Automated Quantitative Measurements and Associated Error Covariances for Planetary Image Analysis.

P.D. Tar, N.A. Thacker, J.D. Gilmour, M.A. Jones.

Last updated
17 /5 / 2015

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.
Automated Quantitative Measurements and Associated Error Covariances for Planetary Image Analysis

P.D. Tar*

G.602 Imaging Science, Stopford Building, University of Manchester, Oxford Road, Manchester, UK

N.A. Thacker

Imaging Science, Stopford Building, University of Manchester, Oxford Road, Manchester, UK

J.D. Gilmour

School of Earth, Atmospheric and Environmental Sciences, University of Manchester, Oxford Road, Manchester, UK

M.A. Jones

School of Earth, Atmospheric and Environmental Sciences, University of Manchester, Oxford Road, Manchester, UK

Abstract

This paper presents a flexible approach for extracting measurements from planetary images based upon the newly developed linear Poisson models technique. The approach has the ability to learn surface textures then estimate the quantity of terrains exhibiting similar textures in new images. This approach is suitable for the estimation of dune field coverage or other repeating structures. Whilst other approaches exist, this method is unique for its incorporation of a comprehensive error theory, which includes contributions to uncertainty arising from training and subsequent use. The error theory is capable of producing measurement error covariances, which are essential for the scientific interpretation of measurements, i.e. for the plotting of error

*Corresponding author

Email address: paul.d.tar@gmail.com (P.D. Tar)
bars. In order to apply linear Poisson models, we demonstrate how terrains can be described using histograms created using a ‘Poisson blob’ image representation for capturing texture information. The validity of the method is corroborated using Monte Carlo simulations. The potential of the method is then demonstrated using terrain images created from bootstrap re-sampling of martian HiRISE data.

**Keywords:** Terrain; Texture; Likelihood; Statistics; Errors

1. Introduction

The analysis of planetary surface images represents an important aspect of many exploration missions. From the early 2000s there have been many high-profile missions launched within the inner solar system which have imaging capabilities. These include: Mars Odyssey (NASA, launched 2001); Mars Express [1] (ESA, launched 2003); Mars Reconnaissance Orbiter [2] (NASA, launched 2005); Mercury orbiter mission MESSENGER [3] (NASA, launched 2004); dual asteroid mission DAWN [4] to Ceres and Vesta (NASA, launched 2007); and the Moon mission Lunar Reconnaissance Orbiter [5] (NASA, launched 2009). These missions have provided planetary scientists with almost complete coverage of Mars, Vesta, Mercury and the Moon, including significant subsets with resolutions down to 0.25m.

Scientific applications for planetary images are numerous and varied. Most involve the identification and quantification of features of interest. For example, impact craters can provide a wealth of information regarding geological evolution. The total density of craters, as summarised using size-frequency distributions (SFD), can be used to infer relative surface ages [6]. There also is much interest in drainage systems on Mars, with recent seasonal flows being observed within the walls of some craters [7]. The distribution and orientation of ancient drainage systems can also be used to infer Mars’ climate history [8] and the timing of some geodynamic events [9]. Other example features are associated with an active carbon dioxide cycle on Mars, which is especially evident around the polar regions [10]. Seasonal eruptions of sublimating CO$_2$ causes dark patches to appear around dunes, radiating fissures to develop known as ‘spiders’ and geysers to eject fans of dark material. The study of these events can help researchers better understand this CO$_2$ cycle. Additionally, the density, morphology and orientation of dunes can be used to estimate the availability and size of grains [11], and also wind speed and
direction [12]. However, the growing quantity of image data is rapidly out-
pacing the resources available to individual researchers wishing to inspect
large numbers of images in detail.

Citizen science and automated methods have both been proposed to al-
leviate the problem. High-resolution images are being mapped with the aid
of large numbers of volunteers using web-based interfaces. These include the
mapping of small lunar craters as part of the MoonZoo [13] and Moon Mappers
projects [14], and the mapping of seasonal carbon dioxide ‘fans’ on Mars
via Planet Four [15]. Automated approaches to crater counting [16]- [26], and
valley and channel network mapping [27][28][29] have also been tested with
some success. Automated approaches tend to follow the same general design
pattern: raw image data is encoded using a higher-level descriptive format
(edge strings, Haar transform, texture descriptors, templates etc.), before
being fed into a classifier (decision trees [30], support vector machines [31],
boosting [32] etc) or searched for signal response peaks (Hough transforms,
template matches etc.) Performance is then evaluated empirically in terms of
numbers of correct versus incorrect classifications. The reported efficiencies
of these algorithms usually range between 60% to 80% for correct detec-
tions, with many false positives reported. These approaches tend to focus on
specific types of features, limiting their applicability in general terrain anal-
ysis tasks. They also involve a minimal amount of error analysis, which is
unfortunate given that quantitative scientific tasks must take careful consid-
eration of noise and uncertainty if data is not to be over-interpreted. Ideally,
all measurements should be presented with accompanying error information,
including honest assessments of statistical errors and any systematic biases.

Whilst under-represented in the literature, theoretical (predictive) meth-
ods of error analysis do exist, but require a good understanding of the inner-
workings of an algorithm, i.e. they represent white-box methods, as opposed
to black-box methods where the inner-workings of an algorithm are hidden
from their users. Advocates of a theoretical approach have analysed numer-
ous low-level algorithms using statistical perturbation models [33][34]. The
use of error propagation in computer vision [35]- [37] has been demonstrated
on tasks such as the extraction of 2D points and determining the accuracy
of parameters of fitted shapes [38]. It has also been applied to assess the
performance of multi-stage shape extraction from 2D projections [39], and in
the use of linear shape models [40]. A comprehensive example of theoretical
error analysis can also be found in [41], where propagated location uncer-
tainty assists in target recognition in image data. Error propagation has also
been used to investigate the effects of noise in Hough transforms [42] and also in iterative algorithms [43], such as expectation maximisation (EM).

We present a generic terrain analysis system, incorporating a detailed error theory for the prediction of statistical and systematic sources of uncertainty, allowing measured values to be used with confidence. An overview of the technique is given in section 2 before a detailed methodology in section 3. An outline of this method can be seen in the block diagram of figure 1. We test this system using a range of terrain images, including craters, dunes and CO2 cycle features in section 4, followed by a discussion of the method’s successes and limitations in section 5 before concluding.

