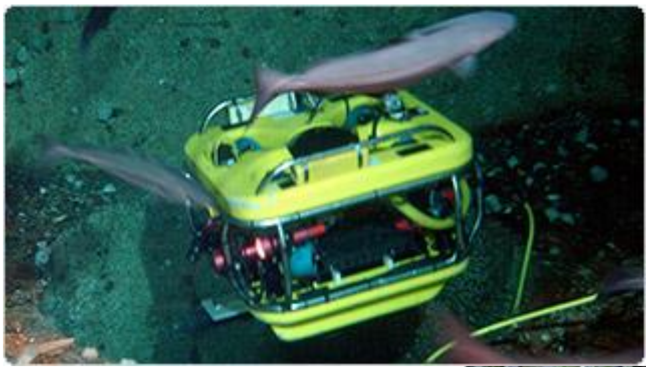
CHALMERS

Analysis of

Sonar

& Underwater Camera Images



Master of Science Thesis

Emil Andersson

Chalmers University of Technology

University of Gothenburg

Department of Computer Science and Engineering
Göteborg, Sweden, September 2010

Research on the potential of image enhancement on Sonar and Underwater Camera Images.

Emil Andersson

Examiner: Ulf Assarsson

Chalmers University of Technology

University of Gothenburg

Department of Computer Science and Engineering

SE-412 96 Göteborg

Sweden

Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering

Göteborg, Sweden September 2010

Abstract

The thesis is researching the possibilities to use filters and advanced algorithms on sonar and underwater camera images. Can filters used for normal images be as helpful on underwater images; and is it possible to use these filters on sonar images that represent sound reflection instead of light reflections?

Besides the normal filters (*image enhancements*) a parallel research on *pattern matching algorithms* will take place. This topic is just recently started to being used in surveillance applications; the implementations and algorithms are advanced. The research will find out if these algorithms are efficient enough to run in real-time and at the same time delivery useful data to the user.

As a part of this thesis, software for underwater inspections will be develop for the ***Australian Custom Service***. This will in the most parts give extra good occasions where filters and algorithms can be tested in real situations and/or on material from real inspections.

Sammanfattning

Detta examensarbete ska forska kring om filter och avancerade algoritmer kan med lyckat resultat köras på sonar och undervattenkamera bilder. Kan förbättringsfilter använda på konventionella bilder även användas på bilder från undervattenskameror; och är det möjligt att använda dessa filter på bilder återgivna från en sonar som visualiserar ljudreflektioner istället för ljusreflektioner?

Vid sida om de vanliga filtren (förbättringsfiltren) ska forskning gällande mer avancerade mönster letande algoritmer göras. Detta område har de senaste åren börjat utvecklas för övervakningssektorn; implementationerna och algoritmerna för detta är mycket komplexa. Forskningen utreda om dessa algoritmer är effektiva nog att köras i realtid samtidigt som de levererar hjälpsam information till användaren.

Som en del I projektet ska mjukvaran för inspelning av undervattens-inspektioner utvecklas för den Australiensiska Tullen. Detta kommer ge goda möjligheter för att testa filter och algoritmer i verkliga situationer och/eller på material inspelade under riktiga inspektioner.

Acknowledgments

I would like to thank the employs at Ocean Modules Australia for giving me the chance to do my master thesis at their office; and for all the support and feedback I've received.

A big thanks to the **Australian Custom Service** for the feedback on the software and the material used to try out filters and algorithms.

Last but absolutely not least, a big thanks to **Ulf Assarson** at *Chalmers Technical University* for a suburb supervising and great feedback during the time the thesis was written.

Innehållsförteckning

Introduction

The marine world has used sonar equipment for a very long time as an aid to locate shallow waters, fish, wrecks etc. The data you get from the sonar equipment is often pretty bad and hard to understand without the right knowledge and as more and more areas of use for sonar are found an aid to interpret the picture from the sonar can be at help.

In these last year’s image analysis software and libraries have exploded and the potentials for using these with sonar equipment as an aid to get out more useful information for the user is getting bigger.

This project is aimed towards underwater sonars used on ROV's (Remotely Operated Vehicle). The ROV has besides the sonar also an underwater camera that delivers a live feed from the bottom. Other useful data such as depth, pitch, tilt and temperature can also be retrieved in real time from the ROV.

A big part of the image analysis is filtering, enhancing and highlighting the sections of the image that is of interest, while removing noise and other part of no importance.

The main idea and purpose of this project is to investigate the possibilities to use image analysis and image recognition on sonars and underwater cameras. Tests and benchmarking will find out if data and other aiding information can be collected to provide the user with helpful information to ease usage of sonars and underwater cameras.

Also, research about the use of Camera and Sonar data together to retrieve even more useful information will be researched. This will be implemented in a simple survey software that a current customer have ordered. The software will be given the possibility to record all the data into a files that later can be retrieved and discussed by the crew etc.

Method

The project will use the Evolutionary project and development model to evolve in. At first we looked at the Incremental Model which in many ways is similar to the Evolutionary model. The main difference between them is that the Evolutionary Model finished one *iteration* (loop) before it starts with the next one. Because there is only one person in this project the feature to start planning for the next *iteration* before the current one is finished is unnecessary to pursue and would take time away from the current *iteration*.

The Evolutionary model is designed for projects where you don´t know all the requirements in advance; this project is meant to have continuous user tests to find out requirements and features along the way to reach a final release of the software.

The project will have three iterations where each will generate a new version; the later better and more rich in features than the previous. Each iteration has three phases which are the following:

Requirements/Specifications

Deciding the features and goals to be achieved in the current iteration.

Development

Developing period where the new version is developed from the requirements decided in the “Requirement” phase.

Testing

The new version will be tested as much and close to the reality as possible with pool dives using the ROV. Small sea expeditions and user tests with the customers will be done as opportunities arise.

