ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 3, March 2016

Optimal Sand Mining Approach Using Image Processing Technique

Asir Sejoenath.R[1], Pratheep Kumar.P[2], Santhiya.C[3], Nathiya.R[4]

Assistant professor, ECE, N.S.N College of Engineering & Technology, karur, India. [1]
UG Scholar, ECE Department, N.S.N College of Engineering & Technology, karur, India [2]
UG Scholar, ECE Department, N.S.N College of Engineering & Technology, karur, India [3]
UG Scholar, ECE Department, N.S.N College of Engineering & Technology, karur, India [4]

Abstract— **Sand mining is a process of extracting sand from beaches, dunes, or open pits. Due to the mining, it has a serious environmental effect such as soil erosion, loss of biodiversity, contamination of groundwater Segmentation. and formation of sinkholes. This gives a detailed explanation about the ill effects of mining on water resources and air quality. This system includes optimal sand mining approach using image processing This system detects vehicle using static images and calculating the amount of sand carried to prevent malpractices of sand smuggling. Different from traditional methods, which use machine learning to detect vehicles, this method introduces a new contour mapping model to find important "vehicle edges" for identifying vehicles. The sand can have various colors under different weather and lighting conditions. The proposed new color segmentation model has excellent capabilities to identify sand pixels from background, even though the pixels are lighted under varying illuminations. The detected amount of sand is checked against the maximum set threshold value specific to the recognized vehicle. Experimental results show that the integration of Hough features and color-image segmentation.**

***Index Terms*—Sand mining,Segmentation,Features.**

I. Introduction

The issue of Illicit sand mining is one of the major problems in developing nations. Weaker government policies for mining leads to illegal sand mining in rural areas. It is difficult to stop such malpractices. The progress of technology has provided ways to monitor this issue. To put a check on such malpractices this paper presents an application developed using computer vision methods. The proposed system recognizes vehicle type and segments the visible amount of sand present on the vehicle which is then compared to the maximum set of amount carried by the vehicle.

II. Overview of our Project.

We present a new system to detect vehicles from static image and calculate the amount of sand it is carrying. It is a real time based two-step process. They are,

- Vehicle classification
- Sand detection

In the first step the system takes the side view image and Finds the number of tires and classifies the vehicle on the basis of number of tires. This information is transferred to the database to fetch data containing the amount of sand the vehicle can carry. For most of the cases the vehicles are weighed on a weighing machine to keep monitoring the amount of sand being mined and transported. But there can be situations where the weighing systems may be tampered with. As a solution, we take the acquired images and segment out the amount of sand visible above the container. The area of the extracted part is calculated and is matched against the set values from the database specific to the type of vehicle.

The parameters for every vehicle includes,

- Maximum allowable weight of sand that can be carried
- Maximum allowable visible sand area over the vehicle's side panels as explained in Figure 1

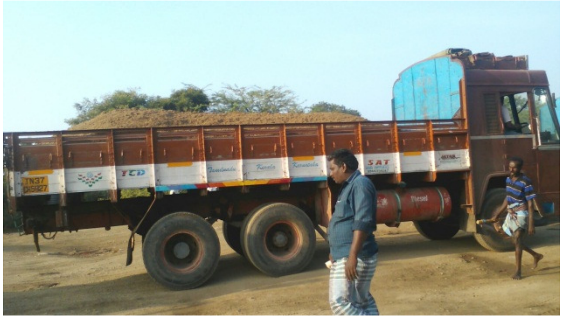


Figure 1: Explains the positioning of side panels and the visible area of sand concept.

To ensure that the application is user friendly, the freedom of obtaining images from any camera with resolution greater than 5 Megapixels is provided. The system is developed using Open Computer Vision Libraries. The working platform must have minimum requirements as shown in Table 1, thus the entire application can be ported onto any smart-phone. The presented system is simple, cost effective and computationally fast.

