SNTC is one of the main software technology centers for Schlumberger; this center has offices in Oslo and Stavanger. In Oslo we concentrate on systems infrastructure, geology and modeling functionality while in Stavanger the focus is on seismic interpretation. More than 300 computer engineers and geosciences professionals from more than 25 nations work in this center creating an environment of innovation, sharing and achievement.

As a result, this tech center has created some of the leading software in the oil and gas industry (Petrel, Ocean, Frontsim), and SNTC work has evolved into many collaboration projects with key members of the software industry, universities and research centers.

As part of our research and innovation efforts, we execute every year a number of projects addressing some of our biggest challenges. These projects can cover different areas of the software development environment some examples of this are: the software building infrastructure, the requirements and project management system, the software documentation system, computer interaction, the software testing system and the software application itself where these projects involve the prototyping of new applications or technologies.

These projects are often executed by interns working in close collaboration with our engineers and lead by the most experienced members of our team as to achieve the expected results.

These innovation projects are developed in different environments, experienced programmers often work on the base code or on Ocean Plugins while less experienced ones work on other environments as our testing automation system and mathematics oriented people work on prototyping new ideas at Matlab or R.

Most of these innovation projects evolve into specific achievements; these achievements are often expressed in terms of new software application prototypes, new internal processes and tools or new patents and technical publications.

Some of the areas of our interest for projects involve

Computer interaction: new ways to communicate between user and computer, with emphasis on mobile devices

Pattern recognition: identification of features (sedimentary features, geological faults) on 3D seismic data

Geostatistics and Multipoint Statistics: Improve ways to simulate physical properties on a 3D volume

Geomathematics: simulation of stratigraphic and sedimentary processes, reservoir connectivity quantification, edge detection on 3D grids, Diffusion Maps For Latent Space Analysis

Fractals: characterize 3D distribution of properties, data compression, evolve into simulation algorithms

Time Series Correlation: Automatic correlation of well geologic records, expressed as electric well logs

Machine vision: be able to quantify similarity between images, be able to identify objects in images

Fluid flow simulation on the porous media: Improve existing algorithms.

You can see some examples of these projects below this letter.

As a conclusion, the ideas I have exposed above are only a small summary of the many projects we are handling at SNTC, we are deeply interested in mentoring students on these projects on all possible ways: as remote mentoring during course projects or in the frame of summer jobs, internships or making these innovation projects a part of their Master projects.

Best regards.

Sergio Courtade

Examples of some of these innovation  projects are described below

**Project: Conversion of Fractal dimension and Lacunarity computation to C# (and eventually to Ocean)**

Fractal dimension and lacunarity are textural attributes for images defined on the study filed of fractal mathematics.

As part of work done by an intern who worked with us this year we have a number of Matlab projects developing computation of Fractal dimension and lacunarity on 1,2,and 3D grids.

These algorithms have shown to work very well on 2D images as a consequence we would like to convert them into C# code and Petrel Ocean Plugins http://www.ocean.slb.com/Pages/default.aspx.

This project involves analysis of the existing code and i9dentification of improvements opportunities as much as rewriting the code in C#

This conversion will involve an extensive work involving

1- conversion of the algorithms from Mat lab to C#
2- Adaptation of the C# code to a Petrel Ocean plugin
3- Implementing the Ocean plugin and testing.

The main difficulties of this project will involve

1- loading the Ocean environment at a PC at Uppsala
2- Learning the Ocean basics as to be able to implement the plugin.

Some of the advantages of this project are:

1- Learning of Ocean for Petrel environment for the student
2- Creation of an Ocean environment at the Uppsala University
3- Creation of a plugin that would test a new technology.

This project still involves a challenge in the algorithmic part since existing algorithms must be reviewed and improvement opportunities analyzed. On the other hand there is also a big challenge in the programming part, this involves the implementation of those algorithms in a new programming language and the implementation of an Ocean for Petrel algorithm

**Project: Conversion of Hidden Markov Model (HMM) computation to C# (and eventually to Ocean)**

A Hidden Markov Model is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. The mathematics behind the HMM was developed by L. E. Baum and coworkers. It is closely related to an earlier work on optimal nonlinear filtering problem (stochastic processes) by Ruslan L. Stratonovich, who was the first to describe the forward-backward procedure

During this summer one of our interns, implemented HMM on well log time series. These algorithms have shown to work very well on well data as a consequence we would like to convert them into C# code and Petrel Ocean Plugins. These algorithms implement the prediction of facies based on shape of e-logs.

This conversion will involve an extensive work since a deep understanding of the involved theory is required for an accurate conversion of this code.

