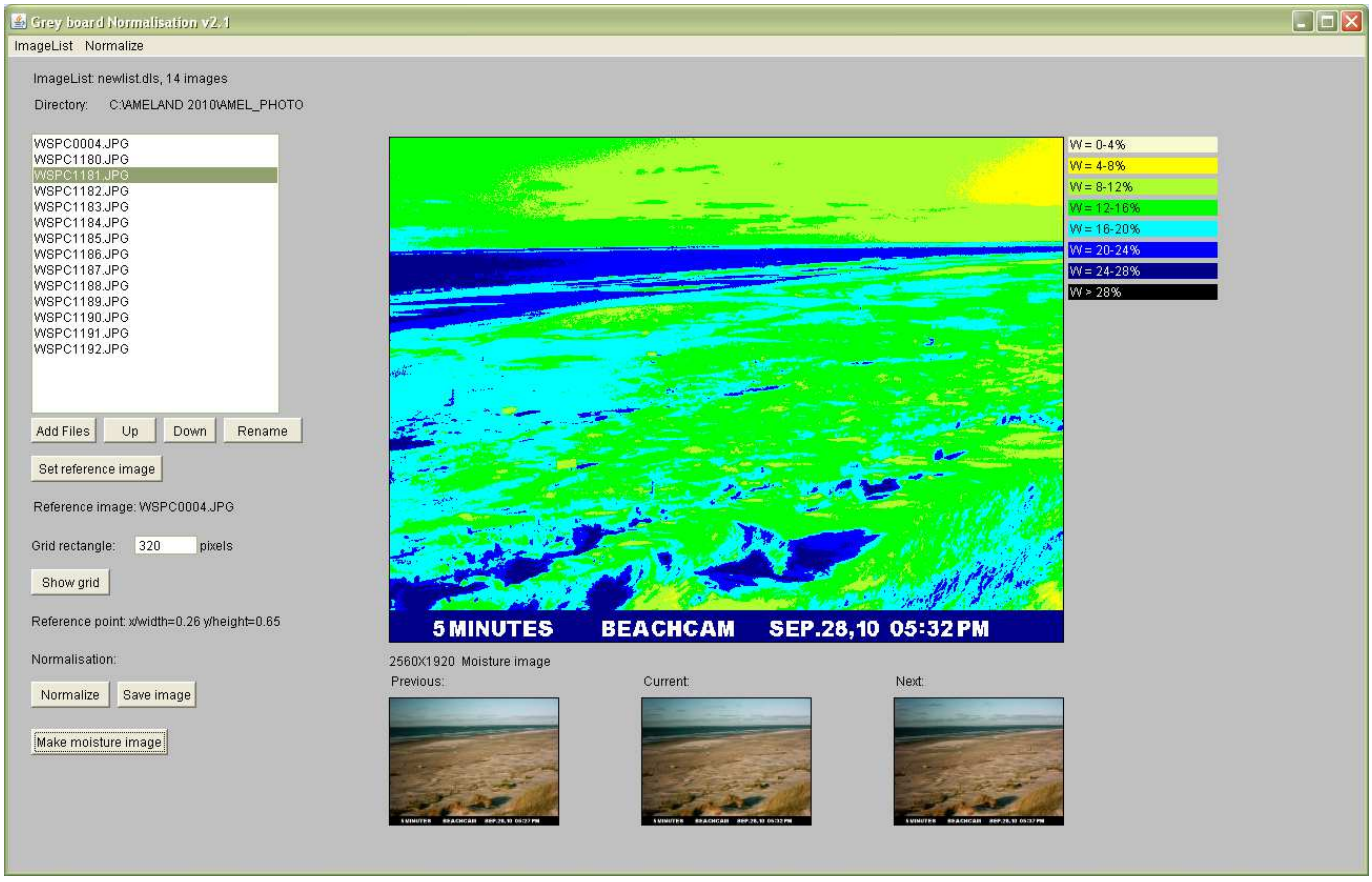
Land Degradation & Development Group

Beach surface moisture mapping as a spatiotemporal parameter for aeolian sand transport prediction



WAGENINGEN UNIVERSITY
WAGENINGEN UR

Beach surface moisture mapping as a spatiotemporal parameter for aeolian sand transport prediction

Corjan Nolet

Master thesis Land Degradation and Development Group submitted in partial fulfillment of the degree of Master of Science in International Land and Water Management at Wageningen University, the Netherlands

Study Program

MSc International Land and Water Management (MIL)

Student registration number

801226-606-050

MSc Thesis Land Degradation and Development

LDD-80339

Supervisors

Dr. Saskia Visser (Wageningen University, the Netherlands)

MSc . Ate Poortinga (Wageningen University, the Netherlands)

Examinators

Dr. Saskia Visser (Wageningen University, the Netherlands)

Dr. Michel Riksen (Wageningen University, the Netherlands)

Abstract

The objective of this study is to develop temporal surface moisture mapping on a sandy beach in a temperate climate, as input parameter for a spatiotemporal model for predicting aeolian sand transport rate and available sand budget for foredune development. It provides parameter input, at an improved spatial and temporal resolution, for the control of surface moisture content on the threshold shear velocity (u_{*t}) at which sand begins to move. The research goals are: (1) to establish a regression model between beach surface moisture content and normalized beach surface reflectance and analyze its goodness of fit, and (2) to develop a data processing method using Java that allows for automated temporal beach surface moisture mapping.

Beach surface reflectance measurements are taken at wavelengths in the visible part of the electromagnetic spectrum. The OLS simple linear regression model $\hat{y} = b_0 + b_1 x + e_i$ for the outlier corrected n pair observations (x_n, y_n) of gravimetric moisture content *w* (%) and normalized luminance *Y* (0 – 255) is described by $\hat{y} = 100.557 - 1.533x$. The regression model is significant at level $\alpha = 0.05$ with a (relatively) strong r of 0.565, but violates the assumptions of normality and constant variance of the errors (skewness 0.548, kurtosis 0.396). As a result the predictive quality of the model is questionable. Still, this relationship is used as input for temporal beach surface moisture mapping.

The data processing method that allows for automated beach surface moisture mapping is effective. There are a number of qualitative concerns about the output. This is due to shortcomings in the field measurement setup, and the device and illumination invariant normalization procedure. Beach surface reflectance measurements at wavelengths in the short-wave infrared electromagnetic spectrum improves the strength of the (then) exponential regression model between gravimetric surface moisture content *w* (%) and normalized luminance *Y* (0 – 255). A more correct field measurement protocol and extending the data processing method for further automation, improves the effectiveness of surface moisture mapping as a spatiotemporal parameter for aeolian sand transport prediction.

Table of Contents

Introduction

In many parts of the world coastal dunes are of vital importance with respect to coastal defense. The ecological significance of coastal dunes is considerable, and in The Netherlands they serve as storage and filtering media for drinking water. A thorough understanding of aeolian processes in the coastal environment is essential for the prediction and evaluation of engineering and management activities (Arens 1996).

This study is part of the PhD-project Knowledge for Climate that aims to understand, simulate, and predict (i.e. model) the impact of climate change on aeolian sand transport rate and available sand budget for foredune development. An essential model component is a function to quantify a sand budget (or rate of sand transport) from the beach to the foredunes. Sand transport by wind is the primary source of sand supply for building foredunes on sandy beaches (Sherman and Bauer 1993; Aagaard et al 2004). Aeolian sand transport and supply rate is governed by spatial and temporal variations in both the erosive force of the wind (erosivity) and the resistance of the surface to erosion (erodibility).

Surface moisture content is an important control on aeolian sand transport on a beach (Cornelis and Gabriels 2003; Davidson-Arnott et. al 2005; Edwards and Namikas 2009). It increases the threshold shear velocity (u_{*t}) at which sand begins to move (Chepil 1956). Gravimetric moisture contents of approximately 0.6% can more than double the u_{*t} of medium-sized sands. Above approximately 5% moisture content, sand-sized material is inherently resistant to entrainment by most natural winds (Belly 1964). Unfortunately, the control of surface moisture content on aeolian sand transport and supply rate is difficult to parameterize. This is because of its high degree of spatiotemporal variability, and its operation at the scale of a particle directly at the surface. Sand transport is triggered by the drying of only the uppermost few particles resting on the surface (Neumann and Langston 2006).

This study provides parameter input, at an improved spatial and temporal resolution, for the control of surface moisture content on the threshold shear velocity (u_{*t}) at which sand begins to move. It contributes to a larger spatiotemporal model for predicting aeolian sand transport rate

and available sand budget for foredune development on sandy beaches in a temperate climate. Using a photographic methodology and regression analysis (Darke and McKenna Neuman 2008; Darke et al. 2009) beach surface moisture content is linked to a corresponding normalized beach surface reflectance. This relation is applied to five minute interval photographs of a sandy beach in a temperate climate (study area: Ameland, The Netherlands), resulting in temporal beach surface moisture maps. Using these maps as an input parameter for temporal threshold friction velocity maps, it is hypothesized that sand transport rate controlled by surface moisture content can more accurately be predicted.

Research objective

To develop temporal surface moisture mapping on a sandy beach in a temperate climate, as input parameter for a spatiotemporal model for predicting aeolian sand transport rate and available sand budget for foredune development.

Research goals

1. To establish a regression model between beach surface moisture content and normalized beach surface reflectance, and analyze its goodness of fit
2. To develop a data processing method using Java that allows for automated temporal beach surface moisture mapping

Theoretical Background

Aeolian sand transport

Aeolian sand transport is the result of individual sand particle movement in response to the wind force; the prevailing transport mode is saltation. Bagnold (1941) used the saltation process to derive an aeolian sand transport equation. Erosivity is expressed in terms of a shear velocity (u_*) and erodibility is expressed by a threshold shear velocity (u_{*t}). Aeolian sand transport occurs when the shear velocity exceeds the threshold shear velocity, i.e. the minimum shear velocity required to overcome the forces holding the particles on the surface.

The wind speed (u) at the surface bed is characterized by the Prandtl–von Karman (1935) equation: $u(z) = \frac{u_*}{k}\ln\left(\frac{z}{z_0}\right)$. Where $u(z)$ (m s^{-1}) is the average wind speed at height z (m), u_* (m s^{-1}) is the shear velocity, k is the von Karman's constant for turbulent flow, and z_0 (m) is the aerodynamic roughness height. For atmospheric air k is 0.4. The shear velocity (u_*) is related to the shear stress (τ) at the surface bed and the air density (ρ) by $u_* = \sqrt{(\tau/\rho)}$. With a minimum of two average wind speed measurements at two different heights, both u_* and z_0 are determined by linear regression (Shao 2000; Bauer 2009)

Sand grains are moved by the wind when the fluid forces (drag and lift) exceed the weight of the particle and cohesive forces between adjacent particles. As drag and lift on the particle increases, there is a critical value of u_* when grain movement is initiated. The threshold shear velocity (u_{*t}) is related to the particle density (ρ_p), the air density (ρ_a), the acceleration due to gravity (g), the particle diameter (d), and an empirical coefficient dependent on the surface grain characteristics (A) by: $u_{*t} = A\sqrt{\frac{\rho_p - \rho_a}{\rho_a}\, gd}$ (Nickling and Neuman 2009).

Soil moisture acts as a bonding agent and increases the threshold shear velocity (u_{*t}) at which sand begins to move (Chepil 1956). This control is expressed as $u_{*tw} = u_* f(moisture)$. Where u_{*tw} is the wet threshold shear velocity (in m s^{-1}), u_{*t} is the threshold under dry conditions, and f(moisture) is a function of the surface moisture expressed in terms of gravimetric moisture content *w* (kg kg) (Cornelis and Gariels 2003).

Aeolian sand transport rate q is given by: $q = K\left[\frac{u_*}{\sqrt{gD}}\right]^3$. Here (q) is the sand transport rate (in kg m^{-1} s^{-1}), (u_*) the shear velocity, (g) the acceleration of gravity, (D) the mean sand grain diameter, and (K) an aeolian sand transport coefficient. Aeolian sand transport occurs when $u_* > u_{*t}$. This equation is useful to estimate sand transport rates for D < 1.0 mm (Chapman 1990).

Color Image Formation

The color of an object is not an actual intrinsic property (or attribute) of the object itself (Bernard et al. 2002ab; Ebner 2007). Rather it is an interaction between light source, object, and observer. This notion is formalized in a model for color image formation (Finlayson et al. 1998). The light source (illuminant) is characterized by the spectral power distribution $E(\lambda)$. This specifies how much energy (i.e. power) the light source emits at each wavelength (λ) of the spectrum. The object is characterized by the reflectance function $S(\lambda)$. This defines what proportion of light energy incident upon it reflects on a per-wavelength basis. And the response of the observer is characterized by the spectral sensitivity function $Q_k(\lambda)$. This specifies its sensitivity to light energy at each wavelength of the spectrum. The subscript *k* denotes that this is the k^{th} sensor of the observer.

Now, a point in a color image q_k is defined as (see figure 1): $q_k = \int_\omega E(\lambda)\, S(\lambda)\, Q_k(\lambda)\, d\lambda$ with $k = 1, \ldots, m$. The integral is taken over the range of wavelengths (ω) a sensor is sensitive. As most optical devices have three sensors (m = 3), the response of a device to a point in a scene (i.e. a pixel) is represented by a triplet of RGB values. A color image is a collection of RGB triplets representing the response of an observer to light from a range of positions in a scene (Hordley et al. 2002). It follows that for color to be a useful cue in recognizing or identifying an object, the color must relate directly to the intrinsic properties of the imaged object. In other words the color of an object must be normalized in some manner, to be independent to the intensity and spectral composition of the illumination as well as the spectral sensitivity of the imaging device (Finlayson et al. 1998). Only then a RGB is an intrinsic representation of an object.

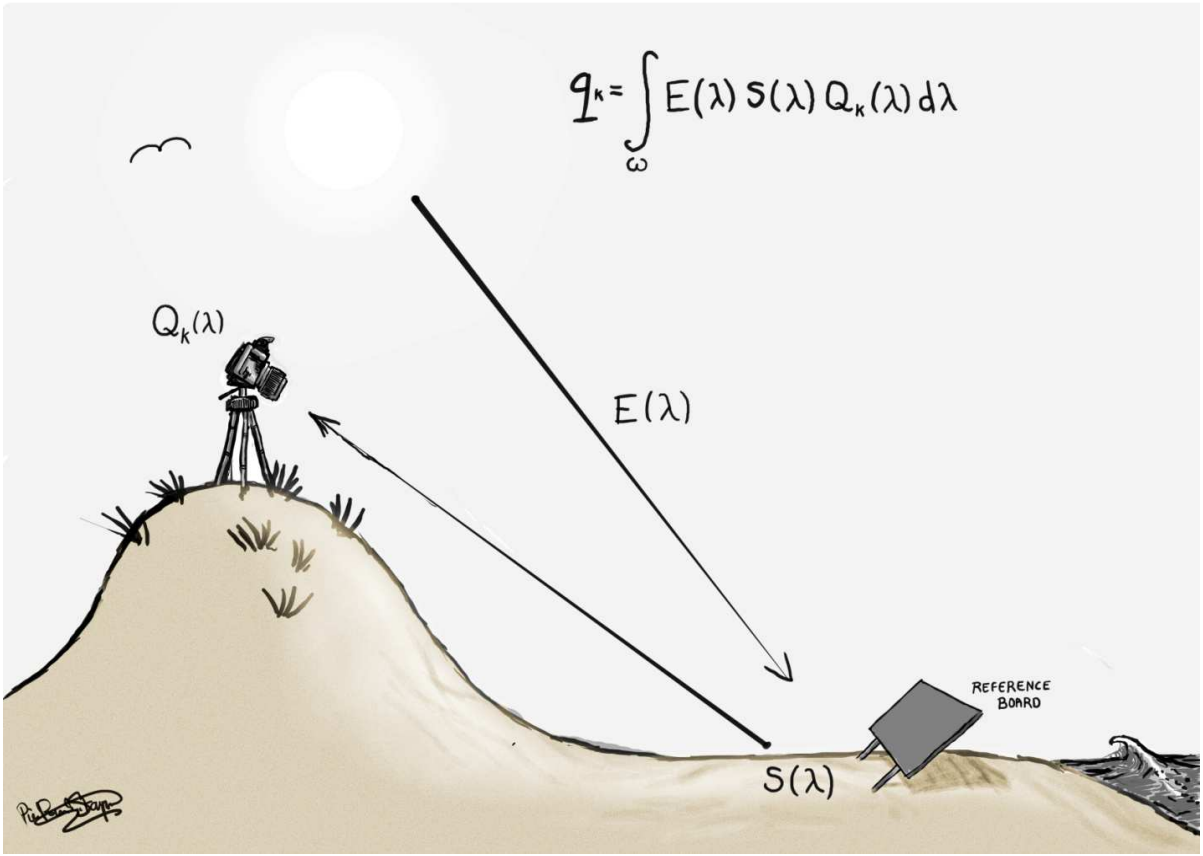


Figure 1. Model of color image formation (Stapel 2011)

In figure 1 the model of color image formation is shown, applied to the setup of this study. The illuminant is the sun, the object is the beach surface, and the observer is a time-lapse camera. The aim is for the spectral reflectance function $S(\lambda)$ of the object to be independent to both the spectral power distribution $E(\lambda)$ of the illuminant and the spectral sensitivity function $Q_k(\lambda)$ of the observer. Or, in other words, the aim is for the color signal $C(\lambda) = E(\lambda)\, S(\lambda)$ of the object to be illuminant and observer (i.e. device) invariant (Finlayson et al. 1998; Hordley et al. 2002). This requires a normalization procedure, in which an important step is the utilization of a gray card. In photography a gray card is an essential piece of equipment. It provides a convenient reference for color balance, because it has a flat (identical RGB) and neutral Lambertian (18%) reflectance spectrum, invariant of illuminant color and geometry (Hunt 1991). Including a gray card (or reference board in figure 1) in a scene allows the color of the object to be independent to the variability of $E(\lambda)$ in terms of intensity and wavelength.

Surface reflectance

The propagation of electromagnetic radiation in sand is dictated by two processes: scattering and absorption. The penetration depth of electromagnetic radiation in the visible and infrared spectrum is in the order of a few millimeters. This implies that surface reflectance and subsequent observed luminance is dominated by the upper most sand layer (Bänninger 2004; Ciani et al 2005).

The addition of water to sand results in a darker color. The manner in which surface moisture content affects the surface reflectance function $S(\lambda)$ of the beach varies across the electromagnetic spectrum. Wavelengths in the shortwave−infrared region of the electromagnetic spectrum (SWIR 1500 to 2500 nm) are most suitable for surface moisture prediction (Kaleita et al. 2005). This is due to spectral absorption features of water; absorption of electromagnetic radiation by water is prominent in the short wave infrared region of the electromagnetic spectrum. Strong absorbance by water (vapor) occurs at wavelengths around 2900, 1950 and 1450 nm, with weaker absorption around 1200 and 970 nm. Here an exponential decrease of reflectance upon wetting is observed, corresponding to the Beer-Lambert law of water absorption (Muller and Decamps 2001; Somers et al. 2010).

At wavelengths in the visible part of the spectrum there is no absorption band for liquid water. Here scattering is the predominant mechanism of decrease of surface reflectance upon wetting (Weidong et al. 2003; Somers et al. 2010). Scattering occurs when light hits an air–water, air–solid, or solid–water interface. The difference in the refractive index (n) between sand minerals (n_{soil}≈1.5) and the air surrounding the sand particles (n_{air}≈1.0) is relatively large. For wet sand the medium around the particles changes from air to water (n_{water}≈1.33). The difference in refractive index at the sand-water-air surfaces diminishes, resulting in a decreased scattering of incident light (Somers et al. 2010). Since scattering acts on the level of individual particles of different material (i.e. air, water, and sand minerals), the behavior of the surface reflection function $S(\lambda)$ of the beach under different moisture conditions cannot be modeled in full detail. Simple linear regression analysis is the best option to establish the overall trend that increasing beach surface moisture content results to a decreased luminance.