Surveys

Two different surveys are used for this project.

First one identifies what the customer thinks about the existing system, highlighting problems and things that are not like they want them to be.

The second survey has the purpose to assist the development and gather requirements for the new software. The first part of it will have questions about features and requirements the customers want to have, while the second part gives more detailed options and descriptions about the software’s UI (User Interface) and user friendly aspects.

Feedback from the survey results will be used to form requirement before starting development of a new version.

Tests

Tests are done continuously throughout the development by the developer and staff at the office. A few tests are to be made by the customer to get valuable feedback and ideas, mostly for the software´s GUI (Graphical User Interface).

Previous work

The beginning of the research focused on what already had been done within the different areas. Most important was what, if any, had been done on sonar image filtering and enhancement. Other areas researched were image enhancement, image analysis (gathering important data) and image recognition for both still images and video feeds.

Sonar

Research showed that not much had been done about the sonar image itself. The companies that were looked into were particularly ***Blueview Technology*** and ***Tritech Ltd***; both because it was with their sonars that would be used later on in the project. The major filtering and enhancement they have done are built into the sonar itself, hardware filters that removed noise. Software wise they have some basic enhancement in form of color maps. Color maps changes the color spectra which gives the image better contrasts. In Blueview´s software *"Proviewer"* another enhancement called *"Auto Intense"* was found which automatically changes how much the image would be intensified; by analyze the image. If the image has nothing to show, the program automatically intensifies the image until something is shown. In a lot of cases the noise is magnified; this usually happens when nothing else then noise is to be found.

Image enhancement

The technology to enhance image quality is not new, but the last couple of years the images analysis field has been exploding with new hardware and software techniques of enhancing.

This is a quite new environment for the image data to come from. Images from underwater cameras usually have a lot of noise in them and how easy it will be to remove these are yet unknown; noise mostly appears as particles and unclear water.

Image analysis

To gather useful information from images is very specific, usually static pictures, but nowadays also videos. In security manors gathering information about people moving around in front of cameras is used for getting better security while having cameras on a football game can gather information about the player's movements and handle statistics. There is not much general information to gather from an image, average color values, brightness etc. During the pre-study nothing was found about analysis on sonar or underwater images.

Image recognition

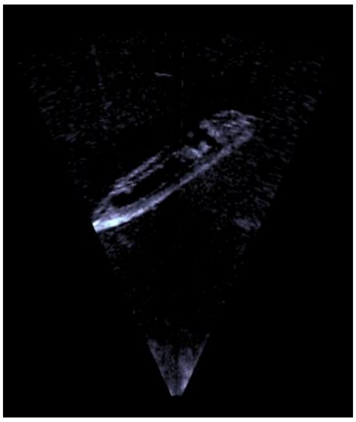
To recognize objects in an image is pretty new in the area of image analysis. The technology lets the computer look up patterns in the image. These patterns can be simple things like lines and circles or more advanced like finding a specific face in a crowd using a picture of the person. This process takes a lot of computer processing power and real time applications may be hard and expensive to achieve.

Analysis

The research of new filters and enhancement strategies on sonar and underwater camera images is tested on real recordings, partly contributed from the ***Australian Customs Service***. This ensured that the filters are tested in the intended work environment and not in an artificial environment like a pool. Water quality, amount of particles and the artificial framework of a pool can have a huge effect on the retrieved sonar and/or underwater camera images.

Image data from sonar equipment and video cameras is saved in a data file format that is easy to playback. Video files are usually recorded in either MPEG2 or MP4 formats, while the sonar files get stored in a special format called *"SON"* that contains a lot of useful information besides the actual feed. (The .son format is specific to ProViewer)

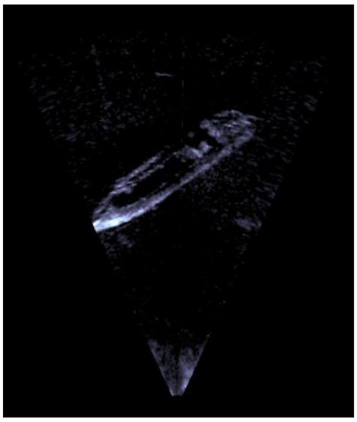
Filters are something that is applied to an individual image (or several times on a video feed) to remove something. It can be a specific color or area. More advanced filtering can be removal of contours or blurry areas (blurry parts usually indicates something with no solid edges, therefore not a real object).



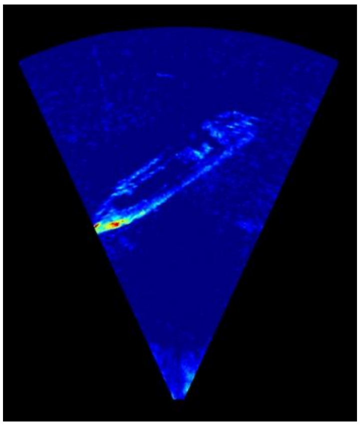
Original Sonar Image

Filtered – Noise removed

Enhancements in its turn are when you make something more visible or highlighting it in the image. Good examples of image enhancements are changing the contrast or applying a color map to the image.



Original Sonar Image



Color mapped Image

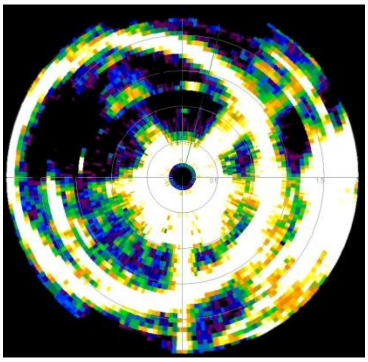
Sonar

Sonars come in many different types and models. The most common types of sonars are explained below. The sonar works like a sound-camera and they only work in water and other liquids. A sonar is an underwater radar; both use sound waves to locate objects in a 2D plane.