2. Problem specification and definition of terms

We have selected the problem of making surface area measurements of user-defined martian terrains and to achieve levels of accuracy predicted by a theoretical error analysis. This problem is to be solved in an automated manner, making use of high resolution digital imagery. To ease understandability of the proposed solution we present an overview of the problem and notations used before a detailed methodology is provided.

Visually, a terrain can be approximated as a mixture of repeating pixel patterns, with different combinations of patterns giving rise to different textures. Additionally, within any class of terrain (e.g. dunes) there will typically be multiple ‘sub-textures’ (e.g. ripples at differing orientations). Any local arrangements of pixels can be encoded as a binary vector \( X = \{X_1, X_2, \ldots, X_l\} \), with elements \( X_i \in \{0, 1\} \), providing a maximum of \( m = 2^l \) discrete observable patterns. A range of potential image encodings is available in the computer vision literature such as BRIEF, which can record local bright-dark patterns using pixel pair comparisons [45]. The specific BRIEF-based encoding selected for this work is described in detail in section 3.1; for now it is sufficient to interpret this vector as an observation of a small image patch. Textures can thus be defined as different statistical distributions over the encoding \( X \), and natural number labels, \( K \in \{1, 2, \ldots, n\} \), which can be used to identify specific textures. A texture can then be described as a probability mass function (PMF), \( P(X = x | K = k) \), which gives the probability of observing any particular pattern realisation, \( x \), given a specific type of texture, \( k \). The training process through which PMFs are estimated is described in section 3.3. The overall distribution of patterns within an image can then be modelled using,
\[ H \approx M = PQ \]  

where \( H = \{H_1, H_2, \ldots, H_m\}^\top \) is a histogram over \( X \) containing sampled data from the image under analysis; \( M = \{M_1, M_2, \ldots, M_m\}^\top \) is the statistical model approximating the data; \( P \) is an \( m \) by \( n \) matrix describing the PMFs of \( n \) linear model components with elements \( P_{ij} = P(X = \xi(i)|K = j) \), with function \( \xi \) translating index \( i \) into the equivalent \( x \) vector realisation, i.e. a conversion from an integer value (1 to \( m \)) into an equivalent vector with individual binary elements; and \( Q = \{Q_1, Q_2, \ldots, Q_n\}^\top \) is a vector corresponding to the amount of each component present within the data.

The model weights, \( Q \), are proportional to the area measurements sought because each sampled pixel pattern is associated with a finite surface area, \( a_d \), where \( a \) is an area in pixels and \( d \) is a natural number index representing an individual sample. Solving the measurement problem then involves a probability ‘inversion’, using Bayes theorem, which requires the estimation of model parameters from the histogram data,

\[ Q \approx \tilde{P}H \]  

followed by a scaling,

\[ A = Q \circ \alpha \]  

where \( \tilde{P} \) is an \( n \) by \( m \) matrix with elements \( \tilde{P}_{ij} = P(K = i|X = \xi(j)) \), i.e. posterior probabilities that texture \( i \) was the source of the observation associated with \( j \); and \( A \) is the vector of area measurements sought, related to model weights by scaling factors, \( \alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}^\top \), applied here using the entrywise product. This process is described in detail in section 3.2.

For scientific applications, it is essential to address the issue of measurement noise so that reliable error bars can be assigned to measurements. The sampling errors found within histogram bins can reasonably be assumed to be Poisson in nature, as the physical processes which generate patterns seen within planetary images are combinations of rare events unfolding in time. These sampling errors will propagate to estimates of \( Q \) and \( A \), and (through the central limit theorem) will become more Gaussian-like. Assuming unbiased estimates, these uncertainties can be modelled as zero-mean Gaussian
noise and summarised with error covariance matrices, $C_Q$ and $C_A$, respectively. Estimating these error covariances is a multi-stage process which is described in section 3.4.

In very simple terrains where patterns are highly stable, it may be sufficient to attribute each linear model component uniquely to an underlying classification for which measurements are sought, i.e. one texture per terrain. However, terrains generally contain several texture components, even within the same classification. For example, dunes with different orientations, scales and morphologies might still be classified as ‘dunes’. Subsets of model weights can therefore be combined together to give summaries of total area measurements for related textures. This permits terrains with variable quantities of different textures to be linearly modelled using combinations of sub-textures. This flexibility permits models to be adjusted on a dataset-by-dataset basis to match their corresponding variable distributions. Total areas of superordinate terrain classifications (e.g. the terrain type dunes, containing the different sub-texture ripples) are then given by,

$$B = KA$$  \hspace{1cm} (4)

where $B = \{B_1, B_2, \ldots, B_z\}^T$ is a vector of total area measurements for superordinate terrain classifications; and $K$ is a $z$ by $n$ mapping matrix with mutually exclusive binary elements which relate individual model components to their superordinate classes. The elements of $K$ are then given by,

$$K_{ij} = \delta(i \in \mathbb{K}_j)$$  \hspace{1cm} (5)

where $\mathbb{K}_j$ is a subset of natural number labels associated with superordinate classification $j$; and $\delta$ evaluates to unity if label $i$ is associated with the classification $j$, and zero otherwise. The uncertainty in these summary measurements can be summarised with an error covariance matrix $C_B$. The process of computing total areas and errors is given in section 3.5.

A conventional notation is adopted here for referencing elements of vectors and matrices, i.e. lower-case subscripts. However, the methodology does often require use of multiple instances of vectors and matrices (e.g. for different images or subregions of images) and also terms which progressively update (e.g. through iterative algorithms). To ensure subscripts denoting vector/matrix elements are not confused with those denoting different vectors/matrices, bracketed subscripts are also used. A lower-case ‘r’ will be adopted to indicate vectors/matrices associated with different data, and a
lower-case ‘t’ will be used to indicate a particular vector/matrix at a given iteration. For example, $P_{ij(r)(t)}$ denotes element $ij$ in matrix $P$ associated with data $r$ at time step $t$. As elements of probability matrices can also be expressed as PMFs, these too may be presented with $r$ and $t$ subscripts. For example, $P_{rt}(x|k)$ denotes the PMF for data $r$ at time $t$. In such cases there is no risk of confusion with an index therefore brackets are not used, making the notation more compact.

