UTRECHT UNIVERSITY

MASTER'S THESIS

High-resolution grain size analysis using photos

Author:
C.F. Bakker

Supervisor:
Dr. T.D. Price
J.W. Bosma, MSc

Faculty of Geosciences
Department of Physical Geography
Earth, Surface and Water

October 25, 2023

Abstract

Grain size is an important factor that influences sediment transport at beaches. Traditional methods of obtaining grain size properties, such as sieving or settling tube, are time-consuming and labour-intensive, which limits the ability to map large sections of a beach in the spatial and temporal domains. This leads to significant simplification of the spatio-temporal variability of the beach sediment, whereby an entire section of the beach is defined by a single property.
In this study, grain size distributions are analysed using pyDGS. This method involves capturing images of sediment samples, significantly reducing fieldwork while enabling higher-resolution sampling.

This study addresses the main challenge when using various grain size analysis methods, where the areal (pyDGS) and volumetric (sieve) measures are not directly comparable. To effectively utilise pyDGS as an alternative to mechanical sieving, we introduce a novel correction method. In contrast to the traditional correction method, which relies on correction exponents obtained through additional sampling methods, our method utilises information extracted from the images. It involves a three-step process for all grain size classes within an image: 1) calculate the total number of grains in the photo frame, 2) estimate the volume of each grain class, and 3) compute the mass of each grain size category by multiplying the number of grains by their volume and density. In this study, the grain shape was estimated and calibration was done with sediment samples that were photographed and sieved.
In total, we analyzed 380 sediment photographs, with 43 sediment samples also sieved for direct comparison. The results of our novel correction method were compared to those of the traditional correction method. Our study reveals that the errors associated with D_{90}, D_{84}, D_{50}, and D_{25} are similar. However, for D_{16} and D_{10}, our novel method achieved a 4% reduction in RMSE and a 7% reduction in MAE for D_{16}, and a 24% reduction in RMSE and a 19% reduction in MAE for D_{10}.
During the most recent field campaign, we photographed a total of 24 cross-shore rows. This extensive dataset demonstrates the potential of our approach, which would be unfeasible with traditional grain size analysis methods.

The main contribution of this research is the introduction of a novel correction approach, enabling the conversion of area-based measurements into volume-by-weight measurements. This approach eliminates the need for generalisation and offers a more precise correction method for individual images. Making it particularly valuable for larger beach sections. Despite challenges related to grain shape uncertainties and sediment variability, the new method provides a more effective way of correcting the images for improved accuracy.

While pyDGS does exhibit significant errors that prevent it from providing absolute values, it cannot fully replace mechanical sieving. However, its high spatial resolution capabilities offer a valuable tool for studying the temporal evolution of grain size at beaches. This method provides insights into relative patterns in grain size evolution, enhancing our understanding of beach sediment dynamics and contributing to more effective coastal management strategies.

Acknowledgements

I'd like to begin by expressing my gratitude to **Dr. Timothy Price** for affording me the opportunity to embark on this research project. I am very thankful for his helpful feedback and guidance throughout this study.

Furthermore, I thanks to **Jorn Bosma** for his role as my second supervisor. His constructive feedback and valuable input greatly contributed to the enhancement of my research methodology.

Additionally, I would thank both my supervisors for nominating me to speak at the **NCK days 2022**. I extend my thanks to the organizers of **NCK** for providing me with the opportunity to present my work to the coastal research community. This was a very fun experience and I have learned alot.

To my parents, brothers, and friends, your encouragement and belief in me have helped me to finish my work. I appreciate all your support throughout this journey. Special thanks to everyone who has shared photographs of sand from their beach visits with me.

Contents

viii

List of Figures

List of Tables

Chapter 1

Introduction

About three-quarters of the sandy coast in the Netherlands consists of dunes (Rapportendatabank, 1990), which serve as a natural barrier for low-lying lands. Since 1990, the Dutch government has implemented a policy of dynamic maintenance. A base coastline (BKL) was established beyond which the retreat of the coastline should not exceed. Annually, the position of the coastline is established and areas experiencing significant erosion are rebuilt using sand nourishments.

The approach of using sand nourishments allows for the development of the beach-dune system through natural processes, e.g. wave-driven or wind-driven processes. Historically, areas that were considered more vulnerable or important were protected with hard-engineering solutions, e.g. sea dikes, groins and breakwaters. However, in recent years, there has been a shift toward nature-based solutions (soft-engineering). For example the Sand Engine (Stive et al., 2013), the Hondsbossche Dunes (Wittebrood et al., 2018) or the Houtribdijk (Steetzel et al., 2017).

The Prins Hendrik Sand Dike (PHSD) is also an example of this switch in coastal protection mentality. When the Prins Hendrik Dike failed the safety requirements it was decided to build a beach-dune system in front of the old dike (HHNK and Witteveen+Bos, 2017). The design combines coastal protection with the development of a new ecosystem (Fordeyn et al., 2019).

The impact of the grain size distribution on the amount of aeolian sand transport of nourished beaches has been shown in previous studies (van der Wal, 1998; van der Wal, 2000). Similarly, grain size influences the sediment transport by wave-action (Yoshioka et al., 2009). Despite the fact that the grain size distribution plays an important role in the sediment transport process, the understanding of its spatial and temporal evolution is limited. This is largely due to the time-consuming and labour-intensive nature of the current sieving method for grain size analysis, which results in relatively few studies being conducted (Prodger et al., 2016; Horn and Walton, 2007; Nugroho and Putra, 2018). The limitations of this method also restricts the spatial and temporal resolution of these studies.

Currently, beach sediments are often characterised by the median grain size and the sorting of the sediment, e.g. a well-sorted medium sandy beach. However, beach sediments do change over time and space. This is often ignored in morphological- or sediment transport models for simplicity, where the sediment is often assumed to be homogeneous and constant over time, which overlooks the actual complex sedimentary conditions.

This study aims to explore a photo-optometry approach for sediment distribution analysis. This method, which involves taking pictures of the sediment, will reduce field work and allow for higher resolution sampling. The study will utilise an algorithm called pyDGS (Buscombe, 2013), which has been adapted for this purpose.

The Prins Hendrik Sand Dike, a poorly sorted sandy beach, was used as a case study to test this new method and its suitability for mapping grain-size distribution across an entire beach.

The structure of this thesis is as follows: Chapter 2 provides the theoretical background of grain size analysis techniques, beach sediments and it introduces the new method of pyDGS. Chapter 2.5 outlines the problem and states the research questions. Chapter 3 details the methodology and the additional correction steps required for implementing pyDGS. The results of applying pyDGS to the different study sites are presented and discussed in Chapters 4 and 5 respectively. Finally, the main findings are summarised in Chapter 6.

Chapter 2

Background

The characterization of sediment samples involves the analysis of their grain size frequency distributions, which can be determined using different analysis methods. First, the advantages and disadvantages of each analysis method will be discussed. Following that, techniques for analysing the output data, including commonly used statistics to describe the sediment sample will be discussed. After that, the area to volume correction methods are discussed and finally, the objective and research questions of this study are stated.

2.1 Grain size analysis techniques

Figure 2.1 shows the commonly used methods these include: sieve analysis (A), settling tube analysis (B) and laser diffraction (C).
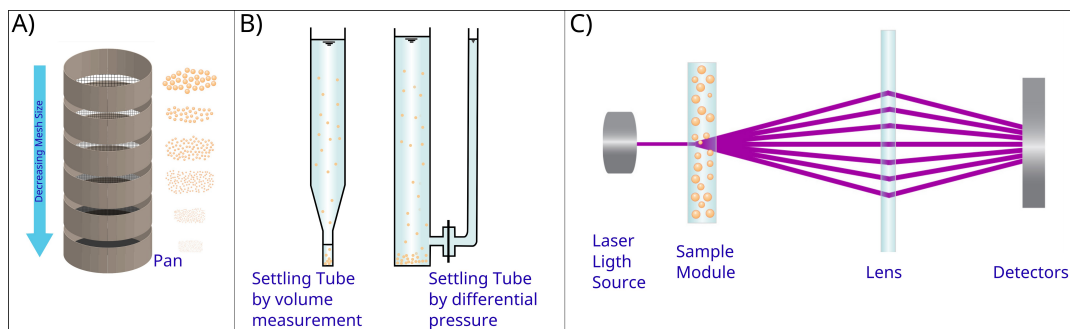
Figure 2.1: Schematic overview of the most commonly used grain size analysis techniques: Sieve (A) (ParticleTechLabs, 2023b); Settling Tube (B) (Modified from: Geldof and Slot, 1979); Laser Diffraction (C) (ParticleTechLabs, 2023a)

2.1.1 Sieve

Sieve analysis (Figure 2.1A) is an accurate method for the general analysis of sand and gravel samples. The process starts with collecting a sediment sample from the beach, which is taken back to the laboratory for analysis. Once in the lab, the sample is dried and weighed before being placed in a shaker, which separates it into different grain sizes. The method involves arranging sieves with varying mesh sizes on top of each other and then weighing the separated sediment on each sieve to determine the size distribution of the sample. This means that sieve analysis is a volume-by-weight measure.

However, sieve analysis has a few limitations. The main disadvantage of sieve analysis is its time-consuming nature, as it requires taking the sediment sample back to

the laboratory for analysis. Another limitation is that sieve analysis can be difficult to use for soft sediments (e.g. shells) as they may break down during the sieving process. Also, irregularly shaped grains or grains with different densities may not be accurately recorded. Despite these limitations, sieve analysis remains a common method for determining the size distribution of sandy samples. This method is considered the standard for grain size analysis.

2.1.2   Settling tube

The settling tube method (Figure 2.1B) is a technique for measuring the grain size of sand that is quick but less accurate. It involves settling the grains through a column of water, where the settling velocity is related to the grain size. The results are obtained by continuously weighing the grains that accumulate at the bottom or by measuring the differences in pressure. Or by recording the vertical accumulation rate of sediment at the bottom of the tube.

However, the settling tube method has several limitations. It can only be used for non-gravelly loose to weakly consolidated sediments and it is unsuitable for soft particles. Relative to sieving it provides better results for the grains with irregular shapes, densities, or porosities.
Therefore, while the settling tube method can be a useful tool for measuring the grain size of sand, it should be used with caution and in combination with other methods to obtain more accurate results (Komar and Cui, 1984).

2.1.3   Laser diffraction

Laser diffraction (Figure 2.1C) is a technique that relies on the principle that grains of a particular size diffract light through a specific angle, known as the angle of diffraction. This angle is inversely proportional to the grain size, i.e. the larger the angle, the smaller the grain size. The intensity of the diffracted beam is related to the number of grains with a specific grain size (Ferro & Mirabile, 2009).

A shortcoming of this method is that it is only effective for relatively small grain sizes, with an upper limit of around 1 mm (Cuttler et al., 2017). This means that it is not suitable for measuring larger particles. Before applying the laser diffraction method, the sample needs to be prepared by removing the coarser fraction through sieving. Which makes it also a time-consuming process.
For the smallest grain sizes, there is little difference between the sieving method and the laser diffraction method. This means that the results obtained from both methods can be used interchangeably (Papalazarou et al., 2010; Sperazza et al., 2004).

2.2   Optical granulometry

Optical granulometry is the measurement of grain size from images, using automated image analysis techniques. The first approach that used images of sediment for analysis was the photosieving method (Adams, 1979; Ibbeken and Schleyer, 1986), which required manually determining the longest axis of grain present in the image. Over the past two decades, significant advancements have led to the automation of this process.
The techniques employed in optical granulometry can generally be categorized into

two main categories: geometrical and statistical. Geometrical methods rely on image processing algorithms (e.g. edge detection, image segmentation) to measure the individual axes of grains (Sime and Ferguson, 2003; Chang and Chung, 2012). However, these methods struggle when the image lacks a distinct background intensity (due to overlapping grains), which makes it difficult to find an accurate threshold for segmenting the grains. Statistical methods analyze image texture using measures like autocorrelation (Rubin, 2004; Warrick et al., 2009) and semivariance (Carbonneau et al., 2004) to characterize grain size. The advantage is that this statistical approach does not measure individual grains, so it can be used on grains smaller than one pixel. The disadvantage of this method is that it requires calibration for specific sediments and/or sites.

This thesis is based on pyDGS, which is an open-source Python framework that has implemented the method of Buscombe (2013). This uses images of sediment as a continuous random field, (Buscombe et al., 2010), meaning that it captures information on all variable scales through the global power density function. By fitting translated and scaled wavelets the grain size distribution of an image of sediment can be determined. Through wavelet translation and scaling, both spatial and spectral details of the images are obtained, which combines the benefits of both the geometrical and statistical approach, without the need for additional calibrations.

However, these approaches have not achieved full integration, and ongoing improvements result in new methods. Alternative methods involve the application of machine learning or artificial intelligence. For example, InstaGrain as presented by Rubin and Chezar (2007), or the citizen-science initiative SandSnap (McFall et al., 2023). Both of these projects work with deep-learning models, with SandSnap relying on the SediNet model (Buscombe, 2020).

A different approach is the use of remote sensing techniques, such as UAV-based (Langhammer et al., 2017; Lang et al., 2021; Mair et al., 2022) or lidar techniques (Deronde et al., 2008; Matsumoto and Young, 2018; Díaz Gómez et al., 2022). The main advantage of using remote sensing lies in its ability to make use of high spatial resolution. However, the sensor resolution limits the capture of the finer sediment and it can not provide grain size distributions. Both methods focus on gravel and coarser bed material and use classifications, where beds that are finer than gravel are classified as sand or mud.

2.3 Representation of grain size composition

The techniques used for analyzing grain sizes, discussed above (Section 2.1), all result in a frequency distribution. These results can be represented in various ways (e.g. histogram, cumulative curve or frequency curve). When it comes to extracting statistical parameters from these distributions there are two techniques.

The first technique is the method of moments. This makes use of the entire frequency distribution, which is the main advantage of this method. However, a significant limitation is that natural sediments often display an 'open-ended' characteristic. For example, a sieved sediment sample has often a 'pan'-fraction of sediment that is finer than the finest sieve. This means that it is not clear how this fine sediment fraction is distributed and the method of moments can not be accurately applied.

The second technique involves the cumulative curve, which is the most commonly used to determine statistical parameters (Folk, 1974). This method involves plotting a cumulative curve and determining the grain size corresponding to various cumulative percentages. In this study, the cumulative curve is used to derive the statistical parameters. To illustrate this approach, a brief example is provided below.

Sieve number (ϕ)	Sieve opening (μm)	Mass of sieve (g)	Mass of sieve + sample retained (g)	Mass of sample retained (g)	% Mass retained	Cumulative % retained	% Finer
	Sample description:	Hypothetical sample					
	Mass of container + dry sample:	157.57	g				
	Mass of container:	34.75	g				
	Mass of dry sample:	122.82	g				
-3	8,000	498.31	498.32	0.01	0.0081	0.0081	99.99
-2	4,000	537.43	538.86	1.43	1.16	1.17	98.83
-1	2,000	337.87	344.41	6.54	5.32	6.49	93.51
0	1,000	302.51	331.41	28.90	23.53	30.02	69.98
0.5	710	264.18	284.80	20.62	16.79	46.81	53.19
1	500	247.06	272.68	25.63	20.87	67.68	32.32
1.25	425	253.73	262.89	9.16	7.45	75.13	24.87
1.5	355	233.32	244.15	10.83	8.82	83.95	16.05
1.75	300	220.73	228.96	8.23	6.70	90.65	9.35
2	250	222.62	230.25	7.63	6.21	96.86	3.14
2.5	180	203.32	206.67	3.35	2.73	99.59	0.41
3	125	190.55	191.05	0.50	0.41	100	0
4	63	178.71	178.71	0	0	100	0
Pan	0	292.71	292.71	0	0	100	0
	Total sample mass sieved:	122.81	g				
	Loss during sieve analysis:	0.0073	%				

Table 2.1: Example of data from a hypothetical sample derived from sieve analysis. And it shows the values for the percent finer.
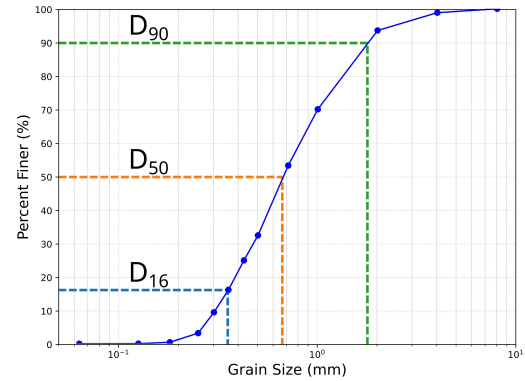
Figure 2.2: Cumulative curve of percent finer from data of the table of the left. Where the coloured lines show the percentile values for the D_{90}, D_{50} and D_{16}.

Table 2.1 shows data from a hypothetical sieved sample. Each sieve holds a specific mass of sediment, enabling the computation of the percentage mass retained in relation to the entire sample mass. The cumulative retained percentage represents the accumulation of both the sieve's retained mass percentage and the cumulative retained percentage of all preceding sieves. The percent finer is the inverse of this cumulative retained percentage.

A cumulative curve of the sample involves plotting the percent finer against the sieve opening (i.e. grain size). Figure 2.2 shows the cumulative curve of a hypothetical sample (Table 2.1). Through this curve the percentiles can be determined; e.g. the 90th percentile (D_{90}) involves locating the corresponding grain size by drawing a horizontal line at the y-value of 90 percent finer and locating the point where the curve intersects this line the resulting x-value represents the D_{90}.

Other often used percentiles are the D_{84}, the D_{50} and the D_{16}. The D_{90} and D_{84} are often used as roughness parameters in sediment transport models. Where the Nikuradse roughness length (k_s) is used to represent the effective height of sand roughness, that can be determined using the following equations: $k_s = 2.5D_{90}$ (Kamphuis, 1974); $k_s = 3.5D_{84}$ (Hey, 1979); $k_s = 3D_{90}$ (van Rijn, 1993).

There are various ways to describe the average grain size, with Trask (1930) proposing the use of median grain size as the descriptor. The median grain size represents the point where half of the sample is finer, and the other half is coarser. Although widely used, it does not provide an accurate description of the average grain size. Inman (1952) showed the difference between the median (D_{50}) and the mean ($M_\sigma = D_{84} + D_{16} / 2$), where the mean is a measure of the average grain size. Otto (1939) (M_z) and McCammon (1962) (M_5) proposed to use a linear combination of multiple percentiles instead ($M_z = \frac{D_{16}+D_{50}+D_{84}}{3}$; $M_5 = \frac{D_{10}+D_{30}+D_{50}+D_{70}+D_{90}}{5}$). Which takes

more of the distribution curve into consideration. The problem with this is that there are no clear rules regarding the number of percentiles to be considered.

σ-Interval	Sorting
< 0.35	Very Well Sorted
0.35 - 0.50	Well Sorted
0.50 - 1.00	Moderately Sorted
1.00 - 2.00	Poorly Sorted
2.00 - 4.00	Very Poorly Sorted
> 4.00	Extremely Poorly Sorted
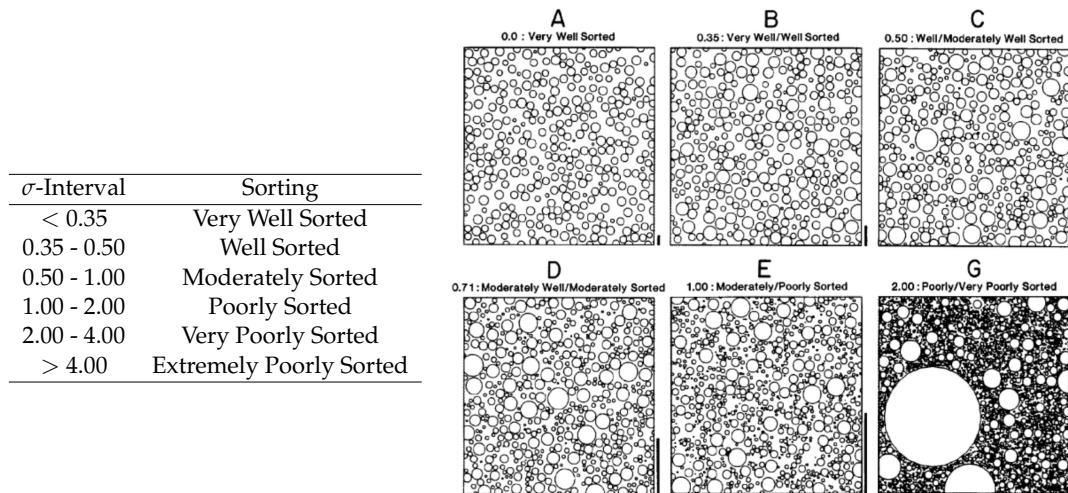
Table 2.2: Intervals that are used to describe different types of sediment, with a verbal scale. (*From: Folk and Ward, 1957*)

Figure 2.3: Visual examples of the different types of sediment, ranging from well sorted (A) to Poorly sorted (G). (*From: Longiaru, 1987*)

A different method to describe the grain size distribution is by examining the gradation of the sample, which can be measured as the fraction between two percentiles. The sorting of the sediment can be described by the range between the 84th and 16th percentiles. A large range indicates poorly sorted sediment, while a small range indicates well-sorted sediment. Folk and Ward (1957) defined the Inclusive Graphic Standard Deviation as:

$$\sigma_s = \frac{D_{84} - D_{16}}{4} + \frac{D_{95} - D_5}{6.6} \tag{2.1}$$

Table 2.2 shows the intervals used for categorizing various sediment types. Where, Figure 2.3 provides a visual representation of the corresponding gradations.

Example

To further explain the previously mentioned techniques for describing grain size distributions, the following examples are shown. In Figure 2.4, five distinct grain size distributions are visible, whereas Figure 2.4F shows the difference in shape between these curves. Specifically, Figure 2.4A and C exhibit an S-shaped curve, whereas Figure 2.4D and E display a broader Z-shape. Figure 2.4B shows an almost linear trend. Table 2.3 provides an overview of the statistical characteristics of the sediment samples presented in Figure 2.4. The table shows the difference between the median and mean grain sizes. Furthermore, it shows the variability in the mean grain size calculations. It is worth noting that the median grain size consistently appears to be finer than the calculated mean grain sizes.

