NSERC CRD Progress Report

Oilsand Slurry Image and Video Analysis

Kevin Gordon   Martin Humphreys   Jui Wen Ting   Hong Zhang

***Background*—In oil sands mining, ore is broken down and combined with water to produce a slurry for hydrotransport. This slurry is fed into an extraction process to yield recovered bitumen and tailings. The recovery rate - the percentage of bitumen recovered from input slurry - is dependent on feed slurry properties as well as processing conditions. Small changes in recovery rate lead to large economical and environmental changes. Our industry partners at Syncrude have developed a sensing device capable of imaging the slurry during the extraction process with the goal of optimizing process control. Our role in the collaborative research and development grant will be to develop the image processing algorithms and real time tools to support this goal.**

I. Scene Modelling

Algorithms and methods capable of modelling the scene captured in the images is an important processing step. A scene can be decomposed into parts belonging to the background and others belonging to the foreground. The original sensor data (video) provided to us by our industry partners was a backlit scene with some immobile occluding objects (residue).

Maximum background

In a backlit scene where we assume no lensing effects, foreground objects may only attenuate light emmitted by the backlight. As a result the scene's background can be modelled by taking the element-wise maximum intensity value across the entire length of an experiment. Changes in lighting intensity or residue can be accommodated by applying the algorithm in a sliding-window manor. This was shown to be both an efficient and effective method of modelling the background, though it suffers the drawback of applying only to backlit scenes.

Learned background

Other more general background modelling algorithms were investigated, including low-rank / sparse decomposition and auto-encoding neural-networks. Low-rank/sparse decomposition methods were shown to work well, but their time complexity made their use in a real time system unfeasible. Research also showed the auto-encoding neural networks were effective in modelling the scene's background, and could be hardware-accelerated for real time application.

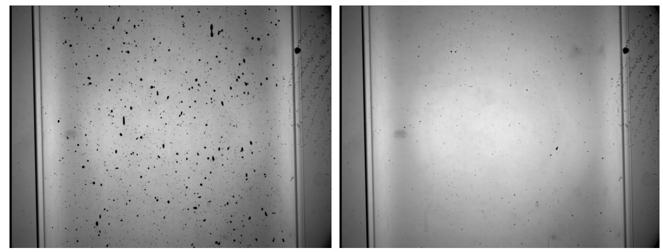

Figure 1: An unprocessed frame from early in an experiment (left), and the maximum background over the entire experiment (right).

Transmittance space

Experimentation with image processing software showed that dividing an image by the modelled background image improved the contrast. It was later found in [1,2] that sediment concentrations could be estimated from first-principles using the Beer-Lambert law. The transmittance T of a sample is the ratio of the incoming radiant flux Φ^i and the transmitted radiant flux Φ^t, which is in turn inversely proportional to the absorbance A of the sample.

$$T = \frac{\Phi^t}{\Phi^i} = 10^{-A}$$

For a fixed path length l and substance absorptivity ε, the concentration c of fines in the solution is proportional to the absorbance of the solution.

$$A = \varepsilon \int_0^l c(z)dz$$

Using the maximum background model we obtain an estimate of the total backlight received Φ^i by the sample, while the observed images yield a measure of the backlight transmitted Φ^t. The quotient of light transmitted over light received by the sample maps the input images to what we call *transmittance space* images, in an algorithm which is comparable to background subtraction. A desirable property of this operation is that division maps the input to a [0,1] range, as values in Φ^i are element-wise maximal. The division also accounts for varying backlight illumination: in theory, intensities in transmittance space are dependent upon physical properties of the sample; qualitatively the resultant images appear uniformly lit.

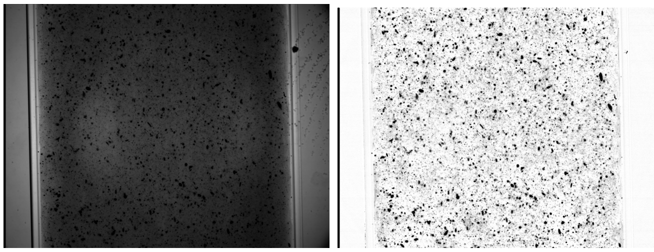

Figure 2: An unprocessed frame(left), the same frame processed by converting to transmittance space and accounting for fines (right).

Foreground modelling and analysis

The images resulting from background modelling and division are sufficiently processed that thresholding can be used to classify pixels as foreground or background in what we internally call the *binary mask*. Several standard thresholding algorithms were tested on the dataset where it was found that global thresholding techniques were as effective and more stable than local thresholding techniques.

