Computing Oil Sand Particle Size Distribution by Snake-PCA Algorithm

Baidya Nath Saha, Nilanjan Ray and Hong Zhang
University of Alberta
Edmonton, Alberta, Canada
{baidya, nray1, zhang} @cs.ualberta.ca

ABSTRACT

An important measure in various stages of oil sand mining is particle size distribution (PSD) of oil sand particles. Currently PSD is found by time consuming manual inspection. An effective automation of PSD computation can play a significant role in improving the mining process. Toward this goal we propose an algorithm (snake-PCA) to detect oil sands from conveyor belt images, which pose considerable challenges to automated analysis. The novelty in snake-PCA is as follows. First, snake-PCA evolves a number of snakes based on a novel variation of gradient vector flow requiring only a point as initialization. Oil sand is then detected by applying a threshold on PCA reconstruction error of a novel pattern image formed on each evolved snake. We show the discriminative property of the proposed pattern image here. Also, our detection experiments with snake-PCA produce a PSD matching well with a manually found PSD.

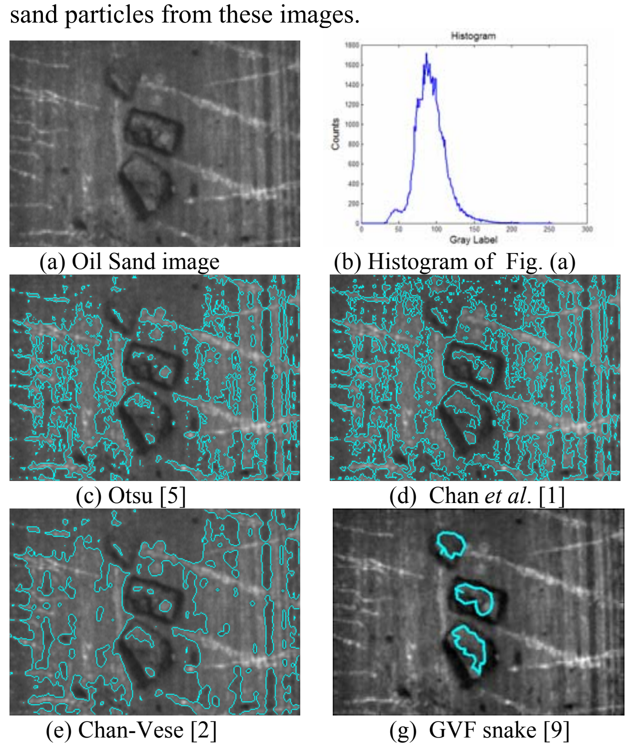
Index Terms**—**Gradient Vector Flow (GVF) snake, principal component analysis (PCA).

1. INTRODUCTION

The world's second-largest oil reserve lies under Alberta, Canada, in the form of oil sand. It is estimated that 174 billion barrels of oil of varying quality could be recovered from the sands, whereas in comparison Saudi Arabia, world's largest oil reserve, possesses 251 billion barrels of oil. Alberta's oil companies produce about one million barrels of oil per day in 2005 and it is expected to become double by 2010 [8].

Oil sand mining can benefit significantly from particle size distribution (PSD) of the oil-sand particles at various stages of the process in which oil is separated from sand. The current approach of detecting oil sand particles is to manually inspect hours of video that is extremely exhaustive, tedious, time consuming as well as prone to error. As oil is becoming scarce and more expensive, oil sand companies are seeking automated means to compute PSD online, where the images will be captured by a video camera mounted over the conveyor belt and looking down at the belt.

Toward automated PSD computation, we propose to detect the oil sand particles from conveyor belt images by an algorithm we call snake-PCA the description of which follows shortly. Oil sand image is unique, and to the best of our knowledge, little research work has been carried out till date. These images are often poorly illuminated and noisy with various kinds of background (conveyor belt) clutter. Moreover, oil sand particles come in a variety of shape, size and texture. The apparent brightness of the individual object varies from object to object. Most of the times, objects are mixed with dirt and fine materials. Additionally, since the mine operates 24 hours a day, and the oil sand material needs to be analyzed outdoors, varying lighting and weather conditions play a significant role in their appearance in the image. The aforementioned factors constitute the main challenges to automatically segmenting the individual oil sand particles from these images.