The linear modelling and error analysis methodology we propose is based upon linear Poisson models (LPM) [44]. Using the terminology of [44], each texture, $k$, may also be referred to as a model component, as LPMs perform an Independent Component Analysis. The following sections include an overview of LPMs and how they are applied in the problem domain. The primary criterion of success within this domain is that repeated independent area measurements should, in practice, achieve levels of accuracy in-line with those predicted from the theoretical error analysis. This requires a significant quantity of ground-truth data to corroborate, therefore experiments are conducted using Monte Carlo simulations and bootstrap re-sampling utilising samples of martian terrain from HiRISE imagery.

3. Methodology

Our proposed use of LPMs for making surface area measurements requires an appropriate selection of image encoding. An encoded observation, $x$, (drawn from $X$) must behave as an independent Poisson event for LPMs to be applied. We present a ‘Poisson blob’ image encoding, from which histograms can be populated (section 3.1) which capture texture information. Once populated, we show that for incoming histograms, $H$, and a set of pre-learned PMFs, $P$, iterative use of the E-step of an expectation maximisation (EM) algorithm can be applied to estimate linear model weights, $Q$, and classification probabilities, $\tilde{P}$, which can be converted into terrain area measurements, $A$ (section 3.2). We also extend the estimation process to implement a full EM algorithm, utilising the M-step to assist in the determination of $P$ from exemplar terrain histograms and use a chi-squared per degree of freedom test to confirm that resulting linear models give good approximations to terrain histogram distributions (section 3.3). We demonstrate how the theories of error estimation and propagation can be applied to provide error covariances on measured quantities (section 3.4). We then detail how related areas can be combined into total area measurements with associated
error estimates (section 3.5). Finally, an algorithm is outlined in pseudo-code to illustrate how the area measurement method can be implemented (section 3.6), before being applied to test data.

3.1. Poisson image encoding

We choose to encode local arrangements of pixels using a modified BRIEF [45] representation. This defines the exact meaning of the random variable \( \mathbf{X} \) in relation to image data. The encoding divides a 12-bit binary vector into an 8-bit \((X_1, \ldots, X_8)\) ‘BRIEF’ part and 4-bit \((X_9, \ldots, X_{12})\) ‘size’ part. The BRIEF part captures the local structure at a location within the image, whilst the size part captures the extent of that structure.

The 8-bit BRIEF part is composed of a binary string indicating the results of 8 pixel pair intensity comparisons within an 8 pixel radius. This produces a compact subsample of the light and dark patterns within a 16 pixel wide sampling disc. From the 8 pixel pairs, \([(\alpha_1, \beta_1), \ldots, (\alpha_8, \beta_8)]\), an 8-bit binary pattern is sampled, \([x_1, \ldots, x_8]\), where a bit, \(x_i\), is set if the corresponding \(\alpha_i\) pixel is brighter than the corresponding \(\beta_i\) pixel by a given threshold, i.e. a sample \(x_i = \delta(\alpha_i - \beta_i > \theta)\). The initial selection of pixel pairs is made randomly from a uniform spatial distribution, relative to the sampling disc’s origin (centre). After initialisation, the relative positions of the pairs are kept fixed to provide a consistent sampling scheme, i.e. the random selection only happens once, from which point that chosen pixel sampling order is maintained. Whilst specific arrangements of pairs might be expected to be more discriminating on specific textures, a random organisation has been shown to be as good as any other in general cases [45]. Samples are taken by exhaustively scanning the origin of the disc across all image locations such that an entire image is described in this sub-sampled format. In this way, a BRIEF pattern is assigned to describe the local structure around each pixel.

In order to improve the spatial independence of image patch samples, adjacent image locations sharing the same light and dark patterns (i.e. exhibiting identical 8-bit binary BRIEF-like strings) are grouped together into ‘blobs’. Individual blobs are denoted with the index \(d \in \{1, 2, \ldots, N\}\), where \(N\) is the total number of blobs found in the image. The area, \(a_d\), of each blob (in pixels) is then recorded. This size information is used for two purposes: firstly, for use in future area calculation; and secondly, as part of the \(\mathbf{X}\) encoding itself. The area is encoded in the remaining 4 bits \((X_9, \ldots, X_{12})\) of the representation in a reduced form. To maintain a reasonably sized pattern space, i.e. \(2^{8+4}\), whilst also permitting a wide range of blob sizes, the
size information is binned into 16 bands on a logarithmic scale. 0000 then corresponds to blobs of 1 pixel; 0001 to blobs up to 2 pixels; 0010 for up to 4 pixels; 0011 up to 8, etc.

Groups of similar BRIEF patterns tend to follow physical structures, such as ridges, which we believe should approximately behave as independent Poisson events. Each possible blob is then used to define a histogram bin in a LPM. This results in $H$ containing $2^{12}$ nominal bins. A histogram is populated by the frequency of occurrence of each blob found within a planetary surface image.

Some surface regions are poorly approximated by Poisson events and are also difficult to attribute to specific terrains. In particular, a blob describing uniform regions, e.g. smooth plains between features, can account for large connected parts of a surface yet provide little information. The quantities of uninformative areas are also anti-correlated with the quantities of those Poisson blobs which do contain useful structure. To avoid contamination from these areas an ‘uninformative’ case is defined as any blob which falls into the largest size band, i.e. samples where bits 9 to 12 are set (1111). These uninformative areas are filtered out, as they require application-specific processing, which is not be addressed here. Figure 4 illustrates the representation using a simplified example.

3.2. Estimation of component weights and terrain areas

The statistical modelling of Poisson samples in this type of scenario is usually attributed to Fermi in the form of the extended maximum likelihood [46] approach, which in terms of PMFs and histogram weighting quantities is given by,

$$
\ln \mathcal{L} = \sum_x \ln \left[ \sum_k P(X = x | K = k) Q_k \right] H_{\xi^{-1}(x)} - \sum_k Q_k \\
$$

(6)

where $\xi^{-1}(x)$ maps $x$ to the corresponding index of the histogram vector. To simplify the notation, $P(X = x | K = k)$ will henceforth be abbreviated to $P(x | k)$. Also, the function $x = \xi(i)$ and its inverse $i = \xi^{-1}(x)$ shall be dropped, allowing the vector $x$ to be used directly as an index, and vice-versa, with the understanding that a binary vector can be interpreted directly as a natural number.

