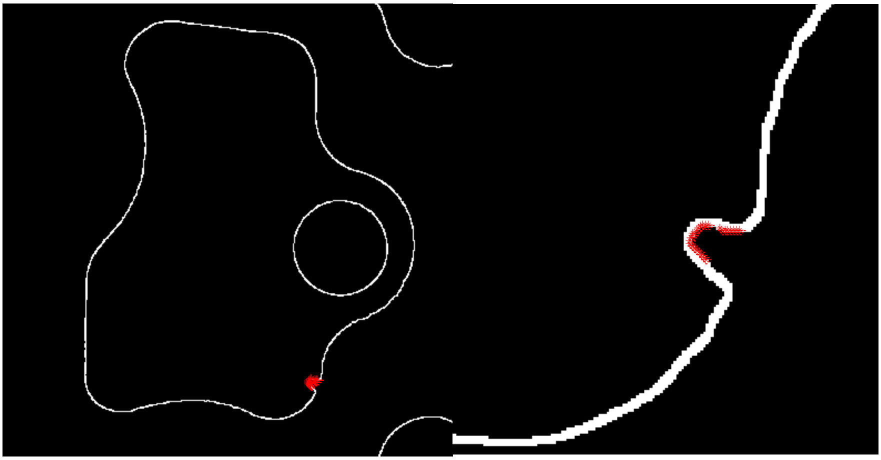

Figure 33 The found defect points from the centroidal signature plotted on the contours of the part

Alternative methods

Another solution would be to use a different type of shape signature called the *[s,ψ] plot* which does not use a center point and instead creates the signature by calculating the angle between the line tangent and a reference line at each contour point [**24**]. Another possible problem might be measuring the size of the defect with this method. Despite looking promising, the *[s,ψ] plot* was not thoroughly investigated due to time constraints and its advantages and shortcomings are to be further investigated for this application.

3.3. Surface Defects and Presenting the Results

The method for finding defects on surface is fairly straightforward and does not require any special techniques. It relies on a good lightning setup to maximize the contrast of defects and then finding the contours of those defects as accurately as possible. If the above methods have worked fine, the surface defect now appears as an object between the contours of the part.

Though, there are some occasions involved that can make finding the defects harder. Firstly, as was mentioned earlier, the surface defects might sometimes be very near the contour making them contour defects and therefore more difficult to find. In other occasions a defect can be so large that it might be confused with one of the drilled holes if measures to avoid this are not in place. Furthermore, as was mentioned earlier, a defect can in some occasions have very little contrast and thus it might be difficult to obtain its exact size or in the worst case the pore might get undetected. Lastly, the final decision whether the defect actually makes the part defective or not depends on, in addition to its size, where it is located. Thus, the algorithm must be able to accurately compare the part with the valid inspection standard.

Measuring defect size

Methods for solving the issues except the pore size measurement and the comparison with the standard have already been proposed. Several ways exists for estimating the size of defect. One method is to estimate the total area of the defects. This could be done by counting all pixels under the pore and since the area corresponding to each pixel is known ($0.01mm^2$) the total area can easily be calculated. Another option is to measure the length of the defect.