(a) Oil Sand image (b) Histogram of Fig. (a)
(c) Otsu [5] (d) Chan *et al*. [1]
(e) Chan-Vese [2] (g) GVF snake [9]

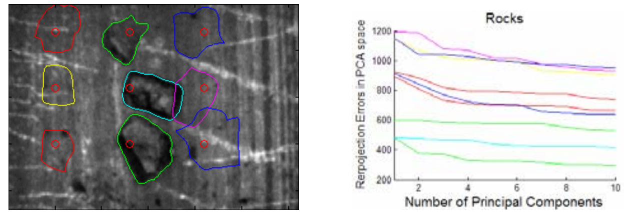
Fig. 1: Results of different well-known segmentation methods on oil sand image.

Fig. 1(a) shows an example of an oil sand image. The associated intensity histogram shown in Fig. 1(b) is seen to be unimodal. As a result, Otsu's global thresholding method [5] fails to segment oil sand particles as shown in Fig. 1(c). Chan *et al.'s* [1] well-known locally adaptive variational thresholding technique does not serve our purpose either as shown in Fig. 1(d). Next, we made an effort using the well-known Chan and Vese's [2] region based level set algorithm. However, as shown here in Fig 1(e), the algorithm fails to detect the oil sands, essentially because there is not much difference in gray level between objects and corresponding background pixels distribution. We also applied Xu and Prince's [9] Gradient Vector Flow (GVF) snake after manually placing an initial snakes inside oil sand particles. The GVF snake results, even with manual initializations near oil sand boundaries, are also seen unsatisfactory (see Fig. 1(g)). These comparisons of some of the state-of-the-art segmentation algorithms on oil sand images immediately reveal difficulty of automated analysis here.

To overcome the difficulty of automated segmentation, here we utilize a variation of GVF snake algorithm. In this variation, we impose a Dirichlet boundary condition while computing the GVF force field. This boundary condition encourages the initial snake to grow, so that if the initial snake is inside the object, eventually it will expand and delineate the object boundary. We also note that oil sand particle boundaries are usually characterized by dark-to-bright intensity transitions going from inside to outside. (Although, the reverse transition is not very rare). We accommodate this prior information inside governing partial differential equation that computes the force field. A version of similar force field computation has been reported in Ray *et al.* [6]; however, we additionally use the directional gradient information here and we evolve the snake and compute the modified GVF in an interleaved fashion until convergence. This proposed modification makes the result of the snake segmentation quite insensitive to initial snake location, as demonstrated later. Thus to detect oil sand particles, we place a number of seed points on an image and evolve snakes from each of these seed points by the proposed snake algorithm.

Note that after the snakes are fully evolved, we need a validation mechanism in place that will discriminate between the snakes delineating the oil sand particles and the snakes clinging onto the conveyor belt and other clutter. We view this validation mechanism as an abnormality detection process and propose to utilize a novel pattern image constructed from each evolved snake. Our proposed pattern image is essentially formed by the area of the original image covered by thickening the snake contour with an equal width both inside and outside. Note that this pattern image is an annular ring. We cut this annular ring at a location and unfold it to a rectangular image for computational convenience. Next, we need a pattern matching technique to work with this pattern image. Motivated from the idea behind face recognition using eigen faces, fisher faces and laplacian faces [4] we utilize Principal Component Analysis (PCA) to reduce spatial dimensionality of this pattern image and use the PCA reconstruction error to detect oil sand particles.

2. SNAKE-PCA ALGORITHM



(a) (b)

Fig. 2: (a) Seeds (red circles) and evolved snakes. (b) PCA reconstruction errors of these snakes.

2.1 Proposed algorithm

Snake-PCA algorithm consists of three sequential steps:

Step 1: Place seed points at uniform grid spacing over the image and evolve one snake using modified gradient vector flow from each seed point (see Fig 2(a)). The proposed snake evolution algorithm is described in Section 2.2.

Step 2: When all the snakes stop their evolution, form pattern images (see Section 2.3 for pattern image formation), one for each snake.

Step 3: Do pattern matching (rather, abnormality detection) by principal component analysis (PCA). The snakes associated with pattern images producing PCA reconstruction errors below a threshold value are retained and recognized as oil-sands. Snakes delineating oil sand particles have typically lower PCA reconstruction errors (Fig 2(b)).