Processor	>1GHz
Camera resolution	>8Megapixels
Web Connectivity	To transmit captured images for off-site processing

Table 1: Minimum System requirements for our application.

$$G_h(x) = \frac{e^{\frac{-|x|}{4h^2}}}{4\pi h^2}$$

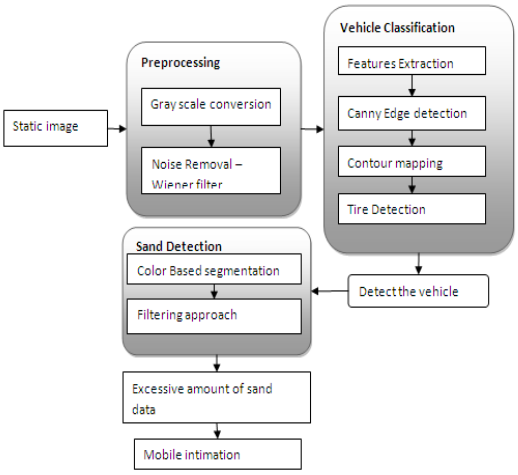
$$u - G_h * u = -h^2 \Delta u + o(h^2)$$

To overcome these complications and to make the applications work with lesser resources we present a vehicle detection system which is customized to this particular situation. Since the user has the liberty to capture the image, the side view and the back view are the inputs given by the user to the system. After this the images are processed using a preprocessing algorithm which equalizes the light conditions of the image. The image is then processed through edge detection and Hough detection algorithms to detect and classify the vehicle on the basis of number of tires and the container size both of which are predefined within the system.

The second part involves extraction of sand from the images. Image segmentation and object detection is by using the colour based segmentation.

III. Proposed System

Both the image-processing systems have been designed for real-time implementation with minimal resource usage. This section is divided into two sub-sections, namely vehicle classification and segmentation of sand. Both the sections include in-depth explanation of the algorithms used with illustrations. Every step explained in the section is accompanied with proper advantage of it as well as its effect in functioning of the entire system.



A. *Vehicle Classification*

One of the main aims of this paper is to detect a vehicle from static images. Rather than using any Machine learning methods, a simpler approach of using edges and contours is used for detection. Since the final implementation is to be carried out on smart phones the algorithm needs to use minimal resources while providing best results. Figure 2 shows the flow diagram of the process of vehicle detection.

The smoothing requirement is usually expressed by the positivity of the kernel:

The paradigm of such kernels is of course the Gaussian kernel *x*. In that case, G_h has standard deviation *h* and it is easily seen that: A similar result is actually valid for any positive radial kernel with bounded variance, so one can keep the Gaussian example without loss of generality. The preceding estimate is valid if *h* is small enough. However since the noise reduction properties in this case depend upon the fact that the neighborhood involved in the smoothing is large enough, so that the noise gets reduced by averaging. So in the following we assume that *h* = *k*, where *k* stands for the number of samples of the function *u* and of the noise in an interval of length *h*, *k* must be much larger than 1 to assure the noise reduction. The effect of a Gaussian smoothing on the noise can be evaluated at a reference pixel *i* = 0. At this pixel,

$$G_h * n(0) = \sum_{i\in I} \int_{P_i} G_h n(x)dx = \sum_{i\in I} \epsilon^2 G_h(i) n_i$$

where we recall that *n(x)* is been interpolated as a piecewise function, the P_i square pixels centered in *i* have size 2 and $G_h(i)$ denotes the mean value of the function G_h on the pixel *i*. Denoting by *Var(X)* the variance of a random variable *X*, the additivity of variances of independent centered random variables yields:

$$Var(G_h * n(0)) = \sum_i \epsilon^4 G_h(i)^2 \sigma^2$$

$$\simeq \sigma^2 \epsilon^2 \int G_h(x)^2 dx = \frac{\epsilon^2 \sigma^2}{8\pi h^2}$$

