IT 09 021

Examensarbete 30 hp
Juni 2009

Evaluation and improvement of image acquisition and processing methods for the BioNanoLab

Magnus Elgh

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Evaluation and improvement of image acquisition and processing methods for the BioNanoLab

Magnus Elgh

BioNanoLab is a project aimed towards developing a prototype for a system capable of fast and sensitive detection of biological warfare agents. One of the important parts in the project is image acquisition and processing. My task was to evaluate and improve these methods. The first step was to choose a machine vision library to use and then to write a program for acquiring images from line scan cameras to disc trough frame grabbers and to write code for processing the images.

The two different machine vision libraries I tested were Sapera Essential and Common vision blox (CVB). Sapera Essential is a machine vision library from Dalsa. It is platform dependent as it only works for Dalsa hardware. CVB on the other hand is a hardware independent machine vision library from Stemmer imaging.

The run times for the two libraries were almost the same but Sapera Essential was chosen as the winner because we wanted the more low level control which Sapera offers, as it is written specifically for the hardware we used. Another reason was that the blob counting for CVB started to miscount some times.

Handledare: Johan Stenberg
Ämnesgranskare: Bo Nordin
Examinator: Anders Jansson
IT 09 021
Tryckt av: Reprocentralen ITC

Table of contents

Introduction

The BioNanoLab project

BioNanoLab is a project within the Swedish Defence Nanotechnology Programme. BioNanoLab is run by Uppsala university (Department of genetics and pathology), Swedish Defence Research Agency (FOI) and Q-linea AB. The project is aimed towards developing a prototype for a system capable of fast and sensitive detection of biological warfare agents. This system consists of laboratory methods for collection and purification of samples, recognition of target molecules (proteins and nucleic acids) and signal generation using molecular biotechnology techniques developed at the department of genetics and pathology.

The laboratory procedures report the presence of target molecules as a signal, consisting of RCPs, rolling-circle-amplification products. These RCPs are long DNA strands that are created through rolling circle amplification and labeled with fluorescent reporter molecules. In solution, RCPs collapse to spherical coils of DNA. The signal generation procedure will be described in further detail below.

The work presented in this thesis has been performed at Q-linea AB in Uppsala. Within the BioNanoLab project, Q-lineas task is to develop an instrument for detection of the amount of RCPs that have been created in a sample. This is done by pumping the sample trough a detection flow cell that is illuminated by a laser line across the flow direction. When RCPs pass the line they will fluoresce, emit light, which is detected using semi-confocal optics and CCD line detectors. Detection is possible in different wavelength channels in order to be able to differentiate RPCs labeled with different fluorophores.

What do we want to detect

The objects that the system is to detect can be divided into two parts. First, there are the traditional meaning of biological warfare agents consisting of viruses, bacteria and bacterial spores and the second part consisting of toxins. Poisons are generally considered agents of chemical warfare, but toxins, poisons that are biologically produced [1], are considered agents of biological warfare. The BioNanoLab project aims to enable detection of viruses, bacteria, bacterial spores and protein-based toxins.

Traditional biological warfare agents: This category includes possible agents of biowarfare or bioterrorism, such as the Anthrax-causing bacterium, *Bacillus anthracis*, the Ebola virus and the cholera-causing bacterium, *Vibrio cholerae*. There are new microorganisms and old ones that have become dangerous for us humans again through modern breakthroughs in genetics. One area is resistances against antibiotics [2].

Traditional biological warfare agents may be divided into three different categories that the system is going to detect.

- Viruses – Detection by both proteins and DNA.

- Bacteria – Detection by its DNA.
- Spores – Detection by proteins on the spore's surface.

Toxins (protein-based): Protein-based toxins include for example ricin and botulinum toxin. Ricin is extracted from the castor bean. A lethal dose for humans is around 500 micrograms, about the size of a grain of salt, if the exposure is from injection or inhalation [3]. Botulinum toxin is the most toxic protein known. Even so commercially used products are made of botulinum toxin for cosmetic treatments (in very small concentrations). One of those products is Botox Cosmetic [4].

A need for the instrument

The instrument that is created through the project could be used both by the military and by the general public. But the primary purpose of this project is for defence applications. So why do the Swedish Armed Forces need an instrument for detecting biological warfare agents? What threats are imposed on Sweden today?

The assessment of the Swedish security service, Säkerhetspolisen, is that there are no present threats directly targeted against Sweden and no deliberate attacks with biological warfare agents have been proven today in Sweden. But there is still an elevated risk for terror attacks against other countries' interests in Sweden. There is also a risk for Swedes and Swedish interests abroad to be attacked (for example in international operations). There is also a possibility that Sweden might be affected by attacks on targets geographically close to Sweden [5].

A recent case of bioterrorism that has been given a lot of attention in the media was the letters with anthrax spores that were mailed to politicians and several news media offices in the USA. The end result was 5 deaths, 17 others infected and great economic costs [6].

There have also been cases of use of biological warfare way back in history (even if it was not called that at that time). Back in the 14th century there were cases of armies catapulting in dead humans infected with the bubonic plague, the Black Death, into towns that they tried to capture. The desired effect was to spread terror and demoralize the defending troops [7].

First step in counting blobs – Biology

The first step of the detection process is getting our sample. The sampling is done in water or air and after that the DNA/protein is extracted from the sample. The next step is the use of padlock probes. A padlock probe is a synthetic linear single stranded DNA molecule that is made circular when it comes into contact with its target DNA sequence (see figure 1). After that, a ligase enzyme is used to connect the two ends together, forming a closed circle. The created circle has a size of 20 nm [8][9][10]. For protein detection the steps are quite similar but the recognition of the target is not performed by DNA base-pairing but rather by two antibodies that bind to the target protein. The antibodies are equipped with short single stranded DNA strands. An additional DNA probe is then circularized by the two DNA strands from the antibodies and closed with a ligase [11][12]. In both cases, the intermediate product is a circular, single stranded DNA molecule, between 50 and 120 nucleotides in length.

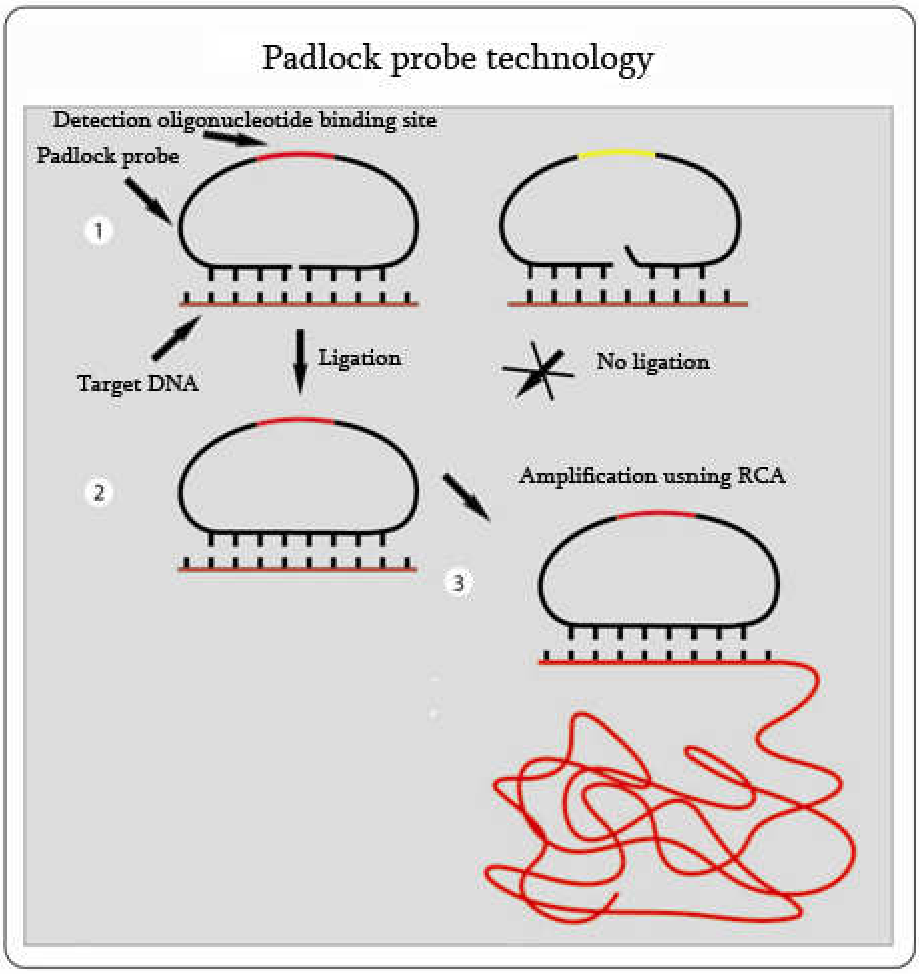


Figure 1 – Padlock probe technology (DNA target).

With a DNA polymerase enzyme a RCA can be performed, creating a linear sequence containing complementary copies of the padlock probe. The number of copies is determined by the time the process is given before the polymerase enzyme is heat inactivated the. During the duplication, the RCA product collapses into a roughly spherical random coil of DNA/protein with a size of approximately 1 µm in diameter.

The RCPs, also called blobs, are thereafter labeled with fluorescent molecules. The fluorescent molecules, or fluorophores, are attached to detection oligonucleotides. The detection oligonucleotides are designed to only bind to a specific DNA binding site. That detection site is present in the padlock probe and therefore copied multiple times in the RCA. The detection oligonucleotides attach to the target positions on the blob. When illuminated with light of the correct wavelength, the blobs will fluoresce and be detectable above the background and may be counted. The RCA step is required for one major reason. It makes it possible to distinguish between free fluorophores in the solution and the ones on the blobs because these will have a higher local concentration of fluorophores within the blobs than in the surrounding solution.

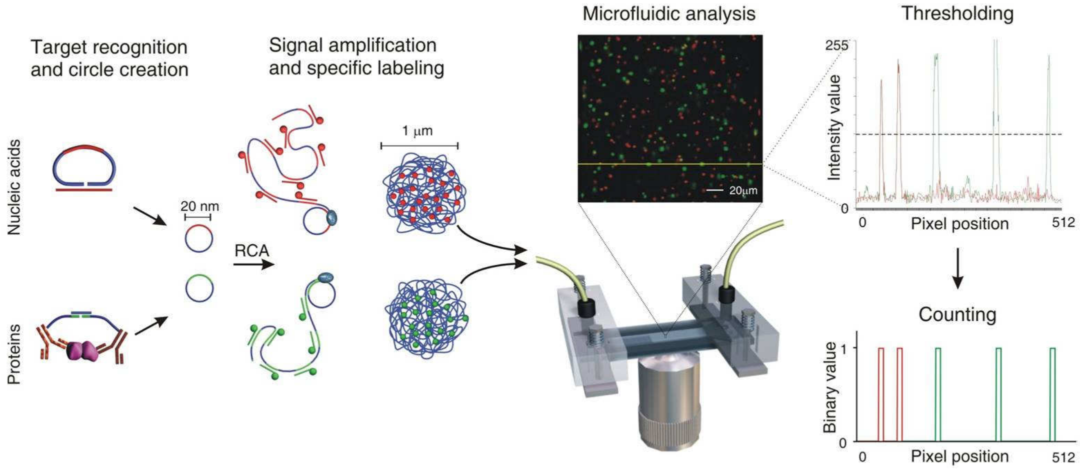


Figure 2 – From DNA/protein to counting blobs. Reprinted by permission from Macmillan Publishers Ltd: NATURE METHODS (Jarvius et al), copyright (2006)[11].

The solution with the blobs is pumped through a micro channel. The micro channel is made in a plastic chip and has dimensions 200 x 40 µm (width by height). When flowing through the channel the blobs are illuminated by laser light (to get an accurate wavelength and focused light). At the same time, image data is collected from the micro channel and analyzed to count the blobs that fluoresce. If the image intensity in a pixel is over a predefined level, this is considered a detected signal. More about the different hardware used for the counting is described below. See figure 2 for the big picture from DNA/protein to the blob counting.

Yesterday's instrument for counting blobs

Before the development of a new detector for biological warfare agents started, a confocal microscope was used to count blobs in the BioNanoLab project. It is still used today but to a lesser extent and as a tool in the development process of the new instrument.

The big difference between confocal microscopes and traditional fluorescence microscopes is that the confocal microscope uses only point light to illuminate the specimen whereas the traditional microscope illuminates the whole specimen all the time (see figure 3 and 4 for the different schematics of the microscopes). One of the benefits of this is that the confocal microscope stops out-of-focus light rays from being detected and better image quality is achieved this way. Two pinholes are used to get point illumination (one at the laser and one at the detector). There are three different types of confocal microscopes. The one used in the BioNanoLab project is a confocal laser scanning microscope.