Comparing with the standard

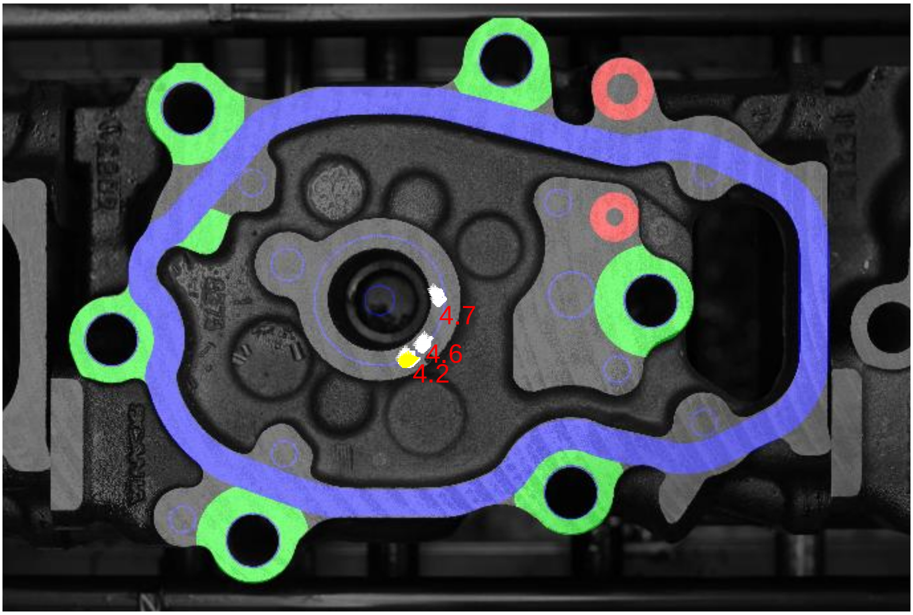

Figure 34 Inspected part compared with the inspection standard. All serious defects with their estimated size plotted on the image.

The algorithm compares the size and location of defect with the inspection standard and if their size (and location) is above maximum allowed limit then the part is declared defective and the responsible defects are displayed together with their estimated size.

Displaying the results

In addition to the presentation shown above, the algorithm also has two other operating modes. In the first of those two all found defects are displayed despite their size or if they were found severe enough. This mode is good for quick troubleshooting, namely for knowing if the algorithm was able to detect the specific pore in question and if it was ignored or if the algorithm missed it entirely. Moreover, the algorithm can run in a mode where the part is not compared with the standard and all found defects are displayed and whether the part was declared defective or not for quality analysis.

The software automatically saves all results of defective part for easy troubleshooting and later reference. Furthermore, it is planned to add functionality for generating various statistics where defects are most often found together with their size and other parameters calculated during the inspection. This kind of database would be useful for improving the quality of the product.

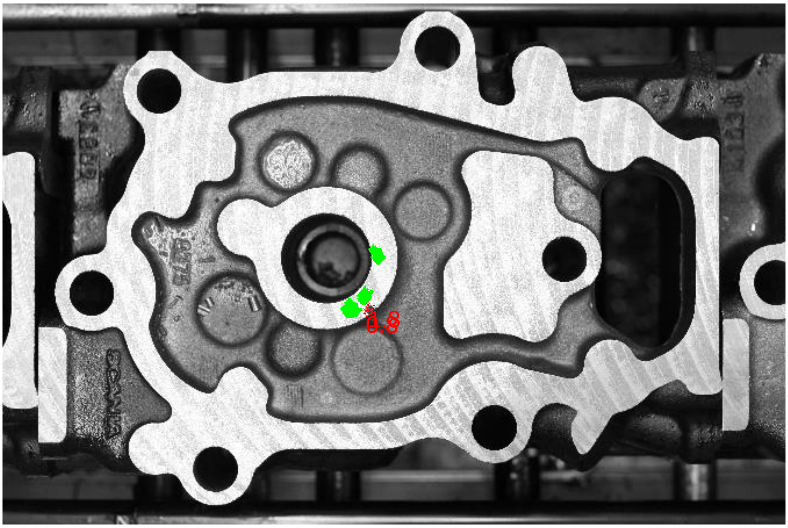

Figure 35 Same image as fig 44 but inspected with a different mode where all significant defects are shown. Note the addition of two found contour errors which were otherwise ignored as they have no effect on the performance