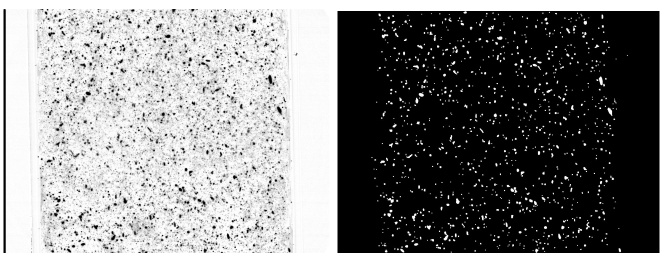

Figure 3: A frame after conversion to transmittance space (left), and its binary mask - the same frame after thresholding(right).

Connected components - a standard algorithm for gathering pixel region properties and statistics - is applied to the binary mask, after which the data from the entire *detection* portion of the pipeline is stored for later analysis.

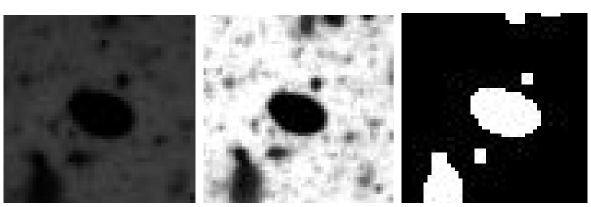

Figure 4: An unprocessed crop (left), the same crop after converting to transmittance space (center), the same crop after thresholding (right).

Deblurring

The imaging process has a small but noticeable edge blurring artifact due to the *point spread function (PSF)* of the lenses. This PSF can be approximated and used to de-blur the input images. This process is currently being researched to determine if the resultant region properties more accurately capture the ground truth data.

Fines modelling

Our Syncrude partners also indicated that it would be valuable to measure the concentration of *fines* - fine clay - in the slurry sample. Fines are too small to be imaged directly by the sensor, but their relative concentration can be estimated, with assumptions, using the Beer-Lambert law as above shown above. After applying a connected components algorithm to the image mask, all pixels will have been classified as foreground or background components. The background component consists primarily of fines and its mean intensity is proportional to the fines concentration.

Another method to determine fines concentration is to apply the background modelling algorithm to images in transmittance space. As long as the fines concentration is sufficiently low that some light is transmitted, a sliding window element-wise maximum algorithm yields an image that approximates the lower bound on fines concentration. This method has the advantage that the fines can be accounted for by again dividing out the modelled background, and is supported by the Beer-Lambert Law.

II. Classification

During ore processing it is known that a mixture of bitumen, clay and sand are input to the separation vessel, where valuable bitumen is separated from the sand and clay. Classification is crucial for obtaining class-based statistics, including particle size distributions, class ratios, and relative-class velocities.

It is important to note that this is a simplified view of the classification problem: in reality there are no known "classes".For example, objects can be bitumen surrounding sand grains, sand attached to bitumen droplets or even air bubbles.

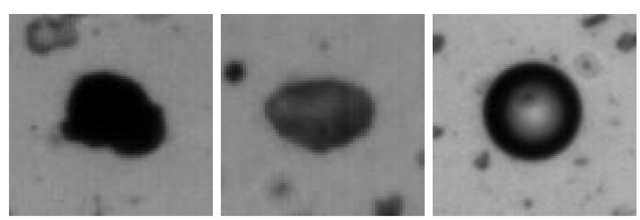

Figure 5: A crop of a bitumenous object (left), a sand object(center), and an air bubble(right).

Classical Features

An initial attempt at classification used mean detection intensity, motivated by bitumen being very dark and absorbing most of the incident light while sand transmits a higher proportion of the incident light. Using a simple threshold value to separate the two distributions resulted in poor performance when used prior to conversion to transmittance space, as the non-uniform lighting led to highly overlapping distributions; results were improved after conversion to transmittance space, but the method still had failure cases due to non-uniform intensity scaling with size.

Learned Features

A small hand-labelled dataset of roughly 5,000 objects were categorized into the set of Undefined, Unknown, Bitumen, Sand, or Bubble by our research team, and used to train a deep neural network. The network was able to learn salient *latent features* - vectors that encode useful information but whose elements do not have explicit meanings. These latent vectors were found to be useful in other stages of processing. The neural network was also able to correctly identify classes according to the human labelling with a high accuracy, significantly outperforming classification by intensity. A major drawback however was in the creation of the dataset: a human may not be able to correctly label detections: the scene is backlit and as such all objects are in shadow, and an object's class can be ambiguous.