Given a set of trained texture component PMFs (which will be considered in section 3.3) the E-step of the EM algorithm can be used to optimise (6)
with respect to free parameters, \( Q \). This is achieved by iteratively updating the elements of \( Q \) by weighting them with the current estimate of the posteriori probability \( P_t(k|x) \), (i.e. from \( \hat{P} \)), starting from some initial estimate at iteration \( t = 0 \),

\[
Q(t) = \hat{P}_{(t-1)}H
\]  

(7)

where \( Q(t) \) is the weighting quantity vector estimate at time \( t \); and \( \hat{P}_{(t-1)} \) is the last estimate of the posterior probabilities. For a single weight, \( Q_k \), this update is implemented using Bayes Theorem \[47\] \[48\],

\[
Q_k(t) = \sum_x P_t(k|x)H_x = \sum_x \frac{P(x|k)Q_{k(t-1)}}{\sum_l P(x|l)Q_{l(t-1)}}H_x
\]

(8)

where \( Q_{k(t-1)} \) is used as the Bayesian ‘prior’, as under a Frequentist interpretation the probability \( P(k) \) is proportional to the frequency of occurrence of \( k \), i.e. the quantity. Upon convergence, the quantities \( \hat{Q} = Q_{(t=\infty)} \) provide the maximum likelihood solutions to the model weights, assuming the PMFs of the model components were correct for the incoming histogram data.

Upon convergence, the area estimates of textures corresponding to each model component are computed via scaling the model weights using equation (3), which for individual elements becomes,

\[
A_k = Q_k \alpha_k
\]

(9)

where the scaling factor is given by,

\[
\alpha_k = \sum_x \frac{P(x|k)a_x}{M_x}
\]

(10)

with \( a_x \) being the total area, in pixels, covered by blobs which are encoded with the specific pattern, \( x \), i.e. the sum of areas of the repeated instances of \( x \),

\[
a_x = \sum_d \delta(x_d = x)a_d
\]

(11)

where the sum over \( d \) is a sum over every individual blob in the image; \( a_d \) is the area of a specific blob; and \( \delta \) evaluates to unity in the case where the specific blob, \( x_d \), encodes the same pattern as the one for which a total area is sought. The total terrain area measurements can be then computed by summing the individual sub-texture areas using equation (4).
3.3. Estimation of component PMFs from exemplar terrains

The distribution of a class of terrain is approximated using a set of \( n \) sub-texture PMFs, i.e. the linear model components. These PMFs must, however, be estimated from somewhere. Given a set of \( N \) \((n \ll N)\) independent terrain exemplars (i.e. training data images), a full EM algorithm can be used to extract the best linear PMF approximations, i.e. perform an independent component analysis (ICA).

Histogram data sampled from a dataset \( r \) can be viewed as being generated by,

\[
H_{x(r)} = \sum_{k=1}^{n} R_r(x|k) \tag{12}
\]

where \( H_{x(r)} \) is the \( x \) bin of the \( r^{th} \) independent training example; and \( R_r(x|k) \) is the data generator contributing to the frequency of \( x \) bin in example \( r \) from texture \( k \). The EM-based ICA method presented here is designed to estimate approximations to the set of functions \( R_r(x|k) \) using a common set of PMFs and data specific weights such that,

\[
R_r(x|k) \approx P(x|k)Q_{k(r)} \tag{13}
\]

Initial estimates of PMFs are generated at iteration 0 giving \( P_{(t=0)} \). These initial estimates can be based upon any random values, as the iterative algorithm will refine them over time. Weighting quantities are then estimated (via the estimation process detailed in section 3.2) for each of the \( N \) examples giving modelled approximations,

\[
H_r \approx M_{(r)(t)} = P_{(t)}Q_{(r)(t)} \tag{14}
\]

where all histogram models share a common definition of \( P \); and each histogram has its own estimate of weighting quantities, \( Q_{(r)} \). Consistent with the EM algorithm, the estimated posteriori probabilities \( \hat{P} \), are used to provide a new estimate of the contribution to each histogram, \( r \), from each component, \( k \), i.e. an estimate of the underlying sub-texture generators,

\[
\hat{R}_{r(t)}(x|k) = P_{r(t-1)}(k|x)H_{x(r)} \tag{15}
\]

These independent estimates are combined and normalised to create a new common estimate of \( P_{(t)} \),
\[ P_{xk(t)} = P_t(x|k) = \frac{\sum_r \hat{R}_r(t)(x|k)}{\sum_r Q_k(r)(t)} \] (16)

This new common estimate of \( P \) is used to reestimate quantities and posteriori probabilities for exemplar histograms, i.e. using the estimation method of section 3.2 again. The process continues until convergence giving \( \hat{P} = P_{t=\infty} \), maximising (6), consistent with the convergence theorem of EM. To avoid the risk of converging into a local minimum the algorithm can be restarted multiple times from different random PMF initialisations.

If the number of required components, \( n \), is unknown the ICA can be repeated until a goodness-of-fit criterion is met. Assuming normally-distributed residuals, the fit between a model with the correct number of components and observed data will have a chi-squared per degree of freedom of 1. However, the models sought have Poisson residuals, with discrepancies between Poisson and Gaussian distributions being especially evident at low sample statistics. A square-root (Anscombe’s) transform [49] can be performed to both model and data to transform the residuals into something better approximating a Gaussian with uniform width of \( \sigma^2 = \frac{1}{4} \), thereby widening the scope of the test to such cases. A chi-squared per degrees of freedom function, \( \chi^2_D \), can then be defined as,

\[ \chi^2_D = \frac{1}{N} \sum_r \frac{4}{m-n} \sum_x \left( \sqrt{M_{x(r)}} - \sqrt{H_{x(r)}} \right)^2 \] (17)

where \( m \) is the number of bins in the examplar histograms; \( n \) is the number of components in the model; and \( M_{x(r)} \) is the modelled frequency in the \( r^{th} \) example histogram’s bin accounted for by the selected components.