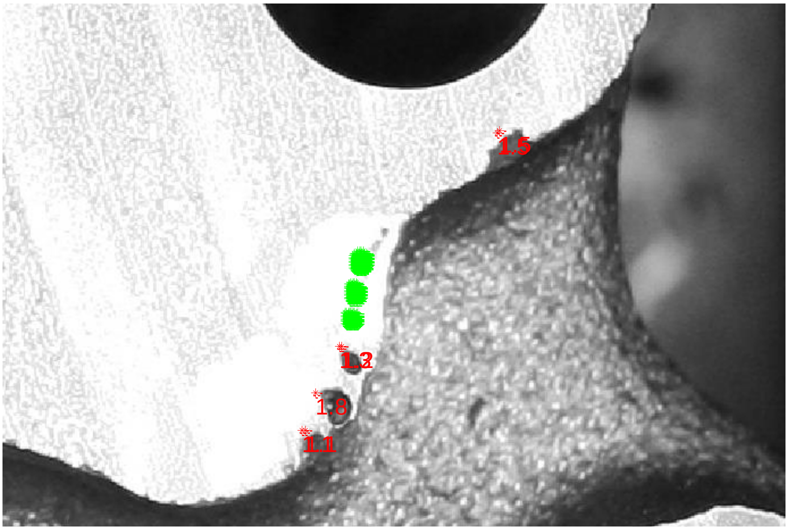

Figure 36 Variety of detected pores. Notice how some surface defects are so close to the contour that they have become part of the contour during processing and are thus found as contour defects. Also, note how two of the surface defects are below the set detection threshold and are thus ignored.

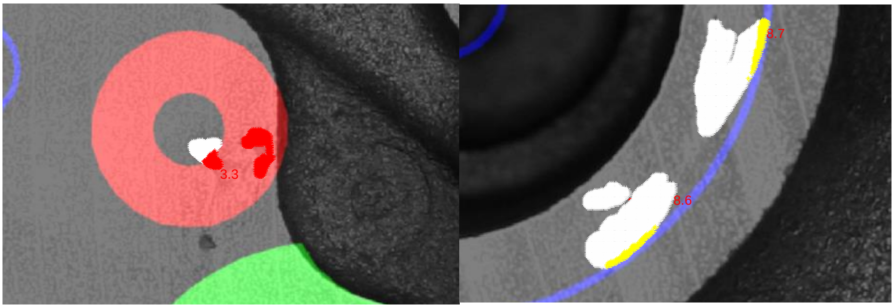

Figure 37 [a,b] Detected pores compared with the inspection standard. Defect on (a) that is not highlighted has no effect on component's performance and can thus be approved.

4. System Calibration and Comparison with Human Inspection

After the prototype system was set up, it was calibrated and the algorithm was tested by doing 150 inspections over a period of week. The number could have been much higher but since bugs were found in the algorithm time was lost.

Despite an initial understanding (before the tests) that the algorithm was near ready the initial tests showed that it lacked robustness for accurately finding pores in various regions and in many cases it was unable to extract the contours of the image due to different factors starting from wrong lighting, poor exposure and bad reflections from the conveyor and nearby objects. Various shortcomings in the existing methods were found which needed to be fixed to increase the systems robustness.

In addition, the system's parameters were calibrated to match the standard as closely as possible but after lengthy discussions with the operators it turned out that the standard was, in some places, not detailed enough. Thus the first test was done by picking out a set of different parts the system declared defective and they were given for different operators to inspect and comment. Based on this new information the system's algorithm was much improved and recalibrated. The system was calibrated to be slightly more sensitive than it normally would in order to find a good mixture of both small and large defects for creating a good sample for the test.

4.1. Blind Experiment

At this stage, the algorithm was considered to be ready for comparison with human inspection. To test the system as objectively as possible while not disturbing production the following blind test experiment was planned and carried out.

1. The system ran continuously and all parts it had declared defective were removed from the conveyor and mixed with randomly selected parts the system had approved
2. When the total of 50 defective and 50 approved parts were collected the parts were given random unique ID numbers from 1 to 100 without any indication if it had been declared defective or approved by the system

3. The parts were then independently inspected by three different quality inspectors and they wrote down numbers of all parts they found defective among those 100 and marked all their defects on a paper drawing. All parts were inspected in two modes; first 'No pores are allowed in all areas', second using the regular inspection standard.
4. Results were analyzed and visualized and the system was further improved and calibrated to match human inspection.