Table 2.3 also includes the values for D_{16} and D_{90}. The D_{16} values remain relatively consistent with each other. The values for the D_{90} seem dependent on the shape of the curve, whereas the broad Z-shape results in larger values for the D_{90}.
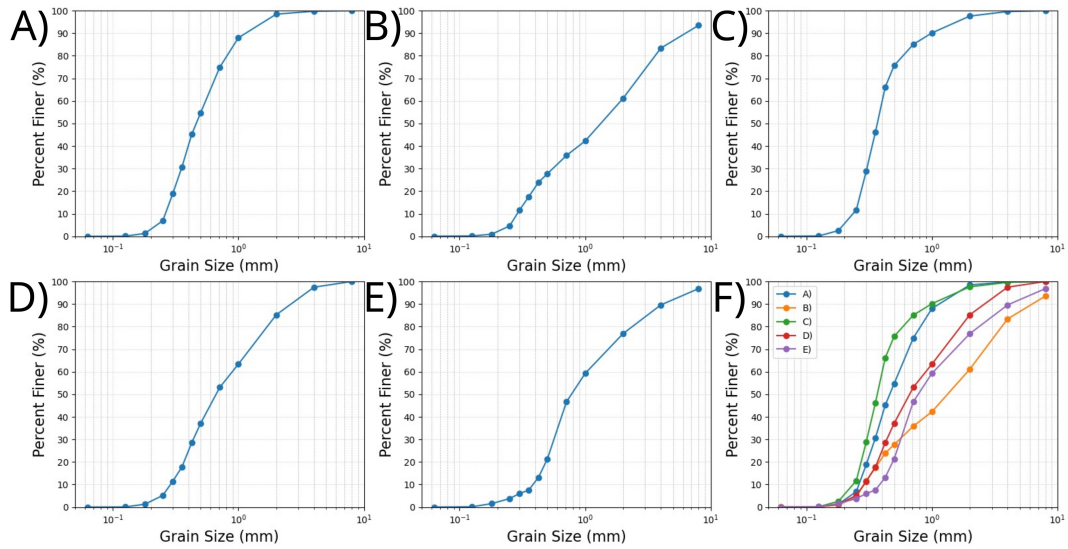
Figure 2.4: Grain size distribution curves showing percent finer on the Y-axis (A-E), and a composite plot combining all previous curves (F)

Locations	Gradation		Average Grain Size			Percentiles		
(Figure 2.4)	σ_s	Verbal Scale	M_σ (Inman, 1952)	M_z (Otto, 1939)	M_5 (McCammon, 1962)	D_{50}	D_{16}	D_{90}
A	0.375	Well Sorted	0.600	0.554	0.586	0.463	0.288	1.19
B	2.15	Very Poorly Sorted	2.30	2.00	2.34	1.41	0.341	6.62
C	0.326	Very Well Sorted	0.475	0.439	0.471	0.368	0.262	0.992
D	0.909	Moderately Sorted	1.14	0.985	1.10	0.670	0.340	2.78
E	1.69	Poorly Sorted	1.79	1.45	1.52	0.787	0.288	1.85

Table 2.3: Statistical parameters of the grain size distribution curves, where the locations correspond to the subplots of Figure 2.4.

2.4   Area to volume conversion coefficient

pyDGS measures the length of individual grains along a horizontal line taken at the image, which results in an 'area-by-size' measurement for grain size. While other methods of grain size analysis (e.g. sieving, settling tube or laser diffraction) are a 'volume-by-weight' measurement. Conversion techniques between different sampling methods already exist (Kellerhals and Bray, 1971; Proffitt, 1980; Diplas and Sutherland, 1988), but there seems confusion regarding how these conversion methods apply to photographic analysis. Thus, in this section, we will provide an overview of these conversion methods and their implications for grain size analysis using photographic techniques.

2.4.1   Cube model

A visual representation of the 'cube model' as proposed by Kellerhals and Bray (1971) is shown in Figure 2.5. It consists of three differently sized cubes (with sizes 1, 2, and 4), which each represent the different grain sizes in a sediment mixture. Each size of the cube is assumed to occupy 1/3 of the total volume of the cube, as visible in Figure 2.5B. Figure 2.5A displays one side of the cube's surface, as would be observed when analysing a surface sample. There are four possible ways to describe the surface sample. By size, by area, by volume and by number of particles. Table

2.4 shows that these different ways are not directly comparable.

Kellerhals and Bray (1971) formulated an equation that can convert an area-by-weight to a volume-by-weight distribution as:

$$p(V-W)_i = Cp(A-W)_iD_i^{-1} \rightarrow p(V-W)_i = p(A-A)_i \qquad (2.2)$$

, where $p(V-W)_i$ and $p(A-W)_i$ are the percentages from the volume-by-weight and area-by-weight respectively and D_i is the grain size. This method is intended to convert a surface sample that has been sieved to a volume sample that has also been sieved. However, this conversion is not made in most image-based analyses.
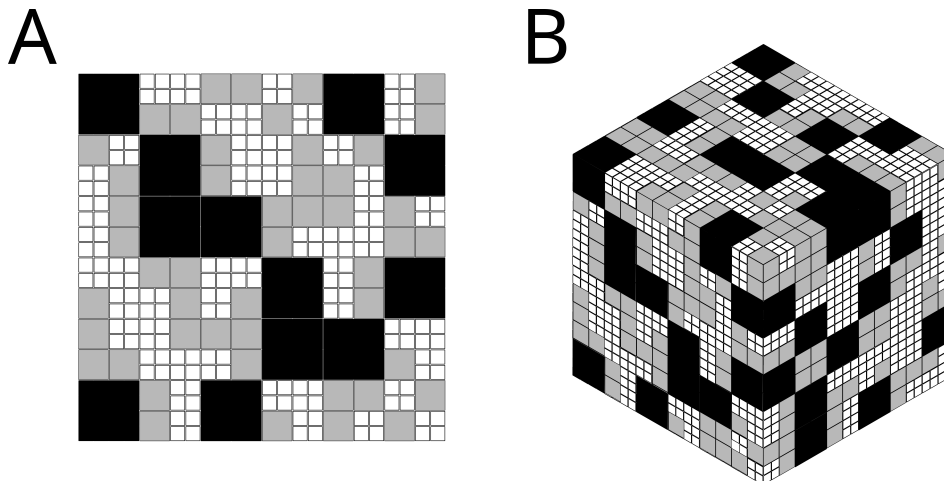
Figure 2.5: Cube model surface sample (A), and 3D representation (B) (*After: Kellerhals and Bray, 1971 and Proffitt, 1980*)

Size	Area	Volume	Nb of cubes in surface sample	% by Size $p(A-S)$	% by Area $p(A-A)$	% by Volume $p(A-W)$	% by Nb	% Cube Volume $p(V-W)$
1	1	1	192	57.14	33.33	14.29	76.19	33.33
2	4	8	48	28.57	33.33	28.57	19.05	33.33
4	16	64	12	14.29	33.33	57.14	4.76	33.33

Table 2.4: Analysis of surface sample versus volume sample. (*Data from: Proffitt, 1980*)

2.4.2 Proffitt's improved method

According to Proffitt (1980), the 'cube model' was unable to convert from area to volume for river sediments due to the absence of various shapes and voids found in natural sediment. Therefore, Proffitt (1980) used the 'cube model' as a starting point and rewrote eq. 2.2 as:

$$p(V-W)_i = Cp(A-W)_iD_i^{-x} \qquad (2.3)$$

To determine the value of the exponent x in eq. 2.3, Proffitt (1980) plotted the percentage volume-by-weight divided by the percentage area-by-weight against the sieve size, as shown in Figure 2.6. The exponent x was found by fitting a power function to the data, resulting in a mean value of -0.47. However, according to Proffitt (1980), the value of x is very sensitive to sampling errors in the finest 1% of the grain sizes, and better results are obtained by ignoring this fine fraction.
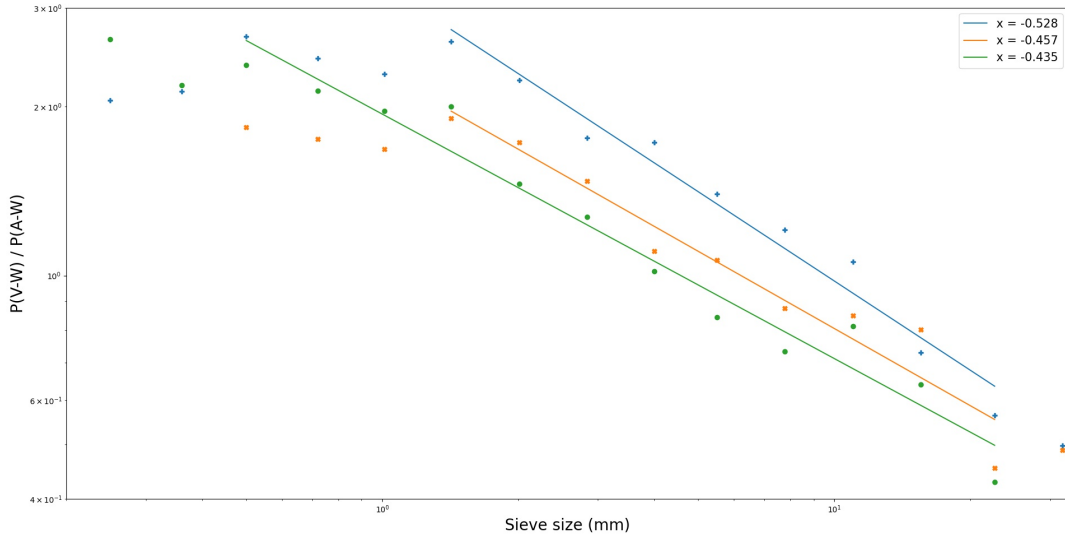
Figure 2.6: Area-by-weight to volume-by-weight conversion. Data is from three different experiments, where the mean exponent has a value of -0.47 (*Data from: Proffitt, 1980*)

2.4.3   Modified cube model

The main problem of the 'cube model' is that all particles are tightly packed together and do not have any voids. Consequently, an areal sample of this voidless cube only considers the particles on the outermost layer of the cube, neglecting the particles beneath it. In natural sediments, however, the presence of voids means that particles underneath the outer layer also need to be taken into account, favouring smaller particles. Diplas and Sutherland (1988) created the 'modified cube model' that incorporates these voids:

$$p(V - W) = Cp(A - W)_i f_1(D_i) f_2(D_i) \tag{2.4}$$

, where $f_1(D_i) = D_i^{-1}$ is the same as in eq. 2.2 and $f_2(D_i)$ include the particles from the lower layer that are visible through the voids. Using a statistical approach, Diplas and Sutherland (1988) determined that $f_2(D_i) = D_i^{-0.42}$, based on a modified cube model with a porosity of 33%. This exponent value of $x = -0.42$ appears to be consistent with the experimental results from Proffitt (1980).

2.4.4   General conversion relation

Graham et al. (2012) used the aforementioned relations to convert (areal) photographic sampling techniques to volumetric samples. The general form to convert different sampling techniques is defined as:

$$p(proc_1)_i = K \cdot p(proc_2)_i \cdot D_i^x \tag{2.5}$$

, where $p(proc_n)_i$ is the percentage of the different sampling techniques for size fraction i, D_i is the grain size, K is a constant and x is the conversion factor. This conversion factor can be determined in the same way as described by Proffitt (1980). By plotting $p(proc_1)_i / p(proc_2)_i$ against the grain size and fitting a power-function to the data, where the exponent of this power-function is equal to the conversion factor.

Recent studies, such as Cuttler et al. (2017) and Harvey et al. (2022), applied this general conversion method to their data. However, they limit the conversion factor *x* to values between 0 and -1, as suggested by Diplas and Sutherland (1988). However, this limit is based on an area-by-weight conversion, which is necessary when surface samples are taken using wax or tape. But since pyDGS is an area-by-size measure, the conversion factor can also be positive and smaller than -1.

2.5 Problem description

The literature review (Ch 2.4) indicates that an areal sediment sample and a volumetric sample are not the same, hence a conversion is necessary when using pyDGS to determine grain size distribution as an alternative to mechanical sieving. According to the conversion method presented in Graham et al. (2012), a second sampling method is required. It is currently believed that the conversion factor for a beach is a constant, but this assumption may only hold true for well-sorted beaches where the mean grain size remains uniform across the beach. On the other hand, mixed sandy beaches, where grain size varies greatly in space and in time, would require a different correction factor for each location. This would result in the need for physical sampling and sieving, which is what pyDGS aims to avoid.

Furthermore, the current grain size analysis method used at PHSD involves collecting sediment samples from the top 5 cm and assuming that the surface layer is uniform throughout this depth. However, for mixed sandy beaches, this assumption may not hold true. Thus, it would be more beneficial to sample only the surface layer, which is relevant to the bed roughness and sediment transport formulations, rather than including the layers below. Sampling the surface layer exclusively could provide more valuable information.

2.5.1 Objective

The primary objective of this study is to determine whether pyDGS is a viable alternative to mechanical sieving for grain size distribution analysis. To achieve this, the correct conversion method must be identified and subsequently applied to the Prins Hendrik Sanddike, which is a poorly sorted sandy beach. When a conversion method is set-up, the pyDGS technique will be implemented, and the results will be compared to those obtained from traditional sieving to assess its accuracy and reliability. In order to reach the main objective the following research questions are stated:

2.5.2 Research questions

1. **What is the most suitable conversion method for accurately determining grain size distribution on a poorly sorted sandy beach like the Prins Hendrik Sanddike?**

   a. What are the steps required to apply pyDGS to a poorly sorted sandy beach?

   b. How does the current General Conversion method preform on the PHSD? And what is the spatial distribution of the shape correction parameter?

c. How do the accuracy and reliability of the results obtained via pyDGS compare to those obtained from traditional mechanical sieving?

2. **What are the potential applications of pyDGS?**

   a. What are the advantages and limitations of using pyDGS as an alternative to traditional mechanical sieving for grain size analysis on the PHSD?

   b. What additional insights can pyDGS offer into the grain size distribution of poorly sorted sandy beaches that are not apparent through traditional mechanical sieving?

Chapter 3

Methods

As discussed in Chapter 2.1, mechanical sieving is the most widely adopted method for grain size analysis and it is considered the gold standard for comparison with other methods. This thesis aims to study the possibility of using pyDGS as a substitute for mechanical sieving. To make an accurate comparison between pyDGS and mechanical sieving, it is important to first convert the results to the same type of measure (Chapter 2.4).

This chapter first introduces the study area. Subsequently, the procedure used for mechanical sieving will be discussed, and all components of pyDGS are explained, including the way the camera is set up, the coin finder GUI, the configuration of pyDGS and the additional corrections steps that are necessary. Finally the sampling method and field campaigns will be introduced.

3.1 Study areas

The main objective of this thesis is to implement the proposed method on mixed sandy beaches. Consequently, the primary study area chosen for this purpose is the Prins Hendrik Sanddike (See Figure 3.1). The sediment of the beach can be characterised as a poorly sorted sandy beach. The research location close to Egmond aan Zee was selected to contrast the outcomes with those from a finer median grain size and well-sorted beach.

3.1.1 Prins Hendrik Sanddike

The Prins Hendrik polder is situated on the southeast coast of the Dutch island of Texel and was formerly protected by a hard-structured dike known as the Prins Hendrik dike. However, in 2007 the Dutch authority for water defence structures, Hoogheemraadschap Hollands Noorderkwartier (HHNK), determined that this dike did not meet safety standards. To address this, a solution called the Prins Hendrik Sanddike (PHSD) was implemented to protect the area in front of the existing dike. This solution

Figure 3.1: Overview map featuring both study sites in the Netherlands: the Prins Hendrik Sanddike situated on the southeast coast of the island Texel, and Egmond aan Zee located on the coast of the North Sea.

features a sand dike, which runs

parallel to the old dike and has a crest level of +8 meters NAP. And a low-level sand spit was built, with a crest level of +3 meters.

The sand for the project was dredged from two locations in the North Sea, Q2N and L15 see Figure 3.2. The majority of the sand came from Q2N, which forms the core of the dike and has a median grain size of medium- to fine sand (Table 3.1). Sand from L15, consisting of coarse sand to fine gravels and shells, was used as a top layer to reduce aeolian transport (Strypsteen et al., 2021).

Situated on the Wadden Sea side of the island, the PHSD is sheltered from the North Sea. Resulting in relatively small significant wave heights (< 1 m), which makes the tidal currents significant. It is commonly referred to as a "back-barrier beach." This study focuses on the intertidal beach, which has a beach width that varies between 15 to 30 meters.

Location	Size Class	Terminology (Jan de Nul)
Q2N	$D_{50} <220 \ \mu$m	Finest
	$D_{50} >220 \ \mu$m	Fine
	$D_{50} >280 \ \mu$m	Medium
L15	$D_{50} >280 \ \mu$m	Medium
	$D_{50} >400 \ \mu$m	Coarse
	$D_{85} >2000 \ \mu$m	Very Coarse
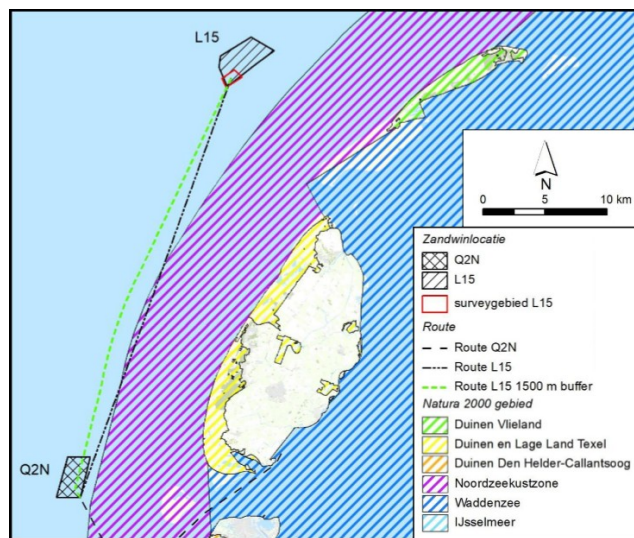
Figure 3.2:   Locations extraction sites (*From: Bruinzeel and Snoek, 2018*)

Table 3.1:   Grain size classes from the extraction sites according to the contractor (Jan de Nul) (*From: Klein Obbink, 2022*)

3.1.2   Egmond aan Zee

The study area near Egmond aan Zee faces the North Sea and is a micro-tidal, storm-wave dominated site (Ruessink et al., 2019). The shoreline is exposed to waves coming from directions ranging from southwest to north. The intertidal beach has a gentle slope of around 1:30. The width of the beach usually varies between 100 m during low water and 50 m during high water, but this can differ widely in space and time.

The sediment near Egmond aan Zee consists of reworked early- and middle-Pleistocene sands from various sources (Eisma, 1968). The continuous reworking of the sediment during the Holocene has resulted in the observed well-sorted sand.

3.2 Mechanical sieving

Sediment samples were collected from the top 50-60 mm of the surface layer, which was then taken to the lab for analysis. Approximately 120 g of each sample was oven-dried at 105 °C overnight ($\approx$ 8 hours) and sieved for 15 minutes using a mechanical sieve shaker. This recommended sieving time ensures full separation of the sediment into size classes without damaging the shells present in the sample (Román-Sierra et al., 2013; Syvitski, 1991;USACE, 2008).

The sieves were weighed with the sediment on top, and the mass of the empty sieve was used to calculate the amount of sediment collected. Doing this for each sieve provided the grain size distribution of a sediment sample. The sieves used range from 4.0 ϕ to -3.0 ϕ and are listed in Table 3.2. The sieve tower used in this study is based on an earlier study performed at the PHSD (Klein Obbink, 2022) and was selected to minimise the number of sieves while maintaining accuracy. The relationship between the used sieve tower and the extensive sieve tower is shown in Figure 3.3, which validates the use of the less extensive sieve tower.

	Very Fine		Fine			Medium			Coarse		Very Coarse	Very Fine	Fine
phi	4	3	2.5	2	1.75	1.5	1.25	1	0.5	0	-1	-2	-3
mm	0.0625	0.125	0.180	0.250	0.300	0.355	0.425	0.5	0.710	1	2	4	8

Table 3.2: The sieve tower employed in this study contained sieves with mesh sizes measured in phi and mm, with the grain classification located at the top.
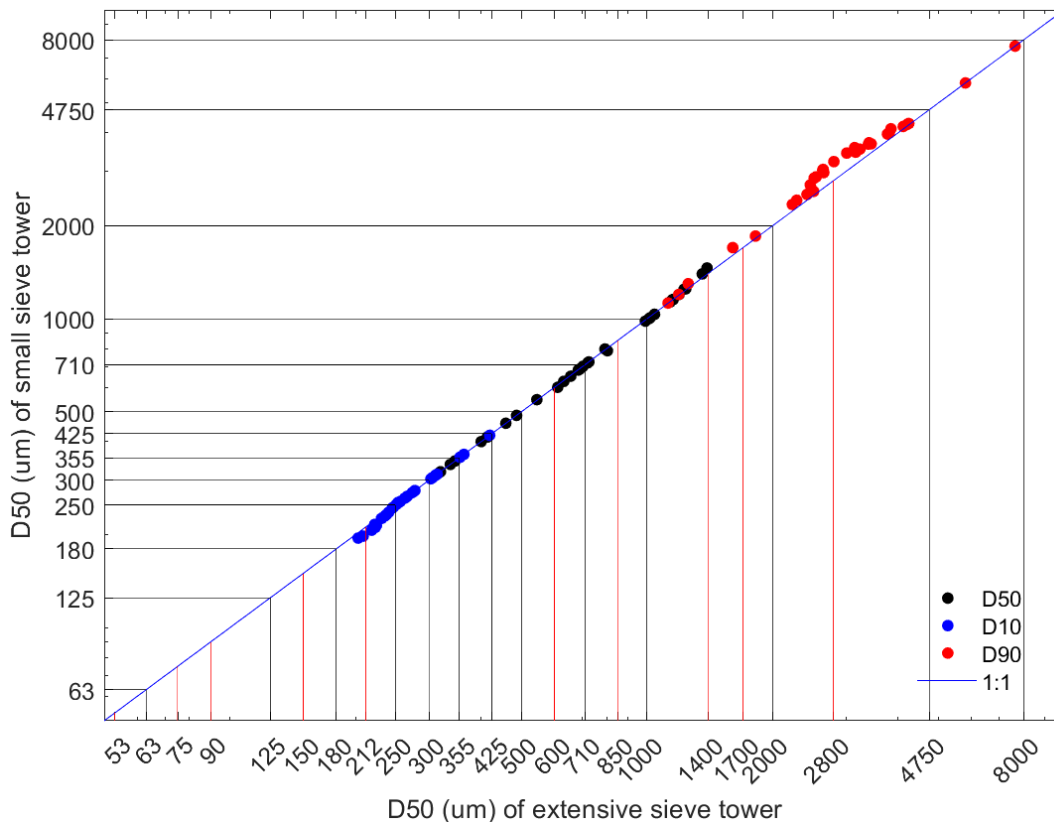
Figure 3.3: Validation of the used sieve tower, where the used sieved tower is compared to a more extensive tower. (*From: Klein Obbink, 2022*)

3.3   Grain size estimation using pyDGS

The initial step in using pyDGS involves capturing images of the sediment. In this study, the usefulness of a smartphone camera is tested. The advantage of using a smartphone is its ease of use and portability, making it convenient for fieldwork. However, holding the camera by hand may result in varying distances between the camera and the sediment bed. This section provides guidelines for taking the pictures and performing necessary correction steps. The process begins with an explanation of the camera setup, followed by an explanation of the graphical user interface (GUI) used to locate the coin in the image. Subsequently, the adjustments made to the pyDGS code are elaborated upon, and finally, the method change in grain size determination is discussed.

3.3.1   Camera settings

The sediment sample images were captured using a Samsung Galaxy A51 smartphone with a 48 MP wide camera, using the default camera application. It is recommended to use the manual focus option instead of the auto-focus, because the latter tends to focus on the coin rather than the grains. Using manual focus gives the added benefit of an edge detection filter (in the default app). Figure 3.4 shows how a light-green filter is added to the screen. The light-green edge detection filter changes dynamically as the focus changes, which helps the user to ensure that the finest grains are in focus and visible. The downside of using the manual focus is that it needs to be applied individually for each image captured.

An alternative approach is to tap on the screen, where the camera should focus. The problem with this method is that it is uncertain whether the camera will successfully focus on the smallest grains. For the most reliable results it is recommended to use the manual focus method.

Additionally, all modern smartphones have a MEMS-gyroscope which tells the orientation of the phone. This can be used to keep the phone flat and when the bed is also flat the photo is always perpendicular to the bed. This keeps the spatial resolution of the image uniform over the image.