The data or image that the sonar sends back represents reflections of sound. Different to an ordinary camera, the sonar have the ability to see behind objects.

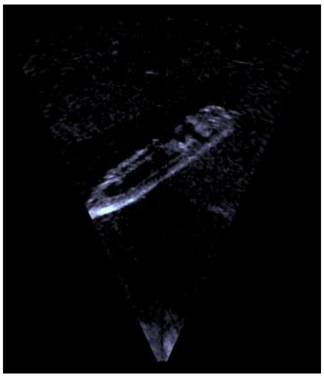
Sonar - Mechanical Head

A traditional sonar gives an radar-like image. It updates one part, a specific direction each time, which gives the typical “sweeping” look of a radar. This is the Tritech Sonar used in this project.



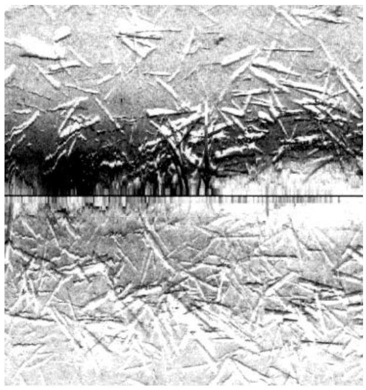
Sonar - Digital Head

The unique feature of this sonar is that it functions much like a normal camera, giving the user one new image per update. This gives a smooth transition between frames. Looks like a video feed. This is the BlueView sonar used in this project.



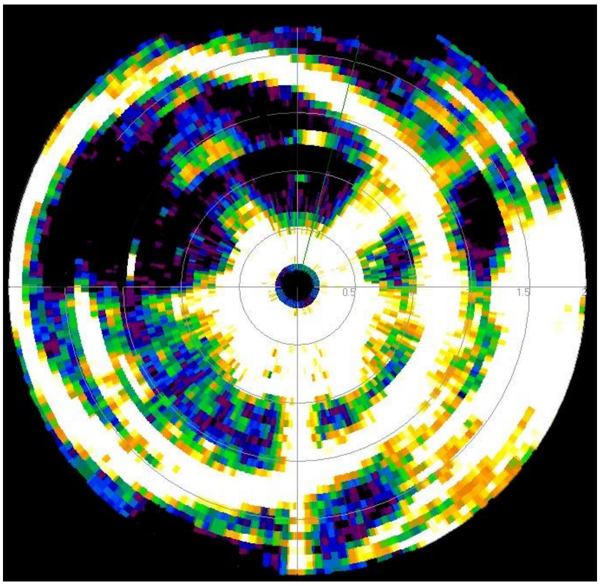
Sidescanning Sonar

This kind of sonar is usually dragged behind a boat, sequentially scanning the seabed and literally adding one new line to the picture in each update.



Potentials

The idea of filtering the sonar image seems new and the few existing attempts are of a very basic nature. As all images can be enhanced in one way or another, this is for sure one of those areas that needs it the most. As seen on example sonar images, they are covered in noise and look like scrambled mess.

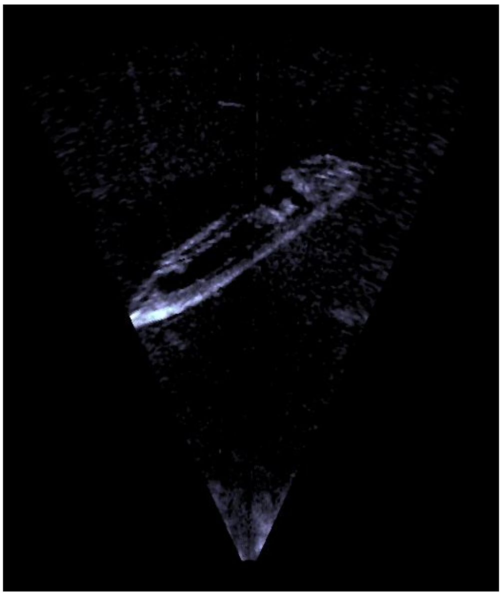


For an inexperienced person the image may looks confusing and worthless. A moving sonar feed is easier to understand because your brain sees patterns and common factors from a frame to frame basis which in its turn gives you a feeling for movement and appearing objects.

Letting the computer process the image before presenting it on the screen gives the opportunity to enhance the quality. By improving contrast between objects, filter noise and “black out” unwanted elements the image will be easier for the user to interpret.

Current Solutions

Filters and enhancements found in today’s sonar software are of a very basic nature. Filters are not commonly used. The only one found was a simple threshold filter used to remove reflections that doesn’t reach the lowest intensity allowed (set by user).



