Planetary Crater Detection and Registration Using Marked Point Processes, Multiple Birth and Death Algorithms, and Region-based Analysis

David Solarna², Gabriele Moser², Jacqueline Le Moigne¹, and Sebastiano B. Serpico²

¹ NASA Goddard Space Flight Center
² University of Genoa

IGARSS 2017, Fort Worth, Texas, July 2017
Content

Introduction

Crater Detection
  – Marked Point Process Model
  – Energy Function
  – Multiple Birth and Death Algorithm
  – Region-of-Interest Approach
  – Experimental Results

Image Registration
  – 2-step Approach
  – Experimental Results

Conclusion
Introduction

Crater Detection
- Marked Point Process Model
- Energy Function
- Multiple Birth and Death Algorithm
- Region-of-Interest Approach
- Experimental Results

Image Registration
- 2-step Approach
- Experimental Results

Conclusion
Introduction

Need for automated methods for image registration

Launch of several planetary missions
Design of new and powerful sensors

Large data sets

- Multisensor
- Multitemporal
- Both

Objective

- Crater detection in planetary images
- Development of an image registration method based on the extracted features
Introduction

**Crater Detection**
- Marked Point Process Model
- Energy Function
- Multiple Birth and Death Algorithm
- Region-of-Interest Approach
- Experimental Results

**Image Registration**
- 2-step Approach
- Experimental Results

Conclusion
Crater detection based on a marked point process (MPP) model

**Marked Point Processes**

MPP: Stochastic Process \[\xrightarrow{\text{Realizations}}\] Configurations of objects, each described by a marked point

**Mathematical Formulation**

A point process \(X\), defined over a bounded subset \(P\) of \(\mathbb{R}^2\) maps from a probability space to a configuration of points in \(P\).

Realizations of the process \(X\) are random configurations \(x\) of points, \(x = \{x_1, \ldots, x_n\}\), where \(x_i\) is the location of the \(i\)th point in the image plane \((x_i \in P)\).

A configuration of an MPP consists of a point process whose points are enriched with additional parameters, called marks and aimed at parameterizing objects linked to the points.

Bayesian approach: Maximum a posteriori (MAP) rule to fit the model to the image is equivalent to minimizing an energy function (computationally challenging).
Marked Point Process for Crater Detection

Modelling
- Crater 
  - Ellipse
  - Distribution of craters on the surface
  - Distribution of ellipses on the image plane

Realization Example

\[
\text{Point Process} \\
x = \{x_1, \ldots, x_7\} \\
\text{Parameters} \\
x_i = (x_{0i}, y_{0i}, a_i, b_i, \theta_i)
\]
**Crater Detection – Energy Function**

**Energy function** of the configuration \( X = \{x_1, x_2, \ldots, x_n\} \) wrt the extracted set \( C \) of **contour pixels** (Canny):

\[
U(X|C) = U_P(X) + U_L(C|X)
\]

**Prior**

**Repulsion** coefficient based on the overlapping of the ellipses (overlapping craters are quite unlikely)

\[
U_P(X) = \frac{1}{n} \sum_{x_i \wedge x_j > 0} \frac{x_i \wedge x_j}{x_i \vee x_j}
\]

\( x_i \vee x_j = \text{area of union of ellipses } x_i \text{ and } x_j \)

\( x_i \wedge x_j = \text{area of intersection of } x_i \text{ and } x_j \)

**Likelihood**

Two terms, one based on a **correlation** measure, the other based on a **distance** measure (fit between contours and realization of \( X \))

\[
U_L(C|X) = \sum_{i=1}^{n} \left[ \frac{d_H(x_i^0, C)}{na_i} - \frac{|x_i^0 \cap C|}{|C|} \right]
\]

\( x_i^0 = \text{set of pixels corresponding to ellipse } x_i \text{ in the image plane} \)

\( d_H(x_i^0, C) = \text{Hausdorff distance between ellipse } x_i \text{ and the contours:} \)

\[
d_H(A, B) = \max \left\{ \sup_{\alpha \in A} \inf_{\beta \in B} d(\alpha, \beta); \sup_{\beta \in B} \inf_{\alpha \in A} d(\alpha, \beta) \right\}
\]

Classical distance between sets \( d(A, B) = 0 \)
Crater Detection – Energy Minimization

Markov chain Monte Carlo-type method
Simulated Annealing scheme

Markov chain sampled by a multiple birth and death (MBD) algorithm

Initialization
- Contour Extraction (Canny) and Parameter Initialization
- Generation of the Birth Map used to speed up the convergence

Birth Step
- Computation of the Birth Probabilities for each pixel
- Birth Phase

Death Step
- Energy Computation for all the ellipses
- Computation of Death Probability and Death Phase according to Likelihood
- Configuration refinement according to Prior

Update
- Convergence Test
Birth Step

For each pixel \( s \) in the image, compute the birth probability as \( \min\{\delta \cdot B(s), 1\} \), where:

\[
B(s) = \frac{b(s)}{\sum s b(s)}
\]

\( b(s) \) is the birth map computed from the contour map using generalized Hough transform and Gaussian filtering.