The vision system was operated in multiple sessions over a course of one week and roughly 500 parts were inspected. Also, no software or hardware changes were allowed during the test.

The purpose of the test

This test was designed to obtain a better understanding of what is actually considered a defect by the human inspectors. It has been shown in the thesis that the parts can have many harmless defects such as chipped edges, shallow scratches and also occasional dimensional inaccuracies in casted contours. Moreover, based on initial testing of the system and discussion with different operators, there was an observation made that each person had a slightly different opinion about the size of a defect that makes the part truly defective. In other words, the threshold level of what is a true defect and what is not is somewhat subjective and therefore it is difficult to calibrate the system.

Moreover, the test might show that the algorithm has deficient accuracy in some areas.

Observations from the experiment

The test showed how human inspectors checked the parts and it can be concluded that for every part first possible defect candidates were spotted, then those candidates were checked from various angles and then usually probed with a fingertip, finger nail or a tool such as a pen to have a better estimate of its severity.

The manual visual inspection of the 100 parts was done in a similar but not exact same location as the quality inspection (to avoid production stops). It took between 60-90 minutes for single inspection and involved some lifting and occasional forced breaks which simulated very well the real conditions in production environment. Moreover, the sample size of 100 parts was enough to be little bit tiring so that the inspectors might not have been able to stay at the peak of their attention throughout the entire test, further matching real life inspection conditions.

As the test was blind the operators did not know if the part they were inspecting had been approved or declared defective by the system so the psychological influence was kept to minimum.

Results

The experiment gave very interesting results which were put into good use to improve the algorithm. The chart below (Figure 38) is constructed so that it shows how many of the parts that the vision system considered defective were defective according to the quality inspectors (averaged). Similarly, it illustrates how many of the parts that the system approved were approved by the inspectors. This first test was done using a custom quality standard 'no defects are allowed in all areas'.

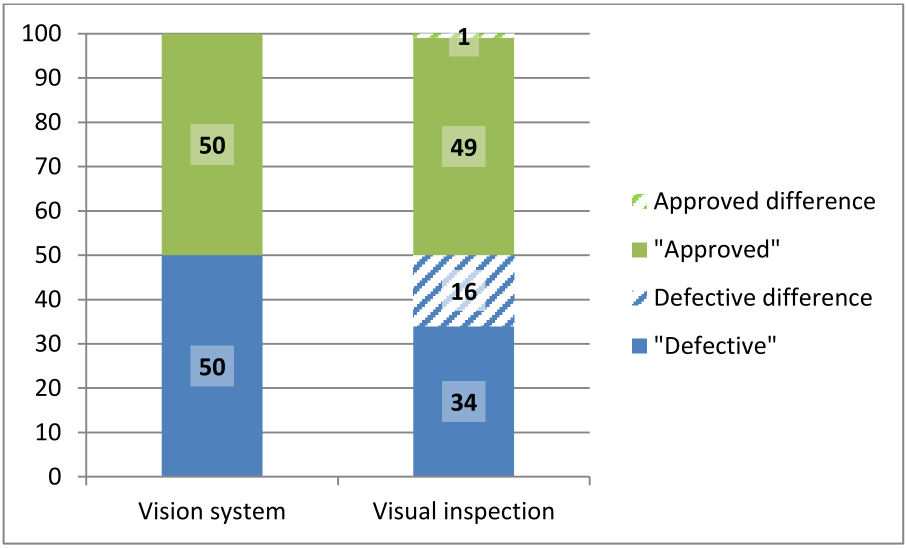

Figure 38 Blind Experiment results with averaged statistics.

As can be seen the system found more defective parts than the inspectors which indicates that the system might be too sensitive. Moreover, the inspectors declared defective one of the parts that the vision system approved.

