Abstract: This paper introduces an algorithm for the estimation of size distribution of crushed aggregate based on 3D-image of a moving conveyor belt obtained by a laser profilometer. The aim of the study is to obtain measurement information that can be utilized mainly for the automatic control of crushers. Preferably the computation should be done in real-time, i.e. faster than the velocity of the conveyor belt. This sets demanding requirements on the computational efficiency of the algorithm. The computation speed is the primary criterion for the algorithm developed and successful results of that development are presented in this paper.

Keywords: Size distribution, Estimation algorithms, Laserprofilometry, 3D measurement, Crushed aggregate, Granular material

1. INTRODUCTION

Crushed rock aggregates are granular rock materials used in construction. The main uses are as a base material of foundations for buildings, roads and railroads, and as raw materials for composites, like concrete and asphalt. Every year, over 3.5 billion tons of aggregates are consumed within the EU, amounting as a total value of 20 billion Euros [UEPG (2010)]. In order to be suitable for use, all aggregates must meet certain specifications in terms of size, shape and strength. Currently, these variables can only be measured with once-a-day laboratory analysis, and thus, this information cannot be used for automatic control.

Due to the low level of system automation, the typical crushing process is either run by an operator (fixed plants) or is simply running autonomously without any operator interaction, while having only safety automation running (mobile plants). Even in operator controlled applications the information obtained from the process is so inadequate that operations, other than the ones keeping the plant in running state, are not being performed. Thus, the fulfillment of given product quality criteria is currently obtained by means of a conservative plant design, i.e. by increasing the average values so much that specifications are met, even in the presence of large disturbances. It is clear that such an arrangement causes significant losses on productivity, energy efficiency and raw material usage. It is also clear that plant performance could be improved by automated control in order to optimize the size distribution after each crusher of a multi-stage crushing plant. Additionally, the information about size distributions in different stages of crushing could also be useful to the operator for manual control.

The crushing process includes several rapidly occurring feed-material-based disturbances [Evertsson (2000); Ruuskunen (2006); Bearman and Briggs (1998)], of which feed material size distribution and feed material moisture are the most influential factors [Itävuo (2009)]. The main reason why presented control schemes [Evertsson (2000); Moshgbar and Parkin (1994)] have not yet been implemented in real world situations is the lack of robust online size distribution measurements that can withstand harsh operating environments common in the production of aggregates. Though there exists a number of commercial applications for measuring size distributions of crushed aggregate (e.g. VisioRock, Split-Online, WipWare), none of them seem to be very well suited for online measurement in demanding environments such as mobile crushing plants. Approaches that typically utilize photographs [Guyot et al. (2004)] instead of 3D-images are known to experience problems in adapting to changing sunlight and, more importantly, in withstanding dust and dirt particles. Other approaches are too slow in terms of computational efficiency [Thurley and Andersson (2007); Thurley (2009)] as the size distribution estimate is needed every second, or they simply require too much attention in terms of calibration, which makes them unsuitable under conditions of ever-changing processes and environments.

Therefore, this paper introduces a fast algorithm for estimation of size distributions of crushed aggregate based on a 3D-image of a moving conveyor belt obtained by a laser profilometer. The aim of the study is to obtain measurement information that can be primarily utilized for the automated online control of crushers. Thus, the computational efficiency has been prioritized over achieving exact measurement accuracy, making the presented algorithm more suitable for less than high-end comput-
ers and demanding environments such as aforementioned. For automated crusher control, instead of complete size distribution information, a simpler scalar characteristic of the distribution was chosen. This is the grading equivalent size, $G_{eq}$, for which the definition is given in Equation 13, and the accuracy of the estimate is measured by the error of the estimation of $G_{eq}$.

This paper is organized as follows. Section 2 presents the definition for particle size and the measurement device used. Section 3 describes the developed algorithm(s) in detail and results are presented in Section 4.

2. PRELIMINARIES

The next subsections present the definition for particle size, and the particular measurement device used in this work, respectively. The definition for particle size is investigated to justify some assumptions used in image processing.