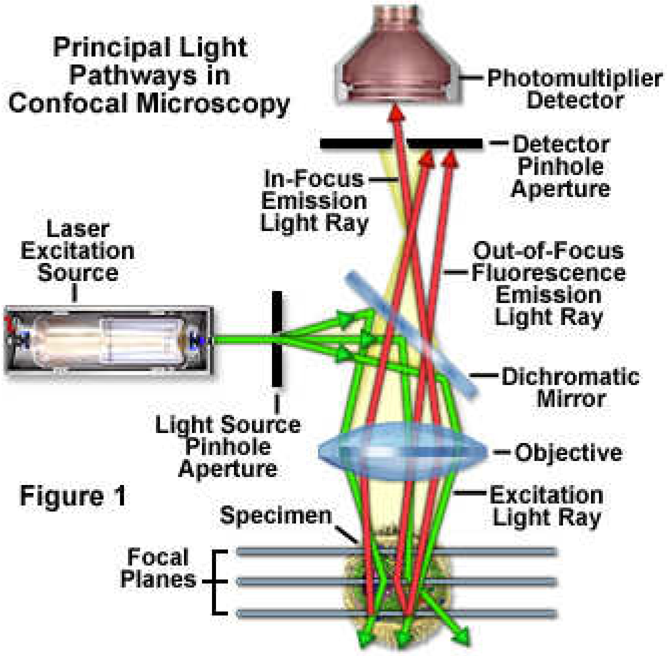

Principal Light Pathways in Confocal Microscopy
Photomultiplier Detector
Detector Pinhole Aperture
In-Focus Emission Light Ray
Laser Excitation Source
Out-of-Focus Fluorescence Emission Light Ray
Dichromatic Mirror
Light Source Pinhole Aperture
Objective
Figure 1
Excitation Light Ray
Specimen
Focal Planes

Figure 3 – A simple schematic representation of a confocal microscope. From http://www.microscopyu.com/articles/confocal/confocalintrobasics.html

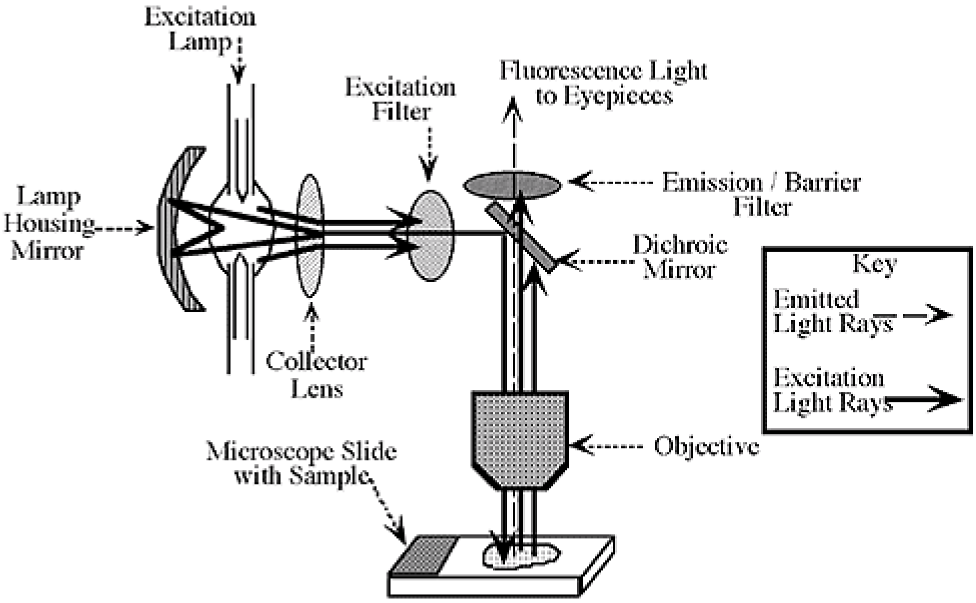


Figure 4 – Traditional fluorescence microscope. From http://international.abbottmolecular.com/DiagramoftheFluorescenceMicroscope_8843.aspx

Because the confocal microscope illuminates a point at a time, the laser beam must be mirrored to scan in x and y coordinates over the whole specimen. Also the z coordinate of the specimen, that is illuminated, may be changed by using different focal planes. That way a three dimensional volume of the specimen could be created by changing the active focal plane, something that is not possible in a traditional fluorescence microscope. Using point-to-point illumination gives that each point on the specimen gets less light and because of that emits less photons compared to when the whole specimen is illuminated. Longer time is therefore needed to get enough photons to avoid building a noisy image. To compensate for that, and to save time, a high intensity light source is used (a laser) [13].

The reason a confocal microscope is used in this project instead of an ordinary fluorescence microscope is because of the ability to choose a focal plane. Without that all of the micro channels height emitting light would be detected and the result would be a lot of overlapping blobs.

Today’s/Tomorrow’s instrument for counting blobs

The instrument that is going to be used in the final prototype is different from the confocal microscope. It uses lasers just like the confocal microscope, but instead of using pinholes to create point-to-point light, a beam expander and beam shaper is used to create lines of light (see figure 5). This is the main reason for creating the new instrument. By using a line instead of a point and using the flow in the micro channel to scan through the y axis a speed up of the width of line is gained. The price of spreading the point laser to a line is less intensity of the light beam. This is compensated by a bit longer exposure time and a laser with higher intensity.

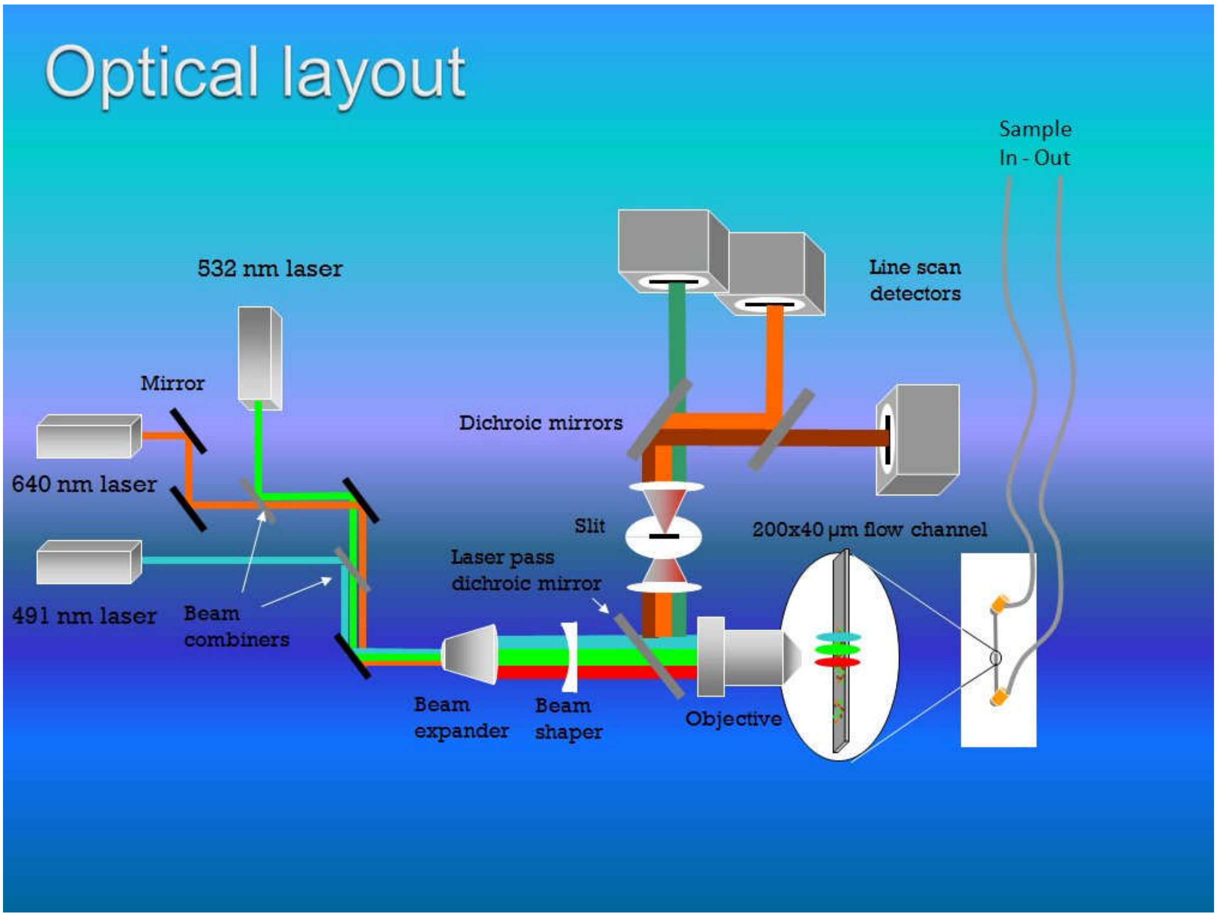


Figure 5 – Optical layout for the new instrument.

The cameras used in the instrument, Dalsa Spyder 3 (camera link interface), are CCD line scan cameras and have a maximal line rate of 60 kHz. That is 60000 exposures per second (60000 lines / second).

The steps of the instrument from the lasers to the line scan detectors are:

1. 3 different lasers (same amount as the line scan detectors).
2. The laser beams are arranged to lay on top of each other by mirrors and beam combiners.
3. The beams are transformed from point light to lines by a beam expander and a beam shaper.
4. The line lights go through a dichroic mirror and the objective and illuminate the micro channel.
5. At the same time is the blob solution pumped through the micro channel.
6. The illuminated blobs emit light (actually the fluorophores).
7. The emitting light goes back trough the objective and reflects on the dichroic mirror.
8. An external slit may be used on the light to ensure not to get out of focus light. The CCD chip on the line detectors may be used to get the same result.
9. Some additional dichroic mirrors are used to separate the three lines of light.
10. Each of the three line scan detectors gets a separate line of light.

That is the optical part of the instrument. The image data is then acquired from the cameras via the frame grabbers, Dalsa X64-CL iPro, and on to RAM memory or the hard drive in the computer (see figure 6).

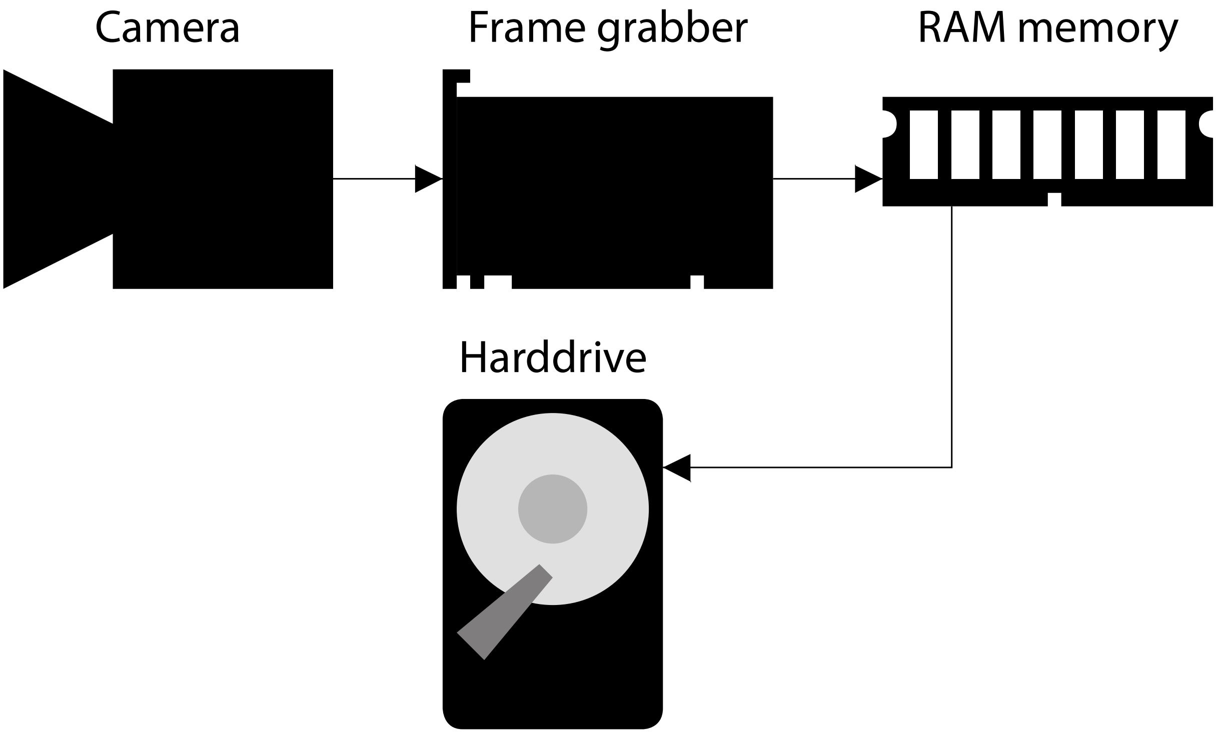


Figure 6 – From the camera to the hard drive.

More about the cameras and other image acquisition hardware will be described below.

Problem description

The development of the new prototype of the detection instrument requires new methods for image acquisition and processing. My thesis is about evaluating and improving these methods. The two main parts was first to chose what machine vision library to use for the image acquisition and processing and then to write programs that used the winning library.

Acquiring and analyzing image data

The first step is the acquisition where the image data is gathered from the cameras. Today the number of cameras in the instrument is three, since you need one camera for every laser channel (see figure 5). A problem that arises when you have more than one camera is keeping them synchronized (and the acquired image data). To solve this problem two signals are used. To synchronize the line scan cameras a line rate signal is used. This signal determines when, and how often, the lines should be exposed in the cameras. The second signal is the frame rate signal that determines when, and how often, a new frame is getting acquired from the cameras trough the frame grabbers to memory. To get synchronized image data both these signals have to be the same for the cameras and frame grabbers. Synchronizing in software was not a major priority in this thesis so an external pulse generator was used instead to create the signal for the frame and line triggers.

Which software to be used for the acquisition stood between two different libraries. CVB, from the German company Stemmer imaging, and Sapera Essential form the Canadian company Dalsa. The goal of this project is to have the acquired data online (only storing some images to disc as snapshots for later inspection). But my thesis is restricted toward offline but with future thoughts for online image acquisition and processing because of the short time (an offline version is also a good tool for the development of the online version of the software).

In this project offline and online represents the intermediate step of storing the acquired images from the cameras on disc for later analysis, or not doing so. Offline analysis is when the images are stored in files on the hard drive and later read in to memory for analysis. Online is when image data are kept in memory and analyzed in real-time (when acquired to memory).

