

Automated searching of Stardust interstellar foils

Ryan C. OGLIORE^{1*}, Christine FLOSS², Frank J. STADERMANN², A. T. KEARSLEY³,
Jan LEITNER⁴, Rhonda M. STROUD⁵, and Andrew J. WESTPHAL⁶

¹Hawai'i Institute of Geophysics and Planetology, University of Hawai'i at Mānoa, Honolulu, Hawai'i 96822, USA

²Laboratory for Space Sciences and Physics Department, Washington University, Saint Louis, Missouri 63130, USA

³Department of Mineralogy, The Natural History Museum, London SW7 5BD, UK

⁴Max Planck Institute for Chemistry, Particle Chemistry Department, P.O. Box 3060, 55020 Mainz, Germany

⁵Materials Science and Technology Division, U.S. Naval Research Laboratory, Washington, District of Columbia 20375, USA

⁶Space Sciences Laboratory, University of California at Berkeley, Berkeley, California 94720, USA

*Corresponding author. E-mail: ogliore@higp.hawaii.edu

(Received 15 June 2011; revision accepted 21 November 2011)

Abstract—The Al foils lining the aerogel tiles of the Stardust interstellar tray represent approximately 13% of the total collecting area, about 15,300 mm². Although the flux is poorly constrained, fewer than 100 impacts are expected in all the Al foils on the collector, and most of these are likely to be less than 1 μm in diameter. Secondary electron (SE) images of the foils at a resolution of approximately 50 nm per pixel are being collected during the Stardust Interstellar Preliminary Examination, resulting in more than two million images that will eventually need to be searched for impact craters. The unknown and complicated nature of 3-dimensional interstellar tracks in aerogel necessitated the use of a massively distributed human search to locate only a few interstellar tracks. The 2-dimensional nature of the SE images makes the problem of searching for craters tractable for algorithmic approaches. Using templates of craters from cometary impacts into Stardust foils, we present a computer algorithm for the identification of impact craters in the Stardust interstellar foils using normalized cross-correlation and template matching. We address the speed, sensitivity, and false-positive rate of the algorithm. The search algorithm can be adapted for use in other applications. The program is freely available for download at http://jake.ssl.berkeley.edu:8000/groups/westphalgroup/wiki/14e52/ISPE_SEM_Crater_Search.html.

INTRODUCTION

NASA's Stardust mission collected cometary particles from the Jupiter-family comet 81P/Wild 2 and returned them to Earth for laboratory analysis. In addition, the mission exposed a second collector to the interstellar dust stream for a total of 195 days in 2000 and 2002. Contemporary interstellar dust, its trajectory and abundance first identified in situ by Ulysses (Grün et al. 1993) and Galileo (Baguhl et al. 1995), is small and rare: approximately 60 impacts of particles < 1 μm and approximately 30 impacts > 1 μm were expected to be present on the returned Stardust interstellar collector (Landgraf et al. 1999). The total aerogel area of the collector is 1037 cm², an additional 153 cm² are provided by Al foils. Hypervelocity impacts into the aerogel

tiles produce three-dimensional “tracks” of complex morphology (e.g., Burchell et al. 2008). Interstellar tracks are expected to be only a few tens of micrometers in length. Manually searching for these small, scarce tracks in a large volume of aerogel would take many years for a few dedicated scientists. The problem was solved by recruiting amateurs to search for interstellar impact tracks using a web-based virtual microscope called Stardust@home (Westphal et al. 2006). This project has been successful, with the identification of possible interstellar tracks along with dozens of small tracks caused by secondary collisions of micrometeoroids with the spacecraft (Westphal et al. 2011).