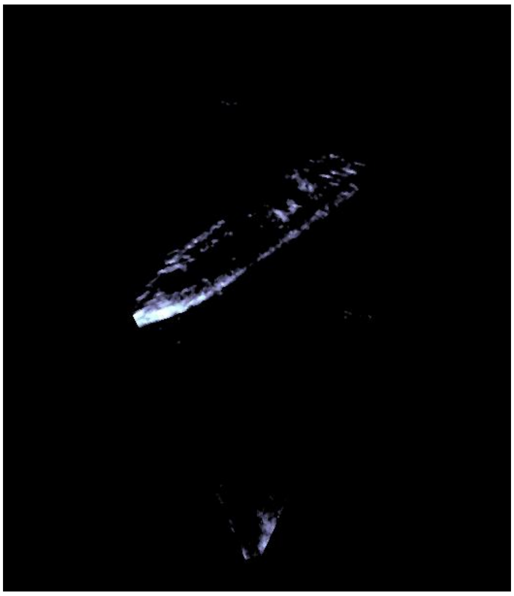
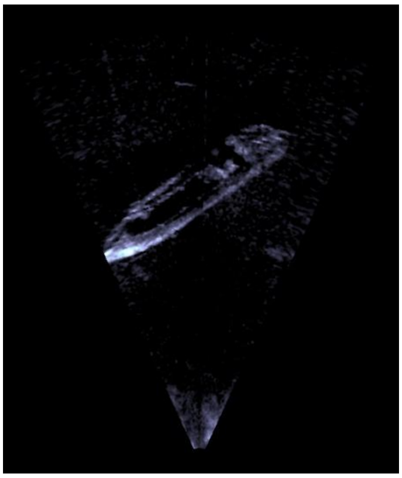
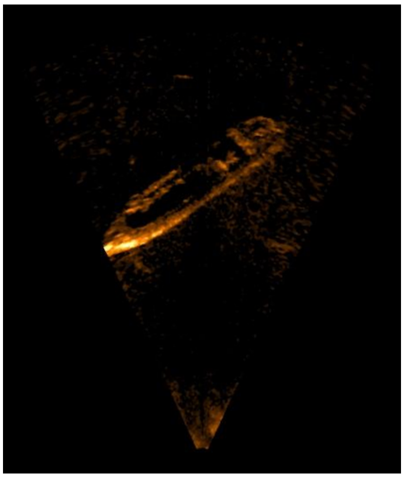
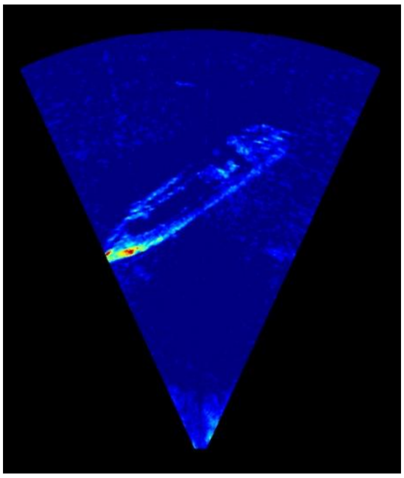


Image With Threshold

The only enhancement found was by using color-maps on the sonar image. Using color-maps is a way to intensify specific spectrum of an image color data, usually grayscale with 8bit or 16bit depth. It basically replaces the gray color with a color that makes the contrasts of the image more obvious for the human eye. Common colors (color-maps) are copper, red and green. Also a rainbow colored map is often used, usually called “jet”. Color-maps are used in various applications and the main purpose is to show contrast (variations) in a range of values, making it easier to find patterns.







Normal Copper Jet

New Ideas

The main tests and research are about trying filters and enhancements that have not yet been tested in this kind of application. Examples are the **binary**

threshold and **rectification** filters and enhancements like **saturation** and **contrast**. The two last ones are used for enhancing the picture quality while the binary threshold and rectify are used for more specialized applications such as heat cameras and in the Wiimote (Nintendo Wii controller).

To research intelligent solutions for pattern recognition and identifying what is worth seeing and what is not on the sonar image. This will probably be hard to do in real-time but some applications may allow this to be done in post-time. Real-time is to be able to run with the same frame rate as the sonar outputs data and without any noticeable delay.

There is a whole group of filters, enhancements and other ideas that are going to be researched; and later tested if the research shows some potential. Most time will be used to finding potentials for image recognition and possibilities on filtering the image through removing things that are not important.

Rectify (Filter)

When an image is full of small variations between pixels, often caused by noise, this filter may improve the quality. The filter works by checks the "size" of a specific variation, if it is smaller than the minimal accepted size it will be blanked out (removed).

Threshold (Filter)

There are a couple of different threshold filters. What they all have in common is that they remove a pre-decided spectrum of color in the image. This spectrum may be a part of the picture that is "too light" or "to dark" and sets the color in that area to zero (*0*).

Usually the rectify filters remove a specific part and leave the other part untouched, but the "binary" group is different. Binary filters makes the part not removed white (*1*), the picture then becomes binary with either black or white pixels (*0* or *1*).

Contrast (Enhancement)

This is a very basic operation and is the same as in video and picture enhancement. By testing different changes in contrast on an image, to see if it gives an image easier or harder to interpret.

Saturation (Enhancement)

The saturation of an image is the amount of color in it. Like contrast, saturation is also used for picture and video enhancement. The idea is to do the same tests as with contrast; see how different values can affect the quality of the sonar image.

Pattern Matching

In some cases it can be of great help if the computer itself could find simple patterns in the image and notify the user about it. Simple patterns like straight lines, circles and rectangles in an image are very likely to be manmade and for that reason often what you are looking for.

It's going to be researched and tested if pattern matching is possible with the hardware used for these applications today; and also to see what can be done or what would be nice to be able to do with this technology in the future.

Image Recognition

When the user wants to find much more complex pattern than *Pattern Matching* can find; Image Recognition is used. Useful examples are when locating pipes or power cables on the seabed or finding a valve on an underwater construction.

The concept is built up on two major parts; first the computer needs to know what it's looking for and secondly it wants to find that in an image.

The first part of getting the computer to "know" what is it looking for is done by having a couple of pictures of the object and then processing those images to find common patterns.

After the "knowing what to look for" part is established the computer uses the complex pattern calculated to look through images.

By research and tests find out if the hardware can handle these complex processing and pattern matching parts without causing the hardware to delay the information to the user.

Test of Ideas

The ideas will first be tested on a theoretical level to see if they have a potential to improve the quality or contribute together with other methods for improvement.

Next step is small scale testing where the ideas will be tested on single images with no time constraints what so ever. This will show if the idea works in reality, "Proof of concept" testing.

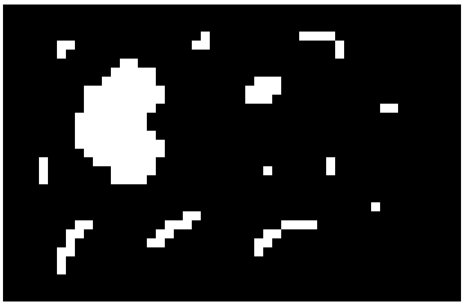
Finally it is important to find out if the ideas can be put into real-time applications, partly if the hardware is powerful enough but also test on the hardware used in the industry (usually older hardware).