2.1 Particle size

Particle size for crushed aggregate is readily defined by screening. The standard screening is done by using standardized test sieves with square apertures [The European Committee for Standardization (CEN) (1995)]. It is intuitively clear that this definition allows very different particles to be categorized in the same size class. The most important factor affecting the screening is the width of a particle together with the height of a particle, which, together, characterizes a particular cross-section. The definitions used for width and height in this work are the second longest and shortest sides of a bounding hyperrectangle, respectively. Thus, the screening is less affected by the length of a particle (i.e. the longest side of the bounding hyperrectangle [The European Committee for Standardization (CEN) (2008)]), which becomes obvious considering how a very long but thin bar is able to pass through a small aperture. Similarly, a long, curved particle is able to pass through a small aperture. In practice, however, as the length of a particle increases to unrealistic levels, the probability of passing through a small aperture decreases as the particle is very unlikely to reach a vertical position perpendicular to the screen. It is fair to assume that no such particles exist among randomly shaped rock particles. Thus, an assumption can be proposed that passing through an aperture is not affected by the length of a particle.

2.2 Measurement device

The measurements of crushed aggregate on a conveyor belt are performed by Ruler E [SICK IVP (2005)]. Ruler E, by SICK IVP, is a laser profilometer that uses laser triangulation to analyze heights of objects passing under a laser-line. Each measurement is done by capturing one image at a time of the laser-line projection on the object. The output of the camera is not a photograph; instead the laser line is detected from the image and height and intensity values are recorded into a profile. Then, each profile shows the cross-section of the measured surface viewed from an angle. Since the geometry of the measurement setup and optics of the camera can be measured, a calibrated profile can be computed by a coordinate transformation. The computation is done through a software-based method provided by SICK IVP. The principal measurement set-up is illustrated in Figure 1.

By combining the captured profiles, the device produces a $512 \times 1024$ pixels (although configurable) 3D-image with approximately 1 mm resolution in both $x$- and $y$-directions. For a typical conveyor belt velocity of 0.5–1 m/s [Eloranta (2008)], a new 3D-image is provided every 0.5–1 second. An example measurement is shown in Figure 2. The conveyor belt has advanced approximately 400 mm during data collection. Height of a measured point is indicated by color, and value zero (blue) represents a missing data point.

3. ALGORITHM DESCRIPTION

Two separate algorithms were developed based on the method used for image segmentation. The structure of both algorithms is as follows:

- Search for particle edges
- Threshold the result to obtain a binary image of partial edges
- Segment the edge image
- Analyze the segmented image

Aforementioned steps are presented in detail in the next subsection.
Logarithm of combined differences

200 400 600 800 ... represent the center points of the objects. When the objects are marked, they can be grown from

Fig. 3. The result of equation (5) applied to the example image.

3.1 Search for edges

Searching for the particle edge points in a 3D-image is related to finding the local minima of a function and thus computing the image gradient seems natural. A number of definitions and methods [Kaartinen (2009); Sonka et al. (1999)] for calculating the image gradient exist usually relying on the convolution of (a) chosen kernel(s). The method used here was chosen on the basis of computational efficiency.

First- and second-order differences in pixel coordinates are computed in the x- and y-directions, respectively, as:

\[
\begin{align*}
\Delta_x f(i, j) &= f(i + 1, j) - f(i, j) \\
\Delta_y f(i, j) &= f(i, j + 1) - f(i, j) \\
\Delta_x^2 f(i, j) &= f(i + 2, j) - 2f(i + 1, j) + f(i, j) \\
\Delta_y^2 f(i, j) &= f(i, j + 2) - 2f(i, j + 1) + f(i, j)
\end{align*}
\]

For computational efficiency, no diagonal computations are used. It should be noted that these differences do not correspond to actual height differences, unless the differences in the x and y coordinates are equal to 1. The actual height differences could be obtained by dividing the first and second order differences in height by \(\Delta x\) or \(\Delta y\), and \((\Delta x)^2\) or \((\Delta y)^2\), respectively. In the chosen approach this does not save computational time and nor does it provide anything of interest as the actual differences are inconsequential. Additionally, this would require the interpolation of missing image points to avoid division by zero.