The second step is the analysis of the collected image data. It was decided not to develop any new code for the blob analysis of the image data and instead test two commercially available libraries (the same as above, CVB and Sapera Essential). One of the problems to tackle here was choosing which one of the libraries to use.

Having chosen one of the libraries, using the right parameters for the blob analysis will be essential. So some kind of parameter optimization will be needed. The optimization will be done with different instrument settings like line rate and pump speed. Each run of the parameter optimization, one instrument setting, will be done over a dilution series; a range of different concentrations. The desired result on each run will be the parameters that give the most linear response (that is of the curve with the concentration on the x axis and the blob count on the y axis). But if all blob counts are 0, for every concentration in the curve, that will give perfect linearity (that is when the line goes through all the points exactly). So the slope of the curve has to be taken into consideration too (a slope of 0 is not wanted). The last thing to take in to consideration is the blob count when the concentration should be 0. A value close to 0 is desired here. The number of blobs detected at 0 concentration represents unwanted background. These three important things will be incorporated in two different formulas (see formula 1 and 2 on page 22).

Theory

Fluorescence

Fluorescence is the ability for a molecule to absorb the energy of a photon and then trigger the emission of a new photon but with a longer wavelength (less energetic) [14]. Fluorophore are organic molecules with this ability. What specific wavelength a molecule absorbs depends on the fluorophore likewise the specific wavelength the photon is emitted in. Some fluorophores requires ultra violet light for fluorescence to happen.

CCD – cameras

The cameras are used to collect image data from the emitted light from the blobs when illuminated by the lasers.

CCD stands for charge-coupled device. A CCD camera uses CCD chips to turn light into electric charges (by photoelectric light sensors). The number of chips it uses depends on if it is a monochrome or color camera. A monochrome camera uses only one chip and a color camera uses 3 (one each for the three channels RGB – red, green and blue) [15][16].

The CCD chip is an array of analog shift registers that transports the analog signals (electric charges) trough successive stages (capacitors), all controlled by a clock signal. The size of the chip depends on if it is an area scan or a line scan camera. An area scan camera has an array size of x width and y height where the aspect ratio x:y often is 4:3. Line scan cameras often have an array size of 512 in width and 1 in height, although it may be as much as a width of 8192 (still at a height close to 1). After that the chip gets very long and the box the chip is mounted in gets really big.

Line scan cameras are used in the instrument. A line scan camera is one that builds the total images by pasting together one line at a time. How long each line is exposed depends on the cameras line rate. That is how many lines the camera acquires per second. A line rate of 60000 lines per seconds gives an exposure time of 1/60000 ≈ 17 µs.

Frame grabber

The frame grabber’s task is to capture image data from the cameras and pass it through to RAM memory (see figure 5). The name comes from those early frame grabbers who only had enough memory to grab and store a single digitized video frame [15][17]. The source for the capturing, the camera’s interface, may be either an analog video signal or a digital video stream. Analog video signals requires a analog frame grabber and digital video streams requires a digital frame grabber, but in both cases is the final result a digital still frame or a sequence of them.

The frame grabbers used in the machine are digital using a Camera Link interface. The maximum bandwidth of the Camera Link interface is 2.04 Gbit/s (255 MB/s).

How fast the images are saved is determined by the frame rate (frames/second). The frame rate is linked to the camera and its line rate. The frame rate cannot be too high compared to the image height and the line rate. Otherwise parts of the frame will not be collected in time.

Thresholding

Thresholding is used in the blob analysis. The process of thresholding an image, here by intensity values, is a way of segmenting it [15]. The simplest case of thresholding is to have one threshold like this

$$g(x,y) = \begin{cases} 1\ if\ f(x,y) > T \\ 0\ if\ f(x,y) \leq T \end{cases}$$

g() and f() are the pixel intensities at a specific pixel in an image and x and y are a pixel coordinate. Hence g(x, y) is the thresholded image, f(x, y) is the original image and T is the threshold.

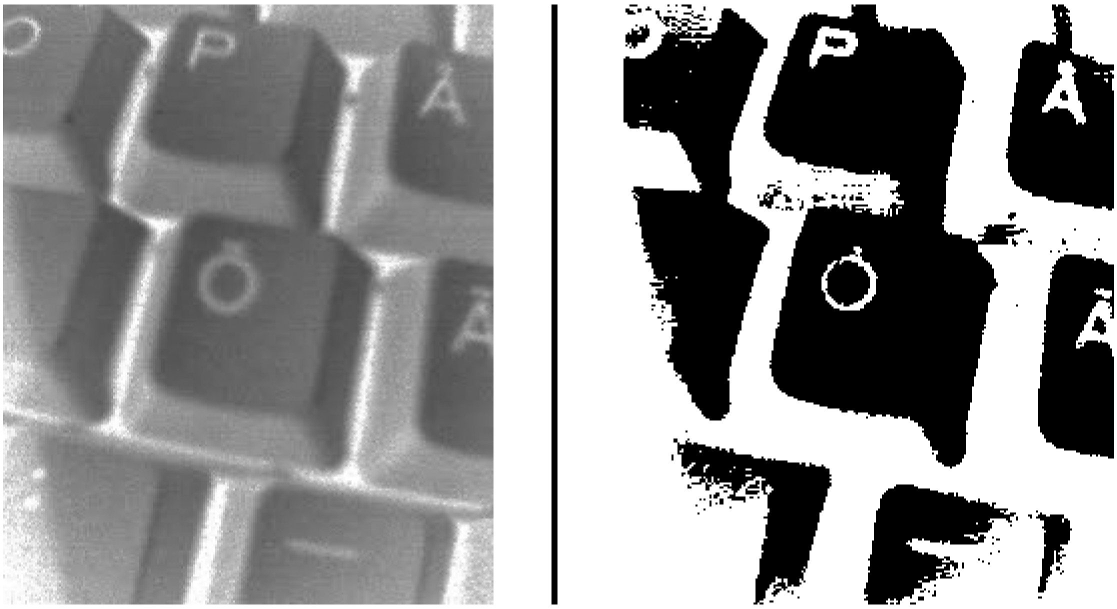


Figure 7 - f(x, y) to the left and g(x, y), the one above with T = 150, to the right.

This way you get a binarized image (intensity values of 0 and 1 only, see figure 7). But you may use many different thresholds and map to different values than just 0 and 1. That depends on your needs.

Using only one fixed set of thresholds on the whole image is called global thresholding. You may also use local thresholding. Then you divide the image in sub images and use different thresholds on the partial images. Local thresholding is a good choice if the image is illuminated unevenly. Global thresholding is easier to use/implement.

Pixel connectivity

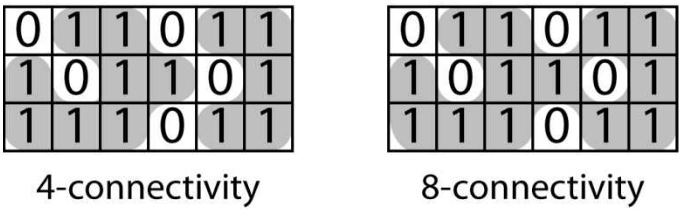
Pixel connectivity is also a component of the blob analysis. Pixel connectivity refers to if two pixels are connected or not. Before we look at that lets look at a pixel's neighbors. A pixel p with the coordinate (x, y) has two horizontal and two vertical neighbors with coordinates

(x + 1, y), (x - 1, y), (x, y +1), (x, y - 1)

This set of pixels is called the 4-neighbors of p and is written $N_4(p)$. We also have the 8-neighbors of p denoted by $N_8(p)$. This is the set $N_4(p)$ with the four diagonal pixels to p added to the set

(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)

Then if we have a set V of intensity values (V = {1} in a binary image). Two pixels, p and q, are then 4-connected if both have a intensity value in the set V and q is in the set $N_4(p)$. In a similar fashion are they 8-connected if both have a intensity value in the set V and q is in the set $N_8(p)$ [15][18].



0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	0	1	1

0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	0	1	1

Figure 8 - 4-connectivity contra 8-connectivity

See figure 8 for an example between 4-connectivity and 8-connectivity. When the ones are 4-connected there are two blobs and when 8-connected there's only one.

Blob analysis

Blob analysis is a key component in the system. Its major role is to find and count the blobs.

A blob in image processing is defined as a set of connected pixels. How the pixels are connected depends on the pixel connectivity. Blob analysis is then the process of identifying the blobs, in an image, and further measurements on the blobs like area, width, count and more. The first step in blob analysis is to divide the image into background and foreground. This is usually done by binary thresholding or some other binarization method. But sometimes when the image is illuminated unevenly or you got some blobs with the same pixels intensity as the background some more advanced segmentation method have to be written. Only the set of foreground pixels makes up the total set of all the blobs where the blob pixel sets are disjoint. The background set is just there as separation between the blobs [15][19].

R^2 and linear regression

R^2 and linear regression is used to calculate the linearity and slope of the curve while doing the parameter optimization.

There is no consensus about the exact definition of *R*^{2}. Only in the case of linear regression are all definitions equivalent. In this case, *R*^{2} is simply the square of a correlation coefficient (the value of *R*^{2} represents the linearity of a curve) [20]. It is that definition of R^2 that have been used in the thesis and the *R*^{2} value is equal to 1 when perfect linearity is reached (when all the points lay on the line).

Linear regression works by using least square error to get a straight line that is the best fit for some points (that is the sum of the square errors from the points to the line is as small as possible).

R^2 is calculated on the set of data, (x_i, y_i), with n data points.

$$R^2 = r^2$$

$$r = \frac{n \sum_{i=0}^{n-1}(x_i y_i) \sum_{i=0}^{n-1}(x_i) \sum_{i=0}^{n-1}(y_i)}{\sqrt{[n \sum_{i=0}^{n-1}(x_i^2) - (\sum_{i=0}^{n-1}(x_i))^2][n \sum_{i=0}^{n-1}(y_i^2) - (\sum_{i=0}^{n-1} y_i)^2]}} \; (\boldsymbol{equation\ 1})$$

I have also used the definitions, in linear regression, for calculating k and m values on the same data set as above [21].

$$y = kx + m \; (\boldsymbol{equation\ 2})$$

$$k = \frac{n \sum_{i=0}^{n-1}(x_i y_i) - \sum_{i=0}^{n-1}(x_i) \sum_{i=0}^{n-1}(y_i)}{n \sum_{i=0}^{n-1}(x_i^2) - (\sum_{i=0}^{n-1}(x_i))^2} \; (\boldsymbol{equation\ 3})$$

$$m = \frac{\sum_{i=0}^{n-1}(y_i) - k \sum_{i=0}^{n-1}(x_i)}{n} \; (\boldsymbol{equation\ 4})$$

Timing in Windows

Timing in Windows may be done in several ways. The big difference between the different ways is their precessions. Here are some ways to do that from lower to higher precision [22].

time() and _time64(): Classic old C style function for getting the time elapsed, in seconds, since midnight, January 1, 1970. For this application this functions precession is too low. Other problems with time() is that it will not work anymore after 19:14:07, January 18, 2038 when the counter is full (not a problem for my thesis tough). But if you want to use time() anyway and that is not enough. Then you have _time64() which is a 64 bit version of time() and that function will work until 23:59:59, December 31, 3000.

GetTickCount() and GetTickCount64(): A function that returns the number of milliseconds that have elapsed since the system was started. The function is included in the Win32 API. It has an resolution that is 10 ms compared to time() 1 second. GetTickCount() returns a DWORD, which is 32-bits wide. So the counter for the function will be reset to zero after about 50 days, as my little calculation below shows.

$$2^{32} = 4294967296$$

$$\frac{4294967296}{1000 * 60 * 60 * 24} \approx 49.7\ days$$

If one uses GetTickCount64() instead, which has the same precision, the counter will be reset to zero first after about 600 million years (the counter is 64 bit instead of 32 bit). That should be more than enough, if you do not need better precision.

QueryPerformanceCounter() and QueryPerformanceFrequency(): If you need even better precision then this is a good choice. While time()/_time64() has a precision of seconds, GetTickCount()/GetTickCount64() of milliseconds, this function has a precision close to microseconds. The actual resolution depends on QueryPerformanceFrequency() for the hardware used. QueryPerformanceCounter() returns the current value, 64 bit integer, of the high-resolution counter and QueryPerformanceFrequency() returns the frequency of that counter.

Methods

CVB (Common vision blox)

Manufacturer: Stemmer imaging, Germany

Version: 10

A generic library that supports different kinds of hardware such as cameras, frame grabbers and other image acquisition hardware. It has both low level code for acquisition and other operations concerning the hardware and high level code for different image analysis like blob analysis, edge detection and others [23].

The benefit of using CVB is that you may write hardware independent code witch make it easy to change hardware without having to rewrite a lot of code.

Sapera Essential

Manufacturer: Dalsa, Canada

Version: Edition 2008-07

A more specific library from Dalsa that only supports their own hardware. Sapera Essential is both Sapera LT, that is a library for low level control of your hardware, and Sapera Processing that is a higher level library for things like blob analysis, bar code reading and other problems in the image analysis area [24].

The benefit of using Sapera Essential, when you have Dalsa cameras and frame grabbers, is that you get a tighter coupled interface for this specific hardware.