3.4. Covariance estimation for component weights

In order to estimate area measurement accuracy, the uncertainties in model weighting quantities must first be estimated. Incoming histogram data, \( H \), will contain statistical Poisson noise. Additionally, noise in extracted PMFs, \( P \), which are used multiple times on new incoming data, will cause systematic effects. Error propagation can be used to estimate a covariance matrix, \( C_Q \), for weighting quantities, which take into account these sources of uncertainty. This propagation of uncertainty must accommodate the iterative nature of the EM algorithm and can be achieved using the following three steps:
1. Estimation of error sources: i.e. variance in $H$ and $P$ (section 3.4.1);
2. Single EM step error: a single EM step estimate of how noise affects one instance of the update function (section 3.4.2);
3. Error amplification: an amplification stage accounting for the accumulative effects of the iterative feedback of errors in subsequent EM steps (section 3.4.3).

3.4.1. Estimation of error sources

The statistical source of error in final quantity estimates stems from Poisson variability in incoming histogram bin frequencies. This error is assumed to be independent from bin to bin. The variance from a single bin is therefore,

$$\sigma^2_{H_x} = <H_x> \approx H_x$$ (18)

where $<H_x>$ indicates the statistical expectation of the bin frequency. For implementation and notational convenience this can be approximated by the observed frequency.

The systematic source of error stems from Poisson variability in exemplar histogram bin frequencies, which are weighted by $P(k|x)$ (giving $\hat{R}_r(x|k)$), summed, then normalised into model PMFs, $P$ in (16). The resulting variances can be determined via a two step argument:

1) For any given approximation of $P(k|x)$, if the total content of an example histogram bin $H_{x(r)}$ were fixed, then the variance on $\hat{R}_r(x|k)$ would follow a Binomial distribution. Each bin entry will have a probability of $P(k|x)$ of being included in the sample and a probability of $1 - P(k|x)$ of being excluded. This variance is given by,

$$H_{x(r)}P(k|x)[1 - P(k|x)] = H_{x(r)}P(k|x) - H_{x(r)}P(k|x)^2$$ (19)

2) However, as $H_{x(r)}$ is not a fixed quantity (as this is a Poisson random variable) there is also an independent Poisson contribution to the error from the originating histogram sample,

$$\left(\frac{\partial \hat{R}_r(x|k)}{\partial H_{x(r)}}\right)^2 \sigma^2_{H_{x(r)}} = P(k|x)^2H_{x(r)}$$ (20)

which adds to the Binomial variance to give a total variance of,

$$\sigma^2_{\hat{R}_r(x|k)} = P(k|x)H_{x(r)} = \hat{R}_r(x|k)$$ (21)
resulting in an overall Poisson behaviour to the estimate, $\hat{R}_r(x|k)$, i.e. the variance of the value is proportional to the value itself, consistent with Poisson statistics.

The sum of the independent example histogram errors therefore also combine in a manner similar to Poisson variances, giving a systematic error equivalent to using a raw histogram sample, $H_{x|k} = \sum_r R_r(x|k)$, containing all of the information for component $k$ from all exemplars,

$$\sigma^2_{H_{x|k}} = H_{x|k} \approx H_{x|k}$$ (22)

### 3.4.2. Single EM step error

For the purposes of error propagation it is convenient to consider the EM update function in terms of a single histogram bin, $x$, and every other bin which is not $x$, which will be denoted as $\bar{x}$, e.g. $H_x = \sum_{y \neq x} H_y$. It is also convenient to consider components in a similar notation using $k$ and all other components, $\bar{k}$, e.g. $Q_k = \sum_{l \neq k} Q_l$. The update function can then be stated in terms of an incoming histogram, $H_x$, (data) and exemplar histogram contributions, $H_{x|k}$, (model) bins giving,

$$Q_k' = P(k|x)H_x + P(k|\bar{x})H_{\bar{x}}$$ (23)

$$= \frac{\left( \frac{H_{x|k}Q_k}{H_{x|k} + H_{\bar{x}|k}} \right) H_x}{\left( \frac{H_{x|k}Q_k}{H_{x|k} + H_{\bar{x}|k}} \right) + \frac{H_{\bar{x}|k}Q_k}{H_{x|k} + H_{\bar{x}|k}}} + \frac{\left( \frac{H_{\bar{x}|k}Q_k}{H_{x|k} + H_{\bar{x}|k}} \right) H_x}{\left( \frac{H_{\bar{x}|k}Q_k}{H_{x|k} + H_{\bar{x}|k}} \right) + \frac{H_{x|k}Q_k}{H_{x|k} + H_{\bar{x}|k}}}$$ (24)

where $Q'$ is the updated quantity; and $Q$ is the previous quantity. Uncertainty from the two sources of error can be propagated through this update function by considering how small changes in the inputs affect the estimated vector of quantities. As the two sources are independent their contributions to the covariance can be derived separately and then summed,

$$C_{EM\text{Step}} = C_{data} + C_{model}$$ (25)

$$C_{ij(data)} = \sum_x \left[ \left( \frac{\partial Q_i'}{\partial H_x} \right) \left( \frac{\partial Q_j'}{\partial H_x} \right) \sigma^2_{H_x} \right]$$ (26)

$$C_{ij(model)} = \sum_x \left[ \sum_k \left( \frac{\partial Q_i'}{\partial H_{x|k}} \right) \left( \frac{\partial Q_j'}{\partial H_{x|k}} \right) \sigma^2_{H_{x|k}} \right]$$ (27)
where $C_{\text{data}}$ is the statistical contribution from the incoming histogram data; and $C_{\text{model}}$ is the systematic contribution from the training exemplar histograms used to construct the component models.