Figure 3.4: Screenshot from default camera application of the Samsung Galaxy A51, using manual focus. The light-green colour visible in the image is an added edge-detection filter.

Buscombe (2013) outlined a few guidelines for the collection of suitable images. The most important criterion is that the smallest grains are well-resolved, i.e. the smallest grains should at least be multiple pixels wide. This is a function of the camera height above the bed and will be discussed in Section 3.3.2.

Another important aspect is that there should be no relief features present in the image. For example, bedforms such as ripples or raindrops can create small impact craters. Depending on the study site, removing the top layer of the sediment surface may be necessary to improve the results from pyDGS.

3.3.2 Image Scaling GUI

The grain size output of pyDGS is measured in terms of a number of pixels, which does not provide useful results as the size of a pixel can vary based on the distance between the camera lens and the captured object. In this study, a coin is used as a reference scale for converting the pixel size to millimetres. This gives the ability to collect the images using only a camera and a coin, eliminating the need for a tripod or other gear, which is advantageous for fieldwork. The diameter of the coin is a known constant, and the resolution of an image is determined by measuring the number of pixels of the coin's diameter.

A Graphical User Interface (GUI) was developed in Python to automate the process of finding the coin in each image, computing, and storing the image resolution. This makes it possible to collect hundreds of images during a field day and simplify the workflow. Figure 3.5 shows a snapshot of the program, displaying a loaded image.
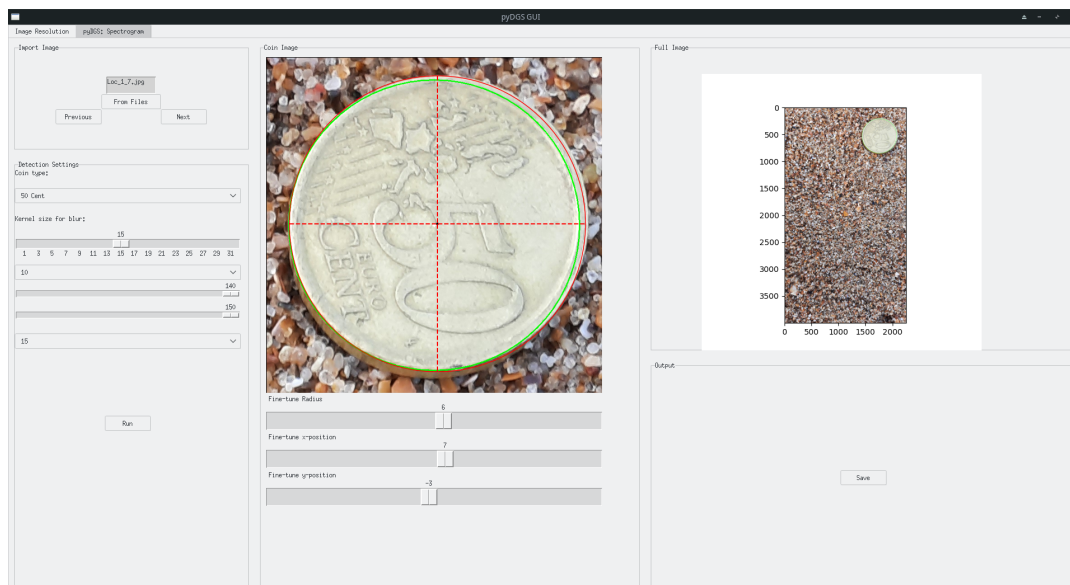
Figure 3.5: Screenshot from image resolution GUI.

The GUI uses OpenCV, which is an open-source Python library focused on computer vision algorithms. The main function that is used to find the coin in the image is *cv.HoughCircles()*. This function implements the Circle Hough Transform, a variation of the Hough (line) Transform.

A circle with radius r and centre (a, b) can be described as:

$$\begin{cases} x = a + r\cos(\theta) \\ y = b + r\cos(\theta) \end{cases} \tag{3.1}$$

Where θ is the angle that goes from 0 to 360 degrees and the points (x, y) will follow the perimeter of the circle. Equation 3.1 has three unknowns (a, b, r). By fixing the radius (r) the problem becomes 2-dimensional, and the centre of the circle (a, b) must be found.

The input for the Hough Transform is an edge-detected image, where a fixed-radius circle is computed at each edge. An accumulator array is used to identify the point with the most circle intersections, which represents the centre (a, b) of the circle.

By iterating over different radii we keep on finding circles. It can be assumed that there is only one circle with the radius of the coin in the image. So the code is written to continue until it only finds 1 circle, this circle is the coin. Appendix A shows a flowchart of this process.

Now the coin's diameter in pixels is known, along with the known diameter in millimetres. This gives the ability to compute the size of a pixel in millimetres.

An issue with this coin detection method is that it stops once it finds only one circle, which may not be the optimal radius for the coin. Fine-tuning the input parameters can improve the results, but this is a time-consuming and difficult task. Therefore the ability to adjust the radius of the circle is implemented. Once the coin is found, the GUI displays a zoomed-in image of the coin's location and allows the radius and centre coordinates (a, b) to be adjusted using the sliders.

Image name	Pixel size (mm/pixel)	Date/time	Device	Latitude	Longitude	Image height (mm)	Image width (mm)	Heigth above bed (mm)
Loc_1_15.jpg	0.030086849	2022:07:08 08:58:32	SM-A515F	53°1′35.7708″ N	4°49′38.54028″ E	67.69540943	120.3473945	91.58790687
Loc_1_14.jpg	0.041810345	2022:07:08 08:58:25	SM-A515F	53°1′35.7708″ N	4°49′38.54028″ E	94.07327586	167.2413793	127.2756085
Loc_1_13.jpg	0.041382253	2022:07:08 08:58:22	SM-A515F	53°1′35.7708″ N	4°49′38.54028″ E	93.11006826	165.5290102	125.9724453
Loc_1_12.jpg	0.072172619	2022:07:08 08:58:15	SM-A515F	53°1′35.7708″ N	4°49′38.54028″ E	162.3883929	288.6904762	219.7019433
Loc_1_11.jpg	0.192460317	2022:07:08 08:56:42	SM-A515F	53°1′36.2158″ N	4°49′38.75196″ E	433.0357143	769.8412698	585.8718487

Table 3.3: Example of all image data that is stored using the Image Resolution GUI.

The resolution of the coin and other image meta-data is stored in a *.csv*-file, as shown in Table 3.3. This table displays the following information: the name of the image-file, the pixel size (mm/pixel), the date and time the image was taken, which device was used to take the image, the latitude and longitude where the image was taken, the height and the width of the image in millimetres and an estimation at which height above the bed the image was taken.

The latitude and longitude can be used as a rough estimate of the coordinates. The height above the bed can be used as an indication of the quality of the image. This distance can be calculated using the following equation:

$$H_{\text{Bed}} = \frac{f \times W_{\text{image}} \times \text{Image Resolution}}{W_{\text{Sensor}}} \tag{3.2}$$

where Figure 3.6 shows the used parameters. The width of the sensor is a property dependent on the used device, this is the reason why the type of device is saved in the *.csv*-file.
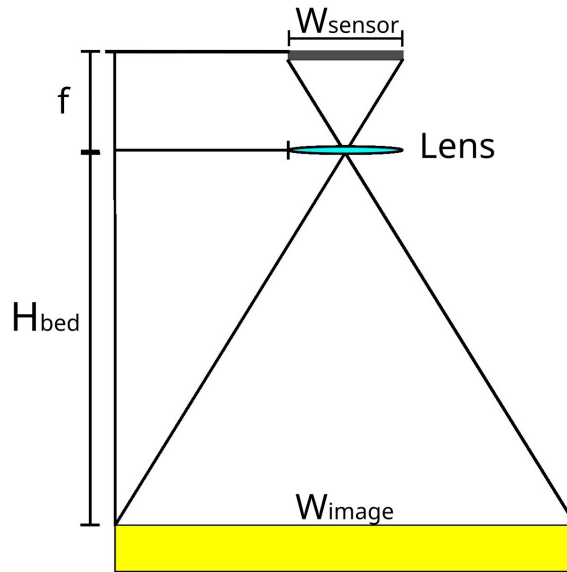
Figure 3.6: Diagram illustrating the variables described in Eq. 3.2, where f is the focal length of the camera sensor, H_{bed} is the height above the bed, W_{sensor} and W_{image} are the width of the sensor and image, respectively

The GUI gives the user the ability to choose, from menus and sliders, the input parameters for the *cv.HoughCircles()* function. The first step is to import an image, supported image file formats include .jpg, .png and .tiff. For different file formats check the OpenCV documentation of the *imread* function. The next step is to select the type of coin that is used in the image. For now, all euro coins are available and a quarter dollar (US). Next, the kernel size for the median blur can be selected using a slider. The larger the kernel size the blurrier the image gets which affects the (canny) edge detection. It is recommended to use an initial value of approximately 15 for this. Next, the size between the minimum and maximum radius can be selected, which determines the window at what radii the circles can be. Recommended value: 10. Next, the start value for the minimum/maximum radius can be selected, which improves the efficiency of the program in finding the coin. If the radius of the coin (in pixels) is approximately known the program can start around that value. The last option is the steps between each window, which are the steps (pixels) between each iteration of the moving window. (default is 10).

When the coin is found the middle window (see Figure 3.5) shows a zoomed-in selection of the location of the coin. The sliders below the image allow the users to change the radius and the x and y position of the centre of the circle. Only the radius is saved, so the x/y position doesn't have to be perfectly aligned.
The window in the bottom right has the save button, which saves all data. This is folder-specific. If all the photos are placed within a single folder, the data will be stored in a single .csv-file. In simpler terms, all the data collected during a field campaign can be neatly stored in one .csv-file.

3.3.3   pyDGS setup

The code of Buscombe (2013) is freely available online (pyDGS link). To make py-
DGS easier to work with and more efficient to use some improvements have been
made to the original code. When the script is run, the user is prompted with a few
questions, as shown in Figure 3.7. First, the name of the directory where the images
are stored is asked. With the name of the directory, the script can automatically find
the image resolution of each image, if the image resolution GUI is used from Section
3.3.2. The script will then iterate through all images in the given directory. The sec-
ond question is if the uncorrected percentages need to be stored, this is useful if you
want to use a different correction method. Third, you can apply the general conver-
sion relation, which will be explained in detail in Section 3.4.1. Finally, a description
can be added to the stored data for clarity.


```
(pydgs) [casper@CasperDesktop pyDGS-GUI]$ python Correct_Data.py
Type the name of the directory: R1/
The working directory will be:  /home/casper/Documents/Aardwetenschappen/MSc Thesis/Photo/26_10_22/Mobile/R1/
_____
Do you want to save the uncorrected percentages? (y/n) yes
Uncorrected percentages will be stored
_____
Do you want to apply and store the general conversion correction? (y/n) yes
Correction will be preformed
What is the power of the small scales (< 0.5 mm)? 3.28
What is the power of the large scales (=> 0.5 mm)? -2.23
_____
Give a description for stored data: temporary
The data will be stored as: /..._R1_temporary.csv
_____
```

Figure 3.7:   Screenshot of terminal where questions are prompted and py-
DGS will run after that.   The code shown in Appendix B and available at
github.com/CasperFBakker/pyDGS-GUI

The image resolution and the initial setup have been solved, now the minimum
and maximum scale should be chosen. Figure 3.8A shows how pyDGS analyses an
image. It takes one hundred horizontal lines and along these lines, it takes the length
of each grain.

The shell fragment in the zoom-in of Figure 3.8A is analysed 16 times, this shell
fragment is approximately 5 mm. This means that pyDGS estimates that in this
zoomed-in section 16 grains with a diameter of 5 mm, while (in this orientation)
there is only 1 grain with a diameter of 5 mm. This leads to an overestimation of the
larger grains.

This problem is solved by running pyDGS twice. The first run is for the small scales,
from 3 pixels ($\approx$ 0.06 mm) to 1 mm, where it is the same as shown in Figure 3.8A.
The second run is for the larger scales, from 1 mm to 8 mm, where the amount of
horizontal lines is reduced to 10. Figure 3.8B shows that the overestimation for the
large grains is reduced.
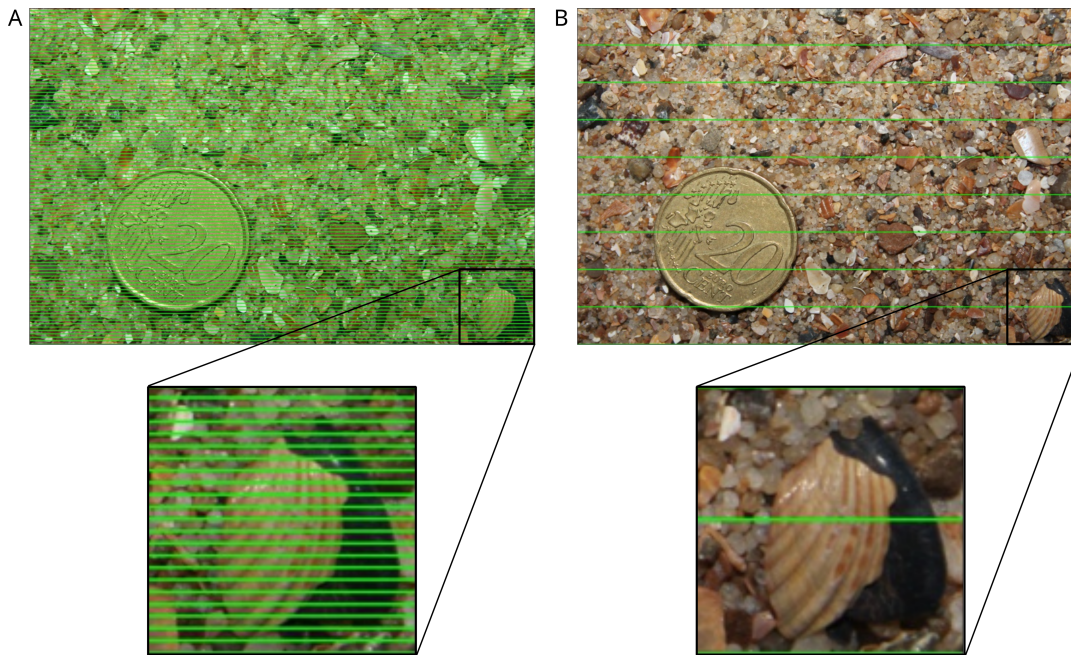
Figure 3.8: Visual representation how pyDGS processes an image. The original method is to perform the analysis along 100 horizontal lines (A), in the new approach for grains larger than 1 mm an additional analysis is done along only 10 lines (B).

Besides variations in grain size, grain shape introduces further uncertainty into the analysis. pyDGS measures the length of each grain and assumes that this is the grain diameter. Using this grain diameter for the grain size distribution assumes that the grain diameter is the same in each orientation. This means that each grain is a perfect sphere. However, as can be seen in Figure 3.8, the grains are not all perfectly round spheres.

Figure 3.9 shows a schematic example of two different grain shapes. For a perfectly round grain (Figure 3.9A) the grain diameter is the same at each orientation, while for an elongated grain (Figure 3.9B) the diameter changes with the orientation. To account for various grain shapes, each image is run multiple times at a different rotation, such that each rotated image gets its own grain size distribution. In the end, the average of each grain size bin for all rotations of the image is taken. Additionally, examining the standard deviation in each grain size bin gives information about the grain shape. If each grain is a perfect sphere, the percentage of each grain size fraction should be constant for each rotated image. If there is a lot of variation within these fractions the grain shape is asymmetrical. This will be used and discussed in Section 3.4.2.
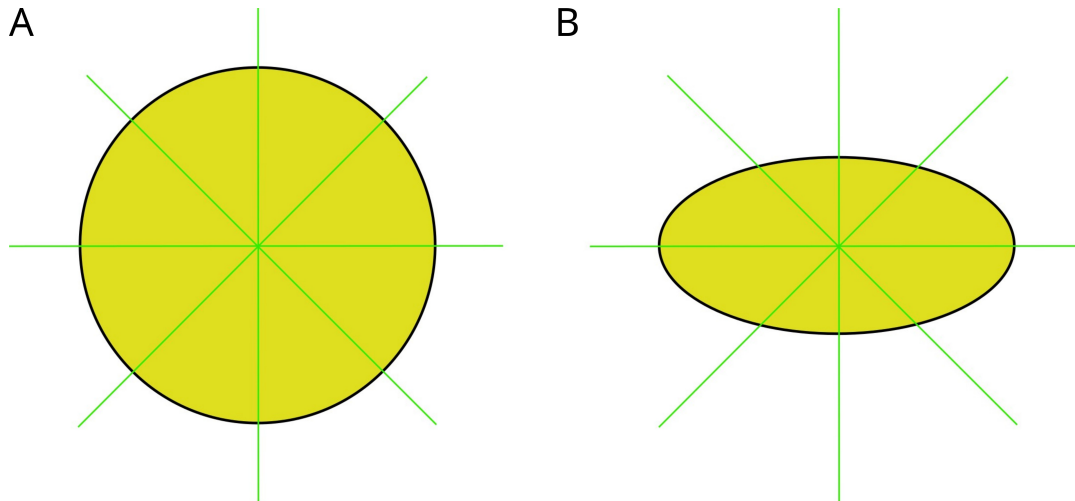
Figure 3.9: A) a schematic example of a grain that is a perfect circle and therefore has uniform diameter in all directions; B), an example of an ellipse-shaped grain that has a varying diameter dependent on its orientation.

3.3.4   Grain size from PDF

During sieve analysis, it is established that the grain size on top of a sieve is larger than the grain size below it and smaller than the grain size on the upper sieve (Ch 2.1.1). However, pyDGS employs a different approach to obtain the grain size, where the whole distribution is determined by using a probability density function (PDF).

The original code of pyDGS calculates grain size bins as interpolated values for the exact sieve size diameters. This difference might be negligible for the smallest sieves since the difference between both sieves is relatively small. However, for the larger sieve sizes, the difference is more pronounced (e.g. the difference between 0.125-180mm or 4-8mm). To ensure accurate comparisons between both two methods, the calculation must be consistent. Therefore, to obtain the grain the area under the curve of the probability density function must be calculated, where the area has the boundaries of the sieve diameters.
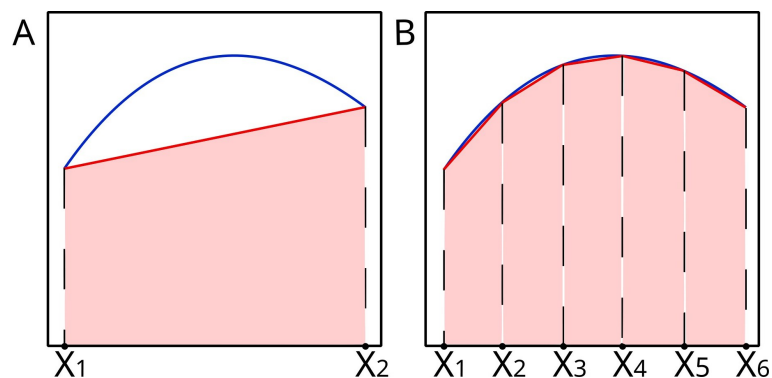
Figure 3.10: Example of the trapezoidal rule, which demonstrates that a single curve can be approximated either with only two points (A) or with six points (B).
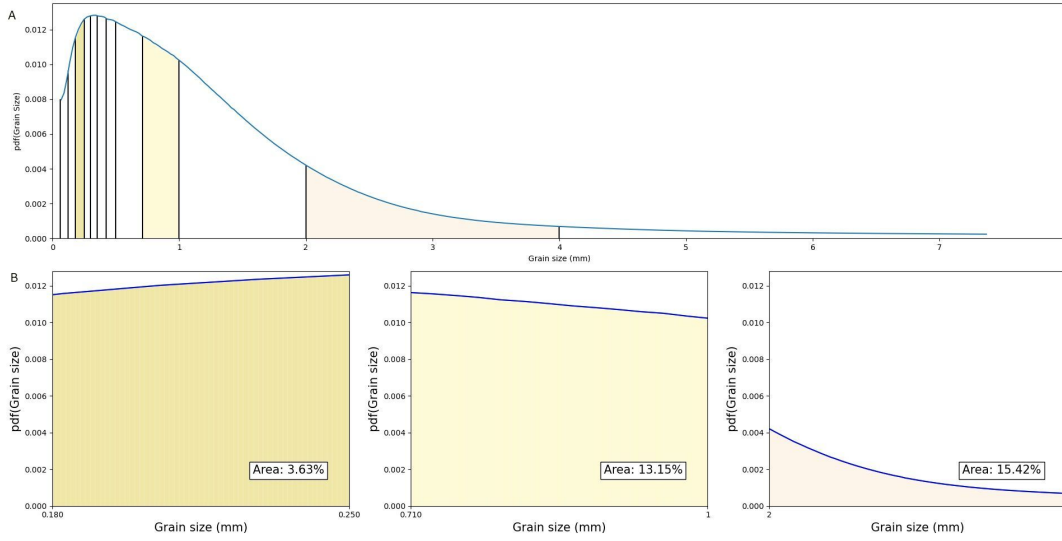
Figure 3.11: The probability density function of grain sizes in an image of sediment is shown as a function of grain size, with vertical black lines representing the sieves (A). The coloured areas provide an example of the area underneath the curve (B

Figure 3.11A shows an example of the probability density function for the grain size. The vertical black lines show where the sieves are present, i.e. the area between the vertical lines is the proportion of the grain class.

The trapezoidal rule is used to calculate the area under a graph. As shown in Figure 3.10A, the area is approximated by a single trapezoid. So, the area can be described as:

$$\int_{x_1}^{x_2} = f(x)dx \cong \frac{(x_2 - x_1)}{2}[f(x_1) + f(x_2)] \tag{3.3}$$

When the function that is described by the trapezoid is not linear the approximation is not very exact. To improve the approximation the interval between the points can be decreased. Figure 3.10B shows this, where six points are used instead of two. When the function is not linear, the trapezoidal approximation is not very exact. To improve the estimation, the distance between the points can be reduced. Figure 3.10B shows this, where six points are used instead of only two.

$$\int_{x_1}^{x_n} = f(x)dx \cong \frac{(x_n - x_1)}{n} \left[\frac{f(x_1) + f(x_2)}{2} + ... + \frac{f(x_n - 1) + f(x_n)}{2} \right] \tag{3.4}$$

In this study, grain classes are identical to those used in sieve analysis. The first step is to use linear interpolation to create an array of 1,000 points for each grain class. These points will be used to calculate the area under the curve, using Eq. 3.4.

Figure 3.11B shows a few examples of the percentages that are calculated for specific grain classes. Since the function is a probability density function the whole area underneath the curve is equal to 100%. However, due to the approximation made using the trapezoidal rule, there may be small errors that result in over- or underestimations of the area. To account for these inaccuracies and ensure that the total sum is equal to 100%, the results are normalised using the following equation.

$$p(S)_{i,cor} = \frac{p(S)_i}{\sum p(S)_i} \tag{3.5}$$

, $p(S)_{i,cor}$ is the corrected percentage and $p(S)_i$ is the percentage as determined from the probability density function, where i is the grain classes.

3.4 Area-to-volume correction method

To compare the results of pyDGS with the mechanical sieve method, an area-to-volume correction is required, as outlined in Chapter 2.4. This section first presents the general conversion relation as described by Graham et al. (2012), and then provides a new method for the area-to-volume conversion.