Image acquisition

The first step in writing software for acquiring and saving the images to disc was to write an easy to use graphical user interface (GUI). As the application was written in C++, in Visual Studio 2008, for Windows, the Win32 API was used because of the well integrated support of the interface in the developing studio. Besides acquiring images, the software also creates a folder structure for the subsequent blob parameter optimization (see figure 9). There is one root folder at the top that is the same folder as the application is run in. The images folder at the root saves all the data created by the software. The first level in the image folder is the Hardware folder. The difference between the folders here is what camera and frame grabber setups that have been used. Each folder has a text file with the information about the folder it is in (as all the folders below in the structure). The hardware folders also contain subfolders with different hardware settings, for example pump speed for the pump in the instrument and line rates for the cameras. The next level is the experiment folders, where information about what used fluorophores is stored. The last level is the concentration folders where the blob concentration used is stored. Each concentration folder stores the acquired images and the information text file.

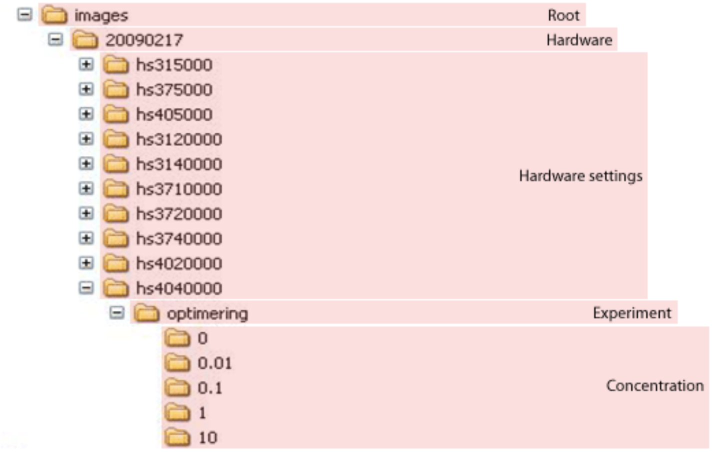


Figure 9 – Folder structure.

Here Sapera Essentials were chosen instead of CVB because the cameras and frame grabbers are from Dalsa and CVB uses only a subset of Dalsas library Sapera LT for low level control.

The biggest problem when acquiring images from the cameras through the frame grabbers to memory is that the buffers get full. For example, if you have only one buffer, which is the same size as one frame, then if the acquisition that fills the buffer is faster than the processing you want to do on the collected data, for example saving it to disc, then there will be a problem if that goes on for too long. Just acquiring one image frame to one buffer is no problem but when the number of acquired images is more than the buffers there is a chance that a buffer, or a part of it, gets overwritten before the processing has been done, resulting in processed data from mixed frames. One way to solve this would be to use the same amount of buffers as processed frames from the camera. But that will create other problems. One problem is that this will demand a lot of memory when the number of buffers grows larger and larger.

Around 3 buffers seemed enough to be able to run one camera and collect images without getting mixed frames. But to be sure another type of buffer was used instead (a trash buffer). The difference between the trash buffer and the ordinary buffer was that if all buffers were full when a new frame was going to be saved it was just thrown away. That way you will not get mixed frames but occasionally lose some frames which was determined as the less evil.

CVB vs. Sapera – Image loading/blob analysis

Two software libraries were considered for developing the blob counting application, CVB and Sapera Essential.

The test of the two different libraries was done by measuring the time required to perform a blob analysis including and excluding the time required for loading the images from file to memory. The timing was implemented by using QueryPerformanceCounter() and QueryPerformanceFrequency() like the partial pseudo code below.

```
BIGINT start = QueryPerformanceCounter();

// code to time

BIGINT end = QueryPerformanceCounter();

double time = (double)(end - start) / QueryPerformanceFrequency();
```

QueryPerformanceCounter() returns the value of the counter and QueryPerformanceFrequency() returns the rate the counter updates at. The elapsed time is calculated by dividing the increase of the counter by the update rate.

The test was done by splitting a large image into smaller ones to see the difference of one big image and many smaller images when it comes to loading images and doing the blob analysis for both libraries. Another reason to split up the big image in smaller ones is to be able to use threading for possible speed up in image loading and blob analysis. But no threading was used in this test. Each set of images was run several times and the shortest run times were recorded because the best times and not the average times were desired. Two timers were used. The first timed both image loading and blob analysis. The second one timed only the blob analysis. This was done to get an estimate for both a future online version, were only the time for blob analysis is important, and with both image loading and blob analysis for the offline version.

The file format used in the test was tiff. The bmp file format was tested also but the libraries are written so much better for the tiff format that bmp images is about a magnitude of 10 slower to load from the hard drive to RAM memory.

R^2 and linear regression implementation

I thought about using an existing library for this like LAPACK++ (Linear Algebra PACKage in C++), that is a software library for numerical linear algebra [25]. But for this fairly simple task a more efficient way is to implement this by creating a simple class in C++. Calculating R^2, k and m is just to sum. So it is just a simple loop over the points. Equation 1, 3 and 4 were used to implement R^2, k and m and only one loop were used in each function (incrementing all of the summations in each loop).

Blob parameter optimization

The choice of blob analysis parameters will affect the number of blobs found in an image. These parameters should be selected to get the best possible blob count for each image. The following parameters are varied in the optimization:

- **Min and max area of the blobs:** Determines by the area if a blob counts. The area consists of the number of pixels in the blob.
- **Min and max width of the blobs:** Determines by the width if a blob counts. The width is the number of pixels, on the x-axis, between the rightmost and the leftmost pixel in the blob.
- **Min and max height of the blobs:** Determines by the height if a blob counts. The height is the number of pixels, on the y-axis, between the topmost pixel and the one furthest to the bottom in the blob.
- **Low and high threshold of the blobs:** Determines by the two thresholds what may count as a blob. The original image is binarized mapping all pixel intensities between the two thresholds to 1 and the rest to 0. Only pixels with the value 1 may constitute the sets of pixels that make up the blobs in the analyzed image.

Then choosing parameters that give you a good blob count comes to getting the right values for the linear regression of the points consisting of blob count contra concentration of the test samples.

Wanted:

- $R^2 \approx 1$, that is going toward perfect linearity.
- $k > 0$, gives that the higher the concentration of the sample is the higher is the blob count.

- $blob\ count\ at\ zero\ concentration\ \approx 0$, gives that the background is close to zero.

Not wanted:

- $k = 0$, gives that the blob count is constant independently of the samples concentration.
- $k < 0$, gives that the blob count decreases when the sample concentration increases.
- $blob\ count\ at\ zero\ concentraion \gg 0$, gives that we have high background and other interference in the blob count.

The optimization is done by two different criteria.

Low detection level: The most relevant for the BioNanoLab project is to get a high distinction, high signal difference, between the background and samples with low concentration. This will make it possible to detect very low concentrations. An important feature when presence is more important than how much there is of something in a sample.

$$Low\ detection\ level = \frac{Blob\ count\ at\ concentration\ 0.01}{(Blob\ count\ at\ concentration\ 0 + 1)}\ (\boldsymbol{formula\ 1})$$

The formula is just the lowest concentration divided by the background (zero concentration). The one is added to remove zero division. The result should be as big as possible to get a good signal difference.

High precision: Another way to do the optimization is to strive for high precision, meaning the ability to differentiate between small differences in sample concentrations. This could be useful for example when measuring the growth of bacteria in a sample. Being able to recognize very small differences in bacteria concentration makes it possible to wait shorter times to determine if the bacteria are reproducing. A useful tool to check for antibiotic resistance.

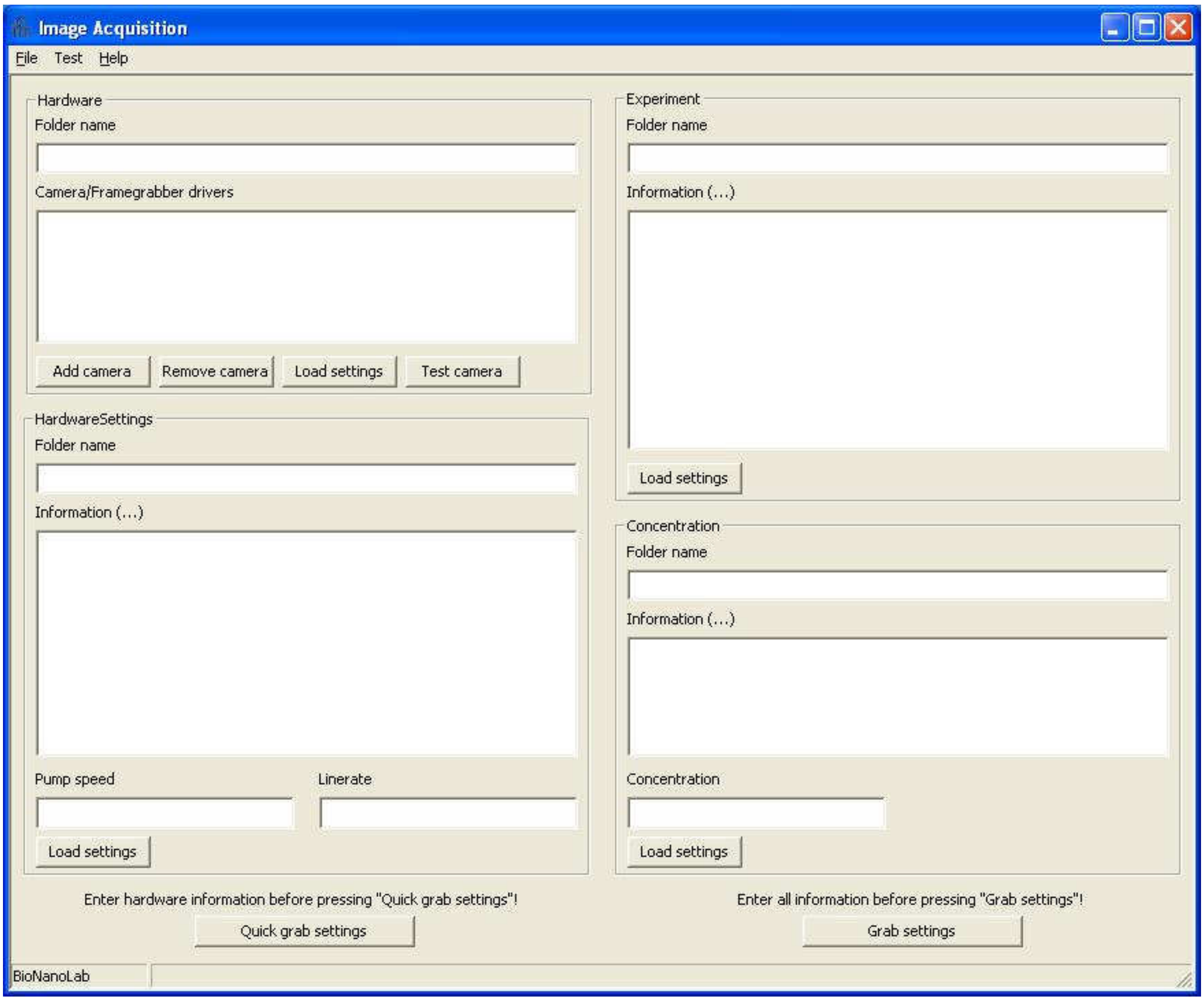
$$High\ precision = R^2 + \log_{10}(k)/35\ \ (\boldsymbol{formula\ 2})$$

The formula has two parts. First is the R^2 value. Good linearity is wanted so that the signals for different concentrations lay proportionally, by the concentration, apart from each other. The second part of the formula is the slope, k, of the concentration contra blob count line for the specific dilution series. A high k value is wanted here so that a small variation in the concentration gives big changes in the signal (blob count). The second part is weighted because otherwise would that part take over the whole formula. Exactly how k was weighted was determined by empirical studies (different weights were tested until the wanted result was reached).

Results

Image acquisition

The GUI for the image acquisition software looks like figure 10 below.



Enter hardware information before pressing "Quick grab settings"!

Enter all information before pressing "Grab settings"!

Figure 10 – Image acquisitions GUI.

The interface was implemented for its special purpose as a tool for reaching online performance later on and simplifying the blob parameter optimization. Not too much time was spent on user friendliness. Instead the focus was aimed towards functionality and getting the program up and running, to be able to do the blob parameter optimization as fast as possible.

See figure 11 for a zoom of an image saved with the software.

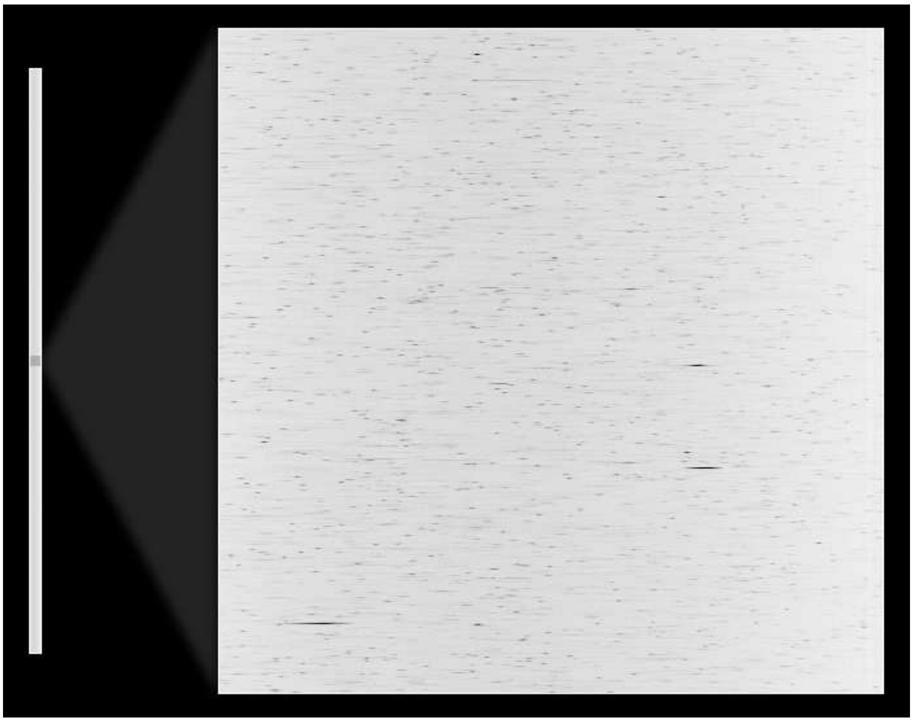


Figure 11 - Image taken with the image acquisition software (512x30000 pixels). The image to the left is zoomed out and the part to the right is almost the original size. Pump speed = 31, line rate = 5000 and blob concentration = 10 picomolar. The image in whole has been inverted to enhance the blobs compared to the background.