Rectify (Filter)

The first thought about this filter came from the usage of a similar filter in a Bachelor thesis (Interactive bar, CTH 2006). The filter can be implemented in several different ways. Below are two ways which are easy to understand.

As a starting point we have a binary picture where **black** is no reflection and **white** is full reflection from an object.

When explaining complexity below, N and M are the height and width of the image while k is the constant for small work being done <u>one time</u> in the method.



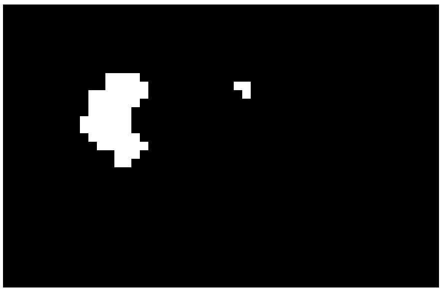
Shrink n' enlarge – method

The filter works exactly like that name says. First it shrinks all the white areas (the hotspots that represent reflections). After that it gets scaled up again.

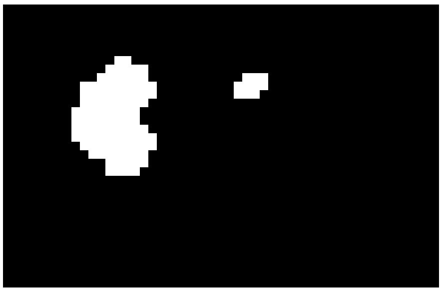
What happens is that small areas will first be shrunk to extension and when they get enlarge the small once don't exist anymore and therefore can't be enlarged. Only the bigger areas will regain their normal size in the end.

The black-out-rule is that if a single white pixel has contact (either vertical, horizontal or diagonal) with a black pixel it should be blacked out in the shrink process.

The rule for the enlarge process is the inverted, all black pixel in direct contact with a white pixel should be changed to white again. This reinstates the old condition of the hot-spots still left after the shrinking process.



Shrinked original image



Enlarged back to original size

The complexity for this method would be 8*(N*M) for the shrinking and 8*(N*M) for the enlarging plus some functionalities in the middle, this gives the complexity (16*(N*M)) + k. Complexity ends up to be O(N*M). A typically image resolution in this case would be 640 x 480 pixels.

“Exclude area” – Method

This method is easier to understand but harder to implement then the “Shrink n’ enlarge”- method. The first step is to calculate the area of all individual hotspots. This will in theory be fast, but it is harder to realize because the function will involve recursion. When all areas have been calculated the method removes hotspots with less area then the minimal area specified in advance by the user.

Complexity for this method would be N*M for area calculating and N*M for removal plus some work in between, O(2*(N*M)+k). The complexity ends up to be O(N*M) where N and M is the height and width of the image.

Tests and Conclusion

In theory this filter works perfect and has a lot of improvements for better image quality on sonar images. The good thing is that it removes a lot of noise that is very common, almost obligatory, on sonar images.

One improvement that has not been tested is to make it applicable on grayscale images. This would take some work but wouldn’t be too hard to implement. How “fast” a hotspot is going hot by looking at how big color intensity difference it is between two pixels and then from this make decisions to keep or not to keep may be one way to solve the grayscale implementation. This will remove even more noise, but even better is that it will still keep small hotspots that have high intensity.

Over all a good filter, but the negative hardware restricting point towards that it takes to much computer power to be able to deliver a real-time filtered image. This is of course in consideration to the hardware that is used in today’s products. On a fast home PC this filter works fine. It should also be taken into consideration that other software and hardware are using some of the computers performance at the same time.

Threshold (Filter)

This filter has a Cut-Off value which gives two scenarios, either the pixel value is over that value or under that value. Lower that value always makes the pixel black (*0* intensity) while over usually leave the pixel untouched, although some variations of the filter modifies the values over the limit too.

As pointed out above there are some variations of threshold filters and underneath is a list of the most common one and how they work.

Threshold To Zero (inverted)

Values from zero (*0*) to the specified threshold gets blacked out (set to zero). This function is optimal to use when the hotspots are represented as a high pixel value. It is also good when filtering noise, because noise usually has a low intensity hotspot on an otherwise black background.

This threshold filter variant also comes in an invert version which does the exact opposite of the normal version. It blacks out everything between the highest possible value and the threshold and leaves the less intense values unchanged. This could be good in cases where dark colors represent high intensity or in combinations with other filters to reach a specific effect.

Binary Threshold (inverted)

Everything that is between zero (*0*) and the threshold value gets blacked out while the rest is set to the "max" value, usually one (*1*) or 255. This gives a "binary" image where a value (pixel) is either zero or has the "max" value specified in advance. Good for speeding up software and less memory use, the images have a depth of 1bit and is optimal for algorithm speed.

The invert version of the function does everything inverted and has the same pros and cons; speed and less memory use but you lose a lot of data that may have been of value when converting to binary images.

Threshold Trunc

This filter truncates the value and lowers the highest possible value to the threshold value; this gives an image where the lower values are preserved while the higher values are set to the maximum value that is predefined.

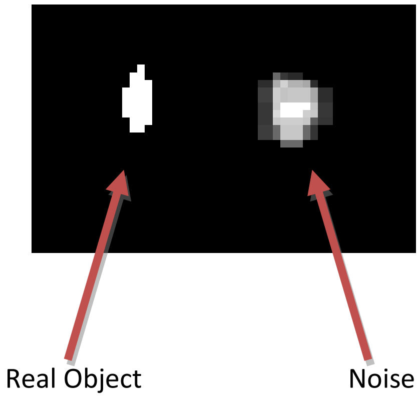
Good in applications where you want to lower the max value of the image. Can be used in situations where some objects give peek reflections that ruin an algorithms work. By lowering the peek reflections to the level of other "real" reflections; an algorithm based on intensity values may work much better.