Some of the steps for this work will be

1- Conversion of the algorithms from matlab to C#
2- Adaptation of the C# code to a Petrel Ocean plugin
3- Implementing the Ocean plugin and testing.

The main difficulties of this project will involve

1- Understanding of the theory involved and accurate conversion from matlab to C#
2- loading the Ocean environment at a PC at Uppsala
3- Learning the Ocean basics as to be able to implement the plugin.

Some of the advantages of this project are:

1- Learning of Ocean for Petrel environment for the student
2- Creation of an Ocean environment at the Uppsala University
3- Creation of a plugin that would test a new technology.

This project involve some important challenge in the algorithmic part (although part of this this was already facilitated by the previous project) as much as in the programming part that involvsthe implementation of those algorithms in a new programming language and the implementation of an Ocean for Petrel algorithm.

**Project: Hidden Markov Model (HMM) for fault identification on 2D and 3D grids (seismic data)**

A Hidden Markov Model is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. The mathematics behind the HMM was developed by L. E. Baum and coworkers. It is closely related to an earlier work on optimal nonlinear filtering problem (stochastic processes) by Ruslan L. Stratonovich, who was the first to describe the forward-backward procedure

The objective of this project is to explore use of HMM workflow for fault identification on 2D and 3D grids (seismic data). A summer internship project explored the value of using HMM to classify well logs facies based on the well logs shape and the position on the sequence.

In this project we consider a similar analysis involving 2D and 3D data with the emphasis of identifying faults on 2D or 3D seismic data. This project will be carried out in Mathlab and it involves

1- Implementing an HMM for grid data (existing 1D code can be used)
2- Testing the implementation with existing grids

**Project: System to vectorize raster log files**

On Geosciences workflows, we often use Raster log files that are scanned images of time series (elogs) calibrated with depth information at some points. The idea for this project is to create a workflow and an algorithm to digitize the electric log curves (dark points in the scan image file) into log curves (vector time series with depth and value) and make all that in a practical working system.

Existing geosciences workflows support raster log files. This kind of files may have an extremely huge size and users possibly can get problems during log visualization and analysis such a slow system response, memory an CPU over consumption. It's also impossible to make any math or automation procession on these log files.

The idea for innovation project is to develop a method to translate raster logs to some digital representation. The digital form may be effectively used in Petrel and probably in other applications or as standalone information source.

Here's a possible project milestones

1- Develop a noise reduction algorithm for pre-digitizing process.

2- Develop algorithm(s) for optical log recognition. Algorithm will take a log raster and produce a some digital representation. This task may be accomplished as standalone utility / application using Java, .NET, mathlab, etc. There are some Open source OCR projects that may give an initial Idea or inspiration to start working on the problem. Task may be accomplished in several stages, for example as first stage log file may be converted to some raw digital data interpretation. Than this raw data is converted to one or many forms like Petrel or other industry recognized formats. Good opportunity for math people
3- Propose a data format to store digital log. It must be a platform independent, machine readable file format. Must allow to store large amount of data effectively.
4- Component(s) for digital log visualization. May be created for different platforms (win, web, mobile, etc.)
5- Petrel plugin for digital log data access
6- Petrel plugin for digital log visualization.

See examples of vectorizing logs at http://www.youtube.com/watch?v=2fOHSTki4uk and http://www.vextrasoft.com/?source=GoogleV&gclid=CKqCqurB8bICFUWnPAodyDgAJQ

**Project: Fractal compression used on geologic or geophysical data**

Fractal compression has become a conventional technology for compressing images. (http://www.cs.northwestern.edu/~agupta/_projects/image_processing/web/FractalImageCompression/) (http://en.wikipedia.org/wiki/Fractal_compression)

The objective of this project is to Investigate efficiency of using fractal compression on geologic 2D and 3D grids and geophysical data, compare with other compression techniquesan alternative objective for this project is to Investigate how to use to fractal compression in a reverse way as to devolve a facies simulation algorithm controlled by fractals mathematics (most of them are controlled by geostatistical mathematics).

The main challenges of this project are:

1- to investigate and develop fractal compression code
2- Explore the efficiency of this code this on geologic and seismic 2D and 3D grids,
3- Explore the use of fractal compression as a facies modeling algorithm

Example of a software application for facies modelingbased on geostatistics can be found here

**Project: Image Dictionary**

This project involves the creation of a dictionary of images that is compared automatically with a target image. The main challenge here is the comparison of images that can be similar but not exactly the same and the quantification of the similarity.

This similarity can be evaluated based on matrix operations, statistics, textural attributes or image segmentation and object identification techniques, all these techniques can be found at image analysis, image processing and machine vision text books.