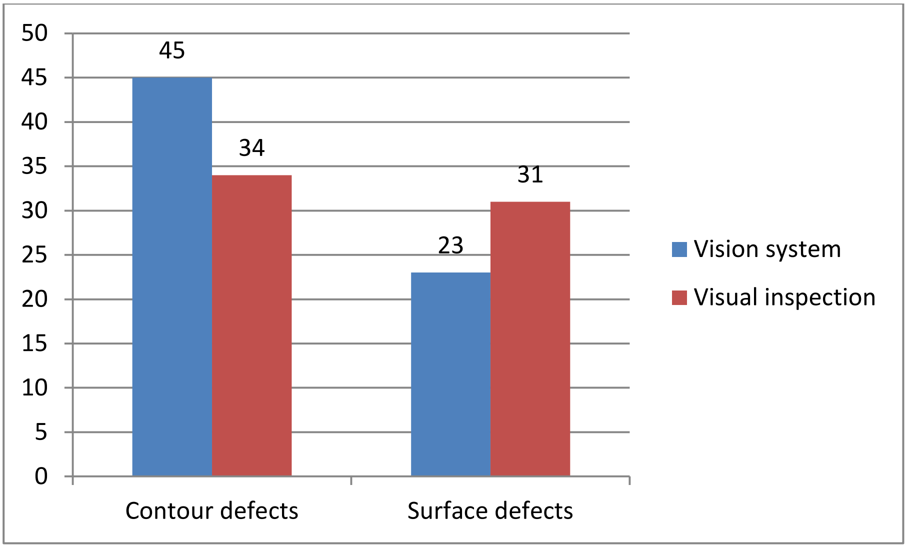

Figure 39 Total number of defects found by system and inspectors

The situation is made more complex by the fact that a defective part can have more than one defect. Therefore, in some cases the vision system found 3 defects while the inspectors found 2 and interestingly vice versa. Although, it must be said that when the part had multiple defects side-by-side then inspectors did not always mark the individual defects instead drawing a single symbol. Moreover, one of the inspectors failed to mark down the location of defects for 9 parts which is why that inspection was excluded from the statistics above.

The numbers above are the total number of each error type found and can contain both cases below:

1) System found a defect, inspector did not
2) Inspector found a defect, system did not

Therefore, the vision system did not find all the 34 contour defects that the operator did. For all the reasons above, the hereby statistics regarding the total number of found defects must be taken with a grain of salt until further analysis and discussion with inspectors for clarification will be held. If conclusions would be drawn from this, it could be said that the system is more sensitive to contour defects than surface defects.

The figure below shows the results of the test if the normal inspection standard was followed. As expected the overall number of detected defective parts is significantly lower. Still, the vision system found three more than the inspectors. It is known that one of the part declared defective by the system was caused by its inability to accurately extract the contours due to a discovered deficiency in the algorithm that has been fixed. Though, this false negative is still included in the statistics as similar event could theoretically happen even when the system is fully implemented.

It is also known that two of the parts that the system declared defective were due to the fact that the surface defects on those parts were situated exactly on the edge of a hole to be drilled later - a fact that was missed by inspectors.