Tests and solutions

The threshold filters are very useful for filtering noise. In some cases they remove objects that are of interest because the filter doesn't take size or how great the intensity of the edges is into consideration before excluding.

The most outstanding feature of a hotspot has more to do with the sharpness of the edge (intensity difference) then the actual intensity of the hotspot. Even thou at first the intensity seems to be the most reliable indicator for a real object; both real objects and noise can have the same intensity in the same picture.

While this is a sign of real objects; noise tends to have more smooth rising (blurry) edges.



Contrast (Enhancement)

This is a simple way to improving the image. By changing the contrast of an image you make different areas stand out more, they get a high contrast to other parts of the image. Because of the already wide use of this improvement, the tests will be small and it will mostly be tested as a combination with other ideas. Some of these ideas may not be mentioned here.

Low contrast will result in a blurry and grey image; hard for computers to understand because no edges and shapes can be found (or at least much more complex). Too high contrast removes a lot of information from the images that may be of interest, color spectrum, small areas of interest etc. disappears.

Tests & Conclusion

Contrast by itself has a minimal use for improving an image. For the human eye the improvement may make it "easier" to look at the image but we see exactly the same as before the contrast improvement was applied. This is because our brain has this built in; it will make it more "comfortable" to look at.

The main idea to research contrast was to see how it could help other algorithms and filters to improve their result. More contrast on an image may help the computer find edges easier; the edges are later used for pattern matching. When very high contrast was tested it showed a similar result as with very low contrast, the image lose a lot of its potential data and this makes it impossible for the computer to go further with improving the image.

Saturation (Enhancement)

As with contrast this is a simple and already widely used image improvement method. Saturation (Swedish, "färgmättnad") is the amount of color used. If increasing the saturation; the image will get a more colorful presentation. This is very common by photographs to make their pictures more vivid. An example is to make trees and grass greener and change the sea and sky to a more intense blue.

As with “Contrast”; the main research here aims to see how Saturation can aid other improvements and algorithms to reach a better result.

Tests & Conclusion

Saturation is more sensitive than contrast when it comes to the “extreme values”. Too much and too little Saturation applied on an image will totally destroy it; make it impossible for both for humans and computer to understand. And the tricky thing is that the limits of when this happens are dynamic and depend on how the image “looks” in ways of color spread and if one color is overrepresented.

After tests concerning mostly Saturation the result was that, by itself, Saturation had the same improvement as Contrast in that meaning that it just made the image more “pleasant” to look at. Computer wise it had little to none effect at all, at least not on grayscale images used today. In the future this may have a greater purpose when color image are used, but there is no real purpose for this today.

Pattern Matching

Pattern matching on underwater images has never been used or tried, at least not as what the pre studies to this thesis found out. The potentials for pattern matching on sonar images may be big and as this research probably will find out later is that the hardware may be the biggest problem for realizing this.

On sonar images you usually look for objects made by humans, and what is it that most significantly tells a man-made object from an object such as a stone made by nature? The shape and material used of course.

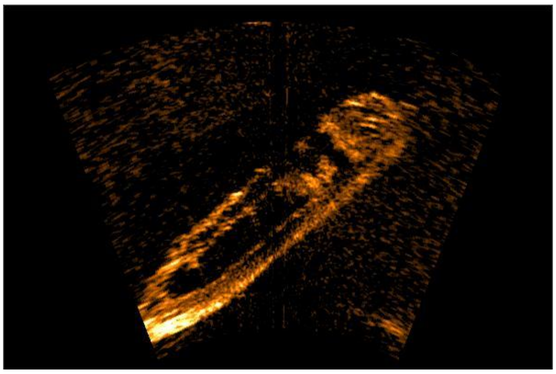
An object of nature has an irregular shape, no moulds have been used. Neither is the material in them always solid and of the same nature, a stone may consist of more than one kind of “stone”.

Man-made object usually are squared, circular or made from a mould that gives it a combination of these. Also solid made with **one** kind of material (per part) which gives them a more uniform reflection on the sonar.

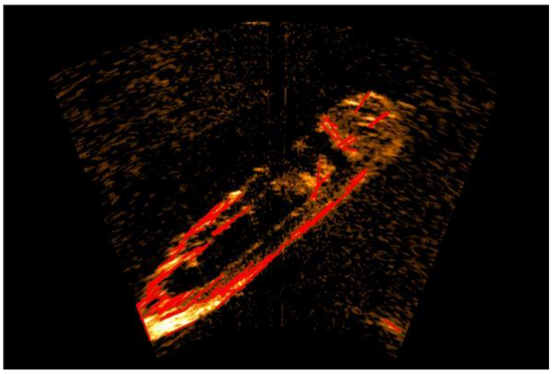
Tests

The major thing tested was to find “lines” in the sonar image. Lines should almost only appear on man-made objects. First tests on a still image took place to see what kind of result it returned. As seen below the results on finding lines in a sample image were good. It was quite easy to find settings which just found lines that were authentic and represented parts of the boat.

It was hard to measure how much computer power it took to analyze just one image, a second test on a short video clearly showed that on a normal PC it could reach 5-6 frames per second at best. This it of course nothing that can be used as a reference outside this test because a lot of parameter such as image size, complexity and color depth play a big role in the FPS result.



Original



Lines detected

Conclusion

The results that came out of these tests showed that the idea, pattern matching, may be useful in applications that run on a High-end PC. But on hardware used in today's sonars applications are just not good enough to get anything useful out of this method.

In the future when hardware gets more efficient; algorithms can run at higher speed and these can play a big role on applications where the sonar is not monitored all the time. To help notify personal if something of interest has been found on the sonar. As when using sonar equipment it takes a lot of time and this supervision aid may help doing it more efficiently.

