NTNU – Trondheim
Norwegian University of
Science and Technology

Realtime auto tracking CCTV cameras for increased safety and productivity

Joakim Skjefstad

2015-xx-xx

MASTER THESIS

Department of Engineering Cybernetics
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology

Supervisor 1: Professor Tor Onshus
Supervisor 2: Doctor Mads Hvilshøj, MHWirth AS
Supervisor 3: Thor Brantzeg, MHWirth AS
Supervisor 4: Johan Hansen, MHWirth AS

Preface

This master thesis is submitted in partial fulfilment of the requirements for the degree MSc. in Engineering Cybernetics at the Norwegian University of Science and Technology.

The motivation has been to study the feasibility of using computer vision offshore, and follow this up with an implementation of a potential application, hopefully aiding in increased safety and reduced cost of operations.

The motivation for this work has been to implement an automated CCTV system for tracking machines that are in motion, improving upon work previously done in the field. By visually following machines, I hope this work will lead to improved safety and productivity in the industry. Part of this thesis will also look into increasing reliability of control systems that are managing drill pipes on an oil rig.

I would like to thank my supervisor Professor Tor Onshus from NTNU and co-supervisor Doctor Mads Hvilshøj from MHWirth AS for their guidance and support throughout the project.

Keywords: computer vision, offshore, extreme environment, decision support, motion tracking, cctv, drilling vessel, oil & gas, drilling applications, machine vision, opencv

Summary

Here comes the summary.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

The Norwegian economy is cooling under a pessimistic oil price of close to 45 USD per barrel as of autumn 2015. Under heavy pressure to perform, companies involved in the exploitation of hydrocarbons are cutting costs while trying to increase productivity.

As a means of improving profitability, decision support systems have been developed, which tries to provide real-time support when drilling oil wells. One example of these was a product of work done at NTNU.

A roadmap for automating drilling systems was established in June 2013 with the objective of providing a guideline to the future emergence of drilling systems automation.

Automation of pipe handling have been pursued by several companies in the oil rig equipment production market with moderate success. The industry have realized that more may be gained from supporting humans in the center of the operation, than to completely eliminate humans. As such, systems to support the human driller need to be developed.

Support systems that may improve safety and productivity includes improvements to the graphical user interface, presentation of relevant and aggrevated data, as well as integrated camera systems.

The main objective of this MSc thesis is to implement a system that provides a better situation overview through automated CCTV tracking of machines.

This MSc thesis should address the following:

- Implementation of an auto tracking CCTV system using glyphs as visual descriptor.
- Analysis of dataset with real-world weather over a long period.

- Prototyping of an algorithm to detect casings and pipes in a fingerboard.

1.1.1 Problem formulation

How can computer vision assisted camera tracking be exploited to increase safety and situational awareness in any process involving heavy machinery and remote operation?

How will GPU-acceleration of the computer vision algorithm affect the system?

How can we reduce the latency of the live camera feed to make feedback control of the camera better?

How can we track an object across multiple cameras?

We will develop a solution to provide benefits to a human driller operating heavy machinery on an oil rig, but the solution is not limited to this application. Our end goal is to reduce the possibility of an undesired event, be it either damage to humans or expensive equipment.

1.1.2 Literature survey

As discussed by Sklet (2005), the concept of a safety barrier is not clearly defined and its meaning is ambiguous. With this in mind, we are looking to implement what **?** describes as a proactive safety barrier, also known as a frequency-reducing barrier, in other words a system to reduce the frequency of undesired events.

Semi-automated CCTV surveillance have been considered by **?** as a method of increasing the capacity of a human operator in a traditional human surveillance situation. The reliability for fully automated systems were not considered good enough for an operator to trust. The findings recommend providing feedback about system confidence and accuracy to the operator, which makes the automated component of such a system more 'visible' to its user. For the case of this thesis, the act of displaying visual cues overlaid on the CCTV images are considered. This would hopefully increase trust, and expose the automated component of our fully automated tracking system.

The computer vision algorithm that was implemented in the authors earlier work, as mentioned in work done by **?** and **?**, will be used to recognize the distinct symbol, hereafter called a glyph.