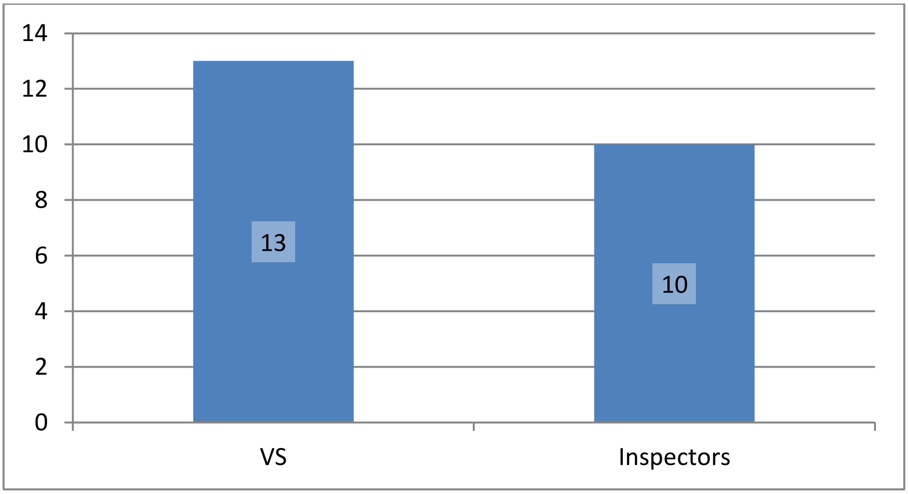

Figure 40 Experiment results when normal inspection standard was followed. Number of defective parts found out of 100.

Analysis of the results

The blind experiment shows that vision system found more contour defects but less surface defects than the inspectors did. This indicates that the system might have been configured to be too sensitive for contour defects during the test and thus it found too small defects that do not affect the part's performance and noticeably the visual appearance. The reason for somewhat weak performance of the system in finding surface defects was due to the aggressive filtering the algorithm does to cope with cutting fluid contamination. Since the fluid isn't really a problem, contrary to what was initially expected, thanks to the brush system, this part of the algorithm could be excluded or reduced which dramatically improves the accuracy and detection rate.

The human inspection is considered superior to the vision system as the latter cannot measure the depth of the defects or see them from multiple angles which is why all parts for which the vision system gave a different decision were further analyzed and the system was re-calibrated to better match human inspection.

5. Conclusions and Discussions

The task of designing a low-cost fully vision system for fully automated quality inspection of the described part proved to be very challenging. The main reasons are the large variety of possible defects in different locations, the complex contour shapes of the part and the large natural variations that the casted parts have. The aim of designing a low-cost system further increased the requirements on the algorithm as it had to handle factors that could in theory have been solved with hardware and using a different setup for the entire inspection. These factors include isolating the part, full control of lighting, noise free background and last but not least cleansing the part.

Handling the above with software requires extensive testing as there can be many possible cases where a part of the algorithm might fail. The testing period in this thesis was short reaching a total number of around 700 inspections. The number should be considered even less as the algorithm was (excluding the time of the experiment) continuously improved. Thus, not the same algorithm was tested with 700 images. According to IEE guidelines a vision system should be tested at least 1000 times during beta testing and up to 100 000 times before final approval [**8**].

The testing has suggested that the algorithm performs well in different conditions and can find a large variety of defects with satisfying accuracy. The blind experiment showed that the system can stand comparison with human inspection. However, the system using a single image can theoretically never outperform human inspection due to lack of depth information.

Testing also showed that the system can be improved in almost all areas. The biggest opportunities for improvements are within its accuracy in some contour areas and more accurately capturing the size of surface defects that lack contrast. Furthermore, the overall robustness of the system can be improved by further developing the edge detection and masking part of the algorithm for more precise extraction of contour information while reducing occasional loss of detail. Moreover, it would be interesting to test the algorithm with higher resolution images to see if this makes a noticeable improvement in the performance.

Various suggestions for improvement have been made throughout the thesis and it can be said with high confidence that improving the algorithm is just a matter of investing more time in it and testing the algorithm further. During the thesis a database of near 700 images has been created which can be used for offline testing and development. Furthermore, the speed of the algorithm could be increased by porting the code to a native programming language such as C++ with OpenCV libraries.

The system and its image processing algorithm are still at prototype stage but they are not far from being ready for implementation as a permanent part of the production line. Though, details regarding the required support systems have not been discussed in this thesis.

Bibliography

[1] Boyle, inventor. 1970. US 3796927 A.