Image Recognition

This idea has a lot common with pattern matching. In a way this is the same but more complex. "Pattern matching" has one step, to find a pattern predefined by the user. Image recognition have two steps which both are necessary. The first one is to calculate the pattern to look for and the other is as in "Pattern matching"; to find it.

Calculating Patterns

The patterns in "Image Recognition" derives from complex patterns in an image, so complex that they are seen upon as "Images" you're later matching against. There is numerous way of calculation these patterns but they will not be described here in detail.

When looking for these patterns the most important to have is a couple of pictures of the same objects. In slightly different environments and angles, if the differences are to big there won't be any patterns for the computer to find. Too few differences and there won't be information enough to tell the object apart from the rest of the background in the picture.

Pattern Matching

The matching is in simple terms the same as normal "Pattern matching", there is a specific pattern and it should be matched to an image to find out if it has that pattern in it.

The difference here is that the pattern consists of so much more "data" to match with, instead of a couple of simple lines it can be thousands of lines, squares and other combinations of primitives. There can also be patterns that are custom shapes made up of vectors.

As it is easily to understand the time and computer power needed is enormously expanding with complexity and the time to match is directly tied to the data that needs to make a positive match.

Tests & Conclusions

The Image recognition was not an area of priority in this thesis and has not been tested as thoroughly as needed to make any absolute conclusions. Simpler tests have been made, but not accordingly to the underwater environment that would have been preferred.

Tests showed that the time it took to match a specific image to another is highly dependent on the size and complexity of the object matching, but also very dependent to the target Image size. A solution that directly comes into mind is to decrease the size of the images to speed up the calculations. This has a big drawback though, by making the image smaller the details who represents the patterns may be distorted or removed complete. This makes the matching impossible.

The best result was achieved by making the generating of patterns for an image on images with the same size as the pictures that would appear on the target image. If the objects normally are 20% of the size of the target image with the size 1024x768 then the pattern should be made up from pictures that are 20% that size.

The conclusion that was reached was that the time taken to make the match is too long and on a dedicated computer would not reach more than 2 FPS (frames per second). Another factor making it unrealistic to use is the big variation in time it takes for different objects, this makes the goal of a steady FPS hard to achieve. It would be a nice feature to have in the future but there are too many unknown factors at the moment. How hardware and algorithms evolve highly affects what can be done in the future, also simply aspects such as if image quality from underwater cameras will improve or not.

Future

The potential of these techniques of enhancement and information collection algorithms are big and the idea of using some of them is very plausible. Especially the image quality enhancements have big potential because of the fact that their processing time is small enough for it to run in real-time; as people in the industry strongly emphasize this to be important.

Image enhancements are far from the simplest aids presented in this thesis, even though they may have the biggest importance. To make the image data more understandable is much more important than trying to interpret it for the user; as the human has extended knowledge to do that themselves. Some of the enhancements mentioned are of the nature that they can be implemented into hardware or even FPGA chips for increased performance. This has partly been done by the Swedish company LYYN.

Image interpretations done by recognition and pattern matching algorithms are far too slow on today's hardware to be of any use. Even though it might be possible to keep an acceptable FPS the delay would still make it "not current enough" to make decisions out of. The idea of letting the computer interpret the image data will be more and more feasible as new technologies arise. Because the complexity in the algorithms is so high it's almost impossible to accelerate it in hardware such as FPGA processors.

Video

An underwater camera is much like a normal camera but with the difference to withstand water and the enormous pressure that comes with it. By enclosing the camera in an underwater housing the camera gets protected from dirt, rough environment and most important, it gets waterproof.

The underwater housing is made out of a cylinder, usually of metal because of its ability to withstand high levels of stress. For lower depth the material and form has less importance. For the deep going camera houses the form is one of the most important factors where a normal tube or pipe isn't cylindrical enough and has to be machined to exact measures.

As with normal cameras there are different kinds and it is possible to get normal cameras as well as HD cameras with underwater housing. The signal is transferred to the surface through either a copper wire or converted into a digital format and sent up by fiber. Copper wire is the most common interface in use but it has a negative effect on the image quality. Copper can be used with cables reaching up to 500 meter in length, after that the other parts of the cable such as the high-power wires interfere too much with the video signal and it gets impossible to reach all the way to the surface.

The Typhoon Camera from Tri-Tech International has a housing handling up to 3000m of depth before cracking. Like a normal camera it has autofocus, zoom and a normal video signal as output.

Researching underwater camera problems and possible ideas of image filters, enhancement and algorithms is of second manor in this thesis. It will be covered in less extent and most of the ideas come from problems discovered while testing the ROV with sonar equipment. Sonar and camera share a lot of their problems such as data quality issues which makes filters and other enhancements go both ways some time.

Potentials & Usage

To use enhancements on underwater camera images has a lot of potential even though the most common filter as contrast and brightness have been used for a long time as they are built into old monitors. A quite new technique introduced by the Swedish company LYYN AB have implemented some semi-complex enhancement algorithms into hardware and are therefore fast and not dependent on a PC. This makes it easier to use and less complex to setup.

There are some drawbacks with moving the algorithms into hardware; first of all you lose the ability for smart algorithms which need a real computer to work. Advanced patterns matching and image recognition needs a much more flexible environment to work in; this can of course be solved with using both a PC and the new hardware.

New ideas

Patterns matching and the more complex image recognition algorithms will be researched and tested if they would work in these kind of applications. The hardware used in the industry today may be a restricting factor and the tests will be run on both modern PC and the more moderate equipment used in the fields.

Because the simpler enhancements such as saturation and contrast have been around some time they will not be researched any further here. The research will concentrate firmly on the more advanced algorithms.