III. Particle Tracking

Algorithms for tracking motion in video sequences can be broken into two categories: tracking by association and tracking by registration. In the former, the output of the detection stage of processing is fed to a matching algorithm that attempts to correctly match detections between frames. Algorithms in the latter category utilize features such as template matching and optical flow in an optimization process which attempts to find the correct parameters in a warping function to correctly estimate the between-frames motion.

Registration-based trackers

Particles in our dataset number in the many thousands per frame with areas ranging from sub-pixel to thousands of pixels. As a result of the sheer number of particles needing to be tracked in a video sequence, registration based trackers were deemed too inefficient and could suffer from problems such as identity switches and non-disjoint paths.

Optical flow

Dense optical flow, an algorithm that computes motion at the pixel level for a video sequence, was tested on the provided datasets as a candidate method to handle the number and density of particles. However, the resultant position deltas were found to be too noisy and unreliable to be useful for this application.

Tracking by association

Within tracking by association can be found global and online methods. Global methods attempt to find optimal matching between particles across an entire video sequence, while online methods only consider matches between two consecutive frames. An online tracker was investigated that used the Munkres (Hungarian) optimal matching algorithm. A global tracker that greedily found the k-shortest paths was also investigated and found to have reasonable time complexity and computed reliable tracks.

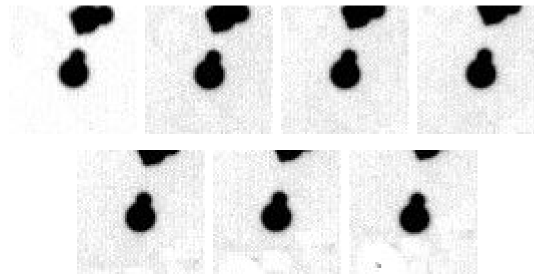

Figure 6: Seven correctly associated detection crops from seven consecutive frames.

Min-cost flow formulation

As an improvement to the greedy k-shortest path approximation algorithm, a min-cost flow graph formulation of the problem was investigated. This formulation correctly models disjoint paths and is convex in the number of tracks solved, allowing for both the optimal number of tracks as well optimal associations between detections.

Graph edge weight cost functions

Min-cost flow trackers represented the state of the art in multiple object tracking [3,4,5], with research focusing on finding cost functions to set graph edge weights. Our team investigated heuristic cost functions, including standard classical heuristics such as temporal slowness (photometric consistency) and appearance similarity (area, shape, intensity). Heuristic costs are easy to explain and work well in practice, showing the power of the min-cost formulation method.

Our team also investigated machine learning algorithms for setting edge weights. One method used the bottleneck feature weights from the classification stage, and set edge weights depending on a high-dimensional euclidean distance, sometimes used in combination with classical features such as position. This cost function outperformed the simple heuristics when the scene became crowded, when particles move fast, or when there were many false positive detections.

Another machine learning algorithm was developed that computed the edge weights directly when supplied with two detections' features, such as their crops and positions. This algorithm outperformed all others investigated and deteriorated more slowly under the presence of heavy crowding, fast motion, and high proportion of false positives.

For both learning algorithms a drawback is poor explainability and a concern that the algorithms could not generalize to unseen data. A large-scale experiment is slated for investigation in the coming months.

IV. Research Tool

The output from the detection, classification and tracking pipelines were analyzed in conjunction with lab-determined recovery rates. Our industry partners indicated that particle size distributions (PSD) and relative class velocities were most important, and features such as intensity, size, and circularity were also of secondary interest.

The research tool itself was developed in a cross-platform open-source environment using the python programming language. The tool consisted of a collection of utilities that could be used from the command line or inline to perform the various operations required by the pipeline (eg: detection, classification, tracking). The utilities could be rapidly swapped or modified to test different algorithms and output desired properties.

The size of the resultant dataset output by the pipeline, where detections easily numbered in the millions per experiment, led to the development of a lightweight database. This enabled quick and flexible data analysis by leveraging existing query languages and database datastructures.

V. Ground Truth

Simulator

A simulator was created that generated qualitatively similar videos to the provided datasets. The simulator was parameterized such that the research team could specify PSD's and class velocity profiles, while the ground truth was stored in a database at a per-particle granularity. The simulator was capable of using detection image crops from the original dataset, generating simple shapes as detections, and generating synthetic image crops using machine learning techniques.

Several corruption methods were developed within the simulator to test the robustness of the pipeline's algorithms. Corruptions included image noise (salt+pepper, gaussian noise), occlusions, lighting conditions, and injecting false positive detections, all of which could be enabled or disabled to create challenging video sequences.