3.4.1 General conversion relation

In Chapter 2.4.4 the General Conversion Method is already briefly explained. Equation 2.5 can be rewritten as:

$$K \cdot D_i^x = \frac{p(proc_1)_i}{p(proc_2)_i} \tag{3.6}$$

where the grain size to the power of a conversion factor (x) is equal to the fraction of both different types of sampling techniques. To find the value for the conversion factor (x) Equation 3.6 can be written as:

$$\log(K) \cdot x = \log_{D_i}\left(\frac{p(proc_1)_i}{p(proc_2)_i}\right) \tag{3.7}$$

which means that by plotting the different sampling techniques on log-log axes as a function of the grain size, there should be a straight line that corresponds with the conversion factor. To find the value for this factor a power-function is fit to the data, where the exponent of this power-function is the conversion factor.

The first step is to apply this method to individual mixed sediment images from the PHSD to assess pyDGS's ability to predict grain size distribution on mixed sandy beaches. Since Buscombe (2013) does not recommend using pyDGS at mixed beaches, without providing a clear reason. Within these individual images, there could be a distribution of the conversion factor which could make it possible to apply this method to a large-scale sampling campaign. The second step is to find if there is a conversion coefficient that can accurately describe all points, or just multiple points, on the beach.

3.4.2 Novel correction method

The correction procedure outlined in the previous section is based on sampling techniques that rely on the use of adhesives to gather surface samples. However, a photo collects different properties, and additional information can now be obtained from these images. The proposed new method utilises the new information obtained from photo-optometry to convert the image into a volume-by-weight measure. The process involves the following steps:

1. Calculate the total number of grains in the photo frame,

2. Estimating the volume of each grain class.

3. Compute the mass of each grain size category by multiplying the number of grains by their volume and density.

The details of these steps will be discussed in this section.

Step 1:
The image taken of sediment covers a certain area, since the resolution is known the total area (A_T) of the image can be computed in square (milli)metres.

$$A_T = (W * R) * (L * R) \tag{3.8}$$

where W and L are the width and the length of the image in pixels and R is the resolution of the image in millimetres/pixel.
The output from pyDGS provides the grain size distribution in terms of percentages for each grain size class ($p(G)_i$). This means that each grain class occupies a specific area within the image frame. The area occupied by each grain class (A_i) can be determined from the percentage of the grain class and the total area of the image frame.

$$A_i = A_T * p(G)_i / 100 \tag{3.9}$$

If we take an idealised case, where all grains are perfectly spherical, the top view of each grain (2D) would resemble a circle, where the radius is equal in all directions. Assuming that the radius of each grain class can be approximated by a single value, the top-view area of each grain size class ($A_{grain,i}$) can be computed.

$$A_{grain,i} = \pi \cdot \left(\frac{D_i}{2}\right)^2 \qquad \text{(Circle)} \tag{3.10}$$

, where D_i corresponds to the grain class diameter (i.e. grain size).
In reality, grains deviate from the idealised spherical shape. In the previous section (Section 3.3.3), the method of rotating each image to obtain the standard deviation for each grain class was discussed. This standard deviation can then be used to quantify the asymmetry of each grain. A boundary value can be used at what value for the standard deviation a grain can be considered to be a circle and above the value the grain is represented as an ellipse. An ellipse has two different axes, where the first axis will represent the grain size radius, while the second axis will be defined as a parameter (γ) proportional to the standard deviation. A larger deviation means a more asymmetrical grain, and the second axis will be larger.

$$A_{grain,i} = \pi \cdot \frac{D_i}{2} \cdot \frac{\gamma * D_i}{2} \qquad \text{(Ellipse)} \tag{3.11}$$

Now the surface area of each individual grain for all grain classes is known. By dividing the surface area of the individual grain by the surface area of its grain class, the number of grains for each grain class can be estimated.

Step 2:
The next step is to estimate the volume of each grain. This method is similar to the one described above for calculating the surface area. The form of the grain is described by the three dimensions, the length (L), the width (I) and the thickness (S). Sneed and Folk, 1958 introduced a form diagram (Figure 3.12) that categorises the grain form into three groups: compact, platy, and elongated. All grains have a form that is a combination of these three primary structures. So in this study, we will consider three different types of grain shapes: perfect spheres, asymmetric ellipsoids, and flat disks. The spheres and ellipsoids are related to the standard deviation, while the flat disks are based on the grain size. It is assumed that grains larger than 2mm

are (fragmented) shells, as they have a large surface area when viewed from the top, but relatively low mass for their size.
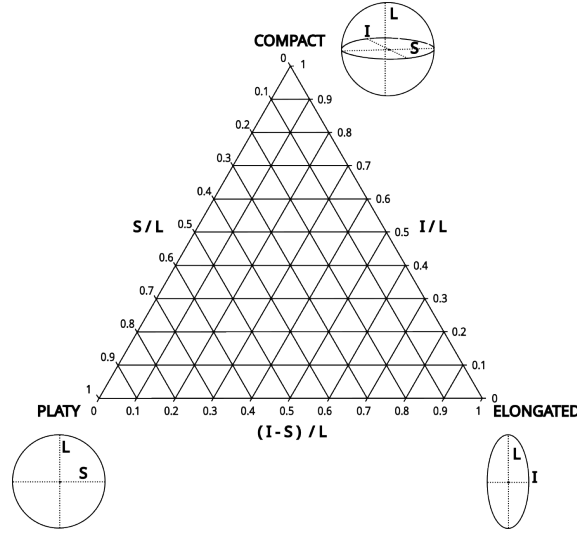
Figure 3.12: Form diagram, where the form is the function of its length (L), width (I) and thickness (S). (*Based of: Sneed and Folk, 1958*)

The volume of each grain class ($V_{grain,i}$) can be computed as:

$$V_{grain,i} = \begin{cases} \frac{\pi}{6} \cdot D_i^3 & \text{(Sphere)} \\ \frac{\pi}{6} * \cdot D_i^2 \cdot \gamma D_i & \text{(Ellipsoid)} \\ \pi * \frac{D_i}{2} \cdot S & \text{(Flat disk)} \end{cases} \qquad (3.12)$$

With the knowledge of the grain volumes, the mass of each grain (m_i) can be computed. The mass of a grain is the function of the density and the volume.

$$m_i = \rho \cdot V_{grain,i} \qquad (3.13)$$

For simplicity in this study, the density of all grains and shells present in the sample is assumed to be constant and set to the density of quartz. However, nothing is stopping future studies from making the density-dependent on the grain class.

Step 3:
The last step is to compute the total mass of each grain class (m_{D_i}), this is done by multiplying the number of grains ($Nb_{grains,i}$) by the mass of each individual grain (m_i).

$$m_{D_i} = Nb_{grains,i} \cdot m_i \qquad (3.14)$$

Now the grain size distribution as volume-by-weight can be computed from the results of pyDGS.

$$p(D)_i = \frac{m_{D_i}}{\sum m_{D_i}} \cdot 100\% \qquad (3.15)$$

Which gives the percentage of each grain class ($p(D)_i$) from the total volume.

Calibration method

An image of sediment can only capture the top-view (2D), so the information necessary to calculate the volume (3D) of the grains is missing. To overcome this limitation, a large-scale sampling campaign has been conducted (Section 3.5), where samples have been both photographed and sieved. Through sieve analysis, the mass of each grain class was obtained, which serves as a basis for calibrating grain volume. This calibration method involved plotting the difference between the sieve results and the pyDGS results.

For example, if the percentage of the grain size class of 1 mm is larger for the sieve, it means that the volume of the 1 mm grains is larger than the computed volume for the corrected pyDGS results. To improve this, the secondary axis of the ellipsoid grains or the thickness of the flat disk grains is adjusted. This calibration process is carried out for each grain class. However, the problem with this approach is that changing the volume of one class impacts the total volume, affecting all other classes. This makes it difficult to find the perfect solution and it remains a heuristic approach.

Due to the heuristic nature of this approach, this study primarily focuses on the coarsest grains for two reasons. First, the coarsest percentiles are the most important for roughness parameters. Second, because of the cumulative nature of percent finer calculations to determine the statistical parameters, errors originating from coarser grain classes accumulate into the finer grain classes (Chapter 2.3).

The Python code for the final calibration method used in this study can be seen in Appendix C. This calibration is dependent on the study site.

3.5   Sampling method

This study is based on three main field campaigns, which will be described in chronological order.

The initial field campaign occurred on October 26, 2022, at the PHSD. This campaign was based on the existing sampling method at the PHSD (Bosma, 2023). It involved the sampling of eight alongshore rows, with sediment collected at five distinct cross-shore elevations at each row: +1m NAP, +0.75m NAP, +0.5m NAP, +0.25m NAP, and 0m NAP. The sampling locations are shown in Figure 3.13. At each location, the sediment was photographed and collected for lab sieving. In total 34 sediment samples were both photographed and gathered for subsequent lab sieving.
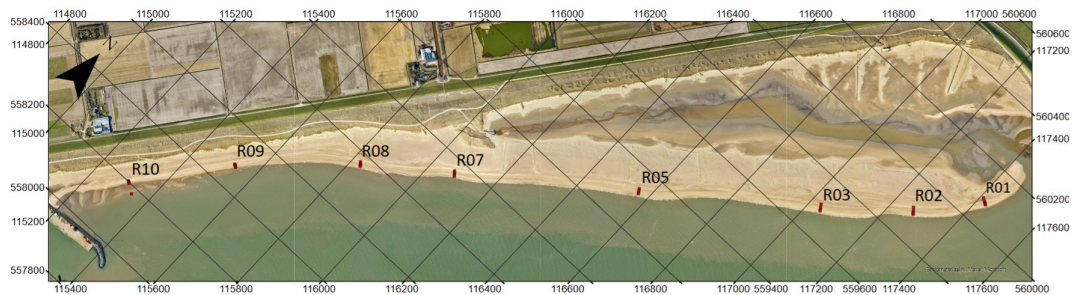
Figure 3.13: Overview of all points sampled and photographed at the PHSD at 26/10/22.

In total 34 sediment samples were both photographed and collected.

Additionally, high-resolution cross-shore photographs were taken at multiple alongshore locations. The photographs were taken at every 1 m or 0.5 m cross-shore interval to obtain a detailed view of the cross-shore sediment evolution of the intertidal beach.

The second field campaign was on December 1, 2022, at Egmond aan Zee. In this instance, three rows were sampled along the beach. At each of these rows, sediment samples were acquired and photographed at three specific cross-shore positions: at the dune toe (position 1), the middle of the beach (position 2), and at the waterline (position 3). In total 9 samples were photographed and sieved. The precise locations of these sampling points are shown in Figure 3.14.
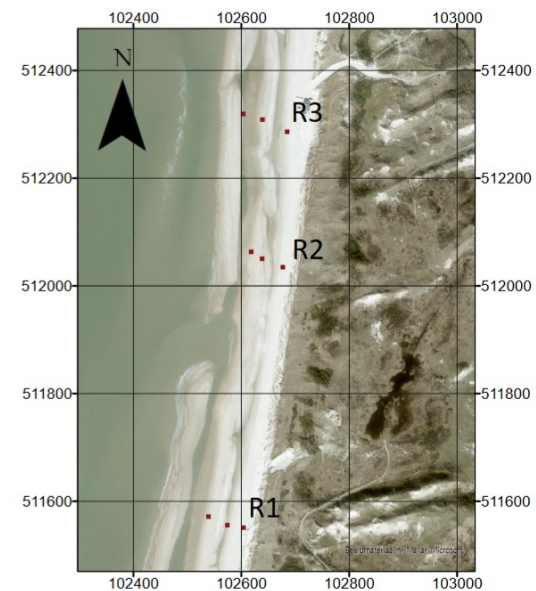
The third field campaign occurred on April 21, 2023, again at the PHSD. This campaign solely involved capturing images. A total of 24 cross-shore rows were investigated, resulting in the collection of 337 images in total. The positions of these collected images are shown in Figure 3.15.

Figure 3.14: Map with all locations where sediment is sampled and photographed, near Egmond aan Zee at 01/12/22.
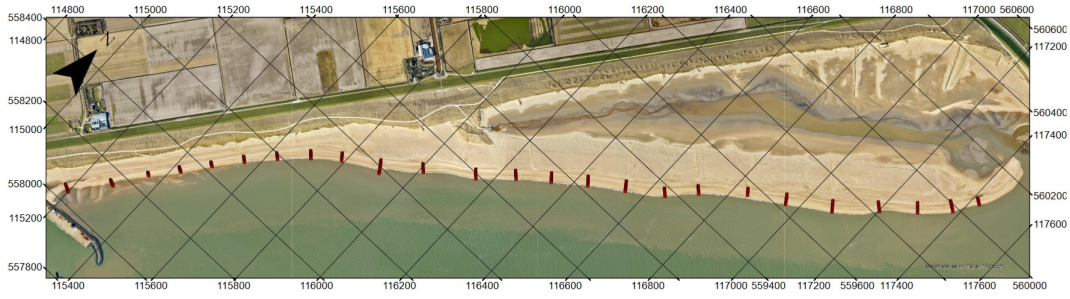
Figure 3.15: Overview of all 337 photographed positions at the PHSD at 21/04/23.

Chapter 4

Results

The aim of the first two field campaigns (Chapter 3.5) was to collect sediment samples for mechanical sieving analysis and to take in situ photographs of the sediment. This gives the ability to compare both grain size analysis techniques. Figure 4.1, shows the difference between the results obtained from mechanical sieving and pyDGS from the first field campaign at the PHSD. This represents the error of pyDGS compared to sieve analysis.

To enable a useful comparison between both methods a conversion of the results from pyDGS from an area-based to a volume-based measure is needed (Chapter 2.4).

This chapter will first show the results from the General Conversion Method, after that the results of the novel conversion method will be shown; at last, the new method will be applied to high spatial resolution sampling, underscoring the potential of this method.
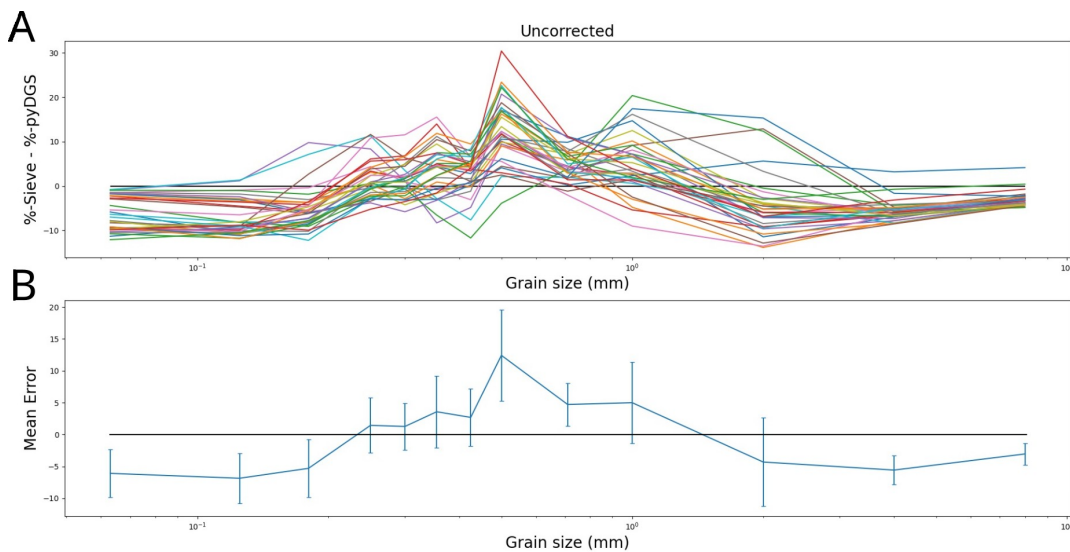
Figure 4.1: A) Difference (error) between the sieved results and the pyDGS results, where each coloured line represents a different location at the PHSD. B) The mean error from the A), together with the standard deviation. The black horizontal line indicates when the result of both methods are the same.

4.1   General conversion method

4.1.1   Prins Hendrik Sanddike

As described in Chapter 3.5 a total of 34 samples were processed using both mechanical sieving and pyDGS. Figure 4.2 shows the method of Proffitt (1980), where the percentage of each size class from the sieve analysis is divided by the percentage of each size class from pyDGS and plotted on a log-log scale. When the fraction of sieve percentage and pyDGS percentage is 1 it means that the percentage of both methods are the same. When it is larger than 1 it means that the sieve percentage is larger than the pyDGS, i.e. pyDGS underestimates the grain size class. Figure 4.2 shows a pattern where pyDGS tends to underestimate both the finest grains (0.063 - 0.180 mm) and the coarsest (1 - 8 mm).

The exponent for the conversion method (Eq. 3.6) can be derived empirically (Proffitt, 1980). This can be done for different locations (e.g. for a single sample) and for different scales (e.g. depending on grain size). Table 4.1 shows, for each individual location, the exponent (x) from the fitted line and the mean coefficient of determination (r^2) and RMSE for each location.

Using a single line for each individual location does not accurately describe the data, with a mean r^2 of 0.123 (ranging from 3.93x10^{-4} to 0.481). Applying a single line fit to all locations results in a r^2 of 3.37x10^{-5}, which is insufficient to correct all locations.

As described above, Figure 4.2 shows a pattern, where the ratio of both methods increases from 0.0625 to 0.500 mm, and then decreases from 0.500 to 8 mm. To more accurately describe this dependency on the grain size, two separate trendlines were fitted to each section, each with its own exponent (x).

Table 4.1 shows the exponents for both fits and their corresponding r^2 and RMSE, with mean r^2 values of 0.831 and 0.817, respectively. The fit to all data also shows a better fit relative to using only a single line.

Figure 4.2: General conversion log-log plot, for all samples at the PHSD collected at 26/10/22. A) using a single line; B) using two line with boundary at 0.500 mm (black lines). The different colours represent the different locations, where A, B, C, D and E are located at cross-shore elevations of +1, +0.75, +0.50, +0.25 and 0 metre NAP respectively. The red-dotted line is the 1:1 line.

Location	Single Line			Two Lines					
	x	r^2	RMSE	x	r^2	RMSE	x	r^2	RMSE
R01 A	-0.176	0.0117	1.32	3.28	0.955	0.761	-2.23	0.952	1.02
R01 B	-0.324	0.0382	1.29	2.48	0.852	1.27	-2.57	0.956	0.814
R01 C	-0.314	0.0338	1.09	2.99	0.963	0.643	-2.37	0.783	1.64
R01 D	0.0240	0.000393	1.09	2.75	0.823	1.41	-1.03	0.674	0.469
R01 E	-0.331	0.0755	1.68	1.30	0.327	2.23	-1.26	0.877	0.379
R02 A	0.107	0.0105	0.690	2.09	0.738	0.685	-1.01	0.707	0.680
R02 B	-0.104	0.00944	1.02	2.35	0.772	1.39	-1.01	0.925	0.318
R02 C	-0.414	0.141	0.811	1.68	0.685	0.604	-1.68	0.936	0.597
R02 D	0.573	0.177	1.02	2.56	0.839	0.827	-1.06	0.774	0.759
R03 A	-0.545	0.326	2.98	0.800	0.191	3.64	-0.470	0.671	0.232
R03 B	0.529	0.451	0.569	1.81	0.806	0.537	0.197	0.313	0.409
R03 C	0.566	0.164	0.991	2.79	0.895	0.850	-1.04	0.786	0.632
R05 A	0.444	0.224	0.916	1.61	0.705	0.777	0.102	0.0107	0.935
R05 B	-0.103	0.00911	1.55	2.67	0.898	1.71	-0.801	0.976	0.0567
R05 C	0.262	0.0475	0.997	2.74	0.964	0.315	-1.18	0.803	0.493
R07 A	0.141	0.0188	1.03	2.41	0.701	1.58	-0.431	0.591	0.202
R07 B	-0.954	0.292	1.89	2.61	0.865	2.03	-2.36	0.956	0.342
R07 C	0.0310	0.000668	1.10	3.23	0.917	0.973	-1.19	0.985	0.217
R07 D	0.522	0.109	1.29	3.22	0.930	1.21	-1.45	0.975	0.317
R07 E	-0.540	0.1105	1.21	2.51	0.939	0.604	-2.49	0.883	1.25
R08 A	-0.316	0.0263	1.30	3.58	0.855	1.18	-2.70	0.910	2.14
R08 B	-1.31	0.481	1.90	2.71	0.899	0.892	-2.72	0.951	0.982
R08 C	-0.222	0.0136	1.53	3.66	0.972	0.768	-2.63	0.867	2.01
R08 D	0.366	0.0537	1.19	3.11	0.848	1.59	-1.49	0.977	0.318
R08 E	0.616	0.151	1.27	3.25	0.919	1.06	-1.27	0.971	0.326
R09 A	-0.137	0.00633	1.71	2.86	0.951	1.02	-2.70	0.982	0.851
R09 B	-0.384	0.0462	1.18	3.01	0.880	1.03	-2.51	0.819	1.86
R09 C	0.807	0.2704	1.32	2.87	0.916	0.413	-1.26	0.881	0.796
R09 D	0.114	0.0106	1.05	2.51	0.860	0.695	-1.12	0.931	0.314
R10 A	0.539	0.114	1.41	3.06	0.979	0.255	-1.81	0.983	0.323
R10 B	0.717	0.316	1.02	2.73	0.939	0.124	-0.712	0.791	0.577
R10 C	-0.271	0.0242	1.48	2.88	0.942	0.239	-2.62	0.881	1.81
R10 D	0.550	0.255	1.48	2.38	0.912	0.268	-0.618	0.605	0.799
R10 E	0.385	0.149	1.51	1.37	0.628	0.464	-0.914	0.682	0.818
Total	-0.00864	3.37e-5	1.25	2.54	0.812	1.04	-1.81	0.618	0.482

Table 4.1: Exponents of the power-fits of the general conversion plot (Figure 4.2) for all locations at the PHSD sampled at 26/10/22, using a single line and two lines to describe the data, where x is the exponent of the fit. With the corresponding mean coefficient of determination (r^2) and Root Mean Square Error (RSME) for each fitted line.

Figure 4.3: Individually corrected samples using the two lines method from the general conversion method. Where the colours represent the different cross-shore position on R07 and the dotted black line is the sieved sample. The plot on the bottom right shows the sieved percent versus the pyDGS percent.

Figure 4.3A-E shows the cumulative grain size distribution curves for all locations at row 7 (R07), with percent finer on the y-axis. The coloured lines are the corrected output from pyDGS, using two trendlines. The dotted black lines are the result of the sieving method. The results from pyDGS at locations R07 C and R07 D show the most resemblance with the sieve data. Both of these locations show a low RMSE, of 0.973 and 0.217 for C, and 1.21 and 0.317 for D. Furthermore, these locations have a high r^2, 0.917 and 0.985 for C, and 0.930 and 0.975 for D.

The high agreement with the fit for the specific grain size does not necessarily indicate that this part is close to the sieve data. For example, location R07 A for the smallest grain sizes (<0.500 mm), pyDGS shows a concave curve, while the sieve curve is convex. However, the trend fitted to the smallest grains is better compared to the coarser grains (r^2 0.710 and 0.591 respectively). This is the result of how the grain size distribution curve is shown, using percent finer on the y-axis, which is a cumulative measure from the highest percent to the lowest ones.