Our camera of choice, an AXIS Q6045 was selected both by its widespread use in the oil- and gas industry, and because this is what the author had at hand. It is a modern high-definition PTZ-camera produced by Axis Communications. Axis Communications have provided a white paper which provides a good overview of the various elements that increase latency in a live video stream. This document is available online. (**?**)

Work done by **?** involved methods to reduce the delays that are inherently found in digital video transmission systems and the control of these. Their research was done on AXIS Q6035 and AXIS Q6032 cameras. The author assumes these to be closely related to the AXIS Q6045, and their findings therefore useful for work done in this thesis. Findings include camera firmware being among the key factors for video delay, as the stock cameras have implemented an inefficient communication scheme, and there are other suggestions to reduce delay presented. For the scope of this thesis, we will have these delays in mind and build around them, as any firmware upgrades have to come from Axis themselves if any company would consider using them.

Tracking of objects across multiple cameras have been explored with most focus on overlapping camera views. Some work has also been done on non-overlapping camera views by **?**, where camera topology and path probabilities are learnt without any inter-camera calibration. As soon as one uses images from several sources, the timestamp for the images becomes crucial as a point of reference. Through the use of Parzen windows, the inter-camera space-time probabilities can be mapped. The method mentioned does require a learning phase.

1.1.3 What remains to be done?

As a summary of the literature survey, we see that much work has been done in the different fields, but not much have been found on combining the results of these into a solution that can provide modern, automated CCTV tracking of industrial processes involving heavy machinery in a way that retains the human operator in the center of the process.

We aim to break down the information silo and combine as much as possible, given constraints of time, into a proof of concept which can be the stepping stone to a commercial product.

1.2 Objectives

The objectives of the work done through this thesis consists of:

1. Implementation of a multi-source GPU-accelerated computer vision program that can control a CCTV camera and follow a symbol

2. Implementation of a proof of concept pipe-detection algorithm for fingerboards

1.3 Limitations

The limitations that relates to this study includes both technical challenges, environmental and operational conditions as well as oil politics in a broad sense, but focus is on the technical challenges for the sake of brevity.

As CCTV cameras have evolved, the video transfer method has gone through some changes to cater for higher resolution and more true representation of the world as observed. By this, analog signals have been replaced by digital signals. Analog transmission is known for being both robust, simple and near-instant, however they are prone to signal deterioration which may affect computer vision algorithms, and their flexibility of location is not as good as modern digital transmission. With digital transmission, usually over Ethernet, the cost of these systems have gone down, flexibility have gone up and resolution as well as control has improved, yet this have introduced new challenges. Data loss is a real possibility, increased latency through encoding and decoding of the video stream and a shared network highway puts more demand on the implementation. One serious limitation to the implementation of the system as presented, is therefore as a feedback system, the upper latency limit for which when the system becomes unstable.

The oil industry is considered a conservative one, as lost production time can lead to extreme expenses and questions regarding responsibility and passing on cost to service providers. Technology is also highly guarded, as the sheer amount of money that can be gained from saving a fraction of time, means that cooperation is not common in the industry, and transparency of systems and solutions may be less than ideal. The idea then, of making changes to what already works, is not easy to plant, and this also limits the available data set for the purpose of making robust systems.

Heterogeneous computing platforms are still considered to be in its infancy, and both CUDA as well as OpenCL is under active developement. Choosing one technology will lead to an exclusion of either benefits or available computing platforms. A limitation here is that the resulting speed and benefits of heterogeneous computing are not set in stone, and that there will always be improvements that can be done to make an otherwise unworkable system to a brilliant idea.

The software world progress quickly, and new solutions can suddenly become obsolete, requiring legacy maintenance and making in-house developed software seem much like building a tower of cards as the flux of developers who knows the system changes over time. This would make an externally maintained solution seem like a better idea, but sharing information to make the development work is not an easy task as each company protects its own interests.

It is considered a hard challenge therefore, to make a truly reliable system that works in the real world. Making a tabletop solution is not nearly enough to allow a big company to test this out in the real world.

Seeing past these limitations, however, is a world of possibilities, which this study tries to explore.

1.4 Approach

1.4.1 CCTV tracking system

The implementation of the CCTV tracking system will be based on the authors previous work and be written in C++ with heterogeneous support from a GPU to increase performance.

After this system has been built, it will be tested with and without GPU acceleration, to determine if the full solution becomes more stable and reliable.

A data-set will be analyzed to test the robustness of the computer vision algorithm and comment how snow, sun and other real-world factors affect the output.