Materials and Methods

Data collection

Normalized beach surface reflectance

Modeling the relationship between a dependent variable y (normalized surface reflectance) and an independent variable x (surface moisture content) involves direct measurement, resulting in n pair observations (x_n, y_n). Figure 2 shows the measurement procedure. A rectangular frame is laid out on the beach surface, with a reference plate that approximates the characteristics of a photographic gray card with spectrally neutral and flat RGB (18% reflectance). A total of 135 color photographs (JPEG in sRGB color space with fixed exposure time 1/125 and aperture F5.0) are taken to quantify the surface reflectance at wavelengths in the visible part of the spectrum. For easier processing, the RGB color photographs are converted to an 8-bit ($2^8 = 256$ integer) luminance image by: $Y = 0.2989R + 0.5870G + 0.1140B$. The luminance images are normalized by assigning a constant mean luminance value to each reference plate. Considering the reference plate an actual gray card, then its mean luminance value is a constant (invariant of illumination conditions and geometry) of $Y = 117.6$ (for sRGB color space). The factor from which the actual mean luminance value differs from the assigned value is calculated, and subsequently applied to the associated mean luminance value of the beach surface. This returns the normalized mean luminance value of the beach surface.

Beach surface moisture content

Directly after taking a photograph the surface moisture content is measured inside the rectangular frame using two methods; a microtome slice and the Theta probe (see figure 2). The Delta-T Devices Theta probe (type ML2x) is based time domain reflectometry. Two stainless steel rods are inserted into the sand and a 100 MHz signal is transmitted. The difference in impedance between the probe and the line of transmission generates a reflected signal. The signal is converted to a voltage (Mv) that is dependent on the apparent dielectric constant of the sand, and is a function of the properties of the sand grains and the (volumetric) moisture content (θ). Following the method of Yang and Davidson-Arnott (2005), the active length of the

Figure 2. Surface moisture content measurement within a rectangular frame with reference plate

probe is reduced to 20 mm by inserting the stainless steel rods through a 40 mm thick cube of dielectric foam. Nine readings are taken within the rectangular fame to determine the average Theta probe reading. The Delta-T Theta probe is calibrated for saline moisture conditions following the method in Atherton et al. (2001). The microtome slice is an adaptation of the Surface Sediment Sampler (developed by Reginato 1975). A surface scraping (or microtome slice) of approximately 3 mm thick from within the rectangular frame is taken. The sample is bagged and weighted for determining mean gravimetric surface moisture content. Water content on a mass basis (w) is defined as the ratio of the mass of the liquid (m_l) in the given sand sample to the mass of the solid material (m_s): $w = \frac{m_l}{m_s}$. Water content on a volume base

(θ) is defined as the ratio of the volume of the liquid (v_l) in the given sand sample to the total volume of the sample (v_t): $\theta = \frac{v_l}{v_t}$. For comparison of both methods is the volumetric moisture content of the Theta probe converted to gravimetric moisture content using sand bulk density: $\theta = \frac{v_l}{v_t} = \frac{m_l/\rho_w}{m_s/\rho_b} = \left(\frac{m_l\,\rho_b}{m_s\,\rho_w}\right) = w\left(\frac{\rho_b}{\rho_w}\right)$. Here ρ_b is the bulk density of the sand (measured at 1367 kg m^{-3}) and ρ_w (1000 kg m^{-3}) is the water density (Wiggs et al. 2004).

Data analysis

Linear regression model

The n pair measured observations (x_n, y_n) of gravimetric moisture content *w* (%) and normalized luminance (0 – 255) provides model input for regression analysis. A regression model relates y to a function of x and β by $y \approx f(x, \beta)$ where β is a parameter of variable x. The regression function in this study is simple linear regression; a linear regression model with a single predictor (independent) variable. Simple linear regression fits a regression function through a set of *n* pair observations using the ordinary least squares (OLS) approach. This method obtains parameter β estimates that minimize the sum of squared residuals (SSE). The residuals are squared to avoid negative values. With n pair observations $(x_1, y_1),\ \ldots, (x_n, y_n)$ the true (i.e. population) linear relationship is described by: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ with $i = 1, \ldots, n$ and $\varepsilon_i \sim N(0, \sigma^2)$ or $y_i \sim N(\mu_y, \sigma^2)$. This denotes that the errors ε_i are assumed to be independent and normally distributed with a mean of zero and equal (but unknown) variance. Also, the observations y_i are assumed to be independent and normally distributed with mean $\mu_y = \beta_0 + \beta_1 x$ and equal (but unknown) variance. Given a random sample of observations from the population, the ordinary least squares approach estimates the population parameters through the simple linear regression model described by: $\hat{y} = b_0 + b_1 x + e_i$ with $i = 1, \ldots, n$. The residual (error) $e_i = y_i - \hat{y}_i$ is the difference between the value of the dependent variable predicted by the model $\hat{y}_i$ and the true value of the dependent variable y_i (Cohen et al. 2003; Miles and Shevlin 2001).

Regression diagnostics

Regression diagnostics analyzes the goodness of fit of the OLS simple linear regression model. Checks of goodness of fit are the coefficient of determination R^2, analysis of the pattern of residuals, and hypothesis testing (Cohen et al. 2003; Miles and Shevlin 2001).

Coefficient of determination

The coefficient of determination R^2 is a measure of how well the least squares equation $\hat{y} = b_0 + b_1 x$ performs as a predictor of y. The better a linear regression fits the data in comparison to the simple average, the closer the value of R^2 is to 1. The square root of R^2 returns the (Pearson product-moment) correlation coefficient (denoted by r). It is used as a measure of the strength of linear dependence (i.e. correlation) between two variables. A correlation is strong for values between 0.5 and 1.0. Below a correlation of 0.3 the linear dependence between two variables is too weak to have significance.

Analysis of residuals

Analysis of residuals is used to verify whether the data meets the criteria of linear regression, and to detect the so-called outliers. There are four assumptions that justify the use of a linear regression model for the purpose of prediction: linearity of the relationship between dependent and independent variables, normality of the error distribution, homoscedasticity (uniformity of variance) of the errors, and independence of the errors (no serial correlation). If an assumption is violated (i.e. nonlinearity, non-normality, heteroscedasticity, and/or serial correlation), then the predictive quality of linear regression model is (at best) inefficient or (at worst) seriously biased or misleading (Cohen et al. 2003; Miles and Shevlin 2001).

An outlier is an observation with a large residual and indicates unusual and/or influential data. Diagnostic statistics indicate for each observation three properties that help to identify an outlier: discrepancy (an observation with deviation of residual in the *y*-direction), leverage (an observation with deviation of residual in the *x*-direction), and influence (an observation with deviation of residual in the *x*- and *y*-direction). Each measure for detecting an outlier has a particular cut-off value, depending on the number of parameters (p) and observations (n). Table 1 shows the cut-off values for the diagnostic statistics used to detect outliers.

Table 1. Cut-off values of measures for detecting outliers

Property	Measure	Cut off value
Discrepancy	Studentized Deleted Residual	> 2
Leverage	Leverage	$> (2p + 2)/n$
Influence	Cook's Distance	$> 4/n$
Influence	DfFit	$> 2/\sqrt{p/n}$

Hypothesis testing

The statistical significance of the OLS linear regression model is checked through hypothesis testing. If there is a significant linear relationship between the independent variable x and the dependent variable y, the slope β_1 is not equal to zero. The null hypothesis and an alternative hypothesis become: $H_0: \beta_1 = 0$ or the slope of the regression line is equal to zero and $H_a: \beta_1 \neq 0$ or the slope of the regression line is not equal to zero. Accepting or rejecting the null hypothesis depends on the significance α level the OLS linear regression model is tested, the test method, and the P-value associated with the test statistic. Here at a significance level $\alpha = 0.05$ the hypothesis is tested with a t-test and F-test. If the P-value is less than the significance level α, the null hypothesis is rejected (Cohen et al. 2003; Miles and Shevlin 2001).

Data processing

The second research goal is to develop a data processing method using Java that allows for automated temporal beach surface moisture mapping. The approach is to apply the OLS linear regression model of normalized surface reflectance (i.e. normalized luminance) and surface moisture content in a spatiotemporal manner, resulting in time-series of normalized luminance beach maps and surface moisture beach maps. It starts with continuous (not normalized) five-minute interval photographs of the beach where direct aeolian sand transport measurements have taken place. This is to provide calibration and validation data for the (to be developed) aeolian sand transport prediction model. The optical device is a waterproof 4 megapixel time-lapse camera. By mounting this device on a pole on top of a dune it is able to take a five minute interval JPEG photograph (in sRGB) of the beach from a bird's eye perspective.

Normalized luminance beach maps

The temporal horizontal distribution of surface moisture content of the beach is visualized after applying two normalization procedures to every individual photograph; device invariance normalization and illuminant invariance normalization. Figure 3 shows the normalization procedure that is applied to each picture element (i.e. pixel) of every photograph. First every pixel is converted to an 8-bit luminance pixel, then it is normalized by multiplying it with a device invariant pixel factor (F_{sr}) and an illuminant invariant photo factor (F_{bs}).

Device invariance normalization deals with differences in spectral response of each (MOS capacitor) image sensor element of an optical device. Each picture element (i.e. pixel) of a photograph has a slightly different sensitivity to electromagnetic radiation. For the color and intensity (i.e. chrominance and luminance) of the object (or point in a scene) to be device invariant, two operations are applied. First, to obtain comparable images, the camera is set to operate with as many fixed settings (aperture, shutter speed, white balance etc.) as possible. Second, to ensure identical spectral response from each (MOS capacitor) image sensor element, a normalization procedure is applied through utilization of a gray card (figure 3). The device invariant pixel factor F_{sr} for normalizing the sensor response is obtained by correcting every pixel (i.e. MOS capacitor sensor element) to the mean luminance pixel of a photograph taken of a gray card. The deviation from the mean determines the factor; it is pixel-specific and is therefore different for every pixel.

Illuminant invariance normalization deals with differences in intensity and spectral composition of the illumination at the time a photograph is taken. For the color and intensity (i.e. chrominance and luminance) of the object (or point in a scene) to be illuminant invariant, a normalization procedure is applied through a reference board placed in the scene that approximates a gray card (figure 3). The illuminant invariant photo factor F_{bs} for normalizing the beach surface is obtained by correcting the mean pixel value (of 16 pixels) of the reference board to the 'true' luminance pixel value of an actual gray card (117.6 in sRGB). The deviation from this value determines the factor; it is one value for every pixel but different for each photograph.

Surface moisture beach maps

The normalized (i.e. device and illuminant invariant) luminance beach maps are transformed into surface moisture beach maps by assigning the associated surface moisture content to the normalized surface reflectance (i.e. normalized luminance). This involves a relatively simple procedure of applying the linear regression model $\hat{y} = b_0 + b_1 x + e_i$ established in research goal 1. Figure 3 shows the procedure of transforming every normalized luminance pixel into a surface moisture pixel through the OLS linear regression model. One extra processing step is necessary to align the range of luminance. A normalized luminance beach map (potentially) employs the full 8-bit range of 256 integers. The OLS linear regression model, however, employs a limited range of normalized luminance (associated with the minimum observed value for dry sand and maximum observed value for wet sand). In order to apply the OLS linear regression model and transform the normalized luminance beach maps into surface moisture beach maps, the range of the normalized luminance of the beach maps is aligned linearly to the range of the minimum $(0, y_{min})$ and maximum $(255, y_{max})$ observation used in the OLS linear regression model. The acquired surface moisture content is visualized by assigning classes of moisture content with a 4% moisture content interval.

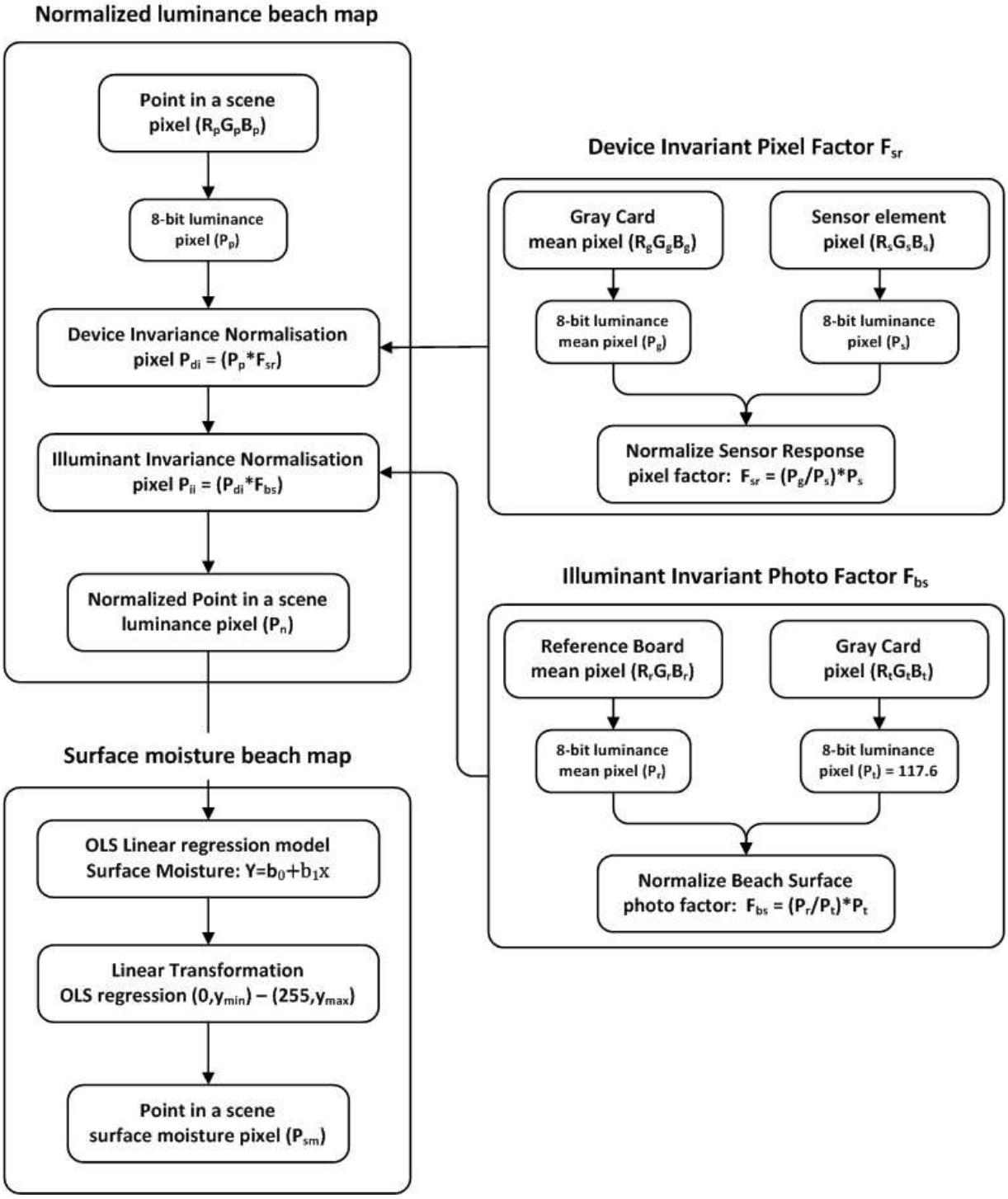


Figure 3. Normalized luminance to surface moisture for every pixel in a photograph

Results

Data collection

Calibration of the Theta probe for saline sandy beach conditions delivers the following linear relation (r of 0.98) between probe output (Mv) and gravimetric soil moisture content w $(\%)$: $Probe\ w = (0.15 * Probe\ output) + 8.02$. This relation is applied to the average probe output (nine measurements per photograph) of the 135 rectangular frame photographs to return its corresponding surface moisture content (figure 4a, annex B).

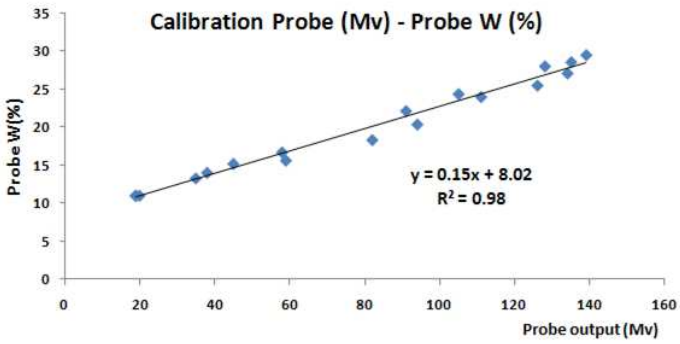


Figure 4a. Calibration Theta probe

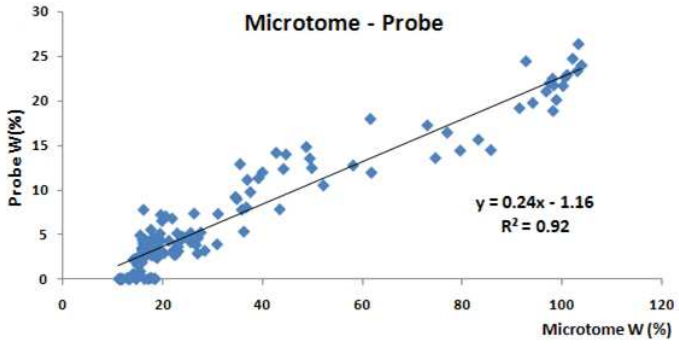


Figure 4b. Correlation Microtome slice and Theta Probe

The relationship between the two methods of measuring gravimetric surface moisture content (microtome slice and Theta probe) is depicted in figure 4b. The surface soil moisture content per photograph of both methods is plotted against each other. Simple linear regression delivers the relation $Probe\ w = (0.24 * Microtome\ w) + 1.16$ with r of 0.96. Although this is a strong correlation, a number of photographs show a large variation in measurement of surface soil moisture content between both methods. This discrepancy is not due to measurement error; rather it exposes the difference in (vertical) scale at which the two methods measure surface soil moisture content. The microtome slice method is able to measure this uppermost surface layer whilst the Theta probe method is not. After discarding the largest discrepancies between the two methods (in 28 photographs), the probe is found to measure surface moisture content with a factor five higher than the microtome slice. As a result, the theta probe does not measure surface moisture content below ten percent. This is higher than the threshold of five percent moisture content for which soils are considered to be too wet for saltation by most

natural winds. Further, the linear regression model of the Theta probe method returns a R^2 of only 0.12 (correlation of 0.35). For these reasons it is decided to discard the results obtained with the Theta probe method and are therefore not used for producing time-series of surface moisture beach maps.

Data analysis

The simple OLS linear regression model $\hat{y} = b_0 + b_1 x + e_i$ with $i = 1, \dots, n$ for the full data set of the microtome slice method is constructed (figure 5, annex B). With a P-value of 0.00 for both the t-test and F-test the linear regression model $\hat{y} = 97.555 - 1.088x$ is significant at level $\alpha = 0.05$. So the null-hypothesis $H_0: \beta_1 = 0$ (the slope of the regression line is equal to zero) is rejected and the alternative hypothesis $H_a: \beta_1 \neq 0$ (the slope of the regression line is not equal to zero) is accepted. The coefficient of determination R^2 for the model is 0.243 and the associated correlation r is 0.493. So the linear dependence between the two variables gravimetric *w* (%) and normalized luminance *Y* (0 – 255) is medium to strong. The regression model violates the assumptions of normality and homoscedasticity of the errors. In a normal distribution both skewness and kurtosis are zero, i.e. perfect symmetry and perfect peakedness. Here, the skewness and kurtosis are 0.476 and 0.540 respectively. The distribution of the residuals plotted against the predicted values is homogeneous in the y-direction but there is considerable skewness in the x-direction; a sign of heteroscedasticity (i.e. non-uniform variance).