The similarity comparison must be able to handle rotation and translation of the images

See here examples of image comparison software http://blogoscoped.com/archive/2008-06-10-n27.html and http://image-comparison.software.informer.com/

The main steps on this project involve

1- Investigate on image comparison techniques
2- Identify the algorithms to use
3- Code these algorithms
4- Build a library of images
5- Automatically compute the similarity between target image and the images in the library

**Project: Augmented Reality**

Apply image recognition techniques and associate specific images to specific files and processes,http://www.ted.com/talks/matt_mills_image_recognition_that_triggers_augmented_reality.html

This project involves associating images with specific files. This must run on mobile devices. The main challenges involve

6- Capture and handling of images
7- Image recognition algorithm (the main challenge)
8- Logic involving the association of the image and the files

**Project: Image segmentation and object classification**

Geosciences workflows involve the recognition of images by experts (for example recognizing features as rivers or lakes on a satellite image). Due to the many different existing processes for images the volume of information is growing faster than the number of experts able to look at them as a consequence we need an expert system able to identify some specific features on an automated way on images.

This process of transforming an image into recognizable features is covered by machine vision, branch or sciences, see also http://www.alexandria.nu/ai/machine_vision/introduction/#WhatIs

This project is aimed at creating or compiling the algorithms for machine vision and creating a system able to identify specific features in an image.

Project stages should be

1- Input an image to the software
2- Apply segmentation algorithm and classify the image into objects
3- Provide training information about objects
4- Automatically classify the objects

Example 1 show a car, the software identify doors windows, weels, Example 2 show a pictire of a cow, the software identifies head, legs, body See similar project executed on the previous year, build on top of that

Example 2 show a satellite image and the software is able to identify rivers and lakes.

**Project: Well correlation**

Well correlation is a very common process in geosciences workflows, it involves finding the points that are equivalent in the geologic time between time series, these time series are recorded as electric logs of rock properties at wells located at different locations.

This could be addressed simply by cross correlation algorithm but things get more complicate when some intervals could be missed by erosion or repeated by faulting, in this case other algorithms as for example dynamic time warping and other algorithms from speech recognition field are required

This project will involve the following steps

1- Analysis of specific cases
2- Analysis of different mathematical approaches
3- Implementation of them as code
4- Compare results

Investigate time series correlation of well data electric logs, use dynamic time warping http://cs.fit.edu/~pkc/papers/tdm04.pdf

The main challenge is to be able to automatically correlate different time series finding missing and repeated intervals. To achieve this you must find and tune a good algorithm, a lot of research and programming to be done, this is an existing industry challenge

**Project: From gradients to surfaces**

Outline:
Imagine we are surveying the topography of the surface of the earth through a remote sensing instrument located on a satellite in space. This instrument is highly accurate in many situations, but has the special characteristic that it cannot accurately calculate the height (above sea level) for locations on earth, but it can calculate the relative height differential between two neighbouring surface points very accurately, as long as the height differentials are not too large. When the differentials are too large we will observe an measurement error very similar to a phenomenon known as "phase wrapping", where we will get a totally wrong differential measurement. The threshold for when a measurement turns unstable depends on the frequency of the signal beamed from the satellite, where the choice of signal frequency is a compromise between accuracy, sensitivity to noise, distance to target, etc.

It should be noted that this problem is very related to 2D phase unwrapping, but a bit less constrained. In phase unwrapping theory we expect any measurement error to be a multiple of 360 degrees, (i.e. $err(x,y) = n(x,y) \times 2 \times PI$, where $n(x,y)$ is an integer). In our problem, as described here, $n(x,y)$ can be any real number (not complex, though).

We have some guidance as to when a differential measurement for a given position is accurate, through various heuristic estimation techniques. One of them is to investigate the signal strength of the reflected signal (the reflected signal amplitude), or changes in the spectra of the reflected signal. This heuristic is based on the thinking that surfaces with a large gradient (e.g. a steep hill slope, or a valley) will scatter the signal from the satellite widely, just like a diffractor, with little emitted energy returning to the satellite. Some high-gradient terrains, in particular cliffs, will however not have this scattering effect, as the prism effect is less expressed, so the estimated signal uncertainty is most of all a guide. In general, we can say that if the uncertainty measurement reports high uncertainty, then that is to be trusted. However, there are cases where the signal cannot be trusted, even when we have a low uncertainty estimate.

I.e. we are trying to solve for the following m x n unknown elevation values Z:

$$Z = z[x,y] \; ; \; 0 <= x <= m \; , \; 0 <= y <= n$$

Given the following (noisy) observations:

$$d/dx(Z) = P = z[x,y] - z[x-1,y] = p[x,y]$$
$$d/dy(Z) = Q = z[x,y] - z[x,y-1] = q[x,y]$$
$$0.0 < u[x,y] <= 1.0$$
$Min(Z) = 0.0$ - i.e. lowest point on the surface should correspond to mean sea level datum.