The first- and second-order differences in both directions are then combined by:

\[
\Xi(i, j) = (\Delta_x f(i, j))^2 + (\Delta_y f(i, j))^2 + (\Delta_x^2 f(i, j))^2 + (\Delta_y^2 f(i, j))^2
\]

The result obtained by Equation (5) for the example image is shown in Figure 3. The values of the result can now differ drastically as differences next to missing points have very large values.

Fig. 4. The cumulative distribution function of combined differences.

3.2 Threshold the edge image

A threshold is applied to the image of combined differences in order to obtain a binary image of the edges. The cumulative distribution function of the result of Equation 5 for the example image is shown in Figure 4. The logarithmic axis illustrates the wide range of values. It was observed that the cumulative distribution function is almost equal for all measurements, even for different product gradings. This removes the need of a variable threshold value, which is favourable for computational efficiency. Thus, a constant threshold value of 1 on the logarithmic scale was chosen based on observations of the distribution functions for different product gradings.

3.3 Image segmentation

A. Watershed transformation

The watershed transformation can be conveniently described by considering the analogy of landscape and rain. Water will find the swiftest descent until it reaches some lake or sea. Mathematically, each lake or sea can be thought as a regional minimum. The boundaries separating different regions are called watersheds or watershed lines which are typically local maxima. In the watershed transformation, the regional minima are called catchment basins and the whole landscape or image is partitioned into separate regional minima. [Sonka et al. (1999)] Partitioning is complete when all catchment basins are filled or analogically, enough water is poured to cover the whole landscape. In the result, the watersheds separating regions have a unity width and are given an equal label, e.g. 0, to denote that they are not part of any region. Each region is given a unique label. Another way to describe the method is to explain that it is performed by locating particle markers and reconstructing the particles by watersheds. Marker extraction resembles human behavior when indicating objects. A human typically points to an object, thus defining a single point or a small set of inner points instead of giving a set of outline boundaries [Sonka et al. (1999)]. In this approach, the markers are extracted by computing the distance transformation of the edge image, i.e. every point in the image is assigned the value of its euclidean distance to the nearest edge point [Danielsson (1980)]. Thus, the local maxima of the distance transformation represent the center points of the objects. When the objects are marked, they can be grown from
the markers using the watershed transformation described above.

B. Simple morphological segmentation Another approach for segmentation was attempted to improve the computational efficiency. The partial edge information as obtained in the first implementation was used as the starting point. This is already close to a segmented image and easily interpreted by the human eye. To compute the actual segmentation, the partial edge information has to be improved. This is done by morphological closing of the image [Soille (2003)]. The edge points are, in this approach, not used for extracting markers for the watershed transformation. Instead, they are used as region boundaries. In order to ensure that all edge points are connected, the closing of the image is done excessively which may introduce a small error to the size estimation (i.e. the sizes of the particles are underestimated). The complement of the image is next computed by:

\[ f^c(i, j) = 1 - f(i, j) \]  

where \( f^c \) denotes the complement of image \( f \). Now, the edge points are given a value of 0 and particles a value of 1. The segmentation is then easily computed by assigning a unique label to each connected component, i.e. an area having adjacent pixels with a value of 1.

In both segmentation methods, particles touching the edge of the image are discarded.