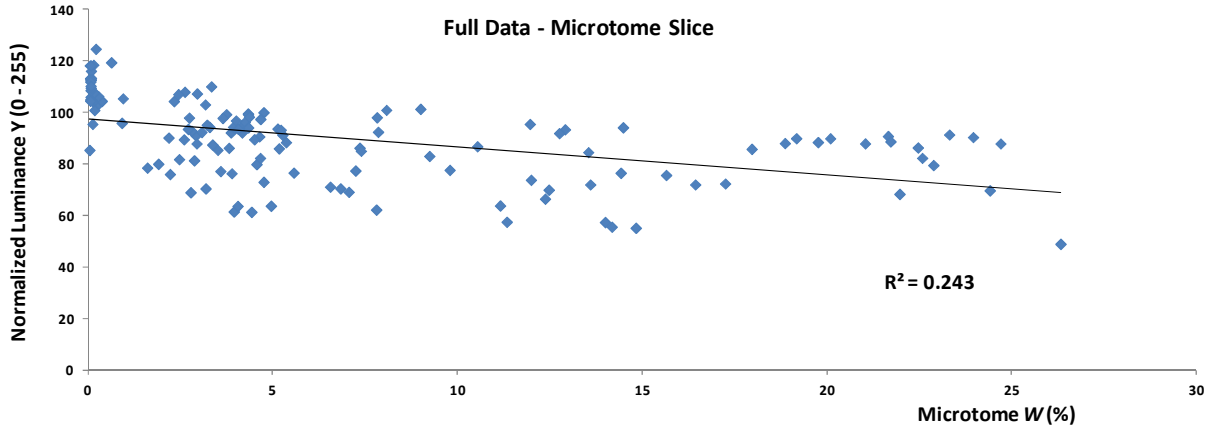


Figure 5. Scatter plot full data microtome slice gravimetric *W* (%) and normalized luminance *Y* (0 – 255)

Outlier detection results in a corrected data set that is used as input for the new OLS linear regression model. Now the linear relationship between gravimetric moisture content *w* (%) and normalized luminance *Y* (0 – 255) is described by $\hat{y} = 100.557 - 1.533x$. This linear relationship is significant at level $\alpha = 0.05$ and the linear dependence is (relatively) strong; the coefficient of determination R^2 for the model is 0.32, the associated correlation r is 0.565. However, this outlier corrected regression model still violates the assumptions of normality of the error distribution and uniform variance of the errors; skewness and kurtosis are 0.548 and 0.396 respectively. Still, this relationship is used as input for producing time-series of surface moisture beach maps even though its predictive quality is questionable.

Data processing

The methodology for reaching the second research goal involves programming in Java. The reader is referred to annex A for the source file normalisation.java. Here the data processing steps are documented in order of the program. The interface building process is left out of this report since it is not relevant for attaining the research objective. Figures 6 to 9 show the results for the data processing method that allows for automated temporal beach surface moisture mapping. Figure 6 shows the input reference JPEG of a gray card that can be set as the reference image for device invariance normalization. Figure 7 shows the original (unnormalized) beach surface luminance image. This image contains the reference board that can be selected for the illuminant invariance normalization step. Figure 8 shows a normalized (device and illuminant invariant) luminance beach map, and figure 9 a beach surface moisture beach map. Figures 10a-b-c show time-series of surface moisture beach maps. Every figure shows four five-minute interval surface moisture beach maps during different parts of the day (morning, mid-afternoon, and late afternoon). This is to illustrate how the illumination conditions change during one day, and how the data processing method performs. In every figure, the first row shows the original input five-minute interval color photographs, the second row the normalized (device and illuminant invariant) luminance beach maps, and the third row the surface moisture beach maps. The data processing method is automated through the batch function that enables input of large amount of five-minute photographs.