For the rest of the images in this dilution series se Appendix A.

CVB vs. Sapera – Image loading/blob analysis

The original images that are used in the test are 512x60000 pixels large. There were five different images with the blob concentrations

- 0 picomolar
- 0.01 picomolar
- 0.1 picomolar
- 1 picomolar
- 10 picomolar

Each image was then divided into sets of sub images

- 1 image with 60000 lines
- 2 images with 30000 lines
- 3 images with 20000 lines
- 4 images with 15000 lines
- 5 images with 12000 lines

The test was run 100 times on each image set and concentration pair. Figure 13 shows the best times of the 100 runs for the concentration 10 picomolar.

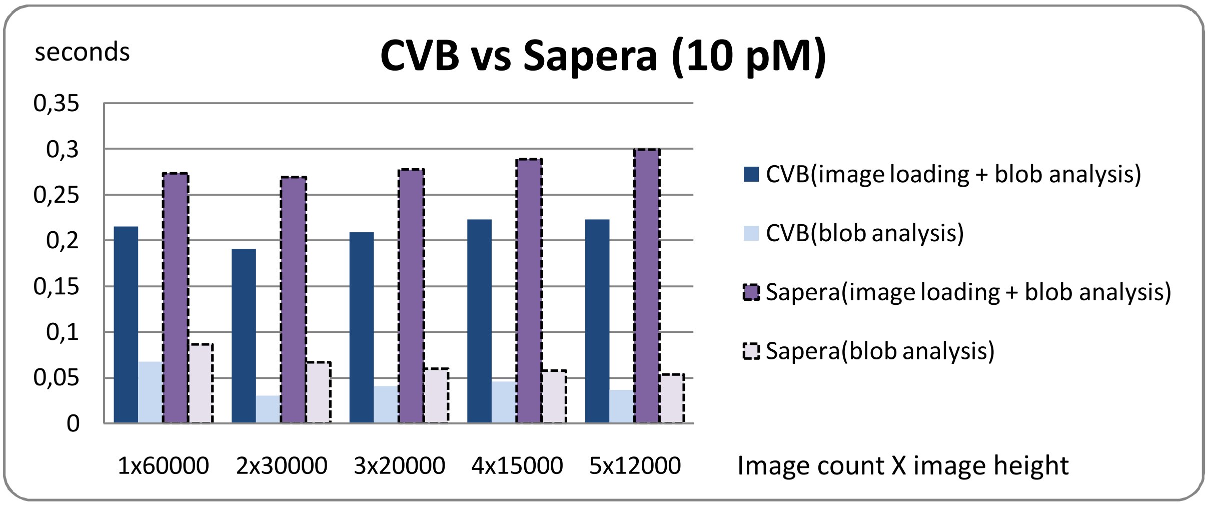


Figure 12 – Results from the images with 10 pM concentration. The test was run on both CVB and Sapera Essential and with image loading and blob analysis and with only blob analysis.

For the rest of the results see Appendix B.

Blob parameter optimization

The test was run on these blob parameter intervals and step sizes

- Min area: 1-4 (step size: 1)
- Max area: 200-200 (step size: 1)
- Min height: 1-5 (step size: 1)
- Max height: 15-15 (step size: 1)
- Min width: 1-5 (step size: 1)
- Max width: 5-20 (step size: 5)
- Low threshold: 10-80 (step size: 10)
- High threshold: 255-255 (step size: 10)

The step sizes were chosen a bit larger on some parameters to save time. This choice of parameter intervals resulted in 3200 different blob parameter settings. The test was run rather roughly to be able to get the big picture of the best parameters for the blob analysis. Runs with finer settings are planned to be run in the future. Three different pump speeds in the instrument, 37.15 µl/minute, 10 µl/minute and 6.25 µl/minute, and three line rates for the cameras, 5000, 20000 and 40000, where used in the test.

Low detection level

Pump speed	Line rate	Min area	Max area	Min height	Max height	Min width	Max width	Low thresh	High thresh
37.15	5000	4	200	1	15	1	15	40	255
37.15	20000	2	200	1	15	1	20	10	255
37.15	40000	1	200	1	15	1	10	10	255
10	5000	1	200	2	15	1	20	30	255
10	20000	2	200	1	15	1	15	20	255
10	40000	1	200	5	15	1	20	10	255
6.25	5000	1	200	2	15	1	20	30	255
6.25	20000	1	200	5	15	5	10	10	255
6.25	40000	1	200	1	15	1	15	10	255

Table 1 - The best results from the low detection level run.

High precession

Pump speed	Line rate	Min area	Max area	Min height	Max height	Min width	Max width	Low thresh	High thresh
37.15	5000	1	200	1	15	1	20	30	255
37.15	20000	4	200	1	15	1	20	10	255
37.15	40000	1	200	1	15	1	15	10	255
10	5000	1	200	1	15	1	20	30	255
10	20000	1	200	1	15	1	20	20	255
10	40000	2	200	1	15	1	20	10	255
6.25	5000	1	200	1	15	1	20	30	255
6.25	20000	1	200	1	15	1	20	20	255
6.25	40000	2	200	1	15	1	20	10	255

Table 2 - The best results from the high precession run.

See appendix C for some extracts from the results.

Conclusion and discussion

Image acquisition

Image acquisition with only one camera using 3 buffers was possible without losing any frames with the image acquisition software written. Using only one camera was enough to solve the key issues of the thesis. Running two cameras with the software was pretty stable also. But frames were lost at random times. Some more effort is needed here to determine the optimal amount of buffers for running two cameras. One way to compensate for the lost frames, if they are not too many, is to remove images in time that not all of the cameras have saved. That way, the image data will be complete and synchronized anyway. Some longer acquisition time may be needed to get the amount of image data wanted though.

Three cameras in the system have not been tested yet. But when the problem of using two cameras have been solved, going to three cameras should not be that hard.

CVB vs. Sapera – Image loading/blob analysis

The results (figure 12) show a steady trend that CVB is a bit faster than Sapera Essential on the image loading and on the blob analysis. But Sapera Essential was declared the winner anyway mainly because it was chosen for the image acquisition. This will make the future online version of the image acquisition and blob analysis simpler, working with only one library for both tasks. Sapera Essentials run times were good enough to be used also for both image loading and blob analysis as it was. Threading might also be used if the time, that the blob analysis take, need to be shorter in the future.

Another reason that Sapera Essential were chosen over CVB was that CVB started to miscount after a period of time. It did count correct sometimes but not too often. One possible reason to the problem could be Windows update as the counting worked just fine earlier in the thesis. The test seemed worth to do anyway at least to see how well Sapera Essential performed. The rest is about Sapera Essential only.

The total time for image loading increased some when the original image was divided into more images. That was expected as there are more overhead when the image loading operations increases. There is also a speed up, in the blob analysis, as the original image is divided into more images because the internal data structure for storing the blobs gets faster to work with split into smaller parts.

Blob parameter optimization

One prominent feature of the result from the test is that for both the low detection level and the high precession should the low threshold be reduced when running on higher line rates compared to lower line rates. That is not a very surprising result as when the line rate goes up does the exposure time go down thus less light have time to be collected for each line in the cameras CCD chip.

Other clear patterns are hard to find in the best parameters, according to this test, for each hardware setting. Anyway this test was not constructed to find exactly the best parameters and then use them here after (more to give a hint in the right direction). This test has to be run again with finer intervals and step sizes to get closer to the parameters to use. Maybe must the two functions for the low detection level and high precession have to be fine tuned a bit also.

Recommendation

Suggestions for improvements

Dust removal: It is hard to avoid particles like dust to get in to the sample, and the instrument, creating unwanted background (variance to the blob counts). One possible solution to this problem could be to use two cameras when counting blobs labeled with only one type of fluorophore. Then there should only be blobs in the images from the camera with the right laser, for the fluorophore, used for that channel. The other cameras images should be black. Blobs that show up on images from both cameras could then be counted as dust and removed from the count.

Getting to online performance: One of the demands on the hardware to get to online performance, saving the image data in RAM memory only, is that you got enough RAM. A little example demonstrating the needed amount of RAM.

$$\begin{array}{c} Acqusition\ time: 30\ second \\ Line\ rate: 60kHz \\ Image\ width: 1024\ pix \\ Color\ depth: 8\ bits \\ Cameras: 3 \\ \hline Total: 30 * 60000 * 1024 * 8 * 3 = 44236800000\ bits \\ \downarrow \\ 5.5296\ GB \end{array}$$

That is over the limits of a 32 bits system were the maximum RAM is approximately 4 GB. But the video card and other resources need some of the address space so around 3 GB is left on a 32 bit system. Then an online version of the software needs a 64 bits operating system and 64 bit versions of CVB or Sapera Essential. The 64 bit version of CVB has not been released yet and the 64 bit version of Sapera Essential was released towards the end of my thesis.

Form factor: Some simple software for checking the form factors, of blobs in an image, was written during the thesis but not to completion due to more important tasks. The formula used for the form factor was

$$\max(widht, height)/\min(widht, height) * sign(width - height) \quad (\boldsymbol{formula\ 3})$$

This way a symmetric blob, same height as width, will give a form factor of 0, blobs with greater height than width will have a negative value and with greater width than height a positive value.

One use of the form factor is to analyze the difference between blobs in the middle of the micro channel and at the perimeters. Differences in blob form factors will probably be due to the fact that the flow of the channel is greater in the middle. That is blobs at the perimeter will be a little stretched by the fact that they will be captured more times by the cameras than blobs in the middle of the channel.

Using software line rate/frame rate generation: The frame grabber used together with Sapera LT, the low level part of Sapera Essentials, have the possibility to send out triggers trough the frame grabber to trigger both line rate and frame rate. That way there is no need for an extra external signal generator to create the triggers.

References

[1] http://en.wikipedia.org/wiki/Toxin 2009-02-01

[2] http://www.unesco.org/courier/1999_03/uk/ethique/txt1.htm 2009-02-21

[3] http://en.wikipedia.org/wiki/Ricin 2009-02-01

[4] http://en.wikipedia.org/wiki/Botulinum 2009-02-01

[5] KBM:S UPPDRAG/UTREDNINGAR (2006) CBRN – Ämnen och hotbilder. ISBN: 91-975934-1-9

[6] http://en.wikipedia.org/wiki/2001_anthrax_attacks 2009-02-03

[7] Wheelis M. Biological warfare at the 1346 siege of Caffa. Emerg Infect Dis [serial online] 2002 Sep [date cited];8. Available from: URL: http://www.cdc.gov/ncidod/EID/vol8no9/01-0536.htm

[8] Jarvius, J., et al. (2006) Direct observation of individual endogenous protein complexes in situ by proximity ligation. *Nature Methods*, 3(12): 995-1000.

[9] Fredriksson, S., et al. (2006) Detection of individual microbial pathogens by proximity ligation. *Nature Methods*, 52(6): 1152-60.

[10] Fredriksson, S., et al. (2002) Protein detection using proximity-dependent DNA ligation assays. *Nature Methods*, 20(5): 473-7.

[11] Jarvius, J., et al. (2006) Digital quantification using amplified single-molecule detection. *Nature Methods*, 3(9), 725-727.

[12] Melin, J., et al. (2007) Homogeneous amplified single-molecule detection: Characterization of key parameters. *Analytical Biochemistry*, 368(2), 230-238.

[13] D Semwogerere & ER Weeks, published in the *Encyclopedia of Biomaterials and Biomedical Engineering*, Taylor & Francis (2005)

[14] http://en.wikipedia.org/wiki/Fluorescence 2009-02-13

[15] Gonzalez, R., et al. (2008) Digital Image Processing, Third Edition. ISBN: 978-0-13-168728-8

[16] http://en.wikipedia.org/wiki/Charge-coupled_device 2009-02-13

[17] http://en.wikipedia.org/wiki/Frame_grabber 2009-02-13

[18] http://homepages.inf.ed.ac.uk/rbf/HIPR2/connect.htm 2009-01-19

[19] http://archive.evaluationengineering.com/archive/articles/0806/0806blob_analysis.asp 2009-02-13

[20] http://en.wikipedia.org/wiki/Coefficient_of_determination 2009-01-09

[21] http://phoenix.phys.clemson.edu/tutorials/regression/index.html 2009-01-09

[22] http://support.microsoft.com/kb/172338 2009-02-13

[23] http://en.commonvisionblox.de/pages/cvb/main.php?view=5&language=en 2009-02-15

[24] http://www.dalsa.com/mv/products/software.aspx 2009-02-15

[25] http://sourceforge.net/projects/lapackpp 2009-02-21

Appendix

Appendix A

Here are the rest of the images with concentrations, from zero to one picomolar, in the dilution series with pump speed equal to 31 and line rate equal to 5000.

Figure 13 - Image taken with the image acquisition software (512x30000 pixels). The image to the left is zoomed out and the part to the right is almost the original size. Pump speed = 31, line rate = 5000 and blob concentration = 0 picomolar. The image in whole has been inverted to enhance the blobs compared to the background.