After the process of equalizing the brightness of the image the output image becomes blurry. The Wiener filter can remove the additive noise and invert the blurring. A Low pass filter is not suitable in this case since it is very sensitive to additive noise. The Wiener filter is proposed to optimize the trade-off between de-noising and inverse filtering. Noise removal in previous steps may lead to blurring of edges. Experiments show that a symmetric low-pass filter of size 7x7 with standard deviation of 0.5 efficiently restores the texture pattern. This pre-processing provides a de-noised pre-segmented image. To summarize (and convert to 2D), given a system:

$$y(h,m) = h(n,m) * x(n,m) + v(n,m)$$

where * denotes convolution and *x* is the (unknown) true image *h* is the impulse response of a linear, time-invariant filter *v* is additive unknown noise independent of *x*, and *y* is the observed image. We find a deconvolution filter g to x:-

$$x'(n,m) = g(n,m) * y(n,m)$$

$$G(w_1,w_2) = \frac{H^*(w_1,w_2)S(w_1,w_2)}{|H(w_1,w_2)|^2 S(w_1,w_2) + N(w_1,w_2)}$$

The equation for *G* can be rewritten as:

$$G(w_1,w_2) = \frac{|H(w_1,w_2)|^2}{H(w_1,w_2)[|H(w_1,w_2)|^2 + \frac{N(w_1,w_2)}{S(w_1,w_2)}]}$$

So the Wiener filter has the inverse filter for *H*, but also a frequency-dependent term that attenuates the gain based on the signal to noise ratio.

Edges characterize boundaries and therefore carry fundamental information of an image. Edges in any image consist of areas with strong intensity contrasts i.e. a jump in intensity from one pixel to the next. As shown edge detection in an image significantly reduces the amount of data to process, while preserving the important structural properties in an image.

One of the best available algorithms present is the canny edge detection algorithm. To further enhance the edge information a Sobel filter is applied. The Sobel operator [16] performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude at each point is found. In the Sobel operator a pair of 3x3 convolution masks is used, where one estimates the gradient in the x-direction (columns) and the other in the y-direction (rows).From these two images.

$$Edge\_gradient(G) = \sqrt{G_x^2 + G_y^2} \qquad Angle(\theta) = \arctan\frac{G_y}{G_x}$$

We find edge gradient and direction for each pixel as follow
After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient. The next step removes any background edges which act as noise for the result. For this, we need two threshold values. Any edges with intensity gradient more than the maximum threshold are sure to be edges and those below minimum are sure to be non-edges, so are discarded. Contour data are an integral part of the application. Contours can be explained simply as curves joining all the continuous points (along the boundary), having same color or intensity. It helps in segregating the real data points from unnecessary background noise. They further enhance the quality of shapes that are to be detected as can be seen in Figure 3.

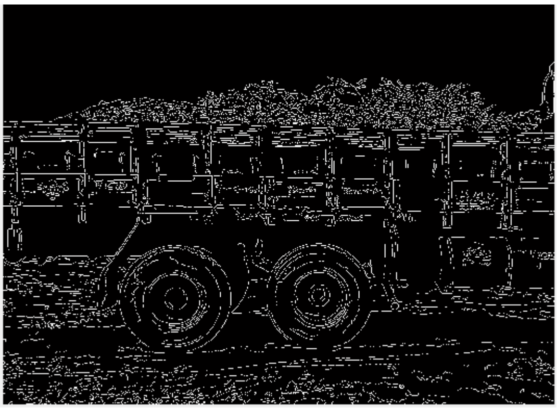


Figure 3: Contour Map.

Once the edge information is extracted, the contour mapping of the image is done to further remove any unnecessary edges and to map the vehicle. Using the contour map the vehicle dimensions can be extracted and classification can be carried out.