2.2 Proposed snake algorithm

Given a seed point (x_0, y_0) for the snake, our proposed snake algorithm first builds an edge map:

$$\rho(x,y) = \max\left(0, \frac{(x-x_0)(\partial I/\partial x) + (y-y_0)(\partial I/\partial y)}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}\right),$$

where $I(x,y)$ is the image. The edge map ρ indicates only dark-to-bright intensity transitions on the image as seen from the point (x_0, y_0). Next, we compute a force field $(u(x, y), v(x, y))$ by:

$$\partial u/\partial t = \exp(-\rho(x,y)/K)\nabla^2 u - (1-\exp(-\rho(x,y)/K))(u - \partial\rho/\partial x),$$

$$\partial v/\partial t = \exp(-\rho(x,y)/K)\nabla^2 v - (1-\exp(-\rho(x,y)/K))(v - \partial\rho/\partial y),$$

subject to the Dirichlet boundary condition:

$$(u(x,y), v(x,y)) = \mathbf{n}(x,y), \text{ for } (x,y) \in \partial\Omega,$$

where $\partial\Omega$ is the initial snake contour (in our case, a small circle centered at (x_0, y_0)) and $\mathbf{n}(x, y)$ is the unit outward normal to the initial snake at (x, y). K is a user defined parameter controlling the degree of smoothness of the snake external force field (u, v). After computing (u, v) we use them to evolve a snake from the initial contour as in [9]. In fact we perform snake evolution and (u, v) computation in an interleaved fashion– first compute (u, v), then evolve a snake with (u, v) until convergence, next compute (u, v)

again with previously evolved snake contour as $\partial\Omega$, and so on, until finally there is no appreciable change in the area enclosed by the snake.

Fig. 3 shows that our proposed snake alogithm is quite insensitive to snake initialization compared to other snake algorithms. It has a broad capture range and it can capture contour from a seed point located inside an oil sand.

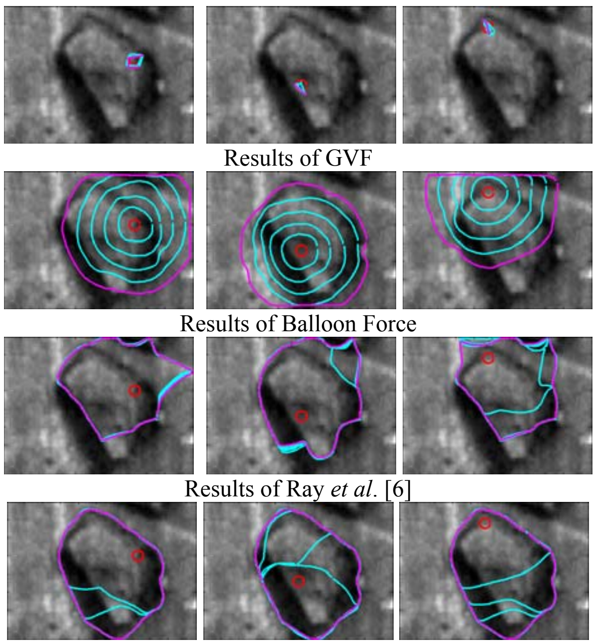
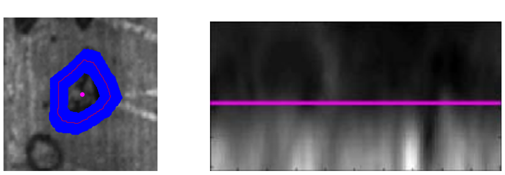


Fig. 3: Initial, intermediate and final snakes are in red, cyan and pink colors respectively.

2.3 Pattern image generation

We choose a couple of images arbitrarily as training images and place a seed point (shown by a dot inside the oil sand particle in Fig. 4(a)) inside the oil-sand particles and we evolve our snake from each seed point. When the snake stops evolution around the boundary of the oil sand particle we consider an annular ring or band along the snake as the pattern image. This annular region forms in such a way that the snake passes along the medial axis of this annular ring. We unfold this annular band and make a rectangular pattern image (as in Fig. 4(b)) for computational convenience. The vector formed by the pixel gray value of the pattern image makes training data set for principal component analysis carried out in the next step.