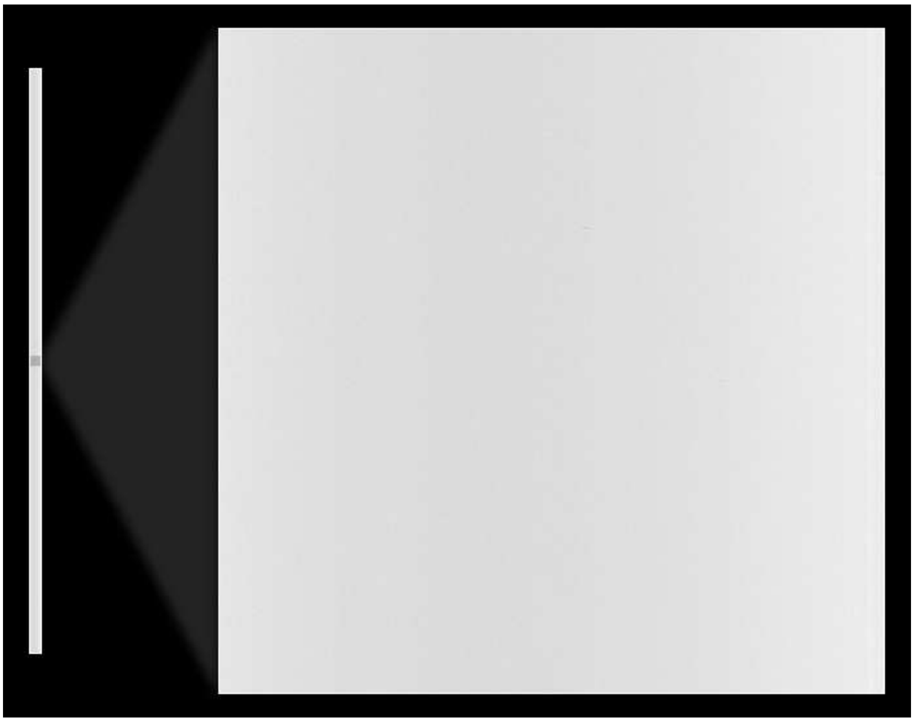


Figure 14 - Image taken with the image acquisition software (512x30000 pixels). The image to the left is zoomed out and the part to the right is almost the original size. Pump speed = 31, line rate = 5000 and blob concentration = 0.01 picomolar. The image in whole has been inverted to enhance the blobs compared to the background.picomolar.

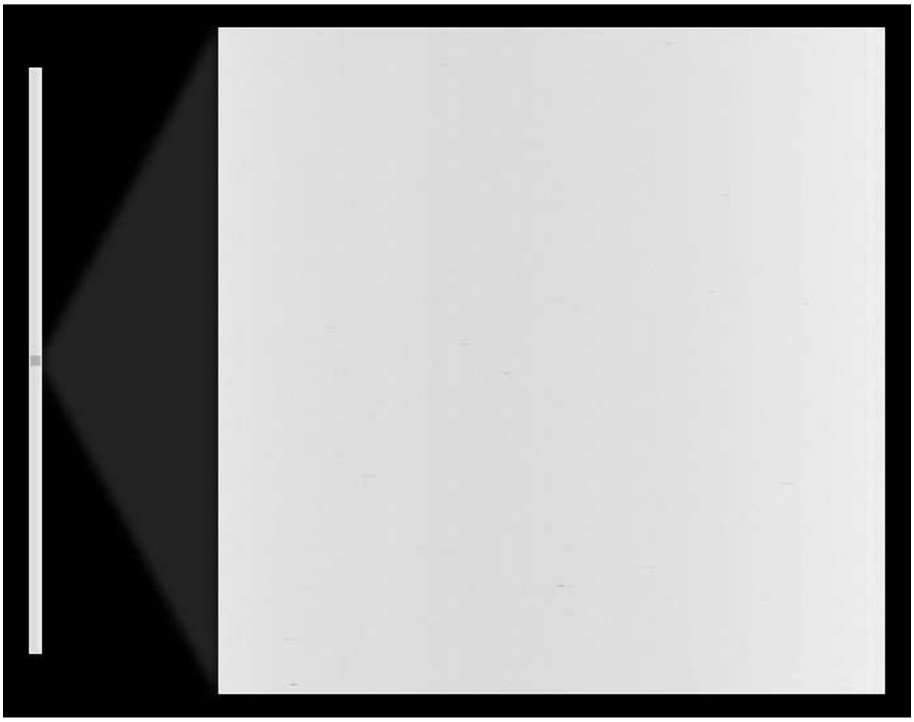


Figure 15 - Image taken with the image acquisition software (512x30000 pixels). The image to the left is zoomed out and the part to the right is almost the original size. Pump speed = 31, line rate = 5000 and blob concentration = 0.1 picomolar. The image in whole has been inverted to enhance the blobs compared to the background.

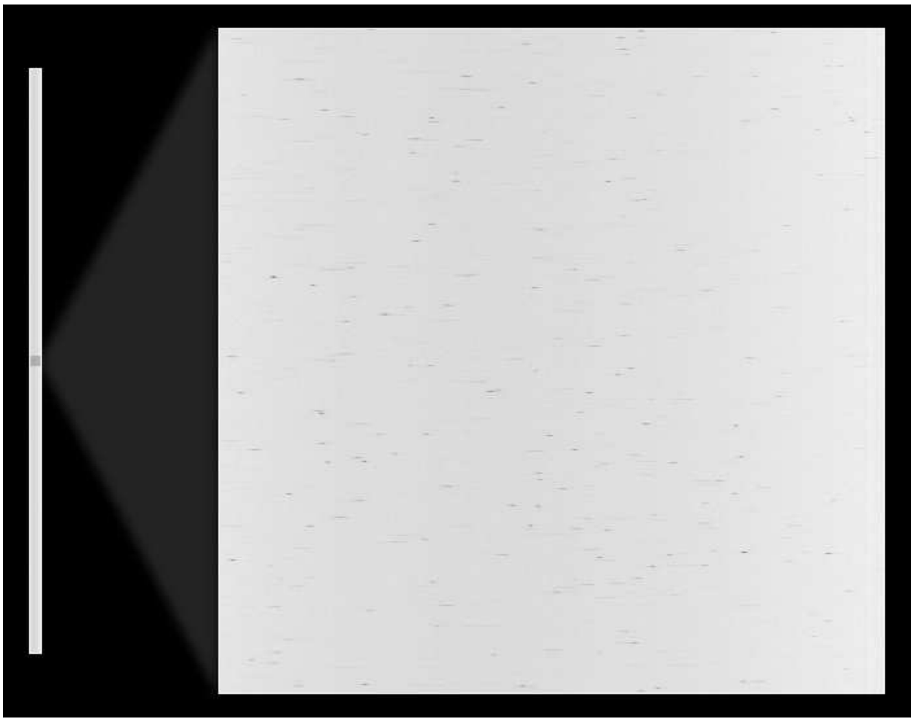


Figure 16 - Image taken with the image acquisition software (512x30000 pixels). The image to the left is zoomed out and the part to the right is almost the original size. Pump speed = 31, line rate = 5000 and blob concentration = 1 picomolar. The image in whole has been inverted to enhance the blobs compared to the background.

Appendix B

The results of the test CVB vs. Sapera (image loading and blob analysis contra only blob analysis).

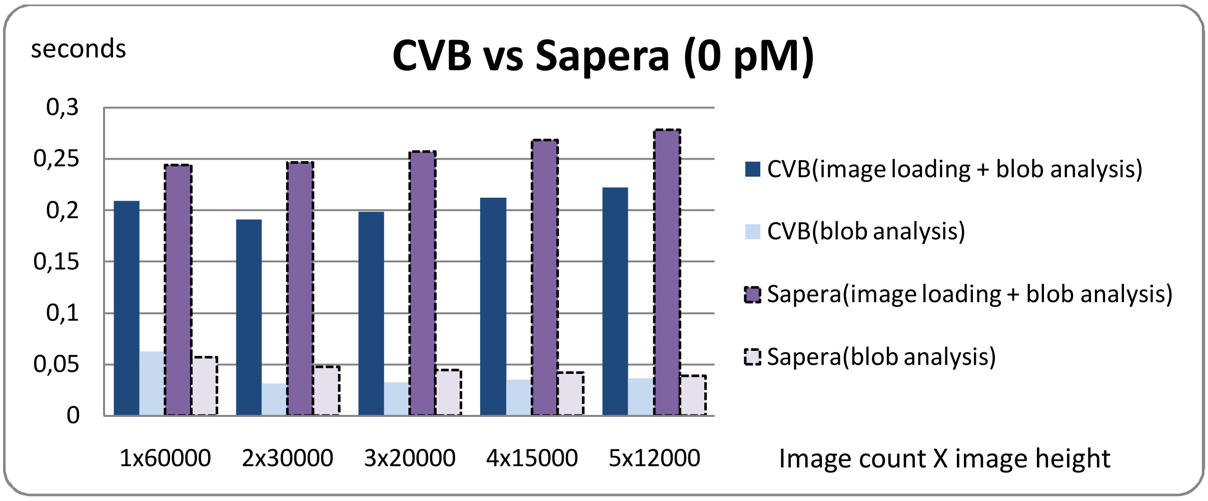


Figure 17 - Results from the images with 0 pM concentration. The test was run on both CVB and Sapera Essential and with image loading and blob analysis and with only blob analysis.

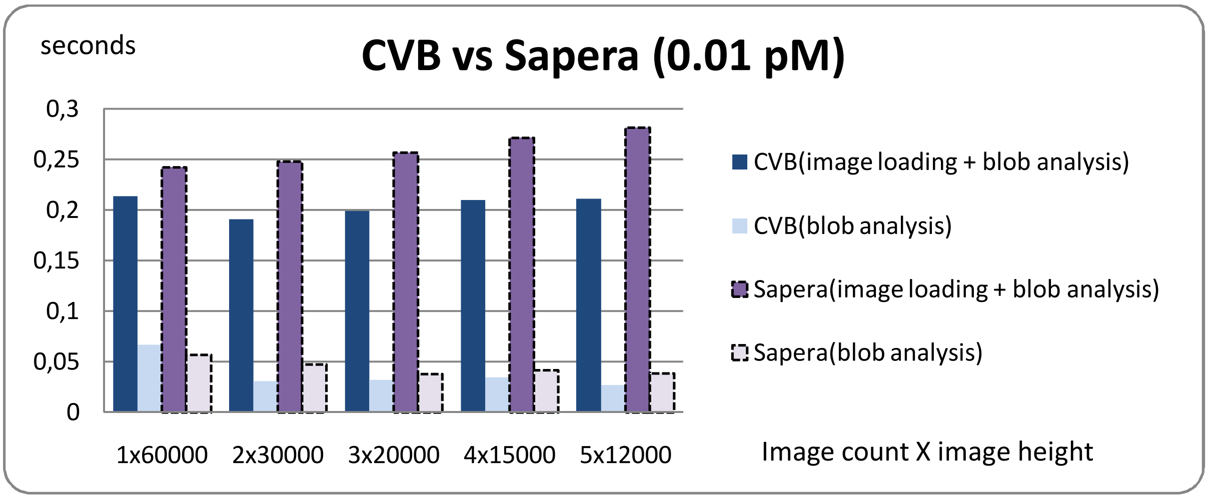


Figure 18 - Results from the images with 0.01 pM concentration. The test was run on both CVB and Sapera Essential and with image loading and blob analysis and with only blob analysis.

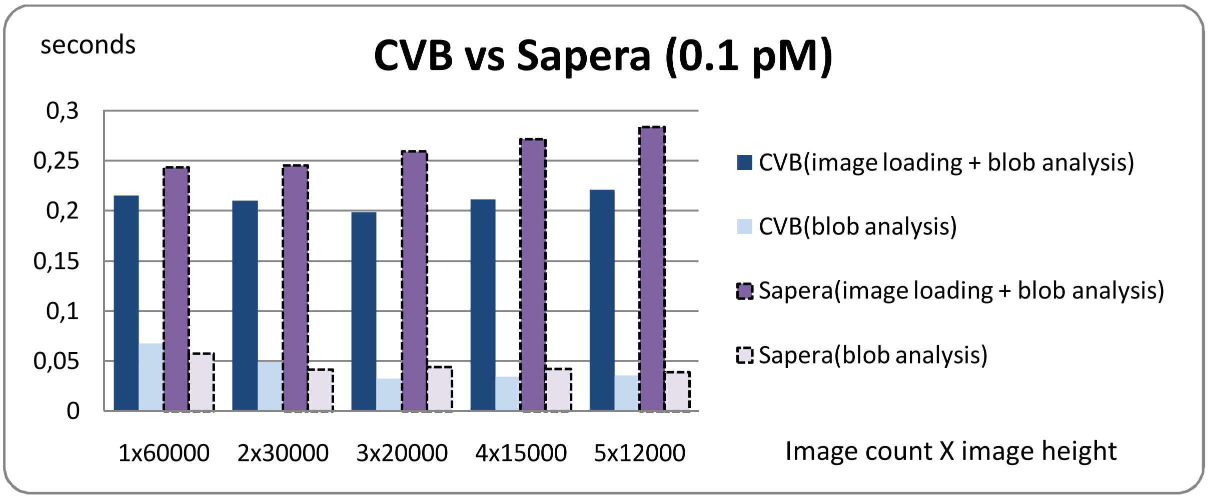


Figure 19 - Results from the images with 0.1 pM concentration. The test was run on both CVB and Sapera Essential and with image loading and blob analysis and with only blob analysis.