Hough Transform uses parametric representation for the family of all circles and transforms each figure point in the obvious way [17]. The classical Hough transform was developed to identify lines in the image, but later the Hough transform extended to identify the positions of arbitrary shapes, most commonly circles or ellipses. Due to imperfections in either the image data or the edge detector, however several points or pixels on the desired curves are missed. Also there are spatial deviations between the ideal circle and noisy edge points are obtained from the edge detector. For these reasons, it is often non-trivial to group the extracted edge features to an appropriate set of circles. Our procedure addresses this problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects as can be seen from Figure 4. A circle is fully defined with three parameters: the center coordinates (a, b) and the radius (R):

$$x = a + R\cos\theta$$

$$y = b + R\sin\theta$$

As ϴ varies from 0 to 360, a complete circle of radius R is created. So the transform function looks for triplets of (x, y, R) from the image. Therefore, we need to construct a 3D accumulator for Hough transform, which would be highly ineffective. So, we use a trickier method, Hough Gradient Method which uses the gradient information of edges. However, the method also includes hidden circles which are meant to be perceived as circles.

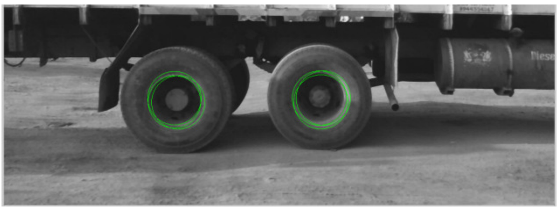


Figure 4: Tire Detection.

In general, without image de-noising, the algorithms tends to extract too many circular features. So, to be more successful, the preprocessing applied is a crucial step.

B. *Segmentation of Sand*

The first step is *Image acquisition*. Two images of the vehicle carrying sand is captured. These images include the side and rear view of the vehicle. The purpose of obtaining these two views is to ensure a dual check process on the quantity estimation of the sand carried by the vehicle. The next step is *Downsampling* the images. The 2D area of the sand in both the views is matched against a set threshold value. This threshold value is subject to the type of the truck and the view. To keep a common base for testing the area the image is down sampled to a width of 500 pixels and height of 500 pixels. The threshold values are obtained from a set of training images and the process is explained later in the paper. The whole reason for down sampling is to create an access image that is a miniaturized duplicate of the optical resolution of master scan. If the image signal and the image noise had similar properties, averaging neighboring pixels in order to reduce the resolution would not improve the signal-to-noise ratio. However, signal and noise have different properties. But in this paper the major reason is to increase signal to noise ratio [18].

The step followed by *Downsampling* the images is *Application of morphological operations*. The segmentation algorithm used in this application is K-Means clustering. To ensure perfect clustering, sharpness of the image has to be reduced and components must emerge out as more connected. This is facilitated by the two morphological operators, namely, dilate and erode. The importance of this step can be understood from Figure 5. The sub-images from Figure 5.(a) to Figure 5.(e) are experimental outputs with different combinations of morphological operations. Figure 5.(a) is the output of clustering without any morphological pre-processing operation, Figure 5.(b) was the output when only dilation operation was applied, Figure 5.(c) was the output when only erosion operation was applied. A region of interest highlighted in the sub-images (a) to (c) of Figure 5, depicts the inefficient clustering outcome. Even the order of using the two operators is important. In the same figure, it can be observed in (e) that smooth clustering output is obtained when dilate operation is followed by erode, while on the other hand, irregular clustering output is obtained when erode operation is followed by dilate operation.

The kernel size used for both the morphological operations is of width 3 px and height 3 px. Another important parameter related to these operations is the number of iterations. In the application the value for this parameters is 1, as it can be observed from Figure 6 that as the number of iterations increase, the clustering of image deteriorates. In figure 6, the number of iterations is 1 in (a), 3 in (b), 5 in (c) and 7 in (d).

After performing the morphological operations, the next step is Noise removal. This step is to make sure that the noise, which the morphological operations are not able to remove, does not affect the application's output. The noise is removed by applying a mean filter of kernel width 5 px and height 5 px. Experiments were conducted on the filter's kernel size value and the results are shown in Figure 7. The optimal kernel for the application was chosen to be of width 5 px and height 5 px. The experiments had symmetric kernels, it was 3 in Figure 7.(a), 5 in 7.(b), 7 in 7.(c), 9 in 7.(d) and 11 in 7.(e). It can be clearly seen that the output with filter kernel size 3 had noise and the ones with kernel size greater than 5 are excessively blurred with loss of important data.