Impacts into Al foil produce features that are easier to identify. For all but the most oblique angles, high-velocity impacts produce circular craters (Pierrazo and

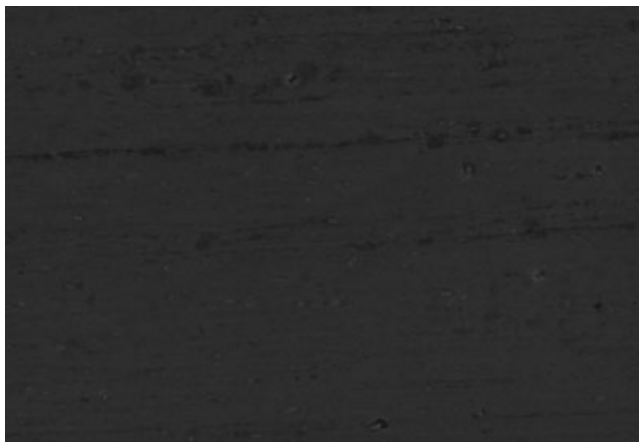


Fig. 1. Secondary electron image from Stardust cometary foil C2034N with no impact craters, covering $35 \times 24 \mu\text{m}$ of the Al foil.

Melosh 2000). Impact craters into the Stardust foils can be imaged by scanning electron microscopy (SEM). The task of identifying these craters is much simpler than that of identifying tracks in aerogel. Two-dimensional recognition of relatively consistent shapes can be done by computational techniques, eliminating the need for humans to manually look through two million SE images for impact craters (although humans are still better at identifying unusual features, as will be discussed later).

In this article, we will describe a computer program that identifies impact craters on SEM images of Stardust interstellar foils.

CRATERS IN STARDUST FOILS

The problem of feature identification is well explored in the computer science discipline of computer vision. The efficacy of various algorithms depends on the properties of the image to be searched and the type of features to be identified.

Samples collected in the Al foils on the Stardust interstellar and cometary collectors have useful characteristics distinct from aerogel-collected material (e.g., concentration of the sample in a small volume and the ability to be scanned by SEM). The foils are not completely pristine: they are typically covered with dust and scratches. Figure 1 shows a typical SE image of a Stardust interstellar foil. The algorithm to identify craters therefore must be robust against this kind of image noise else it will yield an unacceptably high false-positive rate.

Hypervelocity impacts into Al foil produce craters: a depression in the foil with a raised circular rim. The exact size and shape of the crater depend on the size, shape, and composition of the impactor (Kearsley et al. 2008). Impact craters are identifiable in secondary electron

imaging of the Al foils because of the topographic contrast caused by the raised rims and depressed centers of the craters. The rims look bright and the interior looks dark compared with the surrounding foil in the SE image. Some examples of SE crater images from the Al foils on the Stardust cometary collector are shown in Fig. 2.

THE HOUGH TRANSFORM AND NORMALIZED CROSS-CORRELATION

The Hough transform is a well-known tool in computer vision to identify particular shapes in digital images (Illingworth and Kittler 1988). Edges are located in the image, which are then mapped into a parameter space defined by the search shape. The clusters of points which yield similar parameters are then identified as the searched-for shape. In theory, the Hough transform could be used to find circles in the SE images that correspond to the circular rim of impact craters. When applied to SE crater images from Stardust cometary foils, however, the Hough transform had poor sensitivity and a very high false-positive rate: somewhat circularly shaped scratches and dust were often located, whereas many craters with nonideal circular shapes were not identified. After much experimentation with the Hough transform (which has the advantage of being very fast) on Stardust cometary SEM foil images, we determined that it could not be used for crater identification.

The craters on the Stardust cometary foils vary in shape and can deviate significantly from perfect circles (Kearsley et al. 2008). For this reason, and because of the failure of the Hough transform, we decided that a template-matching algorithm would be the most robust way to identify crater-like features in the Stardust interstellar foils. Template-matching algorithms seek to answer the question: Does the image to be searched contain the template, and if it does, what is the template's location in the image?

The correlation between two images is employed for feature detection using template matching (Duda and Hart 1973). Basically, cross-correlation template matching computes the Euclidean distance between the pixel values of the template and the pixel values of a subregion of the image, summed over the size of the template. The cross-correlation term is defined as:

$$c(u, v) = \sum_{x,y} I(x+u, y+v)T(x, y) \quad (1)$$

where I is the image and T is the template. The indices x and y are valid over the size of the template; the indices u and v are valid over the size of the image minus the size of the template. The image and template are both assumed to be one-color (grayscale) images.



Fig. 2. Secondary electron images of four Stardust cometary craters with diameters (left to right): 500 nm, 500 nm, 250 nm, 750 nm.