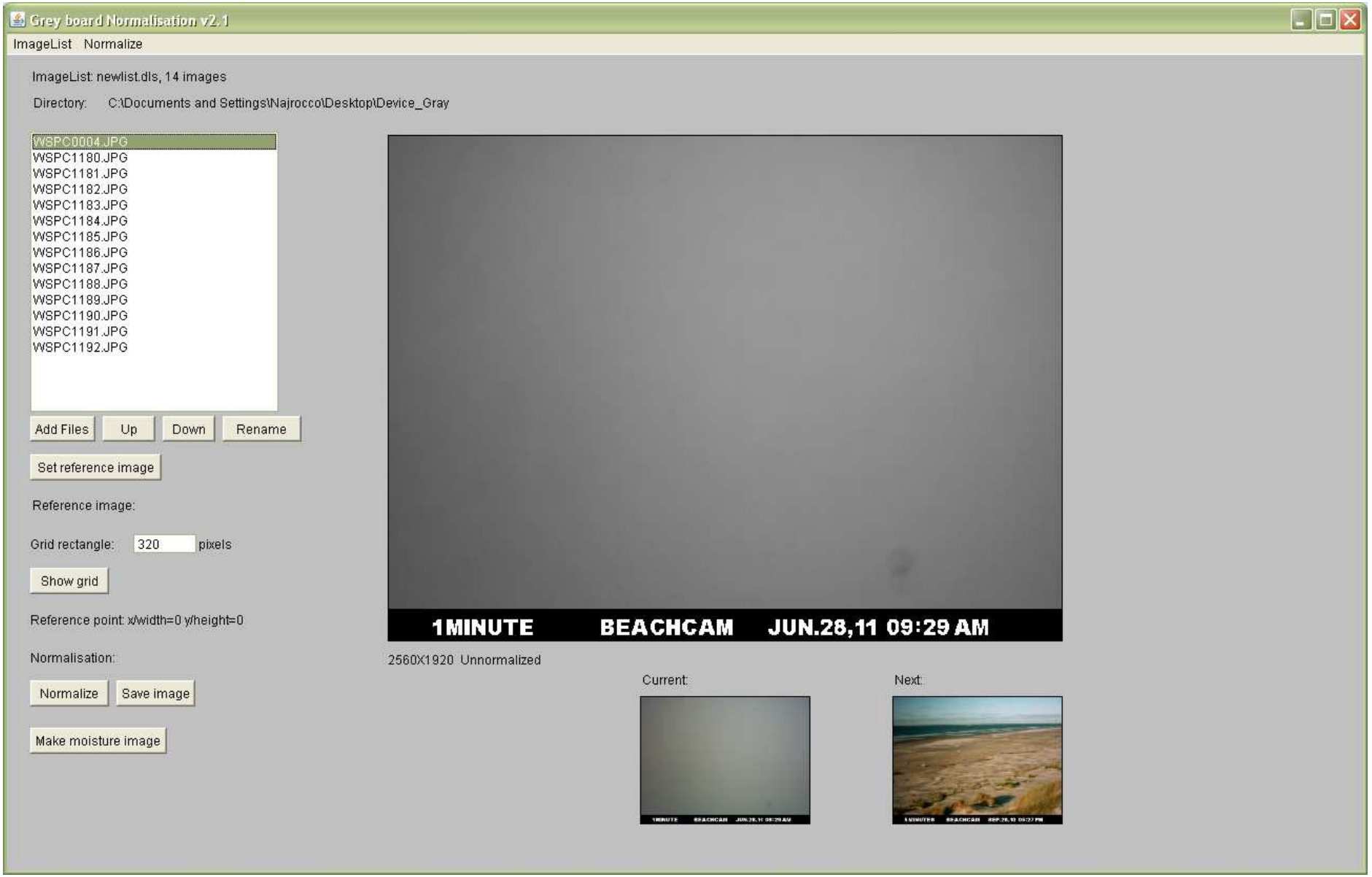


Figure 6. Reference gray card JPEG

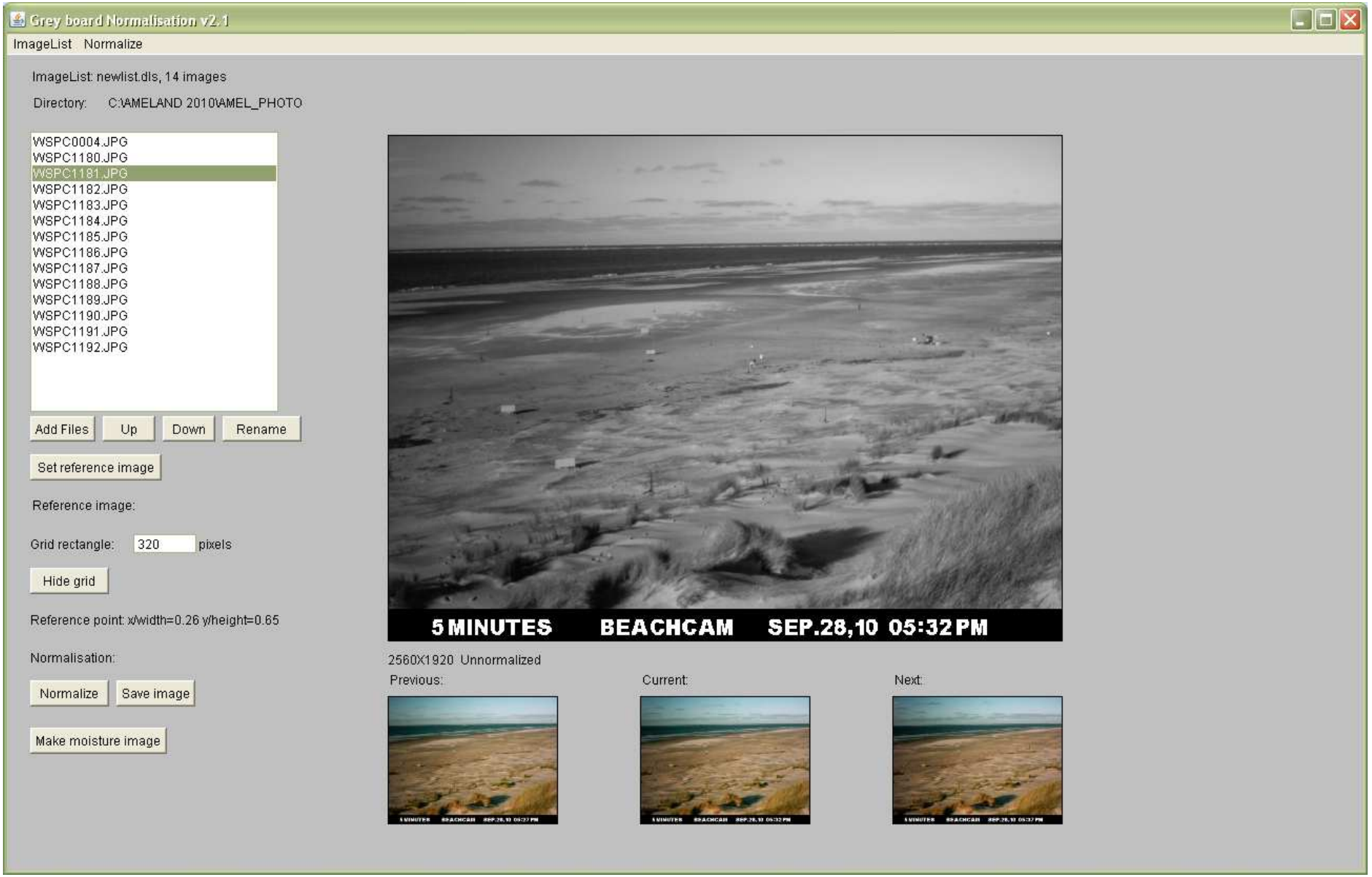


Figure 7. Original beach surface luminance JPEG

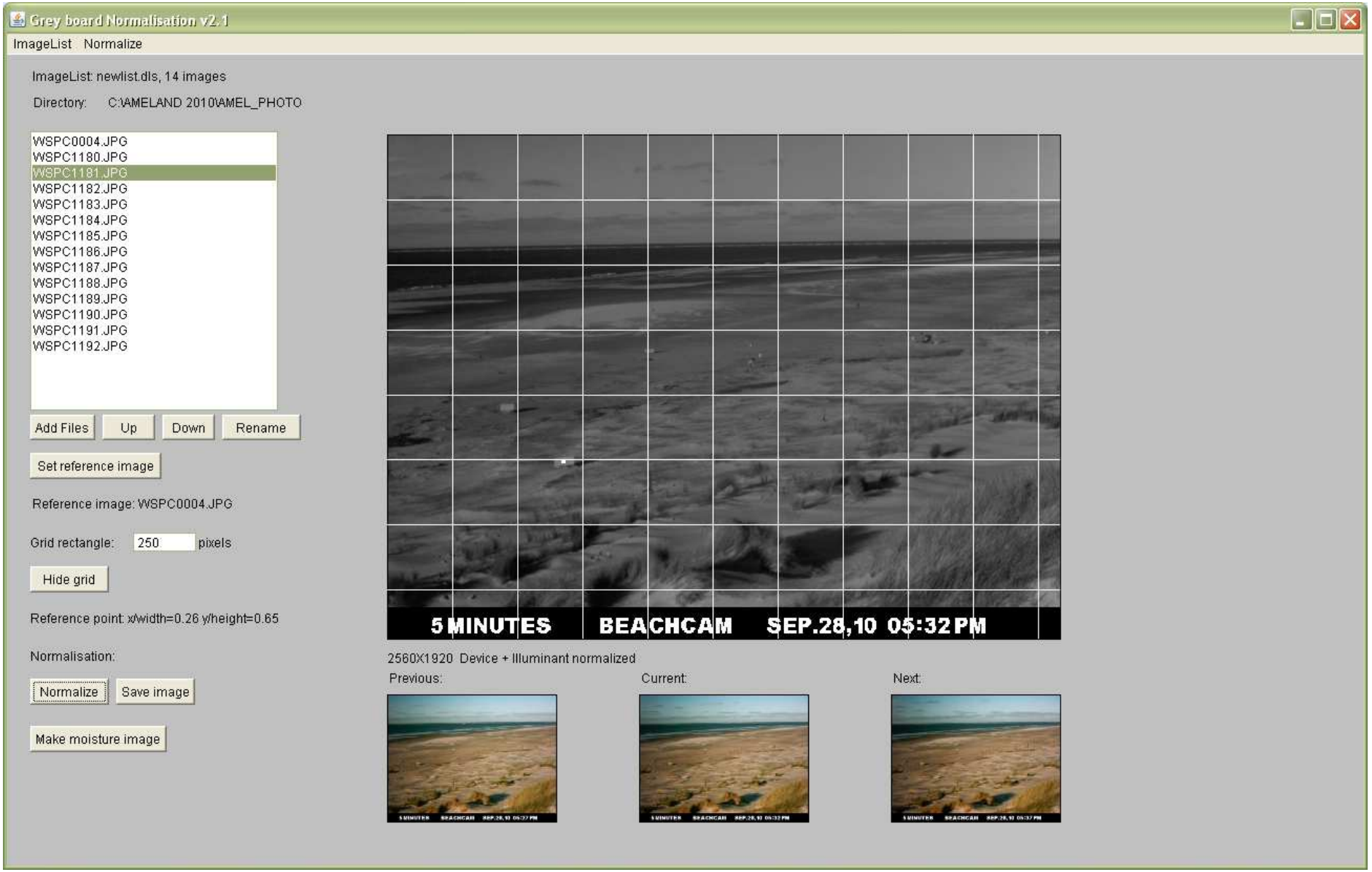


Figure 8. Normalized (device and illumination invariant) luminance beach map

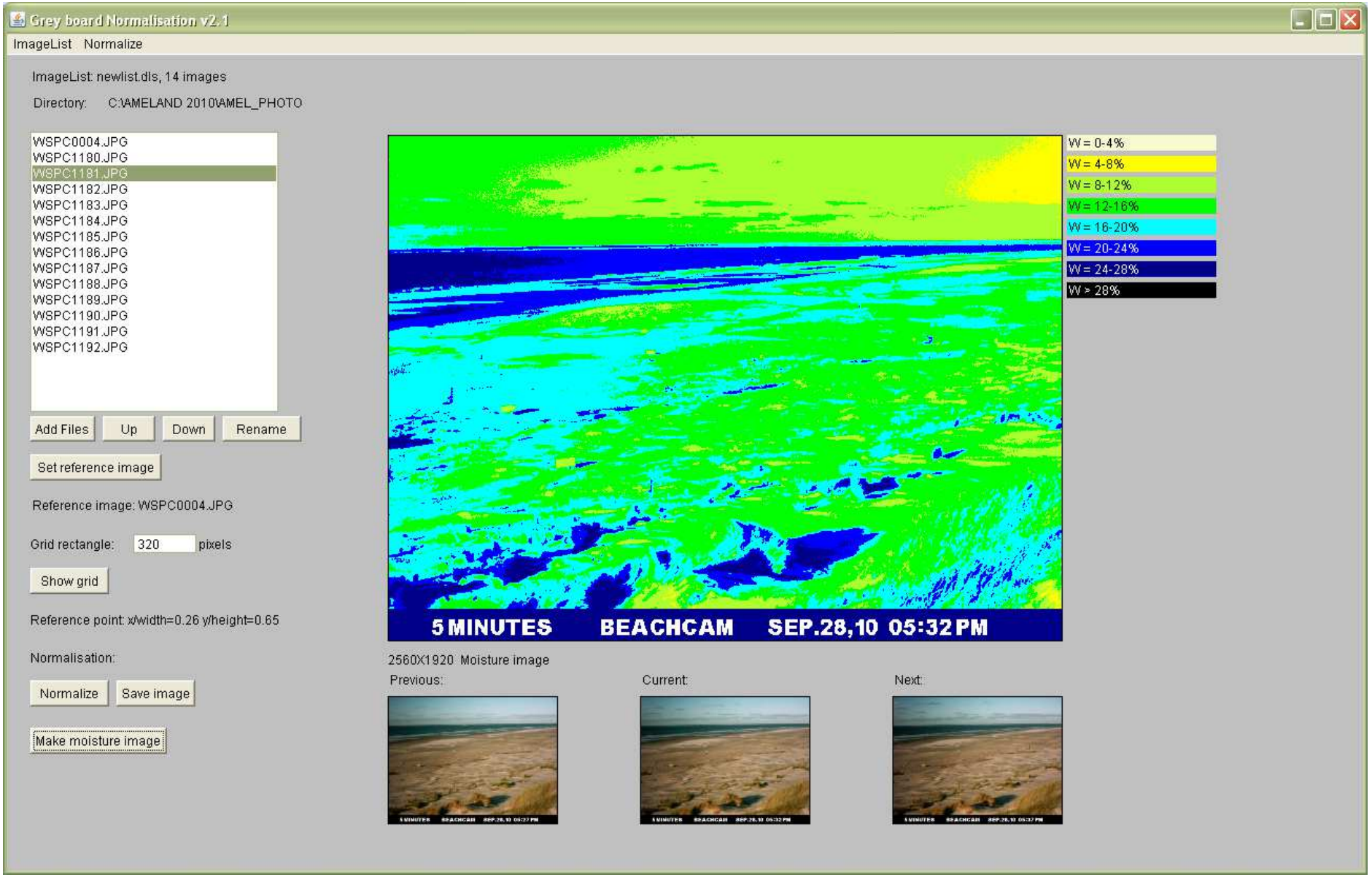


Figure 9. Surface moisture beach map

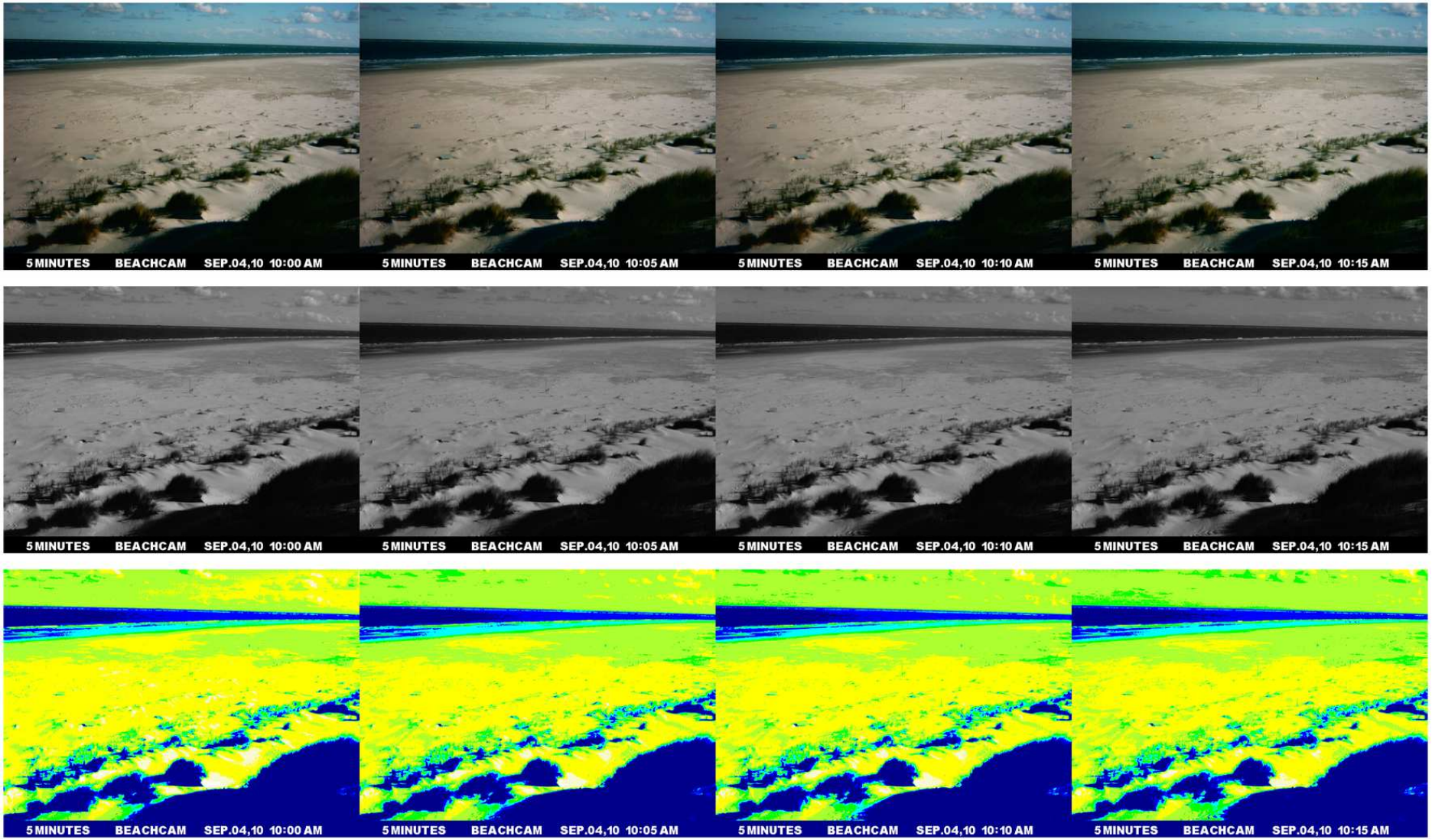


Figure 10a. Time series of surface moisture beach maps (morning 04092010)

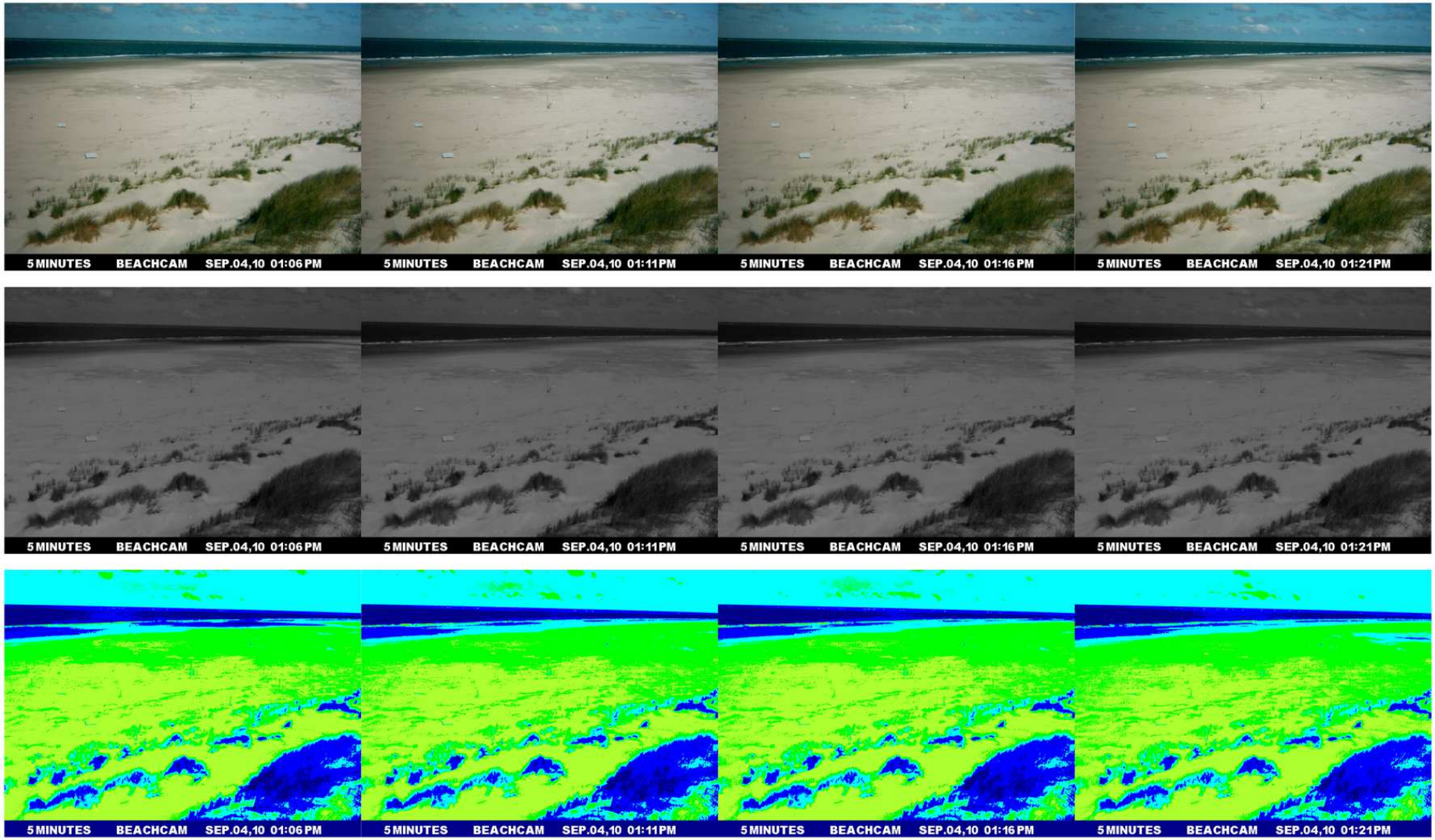


Figure 10b. Time series of surface moisture beach maps (mid-afternoon 04092010)

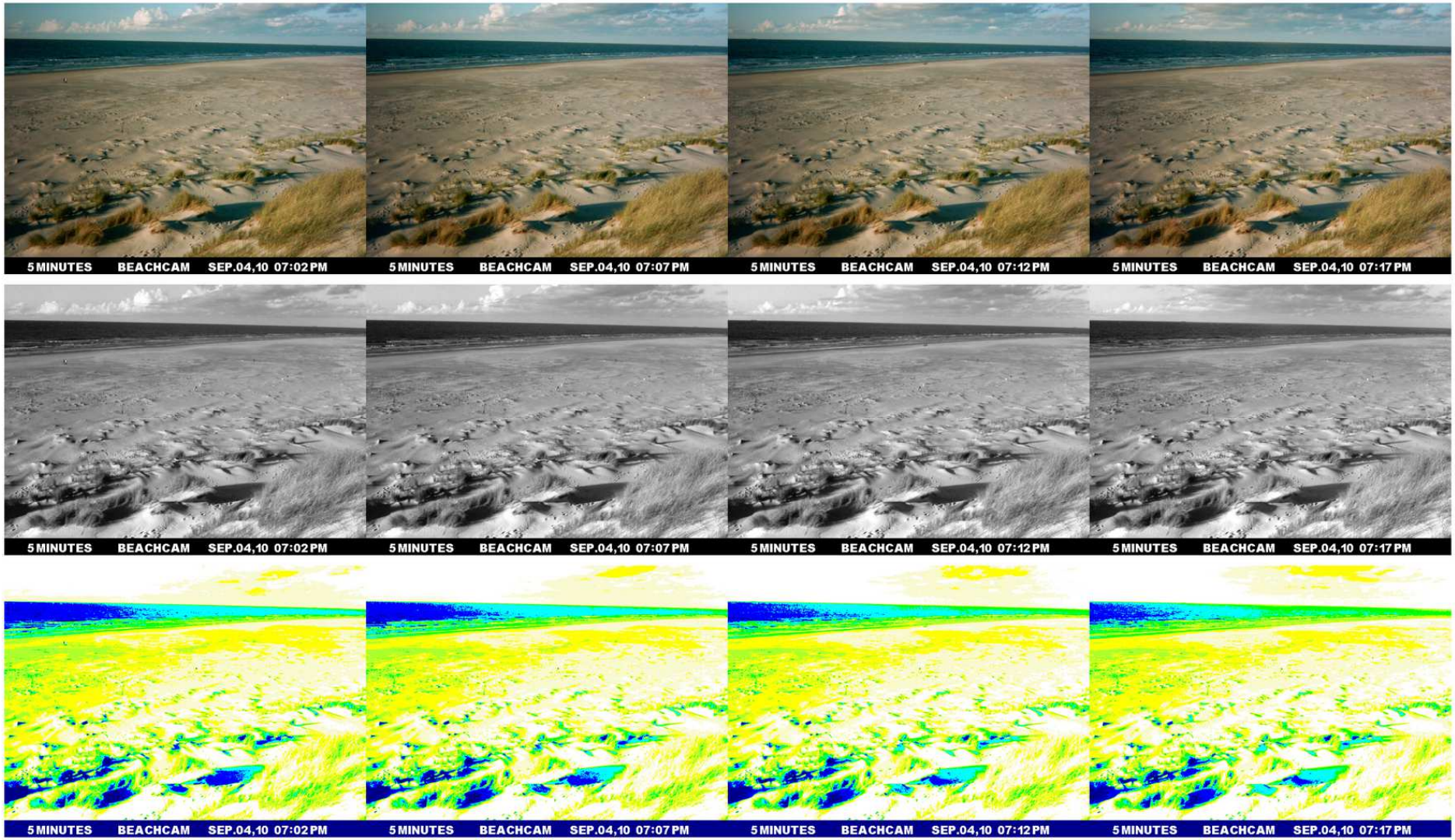


Figure 10c. Time series of surface moisture beach maps (late afternoon 04092010)

Discussion

The linear relationship between gravimetric moisture content *w* (%) and normalized luminance *Y* (0 – 255) $\hat{y} = 100.557 - 1.533x$ is significant at level $\alpha = 0.05$ and has a relatively strong correlation (0.565), but it violates the assumptions of normality of the error distribution and constant variance of the errors. As a result the predictive quality of the model is questionable.

There are a number of factors that contribute to distort (i.e. weaken) the linear relationship. The factor that contributes most is the fact that the surface reflectance (i.e. luminance) measurements have been taken at wavelengths in the visible part of the electromagnetic spectrum. Here, scattering is the mechanism of decrease of surface reflectance upon wetting. As a result the behavior of the surface reflection function $S(\lambda)$ of the beach under different moisture conditions cannot be modeled in full detail. A simple linear regression is the best option, but it only establishes the overall trend that increasing surface soil moisture content results to a decreased luminance.

A site-specific factor that weakens the linear relationship is the different weather conditions under which the photographs for surface reflectance measurements were taken. The photographs are normalized for changing illumination conditions, however not through utilization of a photographic gray card with spectrally neutral and flat RGB (18% reflectance). Instead a reference plate is used that approximates the characteristics of a photographic gray card. As a result, the normalization procedure has an intrinsic error since the reference plate responds unpredictable under varying illumination conditions. Another important site-specific factor that weakens the linear relationship is distortion of the beach surface by materials such as organic matter (seaweed, shells, plant material, etc.), other rubble (such as plastic, glass, etc.), and minerals other than the common constituent of sand in a temperate climate (i.e. quartz). At the study site there is disturbance from deposition of heavy minerals; minerals with a density larger than quartz. Heavy minerals in Dutch beach sands can be concentrated up to thirty percent (Schuiling et al. 1985) but the total mean concentration of heavy minerals in Dutch beach sands is low (about one percent). Effort to reduce this distortion was undertaken,

by discarding photographs where the beach surface has a high standard deviation. A high standard deviation in this context means large deviation from the mean grayscale pixel value, i.e. a beach surface that is not of equal reflectance (or luminance). But applying OLS linear regression to this (standard deviation corrected) data set did not deliver a linear regression model with an improved coefficient of determination. As a result, this route was not investigated further.

Judging the data processing method purely on its ability to allow automated beach surface moisture mapping, it is concluded that the method is effective. A large number of beach surface moisture maps can be automatically generated. However, there are a number of (qualitative) concerns about the output of the data processing method. This is due to shortcomings in the device and illumination invariant normalization steps.

The device invariant normalization step effectively evens out the differences in spectral response from each (MOS capacitor) in a single image sensor element. But it is effective only as long as one single optical device is used throughout the data collection, analysis, and processing steps. Unfortunately, this is has not been the case. In this study two different cameras are used; one for establishing the linear regression model between gravimetric surface moisture content *w* (%) and normalized luminance Y (0 – 255), and another for taking five-minute interval photographs of the beach surface from a bird's eye view. As a consequence, two cameras with two different spectral sensitivity functions $Q_k(\lambda)$ result in two different models for color image formation. This means that the established linear regression model between gravimetric surface moisture content *w* (%) and normalized luminance Y (0 – 255) is optical device specific, and cannot be transferred to photographs taken with another optical device. This error cannot be resolved since this insight came after the field study ended.

The illumination invariant normalization step suffers from a number of complications. Evaluating its effectiveness in normalizing changes in illumination conditions reveals that the changes cannot be too extreme. This is illustrated in figures 11a. When heap clouds, for instance, cast shadows on the beach surface, the illumination invariant normalization step cannot compensate for this. Or when the weather conditions suddenly changes from sunny to

overcast. These changes in illumination are too abrupt to normalize. As a result, the application of this data processing method is limited to more or less constant weather conditions.

Figure 11b shows sea water intrusions during unusual high tide. This distorts the results as the spectral reflectance function $S(\lambda)$ of the sea water different from that of the beach surface. Under the right angle the sea water acts as a mirror and reflects a great deal more visible light than the beach does. As a result it appears to be much drier than the beach surface when, in fact, it is open water. This is not a problem when photographs are manually inspected, but as part of an automated data processing method it leads to complications. The aeolian sand transport prediction model could, for instance, indicate it as a potential source area for sand transport.

Another issue is that there is a need for a data processing method to automatically differentiate between the objects in the photographs (sky, sea, beach, vegetation etc.) The (change in) luminance (or chrominance) value is potentially useful to function as a marker.

Not included in this study, but of practical importance for developing the aeolian sand transport prediction model, is an additional data processing step; georeferencing the input photographs that are used for surface moisture mapping. Georeferencing (or georectification) is a data processing step where an image is aligned to fit in a particular geographic reference coordinate system. This transforms spatial data into geospatial data, which is useful for inclusion into a geographic information system. The most obvious choice for a geodetic datum is the WGS 84 that is used by all GPS equipment. Georeferencing is done by measuring a number of ground control points that are included in the photograph. By interpolating latitude and longitude values of the ground control points an image is warped to fit into a particular geodetic datum.