Table 4.2 shows the errors for each percentile for all locations, which are corrected for each individual location using two trendlines. The errors are all below 20 percent and the errors increase with the finer percentiles, which is a result of the cumulative nature of the percent finer computation. The computation starts with the coarsest grain size class and subsequent classes introduce their individual errors as they contribute to the cumulative value. As a consequence, the error for the finest grain class becomes more pronounced due to the cumulative errors from all other classes.

Percentile	D_{10}	D_{16}	D_{25}	D_{50}	D_{84}	D_{90}
RMSE	0.0496 (19%)	0.0540 (15%)	0.0752 (13%)	0.133 (17%)	0.522 (15%)	0.649 (12%)
MAE	0.0451 (17%)	0.0486 (13%)	0.0562 (9%)	0.0849 (11%)	0.388 (11%)	0.429 (7%)

Table 4.2: Root Mean Square Errors (RMSE) and Mean Absolute Errors (MAE) in percentile estimates each using the individually corrected technique of the general conversion method for the data of PHSD. The value in parentheses is the error normalised by grain size (Appendix D).

The conversion method, as shown by the relatively small errors in Table 4.2, can effectively convert the results from pyDGS to line up with the sieve curve. However, the main limitation of this method is that it requires a sieved sample for each correction. Table 4.1 shows that the exponent for each individual location varies greatly, where the variance $\left(\sigma^2 = \frac{\Sigma(X-\mu)^2}{N}\right)$ is 0.430 and 0.718 respectively for both trendlines. There is no clear spatial distribution visible in exponents, or a relation between the exponents at a location.

A more practical approach is to use the exponents that are fitted to all data-points. Table 4.3 shows the errors for each percentile, where the data is corrected using this approach. Again the coarse percentiles show the smallest errors ($\approx$ 20 %), but for the fine percentiles, the error increases up to 60 percent.

Percentile	D_{10}	D_{16}	D_{25}	D_{50}	D_{84}	D_{90}
RMSE	0.157 (63%)	0.115 (42%)	0.112 (37%)	0.165 (22%)	0.741 (21%)	1.18 (21%)
MAE	0.122 (49%)	0.0952 (35%)	0.0927 (31%)	0.111 (15%)	0.482 (14%)	0.760 (14%)

Table 4.3: Root Mean Square Errors (RMSE) and Mean Absolute Errors (MAE) in percentile estimates of the general conversion method using the the correction exponent of the two-lines 'total'-fit for the data of PHSD. The value in parentheses is the error normalised by grain size (Appendix D).

4.1.2 Egmond aan Zee

At Egmond aan Zee, 9 samples were photographed for pyDGS and collected for sieve analysis. Samples were collected at three alongshore locations (R01, R02, R03) from three distinct cross-shore: at the dune toe (1), at the middle of the cross-shore beach (2), and at the waterline (3). Figure 4.4 shows the fraction of the sieved percentage and pyDGS percentage plotted against the grain size, displayed on a log-log scale, similar to Figure 4.2.

It is observed that for the finest grains (<0.180 mm) there is a steep increase, where from 0.250 mm to 0.500 mm pyDGS underestimates the percentages of each grain size class. From 0.710 mm, pyDGS overestimates the grain size percentages and there is considerable variation between the different locations, and the lines seem to flatten out.

The points near the waterline (R.._3) for a grain size larger than 1 mm are the three points closest to the 1:1 line (Figure 4.4), while locations 1 and 2 (R.._1 and R.._2) show significant over-estimations by pyDGS. This observation suggests that the grains near the waterline are coarser in comparison to those closer to the dunes (location 1,2). This could indicate that for the beach of Egmond aan Zee, a spatial correction exponent could be used, i.e. based on the position on the beach a different correction exponent could be used.
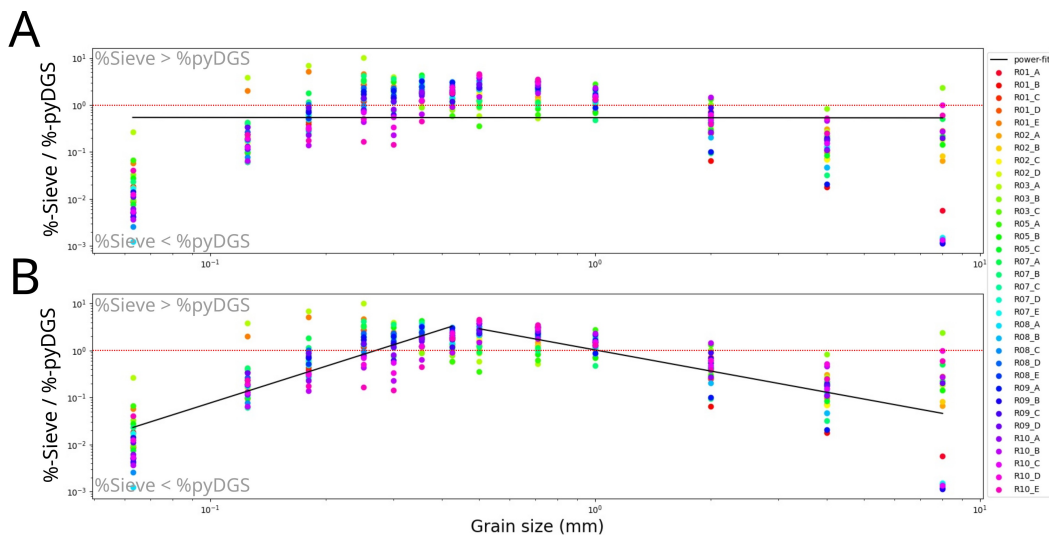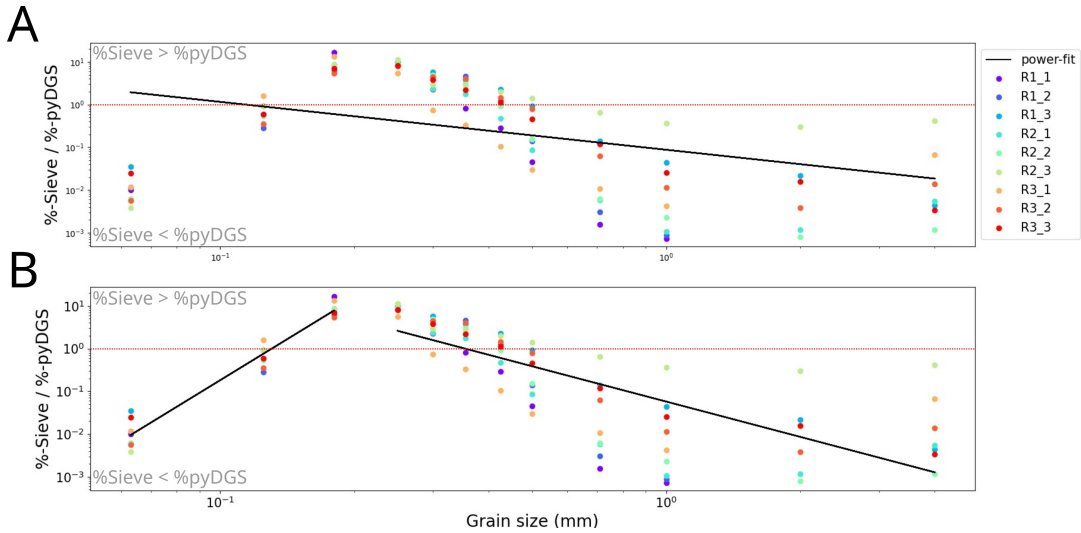
Figure 4.4: General conversion log-log plot, for all samples at Egmond aan Zee collected at 01/12/22. A) using a single line; B) using two line with boundary at 0.500 mm (black lines). The different colours represent the different locations, where there are three rows at different alongshore locations (R1,R2,R3) and at each row there three cross-shore locations located at the dune toe (1), middle of the beach (2) and near the waterline (3). The red-dotted line is the 1:1 line.

Location	Single Line			Two Lines					
	x	r^2	RMSE	x	r^2	RMSE	x	r^2	RMSE
R01 1	-1.79	7.60e-3	5.88	7.02	0.998	1.24	-7.25	0.976	0.163
R01 2	-1.21	0.0263	4.13	5.84	0.973	1.16	-7.52	0.951	5.45
R01 3	-1.25	2.30e-3	4.02	5.05	0.975	1.31	-2.91	0.952	1.01
R02 1	-1.62	2.90e-3	3.57	6.87	0.996	0.809	-3.14	0.691	2.75
R02 2	-1.83	0.0136	3.95	6.83	0.998	0.552	-3.75	0.825	2.12
R02 3	0.155	0.115	3.87	7.39	0.995	1.05	-1.11	0.735	2.53
R03 1	-0.889	0.0194	4.29	6.76	0.998	1.08	-1.45	0.307	1.83
R03 2	-1.04	2.00e-4	3.09	6.48	0.997	0.407	-2.88	0.812	1.44
R03 3	-1.28	5.00e-4	3.20	5.30	0.990	0.789	-2.85	0.946	1.28
All	-1.12	3.00e-6	4.00	6.40	0.964	2.16	-2.75	0.577	2.51

Table 4.4: Exponents of the power-fits of the general conversion plot (Figure 4.4) for all locations at the Egmond aan Zee sampled at 01/12/22, using a single line and two lines to describe the data, where x is the exponent of the fit. With the corresponding mean coefficient of determination (r^2) and Root Mean Square Error (RSME) for each fitted line.
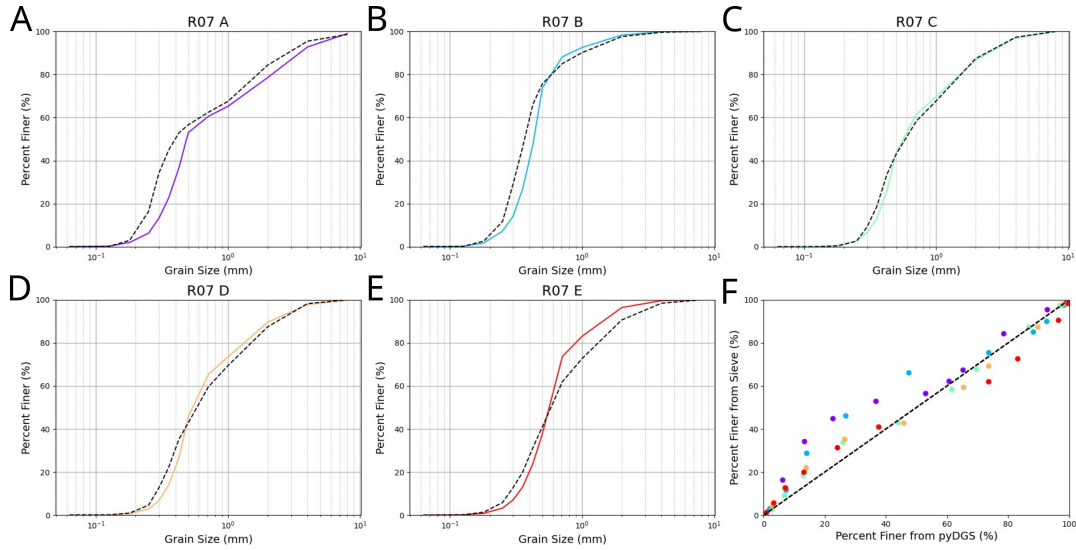
Again the general conversion method is used where two (straight) lines are fitted to the data, now with the boundary between 0.180 mm and 0.250 mm. For the finest grains, the straight line can be fitted quite well, for each individual location and for all locations at once, with a mean coefficient of determination (r^2) of 0.991 and 0.964 respectively. However, for the second part ($\geq$0.250 mm) the points do not follow a clear straight line anymore, where it starts to flatten out for the grain size greater than 1 mm. This can be seen in Table 4.4 where the coefficient of determination is lower, with a mean of 0.799.

Table 4.5 shows the error statistics of the percentiles using the individual locations for the correction. The coarsest percentiles show errors below 10%, where the D_{50} shows the largest RMSE and MAE of 36% and 18% and for the finest percentiles the errors reduce to below 30%.

Percentile	D_{10}	D_{16}	D_{25}	D_{50}	D_{84}	D_{90}
RMSE	0.00722 (27%)	0.0113 (27%)	0.0142 (23%)	0.170 (36%)	0.0625 (5%)	0.176 (8%)
MAE	0.00476 (18%)	0.00720 (17%)	0.0110 (17%)	0.0821 (18%)	0.0491 (4%)	0.114 (5%)

Table 4.5: Root Mean Square Errors (RMSE) and Mean Absolute Errors (MAE) in percentile estimates each using the individually corrected technique of the general conversion method for the data of Egmond aan Zee. The value in parentheses is the error normalised by grain size (Appendix D).

Table 4.6 shows the errors of the percentiles using the two total trendlines. Relative to the individual correction method the errors increase, except for the median grain size (D_{50}) which remains almost constant.

Percentile	D_{10}	D_{16}	D_{25}	D_{50}	D_{84}	D_{90}
RMSE	0.0126 (48%)	0.0197 (47%)	0.0296 (47%)	0.162 (35%)	0.365 (27%)	0.576 (27%)
MAE	0.0104 (39%)	0.0164 (39%)	0.0249 (39%)	0.118 (25%)	0.182 (13%)	0.318 (15%)

Table 4.6: Root Mean Square Errors (RMSE) and Mean Absolute Errors (MAE) in percentile estimates of the general conversion method using the the correction exponent of the two-lines 'total'-fit for the data of Egmond aan Zee. The value in parentheses is the error normalised by grain size (Appendix D).

4.2   Novel correction method

In the following section again both study sites will be discussed, however now the novel correction (Section 3.4.2) method has been applied.

4.2.1   Prins Hendrik Sanddike

Figure 4.5A shows the difference between the sieve percentage and the corrected pyDGS percentage, where the percentage from pyDGS is corrected using the grain shapes as described in Section 3.4.2. Figure 4.5B shows the corresponding mean, with the standard deviation for each grain size class. The pattern shows that the grain size classes in between have the largest error and the largest standard deviation, while the finest and coarsest grain size classes have the smallest errors. For the coarsest (>1 mm) and finest (<0.425 mm) grain size classes the mean error is reduced to <5%. The errors compared to the general conversion method (Section 4.1.1), are reduced where now the maximum mean error is reduced from 63 % to 30 %.

Figure 4.5: A) Difference (error) between the sieved results and pyDGS results, post-correction using the novel correction method. The different colours represent the different locations, where A, B, C, D and E are located at cross-shore elevations of +1, +0.75, +0.50, +0.25 and 0 metre NAP respectively. B) The mean error from the aforementioned plot, together with the standard deviation. The black horizontal line indicates when the result of both methods are the same.

Given that this correction approach is applied universally to all images, it is comparable to the general conversion method that uses the two-line 'total'-line correction (Table 4.3). The comparison of both tables shows that the errors associated with D90, D84, D50, and D25 have similar errors. However, for D16 and D10, this novel method has provided a reduction of 4% in RMSE and 7% in MAE for the D16, respectively. And for the D10, the RMSE has been diminished by 24%, and with a 19% reduction in MAE.

As described in Chapter 3.4.2 is the new correction method focused on the coarsest grains. The result is that the errors of these coarsest grain percentiles are comparable to the general conversion method. While the finest percentiles still show a significant improvement in error reduction.

Percentile	D_{10}	D_{16}	D_{25}	D_{50}	D_{84}	D_{90}
RMSE	0.0981 (39%)	0.102 (38%)	0.115 (38%)	0.195 (26%)	0.702 (20%)	1.09 (19%)
MAE	0.0749 (30%)	0.0750 (28%)	0.0918 (30%)	0.154 (21%)	0.424 (12%)	0.672 (12%)

Table 4.7: Root Mean Square Errors (RMSE) and Mean Absolute Errors (MAE) in percentile estimates of the novel correction method for the data of PHSD. The value in parentheses is the error normalised by grain size (Appendix D).
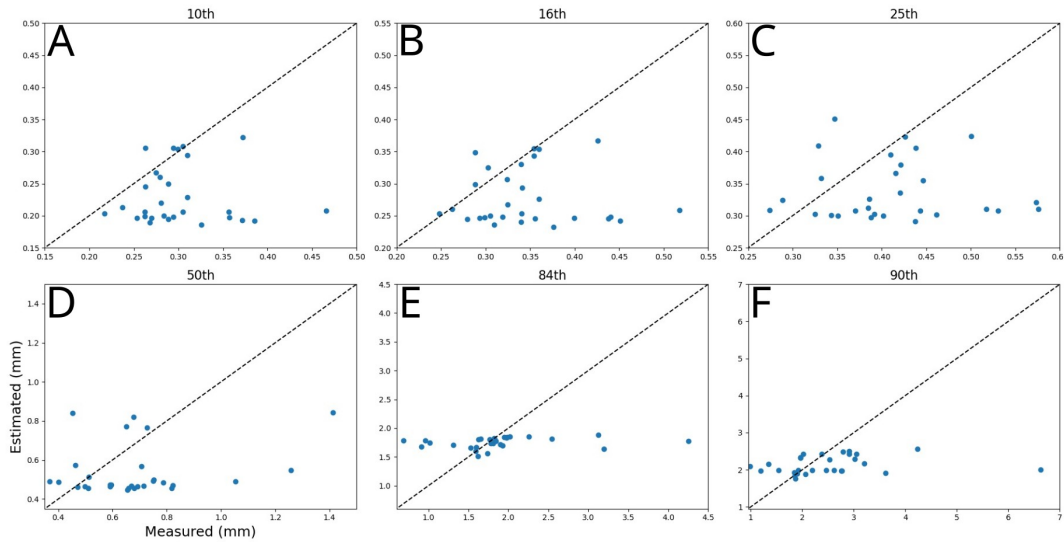
Figure 4.6: Comparison of measured/sieved (x-axis) and estimated/pyDGS (y-axis) percentiles for D_{10} (A), D_{16} (B), D_{25} (C), D_{50} (D), D_{84} (E) and D_{90} (F) based on PHSD dataset. The dashed black line is the 1:1 line.

Figure 4.6 shows the relation between the measured (sieved) and estimated (pyDGS) percentiles. It shows that pyDGS provides estimations within a specific range of grain sizes, with the pattern more pronounced in Figure 4.6 D,E,F. While the measured (sieved) percentiles show a range of values, the estimated values from pyDGS remain more concentrated. This is a result of the correction method, which tends to align with the characteristics of the average grain shape. This could explain why for each percentile plot a few points are situated on the 1:1-line. The identification of outliers with distinct shapes is more difficult to capture with this correction method. This will be further explained in the Discussion (Chapter 5).

4.2.2   Egmond aan Zee

Figure 4.7 is the same as Figure 4.5, but now the grain shapes are corrected based on the photographs and sieved samples from the beach near Egmond aan Zee. Figure 4.7 shows a pattern where the grain sizes >0.425 mm have the smallest errors. While the finer grain sizes (0.180-355 mm) have the largest errors. Noteworthy is the large standard deviation for the 0.180 mm grains, with large a standard deviation (ranging from -30 to 10).
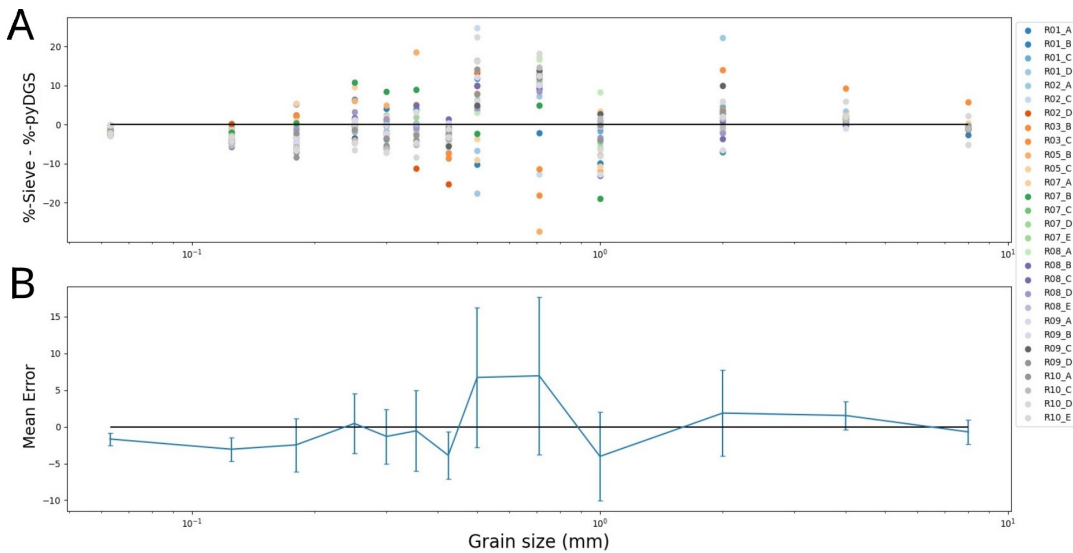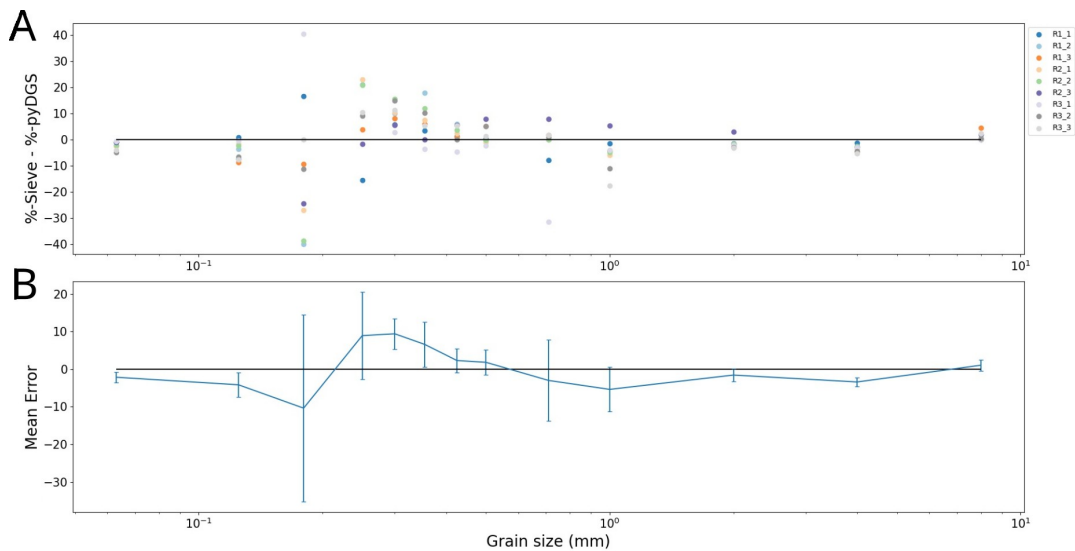

Figure 4.7: A) Difference (error) between the sieved results and pyDGS results, post-correction using the novel correction method. The different colours represent the different locations, The different colours represent the different locations, where there are three rows at different alongshore locations (R1,R2,R3) and at each row there three cross-shore locations located at the dune toe (1), middle of the beach (2) and near the waterline (3). B) The mean error from the aforementioned plot, together with the standard deviation. The black horizontal line indicates when the result of both methods are the same.