Patterns matching

As given by the name the algorithms purpose is to find patterns. A pattern can be simple things like a line, circle or any other common forms. User specified patterns like characters or specific forms like hatches, grids or doors may also be of use in the subsurface industries when looking for structures. This will help the crew using ROV's to locate objects of interest in an unclear image or in an image full of information where you might lose focus to something else catching your eye.

Image recognition

An easy way to explain image recognition is to say it is like a very advanced pattern matching algorithm. The first step is to give the computer several images of the object of interest; from slightly different angles, lightings and surroundings. With these the computer finds common pattern that later is used for matching images against the object of interest.

The purpose with this is the same as matching patterns, to alert the user if wanted objects are matched on the image. This can be useful to have if you don't have a person watching the monitor all the time or if the ROV is navigating for itself and uses different object for its navigation path.

Test of ideas

The ideas for improving the use of underwater cameras are researched mostly on a theoretical level and may be tested in the future if there is time and opportunities arise.

Testing will be done on still images from real surveys and later tested on prerecorded surveys to get a better look at what kind of results may have been seen out in the fields.

Pattern Matching
By looking at common patterns such as straight lines and circles can help to alert the user that man-made objects may be present in the image.

Australian Customs Service is one of the big customers and the ideas of finding an aid for them came early in the process. The majority of their surveys are spent looking for suspicious objects underneath boats; one of the most interesting places to look is in the bow thrusters and sea chests. Sea chests are the latches under the boat where engine sucks in water for cooling.

The research focused on the bow thrusters because their nature of almost always look the same on different boats. The usual design is a big circular hole through the bow of the boat with two big grills of metal; protecting both objects from going into the propellers and stopping humans from hiding illegal material.

Tests & Conclusion
The tests were simple and straight forward tests on pictures of bow thrusters. The same software as used on the sonar images was used to find lines in the underwater camera images.

It took more time to analyze these images because they have 3 colors channels instead of just one with grayscale, but otherwise the same result. It was harder to find one single configuration of parameters that worked good for more than one image; to get one configuration working for more than one image involved compromising the result in a way that a lot of “real” lines got excluded because of the general parameter chosen.

Image Recognition
As mentioned before in the previous chapters, *Image Recognition* is a more advanced version of *Pattern matching*. To read more about the similarities please read the *Image Recognition* part in the *Sonar* chapter above.

Image recognition on video images has more difficulties with it then sonar images. Sonar images are not affected by dirty waters as video feeds are. A swirl from the thruster may stir up dirt in front of the camera which makes it impossible to see anything. Lightning is another problem in underwater images; lights are needed to see more but at the same time highlights particles in the water.

The possibilities here looked very good in the beginning but when the facts of how harsh the environment really is became clear, the hopes of a useful application got slim. The big difference between the tropical's clear water and harbors dirty water made it clear that one solution wouldn't work everywhere; the difference could be 40m in clear water while a staggering low of 0.3m in murky harbor water.

Tests

Pre-recorded underwater material showed directly that most waters that where subjects to any underwater work wasn't as clear as the tropics; usually so dirty that the camera wasn't used for other things then close up inspections.

Image Recognition occupies the computer with heavy duty work as previous tests on Sonar images showed, because standard video images have 3 channels of data instead of one practically made it 3 times slower.

Two strategies of improving the algorithms were invented and both had the negative effects of reduced data content and losing contrast in the image.

Putting together a grayscale image by combining the three color channels gave it the same speed as the algorithms used on sonar images; but when three channels gets combined to one a lot of contrasts in the individual channels gets lost or mixed together into patterns that doesn't exist.

Reducing the resolution to half the pixels and then analyze each color channel by itself prohibits different patterns to be mixed up. Finally adding the results together to a single image gave more "true" patterns then one combined to grayscale beforehand; this even though half the pixels in each image (one for each color) where used.

Conclusion

Algorithms to aid crews to find objects are at first glance a great idea even though it might feel that is has more of a "wow" factor the a real use to it. The hardware and algorithms used today are lacking in computing efficiency and reasonable frame rate are impossible to reach. This is the single biggest reason why Image Recognition is still in a developing stage.

Future

Image enhancing and running helpful algorithms on underwater video feeds will probably grow in interest along with hardware getting better. The potential is very high and could aid a lot in underwater observation even though people in such industries seldom want to try new technology if the one solution they have already works for them.

Image recognition technology will evolve even though this specific industry doesn't show interest in it at the moment. The recognition algorithms discovered and used in other fields will translate directly over to be used on underwater images. Though, to get the most out of image recognition more research have to be commenced on the specific area.

Results

Sonar

Sonar image data showed that there was a lot of noise in them, a lot that could not easily be filtered. Simple filters could remove obvious noise, typically noise that appeared small in size and even distributed over the image. Noise that is hard to discover can in some cases be removed by “smarter” filters; some checks the intensity changed in the image.

Checking the sonar images for classic forms that manmade objects have can aid operators to find objects they look for. Tests showed that it is hard to find one single algorithm-setup that could find these forms in different sonar images. Different levels of noise and refractions intensity makes it hard and not enough time was spent on an automatic adjusting algorithm.

Video

Common enhancements used on video since the old days still works but the most significant enhancement was contrast. Contrast works great here because it does exactly what is wanted; it increases the difference between different details which makes the image easier to understand. Brightness and saturation makes the image nicer to look at but can increase details or help and algorithm find more information.

Algorithms to find patterns and recognize more complex forms showed that it was plausible but not even close to be satisfying on today’s hardware. To find patterns in an image (taken underwater) you first have to filter it and then runt the algorithm; this all together takes too long time to complete. Future hardware may accelerate the time it takes but it will still be hard to reach an acceptable frame rate or a delay that would not be noticeable.

References

Here is further information about products and data mentioned in the thesis.

Sonar

P900E-20 Sonar	*http://www.blueview.com/*
Micron Sonar	*http://www.tritech.co.uk/*

Video

Lyyn	*http://www.lyyn.com/*