It will also be tried to use multiple video sources, however due to the lack of several CCTV cameras, this will only partially be explored using a common webcamera, as a means of doing camera handover.

1.4.2 Pipe detection system

The construction of a proof of concept pipe detection system for fingerboards will be done in a rapid prototyping environment, to show that it is possible to increase control system awareness in existing infrastructure on the oil rig.

1.5 Structure of the report

All the software developed as a part of this project can be found at the authors personal repository at Github Skjefstad (2015), feel free to use this for future non-commercial work.

1.6 Structure of the DVD

The DVD contains a snapshot of the latest Github code repository as of the date of this report.

Chapter 2

Case study

2.1 Glyph tracking

The norm in the industry is that CCTV cameras are manually selected depending on needs, usually through using touch screens that display a set of four pictures at once. They are controllable through PTZ, but they are usually only moved to preset locations, by navigating the user interface.

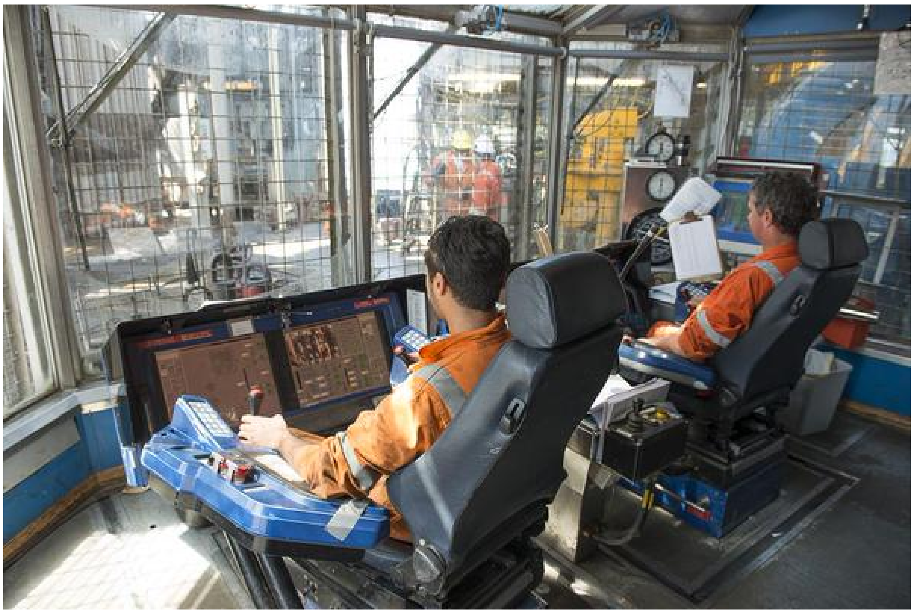

Figure 2.1: Modern driller cabin. Source:**?**

2.1.1 Design of the computer vision algorithm

2.1.2 Analysis of dataset for robustness in weather conditions

Based on work done in the project thesis, a Python program using OpenCV 2 was created. Its purpose was to analyse pictures captured over the span of a year, from the MHWirth test tower located at Dvergsnes, Kristiansand.

$$58h08min18.3secN8h04min17.6secE \tag{2.1}$$

The dataset was gathered using another Python program written and deployed around christmas 2014, which at 15-minute intervals, captured images from a handful of CCTV cameras in the tower and stored them on a local server.

Located at various locations in the fifty meters tall test tower, the idea was that weather impacts would be strong.

As the CCTV cameras have built-in low-light mode, a period during the night is considered too dark for the algorithm to detect the glyph.

The program that gathered the pictures was running without supervision. A result of this was that some gaps of several days in the dataset exists.

Initially, all the pictures were stored as lossless 24-bit RGB Portable Network Graphics. However, after a few months, the amount of data being gathered was filling up the server storage. It was then decided to continue the data gathering using JPEG, which would cut the required storage space close to $\frac{1}{5}$ of the space required with PNG-24

$$\frac{pictures}{camera} = 365days \cdot 24\frac{h}{day} \cdot 4\frac{pics}{h} = 35040 \tag{2.2}$$

2.2 Tubular detection in fingerboard

Drilling pipes and risers, commonly called tubulars, are used to drill deep into the earth.