Percentile	D_{10}	D_{16}	D_{25}	D_{50}	D_{84}	D_{90}
RMSE	0.0100 (38%)	0.0153 (36%)	0.0225 (36%)	0.149 (32%)	0.458 (33%)	0.650 (31%)
MAE	0.00751 (28%)	0.0112 (27%)	0.0165 (26%)	0.0877 (19%)	0.286 (21%)	0.475 (23%)

Table 4.8: Root Mean Square Errors (RMSE) and Mean Absolute Errors (MAE) in percentile estimates of the novel correction method for the data of Egmond aan Zee. The value in parentheses is the error normalised by grain size (Appendix D)

The errors corresponding to each percentile can be found in Table 4.8. These errors should be compared to Table 4.6 because both correction methods are similar. The RMSE for D90 and D84 exhibits an increase of 4-6%, accompanied by an 8% rise of the MAE. While, both the RMSE and MAE values for D50, D25, D16, and D10 all demonstrate a decrease of approximately 10%.
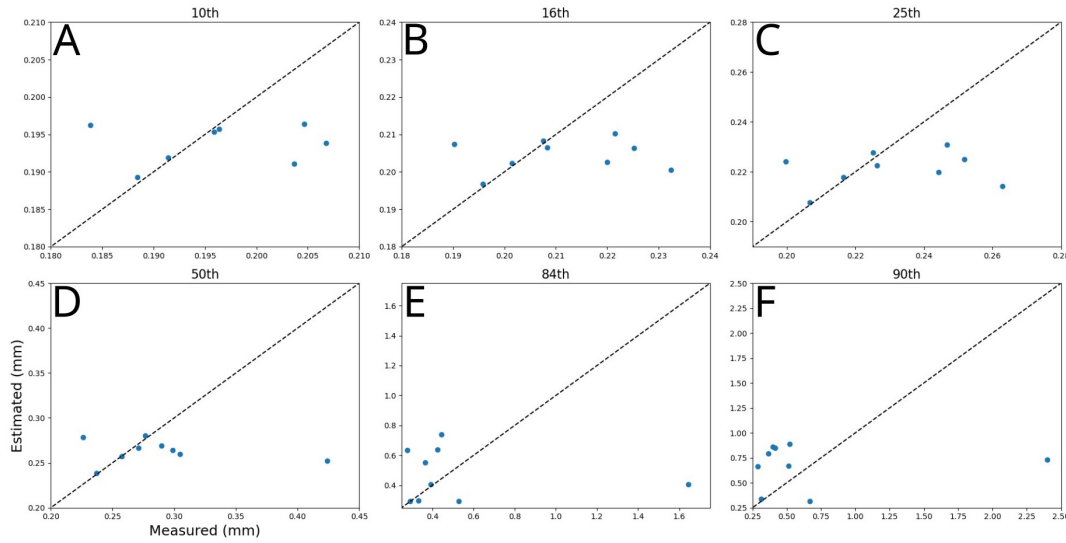
Figure 4.8: Comparison of measured/sieved (x-axis) and estimated/pyDGS (y-axis) percentiles for D_{10} (A), D_{16} (B), D_{25} (C), D_{50} (D), D_{84} (E) and D_{90} (F) based on Egmond aan Zee dataset. The dashed black line is the 1:1 line.

Figure 4.8 shows the estimated versus the measured percentiles. Again the estimated percentiles by pyDGS are reduced to a relatively small range of values. Where the range of the estimated values is approximately 0.5 times the measured range. Figure 4.8 shows for 10th to 50th percentiles (Figure 4.8A,B,C,D) four values that are almost perfectly on the 1:1 line, which means a well-estimated value. These four points are related to the same locations/photos. Visual inspection of these images does not indicate a clear difference from the other images, while the other images show a larger deviation.

The difference between the results of Egmond aan Zee and PHSD can be attributed to the difference in sample size, where the Egmond aan Zee data comprises only 9 samples, in contrast to the 34 samples from the PHSD data.

4.3   High spatial resolution sampling
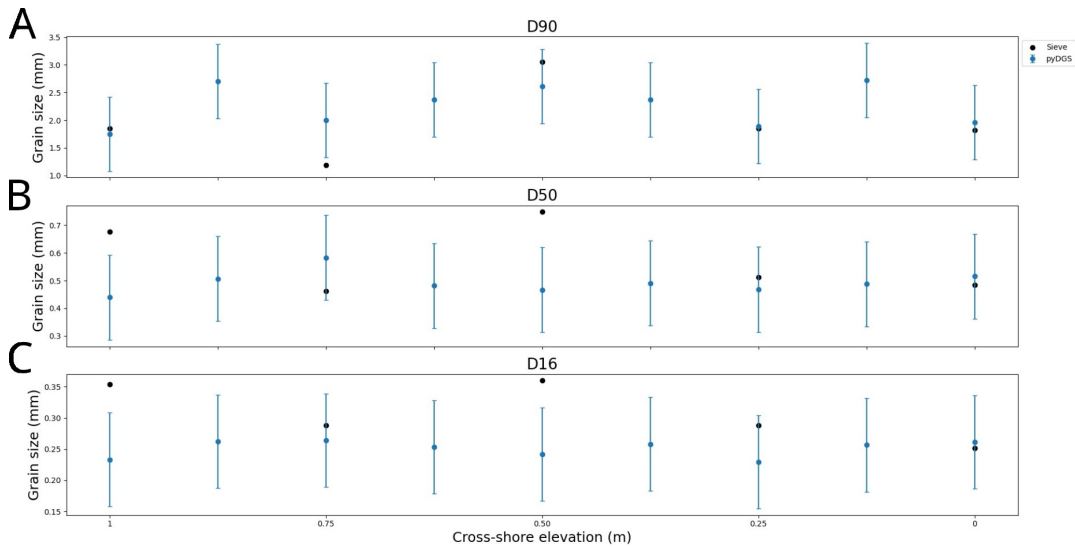
Figure 4.9: Cross-shore evolution of grain size parameters D_{90} (A), D_{50} (B), and D_{16} (C). The x-axis represents cross-shore elevation, corresponding to the specified sampling locations. Black dots indicate sieving results, while the blue dots represent pyDGS results, with error bars indicating mean absolute error. In between the sampled locations photographs have been taken.

Figure 4.9 shows the grain sizes obtained from sieve analysis, at the same locations where photographs were taken. The figure also includes photographs taken at intervals between the sampled locations. It shows that the trends observed in the photographed samples closely mirror those in the sieved result. Particularly for the D_{90} (Figure 4.9A), the same trends are visible: as the grain size increases in the sieved results, the photographed result shows the same increase.

For the next step (Figure 4.10) a different method is applied, where a (very) high resolution is used. This approach involves capturing images at very close intervals along the cross-shore profile, specifically at intervals of 0.5m (Figure 4.10A) and 1m (Figure 4.10B). This shows the potential use of this method, where the quantity of the captured data outperforms the high confidence of the exact grain size.

The main advantage of sampling at this high resolution is the ability to ignore outliers and focus on identifying the visible trends.

For example, Figure 4.10A, the first 6 meters shows a nearly constant D_{90}, followed by an increase around the 8-10 meter cross-shore, after which D_{90} returns back to the previous range. A similar pattern can be observed in Figure 4.10B. However, for the D_{50} and the D_{16} (Figure 4.10C,D,E,F), the observed patterns are less pronounced.
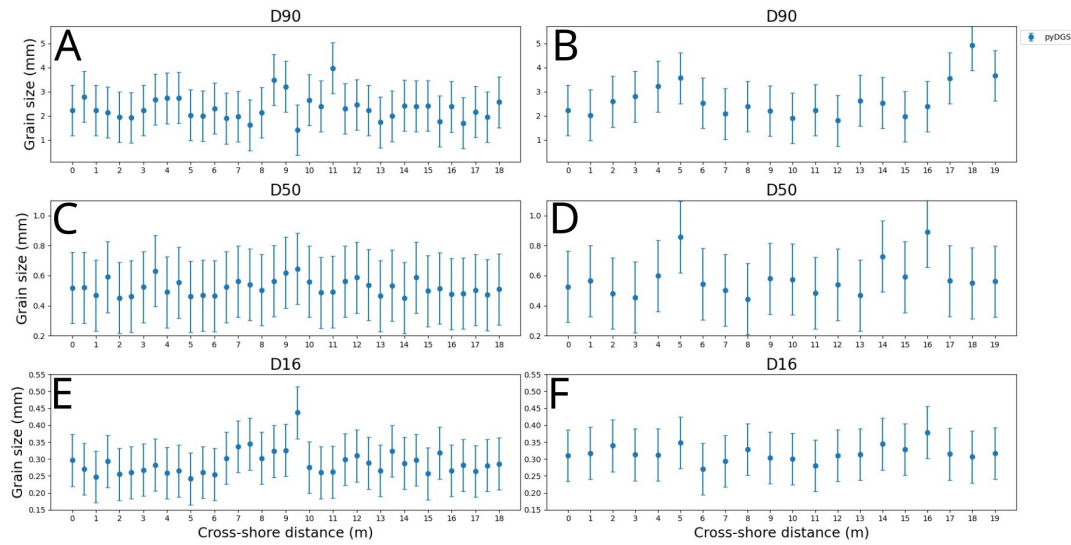
Figure 4.10: Cross-shore evolution of grain size parameters D_{90} [mm] (A,B), D_{50} [mm] (C,D), and D_{16} [mm] (E,F). Two lines with photos taken every 0.5 m (A,C,E) and every 1 m (B,D,F). The error bar shows the mean absolute error.

4.3.1 High resolution mapping of the PHSD

As described in Chapter 3.5 the last field campaign aimed to capture the spatial grain size distribution of the entire beach of the PHSD. It involved only capturing images, coupled with GPS-data.
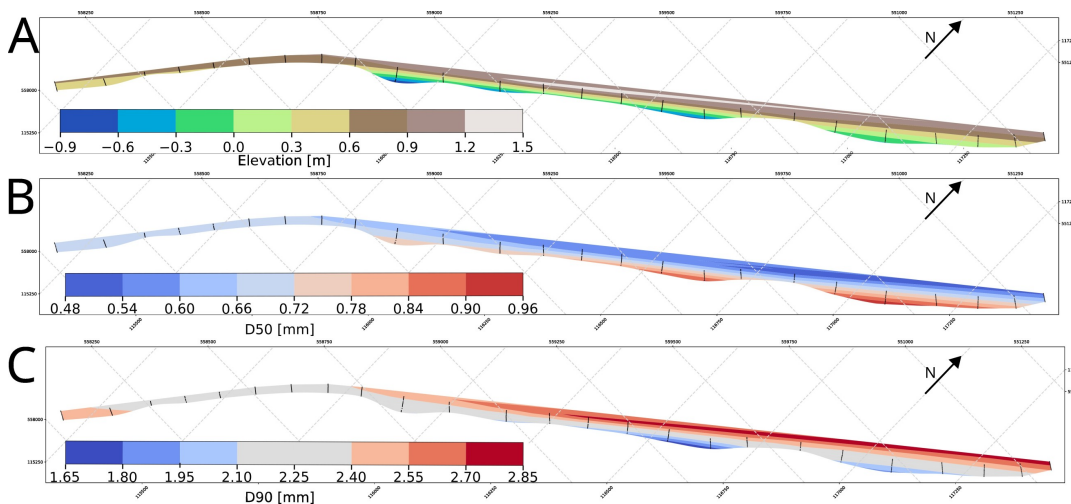
Figure 4.11: A) Elevation map B) D_{50}-map C) D_{90}-map of the intertidal beach area of the PHSD from the data of the last field campaign (21/04/23).

Figure 4.11A) shows the elevation map of the intertidal beach area of the PHSD. Figure 4.11B and C shows the D_{50}-map and the D_{90}-map, respectively.
In the South-Western section of the PHSD (Figure 4.11B), a relatively consistent median grain size of approximately 0.70 mm is visible. Along the bend in the beach, the median grain size decreases: farther from the waterline, the median grain size stays around 0.60 mm, gradually increasing towards the waterline up to 0.85 mm.
For Figure 4.11C the D_{90} is in the South-Western part around 2.50 mm and decreases

in the North-Eastern direction to around 2.25 mm. After the bend the D_{90} decreases from the landward edge towards the waterline, transitioning from approximately 2.75 mm to 2.0 mm. This pattern is the opposite of the D_{50}-map.
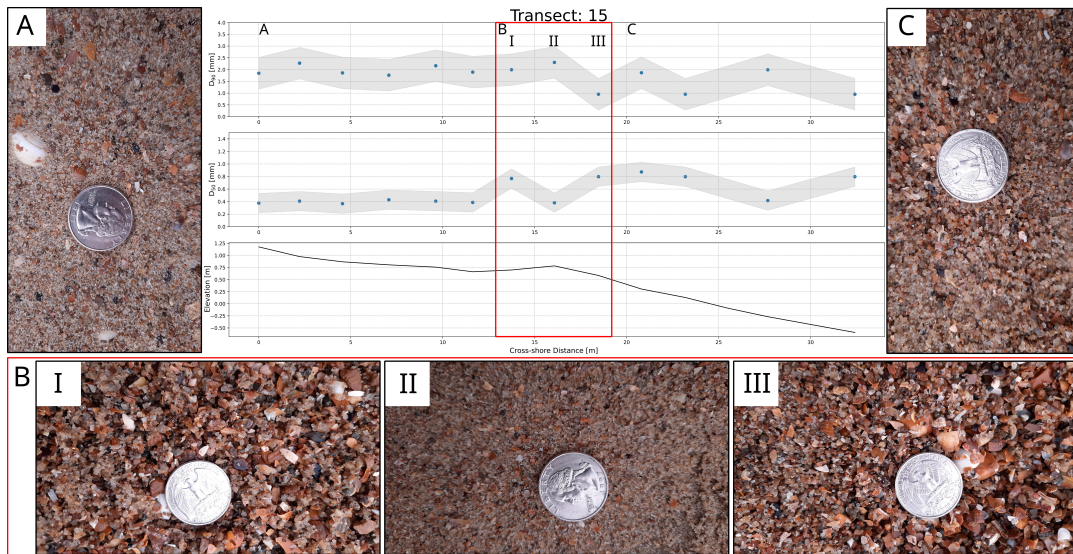
Figure 4.12: Cross-shore evolution of the D_{90} [mm], D_{50} [mm] and the elevation [m NAP] (in center from top-to-down), surrounded by the corresponding photographs of sediment.

The interpolated maps of Figure 4.11 lose the fine details of the grain size distribution evolution. Figure 4.12 shows in the center (from top-to-down) the spatial evolution of the D_{90}, D_{50} and the elevation. This gives the ability to study a transect in high detail. Around the center are the images of sediment added to help visualise the types of sediment around each part. The transect is divided into three sections A, B and C. Where in section A the grain size remains relatively constant, with the D_{90} around 2.0 mm and a D_{50} of 0.40 mm, while the elevation gradually decreases (slope: 1:25 m/m). In section B, a small bump is present, where the elevation increases by 20 cm. The D_{90} seems to slightly increase on the top of this bump and the D_{50} shows a decrease on the top, while on both sides of the bump the D_{90} decreases and the D_{50} increases. After this bump in section C, the elevation again decreases (slope: 1:15 m/m), where the D_{90} decreases further and the D_{50} increases.

Chapter 5

Discussion

5.1 From area-by-size to volume by weight

In this thesis, the pyDGS method has been proposed as a potential alternative to mechanical sieving. However, before it can be considered a suitable replacement, a conversion from the area-by-size measure to the volume-by-weight measure must be performed. As shown by Diplas and Fripp (1992), these measures are not directly comparable. Without this conversion, the results of both methods can not be used interchangeably, and thus pyDGS would not be a reliable substitute for mechanical sieving.

Assuming a theoretical approach such as the cube model, it showed that the area-by-size (relative to an area-by-area measure) of the smallest grains/cubes are over-represented (57%) and the largest grains/cubes are under-represented (14%). This should result in a positive linear trend on log-log plots, using the approach of Proffitt (1980). However, the log-log plots of pyDGS data from both the PHSD and Egmond (Figure 4.2 and 4.4, respectively) show a different pattern, with a positive trend for the finest grains and a negative trend for the larger grains. This indicates that pyDGS overestimates the smallest and largest grains.

The area-by-size method, as described by Kellerhals and Bray (1971) in the cube model, uses the size of each grain, while pyDGS predicts the size from a number of horizontal lines and uses statistics to calculate the total distribution. This leads to an overestimation of the coarsest grain sizes, as grains that are larger than the distance between the lines are counted multiple times. The method described in Chapter 3.3.3 does not seem to be effective in reducing this overestimation, since the increase in distance between two lines does not have to mean that the larger grain is only counted once (i.e. it is dependent on the location of the larger grains). The largest grains that are captured in the image are mainly shells and fragments of shells, which have a large surface area and a relatively small mass. This means that these large grains have two times the impact of overestimation. A potential correction for this will be introduced in Section 5.5.

5.2   General Conversion Method

The errors shown in Table 4.2 showed that the results from using the general conversion method to the PHSD are similar to the errors presented in the original paper of Buscombe (2013). This means that this method is able to correct for the area-by-size measure to a volume-by-weight. In other words, pyDGS can be applied to predict the grain size distribution of mixed sands. However, there seems to be no universal correction exponent that can be applied to all images, since the exponent varies considerably from one image to another. With the application of an average correction factor to all images, the errors increase (Table 4.3).

The exponent of the general conversion method can be characterised as a shape parameter (Harvey et al., 2022; Cuttler et al., 2017), which depends on the grain size, sediment sorting, porosity and the sampling depth (Diplas and Sutherland, 1988; Diplas and Fripp, 1992; Bunte and Abt, 2001). This conversion method was established before image-based grain size analysis was a widely used alternative for areal sampling. This sampling technique involves using an adhesive (e.g. wax (Little and Meyer, 1976), tape (Proffitt and Sutherland, 1980) or epoxy (Gessler, 1968)), which is applied to the surface. The particles are then detached from the adhesive with the use of solvents. The collected sample is then sieved to determine the grain size distribution.

The variation in the exponent across the images can be linked to the different properties on which the images depend, relative to the adhesive sample collection. The following section will discuss the consequences of the method for determining the exponent and the disadvantages of using a single correction factor.

5.2.1   Method of Proffitt

The empirical method that is used to determine the value for the exponent is based on the method of Proffitt (1980). By plotting the difference between both methods against the grain size on log-log axes and fitting a power-function to the data, where the exponent of this power-function is equal to the conversion factor.

The problem with this correction approach is that the absolute estimation differences between both methods might be small, these differences are amplified on a logarithmic scale. For instance, the absolute difference between 0.1 and 1 is 0.9, but on a log-scale, this difference is a factor of 100. This results in a steep decrease in the y-values, resulting in a larger exponent. The small difference especially happens at the finest and coarsest grains, which explains why there is an observable increase trend for the finest grains and a decreasing trend for the coarsest grains.

As a result of this, it was shown in Table 4.1 that relying on a single correction factor for all grain sizes proves insufficient. The use of two trendlines improves the fits and diminishes the overall error. However, the changes in this exponent are due to the rapid changes in the y-axis.

Proffitt (1980) recommended to ignore the finest 1%, but this recommendation was based on gravel sediments and effectively means to ignore grains finer than 1 mm. Which is the grain size range of sands, which is the focus of this study. In other words, this method is not designed and well suited for sands.

Another problem with the use of the exponent is that all properties are confined to a single correction exponent. As discussed above the exponent depends on the grain size, sediment sorting, porosity and sampling depth. This makes it difficult to

explain why the exponent changes from sample to sample. Since a change in exponent can be due to a difference in grading or a difference in grain size. This exponent corrects for all these differences, which makes it difficult to find a spatial distribution of these correction exponents (e.g. an exponent at the dune toe and a different exponent close to the waterline). Table 4.1 does not tell if all differences are due to the change in grain size or due to other factors. This means that the conversion factors can not be universally applied.

5.3 Sampling depth

The sampling method for sieving remained the standard, where the sample was collected from the top 5 cm layer. While the captured image is from only the visible top-layer. The assumption that both samples are comparable only holds true when the sediment is homogeneous in depth. Which is in reality not the case, especially in a mixed sandy beach setting.

Figure 5.1 shows the different grain size distributions between both collected samples. The black line represents the grain size distribution from the image of the top-layer, while the blue line corresponds to the distribution from the sample obtained from the top 5 cm. This latter sample was spread on a tray, and images were taken and analysed with pyDGS. Notably, the depth sample (blue line) exhibits a finer fraction, while the coarse fraction remains relatively consistent. This difference can be attributed to the hiding/exposure effect, a phenomenon in which the smaller grains are hidden behind the larger grains. It could implicate that this method of surface sampling is better for flow estimations, because this already takes corrects for the

Figure 5.1: Grain size distribution curve from pyDGS from the image taken in situ (black) and from the sample spread out on a tray in the lab (blue).

hiding-exposure effect because the surface sample is what directly gets affected by the flow.

Figure 5.1 does show that both samples are not directly comparable. There are two options to correct for this. The first option is to collect the sieve sample with an adhesive. The second option is to take multiple pictures of sediment at different depths. This second option was tested in this study, although its practical implementation in the field proves challenging due to the complexity of precisely targeting specific layers and the presence of larger grains at different depths.

Figure 5.2: Schematic of various penetration depths, with each shade of grey indicates a different depth. The yellow blocks on top represent how pyDGS perceives the sample.

For the adhesive method, the sampling depth (i.e. penetration depth) is also still important, where the sampling depth is the depth at which the adhesive penetrates the ground. Figure 5.2 illustrates how different penetration depths can yield different sediment samples. It also shows how pyDGS perceives the sample. When it comes to surface sampling, the sampling depth should be limited to the top layer in order to ensure accurate results.

5.4   Novel correction method

The correction needed involves the conversion of the results from an area-by-size to a volume-by-weight measure. The proposed novel correction method involves computing the volume of each grain size class, by determining the number of grains that can fit within the image frame. This was done by estimating the volume of each grain based on its respective class.

The main strength of the new approach is that it avoids the need for generalisation using a correction parameter, thereby providing a more accurate correction method for individual images. This makes it better able to apply to larger beach sections. However, due to the complexities arising from uncertainties around grain shapes and the intrinsic variability of mixed sediments, applying this correction method can be challenging.

5.4.1   Grain shape

The grain shape uncertainty can be divided into two categories: top-view (2D) and depth (3D).

The assumption that is made on the top-view sediment is that all the grains align parallel to the image plane, i.e. the longest axis of the grain is parallel to the image plane. However, in natural sediments, grains are not always perfectly sorted in this manner. Mixed sediments, in particular, have a wide range between the coarsest and finest grains. This leads to a more diverse set of grain shapes in the sample, assuming that the grain shape is (at least partly) a function of the grain size. The presence of various grain shapes increases the likelihood of differently oriented-grains.

Because the relationship between parallel and differently oriented grains remains unknown, there is potential for two scenarios. It could be possible that when observing all grains within an image, this effect might be negligible on average, yet it could also play a significant role. The rotation of the image (Chapter 3.3.3) attempts to minimise this effect. The results of the grain size distribution vary across these distinct rotations. This variability is related to the top-view area of the grains within

this correction method. The evaluation of grain surface area takes into account two types of shapes: circles and ellipses. Where an ellipse has a larger surface area relative to a circle, this means that the number of grains in the image frame can be related to the grain shape. This correction shows that the impact of top-view grain shape on the errors is minimal.
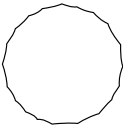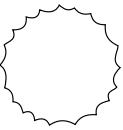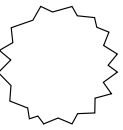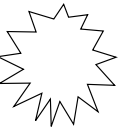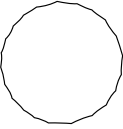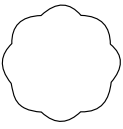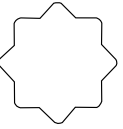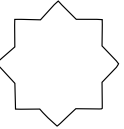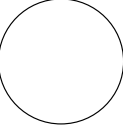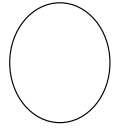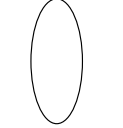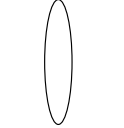
	High			Low
Regularity				
Roundness				
Sphericity				

Figure 5.3: Schematic examples of the different grain shape factors: Regularity, Roundness and Sphericity. (*Based on: Blott and Pye, 2008*)

As described in Chapter 3.4.2 are the grain shapes described by their form. However, there are several ways the shape of a grain can be described. Not only by its form but also by roundness, regularity and sphericity (Blott and Pye, 2008).