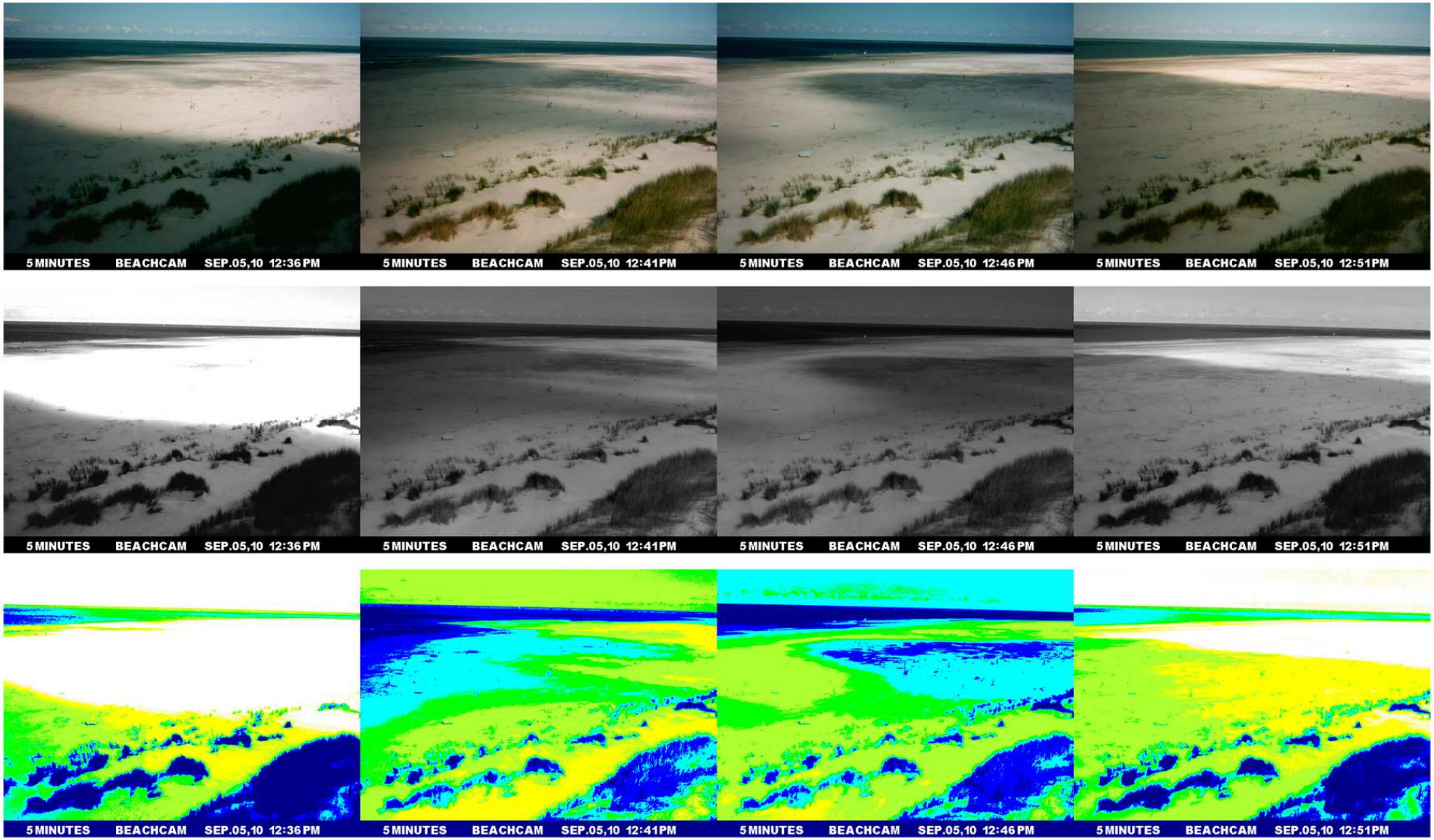


Figure 11a. Distorted results due to shading effects

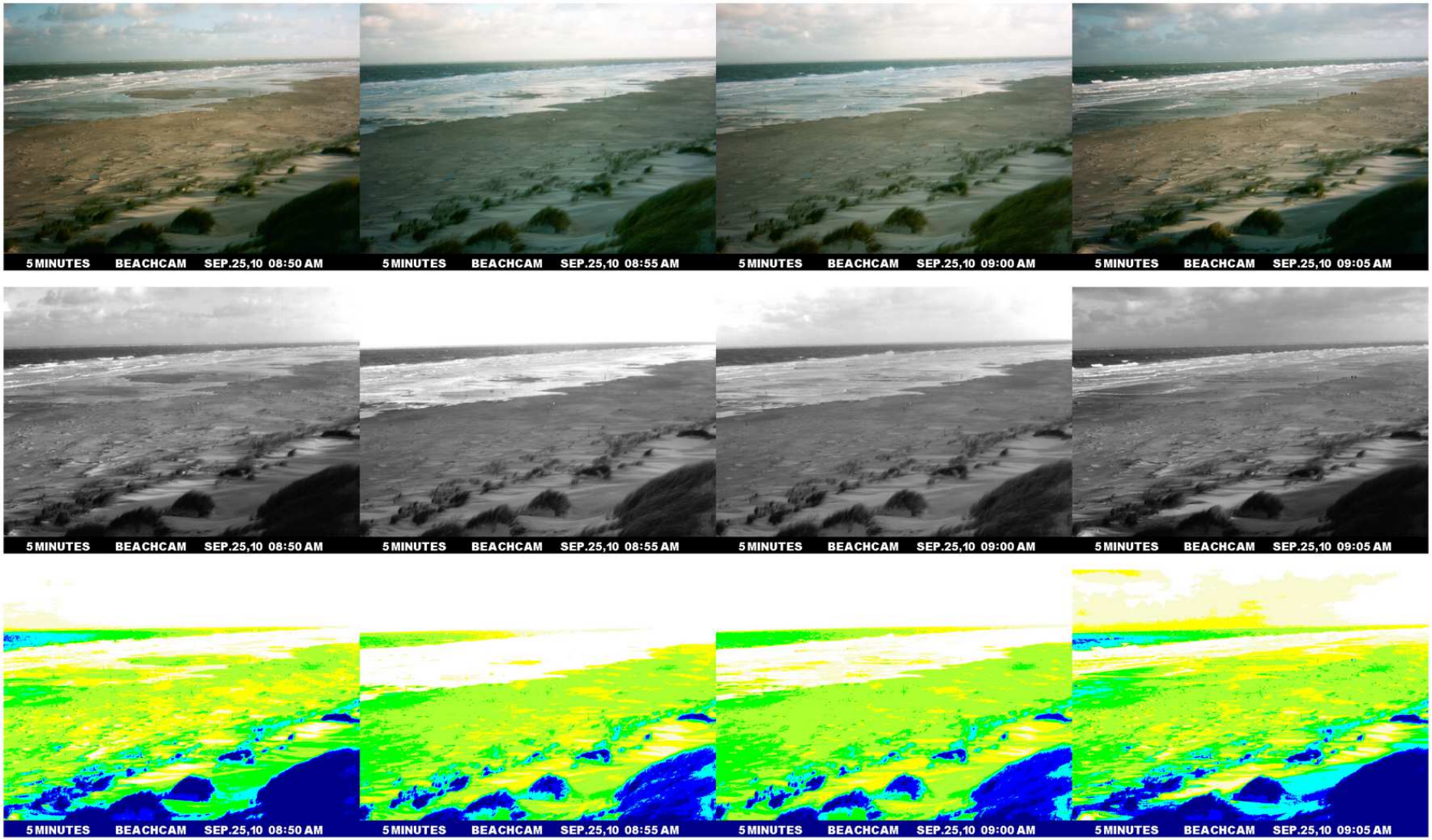


Figure 11b. Distorted results due to sea water intrusion

Conclusion and recommendations

The research objective of this study is to develop and use temporal surface moisture mapping, on a sandy beach in a temperate climate, as input parameter for a spatiotemporal model for predicting aeolian sand transport rate and available sand budget for foredune development. It provides parameter input, at an improved spatial and temporal resolution, for the control of surface moisture content on the threshold shear velocity (u_{*t}) at which sand begins to move. The control of surface moisture content on aeolian sand transport and supply rate is difficult to parameterize because of its high degree of spatiotemporal variability, and its operation at the scale of a particle directly at the surface.

The research objective is attained by two research goals; (1) to establish a regression model between beach surface moisture content and normalized beach surface reflectance, and analyze its goodness of fit, and (2) to develop a data processing method using Java that allows for automated temporal beach surface moisture mapping.

Aeolian sand transport is the result of individual sand particle movement in response to the wind force; the prevailing transport mode is saltation. Erosivity is expressed in terms of a shear velocity (u_*) and erodibility is expressed by a threshold shear velocity (u_{*t}). Aeolian sand transport occurs when $u_* > u_{*t}$. Soil moisture acts as a bonding agent and increases the required threshold shear velocity (u_{*t}) at which sand begins to move. This control is expressed as $u_{*tw} = u_* f(moisture)$. Where u_{*tw} is the wet threshold shear velocity (in m s^{-1}), u_{*t} is the threshold under dry conditions, and f(moisture) is a function of the surface moisture expressed in terms of gravimetric moisture content *w* (kg kg).

The color of an object is not an intrinsic property (or attribute) of the object. This notion is formalized in the model for color image formation, where a point in a color image is defined as: $q_k = \int_\omega E(\lambda)\, S(\lambda)\, Q_k(\lambda)\, d\lambda$. In this study the aim is for the spectral reflectance function $S(\lambda)$ of the object to be independent to both the spectral power distribution $E(\lambda)$ of the illuminant and the spectral sensitivity function $Q_k(\lambda)$ of the observer. This requires an illuminant and device invariance normalization procedure in which the utilization of a gray card provides a

reference for color balance, since it has a flat (identical RGB) and neutral Lambertian (18%) reflectance spectrum, invariant of illuminant color and geometry.

The manner in which surface moisture content affects the surface reflectance function $S(\lambda)$ of the beach varies across the electromagnetic spectrum. Wavelengths in the shortwave−infrared region of the electromagnetic spectrum (SWIR 1500 to 2500 nm) are most suitable for moisture prediction, due to the spectral absorption features of water. Here an exponential decrease of reflectance upon wetting is observed, corresponding perfectly to the Beer-Lambert law of water absorption. At wavelengths in the visible part of the spectrum scattering is the predominant mechanism of decrease of surface reflectance upon wetting. Here the behavior of the surface reflection function $S(\lambda)$ of the beach under different moisture conditions cannot be modeled in full detail, since scattering acts on the level of individual particles of different material (i.e. air, water, and sand minerals). For this reason simple linear regression the best option to establish the overall trend that increasing beach surface moisture content results to a decreased luminance.

Beach surface reflectance measurements are taken at wavelengths in the visible part of the electromagnetic spectrum. The OLS simple linear regression model $\hat{y} = b_0 + b_1 x + e_i$ with $i = 1, \dots, n$ for the outlier corrected n pair observations (x_n, y_n) of gravimetric moisture content w (%) and normalized luminance Y (0 – 255) is described by $\hat{y} = 100.557 - 1.533x$. This linear relationship is significant at level $\alpha = 0.05$ and the linear dependence is (relatively) strong; the coefficient of determination R^2 is 0.320 and the associated correlation r is 0.565. However, the regression model violates the assumptions of normality of the error distribution and constant variance of the errors; skewness and kurtosis are 0.548 and 0.396 respectively. As a result the predictive quality of the model is questionable. Still, this relationship is used as input for temporal beach surface moisture mapping.

Judging the data processing method on its ability to allow automated beach surface moisture mapping, it is concluded that the method is effective. A large number of beach surface moisture maps can be automatically generated. However, there are a number of qualitative concerns

about the output of the data processing method. This is due to shortcomings in the field measurement setup and the device and illumination invariant normalization procedure.

Shortcomings in the field measurement setup involve not using an actual photographic gray card, using two non-identical optical devices, and the omission of easily identifiable ground control points for georeferencing and image stabilization. Shortcomings in the device and illumination invariant normalization are intrinsic to the procedure. It is not (and will not be) practical in too extreme changes in illumination conditions. Its practicality is limited to relative constant weather conditions; either fully sunny or completely overcast.

However, some mentioned concerns can be overcome in future research. Beach surface reflectance measurements at wavelengths in the short-wave infrared region improves the strength of the (then) exponential regression model between gravimetric surface moisture content w (%) and normalized luminance Y (0 – 255). A more correct field measurement protocol and extending the data processing method for further automation, improves the effectiveness of surface moisture mapping as a spatiotemporal parameter for aeolian sand transport prediction.

References

Aagaard, T. Davidson-Arnott, R.G.D. Greenwood, B. Nielsen, J. (2004). Sediment supply from shoreface to dunes: linking sediment transport measurements and long-term morphological evolution. Geomorphology 60(1-2): 205-224.

Arens, S.M. (1996). Rates of aeolian transport on a beach in a temperate humid climate. Geomorphology 17(1-3): 3-18.

Atherton, R.J. Baird, A.J. Wiggs, G.F.S. (2001). Inter-Tidal Dynamics of Surface Moisture Content on a Meso-Tidal Beach. Journal of Coastal Research 17(2): 482-489.

Bagnold, R. A. (1941). The Physics of Blown Sand and Desert Dunes. 265 pp. Willliam Morrow, New York.

Barnard, K. Cardei, V. Funt, B. (2002a). A Comparison of Computational Color Constancy Algorithms—Part I: Methodology and Experiments With Synthesized Data. IEEE Transactions on Image Processing, volume 11 no. 9.

Barnard, K. Cardei, V. Funt, B. (2002b). A Comparison of Computational Color Constancy Algorithms—Part II: Experiments with Image Data. IEEE Transactions on Image Processing, volume 11 no. 9.

Bauer, B.O. (2009). Contemporary research in aeolian geomorphology. Geomorphology 105(1-2): 1-5.

Belly, P.Y. (1964). Sand Movement by Wind. Technical Memorandum 1, US Army Corps of Engineers CERC: Vicksburg, MS; 38 pp.

Chapman, D.M. (1990). Aeolian sand transport - an optimized model. Earth Surface Processes and Landforms 15(8): 751-760.

Chepil, W. S. (1956). Influence of Moisture on Erodibility of Soil by Wind. Soil Sci Soc Am J 20(2): 288-292.

Ciani, A. Goss, K.U. Schwarzenbach, R.P. (2005). Light penetration in soil and particulate minerals. European Journal ofSoil Science 56, 561–574.

Cohen, J. Cohen, P, West, S. G. Aiken, L. S. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Cornelis, W.M. Gabriels, D. (2003). The effect of surface moisture on the entrainment of dune sand by wind: an evaluation of selected models. Sedimentology 50(4): 771-790.

Darke, I. Davidson-Arnott, R. Ollerhead, J. (2009). Measurement of Beach Surface Moisture Using Surface Brightness. Journal of Coastal Research 25(1): 248-256.

Darke, I. Neuman, C.M. (2008). Field Study of Beach Water Content as a Guide to Wind Erosion Potential. Journal of Coastal Research 24(5): 1200-1208.

Davidson-Arnott, R.G.D. MacQuarrie, K. Aagaard, T (2005). The effect of wind gusts, moisture content and fetch length on sand transport on a beach. Geomorphology 68(1-2): 115-129.

Delta-T Devices. (1999). Theta Probe Soil Moisture Sensor Type ML2x User Manual. Cambridge: Delta-T Devices, Ltd.

De Meijer, R.J. 1997. Heavy minerals: from 'Edelstein' to Einstein. Journal of Geochemical Exploration 62 (1998) 81–103

Ebner, M. (2007). Color Constancy. John Wiley & Sons Ltd. Chichester, England

Edwards, B.L. Namikas, S L. (2009). Small-scale variability in surface moisture on a fine-grained beach: implications for modeling aeolian transport. Earth Surface Processes and Landforms 34(10): 1333-1338.

Finlayson, G.D. Schiele, B. Crowley, J.L. (1998). Comprehensive color image normalization. ECCV'98 p. 475-490, 1998

Hordley, S.D. Finlayson, G.D. Schaefer, G. Tian, G.Y. (2002). Illuminant and Device Invariant Color Using Histogram Equalization. UEA Technical Report No. SYS-C02-16

Hunt, R.W.G. (1991). Measuring Color (2th ed.). Ellis Horwood Limited. Chichester, England.

Kaleita, A.L. Tian, L.F. Hirschi, M.C. (2005). Relationship Between Soil Moisture Content and Surface Reflectance. American Society of Agricultural Engineers. Vol. 48(5): 1979−1986

Koomans, R.L. (2000). Sand in motion: effects of density and grain size. PhD-thesis, Rijksuniversiteit Groningen, Netherlands

Miles, J. Shevlin, M. (2001). Applying regression & correlation. London: Sage

Muller, E. Décamps, H. (2001). Modeling soil moisture−reflectance. Remote Sensing Environ. 76(2): 173−180.

Nickling, W.G. Neuman, C.M. (2009). Aeolian Sediment Transport. Geomorphology of Desert Environments: 517-555.

Neuman, C.M. Langston, G. (2006). Measurement of water content as a control of particle entrainment by wind. Earth Surface Processes and Landforms 31(3): 303-317.

Prandtl, L. (1935). The mechanics of viscous fluids. In: F. Durand (ed.), Aerodynamic theory. Volume III, Berlin, Springer, pp. 57–109.

Shao, Y. (2000). Physics and Modelling of Wind Erosion, Kluwer Academic Publishers, Dordrecht, 393 pp.

Sherman, D.J. Bauer, B.O. (1993). Dynamics of beach-dune systems. Progress in Physical Geography 17(4): 413-447.

Somers, B. Gysels, V. Verstraeten, W. Delalieux, S. Coppin, P. (2010). Modelling moisture-induced soil reflectance changes in cultivated sandy soils: a case study in citrus orchards. British Society of Soil Science, European Journal of Soil Science, 61, 1091–1105

Weidong, L. Baret, F. Xingfa, G. Qingxi, T. Lanfen, Z. Bing, Z. (2001). Relating soil surface moisture to reflectance. Remote Sensing of Environment 81 (2002) 238– 246

Wiggs, G.F.S. Baird, A.J. Atherton, R.J. (2004). The dynamic effects of moisture on the entrainment and transport of sand by wind. Geomorphology 59(1-4): 13-30.

Yang, Y. Davidson-Arnott, R.G.D. (2005). Rapid Measurement of Surface Moisture Content on a Beach. Journal of Coastal Research 21(3): 447-452