Cross-correlation can fail if the brightness of the image varies across the image to be searched: a good match between a feature in the image and the template can have a lower cross-correlation than a part of the image, which simply has higher pixel values. This can be seen clearly in Equation 1. Because of this, we normalize the cross-correlation so that the comparison between the image and template corrects to zero mean and unit length:

$$\text{NC}(u, v) = \frac{\sum_{x,y} (I(x+u, y+v) - \bar{I}_{u,v})(T(x,y) - \bar{T})}{\sqrt{\sum_{x,y} (I(x+u, y+v) - \bar{I}_{u,v})^2 \sum_{x,y} (T(x,y) - \bar{T})^2}} \quad (2)$$

This quantity is more computationally expensive than the unnormalized cross-correlation, but is necessary for searching SE images, which have variable brightness. The numerical values returned by normalized cross-correlation vary from -1 to 1 .

THE CRATER TEMPLATE LIBRARY

For a template-matching algorithm to be robust and effective at identifying a diverse set of features, its library must include the full range of features that are sought. Therefore, we must compile a large library of craters that spans the variety of crater-like features that we wish the algorithm to identify in the Stardust interstellar foils.

A crater's distinguishing features in an SE image are a bright rim with a dark interior. The smallest crater therefore is five pixels in width and height: gray-white-black-white-gray in a sequence of pixels. As craters get larger and cover more pixels, they can become more diverse. However, we use the fact that a large crater that is, say, 20 pixels across, still looks like a crater when the image is downsampled by a factor of two in each dimension. We can therefore constrain the size of our library needed to cover the diversity of potential craters by considering only craters smaller than 11 pixels. We downsample craters larger than this by bilinear interpolation until they are less than 11 pixels in their smallest dimension (the template craters are all nearly circular). To search for craters larger than 11 pixels in an

SE image, we downsample the image by a factor of 2, 4, and 8 in each dimension and search each of these downsampled images using the library of small craters. As template-matching by normalized cross-correlation (NCC) scales by the size of the image searched, searching these downsampled images adds little overhead to the computation time (only 33%), while greatly increasing the robustness of the algorithm in finding possible crater matches by averaging over the diversity in larger craters before comparing with the template library. Nonetheless, if the user wishes to search for only small craters (less than 11 pixels), there is an option to turn off the large crater search.

The interstellar dust grains impacting the Stardust collector are expected to be smaller and to have higher impact speed than the cometary dust grains (Landgraf et al. 1999). However, the impacts of interstellar dust into Al foils should still create circular craters, similar in shape (but not necessarily size) to the cometary craters. We assume that the range of crater morphologies is spanned by the set of cometary craters. In addition, we include interstellar analog craters created by high-speed shots from a Van de Graaff accelerator (Postberg et al. 2010). As craters are discovered on the Stardust interstellar foils (whether by automated or human searching), these can easily be added to the crater template library.

Normalized cross-correlation is not robust against image rotation: a crater template rotated 90° will have a different NCC value against an image compared with the unrotated crater. Therefore, we rotate each crater by 90° , 180° , and 270° and add each of these rotations to the crater template library.

Each time an image is searched by NCC using a given template, a new “image” is created with a size that is smaller than the original in each dimension by the size of the template, and with “pixel” values ranging from -1 to 1 . A perfect normalized cross-correlation has a value of 1 (a perfect normalized anticorrelation is -1), a good match to the template has a value less than 1 , but greater than some threshold value. How do we determine this threshold value?