Death Step

For each ellipse \( x_i \) in the configuration, compute the death probability as \( d(x_i) \):

\[
d(x_i) = \frac{\delta \cdot a(x_i)}{1 + \delta \cdot a(x_i)}
\]

\[
a(x_i) = \exp[-\beta (U_L(X \backslash \{x_i\} | C) - U_L(X | C))]
\]
Crater Detection – Region Based Approach

Region-Based Approach

Why?
- MBD is **computationally heavy**
- Computational burden increases with **image size**

Region Based Flowchart and Example

**Initialization**

Detection of **regions on interest** based on the birth map

MBD in each **region**

**Aggregation**
Crater Detection – Data Sets

- 6 THEMIS (Thermal Emission Imaging System) images, TIR, 100m resolution, Mars Odissey mission
- 7 HRSC (High Resolution Stereo Color) images, VIS, ~20m resolution, Mars Express mission
- Image sizes from 1581 × 1827 to 2950 × 5742 pixels

Quantitative Performance Assessment of the crater detection algorithm: Detection Percentage ($D$), Branching Factor ($B$), and Quality Percentage ($Q$)

$$D = \frac{TP}{TP + FN}$$
$$B = \frac{FP}{TP}$$
$$Q = \frac{TP}{TP + FP + FN}$$

<table>
<thead>
<tr>
<th>Data</th>
<th>$D$</th>
<th>$B$</th>
<th>$Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg on all THEMIS</td>
<td>0.91</td>
<td>0.10</td>
<td>0.83</td>
</tr>
<tr>
<td>Avg on all HRSC</td>
<td>0.89</td>
<td>0.06</td>
<td>0.85</td>
</tr>
<tr>
<td>Avg on all images</td>
<td>0.90</td>
<td>0.09</td>
<td>0.84</td>
</tr>
</tbody>
</table>
Crater Detection – Results

<table>
<thead>
<tr>
<th>Crater</th>
<th>( C = (x_0, y_0) )</th>
<th>Semi-axes ((a, b))</th>
<th>Orientation (\theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crater 1</td>
<td>(139, 393)</td>
<td>(35, 33)</td>
<td>64°</td>
</tr>
<tr>
<td>Crater 2</td>
<td>(258, 756)</td>
<td>(51, 50)</td>
<td>115°</td>
</tr>
<tr>
<td>Crater 3</td>
<td>(343, 23)</td>
<td>(13, 12)</td>
<td>180°</td>
</tr>
<tr>
<td>Crater 4</td>
<td>(591, 215)</td>
<td>(19, 18)</td>
<td>31°</td>
</tr>
<tr>
<td>Crater 5</td>
<td>(919, 157)</td>
<td>(15, 14)</td>
<td>106°</td>
</tr>
</tbody>
</table>
Introduction

Crater Detection
- Marked Point Process Model
- Energy Function
- Multiple Birth and Death Algorithm
- Region-of-Interest Approach
- Experimental Results

Image Registration
- 2-step Approach
- Experimental Results

Conclusion
Image Registration – 2-Step Optimization

Why a 2-step Optimization?

Feature-based registration
- Min Hausdorff distance ($d_H$) between extracted craters through genetic algorithm
- Fast but sensitive to accuracy of crater maps

Area-based registration
- Max Mutual Information ($MI$) through genetic algorithm
- Highly accurate but computationally heavy

RST (Rotation-Scale-Translation) transforms $p = (t_x, t_y, \theta, k)$

Initialization

Fast registration based on extracted craters $\rightarrow \hat{p}$

Refinement: registration based on mutual information in a neighborhood of $\hat{p} \rightarrow p^*$
Image Registration – Region of Interest

Transformation found for an interactively selected region of interest \( \rightarrow p^*_B \)
\[
p_B = (t_x, t_y, \theta, k)
\]

Transformation derived for the entire Image \( \rightarrow p^*_A \)
\[
p_A = (T_x, T_y, \beta, \alpha)
\]

\[
p^*_A = \begin{pmatrix}
-k \cos(\theta) x_0 - k \sin(\theta) y_0 + t_x + x_0 \\
k \sin(\theta) x_0 - k \cos(\theta) y_0 + t_y + y_0 \\
\theta \\
k
\end{pmatrix}
\]

Superposition of Reference and Input
Image Registration – Data Sets

Semi-simulated image pairs

20 pairs composed of one real THEMIS or HRSC image and of an image obtained by applying a synthetic transform and AWGN

Quantitative validation with respect to the true transform (RMSE)