Annex A – Normalisation.java


```
1  // JavaDoc: javadoc -private -d docs Normalisation.java
2  import java.awt.*;
3  import java.awt.color.*;
4  import java.awt.image.*;
5  import java.text.*;
6
7  /**
8   * Class for normalizing images.
9   */
10 public class Normalisation
11 {
12     public static final int RED = 0;
13     public static final int GREEN = 1;
14     public static final int BLUE = 2;
15
16     /**
17      * Colors of moisture images
18      */
19     private int[][] colors = {  {4,  0xfafad2},     // W <= 4%  : light goldenyellow
20                                 {8,  0xffff00},     // W <= 8%  : yellow
21                                 {12, 0xadff2f},     // W <= 12% : greenyellow
22                                 {16, 0x00ff00},     // W <= 16% : green
23                                 {20, 0x00ffff},     // W <= 20% : cyan
24                                 {24, 0x0000ff},     // W <= 24% : blue
25                                 {28, 0x00008b}  };  // W <= 28% : darkblue
26
27     /**
28      * RGB to XYZ conversion matrix
29      */
30     private double[][] M   = {{0.4124, 0.3576,  0.1805},
31                               {0.2126, 0.7152,  0.0722},
32                               {0.0193, 0.1192,  0.9505}};
33
34     /**
35      * XYZ to RGB conversion matrix
36      */
37     private double[][] Mi  = {{ 3.2406, -1.5372, -0.4986},
38                               {-0.9689,  1.8758,  0.0415},
39                               { 0.0557, -0.2040,  1.0570}};
40     /**
41      * Image width
42      */
43     private int width;
44     /**
45      * Inage height
46      */
47     private int height;
48     /**
49      * Normalization application
50      */
51     private GreyboardNorm norm;
52
53     /**
54      * Public constructor.
55      * @param  norm Normalization application
56      */
57     public Normalisation(GreyboardNorm norm)
58     {
59         this.norm = norm;
60     }
61
62     /**
63      * Convert sRGB image to grayscale image.
64      * @return Grayscale image
65      * @param  img sRGB image
66      */
67     public BufferedImage sRGBtoGrey (BufferedImage img)
68     {
69         if (img.getType() == BufferedImage.TYPE_BYTE_GRAY)
70             return img;
71
```

```
72          BufferedImage gray = new BufferedImage(img.getWidth(), img.getHeight(), BufferedImage.T
73          WritableRaster raster = gray.getRaster();
74
75          for (int x=0; x<raster.getWidth(); x++)
76          {
77              for (int y=0; y<raster.getHeight(); y++)
78              {
79                  int argb = img.getRGB(x,y);
80                  int r = (argb >> 16) & 0xff;
81                  int g = (argb >>  8) & 0xff;
82                  int b = (argb      ) & 0xff;
83
84                  // luminosity method
85                  int l = (int) Math.round(0.299 * r + 0.587 * g + 0.114 * b);
86                  raster.setSample(x, y, 0, l);
87              }
88          }
89          return gray;
90      }
91
92      /**
93       * Normalize image with Device Normalisation.
94       * @return Normalized image
95       * @param  img Grayscale image
96       * @param  ref matrix [width] [height] with reference pixels
97       * @throws IllegalArgumentException if no grayscale image or reference error
98       */
99      public BufferedImage deviceNormalize (BufferedImage img, double[][] ref)
100     {
101         if (img.getType() != BufferedImage.TYPE_BYTE_GRAY)
102             throw new IllegalArgumentException("No grayscale image");
103         if (ref == null)
104             throw new IllegalArgumentException("Reference error");
105         if (!(img.getWidth() == ref.length && img.getHeight() == ref[0].length))
106             throw new IllegalArgumentException("Reference error");
107
108         WritableRaster raster = img.getRaster();
109         width = raster.getWidth();
110         height = raster.getHeight();
111         int border = findBorder(raster);
112
113         // normalize image
114         double sumpixel = 0;
115         int npixel = 0;
116         for (int x=0; x<width; x++)
117         {
118             for (int y=0; y<height; y++)
119             {
120                 int pixel = raster.getSample(x, y, 0);
121                 if (y < border)
122                 {
123                     int newPixel = (int) Math.round(pixel * ref[x][y]);
124                     if (newPixel > 255)
125                     {
126                         newPixel = 255;
127                     }
128                     raster.setSample(x, y, 0, newPixel);
129                     sumpixel = sumpixel + newPixel;
130                     npixel++;
131                 }
132             }
133         }
134         DecimalFormat twoDec = new DecimalFormat("#.##");
135         double meanpixel = sumpixel / npixel;
136         norm.showStatus("Mean pixel: " + twoDec.format(meanpixel));
137         norm.showStatus("Standard deviation: " + twoDec.format(stdDeviation(img, meanpixel)));
138
139         return img;
140     }
141
142     /**
```

```
143       * Determine y-coordinate of black border of image, if any.
144       * @return y-coordinate of border, or height of image
145       * @param  raster Raster of image
146       **/
147      private int findBorder(WritableRaster raster)
148      {
149          int height = raster.getHeight();
150          int y;
151          for (y=0; y<height; y++)
152          {
153              int pixel = raster.getSample(0, y, 0);
154              if (pixel < 4)  // pixel = black (pixelvalue 0,1,2,3)
155              {
156                  int yb;
157                  for (yb=y; yb<height; yb++)
158                  {
159                      if (raster.getSample(0, yb, 0) > 3) // all pixels black?
160                          break;
161                  }
162                  if (yb == height)
163                      break;
164              }
165          }
166          return y;
167      }
168
169      /**
170       * Normalize image with Illuminant (Grayboard) Normalisation.
171       * @return Normalized image
172       * @param  img Grayscale image
173       * @param  xRef x-coordinate/imagewidth Reference point
174       * @param  yRef y-coordinate/imageheight Reference point
175       * @throws IllegalArgumentException if no grayscale image or reference error
176       */
177      public BufferedImage illuminantNormalize (BufferedImage img, double xRef, double yRef)
178      {
179          if (img.getType() != BufferedImage.TYPE_BYTE_GRAY)
180              throw new IllegalArgumentException("No grayscale image");
181          if (xRef <= 0 && yRef <= 0)
182              throw new IllegalArgumentException("Reference error");
183
184          WritableRaster raster = img.getRaster();
185          width = raster.getWidth();
186          height = raster.getHeight();
187          int border = findBorder(raster);
188          //System.out.println("border= "+ border);
189
190          // set graycard value
191          double grayCard = 118;
192
193          // calculate reference coordnates
194          int refx = (int) (xRef * width);
195          int refy = (int) (yRef * height);
196          norm.showStatus("Reference point: x=" +  refx + " y="+ refy);
197
198          // calculate mean grey value of 16 pixels from reference point
199          int sum = 0;
200          for (int x=0; x<4; x++)
201              for (int y=0; y<4; y++)
202              {
203                  //System.out.println(raster.getSample(refx+x, refy+y, 0));
204                  sum = sum + raster.getSample(refx+x, refy+y, 0);
205              }
206          double mean = Math.round(sum / 16.0);
207          norm.showStatus("Mean board color: " +  mean);
208
209          // calculate factor
210          if (mean == 0) mean = 1;
211          double factor = grayCard / mean;
212          norm.showStatus("Factor: " +  factor);
213
```

```
214          // normalize image
215          double sumpixel = 0;
216          int npixel = 0;
217          for (int x=0; x<width; x++)
218          {
219              for (int y=0; y<height; y++)
220              {
221                  int pixel = raster.getSample(x, y, 0);
222                  if (y < border)
223                  {
224                      int newPixel = (int) Math.round(pixel * factor);
225                      if (newPixel > 255)
226                      {
227                          newPixel = 255;
228                      }
229                      raster.setSample(x, y, 0, newPixel);
230                      sumpixel = sumpixel + newPixel;
231                      npixel++;
232                  }
233              }
234          }
235
236          // calculate mean and standard deviation of image pixels
237          double meanpixel = sumpixel / npixel;
238          DecimalFormat twoDec = new DecimalFormat("#.##");
239          norm.showStatus("Mean pixel: " + twoDec.format(meanpixel));
240          norm.showStatus("Standard deviation: " + twoDec.format(stdDeviation(img, meanpixel)));
241          return img;
242      }
243
244      /**
245       * Calculate standard deviation of image pixels.
246       * @return Standard deviation
247       * @param  img Grayscale image
248       * @param  mean mean of image pixels
249       */
250      private double stdDeviation (BufferedImage img, double mean)
251      {
252          WritableRaster raster = img.getRaster();
253          width = raster.getWidth();
254          height = raster.getHeight();
255          int border = findBorder(raster);
256
257          // calculate variance
258          double sumpixel = 0;
259          int npixel = 0;
260          for (int x=0; x<width; x++)
261          {
262              for (int y=0; y<height; y++)
263              {
264                  if (y < border)
265                  {
266                      double pixel = raster.getSample(x, y, 0);
267                      sumpixel = sumpixel + Math.pow(pixel - mean, 2);
268                      npixel++;
269                  }
270              }
271          }
272          // return standard deviation
273          return Math.sqrt(sumpixel / (npixel - 1));
274      }
275
276      /**
277       * Make moisture image.
278       * @return Moisture image
279       * @param  img Normalized image
280       */
281      public BufferedImage makeMoistureImage(BufferedImage img)
282      {
283          BufferedImage result = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
284
```

```
285         WritableRaster raster = img.getRaster();
286         width = raster.getWidth();
287         height = raster.getHeight();
288         DecimalFormat twoDec = new DecimalFormat("#.##");
289
290         for (int x=0; x<width; x++)
291         {
292             for (int y=0; y<height; y++)
293             {
294                 double pixel = raster.getSample(x, y, 0);
295                 double W;
296
297                 pixel = 0.243 * pixel + 57.67;      // interpolate
298                 W = (pixel - 100.557) / -1.533;     // calculate moisture
299
300                 //norm.showStatus("pixel=" + twoDec.format(pixel) + " W=" + twoDec.format(W));
301
302                 if (W >= 0)
303                 {
304                     int color = getColor(W);
305                     if (color != -1)
306                         result.setRGB(x, y, color);
307                     else
308                         // W > 28
309                         result.setRGB(x, y, 0x000000);  // black
310                 }
311                 else
312                 {
313                     // W < 0
314                     result.setRGB(x, y, 0xffffff);  // white
315                 }
316             }
317         }
318         return result;
319     }
320
321     /**
322      * Get moisture color.
323      * @return Moisture color, or -1 if W > 28%
324      * @param  W Moisture (%)
325      */
326     private int getColor(double W)
327     {
328         int i;
329         for (i=0; i<colors.length; i++)
330         {
331             if (W <= colors[i][0])
332             {
333                 break;
334             }
335         }
336         if (i == colors.length)
337             return -1;
338         else
339             return colors[i][1];
340     }
341
342     /**
343      * Convert sRGB image to 3 pixel matrix.
344      * @return Pixel matrix [width * height][3]
345      * @param  img sRGB image
346      */
347     public double[][] sRGBtoMatrix (BufferedImage img)
348     {
349         width = img.getWidth();
350         height = img.getHeight();
351         int n = width * height;
352         double[][] matrix = new double[n][3];
353
354         int k = 0;
355         for (int i=0;i<width;i++)
```

```
356             for (int j=0;j<height;j++)
357             {
358                 int pixel = img.getRGB(i,j);
359                 int alpha = (pixel & 0xff000000) >>> 24;
360                 int red   = (pixel & 0x00ff0000) >>> 16;
361                 int green = (pixel & 0x0000ff00) >>> 8;
362                 int blue  = (pixel & 0x000000ff) >>> 0;
363                 //System.out.println("sRGB=" + red + " " + green + " " + blue);
364                 //compNorm.textArea.append(red + ";" + green + ";" + blue + "\n");
365 
366                 // transformeer naar linear RGB
367                 matrix[k] = sRGBtoRGB(red, green, blue);
368                 //System.out.println("RGB=" + matrix[k][RED] + " " + matrix[k][GREEN] + " " + m
369                 //compNorm.textArea.append(matrix[k][RED] + ";" + matrix[k][GREEN] + ";" + matr
370 
371                 /*
372                 // transformeer naar CIE XYZ
373                 matrix[k] = RGBtoXYZ(matrix[k]);
374                 //System.out.println("XYZ=" + matrix[k][RED] + " " + matrix[k][GREEN] + " " + m
375                 */
376 
377                 k++;
378             }
379         return matrix;
380     }
381 
382     /**
383      * Convert pixel matrix to sRGB image.
384      * @return sRGB image
385      * @param  matrix Pixel matrix [width * height][3]
386      */
387     public BufferedImage MatrixtosRGB (double[][] matrix)
388     {
389         BufferedImage result = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
390 
391         int[] rgb = null;
392         int k = 0;
393         for (int i=0;i<width;i++)
394             for (int j=0;j<height;j++)
395             {
396                 /*
397                 // transformeer naar RGB
398                 double[] RGB = XYZtoRGB(matrix[k]);
399                 // transformeer naar sRGB
400                 int[] rgb = RGBtosRGB(RGB[RED], RGB[GREEN], RGB[BLUE]);
401                 */
402 
403                 // transformeer naar sRGB
404                 rgb = RGBtosRGB(matrix[k][RED], matrix[k][GREEN], matrix[k][BLUE]);
405 
406                 int red = rgb[RED];
407                 int green = rgb[GREEN];
408                 int blue = rgb[BLUE];
409 
410                 //System.out.println(Integer.toHexString(red) + " " + Integer.toHexString(green
411                 //System.out.println(red + " " + green + " " + blue);
412 
413                 int argb = (int)((blue << 0) + (green << 8) + (red << 16) + (255 << 24));
414 
415                 result.setRGB(i,j, argb);
416 
417                 k++;
418             }
419         norm.showStatus("srgb[" + (k-1) + "] " + rgb[RED] + " " + rgb[GREEN] + " " + rgb[BLUE])
420 
421         return result;
422     }
423 
424     /**
425      * Normalize pixel matrix with Chromaticity normalisation.
426      * @return Normalized pixel matrix
```

```
427       * @param  matrix Pixel matrix [width * height][3]
428       */
429      public double[][] chromaticity (double[][] matrix)
430      {
431          double[][] result = new double[matrix.length][3];
432
433          for (int i=0;i<matrix.length;i++)
434          {
435              // calculate chromaticities
436              double sum = matrix[i][RED] + matrix[i][GREEN] + matrix[i][BLUE];
437              if (sum != 0)
438              {
439                  result[i][RED] = matrix[i][RED] / sum;
440                  result[i][GREEN] = matrix[i][GREEN] / sum;
441                  result[i][BLUE] = matrix[i][BLUE] / sum;
442              }
443              //System.out.println("rgb=" + result[i][RED] + " " + result[i][GREEN] + " " + resul
444              //compNorm.textArea.append(result[i][RED] + ";" + result[i][GREEN] + ";" + result[i
445          }
446          return result;
447      }
448
449      /**
450       * Normalize pixel matrix with Greyworld normalisation.
451       * @return Normalized pixel matrix
452       * @param  matrix Pixel matrix [width * height][3]
453       */
454      public double[][] greyworld (double[][] matrix)
455      {
456          double[][] result = new double[matrix.length][3];
457
458          double sumR = 0.0;
459          double sumG = 0.0;
460          double sumB = 0.0;
461
462          for (int i=0;i<matrix.length;i++)
463          {
464              // calculate sum R, sum G en sum B
465              sumR = sumR + matrix[i][RED];
466              sumG = sumG + matrix[i][GREEN];
467              sumB = sumB + matrix[i][BLUE];
468          }
469
470          //System.out.println("sumR=" + sumR + " sumG=" + sumG + " sumB=" + sumB);
471
472          // bereken gemiddelde R, gemiddelde G en gemiddelde B
473          double meanR = sumR / matrix.length;
474          double meanG = sumG / matrix.length;
475          double meanB = sumB / matrix.length;
476
477          //System.out.println("meanR=" + meanR + " meanG=" + meanG + " meanB=" + meanB);
478
479          for (int i=0;i<matrix.length;i++)
480          {
481              if (meanR != 0) result[i][RED] = matrix[i][RED] / meanR;
482              if (meanG != 0) result[i][GREEN] = matrix[i][GREEN] / meanG;
483              if (meanB != 0) result[i][BLUE] = matrix[i][BLUE] / meanB;
484
485              //System.out.println("rgb=" + result[i][RED] + " " + result[i][GREEN] + " " + resul
486              //compNorm.textArea.append(result[i][RED] + ";" + result[i][GREEN] + ";" + result[i
487
488          }
489          return result;
490      }
491
492      /**
493       * Convert sRGB pixel values to linear RGB pixel values.
494       * @return vector of 3 linear RGB pixel values
495       * @param  red Red sRGB pixel value
496       * @param  green Green sRGB pixel value
497       * @param  blue Blue sRGB pixel value
```

```
498       */
499      public double[] sRGBtoRGB(int red, int green, int blue)
500      {
501          double[] result = new double[3];
502
503          // convert 0 .. 255 to 0.0 .. 1.0
504          double r = red / 255.0;
505          double g = green / 255.0;
506          double b = blue / 255.0;
507
508          double R, G, B;
509
510          // convert sRGB to linear RGB
511          if (r <= 0.04045)
512          {
513              R = r / 12.92;
514          }
515          else
516          {
517              R = Math.pow(((r + 0.055) / 1.055), 2.4);
518          }
519
520          if (g <= 0.04045)
521          {
522              G = g / 12.92;
523          }
524          else
525          {
526              G = Math.pow(((g + 0.055) / 1.055), 2.4);
527          }
528
529          if (b <= 0.04045)
530          {
531              B = b / 12.92;
532          }
533          else
534          {
535              B = Math.pow(((b + 0.055) / 1.055), 2.4);
536          }
537
538          result[0] = R;
539          result[1] = G;
540          result[2] = B;
541
542          return result;
543      }
544
545      /**
546       * Convert linear RGB pixel values to sRGB pixel values.
547       * @return vector of 3 sRGB pixel values
548       * @param  R Red linear RGB pixel value
549       * @param  G Green linear RGB pixel value
550       * @param  B Blue linear RGB pixel value
551       */
552      public int[] RGBtosRGB(double R, double G, double B)
553      {
554          int[] result = new int[3];
555
556          double r, g, b;
557
558          // convert linear RGB to sRGB
559          if (R > 0.0031308)
560          {
561              r = ((1.055 * Math.pow(R, 1.0 / 2.4)) - 0.055);
562          }
563          else
564          {
565              r = (R * 12.92);
566          }
567
568          if (G > 0.0031308)
```

```
569          {
570              g = ((1.055 * Math.pow(G, 1.0 / 2.4)) - 0.055);
571          }
572          else
573          {
574              g = (G * 12.92);
575          }
576
577          if (B > 0.0031308)
578          {
579              b = ((1.055 * Math.pow(B, 1.0 / 2.4)) - 0.055);
580          }
581          else
582          {
583              b = (B * 12.92);
584          }
585
586          // be sure r, g, b are in interval 0.0 .. 1.0
587          if (r > 1)
588          {
589              r = 1.0/r * r;
590              g = 1.0/r * g;
591              b = 1.0/r * b;
592          }
593          if (g > 1)
594          {
595              r = 1.0/g * r;
596              g = 1.0/g * g;
597              b = 1.0/g * b;
598          }
599          if (b > 1)
600          {
601              r = 1.0/b * r;
602              g = 1.0/b * g;
603              b = 1.0/b * b;
604          }
605
606          // convert 0.0 .. 1.0 to 0 .. 255
607          result[RED] = (int) Math.round(r * 255.0);
608          result[GREEN] = (int) Math.round(g * 255.0);
609          result[BLUE] = (int) Math.round(b * 255.0);
610
611          //System.out.println("R=" + Integer.toHexString(result[RED]) + " G=" + Integer.toHexStr
612
613          return result;
614      }
615
616      /**
617       * Convert linear RGB pixel values to XYZ pixel values.
618       * @return vector of 3 XYZ pixel values
619       * @param  RGB vector of 3 linear RGB pixel values
620       */
621      public double[] RGBtoXYZ(double[] RGB)
622      {
623          double[] result = new double[3];
624
625          double r = RGB[RED];
626          double g = RGB[GREEN];
627          double b = RGB[BLUE];
628
629          // [X Y Z] = [r g b][M]
630          result[RED] = (r * M[0][0]) + (g * M[0][1]) + (b * M[0][2]);
631          result[GREEN] = (r * M[1][0]) + (g * M[1][1]) + (b * M[1][2]);
632          result[BLUE] = (r * M[2][0]) + (g * M[2][1]) + (b * M[2][2]);
633
634          return result;
635      }
636
637      /**
638       * Convert XYZ pixel values to linear RGB pixel values.
639       * @return vector of 3 linear RGB pixel values
```

```
640       * @param  XYZ vector of 3 XYZ pixel values
641       */
642      public double[] XYZtoRGB(double[] XYZ)
643      {
644        double[] result = new double[3];
645
646        double x = XYZ[RED];
647        double y = XYZ[GREEN];
648        double z = XYZ[BLUE];
649
650        // [r g b] = [X Y Z][Mi]
651        result[RED] = (x * Mi[0][0]) + (y * Mi[0][1]) + (z * Mi[0][2]);
652        result[GREEN] = (x * Mi[1][0]) + (y * Mi[1][1]) + (z * Mi[1][2]);
653        result[BLUE] = (x * Mi[2][0]) + (y * Mi[2][1]) + (z * Mi[2][2]);
654
655        return result;
656      }
657
658  }
659
```

Annex B – Field Data

Luminance Normalization

JPEG_NR	CROP_REF		Std.Dev	graycard_factor	CROP_SAND		Std.Dev	CORRECT_SAND		Microtome W (%)	Probe (Mv)	Probe *W* (%)	Probe W / Microtome W
44	148.257	0.581	3.382	0.793	157.535	0.618	15.642	124.959	0.490	0.21	44	14.45	69
45	146.717	0.575	3.083	0.802	119.180	0.467	30.995	95.528	0.375	4.20	58	16.49	4
46	196.507	0.771	2.960	0.598	161.339	0.633	25.768	96.554	0.379	4.24	58	16.57	4
47	176.508	0.692	2.269	0.666	142.017	0.557	15.764	94.620	0.371	3.29	53	15.79	5
48	172.665	0.677	2.123	0.681	125.808	0.493	14.937	85.686	0.336	3.51	53	15.76	4
49	172.686	0.677	2.125	0.681	125.813	0.493	14.933	85.679	0.336	0.04	23	11.37	280
50	144.564	0.567	2.648	0.813	76.812	0.301	13.491	62.485	0.245	7.80	55	16.05	2
60	211.596	0.830	2.499	0.556	143.630	0.563	7.563	79.826	0.313	22.88	633	100.90	4
61	212.072	0.832	2.396	0.555	148.961	0.584	8.086	82.603	0.324	22.57	630	100.48	4
62	234.882	0.921	2.453	0.501	187.083	0.734	21.593	93.668	0.367	12.91	186	35.39	3
63	219.073	0.859	2.275	0.537	160.377	0.629	9.348	86.092	0.338	17.96	364	61.49	3
64	239.890	0.941	1.741	0.490	200.643	0.787	44.783	98.360	0.386	7.82	189	35.75	5
65	212.237	0.832	3.312	0.554	156.150	0.612	33.412	86.522	0.339	7.35	156	30.99	4
66	197.625	0.775	3.098	0.595	173.695	0.681	16.008	103.360	0.405	3.17	102	23.01	7
68	223.648	0.877	2.429	0.526	164.125	0.644	20.261	86.301	0.338	5.17	101	22.79	4
69	246.944	0.968	1.436	0.476	236.913	0.929	6.850	112.823	0.442	0.07	61	16.96	244
70	158.569	0.622	2.479	0.742	126.695	0.497	21.173	93.961	0.368	5.13	77	19.38	4
71	190.509	0.747	5.067	0.617	118.716	0.466	57.294	73.283	0.287	4.75	107	23.67	5
72	169.454	0.665	3.059	0.694	143.797	0.564	12.015	99.794	0.391	4.33	79	19.61	5
78	146.254	0.574	2.490	0.804	103.642	0.406	15.398	83.337	0.327	9.24	180	34.48	4
79	118.449	0.465	2.258	0.993	91.967	0.361	5.990	91.308	0.358	2.92	128	26.88	9
80	233.498	0.916	1.868	0.504	175.099	0.687	13.810	88.188	0.346	24.69	640	102.01	4
81	236.045	0.926	1.757	0.498	181.999	0.714	14.050	90.674	0.356	23.95	652	103.79	4
82	234.687	0.920	1.878	0.501	182.915	0.717	15.075	91.657	0.359	23.31	647	102.94	4
83	235.888	0.925	1.696	0.499	182.658	0.716	11.800	91.063	0.357	21.65	627	100.10	5
84	235.743	0.924	1.795	0.499	180.815	0.709	11.532	90.199	0.354	20.09	618	98.75	5
85	236.520	0.928	1.749	0.497	181.419	0.711	12.150	90.203	0.354	19.17	568	91.36	5
86	239.007	0.937	2.159	0.492	191.960	0.753	16.404	94.451	0.370	14.48	529	85.63	6
87	238.633	0.936	2.160	0.493	194.321	0.762	18.050	95.763	0.376	11.96	366	61.69	5
88	237.632	0.932	2.314	0.495	205.371	0.805	24.159	101.635	0.399	9.00	182	34.74	4
89	236.766	0.928	1.830	0.497	203.803	0.799	25.008	101.228	0.397	8.07	195	36.65	5
90	232.885	0.913	1.997	0.505	180.072	0.706	44.448	90.931	0.357	4.63	129	26.98	6
91	233.474	0.916	1.997	0.504	199.107	0.781	36.353	100.289	0.393	4.75	109	23.99	5
92	229.506	0.900	2.117	0.512	186.223	0.730	56.400	95.422	0.374	3.22	138	28.30	9