Regularity is used to describe the surface of the grain, when the surface has a lot of indentations or bumps the grain is described as irregular (Blott and Pye, 2008). The regularity is independent of the form and roundness because it is focused on the surface area.

The sphericity is the degree at which the grain can be approximated to a true sphere, where the radius is uniform in all directions (Krumbein, 1941). The sphericity is dependent on the roundness and the form of the grain since a true sphere is also perfectly round and compact.

Powers (1953) defined the degree of roundness for a grain as the function of the sharpness of the angles and edges. The roundness scale ranges from well-rounded to highly angular.

Figure 5.3 provides a schematic overview of the different ways to describe a grain, with different degrees of regularity, roundness and sphericity.

The novel correction depends entirely on the form of the grains to compute the volume. The form of the grain does play the largest role in the total volume of the grain. While the roundness, sphericity, and regularity of individual grains may not singularly influence the volume significantly, their combined effect cannot be overlooked. Ignoring this aspect could be problematic since the volume is the primary focus of this correction method.

5.4.2 Calibration technique

The main limitation of this method is that the shape of the grains can not be automatically determined from images of sediment. As a consequence of that unknown,

the calibration method is applied to approximate the grain volumes. The problem with the calibration technique is that the rules are arbitrary and based on empirical observations. For example, grains larger than 1 mm are often observed to be (fragmented) shells and are therefore assumed to be flat disks. However, this overlooks the influence of the sand grains that are larger than 1 mm.

Furthermore, the calibration correction tends to approximate an average grain shape. As visible in Figures 4.6 and 4.8, where the corrected pyDGS results lie within a relatively narrow range of values. Unfortunately, this correction method fails to correct for outliers with distinct shapes, which are the most interesting parts to study.

Lastly, the calibration technique is subject to an additional problem as it relies on the assumption that the differences between the pyDGS and sieve results are entirely related to volume, ignoring the influence of other factors like the presence of different lithologies or shell fragments.

5.4.3 Application to Egmond aan Zee

It was expected that the novel correction method works better with the well-sorted sands near Egmond aan Zee, because the grain shapes are expected to be more homogeneous. However, the results showed a large mean error for the Egmond data relative to the data of PHSD. This difference compared to the data of PHSD could be explained by the type of sand near Egmond aan Zee. For both sieve analyses the same mesh sizes for the sieves are used. However, these mesh sizes were selected to be used specifically for the PHSD poorly sorted sediment. For the well-sorted sands near Egmond aan Zee the finer details of the grain size distribution are less well captured and there are relatively few grains compared to the PHSD. This means that the calibration for the novel correction method is inadequate, because it misses information about the finer grain size distribution. Resulting in larger errors for these grain sizes.

5.5 Implications and recommendations
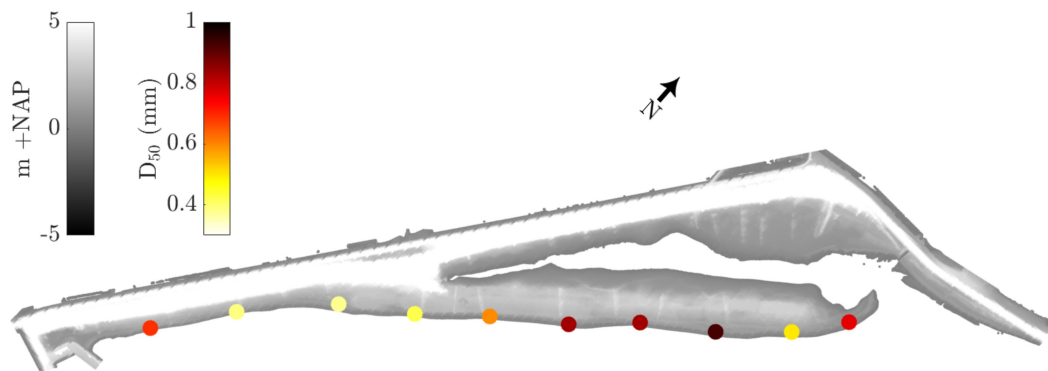
Figure 5.4: Elevation map of the PHSD with 10 D_{50} points, each computed as the cross-shore average. (*From: Bosma, 2023*)

The novel correction method has the ability to correct the sediment images using a non-general correction. This means that it can be better applied across extensive beach sections, whereas the general conversion method does not have this ability. For now, the new method is focused on the coarsest grain class, which results in the smallest errors in the coarse percentiles.

Figure 5.4 shows a map of the PHSD, where the current sieving methodology has been applied. It shows 10 alongshore points, where each point reflects the average of 4-5 cross-shore points. Compared to Figure 4.11 it is clear what the potential of the image-based analysis has. Instead of 10 lines, 24 lines can be sampled with approximately 14 samples per line being taken instead of 4-5. Also, the processing of the samples is easier, where the images can be processed automatically instead of the labour-intensive process of sieving. Moreover, this method is less time-consuming, allowing for more frequent data collection to study the temporal evolution of the coarse fraction.

Although the corrected output from pyDGS does not currently produce reliable absolute values, it can still be used to show relative differences in high-resolution lines. This means the method could be used to study the spatial resolution of sediment sampling, where each meter cross-shore (or 20 meters alongshore) can reveal patterns in the D_{90} values, including areas of coarser and finer fractions. While the absolute values may not be entirely accurate, these patterns could provide valuable data on the spatial variation of coarse fractions.

To make use of the full potential of the image-based grain size analysis, there are a few adjustments needed to apply to the current workflow as presented in this study.

1. The first option is to sample the surface, such as using an adhesive, and compare the sieve result to the output of pyDGS. This approach would remove the uncertainty in-depth and provide a more precise comparison. The next step would involve attempting to find a general relationship between the sample in-depth and on the surface. However, this could be challenging on a poorly sorted beach, where the variation of grain size in depth is likely to vary spatially.

2. The other option is to find a representative grain shape analysis method. A previous study by Lira and Pina (2009) presented an automated grain shape measurement technique involving photographing each grain size class after sieving. The output would include the previously discussed grain shape properties. If a representative grain shape distribution could be identified, the entire calibration method could be removed, and a different correction method could be applied to the difference between the sieve and pyDGS results. For example, that could be corrected for the difference in depth.

3. Matsumoto and Young (2023) showed the potential of using an ATV to improve the process of capturing photographs. This method can be extended to the PHSD as well, thereby enabling to study of a larger section of the beach.

4. Another approach to determine the grain size distribution is the use of deep-learning models, such as SediNet (Buscombe, 2020). These models continue to improve (McFall et al., 2023). However, it is important to note that the results from these models still need to be correct for comparison with sieved samples.

Chapter 6

Conclusion

The aim of this study was to find out if the method of optical granulometry can be an alternative to mechanical sieving which is a time-consuming process that limits the spatial- and temporal resolution. Therefore, the method of pyDGS was tested. Where the different correction methods have been explored and applied. To help to find the answer to the objective two main research questions were answered.

What is the most suitable conversion method for accurately determining grain size distribution on a poorly sorted sandy beach like the Prins Hendrik Sanddike?

The novel correction approach offers a significant advantage as it can correct the area-by-size measure to a volume-by-weight measure, allowing pyDGS to sufficiently accurately predict the grain size distribution of mixed sands. Unlike the traditional methods that rely on a correction parameter, this approach eliminates the need for generalisation and offers a more precise correction method for individual images. This makes it particularly useful for larger beach sections where the traditional correction method may not be effective. However, the correction process can be challenging due to the uncertainties related to grain shapes and the intrinsic variability of mixed sediments. Despite these challenges, the new method provides a more effective way of correcting the images for improved accuracy.

What are the potential applications of pyDGS?

Since the current correction methods do not yet provide sufficiently accurate estimations for the grain size compared to sieve results, pyDGS can not be used as a complete replacement. However, pyDGS does allow for much higher spatial resolution of the beach characteristics, allowing for monitoring at one-meter cross-shore intervals or even finer resolutions. This high resolution can provide insights in the relative patterns of the grain size evolution.

pyDGS offers an alternative approach by allowing for the fast and efficient monitoring of a larger area of the beach at high resolution. This makes it possible to study the temporal evolution of the coarse fraction with much greater detail. Additionally, the ability of pyDGS to identify relative patterns in grain size evolution that would be challenging to observe using traditional mechanical sieving. Consequently, it becomes a valuable tool for exploring the diverse factors influencing sediment transport, including wave action, tides, and storm events.

In summary, while pyDGS cannot yet entirely replace mechanical sieving, its high spatial resolution capabilities offer a valuable tool for studying the temporal evolution of grain size at a beach. The ability to identify relative patterns in grain size evolution allows researchers to investigate the factors that influence sediment transport, providing valuable new insights into beach dynamics.

To improve this method it should be extended to cover more beach sections and at a finer temporal scale. This could be done by using an ATV, which has the potential to substantially decrease the fieldwork duration.

Appendix A

Flowchart coin detection algorithm
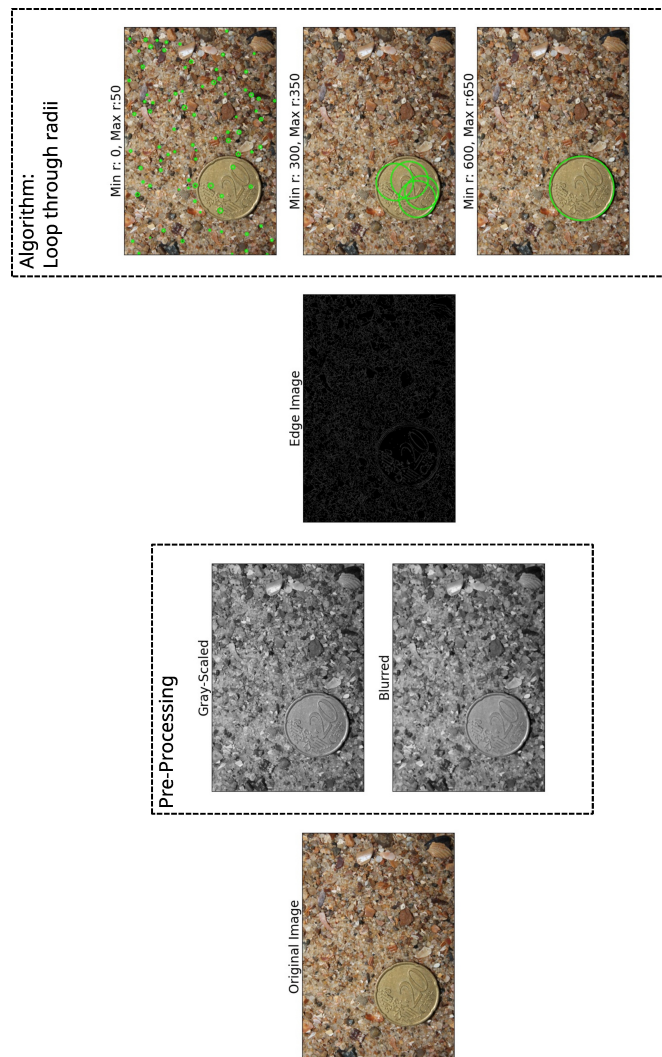
Appendix B

Run pyDGS script

Listing B.1: For current version: Github link

```python
import os
import sys
from functools import partial

import numpy as np
import pandas as pd
import pywt
from imageio.v2 import imread
from skimage.restoration import denoise_wavelet, estimate_sigma
from tqdm import tqdm

# rescale_sigma=True required to silence deprecation warnings
_denoise_wavelet = partial(denoise_wavelet, rescale_sigma=True)
import tkinter as tk
from datetime import datetime
from random import uniform
from time import sleep
from tkinter import messagebox

import cv2
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as stats
from scipy.stats import gmean
from tqdm import tqdm

start_time = datetime.now()
# =========================================================
def GetImageRes(img_path):
    filename = os.path.basename(img_path)
    dir_path = os.path.dirname(img_path)
    dir_name = os.path.basename(dir_path)

    try:
        DataFrame = pd.read_csv("/home/casper/Documents/Python/pyDGS_GUI/Output_data/Image_data/Bloemendaal/data_" + dir_name +".csv")
        row = DataFrame[DataFrame["Image_name"] == filename].index[0]
        resolution = DataFrame.at[row, 'Pixel_size_(mm/pixel)']
    except FileNotFoundError:
        DataFrame = pd.read_csv("/home/casper/Documents/Python/pyDGS_GUI/pyDGS-GUI/Output_data/Image_data/Bloemendaal/data_" + dir_name +".csv")
        row = DataFrame[DataFrame["Image_name"] == filename].index[0]
        resolution = DataFrame.at[row, 'Pixel_size_(mm/pixel)']
    return resolution

# =========================================================
def rescale(dat,mn,mx):
    """
    rescales an input dat between mn and mx
    """
    m = min(dat.flatten())
    M = max(dat.flatten())
    return (mx-mn)*(dat-m)/(M-m)+mn

# =========================================================
def standardize(img):
    img = np.array(img)
    #standardization using adjusted standard deviation
    N = np.shape(img)[0] * np.shape(img)[1]
    s = np.maximum(np.std(img), 1.0/np.sqrt(N))
    m = np.mean(img)
    img = (img - m) / s
    img = rescale(img, 0, 1)
    del m, s, N

    return img

# =========================================================
def Correction_Volume():

    pass

# =========================================================
def Percentage2Percentile(Percentage):
    Percentage = Percentage[::-1]
    Percentile = 100 - np.nancumsum(Percentage)
```

```python
        Percentile_cor = []
        for index, value in enumerate(Percentile):
            if value < 0:
                Percentile_cor.append(0)
            else:
                Percentile_cor.append(value)

        Percentile_cor = Percentile_cor[::-1]

        return Percentile_cor

# ============================================================
def Store_Percentile(path_of_the_directory, Image_Name, Percentile, Description):

        data = Image_Name, Percentile[0], Percentile[1], Percentile[2], Percentile[3], Percentile[4], Percentile[5], Percentile[6], Perc
        columns = ['Image_name', '0_mm', '0.063_mm', '0.125_mm', '0.180_mm', '0.250_mm', '0.300_mm', '0.355_mm', '0.425_mm', '0.500_mm',

        dir_path = os.path.dirname(path_of_the_directory)
        dir_name = os.path.basename(dir_path)

        temp = pd.DataFrame([data], columns=columns)
        temp.to_csv('Output_data/Bloemendaal/Percentiles/temp_percentile.csv', index=False)

        try:
            DF = pd.read_csv("Output_data/Bloemendaal/Percentile_" + dir_name + "_" + Description + ".csv")

            if Image_Name in DF.values:
                pass
            else:
                temp = pd.DataFrame([data], columns=columns)
                merged = pd.concat([temp, DF])
                merged.to_csv("Output_data/Bloemendaal/Percentile_" + dir_name + "_" + Description + ".csv", index=False)

        except FileNotFoundError:
            temp = pd.DataFrame([data], columns=[columns])
            temp.to_csv("Output_data/Bloemendaal/Percentile_" + dir_name + "_" + Description + ".csv", index=False)

# ============================================================
def Store_Percentage(path_of_the_directory, Image_Name, Percentage, Description):

        data = Image_Name, Percentage[0], Percentage[1], Percentage[2], Percentage[3], Percentage[4], Percentage[5], Percentage[6], Perc
        columns = ['Image_name', '0_mm', '0.063_mm', '0.125_mm', '0.180_mm', '0.250_mm', '0.300_mm', '0.355_mm', '0.425_mm', '0.500_mm',

        dir_path = os.path.dirname(path_of_the_directory)
        dir_name = os.path.basename(dir_path)

        temp = pd.DataFrame([data], columns=columns)
        temp.to_csv('Output_data/Percentage/temp_percentage.csv', index=False)

        try:
            DF = pd.read_csv("Output_data/Bloemendaal/Original_" + dir_name + "_" + Description + ".csv")

            if Image_Name in DF.values:
                pass
            else:
                temp = pd.DataFrame([data], columns=columns)
                merged = pd.concat([temp, DF])
                merged.to_csv("Output_data/Bloemendaal/Original_" + dir_name + "_" + Description + ".csv", index=False)

        except FileNotFoundError:
            temp = pd.DataFrame([data], columns=[columns])
            temp.to_csv("Output_data/Bloemendaal/Original_" + dir_name + "_" + Description + ".csv", index=False)

# ============================================================
input_dir = False

while input_dir == False:
    input_directory = input('Type_the_name_of_the_directory_(close_with_/):_')

    if input_directory.endswith("/"):
        input_dir = True
    else:
        print('Please_end_the_directory_name_with:_"/"_')
        input_dir = False

path_of_the_directory = os.path.join('/home/casper/Documents/Aardwetenschappen/MSc_Thesis/Photo/Bloemendaal/', input_directory)
dir_path = os.path.dirname(path_of_the_directory)
dir_name = os.path.basename(dir_path)
print('The_working_directory_will_be:_', path_of_the_directory)
print('_____')
answer = False

while answer == False:
    save_Percentages = input('Do_you_want_to_save_the_uncorrected_percentages?_(y/n)_')

    if save_Percentages == "y":
        print("Uncorrected_percentages_will_be_stored")
        answer = True
    elif save_Percentages == "n":
        print("Uncorrected_percentages_will_not_be_stored")
        answer = True
    else:
        print("please_answer_with_y_or_n")
print('_____')
Correction = False

while Correction == False:
```

```python
    save_Correction = input('Do_you_want_to_use_the_multi-trend_correction?_(y/n)_')

    if save_Correction == "y":
        print("Correction_will_be_preformed")
        Correction = True
    elif save_Correction == "n":
        print("No_correction_is_done")
        Correction = True
    else:
        print("please_answer_with_y_or_n")

print('_____')
Description_Data = input('Give_a_description_for_stored_data:_')
print("The_data_will_be_stored_as:_/..._" + dir_name + "_" + Description_Data + ".csv")
print('_____')


for files in os.listdir(path_of_the_directory):
    if files.endswith('.jpg') or files.endswith('.JPG'):
        image = path_of_the_directory + files
        resolution = GetImageRes(image)
        print(files, resolution)

        img = cv2.imread(image)
        nxx, nyy, _ = img.shape
        width = max(nxx, nyy)

        im = imread(image)     # read the image straight with imread
        im = np.squeeze(im)    # squeeze singleton dimensions
        if len(np.shape(im))>3:
            im = im[:, :, :3]              # only keep the first 3 bands

        if len(np.shape(im))==3: # if rgb, convert to grey
            im = (0.299 * im[:,:,0] + 0.5870*im[:,:,1] + 0.114*im[:,:,2]).astype('uint8')

        nx,ny = np.shape(im)
        if nx>ny:
            im=im.T

        im = standardize(im)

        region = im.copy()

        original = rescale(region,0,255)

        nx, ny = original.shape
        P = []; M = []
        for k in tqdm(np.linspace(1,nx-1,100)):
            [cfs, frequencies] = pywt.cwt(original[int(k),:], np.arange(5, np.maximum(nx,ny)/(width*resolution / 8), 1),
'morl', .5)
            period = 1. / frequencies
            power =(abs(cfs)) ** 2
            power = np.mean(np.abs(power), axis=1)/(period**2)
            P.append(power)

            M.append(period[np.argmax(power)])
            sleep(uniform(0.005, 0.01))
        p = np.mean(np.vstack(P), axis=0)
        p = np.array(p/np.sum(p))

        # get real scales by multiplying by resolution (mm/pixel)
        scales = np.array(period)

        srt = np.sqrt(np.sum(p*((scales-np.mean(M))**2)))

        p = p+stats.norm.pdf(scales, np.mean(M), srt/2)
        p = np.hstack([p])
        scales = np.hstack([scales])
        p = p/np.sum(p)
        x = -0.47
        # area-by-number to volume-by-number
        r_v = (p*scales**x) / np.sum(p*scales**x) #volume-by-weight proportion

        # Calculating Percentages
        a = (scales*resolution)
        minSz = np.array([0, 0.063, 0.125, 0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8])
        maxSz = np.array([0.063, 0.125, 0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8, 12])
        percentage = []
        if a[0] > 0.063:
            percentage.append(0)
            minSz = np.array([0.125, 0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8])
            maxSz = np.array([0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8, 12])

            _, length =  np.shape(np.where(a<0.125))
            percentage.append(((np.trapz(np.interp([np.linspace(a[0], 0.125, 1000)], (scales*resolution), r_v)[0])*length/1000))*100)
            for i in range(len(minSz)):
                _, length = np.shape(np.where((a>minSz[i])&(a<maxSz[i])))
                percentage.append(((np.trapz(np.interp([np.linspace(minSz[i], maxSz[i], 1000)], (scales*resolution), r_v)[0])*length/1000))*100)
        else:
            for i in range(len(minSz)):
                _, length = np.shape(np.where((a>minSz[i])&(a<maxSz[i])))
                percentage.append(((np.trapz(np.interp([np.linspace(minSz[i], maxSz[i], 1000)], (scales*resolution), r_v)[0])*length/1000))*100)

        if save_Percentages == "y":
            Store_Percentage(path_of_the_directory, files, percentage, Description_Data)

        for index, value in enumerate(minSz):
            if (value/resolution) < 3:
```

```
                    percentage[index] = np.nan
                else:
                    pass


        if save_Correction == "y":
            minSz = np.array([0, 0.063, 0.125, 0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8])
            # Multi-trend correction
            C_s = 191.13808985413237
            P_s = 2.804953092790458
            C_m = 0.1854802814657743
            P_m = -2.1493016950684303
            C_l = 0.3530984202702832
            P_l = 1.887249670551427

            first_step = []

            for index, value in enumerate(percentage):
                if value != 0:
                    if minSz[index] <=0.250:
                        first_step.append(value*(C_s * pow(minSz[index], P_s)))
                    elif minSz[index] > 0.250 and   minSz[index] < 1:
                        first_step.append(value*(C_m * pow(minSz[index], P_m)))
                    elif minSz[index] >= 1:
                        first_step.append(value*(C_l * pow(minSz[index], P_l)))
                else:
                    first_step.append(np.nan)

            total_sum = np.nansum(first_step)
            Corrected_Percentage = []
            for index, value in enumerate(first_step):
                Corrected_Percentage.append(((value/total_sum)*100))

            Percentiles = Percentage2Percentile(Corrected_Percentage)

            Store_Percentile(path_of_the_directory, files, Percentiles, Description_Data)

    else:
        continue


end_time = datetime.now()
print('Duration:_{}'.format(end_time - start_time))
```