Here U = u[x,y] is the estimated uncertainty in the differential measurement for a point, where a small value indicates low uncertainty, and a high value indicates high likelihood for an measurement error. In general we can say that either the measurement Z is very accurate, or is totally wrong. Also, there are generally many more accurate measurements for a surface than erroneous one. This is based on the thinking that the earth is generally flat and continuous, but has scattered rifting and other discontinuities.

Obviously this is an over-determined system, as we have m x n unknowns (Z), and 2 x m x n noisy observations (P and Q). Dr. Jesse Lomask (Stanford University PhD thesis) proposed to solve a 3D version of this problem in a least-squares sense, minimizing in varying ways the differences:

d/dx(ZZ) - P

d/dy(ZZ) - Q

where ZZ is an estimate of Z. He also suggested doing weighted differences, based on the uncertainty U. As Dr. Lomask tried to solve this problem in 3D, which is much more difficult than to solve the 2D version of the problem, and because the 2D problem is recognized to be complex on its own, I would rather propose to study further the behavior of his method in 2D first, hence this project.

This image is the sum of observed P and Q of a surface circa 2.5 kilometer below the seabed in the Danish sector of the North Sea. Note how closely this image resembles a shaded image of the actual surface. From these differentials we can clearly get a general idea about the topology of the surface. To calculate the actual surface is though immensely difficult. Image courtesy of Hess Denmark.

Project Outline:

Given several example datasets; propose, implement and test several solutions to this numerical problem, including the 2D versions of Dr. Lomask´s least-squares approach. The algorithms should preferably be coded in C#, but code in Matlab, C++ or C is also ok. Evaluate results, and discuss benefits and drawbacks with each method. Time permitting, also test other methods known in the industry, e.g. coming from Liverpool University.

References:

[1] Lomask J.; "Seismic volumetric flattening and segmentation"; PhD Thesis, Stanford University, CA; 2006
http://sep.stanford.edu/sep/lomask/thesis.pdf

[2] Rocca; "Interferometry: Phase unwrapping"; European Space Agency, Lecture; Sep. 3, 2007 http://earth.esa.int/landtraining07/D1LB4-Rocca.pdf

[3] Phase Unwrapping project at Liverpool John Moores University: http://www.ljmu.ac.uk/GERI/90202.htm

**Project: Proposal for a short-term position for developing a flow-line based free-surface flow model for use in stratigraphic and sedimentary simulation**

**Mentor: Daniel Tetzlaff**

GPM (Geologic Process Modeler) is a project currently under development at Schlumberger Stavanger Research. Its purpose is to update existing research computer code and make it into a Petrel plugin for stratigraphiuc forward modeling. GPM simulates numerically the physical process of sediment erosion, transport and deposition under various conditions of paleotopography, sea-level change, paleoclimate, and sediment supply. It predicts the shape, geometry and properties of the sedimentary deposits that will be formed under a set of assumed conditions, much in the same way as a laboratory in sedimentation.

The conversion of the code into a Petrel plugin is already being carried out successfully within Schlumberger. However, the code also needs new options and modules to improve its modeling capabilities. One of the necessary improvements is a better method to model free-surface flow. Because of the need to model many thousands or even millions of years, GPM has special requirements and limitations: The flow cannot be modeled in full 3-D as a modern CFD program used in engineering would do because resources would be prohibitive. Therefore, GPM uses shallow-water equations in which only the vertically averaged horizontal velocity plus the flow depth are simulated in a quasi-2D manner over an arbitrary topographic surface. Even that is not straightforward because the flow surface and the "dry" areas of the model are frequently changing. Also, sources and sinks can be at arbitrary locations or over entire areas, and the flow must also be modeled over these areas. Presently there are two alternative methods to simulate flow in GPM: (1) A finite-difference method, used mainly for simulating steady flow (as in rivers and deltas), and (2) a particle method, used to simulate unsteady flow (turbidites and river floods).

To improve efficiency, we would like to add a third method that will be used to approximate steady or slowly varying flow. This method could for example be based on flowlines, implicit time discretization, approximations of the coverning PDEs, or a combination. The ideal candidate to carry out this work would be somebody with a strong background in mathematics and some background in computer science. Knowledge of fluid mechanics is a plus but is not an absolute necessity, as the requirements of this model differ significantly from those of traditional fluid mechanics models. It should be possible to complete the work in three to six months, under the supervision of one or two Schlumberger scientists.