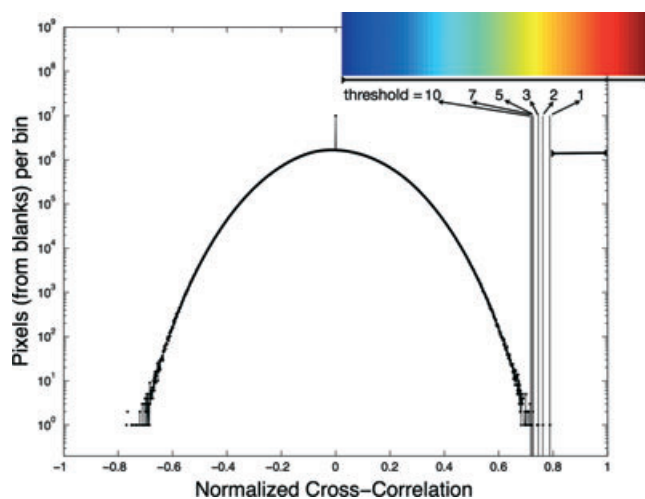


Fig. 3. Histogrammed normalized cross-correlation values for a crater template over 332 Stardust foil images that contain no craters. The strictest threshold (1) is the highest NCC value in any of the blank images (0.788). More false-positives are allowed for lower thresholds at the 2nd, 3rd, 5th, 7th, and 10th highest NCC value in the blanks. The color bar indicates the strength of a “hit” for the strictest threshold: a NCC value just above threshold is circled in blue on the output image, a value close to one is circled in red.

Using a series of randomly chosen blanks (SE images of the Stardust foils without craters as determined by human inspection), we determined the distribution of NCC values for a given crater template over more than 300 blanks. Due to debris and scratches on the foil, some regions of these blanks will yield high NCC values for the template. Our strictest threshold level is the value of the single highest NCC value recorded over all of the blanks (the maximum blank NCC value varies between approximately 0.75 and approximately 0.95 for the 2156 crater templates). This threshold level will yield the fewest false positives, but also the highest chance of missing a crater-like feature (low sensitivity). The lower threshold levels (higher false-positives, higher sensitivity) are where the 2nd, 3rd, 5th, 7th, and 10th highest NCC values (chosen to span the range of useful sensitivity and false-positive rate) are obtained for the SE blanks. See Fig. 3 for an illustration of the threshold determination for a crater in the template library.

OPTIMIZATION OF THE LIBRARY

The time to search a given SEM image scales linearly with the size of the library because each crater template has to be compared with the search image individually by normalized cross-correlation. A large library increases the chance that an interesting feature will be identified by

the algorithm, but also increases the computation time needed to perform the search. Therefore, it is beneficial to construct the optimum library of a given size. The most optimum subset of n templates from a master library of N templates is that subset which yields a normalized cross-correlation value greater than the threshold value for the most number of templates in the master library. To illustrate, a subset of 50 craters is matched, via normalized cross-correlation, against the master library of 1000 craters. This generates 1000 NCC values for each template in the master library. Each of these 1000 craters has a threshold value above which is considered a good match, a “hit.” The number of these 1000 NCC values which are above the threshold value is the number of templates fit by this subset of 50 craters; we call this value N^* . We seek to maximize N^* for a library of a given size and threshold level.

We use the genetic algorithm (Goldberg 1989) to solve this optimization problem. The genetic algorithm uses principles of genetic evolution to find a global solution to an optimization problem without the normal requirements of the functions involved to be differentiable or continuous. Other optimization algorithms we explored either did not find a global solution, failed to converge, or were too slow for this particular problem. We seek optimal sublibraries of 100, 250, 500, 1000, and 1500 crater templates from the master library of 2156 craters using the threshold levels of 1, 2, 3, 5, 7, and 10 as described above. We start with a population of 1000 random 100-template subsets of the 2156 master library. The fitness function that the genetic algorithm will maximize is the N^* value (number of matching templates above their threshold) of the subset. The genetic algorithm selects parents for the next generation from those 1000 possible sublibraries which have the higher N^* values. The next generation is then created by crossover (mixing individual crater templates from a pair of parents), mutation (randomly switching a template out for another from the master library), and direct inheritance of the templates with the highest N^* values. We experimented with the mutation rate, fraction of craters for crossover, number of direct inheritance, and several other parameters of the genetic algorithm to arrive at a suitably robust and efficient optimization routine for this problem. It required 3 days of computer time utilizing four processor cores on a circa-2009 Linux machine to compute the five crater-template sublibraries for each of the six threshold levels. An example of an optimized crater library is shown in Fig. 4. As the threshold level goes from 1 to 10, N^* approaches 2156. Once $N^* = 2156$, the algorithm has acquired its maximum fitness, and relaxing the detection threshold will not change the templates selected for the sublibrary. The sublibrary with 500 templates, for example, acquires