Appendix C

Volume Correction Code

Listing C.1: For current version: Github link

```python
for i in range(len(Resolution)):
    Nb_Grains = []
    Image_Area = (4000*Resolution[i]) * (2250 * Resolution[i])

    DGS_Fractions = dgs_data[i, 1:] /100

    Area_Fraction = (DGS_Fractions * Image_Area)

    Grain_Volume = []
    for j in range(len(Area_Fraction)):
        if Area_Fraction[j] == 0:
            Nb_Grains.append(0)
        else:
            if STdev[i, j+1] < 0.5:
                Nb_Grains.append((Area_Fraction[j]) / (np.pi*(GrainSz[j]/2)**2))
            elif STdev[i, j+1] >= 0.5 and STdev[i, j+1] < 1:
                if GrainSz[j] < 1:
                    Nb_Grains.append((Area_Fraction[j]) / (np.pi*(0.75 *(GrainSz[j]/2) * (GrainSz[j]/2))))
                else:
                    Nb_Grains.append((Area_Fraction[j]) / ((1/STdev[i, j+1]* (GrainSz[j]/2))* 0.1 * np.pi*(GrainSz[j]/2)))
            else:
                Nb_Grains.append((Area_Fraction[j]) / ((STdev[i, j+1]* (GrainSz[j]/2) * 0.001) * np.pi*(GrainSz[j]/2)))


        #     Volume part
        if STdev[i, j+1] < 0.5:
            if GrainSz[j] == 0.71:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])**2 * 0.001*GrainSz[j])
            elif GrainSz[j] == 0.5:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])**2 * 0.05*GrainSz[j])
            else:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])**2 * 0.1*GrainSz[j])

        elif STdev[i, j+1] >= 0.5 and STdev[i, j+1] < 1:
            if GrainSz[j] == 0.71:
                Grain_Volume.append((np.pi/6)*(GrainSz[j]*GrainSz[j]*0.75*GrainSz[j])*0.1)
            elif GrainSz[j] == 0.5:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*STdev[i, j+1]*GrainSz[j]*0.1)
            elif GrainSz[j] == 2:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*GrainSz[j]*0.0005)
            elif GrainSz[j] == 8:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*GrainSz[j]*0.00075)
            else:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*STdev[i, j+1]*GrainSz[j]*0.01)

        else:
            if GrainSz[j] == 0.5:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j]*0.000001)
            elif GrainSz[j]<1:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j])
            elif GrainSz[j] == 2:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*GrainSz[j]*0.000001)
            elif GrainSz[j] == 1:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j]*0.00025)
            elif GrainSz[j] == 0.710:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j])
            elif GrainSz[j] == 4:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j]*0.000001)
            elif GrainSz[j] == 8:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j]*0.000001)
            else:
                Grain_Volume.append((np.pi/6)*(GrainSz[j])*GrainSz[j]*1/STdev[i, j+1]*GrainSz[j])

    Grain_Mass = np.array(Grain_Volume) * Density_Sand
    for index, value in enumerate(Grain_Mass):
        Mass_Fraction[i,index] = (Nb_Grains[index]*value)


    Mass_Fraction[i,:] = (Mass_Fraction[i,:] / np.nansum(Mass_Fraction[i,:])) * 100

df = pd.DataFrame(Mass_Fraction)
# df.to_csv('/home/casper/Documents/Python/pyDGS-GUI/Output data/26_10_22/Mobile/Corrected/Corrected_R10Inb.csv')
```

```
plt.subplot(2,1,1)
sieve_open = [0.063, 0.125, 0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8]
for i in range(len(sieve_data)):
    plt.scatter(sieve_open, ( sieve_data[i,2:] - Mass_Fraction[i,1:] ), color=color[i], label= sieve_data[i,0])
plt.legend(bbox_to_anchor=(1,1), loc="upper_left")
plt.xscale("log")
plt.hlines(0, 0.063, 8, color='k')
plt.ylabel('%-Sieve_-_%-pyDGS_', fontsize=20)


error = sieve_data[:34, 2:]- Mass_Fraction[:34, 1:]
std = np.std(error.astype(np.float64), axis=0)

sieve_open = [0.063, 0.125, 0.180, 0.250, 0.300, 0.355, 0.425, 0.500, 0.710, 1, 2, 4, 8]
plt.subplot(2,1,2)

plt.errorbar(x=sieve_open, y=np.mean(error, axis=0), yerr=std, capsize=3)
plt.hlines(0, 0.063, 8, color='k')
plt.xscale('log')
plt.ylabel('Mean_Absolute_Error', fontsize=20)
plt.xlabel('Grain_size_(mm)', fontsize=20)

plt.show()
```

Appendix D

Error statistics computation

The Root Mean Square Error (RMSE) is a measure of the average error between predicted and actual values, with lower values indicating better model performance. It is defined as:

$$RSME = \frac{\sqrt{\sum \left[(Measured - Estimated)^2 \right]}}{n} \qquad \text{(D.1)}$$

The Mean Absolute Error (MAE) is a metric used to assess the average absolute difference between predicted and actual values, with lower values indicating better model performance, which is defined as:

$$MAE = \frac{1}{n} \sum \left(|Measured - Estimated| \right) \qquad \text{(D.2)}$$

To compare the results of the RMSE and MAE between different methods it is useful to normalise the values, which means that Equations D.1 and D.2 are rewritten as:

$$NRSME = \frac{\frac{\sqrt{\sum [(Measured - Estimated)^2]}}{n}}{Measured_{max} - Measured_{min}} * 100\% \qquad \text{(D.3)}$$

$$NMAE = \frac{\frac{1}{n} \sum \left(|Measured - Estimated| \right)}{Measured_{max} - Measured_{min}} * 100\% \qquad \text{(D.4)}$$

Bibliography

Adams, J. (1979). Gravel Size Analysis from Photographs. *Journal of the Hydraulics Division*, *105*(10), 1247–1255. https://doi.org/10.1061/JYCEAJ.0005283

Blott, S. J., & Pye, K. (2008). Particle shape: A review and new methods of characterization and classification. *Sedimentology*, *55*(1), 31–63. https://doi.org/10.1111/j.1365-3091.2007.00892.x

Bosma, J. (2023). Effective Upgrades and Retrofits for Coastal Climate Adaptation. *[Unpublished PhD thesis]*, Utecht University.

Bruinzeel, L. W., & Snoek, R. (2018). *Addendum Passende Beoordeling zandwinning en -transport PHZD Texel (Q2N en L15)* (tech. rep. No: A&W-rapport 2432). Altenburg & Wymenga. https://rb.gy/b2os4

Bunte, K., & Abt, S. (2001). Sampling Surface and Subsurface Particle-size Distributions in Wadable Gravel- and Cobble-bed Streams for Analyses in Sediment Transport, Hydraulics, and Streambed Monitoring. *US Department of Agriculture, Forest Service, Rocky Mountain Research Station.* https://doi.org/10.2737/rmrs-gtr-74

Buscombe, D., Rubin, D. M., & Warrick, J. A. (2010). A universal approximation of grain size from images of noncohesive sediment. *Journal of Geophysical Research: Earth Surface*, *115*(F2). https://doi.org/10.1029/2009JF001477

Buscombe, D. (2013). Transferable wavelet method for grain-size distribution from images of sediment surfaces and thin sections, and other natural granular patterns. *Sedimentology*, *60*(7), 1709–1732. https://doi.org/10.1111/sed.12049

Buscombe, D. (2020). SediNet: A configurable deep learning model for mixed qualitative and quantitative optical granulometry. *Earth Surface Processes and Landforms*, *45*(3), 638–651. https://doi.org/10.1002/esp.4760

Carbonneau, P. E., Lane, S. N., & Bergeron, N. E. (2004). Catchment-scale mapping of surface grain size in gravel bed rivers using airborne digital imagery. *Water Resources Research*, *40*(7). https://doi.org/10.1029/2003WR002759

Chang, F.-J., & Chung, C.-H. (2012). Estimation of riverbed grain-size distribution using image-processing techniques. *Journal of Hydrology*, *440-441*, 102–112. https://doi.org/10.1016/j.jhydrol.2012.03.032

Cuttler, M. V. W., Lowe, R. J., Falter, J. L., & Buscombe, D. (2017). Estimating the settling velocity of bioclastic sediment using common grain-size analysis techniques. *Sedimentology*, *64*(4), 987–1004. https://doi.org/10.1111/sed.12338

Deronde, B., Houthuys, R., Henriet, J.-P., & Lancker, V. V. (2008). Monitoring of the sediment dynamics along a sandy shoreline by means of airborne hyperspectral remote sensing and LIDAR: A case study in Belgium. *Earth Surface Processes and Landforms: The Journal of the British Geomorphological Research Group*, *33*(2), 280–294. https://doi.org/10.1002/esp.1545

Diplas, P., & Fripp, J. B. (1992). Properties of Various Sediment Sampling Procedures. *Journal of Hydraulic Engineering*, *118*(7), 955–970. https://doi.org/10.1061/(ASCE)0733-9429(1992)118:7(955)

Diplas, P., & Sutherland, A. J. (1988). Sampling Techniques for Gravel Sized Sediments. *Journal of Hydraulic Engineering*, *114*(5), 484–501. https://doi.org/10.1061/(ASCE)0733-9429(1988)114:5(484)

Díaz Gómez, R., Pasternack, G. B., Guillon, H., Byrne, C. F., Schwindt, S., Larrieu, K. G., & Solis, S. S. (2022). Mapping subaerial sand-gravel-cobble fluvial sediment facies using airborne lidar and machine learning. *Geomorphology*, *401*, 108106. https://doi.org/10.1016/j.geomorph.2021.108106

Eisma, D. (1968). Composition, origin and distribution of Dutch coastal sands between Hoek Van Holland and the island of Vlieland. *Netherlands Journal of Sea Research*, *4*(2), 123–267. https://doi.org/10.1016/0077-7579(68)90011-2

Ferro, V., & Mirabile, S. (2009). Comparing particle size distribution analysis by sedimentation and laser diffraction method. *Journal of Agricultural Engineering*, *40*(2), 35–43. https://doi.org/10.4081/jae.2009.2.35

Folk, R. L. (1974). Petrology of sedimentary rocks. *Hemphill Publishing Company., Austin (Texas)*. https://rb.gy/vg5ls

Folk, R. L., & Ward, W. C. (1957). Brazos River bar [Texas]; a study in the significance of grain size parameters. *Journal of Sedimentary Research*, *27*(1), 3–26. https://doi.org/10.1306/74D70646-2B21-11D7-8648000102C1865D

Fordeyn, J, Lemey, E, & Perk, L. (2019). A holistic approach to coastal protection for the Prins Hendrik polder. *22nd World Dredging Congress, WODCON 2019*, *pp*, 834–846. https://rb.gy/8044i

Geldof, H. J., & Slot, R. E. (1979). *Settling Tube Analysis of Sand* (tech. rep. No: 4-79). TU Delft. https://rb.gy/i6ou5

Gessler, J. (1968). The Beginning of Bedload Movement of Mixtures Investigated as Natural Armoring in Channels. *PhD thesis, Swiss Federal Institute of Technology*. https://resolver.caltech.edu/CaltechAUTHORS:20200624-114805343

Graham, D. J., Rollet, A.-J., Rice, S. P., & Piégay, H. (2012). Conversions of Surface Grain-Size Samples Collected and Recorded Using Different Procedures. *Journal of Hydraulic Engineering*, *138*(10), 839–849. https://doi.org/10.1061/(ASCE)HY.1943-7900.0000595

Harvey, E. L., Hales, T. C., Hobley, D. E. J., Liu, J., & Fan, X. (2022). Measuring the grain-size distributions of mass movement deposits. *Earth Surface Processes and Landforms*, *47*(6), 1599–1614. https://doi.org/10.1002/esp.5337

Hey, R. D. (1979). Flow Resistance in Gravel-Bed Rivers. *Journal of the Hydraulics Division*, *105*(4), 365–379. https://doi.org/10.1061/JYCEAJ.0005178

HHNK & Witteveen+Bos. (2017). *Prins Hendrikzanddijk - Projectplan*, (tech. rep. No: 17.29511). Hoogheemraadschap Hollands Noorderkwartier. https://rb.gy/y68qf

Horn, D. P., & Walton, S. M. (2007). Spatial and temporal variations of sediment size on a mixed sand and gravel beach. *Sedimentary Geology*, *202*(3), 509–528. https://doi.org/10.1016/j.sedgeo.2007.03.023

Ibbeken, H., & Schleyer, R. (1986). Photo-sieving: A method for grain-size analysis of coarse-grained, unconsolidated bedding surfaces. *Earth Surface Processes and Landforms*, *11*(1), 59–77. https://doi.org/10.1002/esp.3290110108

Inman, D. L. (1952). Measures for describing the size distribution of sediments. *Journal of Sedimentary Research*, *22*(3), 125–145. https://doi.org/10.1306/D42694DB-2B26-11D7-8648000102C1865D

Kamphuis, J. W. (1974). Determination of sand roughness for fixed beds. *Journal of Hydraulic Research*, *12*(2), 193–203. https://doi.org/10.1080/00221687409499737

Kellerhals, R., & Bray, D. I. (1971). Sampling Procedures for Coarse Fluvial Sediments. *Journal of the Hydraulics Division*, *97*(8), 1165–1180. https://doi.org/10.1061/JYCEAJ.0003044

Klein Obbink, M. (2022). Multi-fraction sediment sorting and entrainment at the Prins Hendrikzanddijk. *MSc Thesis, Utrecht University*. https://studenttheses.uu.nl/handle/20.500.12932/41609

Komar, P. D., & Cui, B. (1984). The analysis of grain-size measurements by sieving and settling-tube techniques. *Journal of Sedimentary Research*, *54*(2), 603–614. https://doi.org/10.1306/212F8481-2B24-11D7-8648000102C1865D

Krumbein, W. C. (1941). Measurement and geological significance of shape and roundness of sedimentary particles. *Journal of Sedimentary Research*, *11*(2), 64–72. https://doi.org/10.1306/D42690F3-2B26-11D7-8648000102C1865D

Lang, N., Irniger, A., Rozniak, A., Hunziker, R., Wegner, J. D., & Schindler, K. (2021). GRAINet: Mapping grain size distributions in river beds from UAV images with convolutional neural networks. *Hydrology and Earth System Sciences*, *25*(5), 2567–2597. https://doi.org/10.5194/hess-25-2567-2021

Langhammer, J., Lendzioch, T., Miřijovský, J., & Hartvich, F. (2017). UAV-based optical granulometry as tool for detecting changes in structure of flood depositions. *Remote Sensing*, *9(3)*, 240. https://doi.org/10.3390/rs9030240

Lira, C., & Pina, P. (2009). Automated grain shape measurments applied to beach sands. *Journal of Coastal Research*, 1527–1531. https://www.jstor.org/stable/25738045

Little, W. C., & Meyer, P. G. (1976). Stability of Channel Beds by Armoring. *Journal of the Hydraulics Division*, *102*(11), 1647–1661. https://doi.org/10.1061/JYCEAJ.0004651

Longiaru, S. (1987). Visual Comparators for Estimating the Degree of Sorting from Plane and Thin Section. *Journal of Sedimentary Research*, *57*(4). https://doi.org/10.1306/212f8c60-2b24-11d7-8648000102c1865d

Mair, D., Do Prado, A. H., Garefalakis, P., Lechmann, A., Whittaker, A., & Schlunegger, F. (2022). Grain size of fluvial gravel bars from close-range UAV imagery–uncertainty in segmentation-based data. *Earth Surface Dynamics*, *10*(5), 953–973. https://doi.org/10.5194/esurf-10-953-2022

Matsumoto, H., & Young, A. P. (2018). Automated Cobble Mapping of a Mixed Sand-Cobble Beach Using a Mobile LiDAR System. *Remote Sensing*, *10*(8), 1253. https://doi.org/10.3390/rs10081253

Matsumoto, H., & Young, A. P. (2023). Development of an automated mobile grain size mapping of a mixed sediment beach. *Earth Surface Processes and Landforms*. https://doi.org/10.1002/esp.5655

McCammon, R. (1962). Efficiencies of Percentile Measures for Describing the Mean Size and Sorting of Sedimentary Particles. *The Journal of Geology*, *70*((4)), 453–465. https://doi.org/10.1086/626836

McFall, B. C., Young, D. L., Whitmeyer, S. J., Buscombe, D., Stever, S. N., & Walker, B. M. (2023). Sandsnap: Creating a nationwide beach grain size database by engaging citizen scientists. *Coastal Sediments 2023: The Proceedings of the Coastal Sediments 2023*, 906–918. https://doi.org/10.1142/9789811275135_0086

Nugroho, S. H., & Putra, P. S. (2018). Spatial distribution of grain size and depositional process in tidal area along Waikelo Beach, Sumba. *Marine Georesources & Geotechnology*, *36*(3), 299–307. https://doi.org/10.1080/1064119X.2017.1312649

Otto, G. (1939). A modified logarithmic probability graph for the interpretation of mechanical analyses of sediments. *Journal of Sedimentary Research, 9*((2)), 62–76. https://doi.org/10.1306/d4269044-2b26-11d7-8648000102c1865d

Papalazarou, A, Albanakis, K, & Laopoulos, T. (2010). A Low Cost Implementation of the Settling-Tube Measurement Technique for Sand Granulometry. *17th Symposium IMEKO TC 4, 3rd Symposium IMEKO TC 19 and 15th IWADC Workshop*. https://rb.gy/9kcyr

ParticleTechLabs, P. (2023a). Laser Diffraction. https://particletechlabs.com/analytical-testing/laser-diffraction/

ParticleTechLabs, P. (2023b). Sieve Analysis. https://particletechlabs.com/analytical-testing/sieve-analysis/

Powers, M. (1953). A New Roundness Scale for Sedimentary Particles. *SEPM Journal of Sedimentary Research, Vol. 23*. https://doi.org/10.1306/D4269567-2B26-11D7-8648000102C1865D

Prodger, S., Russell, P., Davidson, M., Miles, J., & Scott, T. (2016). Understanding and predicting the temporal variability of sediment grain size characteristics on high-energy beaches. *Marine Geology, 376*, 109–117. https://doi.org/10.1016/j.margeo.2016.04.003

Proffitt & Sutherland. (1980). Self Armouring of Non-uniform Alluvial Sediments. *In: 7th Australasian Conference on Hydraulics and Fluid Mechanics (7th: 1980: Brisbane, Qld.)*, 225–228. https://doi.org/10.3316/informit.561373565280215

Proffitt, G. T. G. T. (1980). Selective transport and armouring of non-uniform alluvial sediments. *PhD thesis, University of Canterbury*. https://doi.org/10.26021/11090

Rapportendatabank, R. (1990). *Kustverdediging na 1990: Beleidskeuze voor de kustlijnzorg.* (Kamerstuk 21.136 No: 5-6). Rijkswaterstaat, RIKZ. https://rb.gy/m3fc5

Román-Sierra, J., Muñoz-perez, J. j., & Navarro-Pons, M. (2013). Influence of sieving time on the efficiency and accuracy of grain-size analysis of beach and dune sands. *Sedimentology, 60*(6), 1484–1497. https://doi.org/10.1111/sed.12040

Rubin, D. M., & Chezar, H. (2007). Instant Grainification: Real-Time Grain-Size Analysis from Digital Images in the Field. *AGU Fall Meeting Abstracts, 2007*, H51I–0899. https://rb.gy/btnmf

Rubin, D. M. (2004). A Simple Autocorrelation Algorithm for Determining Grain Size from Digital Images of Sediment. *Journal of Sedimentary Research, 74*(1), 160–165. https://doi.org/10.1306/052203740160

Ruessink, G., Schwarz, C. S., Price, T. D., & Donker, J. J. A. (2019). A Multi-Year Data Set of Beach-Foredune Topography and Environmental Forcing Conditions at Egmond aan Zee, The Netherlands. *Data, 4*(2), 73. https://doi.org/10.3390/data4020073

Sime, L., & Ferguson, R. (2003). Information on Grain Sizes in Gravel-Bed Rivers by Automated Image Analysis. *Journal of Sedimentary Research, 73*(4), 630–636. https://doi.org/10.1306/112102730630

Sneed, E. D., & Folk, R. L. (1958). Pebbles in the Lower Colorado River, Texas a Study in Particle Morphogenesis. *The Journal of Geology, 66*(2), 114–150. https://doi.org/10.1086/626490

Sperazza, M., Moore, J. N., & Hendrix, M. S. (2004). High-Resolution Particle Size Analysis of Naturally Occurring Very Fine-Grained Sediment Through Laser Diffractometry. *Journal of Sedimentary Research, 74*(5), 736–743. https://doi.org/10.1306/031104740736

Steetzel, H., Penning, E., van Santen, R., & Vuik, V. (2017). Building with nature pilot sandy foreshore Houtribdijk design and behaviour if a sandy dike defence in a lake system. *Coastal Dynamics: Port of Spain, Trinidad and Tobago*, (063). https://rb.gy/2al9l

Stive, M. J., de Schipper, M. A., Luijendijk, A. P., Aarninkhof, S. G., van Gelder-Maas, C., van Thiel de Vries, J. S., de Vries, S., Henriquez, M., Marx, S., & Ranasinghe, R. (2013). A New Alternative to Saving Our Beaches from Sea-Level Rise: The Sand Engine. *Journal of Coastal Research*, *29*(5), 1001–1008. https://doi.org/10.2112/JCOASTRES-D-13-00070.1

Strypsteen, G., van Rijn, L. C., Hoogland, M. D., Rauwoens, P., Fordeyn, J., Hijma, M. P., & Lodder, Q. J. (2021). Reducing aeolian sand transport and beach erosion by using armour layer of coarse materials. *Coastal Engineering*, *166*, 103871. https://doi.org/10.1016/j.coastaleng.2021.103871

Syvitski, J. P. (1991). Principles, methods, and application of particle size analysis. *Cambridge: University Press Cambridge*, *388*. http://www.grano.de/basinr01.rtf

Trask, P. (1930). Mechanical analyses of sediments by centrifuge. *Economic Geology*, *25*(581-599). https://rb.gy/hv637

USACE. (2008). Coastal sediment properties. *Coastal Engineering Manual − Part III*, *US Army Corps of Engineers*(EM 1110-2-1100). https://rb.gy/xhel7

van Rijn, L. Chapter: 7.41–7.43: Stochastic predictor. In: In *Principles of sediment transport in rivers, estu aries and coastal seas.* Oldemarkt, Aqua, The Netherlands, 1993. https://rb.gy/j6jnb

van der Wal, D. (1998). The Impact of the Grain-Size Distribution of Nourishment Sand on Aeolian Sand Transport. *Journal of Coastal Research*, *14*(2), 620–631. https://www.jstor.org/stable/4298816

van der Wal, D. (2000). Grain-Size-Selective Aeolian Sand Transport on a Nourished Beach. *Journal of Coastal Research*, *16*(3), 896–908. https://www.jstor.org/stable/4300100

Warrick, J. A., Rubin, D. M., Ruggiero, P., Harney, J. N., Draut, A. E., & Buscombe, D. (2009). Cobble cam: Grain-size measurements of sand to boulder from digital photographs and autocorrelation analyses. *Earth Surface Processes and Landforms*, *34*(13), 1811–1821. https://doi.org/10.1002/esp.1877

Wittebrood, M., Vries, S. d., Goessen, P., & Aarninkhof, S. (2018). Aeolian sediment transport at a man-made dune system; building with nature at the Hondsbossche dunes. *Coastal Engineering Proceedings*, (36), 83–83. https://doi.org/10.9753/icce.v36.papers.83

Yoshioka, A., Uda, T., Aoshima, G., Furuike, K., & Ishikawa, T. (2009). Field experiment of beach nourishment considering change in grain size and prediction of beach changes. *Coastal Engineering 2008*, 2694–2706. https://rb.gy/bbaaf