[2] Alper. Adimec Offical Blog. 2012 [Internet]. Available from: http://info.adimec.com/blogposts/bid/92621/Machine-vision-has-gone-from-niche-to-mainstream-Machine-Vision-over-the-last-25-years.

[3] AIA. Automated Imaging Association Official Webpage. 2013 [Internet]. Available from: http://www.visiononline.org/userAssets/aiaUploads/file/AIA_Business_Conference_AES_2013_FINAL.pdf [Accessed 2014 May 21].

[4] Kellet P. Machine Vision Markets. Automated Imaging Association (AIA); 2005 May.

[5] Dr.-Ing Waszkewitz, Peter; Robert Bosch GmbH. Machine Vision in Manufacturing. p. 693-775; 712; 754.

[6] Mital A, Govindaraju B, Subramani A. A Comparison Between Manual and Hybrid Methods in Parts Inspection. Integrated Manufacturing Systems. 1998:344-349.

[7] Malamas EN, Petrakis EGM, Zervakis M, Petit L, Legat JD. A survey on industrial vision systems, applications and tools. Image and Vision Computing. 2003;21:171-188.

[8] Davies ER. Computer and Machine Vision: Theory,Algorithms,Practicalities. 4th Ed. Elsevier; 2012. p. 526;757-759;271;758.

[9] Ying-dong Q, Cheng-song C, Shan-ben C, Qing-chun L. On-line measurement of deposit dimension in spray. Journal of Materials Processing Technology. 2006;172:195-201.

[10] Dworkin SB, Nye TJ. Image processing for machine vision measurement of hot formed parts. Journal of Materials Processing Technology. 2006;174:1-6.

[11] Bay H, Ess A, Tuytelaars T, Gool LV. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding. 2008;110:349-359.

[12] Lowe D. Distinctive image features from scale-invariant keypoints. IJCV. 2004;60(2):91-110.

[13] RoboRealm. RoboRealm official webpage. 2014 [Internet]. Available from: www.roborealm.com [Accessed 2014 Jun 01].

[14] OpenCV. OpenCV Documentation. 2014 [Internet]. Available from: docs.opencv.org [Accessed 2014 Jun 01].

[15] OpenCV. OpenCV CheatSheet. [Internet]. 2014 Available from: docs.opencv.org/trunk/opencv_cheatsheet.pdf [Accessed 2014 Jun 01].

[16] Tetherscript Technology Corporation. ControlMyCanon™ feature list. 2014 [Internet]. Available from: www.controlmycanon.com/#!features/c13w5 [Accessed 2014 Jun 01].

[17] Martin D. A Practical Guide to Machine Vision Lighting - Part I. [Internet]. 2012 Available from: http://zone.ni.com/devzone/cda/tut/p/id/6901 [Accessed 2014 May 01].

[18] Martin D. A Practical Guide to Machine Vision Lighting - Part II. [Internet]. 2013 Available from: http://zone.ni.com/devzone/cda/tut/p/id/6902 [Accessed 2014 May 01].

[19] Martin D. A Practical Guide to Machine Vision Lighting - Part III. [Internet]. 2010 Available from: http://zone.ni.com/devzone/cda/tut/p/id/6903 [Accessed 2014 May 01].

[20] Amazon. 2014 [Internet]. Available from: www.amazon.com/BIS-6922-Fanless-i7-3740QM-Factor-System/dp/B00DH3WQQ8 [Accessed 2014 Jun 01].

[21] Anandtech. Logic Supply ML-320 fanless industrial NUC review. 2014 [Internet]. Available from: www.anandtech.com/show/7982/logic-supply-coreml320-fanless-industrial-nuc-review [Accessed 2014 Jun 01].

[22] Anandtech. Intel D54250WYKH Haswell NUC kit with 2.5" drive slot minireview. 2014 [Internet]. Available from: http://www.anandtech.com/show/8028/intel-d54250wykh-haswell-nuc-kit-with-25-drive-slot-minireview [Accessed 2014 Jun 02].