(a) (b)

Fig. 4: Proposed pattern image formation.

The existence of the prominent dark to bright transition (as in Fig. 4(b) where the top of the line is dark and bottom of it is relatively bright) across the boundary or contour of the oil-sand particle characterizes an oil-sand particle. Any dark to bright transition similar to the training pattern image in the annular band across any evolved snake determines the presence of oil sand particle in the test image.

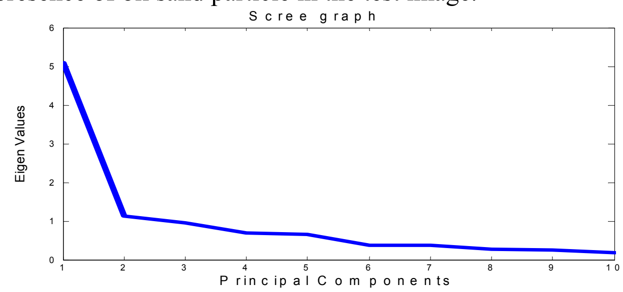
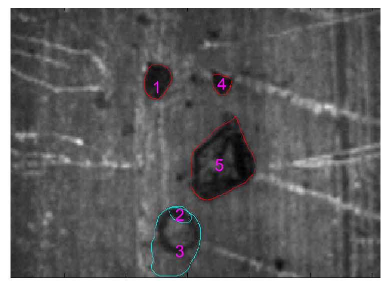
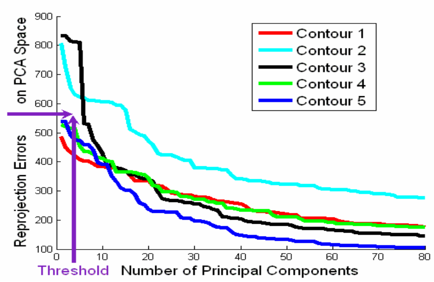


Fig. 5: Scree plot.



(a)



(b)

Fig. 6: (a) Snakes 1, 4 and 5 correspond to oil sands, while 2 and 3 correspond to clutter. (b) PCA reprojection errors corresponding to 1, 4 and 5 are smaller than those for 2 and 3, for the first few principal components.

2.4 Principal Component Analysis (PCA)

We have generated a training set of 51 oil sand contour samples from 10 oil sand images. Our training dataset for PCA consists of a two dimensional matrix where each column consists of the the column vector formed by the pixel colors (gray values) of a pattern image. Primarily we have carried out Bartlett's sphericity test [3] on the training dataset and it shows that spatial dimensionality reduction through PCA is meaningful.

Our experiments on training data shows that only the first two principal components can be retained, since their eigenvalues are greater than unity [7]. Extended Bartlett's test [3] shows that only the first principal component corresponding to the maximum eigenvalue shown in Scree plot (Fig. 5) could be retained and it explains maximum percentage of variance (50% of the total variance).

The following instance also confirms that the larger principal components have better discriminative properties than the smaller components (Fig. 6). We have also carried out Kolmogorov-Smirnov test on PCA reconstruction errors associated with training pattern images and it shows that it follows a normal distribution. We have used 25th, 50th, 75th percentile and mean of this training distribution as the thresholds to find PSD on the test set.

3. RESULTS AND DISCUSSIONS

We have used 100 oil sand images as the test set sampled randomly from an inspection system. In Step 1 of the snake-PCA algorithm, we place 30 seed points at uniform spacing on each of 100 test images. Next, we follow steps 2 and 3 of the algorithm. Fig. 7 shows the performance of step 3, where each snake is classified as oil sands or clutter. Note that recall increases but precision decreases as PCA reconstruction error threshold increases. Detections on 4 images at threshold 770 are shown in the Fig. 8.