93	219.288	0.860	2.717	0.536	170.747	0.670	59.512	91.568	0.359	5.26	133	27.52	5
94	219.524	0.861	2.962	0.536	202.073	0.792	21.016	108.251	0.425	2.62	70	18.28	7
95	205.440	0.806	3.056	0.572	187.530	0.735	28.837	107.348	0.421	2.44	73	18.74	8
96	206.613	0.810	5.314	0.569	189.032	0.741	23.890	107.593	0.422	2.95	83	20.14	7
97	204.574	0.802	4.831	0.575	169.967	0.667	30.533	97.706	0.383	4.67	109	23.98	5
98	195.268	0.766	5.395	0.602	183.236	0.719	33.718	110.354	0.433	3.33	56	16.20	5
99	191.961	0.753	5.624	0.613	195.410	0.766	22.074	119.713	0.469	0.63	40	13.95	22
100	174.182	0.683	6.893	0.675	137.884	0.541	55.942	93.093	0.365	2.80	96	22.15	8
101	191.913	0.753	5.503	0.613	153.134	0.601	53.569	93.837	0.368	2.71	66	17.65	7
110	227.342	0.892	2.073	0.517	167.477	0.657	15.760	86.633	0.340	22.46	613	98.00	4
111	228.774	0.897	2.483	0.514	171.583	0.673	15.713	88.201	0.346	21.03	604	96.74	5
112	239.118	0.938	2.035	0.492	180.945	0.710	16.175	88.990	0.349	21.72	614	98.22	5
113	238.652	0.936	1.843	0.493	179.257	0.703	14.982	88.332	0.346	18.86	614	98.09	5
114	236.734	0.928	1.813	0.497	178.608	0.700	13.570	88.725	0.348	19.75	586	94.02	5
115	181.751	0.713	3.443	0.647	117.396	0.460	8.671	75.960	0.298	15.64	512	83.12	5
116	194.811	0.764	3.675	0.604	127.197	0.499	9.294	76.784	0.301	14.42	487	79.50	6
117	184.340	0.723	3.388	0.638	113.310	0.444	8.299	72.286	0.283	13.59	453	74.52	5
118	236.929	0.929	1.711	0.496	185.759	0.728	14.856	92.202	0.362	12.75	341	58.05	5
119	186.503	0.731	5.100	0.631	187.940	0.737	10.641	118.506	0.465	0.05	45	14.61	287
120	170.172	0.667	3.756	0.691	164.165	0.644	10.442	113.449	0.445	0.06	56	16.21	278
121	216.235	0.848	5.014	0.544	218.376	0.856	5.776	118.764	0.466	0.15	64	17.44	119
122	151.716	0.595	2.702	0.775	135.250	0.530	9.699	104.837	0.411	0.37	39	13.78	38
123	144.061	0.565	2.040	0.816	126.250	0.495	7.573	103.061	0.404	0.24	45	14.61	62
125	154.178	0.605	2.320	0.763	121.157	0.475	8.712	92.413	0.362	4.17	119	25.51	6
126	138.577	0.543	2.068	0.849	123.952	0.486	7.031	105.189	0.413	0.19	65	17.50	93
127	123.869	0.486	2.330	0.949	97.582	0.383	6.146	92.643	0.363	3.07	78	19.52	6
128	138.799	0.544	2.684	0.847	119.398	0.468	8.000	101.162	0.397	0.17	32	12.77	74
129	144.442	0.566	2.145	0.814	100.277	0.393	9.519	81.642	0.320	2.87	58	16.54	6
130	161.192	0.632	2.192	0.730	123.988	0.486	9.124	90.457	0.355	2.18	43	14.40	7
131	161.839	0.635	2.479	0.727	106.628	0.418	15.686	77.481	0.304	3.58	69	18.15	5
132	162.810	0.638	2.529	0.722	119.742	0.470	11.122	86.491	0.339	3.81	64	17.37	5
133	154.769	0.607	2.168	0.760	115.506	0.453	11.826	87.766	0.344	3.37	62	17.16	5
134	129.096	0.506	1.981	0.911	115.531	0.453	7.006	105.243	0.413	0.06	25	11.65	184
135	129.798	0.509	2.052	0.906	117.298	0.460	6.434	106.275	0.417	0.07	36	13.26	183
136	120.309	0.472	2.343	0.977	78.065	0.306	7.260	76.307	0.299	2.22	41	14.01	6
137	119.673	0.469	2.092	0.983	106.591	0.418	6.774	104.745	0.411	0.06	35	13.16	214
138	117.491	0.461	2.131	1.001	88.102	0.345	7.081	88.184	0.346	2.94	53	15.76	5
139	93.824	0.368	2.267	1.253	44.580	0.175	4.798	55.877	0.219	14.17	236	42.64	3
140	99.905	0.392	2.326	1.177	47.108	0.185	4.926	55.451	0.217	14.82	277	48.66	3
141	105.416	0.413	2.125	1.116	51.697	0.203	5.321	57.672	0.226	13.99	249	44.58	3
142	121.474	0.476	2.039	0.968	80.239	0.315	8.128	77.680	0.305	7.24	79	19.56	3
143	118.603	0.465	2.256	0.992	64.646	0.254	7.023	64.099	0.251	11.15	196	36.85	3
144	99.224	0.389	2.172	1.185	48.749	0.191	7.174	57.777	0.227	11.33	211	39.06	3

145	185.236	0.726	2.524	0.635	151.542	0.594	16.124	96.209	0.377	0.91	50	15.36	17
146	193.976	0.761	2.657	0.606	174.375	0.684	12.171	105.717	0.415	0.94	49	15.25	16
147	176.785	0.693	2.733	0.665	133.308	0.523	8.203	88.678	0.348	5.36	192	36.16	7
148	186.816	0.733	2.743	0.629	146.918	0.576	8.325	92.484	0.363	3.86	126	26.57	7
149	189.052	0.741	2.844	0.622	157.909	0.619	8.907	98.227	0.385	2.74	59	16.70	6
150	204.949	0.804	2.832	0.574	139.945	0.549	14.962	80.301	0.315	1.90	53	15.76	8
151	207.359	0.813	2.742	0.567	184.559	0.724	13.121	104.669	0.410	0.37	47	14.89	40
152	214.198	0.840	2.272	0.549	194.132	0.761	11.258	106.583	0.418	0.28	46	14.83	53
153	205.353	0.805	2.775	0.573	137.608	0.540	13.142	78.804	0.309	1.60	47	14.89	9
154	200.847	0.788	2.460	0.586	118.289	0.464	13.643	69.261	0.272	2.78	98	22.48	8
155	194.525	0.763	2.991	0.605	117.032	0.459	15.375	70.752	0.277	3.18	95	21.92	7
156	196.894	0.772	3.311	0.597	103.458	0.406	18.668	61.793	0.242	3.95	155	30.79	8
157	227.073	0.890	2.442	0.518	208.555	0.818	13.682	108.010	0.424	0.16	36	13.24	81
158	229.875	0.901	3.001	0.512	204.583	0.802	16.602	104.661	0.410	2.32	44	14.48	6
159	225.684	0.885	2.731	0.521	183.831	0.721	16.704	95.791	0.376	0.12	69	18.12	150
160	225.639	0.885	1.944	0.521	157.527	0.618	24.058	82.101	0.322	2.47	74	18.89	8
161	227.074	0.890	2.028	0.518	219.342	0.860	12.032	113.596	0.445	0.08	71	18.43	231
162	211.101	0.828	1.864	0.557	161.212	0.632	16.596	89.808	0.352	2.60	74	18.87	7
163	199.407	0.782	1.926	0.590	197.321	0.774	15.915	116.370	0.456	0.08	36	13.24	165
176	182.204	0.715	2.610	0.645	106.286	0.417	6.247	68.600	0.269	21.96	609	97.41	4
177	184.871	0.725	2.820	0.636	110.021	0.431	6.310	69.986	0.274	24.40	576	92.65	4
178	185.498	0.727	3.021	0.634	114.001	0.447	7.173	72.273	0.283	16.43	469	76.89	5
179	181.481	0.712	3.112	0.648	112.094	0.440	9.368	72.637	0.285	17.24	442	72.91	4
180	177.194	0.695	3.125	0.664	131.214	0.515	13.184	87.084	0.342	10.53	300	52.08	5
181	172.183	0.675	3.342	0.683	124.196	0.487	16.058	84.825	0.333	13.54	282	49.37	4
182	169.588	0.665	2.681	0.693	133.670	0.524	12.562	92.693	0.364	7.85	241	43.34	6
183	161.411	0.633	2.295	0.729	128.342	0.503	11.541	93.507	0.367	5.22	119	25.51	5
184	157.986	0.620	2.429	0.744	126.851	0.497	10.412	94.424	0.370	4.34	89	21.14	5
185	152.473	0.598	2.393	0.771	122.693	0.481	9.552	94.631	0.371	3.92	61	16.96	4
186	160.168	0.628	2.371	0.734	135.549	0.532	12.580	99.524	0.390	3.74	70	18.33	5
187	157.354	0.617	2.285	0.747	131.266	0.515	13.523	98.103	0.385	3.64	102	22.95	6
188	152.905	0.600	2.475	0.769	126.294	0.495	13.621	97.133	0.381	4.00	101	22.87	6
189	134.530	0.528	2.174	0.874	97.607	0.383	14.589	85.323	0.335	7.39	124	26.18	4
190	122.466	0.480	2.428	0.960	81.156	0.318	14.016	77.932	0.306	9.79	200	37.45	4
191	125.040	0.490	2.437	0.940	78.774	0.309	11.717	74.087	0.291	11.99	217	39.88	3
192	119.466	0.468	2.535	0.984	71.370	0.280	6.751	70.255	0.276	12.47	284	49.76	4
193	110.697	0.434	2.248	1.062	62.819	0.246	6.636	66.736	0.262	12.37	246	44.09	4
194	243.285	0.954	1.608	0.483	245.173	0.961	5.214	118.513	0.465	0.08	24	11.60	151
195	178.055	0.698	3.239	0.660	171.028	0.671	9.101	112.959	0.443	0.08	23	11.40	149
196	181.415	0.711	3.051	0.648	173.115	0.679	8.490	112.220	0.440	0.07	21	11.12	169
197	177.597	0.696	1.941	0.662	164.475	0.645	8.648	108.911	0.427	0.07	22	11.21	151
198	180.984	0.710	2.630	0.650	168.992	0.663	9.366	109.808	0.431	0.08	26	11.87	155
199	158.431	0.621	2.917	0.742	148.951	0.584	8.737	110.563	0.434	0.07	24	11.51	160

200	183.531	0.720	2.392	0.641	140.179	0.550	8.775	89.822	0.352	4.50	56	16.31	4
201	161.200	0.632	2.026	0.730	109.912	0.431	11.530	80.184	0.314	4.56	68	17.96	4
202	232.464	0.912	2.070	0.506	196.699	0.771	10.930	99.507	0.390	4.33	74	18.92	4
203	192.713	0.756	4.076	0.610	135.313	0.531	13.750	82.573	0.324	4.66	67	17.91	4
204	228.553	0.896	2.702	0.515	191.577	0.751	21.033	98.574	0.387	4.36	66	17.65	4
205	174.745	0.685	1.949	0.673	114.242	0.448	13.883	76.883	0.302	5.57	65	17.55	3
206	185.868	0.729	2.852	0.633	148.438	0.582	15.826	93.918	0.368	4.06	54	15.98	4
207	165.411	0.649	2.298	0.711	99.537	0.390	10.551	70.766	0.278	6.83	94	21.77	3
208	157.948	0.619	2.098	0.745	95.901	0.376	13.034	71.403	0.280	6.55	80	19.75	3
209	137.265	0.538	2.447	0.857	81.029	0.318	10.531	69.421	0.272	7.06	85	20.50	3
210	113.914	0.447	2.427	1.032	62.005	0.243	8.730	64.012	0.251	4.95	50	15.40	3
211	147.105	0.577	2.437	0.799	77.082	0.302	11.298	61.621	0.242	4.42	54	15.89	4
212	127.028	0.498	2.119	0.926	69.047	0.271	10.099	63.922	0.251	4.05	61	16.96	4
213	95.055	0.373	2.222	1.237	61.935	0.243	9.055	76.624	0.300	3.89	67	17.84	5
214	100.784	0.395	2.240	1.167	42.188	0.165	4.575	49.227	0.193	26.32	648	103.17	4
	181.284	av.std.	2.639			av.std	15.175						

sunny

Theta probe

JPEG	Mv									Gem
44	40	46	40	41	47	47	41	46	46	44
45	51	55	57	67	58	55	57	61	58	58
46	54	60	53	59	60	60	56	64	58	58
47	52	53	54	54	53	52	52	54	52	53
48	51	52	53	52	51	53	53	55	54	53
49	22	23	23	22	22	23	22	25	23	23
50	55	53	57	53	56	51	54	53	60	55
60	646	644	610	647	649	630	614	624	630	633
61	638	651	659	626	619	638	606	580	651	630
62	184	206	209	185	175	212	172	179	156	186
63	258	338	267	437	411	354	399	430	384	364
64	186	179	188	163	199	200	186	185	214	189
65	151	159	135	163	161	155	176	170	138	156
66	96	102	97	101	110	102	103	101	107	102
68	85	107	109	107	91	95	93	100	118	101
69	35	35	33	89	76	43	90	68	79	61
70	77	76	77	78	83	84	75	72	74	77
71	108	113	127	84	114	110	90	98	115	107
72	80	80	79	78	81	84	73	77	78	79
78	186	245	149	205	208	115	180	197	137	180
79	118	150	135	126	113	133	126	130	125	128
80	620	637	653	639	644	649	643	639	638	640
81	640	660	658	648	652	657	648	655	653	652
82	641	643	646	653	646	649	654	636	651	647
83	606	632	637	629	636	630	631	625	619	627
84	610	614	621	613	617	620	618	624	625	618
85	508	598	584	557	537	593	561	597	574	568
86	495	590	472	520	512	564	498	537	570	529
87	336	330	381	381	349	418	335	408	352	366
88	179	175	201	190	169	170	186	190	178	182
89	174	214	224	190	195	210	163	193	192	195
90	113	114	137	140	138	138	127	139	116	129
91	102	110	122	105	112	94	109	110	115	109
92	134	132	123	157	130	157	143	138	129	138
93	122	120	143	132	113	131	145	131	158	133
94	59	60	71	71	73	73	77	77	68	70
95	72	73	70	78	73	70	76	76	69	73
96	79	82	85	72	85	80	85	90	85	83
97	111	99	101	121	106	104	112	116	108	109
98	56	60	52	56	57	56	52	55	57	56
99	41	40	40	37	43	43	39	40	40	40
100	89	106	100	89	101	97	100	99	85	96
101	82	90	83	76	53	52	53	50	51	66
110	604	627	595	613	629	624	595	589	640	613
111	590	612	618	606	589	594	598	625	607	604
112	604	623	633	594	631	626	607	609	603	614
113	616	611	604	619	606	611	620	619	616	614
114	592	567	573	590	593	591	597	578	591	586
115	515	513	530	508	523	472	516	530	497	512
116	479	440	518	480	500	497	517	509	442	487
117	403	452	448	442	471	489	438	470	464	453

118	415	223	293	370	346	317	414	349	340	341
119	38	37	49	49	49	39	46	50	47	45
120	54	53	56	60	53	58	58	54	56	56
121	53	59	77	65	66	70	59	71	57	64
122	35	43	35	39	40	40	38	41	42	39
123	42	46	45	47	47	43	43	45	46	45
125	120	140	115	137	136	101	107	130	86	119
126	64	63	59	67	64	72	60	58	74	65
127	80	83	85	72	80	75	75	78	77	78
128	28	32	36	34	30	35	31	33	32	32
129	59	58	55	58	60	58	60	57	57	58
130	41	45	45	43	44	43	45	43	42	43
131	64	76	80	62	59	85	62	64	69	69
132	64	65	62	67	68	62	66	62	57	64
133	61	60	60	69	64	58	65	64	59	62
134	23	21	22	22	24	26	24	28	32	25
135	33	35	36	34	37	37	35	36	38	36
136	38	41	41	40	43	41	41	42	40	41
137	30	34	37	32	36	39	35	36	36	35
138	51	51	54	50	53	53	55	54	53	53
139	227	257	240	248	226	234	235	251	204	236
140	253	288	293	318	286	250	260	291	252	277
141	234	246	258	228	240	271	259	268	237	249
142	69	79	75	77	82	80	85	80	80	79
143	190	200	218	179	202	211	183	187	197	196
144	209	199	211	187	217	222	214	234	210	211
145	46	54	53	45	53	52	46	50	51	50
146	51	47	49	54	51	45	54	43	49	49
147	173	193	202	184	200	216	160	175	222	192
148	114	130	136	114	111	134	132	126	140	126
149	59	63	63	54	58	58	56	60	61	59
150	51	50	50	55	52	49	58	57	52	53
151	46	47	46	47	45	54	45	41	50	47
152	40	47	44	43	46	47	51	49	50	46
153	46	48	50	44	47	48	47	44	47	47
154	85	93	79	98	112	101	97	108	113	98
155	96	102	112	102	86	81	87	92	94	95
156	144	161	170	155	165	155	161	142	143	155
157	33	36	38	36	36	35	37	33	36	36
158	42	45	44	43	45	44	45	44	44	44
159	59	62	60	67	63	67	81	81	79	69
160	62	69	79	74	71	77	78	77	79	74
161	63	72	70	71	69	67	78	79	69	71
162	77	76	80	73	79	84	63	63	70	74
163	32	35	36	39	36	37	36	30	39	36
176	593	628	610	631	535	643	644	602	594	609
177	624	590	495	630	599	541	633	545	531	576
178	489	463	491	442	456	524	419	415	523	469
179	452	436	431	446	444	461	421	430	457	442
180	282	297	309	290	313	306	283	306	315	300
181	281	275	293	285	284	300	260	274	283	282
182	225	204	245	265	266	252	233	239	236	241
183	103	123	119	115	125	128	106	121	132	119
184	80	91	88	94	92	96	86	89	88	89
185	58	60	60	58	64	62	62	63	61	61

186	65	67	69	69	70	73	68	74	77	70
187	101	99	107	100	98	114	109	83	104	102
188	93	106	94	108	101	98	91	115	104	101
189	127	133	119	144	117	118	134	109	112	124
190	211	198	185	215	199	198	189	206	203	200
191	223	224	235	207	232	216	203	200	213	217
192	274	281	241	301	288	277	306	295	296	284
193	270	251	227	256	249	273	210	207	268	246
194	24	25	27	22	25	25	23	24	24	24
195	23	21	21	29	22	20	24	25	22	23
196	21	20	20	22	21	21	22	22	21	21
197	21	21	22	22	22	21	22	22	22	22
198	31	26	24	27	26	24	26	26	26	26
199	26	23	24	24	23	23	23	24	24	24
200	52	56	51	58	56	54	62	61	58	56
201	62	69	63	68	71	65	74	71	66	68
202	67	75	72	70	76	75	79	78	76	74
203	65	68	68	67	67	67	69	67	68	67
204	54	59	60	67	66	69	69	74	72	66
205	66	66	65	66	66	63	63	66	63	65
206	45	51	53	58	53	52	71	54	51	54
207	76	83	90	104	98	101	96	97	98	94
208	75	75	72	82	91	90	76	80	78	80
209	75	81	86	80	88	86	88	94	87	85
210	51	54	54	46	51	49	49	49	49	50
211	53	53	49	57	54	55	59	50	52	54
212	51	59	62	63	66	63	58	61	65	61
213	56	68	73	68	70	62	70	66	69	67
214	639	647	628	649	657	658	644	652	659	648