3.4 Analysis of the segmented image

Size estimation In image processing, it is often desired to evaluate the size of an object. One fundamental approach is to fit some geometric shape around the object. The choice of the geometric shape is typically case dependent. For this case, a rectangle is chosen due to its relation to square apertures used in manual screening and, more importantly, due its high computational speed. To further improve the computational efficiency, we omit the computation of particle orientation, which is needed to compute the best-fit rectangle (BFR) [Wang et al. (2006)]. Instead, we compute the simple bounding rectangle (Ferret box) with sides parallel to the x- and y-axes. Thus, the lengths of the sides of the BFR are given by:

\[ l_1 = \max(x) - \min(x) \]  
\[ l_2 = \max(y) - \min(y) \]

where \( x \) and \( y \) belong to a single region in the segmented image. Now, the size of the particle is estimated as the smaller side of the BFR by:

\[ d = \min(l_1, l_2) \]

which is also called the width of the particle. The length of the particle is estimated by:

\[ e = \max(l_1, l_2) \]

It is assumed that the movement of the conveyor belt and gravity causes the particles to align in a way in which the smallest dimension of a particle is parallel to the z-axis. The assumption is based on practical observations. Thus, the height information is not needed to estimate the size of a particle.

Mass-weighing The standard method of presenting screening results is through a mass-weighed histogram or more commonly the cumulative sum of it. Thus, we need to estimate the mass of each segment or simply the volume of each segment as the density of each particle can be assumed to be equal. The 3D-image data would allow seemingly accurate estimates for volume to be computed by summing the z-values representing the height. However, increased computational efficiency can be achieved by making an appropriate assumption about the geometry of the particles. When the particles are assumed to be ellipsoidal, the volume of the particle can be computed as:

\[ V(d, e) = \frac{4}{3} \pi d^2 e \]

With this choice, the height of the ellipsoid is assumed equal to the width of the ellipsoid. A somewhat more accurate assumption would be to allow the height of the ellipsoid to differ from the width. The height could be obtained by finding the largest height value within the segment, \( z_{\text{max}} \), and the volume of the ellipsoid would then become:

\[ V(d, e) = \frac{4}{3} \pi d e z_{\text{max}} \]

It was observed however that the previous assumption about the height being equal to the width gives sufficient results and is naturally faster to compute. This simplification is motivated by shape conversion factor considerations presented in [Woodcock and Mason (1987)].

4. RESULTS

In this section, the estimates given by the algorithms are compared to the results of manual screening for three different product gradings. The measurements were performed at a Metso Mining and Construction Technology research facility. Results are shown graphically in figures 5, 6, and 7. In the tables, the algorithm utilizing watershed transformation is denoted by algorithm A and the alternate algorithm is denoted by algorithm B. Key figures in the tables are the computational time, CT, the estimated grading equivalent size, \( G_{eq} \), the error of the estimated \( G_{eq} \), and the sum of squares of errors, SSE, which is computed using the cumulative distribution function. The computations are performed on a desktop computer with Intel Core 2 Duo processor (1.86 GHz) and 4 GB of RAM. The measurement consists of 524,288 3D-points (approximately on a 0.5 meter long section of the conveyor belt). The grading equivalent size [Ruu5kanen (2006)] is given by:

\[ G_{eq} = \sum_{i=1}^{n-1} \left( \frac{s_{i+1} - s_i}{100} \sqrt{f_i \cdot f_{i+1}} \right) \]

where \( s_i \) is a passing percentage of the discrete, cumulative distribution function at \( f_i \) and \( n \) is the number of sieves. Due to being a scalar, \( G_{eq} \) is a simple characteristic of the size distribution function. Thus, \( G_{eq} \) has been chosen to be the measurement information to be utilized in crusher control related to this work. However, \( G_{eq} \) is obviously not a complete representation of the size distribution as uniqueness is lost.

The results are very accurate except for the last case with the product grading 0–200, which is representative of
5. CONCLUSION

In terms of accuracy, both proposed algorithms perform extremely well unless there exists a significant amount of very small particles, which are not visible in the measured 3D-image. In terms of computational efficiency, both al-
ACKNOWLEDGEMENTS

This work was funded by Tekes, the Finnish Funding Agency for Technology and Innovation. We thank the staff of Metso Mining and Construction Technology Research and Test Center at Tampere for performing the laserpro-
filometer measurements and the manual screenings.

REFERENCES