These tubulars need to be stored in between their use in the drilling process. The most common method for short-term to mid-term storage is in groups of a few units in a machine called fingerboard, a part of the pipe handling equipment on a drilling rig.

The circularity of a circle can be described using the Heywood circularity factor. This factor can be used to narrow feature detection.

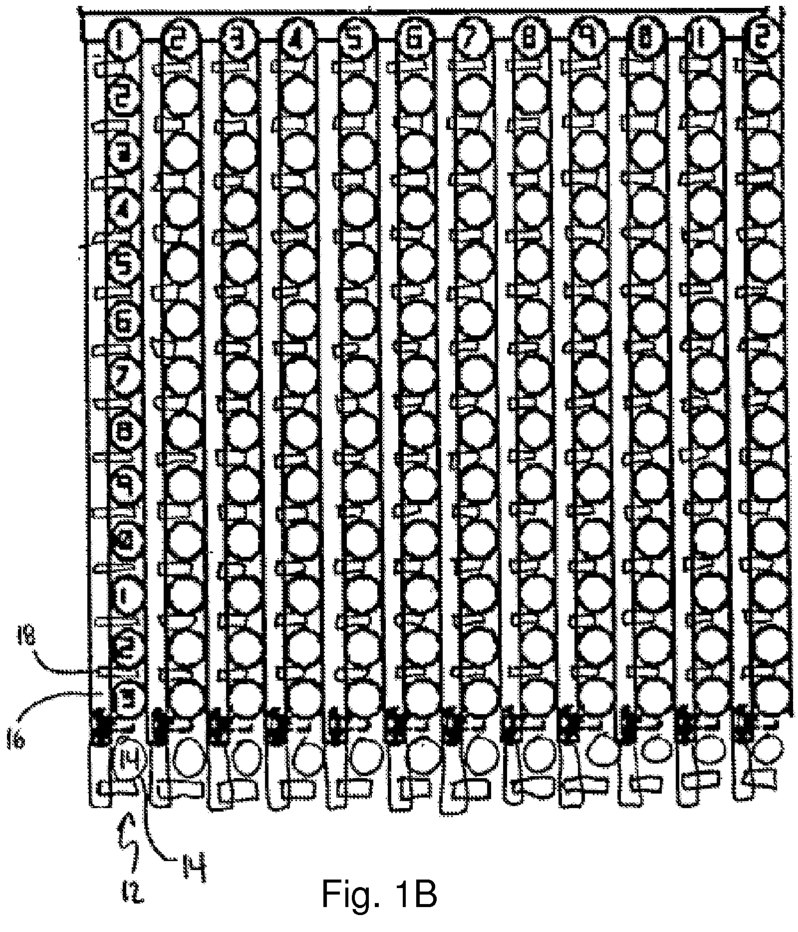

Fig. 1B

Figure 2.2: Diagram of a fingerboard with tubulars. Source:**?**

The first patent for a finger board was filed 1929 in the United States, and ever since, new patents that increase their capabilities have been filed. The latest ones use pneumatic cylinders to latch and secure tubulars in place, and a control system managing and tracking the pipes that should be in the fingerboard.

For more information regarding the history of the fingerboard, the reference list of a patent publicized February 2013 by **?** is suggested. The patent in question discuss a method to report finger position data to the control system. With this feature, the control system becomes aware of its fingers, but knowing if a tubular exists in a given cell requires a fail-free logging of the actions done by other machines.

The goal of this case study is to implement a proof of concept computer vision algorithm to detect tubulars.

Figure 2.3: Fingerboard without tubulars. Source:**?**

One possible solution involves using the Hough transform method to find ellipses in the picture, then analyze the region contained within to determine wheter the detected ellipse may outline a tubular. The algorithm implemented in OpenCV is based on a variant called the 2-1 Hough Transform by **?**. Partial occlusion of tubulars are a challenge.

Another more recent method of ellipse detection is presented by **?** based on sorted merging. This algorithm is not yet implemented in OpenCV.