Real multi-temporal image pairs

Real multi-temporal pair of LROC (Lunar Reconnaissance Orbiter Camera) images

100m resolution

Only qualitative visual analysis is available, as no ground truth is available
Registration Results with Semi-synthetic Data

<table>
<thead>
<tr>
<th>Data set</th>
<th>RMSE [pixel]</th>
<th>Left Image</th>
<th>Right Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEMIS (10 data sets)</td>
<td>0.31</td>
<td>$(7.05, 35.91, 0.18^\circ, 1.071)$</td>
<td>$(76.59, 19.96, 2.17^\circ, 1.031)$</td>
</tr>
<tr>
<td>HRSC (10 data sets)</td>
<td>0.22</td>
<td>$(7.04, 35.92, 0.19^\circ, 1.071)$</td>
<td>$(76.41, 20.06, 2.18^\circ, 1.031)$</td>
</tr>
<tr>
<td>Average (20 data sets)</td>
<td>0.26</td>
<td>RMSE 1(^{st}) Step 0.79</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RMSE 2(^{nd}) Step 0.16</td>
<td>0.33</td>
</tr>
</tbody>
</table>

RGB of original non-registered data
RGB of registered data
Registration Results with Real Data

Visually accurate matching between reference and registered images in the real multitemporal data set

Checkerboard representation of the registered images (zoom on details)
Registration Results with Real Data

Visually accurate matching between reference and registered images in the real multitemporal data set
Content

Introduction

Crater Detection
- Marked Point Process Model
- Energy Function
- Multiple Birth and Death Algorithm
- Region-of-Interest Approach
- Experimental Results

Image Registration
- 2-step Approach
- Experimental Results

Conclusion
Conclusions

• **Accurate** crater maps, useful for both image registration and planetary science, were obtained from data from different sensors.

• **Higher accuracy** as compared to previous work on crater detection (not shown for brevity)

• Reduced time for convergence thanks to a **region-based approach**

• **Sub-pixel accuracy and visual precision** in registration: effectiveness of the proposed 2-step registration method

Future Developments

• Test in conjunction with a **parallel** implementation (e.g. computer cluster)

• Validation with **multi-sensor** real images

• Extension to **other applications** requiring the extraction of ellipsoidal or circular features, e.g. optical Earth observation images or medical images
Short Bibliography

For each pixel in the image compute the Birth Probability as $\min\{\delta \cdot B(s), 1\}$, where:

$$B(s) = \frac{b(s)}{\sum_s b(s)}$$

Being $b(s)$ the Birth Map computed from the Canny Contour Map
For each ellipse $x_i$ in the configuration compute the **Death Probability** as $d(x_i)$, where

$$d(x_i) = \frac{\delta \cdot a(x_i)}{1 + \delta \cdot a(x_i)} \quad \text{and} \quad a(x_i) = e^{-\beta (U_L(x|\{x\setminus x_i\}|g) - U_L(x|g))} = e^{\beta \cdot U_L^i(x_i|g)}$$

The complete **Flowchart** of the **Death Step** is as follows:

1. Computation of the Energy Terms
2. Computation of Death Prob. based on **Likelihood**
3. Elimination of ellipses with prob. $d(x_i)$
4. Configuration refinement thanks to **Prior** term
Similarity Measures

Hausdorff Distance

\[ \text{Similarity} = \text{mean}_c \left\{ \sum_{i=1}^{N^c} \sum_{t=1}^{P} d_H(x_i^c, x_t) \right\} \]

\( c = \text{craters in Input Image} \)
\( N^c = \text{sum(pixels in crater c in Input Image)} \)
\( P = \text{sum(craters’border pixels in Ref Image)} \)
\( x_i^c = \text{coord of pixel i in crater c in Input Image} \)
\( x_t = \text{coord of pixel t in Ref Image’s craters} \)

Mutual Information

\[ MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x, y) \log \left( \frac{p_{X,Y}(x, y)}{p_X(x) p_Y(y)} \right) \]

\( X: \text{pixel intensity in Reference Image} \)
\( Y: \text{pixel intensity in Input Image} \)
\( p_X(x): \text{probability density function (pdf) of } X \)
\( p_Y(y): \text{probability density function (pdf) of } Y \)
\( p_{X,Y}(x, y): \text{joint pdf of } X \text{ and } Y \)

estimated through the corresponding image histograms
RST Transformation

Rotation – Scale – Translation Transformation

Transformation vector

\[ p = (t_x, t_y, \theta, k) \]

\( \{t_x, t_y\} \): Translations in \( x \) and \( y \)
\( \theta \): Rotation angle
\( k \): Scaling Factor

Matrix Formulation

\[ T_p(x, y) = \begin{pmatrix} k \cos(\theta) & k \sin(\theta) \\ -k \sin(\theta) & k \cos(\theta) \end{pmatrix} \begin{pmatrix} t_x \\ t_y \end{pmatrix} \]
Region of Interest Approach