Microtome Slice

JPEG	Keramic	WET	DRY	M(water)	M(sand)	Microtome *W*	Microtome *W*%	Microtome θ (%)	Probe (Mv)	Probe θ (%)	Probe W (%)	
44	22.877	58.283	58.209	0.0740	35.3320	0.0021	0.21	0.29	44	10.57	14.45	69.0
45	22.433	54.470	53.180	1.2900	30.7470	0.0420	4.20	5.74	58	12.06	16.49	3.9
46	25.217	45.725	44.890	0.8350	19.6730	0.0424	4.24	5.80	58	12.12	16.57	3.9
47	25.444	53.210	52.326	0.8840	26.8820	0.0329	3.29	4.50	53	11.55	15.79	4.8
48	25.829	58.255	57.156	1.0990	31.3270	0.0351	3.51	4.80	53	11.53	15.76	4.5
49	26.768	73.628	73.609	0.0190	46.8410	0.0004	0.04	0.06	23	8.32	11.37	280.3
50	22.879	63.356	60.427	2.9290	37.5480	0.0780	7.80	10.66	55	11.74	16.05	2.1
60	23.914	54.788	49.040	5.7480	25.1260	0.2288	22.88	31.27	633	73.82	100.90	4.4
61	23.593	63.449	56.109	7.3400	32.5160	0.2257	22.57	30.86	630	73.51	100.48	4.5
62	20.391	51.681	48.104	3.5770	27.7130	0.1291	12.91	17.64	186	25.90	35.39	2.7
63	23.143	52.722	48.218	4.5040	25.0750	0.1796	17.96	24.55	364	44.99	61.49	3.4
64	20.244	46.051	44.180	1.8710	23.9360	0.0782	7.82	10.69	189	26.16	35.75	4.6
65	26.242	62.442	59.964	2.4780	33.7220	0.0735	7.35	10.05	156	22.67	30.99	4.2
66	23.022	49.998	49.169	0.8290	26.1470	0.0317	3.17	4.33	102	16.84	23.01	7.3
68	23.030	54.648	53.095	1.5530	30.0650	0.0517	5.17	7.06	101	16.67	22.79	4.4
69	22.924	63.205	63.177	0.0280	40.2530	0.0007	0.07	0.10	61	12.41	16.96	243.9
70	22.244	53.152	51.644	1.5080	29.4000	0.0513	5.13	7.01	77	14.18	19.38	3.8
71	21.536	60.531	58.762	1.7690	37.2260	0.0475	4.75	6.50	107	17.32	23.67	5.0
72	24.614	51.152	50.051	1.1010	25.4370	0.0433	4.33	5.92	79	14.34	19.61	4.5
78	25.403	53.643	51.255	2.3880	25.8520	0.0924	9.24	12.63	180	25.23	34.48	3.7
79	22.514	52.896	52.035	0.8610	29.5210	0.0292	2.92	3.99	128	19.67	26.88	9.2
80	22.946	67.629	58.780	8.8490	35.8340	0.2469	24.69	33.76	640	74.63	102.01	4.1
81	20.265	61.776	53.755	8.0210	33.4900	0.2395	23.95	32.74	652	75.93	103.79	4.3
82	23.163	66.070	57.960	8.1100	34.7970	0.2331	23.31	31.86	647	75.31	102.94	4.4
83	20.411	61.551	54.230	7.3210	33.8190	0.2165	21.65	29.59	627	73.23	100.10	4.6
84	21.553	58.913	52.664	6.2490	31.1110	0.2009	20.09	27.46	618	72.24	98.75	4.9
85	23.040	56.334	50.979	5.3550	27.9390	0.1917	19.17	26.20	568	66.84	91.36	4.8
86	25.421	58.403	54.232	4.1710	28.8110	0.1448	14.48	19.79	529	62.65	85.63	5.9
87	26.262	56.973	53.693	3.2800	27.4310	0.1196	11.96	16.35	366	45.13	61.69	5.2
88	23.933	54.370	51.858	2.5120	27.9250	0.0900	9.00	12.30	182	25.42	34.74	3.9
89	23.613	50.963	48.920	2.0430	25.3070	0.0807	8.07	11.04	195	26.81	36.65	4.5
90	22.263	58.159	56.569	1.5900	34.3060	0.0463	4.63	6.34	129	19.74	26.98	5.8
91	23.052	53.989	52.586	1.4030	29.5340	0.0475	4.75	6.49	109	17.55	23.99	5.1
92	24.635	50.760	49.946	0.8140	25.3110	0.0322	3.22	4.40	138	20.70	28.30	8.8
93	22.536	52.241	50.756	1.4850	28.2200	0.0526	5.26	7.19	133	20.13	27.52	5.2
94	18.921	44.503	43.851	0.6520	24.9300	0.0262	2.62	3.58	70	13.38	18.28	7.0
95	26.397	58.187	57.429	0.7580	31.0320	0.0244	2.44	3.34	73	13.71	18.74	7.7
96	22.893	51.098	50.290	0.8080	27.3970	0.0295	2.95	4.03	83	14.74	20.14	6.8
97	22.451	57.108	55.563	1.5450	33.1120	0.0467	4.67	6.38	109	17.54	23.98	5.1

98	25.849	60.525	59.406	1.1190	33.5570	0.0333	3.33	4.56	56	11.85	16.20	4.9
99	26.784	60.165	59.957	0.2080	33.1730	0.0063	0.63	0.86	40	10.20	13.95	22.2
100	22.895	54.295	53.440	0.8550	30.5450	0.0280	2.80	3.83	96	16.21	22.15	7.9
101	25.462	57.782	56.930	0.8520	31.4680	0.0271	2.71	3.70	66	12.91	17.65	6.5
110	23.075	60.770	53.857	6.9130	30.7820	0.2246	22.46	30.70	613	71.70	98.00	4.4
111	21.263	57.324	51.058	6.2660	29.7950	0.2103	21.03	28.75	604	70.78	96.74	4.6
112	23.284	60.663	53.994	6.6690	30.7100	0.2172	21.72	29.69	614	71.86	98.22	4.5
113	22.458	59.657	53.755	5.9020	31.2970	0.1886	18.86	25.78	614	71.77	98.09	5.2
114	22.676	53.903	48.752	5.1510	26.0760	0.1975	19.75	27.00	586	68.78	94.02	4.8
115	24.994	59.677	54.985	4.6920	29.9910	0.1564	15.64	21.39	512	60.81	83.12	5.3
116	20.514	52.816	48.746	4.0700	28.2320	0.1442	14.42	19.71	487	58.16	79.50	5.5
117	22.685	54.770	50.931	3.8390	28.2460	0.1359	13.59	18.58	453	54.52	74.52	5.5
118	25.239	44.833	42.617	2.2160	17.3780	0.1275	12.75	17.43	341	42.47	58.05	4.6
119	25.448	64.695	64.675	0.0200	39.2270	0.0005	0.05	0.07	45	10.69	14.61	286.6
120	23.225	67.766	67.740	0.0260	44.5150	0.0006	0.06	0.08	56	11.86	16.21	277.6
121	22.990	62.476	62.418	0.0580	39.4280	0.0015	0.15	0.20	64	12.76	17.44	118.5
122	22.278	58.710	58.577	0.1330	36.2990	0.0037	0.37	0.50	39	10.08	13.78	37.6
123	26.415	67.027	66.931	0.0956	40.5164	0.0024	0.24	0.32	45	10.69	14.61	61.9
125	22.553	55.937	54.602	1.3350	32.0490	0.0417	4.17	5.69	119	18.66	25.51	6.1
126	26.804	62.506	62.439	0.0670	35.6350	0.0019	0.19	0.26	65	12.80	17.50	93.1
127	23.618	54.098	53.189	0.9090	29.5710	0.0307	3.07	4.20	78	14.28	19.52	6.4
128	22.295	66.374	66.298	0.0760	44.0030	0.0017	0.17	0.24	32	9.34	12.77	73.9
129	23.638	57.967	57.009	0.9580	33.3710	0.0287	2.87	3.92	58	12.10	16.54	5.8
130	26.431	65.089	64.264	0.8250	37.8330	0.0218	2.18	2.98	43	10.54	14.40	6.6
131	22.570	62.108	60.740	1.3680	38.1700	0.0358	3.58	4.90	69	13.28	18.15	5.1
132	24.047	58.916	57.636	1.2800	33.5890	0.0381	3.81	5.21	64	12.71	17.37	4.6
133	20.448	56.562	55.386	1.1760	34.9380	0.0337	3.37	4.60	62	12.55	17.16	5.1
134	20.335	66.247	66.218	0.0290	45.8830	0.0006	0.06	0.09	25	8.52	11.65	184.3
135	20.538	57.879	57.852	0.0270	37.3140	0.0007	0.07	0.10	36	9.70	13.26	183.3
136	22.920	64.790	63.882	0.9080	40.9620	0.0222	2.22	3.03	41	10.25	14.01	6.3
137	24.650	62.026	62.003	0.0230	37.3530	0.0006	0.06	0.08	35	9.63	13.16	213.8
138	25.018	58.343	57.392	0.9510	32.3740	0.0294	2.94	4.02	53	11.53	15.76	5.4
139	23.036	57.014	52.796	4.2180	29.7600	0.1417	14.17	19.38	236	31.19	42.64	3.0
140	23.240	57.328	52.927	4.4010	29.6870	0.1482	14.82	20.27	277	35.60	48.66	3.3
141	26.821	67.960	62.910	5.0500	36.0890	0.1399	13.99	19.13	249	32.61	44.58	3.2
142	25.867	59.373	57.112	2.2610	31.2450	0.0724	7.24	9.89	79	14.31	19.56	2.7
143	21.325	57.277	53.670	3.6070	32.3450	0.1115	11.15	15.24	196	26.96	36.85	3.3
144	25.461	57.699	54.418	3.2810	28.9570	0.1133	11.33	15.49	211	28.58	39.06	3.4
145	21.300	55.325	55.017	0.3080	33.7170	0.0091	0.91	1.25	50	11.24	15.36	16.8
146	20.558	55.326	55.001	0.3250	34.4430	0.0094	0.94	1.29	49	11.16	15.25	16.2
147	23.974	58.262	56.518	1.7440	32.5440	0.0536	5.36	7.33	192	26.46	36.16	6.7
148	23.005	53.354	52.225	1.1290	29.2200	0.0386	3.86	5.28	126	19.44	26.57	6.9
149	22.941	58.880	57.923	0.9570	34.9820	0.0274	2.74	3.74	59	12.22	16.70	6.1

150	25.034	68.577	67.764	0.8130	42.7300	0.0190	1.90	2.60	53	11.53	15.76	8.3
151	20.452	63.741	63.582	0.1590	43.1300	0.0037	0.37	0.50	47	10.90	14.89	40.4
152	23.220	57.097	57.003	0.0940	33.7830	0.0028	0.28	0.38	46	10.85	14.83	53.3
153	20.390	71.500	70.694	0.8060	50.3040	0.0160	1.60	2.19	47	10.90	14.89	9.3
154	22.709	75.333	73.912	1.4210	51.2030	0.0278	2.78	3.79	98	16.44	22.48	8.1
155	23.304	74.361	72.787	1.5740	49.4830	0.0318	3.18	4.35	95	16.04	21.92	6.9
156	25.886	71.307	69.583	1.7240	43.6970	0.0395	3.95	5.39	155	22.53	30.79	7.8
157	22.587	65.520	65.450	0.0700	42.8630	0.0016	0.16	0.22	36	9.69	13.24	81.1
158	26.451	61.640	60.841	0.7990	34.3900	0.0232	2.32	3.18	44	10.60	14.48	6.2
159	24.666	71.147	71.091	0.0560	46.4250	0.0012	0.12	0.16	69	13.26	18.12	150.2
160	22.306	67.317	66.234	1.0830	43.9280	0.0247	2.47	3.37	74	13.82	18.89	7.7
161	23.649	65.048	65.015	0.0330	41.3660	0.0008	0.08	0.11	71	13.48	18.43	231.0
162	26.840	64.858	63.896	0.9620	37.0560	0.0260	2.60	3.55	74	13.81	18.87	7.3
163	25.477	59.194	59.167	0.0270	33.6900	0.0008	0.08	0.11	36	9.69	13.24	165.3
176	23.585	64.004	56.726	7.2780	33.1410	0.2196	21.96	30.02	609	71.27	97.41	4.4
177	24.621	71.530	62.328	9.2020	37.7070	0.2440	24.40	33.36	576	67.78	92.65	3.8
178	20.571	53.065	48.479	4.5860	27.9080	0.1643	16.43	22.46	469	56.25	76.89	4.7
179	25.494	63.698	58.079	5.6190	32.5850	0.1724	17.24	23.57	442	53.34	72.91	4.2
180	22.890	52.892	50.033	2.8590	27.1430	0.1053	10.53	14.40	300	38.10	52.08	4.9
181	22.257	56.089	52.055	4.0340	29.7980	0.1354	13.54	18.51	282	36.12	49.37	3.6
182	25.835	58.883	56.478	2.4050	30.6430	0.0785	7.85	10.73	241	31.71	43.34	5.5
183	26.777	56.555	55.079	1.4760	28.3020	0.0522	5.22	7.13	119	18.66	25.51	4.9
184	20.327	52.667	51.323	1.3440	30.9960	0.0434	4.34	5.93	89	15.47	21.14	4.9
185	23.310	53.397	52.261	1.1360	28.9510	0.0392	3.92	5.36	61	12.41	16.96	4.3
186	23.920	52.973	51.925	1.0480	28.0050	0.0374	3.74	5.12	70	13.41	18.33	4.9
187	22.942	59.121	57.849	1.2720	34.9070	0.0364	3.64	4.98	102	16.79	22.95	6.3
188	23.149	52.220	51.101	1.1190	27.9520	0.0400	4.00	5.47	101	16.73	22.87	5.7
189	22.696	58.850	56.363	2.4870	33.6670	0.0739	7.39	10.10	124	19.15	26.18	3.5
190	25.002	62.792	59.423	3.3690	34.4210	0.0979	9.79	13.38	200	27.40	37.45	3.8
191	22.517	63.172	58.820	4.3520	36.3030	0.1199	11.99	16.39	217	29.18	39.88	3.3
192	25.412	60.733	56.817	3.9160	31.4050	0.1247	12.47	17.05	284	36.41	49.76	4.0
193	22.479	59.547	55.467	4.0800	32.9880	0.1237	12.37	16.91	246	32.26	44.09	3.6
194	22.713	67.130	67.096	0.0340	44.3830	0.0008	0.08	0.10	24	8.48	11.60	151.4
195	22.966	68.759	68.724	0.0350	45.7580	0.0008	0.08	0.10	23	8.34	11.40	149.1
196	24.650	70.215	70.185	0.0300	45.5350	0.0007	0.07	0.09	21	8.14	11.12	168.8
197	20.607	66.341	66.307	0.0340	45.7000	0.0007	0.07	0.10	22	8.20	11.21	150.6
198	20.338	67.352	67.316	0.0360	46.9780	0.0008	0.08	0.10	26	8.69	11.87	154.9
199	23.165	69.154	69.121	0.0330	45.9560	0.0007	0.07	0.10	24	8.42	11.51	160.4
200	23.624	57.152	55.709	1.4430	32.0850	0.0450	4.50	6.15	56	11.93	16.31	3.6
201	22.912	56.933	55.449	1.4840	32.5370	0.0456	4.56	6.23	68	13.14	17.96	3.9
202	26.804	58.802	57.473	1.3290	30.6690	0.0433	4.33	5.92	74	13.84	18.92	4.4
203	22.283	58.839	57.212	1.6270	34.9290	0.0466	4.66	6.37	67	13.10	17.91	3.8
204	22.512	56.242	54.834	1.4080	32.3220	0.0436	4.36	5.95	66	12.91	17.65	4.1

205	25.436	61.892	59.970	1.9220	34.5340	0.0557	5.57	7.61	65	12.84	17.55	3.2
206	25.018	59.290	57.953	1.3370	32.9350	0.0406	4.06	5.55	54	11.69	15.98	3.9
207	25.041	65.975	63.358	2.6170	38.3170	0.0683	6.83	9.34	94	15.93	21.77	3.2
208	26.812	64.507	62.189	2.3180	35.3770	0.0655	6.55	8.96	80	14.45	19.75	3.0
209	25.453	57.044	54.962	2.0820	29.5090	0.0706	7.06	9.64	85	15.00	20.50	2.9
210	22.725	59.467	57.734	1.7330	35.0090	0.0495	4.95	6.77	50	11.27	15.40	3.1
211	20.607	63.224	61.419	1.8050	40.8120	0.0442	4.42	6.05	54	11.62	15.89	3.6
212	22.974	63.110	61.548	1.5620	38.5740	0.0405	4.05	5.54	61	12.41	16.96	4.2
213	20.353	53.759	52.509	1.2500	32.1560	0.0389	3.89	5.31	67	13.06	17.84	4.6
214	24.666	66.354	57.668	8.6860	33.0020	0.2632	26.32	35.98	648	75.48	103.17	3.9

Calibration Theta Probe

M(gr) wet_total	M(kg) wet_sand	M(kg) dry_sand	M(kg) water	*W(%)*	θ(%)	Probe (Mv)	Probe (V)
2952.70	2.72625	2.05012	0.67613	32.98	24.13	296	0.296
2597.20	2.37075	2.05012	0.32063	15.64	11.44	59	0.059
3011.70	2.78525	2.05012	0.73513	35.86	26.24	304	0.304
2965.55	2.73910	2.05012	0.68898	33.61	24.59	314	0.314
2927.20	2.70075	2.05012	0.65063	31.74	23.22	162	0.162
2849.75	2.62330	2.05012	0.57318	27.96	20.46	128	0.128
2775.30	2.54885	2.05012	0.49873	24.33	17.80	105	0.105
2978.40	2.75195	2.05012	0.70183	34.23	25.05	290	0.29
2949.20	2.72275	2.05012	0.67263	32.81	24.01	281	0.281
2921.25	2.69480	2.05012	0.64468	31.45	23.01	238	0.238
2898.10	2.67165	2.05012	0.62153	30.32	22.18	186	0.186
2880.05	2.65360	2.05012	0.60348	29.44	21.54	139	0.139
2860.83	2.63438	2.05012	0.58426	28.50	20.85	135	0.135
2830.75	2.60430	2.05012	0.55418	27.03	19.78	134	0.134
2798.20	2.57175	2.05012	0.52163	25.44	18.62	126	0.126
2767.55	2.54110	2.05012	0.49098	23.95	17.52	111	0.111
2729.83	2.50338	2.05012	0.45326	22.11	16.18	91	0.091
2693.80	2.46735	2.05012	0.41723	20.35	14.89	94	0.094
2652.30	2.42585	2.05012	0.37573	18.33	13.41	82	0.082
2619.30	2.39285	2.05012	0.34273	16.72	12.23	58	0.058
2588.40	2.36195	2.05012	0.31183	15.21	11.13	45	0.045
2564.41	2.33796	2.05012	0.28784	14.04	10.27	38	0.038
2549.46	2.32301	2.05012	0.27289	13.31	9.74	35	0.035
2503.21	2.27676	2.05012	0.22664	11.05	8.09	20	0.02
2502.68	2.27623	2.05012	0.22611	11.03	8.07	19	0.019
2503.17	2.27672	2.05012	0.22660	11.05	8.09		