[23] Gonzalez RC, Woods RE, Eddins SL. Digital Image Processing Using MATLAB®. Gatesmark Publishing; 2009. p. 546;82;619.

[24] R.C G, Woods RE. Digital Image Processing. Pearson Education International; 2008. p. 832.

Appendix 1 Sample Code from the Algorithm

```
Function [CenterPoints,p,P,t,ind]=findcenterV3(p,dt)
%by A.Kiviorg, KTH (student)
%
%Returns the center pixel(s) in a segment of consecutive %pixels given in nx2
matrix of pixel coordinates with unknown %amount of segments and length.
%Consecutive pixels are defined by the distance between them. %If the distance
is less or equal than dt then they are %consecutive.
%If the segment consists of equal number of pixels the %function returns the two
centermost pixels of that segment.
%
%   INPUTS
%       p - points in n x 2 format ( [r, c] ), p can contain %other information
as long as coordinates are in first two %columns.
%       dt - the distance threshold. if distance between two %consecutive pixels
is >dt, they are considered to be part of %different segments(different pores).
%
%   OUTPUTS
%       CenterPoints - the center pixels of all found segments

%Makes sure all points are unique. (removes duplicate entries from rounding).
px=p(:,1:2);
[~,uniq]=unique(px,'rows','stable');
p=p(uniq,:);
P=[]; t=[]; %Allocate matrices to outputs, to avoid errors.

%Calculate the distances from one pixel to the next
Distances= sqrt(sum(diff(p(:,1:2),[],1).^2,2));
Distances=cat(1,0,Distances);
p=cat(2,p,Distances);

ind=find(p(:,end)>dt);

CenterPoints=[];

if isempty(ind)
        n=size(p,1);
        if n==1 || n==2
            CenterPoints=cat(1,CenterPoints,p);
        else
            t=round(n/2);
            P=p(t-1:t,:);
            CenterPoints=cat(1,CenterPoints,P);
        end
else
N=numel(ind);
    for i=1:N+1
        if i==1
            P=p(1:ind(i)-1,:);
            n=size(P,1);
            if n==1 || n==2
                CenterPoints=cat(1,CenterPoints,P);
            else
                if mod(n,2)==1
                    t=round(n/2);
                    P=P(t,:);
                else
                    t=n/2;
                    P=P(t:(t+1),:);
                end
                CenterPoints=cat(1,CenterPoints,P);
            end
                    if N==1
```

```
                    P=p(ind(i):end,:);
                    n=size(P,1);
                        if n==1 || n==2
                            CenterPoints=cat(1,CenterPoints,P);
                        else
                            if mod(n,2)==1
                            t=round(n/2);
                            P=P(t,:);
                            else
                                t=n/2;
                                P=P(t:(t+1),:);
                            end
                            CenterPoints=cat(1,CenterPoints,P);
                            break;
                        end
                    end
            else
                if i==N+1
                    P=p(ind(i-1):end,:);
                    n=size(P,1);
                    if n==1 || n==2
                        CenterPoints=cat(1,CenterPoints,P);
                    else
                        if mod(n,2)==1
                        t=round(n/2);
                        P=P(t,:);
                        else
                            t=n/2;
                            P=P(t:(t+1),:);
                        end
                        CenterPoints=cat(1,CenterPoints,P);
                    end
                else
                    P=p(ind(i-1):ind(i)-1,:);
                    n=size(P,1);
                    if n==1 || n==2
                        CenterPoints=cat(1,CenterPoints,P);
                    else
                        if mod(n,2)==1
                        t=round(n/2);
                        P=P(t,:);
                        else
                            t=n/2;
                            P=P(t:(t+1),:);
                        end
                        CenterPoints=cat(1,CenterPoints,P);
                    end
                end
            end
    end
end
CenterPoints=CenterPoints(:,1:(end-1)); %Remove the distance information
return
```

Appendix 2 Process Flow in Prototype System