$I_A(X, Y), I_B(x, y)$: Two Images

$I_B$: sub-image of $I_A$ such that $I_B(0,0) = I_A(x_0, y_0)$

$p_A = (T_X, T_Y, \beta, \alpha)$: RST transformation vector transforming $I_A$ into $I_A^{tr}$

$p_B = (t_x, t_y, \theta, k)$: RST transformation vector transforming $I_B$ into $I_B^{tr}$

$I_B^{tr}(0,0) = I_A^{tr}(x_0, y_0)$

Given:

Transformation: $p_B$

Reference of Region: $(x_0, y_0)$

Find: Transformation: $p_A$

Expressing the transformation in Matrix Form

From the image

\[
\begin{align*}
X &= x + x_0 \\
Y &= y + y_0
\end{align*}
\]

\[
T_{pA} = \begin{pmatrix}
\alpha \cos(\beta) & \alpha \sin(\beta) \\
-\alpha \sin(\beta) & \alpha \cos(\beta)
\end{pmatrix}
\begin{pmatrix}
T_X \\
T_Y
\end{pmatrix}
\]

$T_{pA}(X, Y) = (X', Y')$

\[
T_{pB} = \begin{pmatrix}
k \cos(\theta) & k \sin(\theta) \\
-k \sin(\theta) & k \cos(\theta)
\end{pmatrix}
\begin{pmatrix}
t_x \\
t_y
\end{pmatrix}
\]

$T_{pB}(x, y) = (x', y')$

This should also hold

$T_{pA}(x + x_0, y + y_0) = (x' + x_0, y' + y_0)$

Plugging $T_{pA}$ into this equation and replacing $x'$ and $y'$ according to $T_{pB}$

\[
\begin{align*}
\alpha \cos(\beta)(x + x_0) + \alpha \sin(\beta)(y + y_0) + T_x \\
-k \sin(\theta)(x + k \cos(\theta)y + t_y + y_0) =
\end{align*}
\]

Knowing $\alpha = k$ and solving in $P_1 = (0,0)$ and $P_2 = (-x_0, -y_0)$

\[
p_A = \begin{pmatrix}
-k \cos(\theta)(x_0 - k \sin(\theta)y_0 + t_x + x_0) \\
k \sin(\theta)(x_0 - k \cos(\theta)y_0 + t_y + y_0) \\
\theta \\
k
\end{pmatrix}
\]
RMS Error Computation

**Ground Truth Transformation**

\[ p_{GT} = (t_{x1}, t_{y1}, \theta_1, k_1) \rightarrow T_{p_{GT}}(x, y) = Q_{p_{GT}} \cdot [x, y, 1]^T \]

**Computed Transformation**

\[ p = (t_x, t_y, \theta, k) \rightarrow T_p(x, y) = Q_p \cdot [x, y, 1]^T \]

\((x, y) \in \text{Image}, [x', y', 1]^T = Q_{Pe} \cdot [x, y, 1]^T\)

**Error Transformation**

\[ p_e = (t_{xe}, t_{ye}, \theta_e, k_e) \rightarrow Q_{P_e} = Q_p \cdot Q_{p_{GT}}^{-1} \]

\[
\begin{cases}
    k_e = \frac{k_2}{k_1}, & \theta_e = \theta_2 - \theta_1 \\
    t_{xe} = t_{x2} - k_e(t_{x1} \cos(\theta_e) + t_{y1} \sin(\theta_e)) \\
    t_{ye} = t_{y2} - k_e(t_{y1} \cos(\theta_e) - t_{x1} \sin(\theta_e))
\end{cases}
\]

\[ [x'] = k_e \begin{pmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{pmatrix} [x] + [t_{xe}] \\
[y'] = k_e \begin{pmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{pmatrix} [y] + [t_{ye}] \]

**RMS Error:**

\[ E(p_e) = \sqrt{\frac{1}{AB} \int_0^A \int_0^B (x' - x)^2 + (y' - y)^2 \, dx \, dy}, \quad \alpha = A^2 + B^2 \]

\[ E^2(p_e) = \frac{1}{AB} \int_0^A \int_0^B (k_e \cos(\theta_e)x + k_e \sin(\theta_e)y + t_{xe} - x)^2 + (-k_e \sin(\theta_e)x + k_e \cos(\theta_e)y + t_{ye} - y)^2 \, dx \, dy \]

\[ E^2(p_e) = \alpha \left( k_e^2 - 2k_e \cos(\theta_e) + 1 \right) + (t_{xe}^2 + t_{ye}^2) - (A^2 x_{xe} + B^2 y_{ye})(1 - k_e \cos(\theta_e)) - k_e(A t_{ye} - B t_{xe}) \sin(\theta_e) \]