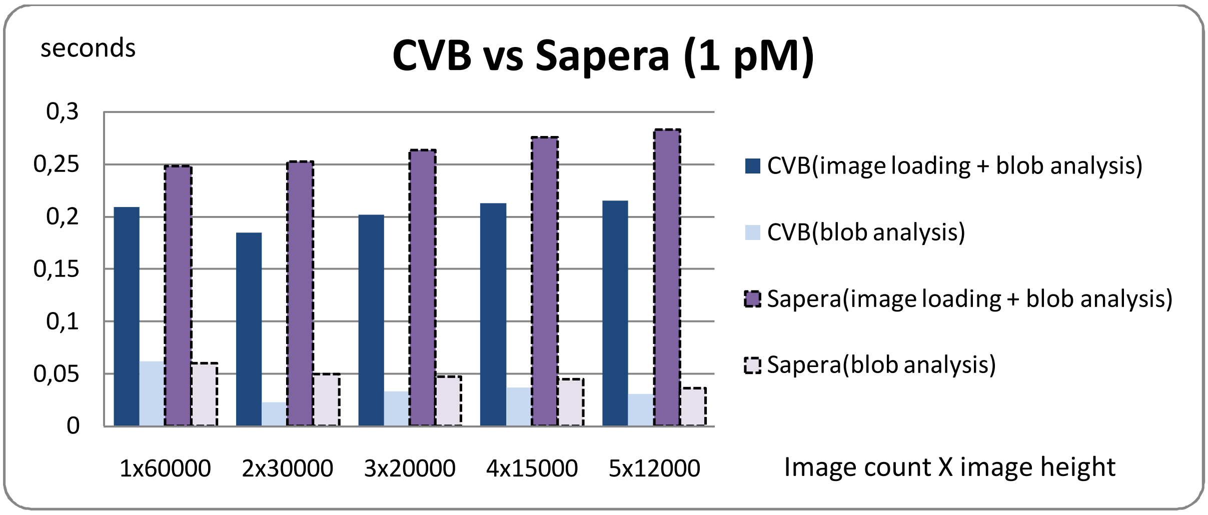


Figure 20 - Results from the images with 1 pM concentration. The test was run on both CVB and Sapera Essential and with image loading and blob analysis and with only blob analysis.

Appendix C

Below are the top ten results on both low detection level and high precession on all the tests. The columns in order from the right to the left are first the blob parameters that is follow by the blob count at the five blob concentrations. Next are the slope, k, and the intersection with the y axis, m, and the R^2 value. The last two are the high precession formula value, hP, and the low detection level formula value, lDL.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
4/200/1/15/1/15/40/255:	0	85	611	10625	86568	10747.2	-152.111	0.998302	1.11348	85
4/200/1/15/1/20/40/255:	0	85	613	10635	87020	10757	-151.826	0.998313	1.1135	85
4/200/1/15/2/15/40/255:	0	85	611	10625	86568	10747.2	-152.111	0.998302	1.11348	85
4/200/1/15/2/20/40/255:	0	85	613	10635	87020	10757	-151.826	0.998313	1.1135	85
4/200/1/15/3/15/40/255:	0	85	607	10529	86042	10649.5	-149.976	0.998311	1.11338	85
4/200/1/15/3/20/40/255:	0	85	609	10539	86494	10659.2	-149.692	0.998322	1.1134	85
1/200/1/15/4/15/40/255:	0	84	588	10135	83055	10249.2	-142.399	0.998333	1.11292	84
1/200/1/15/4/20/40/255:	0	84	590	10145	83507	10259	-142.115	0.998344	1.11295	84
2/200/1/15/4/15/40/255:	0	84	588	10135	83055	10249.2	-142.399	0.998333	1.11292	84
2/200/1/15/4/20/40/255:	0	84	590	10145	83507	10259	-142.115	0.998344	1.11295	84

Table 3 - Pump speed = 31 and line rate = 5000. Low detection level tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/20/30/255:	8	698	4735	61979	426771	62315.9	-437.661	0.999416	1.1364	77.5556
1/200/1/15/1/15/30/255:	8	698	4727	61666	416835	61995.8	-429.087	0.999427	1.13635	77.5556
1/200/1/15/1/10/30/255:	8	671	4589	58705	386331	58993.8	-377.542	0.999494	1.1358	74.5556
2/200/1/15/1/20/30/255:	7	431	3319	44079	305625	44355.3	-349.585	0.999396	1.13217	53.875
1/200/1/15/2/20/30/255:	7	430	3316	44000	304591	44275.1	-348.091	0.9994	1.13215	53.75
2/200/1/15/2/20/30/255:	7	430	3316	44000	304591	44275.1	-348.091	0.9994	1.13215	53.75
2/200/1/15/1/15/30/255:	7	431	3311	43766	295689	44035.2	-341.013	0.999412	1.13209	53.875
1/200/1/15/2/15/30/255:	7	430	3308	43687	294655	43955	-339.518	0.999415	1.13207	53.75
2/200/1/15/2/15/30/255:	7	430	3308	43687	294655	43955	-339.518	0.999415	1.13207	53.75
1/200/1/15/1/5/30/255:	7	528	3486	41921	284011	42055.3	-184.845	0.999669	1.13178	66

Table 4 - Pump speed = 31 and line rate = 5000. High precession tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
2/200/1/15/1/20/10/255:	2	15112	22326	28236	84506	17654.3	11519.9	0.490888	0.612226	5037.33
2/200/1/15/1/15/10/255:	2	15111	22315	28060	83027	17476.3	11522.3	0.485688	0.606901	5037
2/200/1/15/1/10/10/255:	2	15106	22242	26997	76018	16402.8	11535	0.453415	0.573842	5035.33
2/200/1/15/1/5/10/255:	2	15079	22038	24275	56018	13663.4	11556.9	0.364064	0.482222	5026.33
1/200/2/15/1/20/10/255:	2	14325	20953	23534	57307	13462.5	10967.7	0.381753	0.499728	4775
2/200/2/15/1/20/10/255:	2	14325	20953	23534	57307	13462.5	10967.7	0.381753	0.499728	4775
1/200/2/15/1/15/10/255:	2	14324	20944	23385	55946	13311.8	10969.7	0.376335	0.49417	4774.67
2/200/2/15/1/15/10/255:	2	14324	20944	23385	55946	13311.8	10969.7	0.376335	0.49417	4774.67
1/200/2/15/1/10/10/255:	2	14321	20892	22529	50324	12445.2	10982.5	0.344624	0.461624	4773.67
2/200/2/15/1/10/10/255:	2	14321	20892	22529	50324	12445.2	10982.5	0.344624	0.461624	4773.67

Table 5 - Pump speed = 31 and line rate = 20000. Low detection level tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
4/200/1/15/1/20/10/255:	1	91	568	5852	42361	5843.58	6.40705	0.999948	1.10757	45.5
4/200/1/15/2/20/10/255:	1	90	562	5849	42348	5842.39	4.23712	0.999939	1.10756	45
4/200/1/15/3/20/10/255:	1	77	528	5777	41820	5781.84	-8.71139	0.999889	1.10738	38.5
4/200/1/15/1/15/10/255:	1	90	557	5676	40882	5665.62	8.7904	0.999956	1.10719	45
4/200/1/15/2/15/10/255:	1	89	551	5673	40869	5664.43	6.62047	0.999948	1.10718	44.5
4/200/1/15/3/15/10/255:	1	76	517	5601	40341	5603.88	-6.32804	0.999903	1.107	38
1/200/1/15/3/20/10/255:	1	395	1125	7091	48323	6861.97	248.802	0.996866	1.10648	197.5
2/200/1/15/3/20/10/255:	1	395	1125	7091	48323	6861.97	248.802	0.996866	1.10648	197.5
3/200/1/15/3/20/10/255:	1	395	1125	7091	48323	6861.97	248.802	0.996866	1.10648	197.5
1/200/1/15/4/20/10/255:	0	54	447	5397	39241	5420.43	-29.668	0.999706	1.10639	54

Table 6 - Pump speed = 31 and line rate = 20000. High precession tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/10/10/255:	2	52	90	1433	10336	1431.33	-2.94468	0.997078	1.08724	17.3333
1/200/1/15/1/15/10/255:	2	52	90	1459	10491	1458.09	-3.86956	0.997021	1.08741	17.3333
1/200/1/15/1/20/10/255:	2	52	90	1459	10506	1458.09	-3.86956	0.997021	1.08741	17.3333
1/200/1/15/1/5/10/255:	2	50	81	1179	8567	1172.98	2.49802	0.997112	1.08481	16.6667
1/200/1/15/2/10/10/255:	0	12	41	930	6839	942.118	-15.6877	0.9968	1.08177	12
1/200/1/15/2/15/10/255:	0	12	41	956	6994	968.874	-16.6126	0.996687	1.08201	12
1/200/1/15/2/20/10/255:	0	12	41	956	7009	968.874	-16.6126	0.996687	1.08201	12
2/200/1/15/1/10/10/255:	0	12	43	944	6981	956.02	-15.5455	0.996969	1.08213	12
2/200/1/15/1/15/10/255:	0	12	43	970	7136	982.776	-16.4703	0.996856	1.08235	12
2/200/1/15/1/20/10/255:	0	12	43	970	7151	982.776	-16.4703	0.996856	1.08235	12

Table 7 - Pump speed = 31 and line rate = 40000. Low detection level tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/15/10/255:	2	52	90	1459	10491	1458.09	-3.86956	0.997021	1.08741	17.3333
1/200/1/15/1/20/10/255:	2	52	90	1459	10506	1458.09	-3.86956	0.997021	1.08741	17.3333
1/200/1/15/1/10/10/255:	2	52	90	1433	10336	1431.33	-2.94468	0.997078	1.08724	17.3333
1/200/1/15/1/5/10/255:	2	50	81	1179	8567	1172.98	2.49802	0.997112	1.08481	16.6667
2/200/1/15/1/15/10/255:	0	12	43	970	7136	982.776	-16.4703	0.996856	1.08235	12
2/200/1/15/1/20/10/255:	0	12	43	970	7151	982.776	-16.4703	0.996856	1.08235	12
2/200/1/15/1/10/10/255:	0	12	43	944	6981	956.02	-15.5455	0.996969	1.08213	12
1/200/1/15/2/15/10/255:	0	12	41	956	6994	968.874	-16.6126	0.996687	1.08201	12
1/200/1/15/2/20/10/255:	0	12	41	956	7009	968.874	-16.6126	0.996687	1.08201	12
2/200/1/15/2/15/10/255:	0	12	41	956	6994	968.874	-16.6126	0.996687	1.08201	12

Table 8 - Pump speed = 31 and line rate = 40000. High precession tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/2/15/1/20/30/255:	1	74	519	7554	37542	7613.96	-75.8736	0.999035	1.10994	37
2/200/2/15/1/20/30/255:	1	74	519	7554	37542	7613.96	-75.8736	0.999035	1.10994	37
1/200/2/15/2/20/30/255:	1	72	510	7402	37108	7460.57	-74.0594	0.999048	1.1097	36
2/200/2/15/2/20/30/255:	1	72	510	7402	37108	7460.57	-74.0594	0.999048	1.1097	36
3/200/2/15/1/20/30/255:	1	72	506	7297	36766	7353.53	-71.6047	0.999071	1.10954	36
3/200/2/15/2/20/30/255:	1	72	506	7296	36764	7352.5	-71.5692	0.999072	1.10954	36
1/200/2/15/1/20/40/255:	0	35	231	3209	19130	3230.62	-27.747	0.999199	1.09946	35
1/200/2/15/2/20/40/255:	0	35	230	3200	19040	3221.61	-27.7471	0.999193	1.09942	35
1/200/2/15/3/20/40/255:	0	35	222	3134	18489	3155.71	-27.9605	0.999127	1.0991	35
1/200/2/15/4/20/40/255:	0	35	204	2932	16982	2952.39	-26.5375	0.999028	1.09818	35

Table 9 - Pump speed = 37 and line rate = 5000. Low detection level tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/20/30/255:	27	477	2850	34691	147742	34787.3	-142.222	0.999599	1.12935	17.0357
1/200/1/15/1/15/30/255:	26	453	2730	32691	140135	32769	-118.392	0.999648	1.12866	16.7778
1/200/1/15/1/10/30/255:	25	407	2403	27559	118463	27588.3	-57.2449	0.999753	1.12663	15.6538
2/200/1/15/1/20/30/255:	18	250	1890	23160	109894	23253.6	-123.367	0.999636	1.12439	13.1579
1/200/1/15/2/20/30/255:	18	248	1881	23008	109460	23100.2	-121.553	0.999642	1.12432	13.0526
2/200/1/15/2/20/30/255:	18	248	1881	23008	109460	23100.2	-121.553	0.999642	1.12432	13.0526
2/200/1/15/1/15/30/255:	17	226	1770	21160	102287	21235.3	-99.5374	0.999711	1.12334	12.5556
1/200/1/15/2/15/30/255:	17	224	1761	21008	101853	21081.9	-97.7231	0.999717	1.12326	12.4444
2/200/1/15/2/15/30/255:	17	224	1761	21008	101853	21081.9	-97.7231	0.999717	1.12326	12.4444
1/200/1/15/1/5/30/255:	21	339	1780	20198	80342	20198.1	-20.4787	0.999738	1.12275	15.4091