The statistical contribution is straightforward, giving,

$$C_{ij(\text{data})} = \sum_x P(i|x)P(j|x)H_x$$  \hspace{1cm} (28)

In contrast, the systematic contribution involves relatively complex derivatives. Defining the total quantity of training data for component $k$ as $T_k = H_{x|k} + H_{\bar{x}|k}$, in the cases where $i = k$ or $j = k$, the derivatives are,

$$\frac{\partial Q_i}{\partial H_{x|k}} = \frac{\partial Q_j}{\partial H_{x|k}} = \frac{P(k|x)P(\bar{k}|x)P(\bar{x}|k)H_x - P(\bar{k}|\bar{x})P(k|\bar{x})H_x}{T_k P(x|k)}$$  \hspace{1cm} (29)

In the cases where $i \neq k$ (or similarly when $j \neq k$, substituting $j$ for $i$ in the following) the same terms become,

$$\frac{\partial Q_i}{\partial H_{x|k}} = \frac{P(i|\bar{x})P(k|\bar{x})H_x - P(i|x)P(k|x)P(\bar{x}|k)H_x}{T_k P(x|k)}$$  \hspace{1cm} (30)

Substituting these results back into the covariance calculation (27) completes the estimate of the error contribution on quantities after performing a single EM step.

### 3.4.3. Error amplification

Our simulations have shown that final errors are larger than those seen during a single EM step. Others have also observed that initial error are amplified in iterative processes [50]. For EM, this amplification can be approximated by a convergent geometric series and applied to the single step covariance using a matrix, $S$,

$$C = S^t C_{\text{EM step}} S$$  \hspace{1cm} (31)

$$S = [I - \nabla Q]^{-1}$$  \hspace{1cm} (32)

where $I$ represents the identity matrix; and $\nabla Q$ is the Jacobian,

$$\nabla Q_{ij} = \frac{\partial Q_i}{\partial \Delta_j}$$  \hspace{1cm} (33)

15
where \( \Delta \) is an initial small error in quantity. The diagonal terms are given by,

\[
\nabla Q_{ii} = \sum_x P(x|i) - P(x|i)P(i|x)
\]

(34)

and the off-diagonal terms are given by,

\[
\nabla Q_{ij} = \sum_x -P(x|j)P(i|x)
\]

(35)

### 3.5. Covariance estimation for terrain area measurements

We first compute an error covariance matrix for sub-texture area measurements, i.e. per linear model component, then sum these into a total terrain class area measurement error covariance, i.e. for superordinate terrain classifications. To apply error propagation, the area measurement calculation is written in terms of weighting quantities (for which a covariance is already available, \( C_Q \)) combining equations (9) and (10),

\[
A_k = Q_k \sum_x \frac{P(x|k)a_x}{M_x}
\]

(36)

where \( M_x \) is the modelled frequency of \( x \) from (14). Assuming the major source of error is the weighting quantities, error propagation can be applied giving an \( n \) by \( n \) area covariance,

\[
C_A = \nabla A C_Q \nabla^T A
\]

(37)

where \( \nabla A \) is the matrix of partial derivatives

\[
\nabla A_{ij} = \frac{\partial A_i}{\partial Q_j}
\]

(38)

For the case when \( i = j \) this gives

\[
\nabla A_{ij} = \sum_x \frac{P(x|j)a_x}{M_x}
\]

(39)

\[
= \sum_x \frac{P(x|j)Q_ja_x}{M_xQ_j} = \sum_x \frac{P(j|x)a_x}{Q_j}
\]
\[
= \frac{A_j}{Q_j}
\]

which is zero for \( i \neq j \) forming a diagonal matrix.

The sub-texture areas are summed to give the total area of each terrain using (4). This simple linear addition of component areas imply a simple linear combination of errors. Error propagation can be applied to give final measurement accuracies,

\[
C_B = KAK^T
\]  \hspace{1cm} (40)

This covariance assumes the training data is representative of the incoming image. However, discrepancies may exist in practice with incoming images being subtly different from exemplars. Boundary conditions, the occasional inclusion of unknown features, or irregular parts of terrain textures can all invalidate modelled component PMFs. The \( \chi^2_D \) (17) model fit criterion, if greater than unity, can indicate data-model discrepancies, which can then be used to approximately correct error estimates by scaling covariances [46],

\[
C'_B = C_B \chi^2_D
\]  \hspace{1cm} (41)

Alternatively, if the fit is deemed too poor then the problem dataset can be rejected on the grounds that it is too unrepresentative of the trained model. This rejection option is not applicable during our testing because the Monte Carlo and bootstrap re-sampling evaluations are designed to avoid such problems.

3.6. Pseudo-code

The following algorithm gives the necessary steps for training a linear model and then applying it to make measurements of known terrain types.

3.6.1. Training algorithm

For each superordinate class of terrain (sets of sub-textures \( K_1, K_2, \ldots, K_z \)):

1. Gather \( N \) exemplar histograms from images of the terrain \( K \) \( \{H_{(r=1)}, H_{(r=2)}, \ldots, H_{(r=N)}\} \)
2. Create an initial common definition of \( P \) using a random number generator to assign any value to each element \( ij \)
3. Using an increasing number of components (sub-textures), whilst model fit, \( \chi^2_D \) equ. (17), is greater than unity and \( P \) has not converged
(a) For each terrain histogram $H(r)$
   i. Iterate equ. (7) and equ. (8) until $\tilde{P}(r)$ and $Q(r)$ converge
   ii. Use $\tilde{P}(r)$ to estimate individual component contributions, $\hat{R}_r$, using equ. (15)
(b) Sum and normalise each component to give a new common estimate of $P$ using equ (16)

4. Update sub-texture/terrain mapping matrix, $K$, to indicate which sub-texture components belong to terrain class $K$

3.6.2. *Measurement algorithm*

For an incoming terrain image histogram $H$

1. Iterate equ. (7) and equ. (8) until $\tilde{P}$ and $Q$ converge
2. Compute sub-texture areas from model weights giving $A$, using equ. (9)
3. Compute total terrain areas, $B$, using equ. (4)
4. Compute statistical and systematic EM covariance terms using equ. (26) and equ. (27)
5. Compute amplification matrix using equ. (32)
6. Compute component weighting quantity covariances, $C_Q$, using equ. (31)
7. Compute sub-texture area covariances, $C_A$, using equ. (37)
8. Compute total terrain area covariances, $C_B$, using equ. (40)
9. If the model fit equ. (17) is above 1 then scale the covariance using equ. (41)
10. Take the square-root of diagonal terms to give +/- 1 standard deviation error on measurements, i.e. $B_i \pm \sqrt{C_{Bii}}$

4. *Experiments*

Experiments were designed primarily to corroborate our error estimation theory, i.e. that predicted measurement accuracies can be achieved in practice [36]. Alternative representations, or optimised versions of the Poisson blob representation, may yield better levels of accuracy on a case-by-case basis. However, we opted to test the general measurement technique, which if successful would be equally applicable to improved representations.