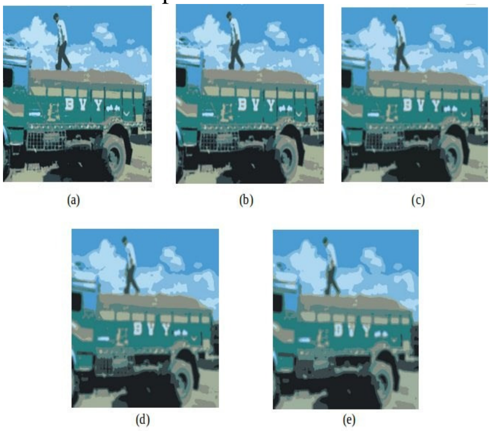


image is transformed into HSV format. The step following is Thresholding. In this step a pre-obtained threshold level for the H, S, and V values is put against the transformed test image. The purpose is to extract all the sand pixels that are over the vehicle. To ensure that the sand pixels other than the required, i.e., over the vehicle's tires or that on the ground level are detected, the thresholding is carried over a specific region of interest. The region of interest (rectangular) for a side view has properties shown in Table 2.

image is transformed into HSV format. The step following is Thresholding. In this step a pre-obtained threshold level for the H, S, and V values is put against the transformed test

image. The purpose is to extract all the sand pixels that are over the vehicle. To ensure that the sand pixels other than the required, i.e., over the vehicle's tires or that on the ground level are detected, the thresholding is carried over a specific region of interest. The region of interest (rectangular) for a side view has properties shown in Table 2.

The values are obtained over experimentation. The threshold values have been set after understanding the different gradients in the color of sand at different regions, weather conditions and lighting conditions. The set value functions in such a way that all the pixels with H between 0 and 160, S between 0 and 50, V between 0 and 200 are extracted out and labelled as sand pixels.

The final step in the proposed approach is Correction Algorithm. The aim of this step is to remove the undesired and false detections by exploiting the connected nature of the sand present in the image. An all-region sweep stores the data of the pixels which are most illuminated as per the clustering algorithm. The sweep is divided on the basis of rows. A particular row is selected and in that row, any pixel in the threshold range is labelled as white and the rest as black. For any row if *[white/(white+black)] <= 0.48,* then all the pixels in that row labelled as sand pixels are rejected.
This approach removes 93% of the false positive patches and provides a stable detection as it can be seen in Figure 11. In the Figure 11.(a), the region of false detection is highlighted, and Figure 11.(b) is processed out by applying the Correction algorithm.

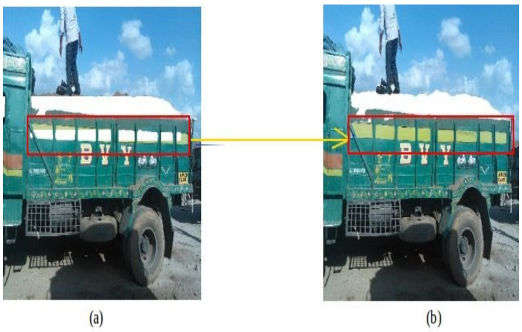


Figure 11: Advantage Of The Correction Algorithm.
Figure 12: Sand Segmentation Process.

The entire segmentation algorithm is shown in Figure 12 as a flow-chart format.

In order to analyze the performance of our proposed method, various static images captured under different weather conditions and lighting conditions were used. The approach presented in this paper supports direct measurement of detection and tracking, facilitates iterative algorithm development, and provides important diagnostic feedback. The flowchart in Figure 13 presents the overall process from capturing the image to the end output. The process begins by determining the vehicle class according to its sand carrying capacity. As shown in Figure 14 after detection the trucks are classified into one of the following classes: Large, Medium or Small. After the process of classification the input image is fed into the sand detection algorithm.