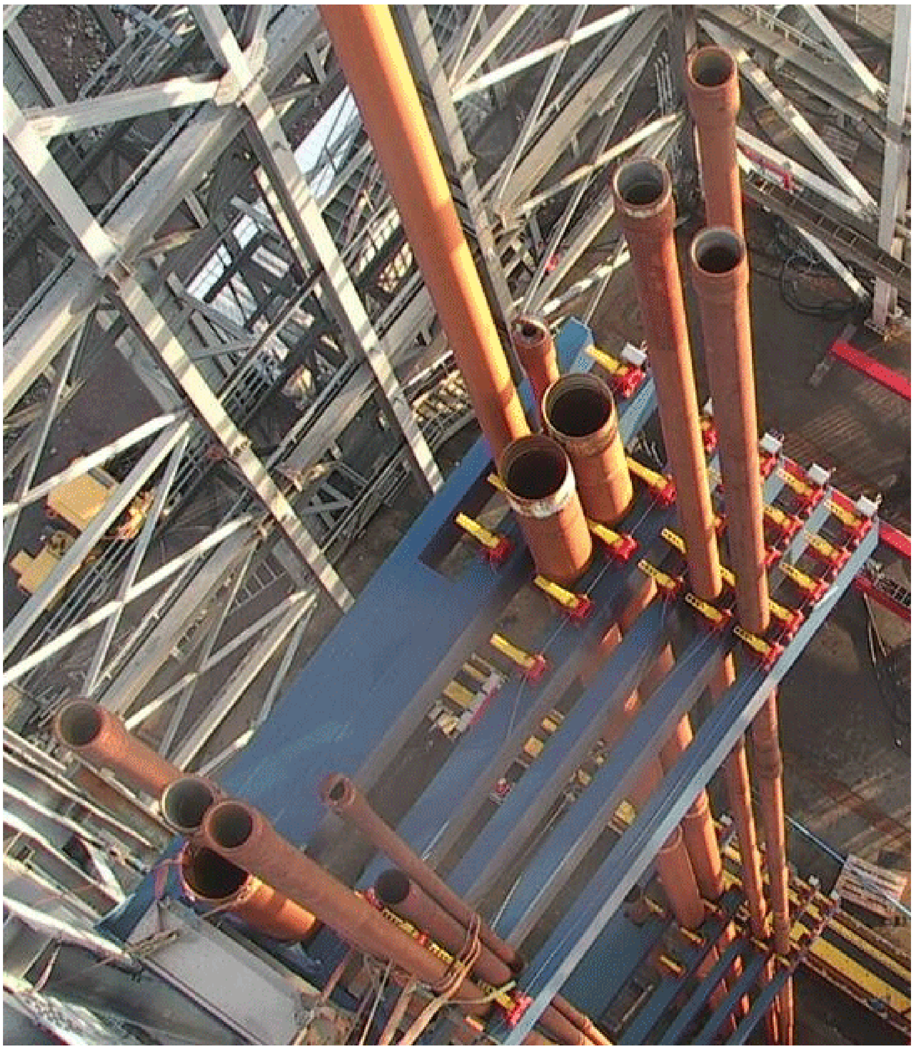

Figure 2.4: Fingerboard with tubulars. Source: Own work (MHWirth AS)

Chapter 3

Discussion and results

3.1 Discussion

3.2 Recommendations for future work

Appendix A

Acronyms

API Application Programming Interface

AR Augmented Reality

CD Compact Disc

CGI Common Gateway Interface

CV Computer Vision

IADC International Association Of Drilling Contractors

MJPEG Motion JPEG

OPC UA Open Platform Communications Unified Architecture

PTZ Pan Tilt Zoom

SOAP Simple Object Access protocol

SSL Secure Socket Layer

WITSML Wellsite Information Transfer Standard Markup Language

Appendix B

Axis Q6045 Camera

Introduction The Axis Q6045 PTZ Dome Network Camera is a high-performance IP-camera which can deliver its video stream over a computer network. It is delivered in both indoor and outdoor models, and have a 20x optical zoom. Endless 360 degree pan and a high resolution.

Outdoor edition The outdoor edition also supports electronic image stabilization to reduce effects of camera vibration. It also includes automatic defogging capabilities using digital filters.