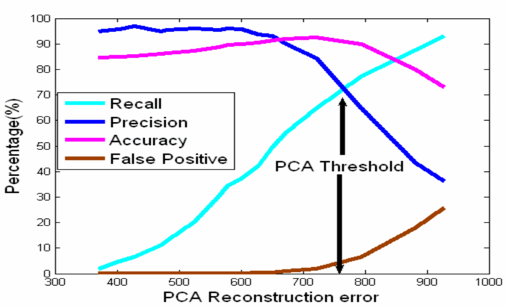


Fig. 7: Recall, precision, accuracy and false positive rates

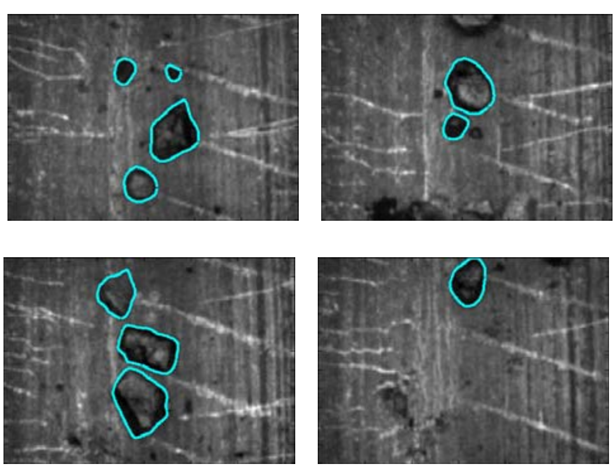


Fig. 8: Results of the proposed Snake-PCA algorithm

Finally the PSDs computed on the test set are shown in Fig. 9. Here we set the error threshold values at 25th, 50th, 75th percentiles and the mean of the training set error distribution. The Kolmogorov-Smirnov test shows that the test set PSDs obtained when thresholds are set at 50th percentile and the mean are similar to the distribution of the manually found oil sand particles on the test 100 images.

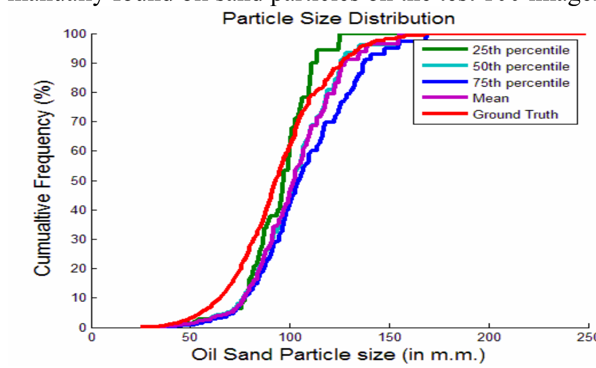


Fig. 9: Oil sand particle size distributions.

4. FUTURE WORK

We have proposed a combination of snake and PCA algorithm on oil sand images. This is a general framework to segment and delineate objects boundaries from poorly illuminated images. We look forward to implementing this proposed method in other application domains, such as different medical/biomedical engineering applications.

Acknowledgements: This work is supported by NSERC discovery grant, iCORE, Syncrude Canada, Ltd., Matrikon and University of Alberta.

REFERENCES

[1] Francis H. Y. Chan, F. K. Lam, and Hui Zhu, "Adaptive Thresholding by Variational Method," IEEE *Transactions on Image Processing*, Vol. 7, No. 3, pp. 468-473, March, 1998.
[2] Tony F. Chan and Luminita A. Vese, "Active Contours Without Edges," *IEEE Transactions on Image Processing*, vol. 10, No. 2, pp. 266-277, February, 2001.
[3] Paul E. Green and J. Douglas Carroll, *Analyzing Multivariate Data* , The Dryden Press, Hinsdale, Illinois, 1978.
[4] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi and Hong-Jiang Zhang, "Face Recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 3,pp. 328-340, March, 2005.
[5] N. Otsu, "A threshold selection method from gray– level histogram," *IEEE Transactions on System Man Cybernatics*, Vol. SMC-9, No.1 , pp. 62-66, 1979.
[6] N. Ray and Scott T. Acton, "Merging Parametric Active Contours Within Homogeneous Image Regions for MRI – Based Lung Segmentation," *IEEE Transactions on Medical Imaging*, Vol. 22, No. 2, February, 2003.
[7] Alvin C. Rencher, Methods of Multivariate Analysis, John Wiley & Sons, Inc. , 1995.
[8] Technology Review: Photo Essay: Dirty Oil, December, 2005 found at http://www.technologyreview.com/Energy/16059/.
[9] Chenyang Xu and Jerry L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Transactions on Image Processing*, Vol. 7, pp. 359-369, 1998.