Our area measurement and error prediction methods were tested using Monte Carlo simulated histograms and also composite martian terrain images
bootstrapped from real martian data. Monte Carlo simulations demonstrate that the first-order approximations used, for error propagation and their iterative amplification, provides sufficient predictive capabilities in the case of known distributions. The use of re-sampling of martian data demonstrates the levels to which real data - under the selected image encoding - meets the assumptions made by the theory when applied in practice.

During both Monte Carlo and bootstrapped image tests, areas and error predictions were repeatedly estimated. Deviations from known ground truths were compared to expected errors, allowing our predicted accuracies to be corroborated against empirically observed area measurement accuracies. The key to this approach is the ability to provide large quantities of independent ground truths, uninfluenced by any subjective human interpretations, hence the use of Monte Carlo and bootstrap re-sampling, which allows complete control over test data. This control over the data also prevents unrepresentative regions from contaminating results, e.g. the appearance of textures or features which were not seen during training. Each test was repeated with differing ratios of training to testing data.

4.1. Monte Carlo

Before applying the method to HiRISE data, the mathematics and implementation was tested in Monte Carlo simulation. Such a simulation generates ideal data using known reference distributions and errors. As such, theoretically predicted behaviour should trivially be observed. Behaviour deviating from prediction would indicate either a mistake in the theory or a bug in the software (or both). Monte Carlo also permits straightforward testing of the system using differing numbers of histogram bins than that dictated by the Poisson blob representation, which is otherwise fixed at $2^{12}$.

Training and testing histograms are generated by scaling reference distributions (in the form of known PMFs, $P$) by randomly selected known quantities. The scaling quantities (i.e. dictating $Q$) are all real-valued uniform random numbers between 1,000 and 1,000,000. For each histogram bin, a random Poisson number generator was used to draw independent samples from each bin and simulated component of terrain before summing to give total histogram frequencies. In terms of the methodology, this amounts to artificially implementing equation (12) for different datasets, which is otherwise an unknown function of real martian terrains.

Models are tested for histograms containing 64, 4096 and 16384 bins, simulating alternative BRIEF parameters giving different sized histograms,
demonstrating the flexibility of the method over a range of variables. The 64 bin reference distributions were manually created and can be seen in figure 2. These distributions are used as templates for creating up to 10 different components (e.g. simulated texture distributions) by shifting the distributions along the x-axis. For example, figure 3 shows the approximate distributions extracted at the point where 3 components per model are generated. In contrast, the 4096 and 16384 bin histogram component reference distributions were generated automatically with each reference bin frequency being drawn from a random number generator to give a range of synthetic data.

4.2. Bootstrapped terrain

In order to support the bootstrap re-sampling of real data, greyscale synthesised martian images were formed using terrains taken from the HiRISE project [2]. Samples of these images can be seen in figure 5. Each terrain is divided into 200 small rectangular tiles, which are randomly composited together to create arbitrarily sized regions containing combinations of different terrain types. Large numbers of independent samples are created from this finite set of tiles by randomly stretching tiles by up to 10% in both the X and Y directions, and adding additional grey-level noise before being combined. To prevent unrealistic discontinuities producing artifacts at tile boundaries, the tiles were blended together, separating the high frequency spatial features from the low frequency components. This allows the structures indicative of each texture to be maintained whilst locally averaging overall grey levels in order to prevent step changes in grey levels at boundaries. An example synthesised terrain from this bootstrap re-sampling of real data can be seen in figure 6.

The exact process can be summarised as follows:

1. Gather large images containing examples of martian terrains;
2. Using smoothing kernels with different widths, separate the low- and high-frequency spatial components of the terrain images;
3. Divide the example images into many small tiles;
4. Provide a labelled template showing the desired layout for a synthetic martian image;
5. Randomly select tiles (with replacement) for the terrain types prescribed by the template;
6. Slightly perturb the shape of each tile through stretching by random amounts up to 10% in both X and Y directions;
Dataset A: EPS 023675 0930  EPS 024889 2605  EPS 024926 2525
Dataset B: EPS 017810 1850  EPS 019243 2550  EPS 024984 2610
Dataset C: EPS 023661 0931  EPS 017866 2855  EPS 024979 2550
Dataset D: EPS 023738 0915  EPS 023744 1775  EPS 024927 2555
Dataset E: EPS 020558 0930  EPS 022543 0950  EPS 024949 2440
Dataset F: EPS 023767 0925  EPS 024662 2555  EPS 024950 1870
Dataset G: EPS 017866 2855  EPS 019104 1740  EPS 024899 2540
Dataset H: EPS 023729 0935  EPS 022882 2030  EPS 024991 2540
Dataset I: EPS 023621 0970  EPS 023676 0925  EPS 024983 2160
Dataset J: EPS 018948 2250  EPS 024724 1760  EPS 024883 1525

Table 1: The 10 datasets (A-J) used within experiments each contain texture samples taken from 3 HiRISE images. This table lists the HiRISE codes associated with these images and how they are grouped into the 10 datasets used during testing.

7. Form a composite image using the tile’s high-frequency spatial components;
8. Add uniform Gaussian noise to pixel values in the high-frequency image;
9. Form a composite image using the tile’s low-frequency spatial components;
10. Heavily smooth the low-frequency image;
11. Add together the high and low frequency images giving the final synthetic terrain.

Synthesised regions of individual terrain types are used with between 3 to 8 linear components extracted from each in order to satisfy the $\chi^2_D$ (17) model fit criterion. Data is constructed for 30 terrains grouped into 10 sets of 3 (Table 1), with images composed using differing quantities of each. 100 images are created per quantity of data and per group to allow repeated measurements to be acquired in order to observe the empirical measurement error. Different size images are again used to test different ratios of training to testing data.