Camera

Image Sensor	1/3" progressive scan CMOS
Lens	f=4.45 - 89 mm, F1.6 - 2.9, autofocus Horizontal angle of view: 62.98 deg - 3.49 deg
Min. illumination	Color: 0.6 lux at 30 IRE F1.6 B/W: 0.04 lux at 30 IRE F1.6
Shutter time	1/33000 s to 1/3 s (50 Hz), 1/33000 s to 1/4 s (60 Hz)
Pan/Tilt/Zoom	Pan: 360 degrees endless, 0.05 - 450 degree/s Tilt: 180 degree, 0.05 - 450 degree/s 20x optical zoom and 12x digital zoom, total 240x zoom

Video

Video compression	H.264 Main and Baseline Profiles Motion JPEG
Resolutions	1920x1080 to 320x180
Frame rate	H.264: Up to 25/30 fps (50/60 Hz) in all resolutions Motion JPEG: Up to 25/30 fps (50/60 Hz) in all resolutions

System integration

API	VAPIX AVHS ONVIF Profile S

General

Operating conditions	0 degree to 50 degree Celsius Humidity 10-85% RH (non-condensing)

Appendix C

Python Profiling

Introduction It is well known that Python as an interpreted language is considerably slower than both compiled C and C++ code. However, the Python module that provides our OpenCV-interface is not much more than a wrapper for the C++ core in OpenCV.

As such, we use Python for quickly developing and structuring the program, while the actual heavy lifting is done by compiled C++ code. Our design goal is then to write efficient Python and using quick algorithms where possible, leveraging the built-in functionality in the C++ core of OpenCV.

In order to investigate both time and memory used for our solution, a few good methods exists.

line_profiler A module for Python that provides line by line profiling capabilities is the line_profiler program. The actual time spent inside a function is outputted.

We can install the module through pip.

```
pip install line_profiler
```

After adding the keyword profile ahead of each function we would like to profile, we run the profiler as kernprof.py through Python.

```
python kernprof.py -l -v program.py
```

By running line_profiler on the initial proof of concept, we will get the time spent on each function we used @profile in front.

```
Total time: 0.475104 s
File: jsg.py
Function: find_potential_glyphs at line 90

Total time: 0.116176 s
File: main.py
```

```
Function: init_capture_device at line 18

Total time: 2.35552 s
File: main.py
Function: grab_frame at line 29

Total time: 5.41517 s
File: ptz.py
Function: execute_command at line 25
```

This tells us that through the running time of the program, most time was spent inside ptz.py.

We can for example, see the following output for code from ptz.py, the file that sends ptz commands to the camera.

We see that the function execute_command contains, at line 30, a http request which consumes 99.7% of the time used inside the function. Out of this, it is suggested that the overhead by using the CGI element for doing PTZ is a function of network latency.

memory_profiler Another important resource is the amount of memory being used by the program. Using this, we can discover memory leaks and further optimize our code. We can install the module through pip. We are also adviced to install psutil, as it improves performance of memory_profiler.

```
pip install memory_profiler
pip install psutil
```

Appendix D

Compiling OpenCV 3.0 for OS X

OpenCV 3.0 gold release arrived in June 2015. Being this new, the resources to properly compile OpenCV 3.0 for OS X are not easy to come by. A few blogs have a partial solution for setting up the software, so this appendix will not contain a complete solution itself, but mention the most important points when building OpenCV 3.0 for OS X.

Homebrew is utilized to settle most build-dependencies, with good success.

When building opencv, we also would be interested in building opencv_contrib modules.

```
cd ~
git clone https://github.com/Itseez/opencv.git
git clone https://github.com/Itseez/opencv_contrib.git
```

We use the command-line interface for cmake, but it should also be possible to use the GUI version. Notice that we explicitly disable CUDA in this case, and enable OpenCL. We also build with support for Python 2.7, as the author considers this a good way to rapidly test functionality.

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D PYTHON2_PACKAGES_PATH=~/.virtualenvs/cv/lib/python2.7/site-packages \
-D PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Versions/2
-D PYTHON2_INCLUDE_DIR=/usr/local/Frameworks/Python.framework/Headers \
-D INSTALL_C_EXAMPLES=ON \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D BUILD_EXAMPLES=ON \
-D WITH_OPENCL=ON \
-D WITH_CUDA=OFF \
-D WITH_FFMPEG=ON \
-D BUILD_DOCS=ON \
```

```
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules ..
```

We have not installed clBLAS nor clFFT, two libraries that are considered advantageous when using OpenCL. These libraries will have to be compiled with their own specific dependencies.

Bibliography

S. Sklet, “Safety barriers on oil and gas platforms,” Ph.D. dissertation, Norwegian University of Science and Technology, 2005.

J. Skjefstad. (2015) Github. [Online]. Available: www.google.com