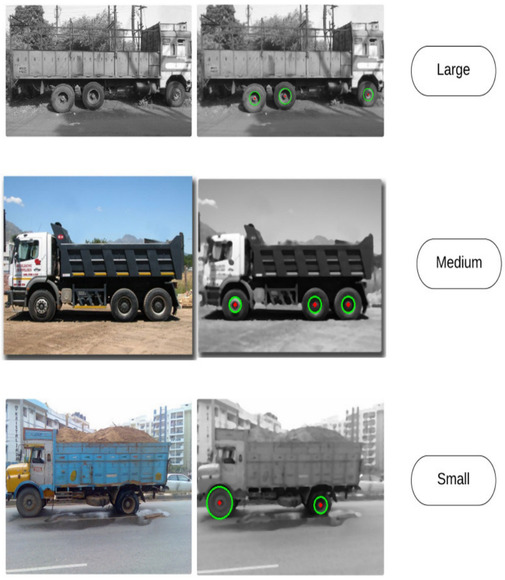


Figure 14: Vehicle Classification.

The robust sand detection results are shown in Figure 15. The preset threshold values determine whether the vehicle is overloaded or not. Experimental results suggest that in sub-images 15.(a) and 15.(c) the vehicle is carrying sand beyond the maximum capacity while in 15.(b) and 15.(d) it is well within the limits.

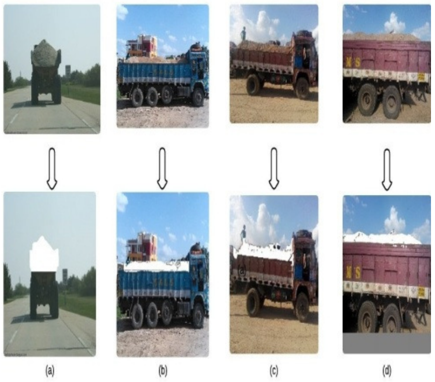


Figure 15**:** Results Of Sand Detection

IV. Conclusion

In this paper a vehicle classification and sand detection algorithm is presented. The system presented in this paper provides a layer of abstraction to the user which simplifies the whole process of implementing and using the application. The user is provided with a black box which can take, as input, the side and front view of the vehicle and give the vehicle type and the amount of sand present as output. The algorithm developed here is error free even though a lot of variability in the capturing conditions acts as hindrances. Since it can well record different changes of vehicle appearances, desired vehicles can be very accurately and effectively detected from static images. However there is still a lot of scope for development. The algorithm presented can be used to develop applications with minimal hardware requirements. The customizability of such an application is endless and such systems can be employed to several other industries.

REFERENCES

1. AKASH DEEP SINGH,ABISHEK KUMAR ANNAMRAJU " A NOVEL SYSTEM TO MONITOR ILLEGAL SAND MINING USING CONTOUR MAPPING AND COLOR BASED IMAGE SEGMENTATION"[MARCH 2015]
2. .Ojos Negros Research Group. "Sand Mining Impacts" USENET: http://threeissues.sdsu.edu/three_issues_sandminingfacts01.html. [April. 29, 2015].
3. Shi Na, Guan Yong and Liu Xumin, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," in Third International Symposium on Intelligent Information Technology and Security Informatics, 2010, pp. 63-67.
4. Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines," presented at the IEEE Int. Conf. Digital Signal Processing, Santorini, Greece, Jul. 2002.
5. M. Bertozzi, A. Broggi, and S. Castelluccio, "A real-time oriented system for vehicle detection," Journal of Systems Architecture, pp. 317–325, 1997.
6. Tzomakas and W. Seelen, "Vehicle detection in traffic scenes using shadow," Tech. Rep. 98-06 Inst. fur neuroinformatik, Ruhtuniversitat, Germany, 1998.