5. Discussion

The experiments described above were designed to test agreement between predicted errors and observed errors (as opposed to estimated areas compared
with true areas), as errors embody deviations of measurements from ground-truth, thereby corroborating both area measurements and error estimates simultaneously.

Monte Carlo tests confirm our area measurement and error prediction methods are valid, in the case where training data is guaranteed to be representative. Figure 7 shows that for different ratios of training to testing data our method can make area measurements within accuracies correctly predicted by the associated covariances. This plot is a summary which includes results from different sized histograms and model complexities. Figure 8, which shares a common x-axis to Figure 7, shows that the systematic component of the error is negligible when the quantity of testing data is small in comparison to the training data, but becomes the dominant source of uncertainty when the quantity of testing data becomes large. Figure 9 confirms that the method provides an adequate approximation to measurements and errors when martian data is presented, with most results achieving error estimates within a factor of 2 of real errors. Figure 10 illustrates the actual accuracies attainable, with measurements being estimated to within as little as half a percent and up to two percent of the true areas, depending upon the terrain. Allowing for the typical underestimate of a factor of 2, the actual accuracies are still within only a few percent of ground truth values.

Our method is designed for quantitative scientific use which we believe requires a good understanding of uncertainty - hence our emphasis on achieving good error predictions. To be used practically, we believe that errors must be estimated to within a factor of 2. If observed errors are more than a factor of 2 larger than predicted then this implies that the majority of the error is not accounted for by our theory. In such cases the associated measurement is not sufficiently well understood and should not be trusted. This minimum level of agreement is successfully achieved for most of the martian terrains. However, there are mixtures of terrains which fall outside of this limit.

We believe that discrepancies between predicted errors and observed errors in the martian data can be explained by spatial correlations between adjacent Poisson blobs. The theory assumes all blobs are independent Poisson events and the Monte Carlo simulation enforces this. However, in martian data it is possible that, within some textures, small groups of blobs systematically appear in pairs or triplets manifesting as single events which are then double (or triple) counted. This explains the factors of 2 to 3 discrepancies which could be corrected with a straightforward calibration. Improvements could also be made by optimising the Poisson blob encoding via the pixel pairs,
radius and threshold parameters. Calibrations and optimisations will require quantitative testing on an application-specific basis in order to check the appropriateness of the method when used in real tasks.

Successful area measurements can potentially be used to solve the types of problems noted in the introduction. Examples include the estimation of crater SFDs from crater areas, as there is a fixed relationship between crater area and diameter; the orientation of dunes can be identified by training different sub-texture components using differently oriented examples; and the wavelength of dunes can be estimated from the density of dunes in comparison to the inter-dune uninformative spaces. However, there are challenges to overcome including gathering sufficient training data to generalise models to different illumination conditions; dealing with occlusion; extending to multiple scales; and identifying and avoiding regions of images containing textures which are not modelled by the trained components.

6. Conclusion

We have presented a quantitative method for making supervised texture-based area measurements from surface terrain images appropriate for scientific applications. Our method has been corroborated using Monte Carlo simulated Poisson 'blob' histograms confirming the validity of our area measurement and associated error estimation theories. These tests validate our use of linear Poisson models and their first-order approximations for making error predictions. We have demonstrated the performance of our method using images synthesised from real martian data where measurement accuracies were achieved to within up to half a percent of the true values. Bootstrap re-sampling from martian images revealed that assumptions made in the formulation of LPM theory approximately hold in real data. This permitted the successful prediction of the majority of uncertainties in martian data, typically to within a factor of 2, which could potentially be calibrated on a terrain-by-terrain basis leading improved error agreements.

The next stage in development will be to improve error agreements and to apply the method to tackle real planetary science problems. Our software implementation will be made available via the www.tina-vision.net website as part of the open-source TINA image analysis libraries. This will permit others to corroborate our initial findings and also investigate for themselves potential science applications.
7. Acknowledgements

We would like to thank The Leverhulme Trust and STFC for project and PhD studentship funding, respectively; The BMVA for travel bursaries; K.H. Joy and A. Seepujak, of the University of Manchester for support.

References


[23] Y. Sawabe et al., Automated detection and classification of lunar craters using multiple approaches, COSPAR (Advances in Space Research), 2005


[37] W. Foerstner, Diagnostics and Performance evaluation in Computer Vision, Workshop on Robust Computer Vision, 1994


[40] H. Ragheb et al., Quantitative Shape Analysis with Weighted Covariance Estimates for Increased Statistical Efficiency, Frontiers in Zoology, 10(16), 2013


Figure 1: Outline of proposed method. Training images are encoded to give a statistical distribution of pixel patterns. An Independent Component Analysis extracts stable textures within terrains which can then be fitted to new data. Error propagation is used to compute measurement errors from incoming sampling noise.
Figure 2: Examples of 64 bin reference distribution templates used during Monte-Carlo simulations.

Figure 3: Examples of components extracted using EM ICA training algorithm during 64 bin Monte-Carlo simulations
Figure 4: Simplified illustration of circular BRIEF samples taken from a martian texture exhibiting dark linear features. Feature $X$ is a Poisson ‘blob’ containing identical BRIEF samples in a connected linear feature. Feature $Y$ is part of an uninformative area. Here, the BRIEF part of the representation has been reduced to 6 bits and the size part is presented in decimal.

Figure 5: Examples of HiRISE martian images.
Figure 6: Example of synthesised terrain created from bootstrap re-sampling from tiles of 3 terrain types. It is acknowledged that such images are only semi-realistic, in that the artificial boundaries between different textures are never observed in HiRISE images. However, the constituent textures are genuine, which is the true object of the statistical modeling and area estimation.
Figure 7: Comparison of observed area measurement errors to predicted errors for different quantities of training and testing data. Each point represents a different group of simulated distributions in Monte-Carlo study.
Figure 8: Statistical and systematic components of predicted area measurement errors for different quantities of training and testing data.
Figure 9: Comparison of observed area measurement errors to predicted errors for different quantities of training and testing data. Each curve represents a different mixture of 3 simulated martian terrain images.
Figure 10: Percentage accuracy of area measurements for different quantities of training and testing data. Each curve is a predicted 1 standard deviation error as a percentage of measured areas within simulated terrains.