Table 10 - Pump speed = 37 and line rate = 5000. High precession tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
2/200/1/15/1/15/20/255:	1	29	248	3340	15937	3363.02	-28.7392	0.999365	1.10013	14.5
2/200/1/15/1/20/20/255:	1	29	248	3356	16008	3379.49	-29.3083	0.999348	1.10017	14.5
1/200/1/15/2/15/20/255:	1	28	247	3325	15814	3348.22	-28.8814	0.999372	1.10008	14
1/200/1/15/2/20/20/255:	1	28	247	3341	15885	3364.69	-29.4506	0.999355	1.10012	14
2/200/1/15/1/10/20/255:	1	28	236	3159	15208	3180.17	-26.498	0.999383	1.09945	14
2/200/1/15/2/15/20/255:	1	28	247	3325	15814	3348.22	-28.8814	0.999372	1.10008	14
2/200/1/15/2/20/20/255:	1	28	247	3341	15885	3364.69	-29.4506	0.999355	1.10012	14
1/200/1/15/2/10/20/255:	1	27	235	3144	15085	3165.37	-26.6404	0.999391	1.0994	13.5
2/200/1/15/2/10/20/255:	1	27	235	3144	15085	3165.37	-26.6404	0.999391	1.0994	13.5
3/200/1/15/1/15/20/255:	1	27	225	2966	14174	2984.72	-23.5099	0.999433	1.09872	13.5

Table 11 - Pump speed = 37 and line rate = 20000. Low detection level tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/20/20/255:	2	36	287	3859	18336	3884.2	-31.8656	0.999352	1.1019	12
1/200/1/15/1/15/20/255:	2	36	287	3843	18265	3867.74	-31.2965	0.999366	1.10187	12
1/200/1/15/1/10/20/255:	2	35	275	3662	17536	3684.88	-29.0554	0.999382	1.10128	11.6667
2/200/1/15/1/20/20/255:	1	29	248	3356	16008	3379.49	-29.3083	0.999348	1.10017	14.5
2/200/1/15/1/15/20/255:	1	29	248	3340	15937	3363.02	-28.7392	0.999365	1.10013	14.5
1/200/1/15/2/20/20/255:	1	28	247	3341	15885	3364.69	-29.4506	0.999355	1.10012	14
2/200/1/15/2/20/20/255:	1	28	247	3341	15885	3364.69	-29.4506	0.999355	1.10012	14
1/200/1/15/2/15/20/255:	1	28	247	3325	15814	3348.22	-28.8814	0.999372	1.10008	14
2/200/1/15/2/15/20/255:	1	28	247	3325	15814	3348.22	-28.8814	0.999372	1.10008	14
2/200/1/15/1/10/20/255:	1	28	236	3159	15208	3180.17	-26.498	0.999383	1.09945	14

Table 12 - Pump speed = 37 and line rate = 20000. High precession tops ten.

minA/maxA/minH/maxH/minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/5/15/1/20/10/255:	0	14	67	921	5212	925.521	-6.33203	0.99911	1.08386	14
1/200/5/15/2/20/10/255:	0	14	67	919	5212	923.463	-6.26089	0.999119	1.08384	14
1/200/5/15/3/20/10/255:	0	14	67	915	5191	919.346	-6.11859	0.999135	1.08381	14
1/200/5/15/4/20/10/255:	0	14	66	904	5122	908.279	-6.04742	0.999116	1.08364	14
2/200/5/15/1/20/10/255:	0	14	67	921	5212	925.521	-6.33203	0.99911	1.08386	14
2/200/5/15/2/20/10/255:	0	14	67	919	5212	923.463	-6.26089	0.999119	1.08384	14
2/200/5/15/3/20/10/255:	0	14	67	915	5191	919.346	-6.11859	0.999135	1.08381	14
2/200/5/15/4/20/10/255:	0	14	66	904	5122	908.279	-6.04742	0.999116	1.08364	14
3/200/5/15/1/20/10/255:	0	14	67	921	5212	925.521	-6.33203	0.99911	1.08386	14
3/200/5/15/2/20/10/255:	0	14	67	919	5212	923.463	-6.26089	0.999119	1.08384	14

Table 13 - Pump speed = 37 and line rate = 40000. Low detection level tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
2/200/1/15/1/20/10/255:	9	55	252	3021	15553	3020.66	-3.98422	0.99951	1.09894	5.5
2/200/1/15/1/15/10/255:	9	53	245	2928	15109	2927.49	-3.62848	0.999523	1.09857	5.3
1/200/2/15/1/20/10/255:	8	50	238	2793	14443	2791.87	-2.4941	0.999594	1.09805	5.55556
2/200/2/15/1/20/10/255:	8	50	238	2793	14443	2791.87	-2.4941	0.999594	1.09805	5.55556
1/200/1/15/2/20/10/255:	8	48	231	2789	14594	2790.29	-5.30441	0.999513	1.09796	5.33333
2/200/1/15/2/20/10/255:	8	48	231	2789	14594	2790.29	-5.30441	0.999513	1.09796	5.33333
1/200/2/15/1/15/10/255:	8	48	231	2700	13999	2698.7	-2.13837	0.999609	1.09764	5.33333
2/200/2/15/1/15/10/255:	8	48	231	2700	13999	2698.7	-2.13837	0.999609	1.09764	5.33333
1/200/1/15/2/15/10/255:	8	46	224	2696	14150	2697.11	-4.94867	0.999527	1.09755	5.11111
2/200/1/15/2/15/10/255:	8	46	224	2696	14150	2697.11	-4.94867	0.999527	1.09755	5.11111

Table 14 - Pump speed = 37 and line rate = 40000. High precession tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/2/15/1/20/30/255:	1	40	439	5107	55799	5128.95	-26.5336	0.999824	1.10582	20
2/200/2/15/1/20/30/255:	1	40	439	5107	55799	5128.95	-26.5336	0.999824	1.10582	20
1/200/2/15/2/20/30/255:	1	39	426	5031	54860	5054.41	-28.3479	0.999789	1.10561	19.5
2/200/2/15/2/20/30/255:	1	39	426	5031	54860	5054.41	-28.3479	0.999789	1.10561	19.5
3/200/2/15/1/20/30/255:	1	39	420	4953	54139	4975.65	-27.4941	0.999792	1.10542	19.5
3/200/2/15/2/20/30/255:	1	39	420	4953	54133	4975.65	-27.4941	0.999792	1.10542	19.5
4/200/2/15/1/20/30/255:	1	37	408	4844	52901	4867.28	-28.1699	0.999777	1.10513	18.5
4/200/2/15/2/20/30/255:	1	37	408	4844	52901	4867.28	-28.1699	0.999777	1.10513	18.5
1/200/1/15/3/5/40/255:	0	18	254	2236	25420	2229.98	8.18177	0.999786	1.09545	18
1/200/2/15/1/15/30/255:	1	36	389	4416	47337	4432.01	-19.3833	0.999872	1.10406	18

Table 15 - Pump speed = 40 and line rate = 5000. Low detection level tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/20/30/255:	8	150	1390	14464	147282	14473.1	-13.2727	0.999981	1.11885	16.6667
1/200/1/15/1/15/30/255:	8	144	1307	13359	135806	13359.2	-2.67187	0.999992	1.11787	16
1/200/1/15/1/10/30/255:	8	121	1113	10877	111421	10862.8	15.3282	0.999998	1.11531	13.4444
2/200/1/15/1/20/30/255:	6	90	1019	10436	106733	10445.3	-10.8262	0.999993	1.11482	12.8571
1/200/1/15/2/20/30/255:	6	89	1006	10360	105794	10370.8	-12.6405	0.999991	1.11473	12.7143
2/200/1/15/2/20/30/255:	6	89	1006	10360	105794	10370.8	-12.6405	0.999991	1.11473	12.7143
2/200/1/15/1/15/30/255:	6	84	936	9331	95257	9331.44	-0.22557	0.999998	1.11343	12
1/200/1/15/2/15/30/255:	6	83	923	9255	94318	9256.9	-2.03962	0.999998	1.11333	11.8571
2/200/1/15/2/15/30/255:	6	83	923	9255	94318	9256.9	-2.03962	0.999998	1.11333	11.8571
3/200/1/15/1/20/30/255:	4	77	875	9212	94541	9227.86	-18.731	0.999977	1.11327	15.4

Table 16 - Pump speed = 40 and line rate = 5000. High precession tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/5/15/5/10/10/255:	2	61678	59220	48497	30019	11434.8	39176.1	0.036878	0.152827	20559.3
2/200/5/15/5/10/10/255:	2	61678	59220	48497	30019	11434.8	39176.1	0.036878	0.152827	20559.3
3/200/5/15/5/10/10/255:	2	61678	59220	48497	30019	11434.8	39176.1	0.036878	0.152827	20559.3
4/200/5/15/5/10/10/255:	2	61678	59220	48497	30019	11434.8	39176.1	0.036878	0.152827	20559.3
1/200/5/15/4/10/10/255:	4	87461	84131	70474	42795	17928.7	55542.3	0.044736	0.166266	17492.2
2/200/5/15/4/10/10/255:	4	87461	84131	70474	42795	17928.7	55542.3	0.044736	0.166266	17492.2
3/200/5/15/4/10/10/255:	4	87461	84131	70474	42795	17928.7	55542.3	0.044736	0.166266	17492.2
4/200/5/15/4/10/10/255:	4	87461	84131	70474	42795	17928.7	55542.3	0.044736	0.166266	17492.2
1/200/5/15/5/15/10/255:	3	66337	63369	51823	36695	12033.1	42043.8	0.035503	0.152085	16584.3
2/200/5/15/5/15/10/255:	3	66337	63369	51823	36695	12033.1	42043.8	0.035503	0.152085	16584.3

Table 17 - Pump speed = 40 and line rate = 20000. Low detection level tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/20/20/255:	16	15	107	1555	21761	1561.15	-9.96836	0.998652	1.08989	0.882353
1/200/1/15/1/15/20/255:	15	15	107	1545	21531	1551.25	-9.97232	0.99871	1.08987	0.9375
1/200/1/15/1/10/20/255:	15	13	102	1460	20179	1465.8	-9.26084	0.998738	1.0892	0.8125
2/200/1/15/1/20/20/255:	12	12	91	1372	19129	1379.59	-11.0869	0.998566	1.08827	0.923077
2/200/1/15/1/15/20/255:	11	12	91	1362	18899	1369.7	-11.0909	0.998632	1.08825	1
1/200/1/15/2/20/20/255:	11	12	89	1357	18920	1365.06	-11.5534	0.998541	1.08812	1
2/200/1/15/2/20/20/255:	11	12	89	1357	18920	1365.06	-11.5534	0.998541	1.08812	1
1/200/1/15/2/15/20/255:	10	12	89	1347	18690	1355.16	-11.5573	0.998607	1.08809	1.09091
2/200/1/15/2/15/20/255:	10	12	89	1347	18690	1355.16	-11.5573	0.998607	1.08809	1.09091
2/200/1/15/1/10/20/255:	11	10	86	1277	17547	1284.25	-10.3794	0.998663	1.08748	0.833333

Table 18 - Pump speed = 40 and line rate = 20000. High precession tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
1/200/1/15/1/15/10/255:	109	8851	9447	11240	51384	5763.16	5812.47	0.314271	0.421718	80.4636
1/200/1/15/1/20/10/255:	109	8851	9449	11280	51995	5803.81	5811.69	0.317371	0.424906	80.4636
1/200/1/15/1/10/10/255:	109	8850	9436	11092	49330	5614.01	5813.86	0.302973	0.410095	80.4545
1/200/1/15/1/5/10/255:	109	8846	9403	10754	44079	5276.05	5813.9	0.277403	0.383755	80.4182
1/200/2/15/1/15/10/255:	1	15	118	1066	15878	1061.07	5.55339	0.99992	1.08637	7.5
1/200/2/15/1/20/10/255:	1	15	120	1106	16489	1101.73	4.77074	0.999952	1.08687	7.5
2/200/1/15/1/15/10/255:	1	15	124	1123	16627	1118.21	5.44663	0.999922	1.08702	7.5
2/200/1/15/1/20/10/255:	1	15	126	1163	17238	1158.87	4.66401	0.999952	1.0875	7.5
2/200/2/15/1/15/10/255:	1	15	118	1066	15878	1061.07	5.55339	0.99992	1.08637	7.5
2/200/2/15/1/20/10/255:	1	15	120	1106	16489	1101.73	4.77074	0.999952	1.08687	7.5

Table 19 - Pump speed = 40 and line rate = 40000. Low detection level tops ten.

minA/maxA/minH/maxH/ minW/maxW/lowT/highT:	0	0.01	0.1	1	10	k	m	R2	hP	lDL
2/200/1/15/1/20/10/255:	1	15	126	1163	17238	1158.87	4.66401	0.999952	1.0875	7.5
2/200/1/15/1/15/10/255:	1	15	124	1123	16627	1118.21	5.44663	0.999922	1.08702	7.5
1/200/2/15/1/20/10/255:	1	15	120	1106	16489	1101.73	4.77074	0.999952	1.08687	7.5
2/200/2/15/1/20/10/255:	1	15	120	1106	16489	1101.73	4.77074	0.999952	1.08687	7.5
1/200/1/15/2/20/10/255:	1	12	103	1058	15749	1057.77	-0.03162	0.999986	1.0864	6
2/200/1/15/2/20/10/255:	1	12	103	1058	15749	1057.77	-0.03162	0.999986	1.0864	6
1/200/2/15/1/15/10/255:	1	15	118	1066	15878	1061.07	5.55339	0.99992	1.08637	7.5
2/200/2/15/1/15/10/255:	1	15	118	1066	15878	1061.07	5.55339	0.99992	1.08637	7.5
1/200/1/15/2/15/10/255:	1	12	101	1018	15138	1017.11	0.750993	0.999995	1.08592	6
2/200/1/15/2/15/10/255:	1	12	101	1018	15138	1017.11	0.750993	0.999995	1.08592	6

Table 20 - Pump speed = 40 and line rate = 40000. High precession tops ten.