Detection was determined to be essentially solved for the original dataset, and so the simulator was heavily developed as a tracking validation tool. Ground truth detections were supplied to the various tracking algorithms and the tracking performance evaluated using a standard multiple object tracking metric *MOTA* [6].

Labelling tool

A ground truth labelling tool was also developed, allowing for quick building of hand-annotated datasets and give statistical estimates of an algorithms performance. An operator used a keyboard and mouse to label the output of the pipeline processing via a web interface and these labels stored in the experiment database. Later, these labels could be included in queries to generate datasets or calculate statistics.

Physical ground truth

Another source of ground truth was provided by our industry partner in the form of particles - engineered glass beads - of known size. These particles were fed through the sensing equipment and resultant dataset used to evaluate the accuracy of computed particle size distributions, and to compare the similarity of various algorithm and algorithm combination outputs.

VI. Runtime tool

The runtime specification from our industry partner was that a single "experiment" be processed every 5 minutes. An experiment is comprised of roughly 1500 frames at 2336x1729 grayscale video with a typical video frame having detections numbering in the thousands. This represents a significant computational load, requiring continuously processing data at ~150MBps.

Modern computing architecture supports multiple physical processing cores, while the python programming language is inherently single-core, but supports multiprocessing. In order to leverage the available computing power and meet the target processing time, the research tool utilities were refactored into an **asynchronous multiprocessing software base**. An open source multiprocessing library (**mpyx**) was developed to facilitate this effort, along with a method of increasing inter-process communication more than tenfold above baseline.

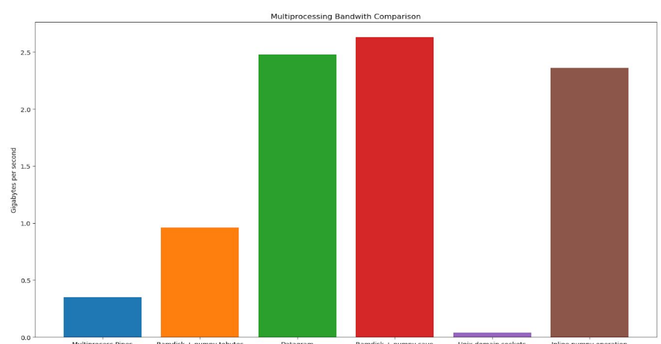

Figure 7: A comparison of various inter-process communication methods using the python programming language. The language default is shown on the far left in blue; the new method "Datagram" in green shows an increase in throughput.

In addition to better infrastructure, **efficient implementations of algorithm**s were swapped out for less efficient ones. Examples of this include the min-cost flow solver where state of the art C implementation was swapped for a python version, in the MOTA evaluation where a C implementation of the Munkres algorithm was used in place of a python version. **Hardware acceleration** is used for tensor processing wherever possible. A tool to **analyze processing bottlenecks** was used to tune the system, allocating more parallel processing to slower algorithms.

The simple database scheme developed in the research tool was also ported to a **large-scale database** version that could handle data points easily numbering in the hundreds of millions per day.

Development of a runtime tool was determined to also be a critical component for large-scale research interests. Today's machine learning systems require large volumes of data, and the infrastructure to efficiently handle it.

VII. Conclusion

Focusing on accurate scene modelling by high performance decomposition of the video frames into background, fines, and foreground components, followed by object detection, classification, and a min cost flow formulation of object tracking, we have successfully closed in on the target objectives of this research project. Our results are confirmed by the generation of synthetic data, which when run through our processing pipeline shows very high levels of correspondence with the generated ground truth and is robust to a wide range of parameters, failing only when the object velocities are set to relatively extreme values. Our team is excited to continue making refinements to our process to further improve accuracy and performance to achieve the project goal of high quality measurement of the properties of oil sand slurry and do so as efficiently and expediently as possible.

References

[1] Wu et al. *Laboratory Measurements of Sediment Flux and Bed Level Evolution in the Swash Zone, Coast. Eng. J. (2016)*

[2] Wu et al. *Sediment Flux Measurement at High Concentration Based on Image Analysis with Combined Illumination*. Coast. Eng. J (2014)

[3] Schulter et al. *Deep Network Flow for Multi-Object Tracking. CVPR (2017)*

[4] Zhang et al. Global data association for multi-object tracking using network flows. (2008). IEEE CVPR (2008)

[5] Leal-Taixé et al. *Learning by tracking: Siamese CNN for robust target association,* CVPR workshops (2016)

[6] Milan et al. *MOT16: A Benchmark for Multi-Object Tracking.* CoRR abs/1603.00831 (2016)