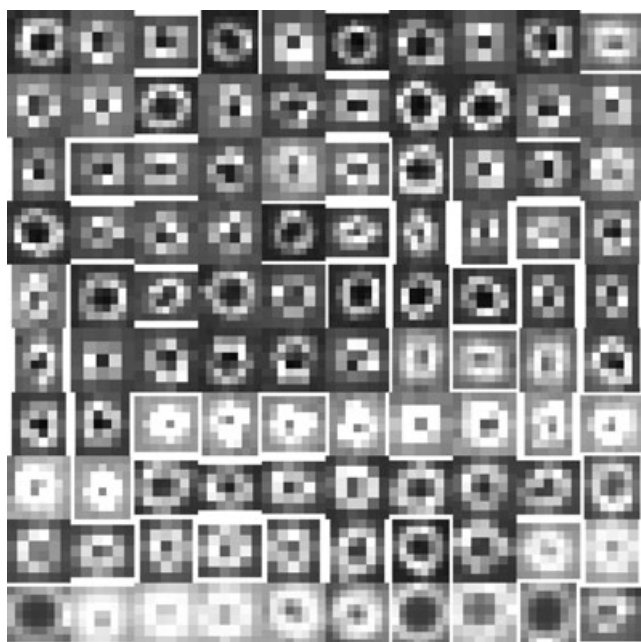


Fig. 4. Sample library of crater templates to be used for searching Stardust interstellar foils. This library has 100 templates and the highest threshold from the analyzed blank foils.

$N^* = 2156$ when the threshold level is 2, so the 3, 5, 7, and 10 threshold libraries are not computed.

IMPLEMENTATION OF THE ALGORITHM

We implemented the normalized cross-correlation and template-matching algorithm in Matlab, a high-level interpreted language. As nearly the entire computational cost of the template-matching routine is in the normalized cross-correlation, we chose to implement an exact normalized cross-correlation routine that is significantly faster than Matlab's native *normxcorr2*. This routine uses compiled C++ code from OpenCV (<http://sourceforge.net/projects/opencvlibrary/>) and is called directly from Matlab via a MEX file written by Daniel Eaton (http://www.cs.ubc.ca/~deaton/remarks_ncc.html), which results in an approximately 50% increase in speed compared with Matlab's native routine.

The code is compiled as an executable for Mac and Windows, called Crater Finder. The user selects the library size and threshold value, whether to search for large craters via the down-sampling routine as described above, and a folder containing the images to be searched. The program then matches each template in the chosen library with the search image and records matches that are above the threshold level for that template. After every template has been matched across the full-size and

down-sampled images, the template hits are tallied for that SE image. The strength of the NCC is recorded as the "score":

$$\text{Score} = \frac{\text{NCC} - \text{threshold}}{1 - \text{threshold}} \quad (3)$$

For each "hit," the score varies from 0 (equal to the threshold value) to 1 (the maximum NCC value, a perfect match to the template). For each hit, a circle is drawn around the identified feature on the SE image that corresponds to the score, where the color of the circle varies from blue (score = 0) to red (score = 1) using Matlab's "jet" color map. For a feature on the SE image that has a "hit" from two or more templates, the scores are summed. For each image, the output from Crater Finder consists of the following: a copy of the original SE image with the identified features circled in color corresponding to the score, image crops of the identified features, a text file listing the x - and y - pixel locations of the identified features, their score, pixel radius, the name of the matched template, and the factor at which the original image was down-sampled to find the match. After all the images in a given directory have been searched, a final text file compiling all the results is written containing the information described above, with the "hits" listed in descending order by their score. In addition, a pdf file with a cropped image of the feature and the template that matched to it, also in descending order by score and containing the information in the summary text file, is written to the output directory. This makes it convenient to manually examine the crater features with the highest scores. All of these files are written to a subdirectory inside the directory of the original SE images.

The program is available for download here: http://jake.ssl.berkeley.edu:8000/groups/westphalgroup/wiki/14e52/ISPE_SEM_Crater_Search.html

PERFORMANCE OF THE CRATER FINDER PROGRAM

To evaluate the performance of the crater-finding algorithm, we assume that the diversity in the "unknown" crater population is reflected in the diversity of the craters in our library. With these assumptions, along with blank SE images (as searched by humans), we can estimate the sensitivity of a given sublibrary (the percentage of craters of the master catalog that would be identified as "hits" by the algorithm) and the false-positive rate per 3 megapixel (e.g., 2048×1536) image (determined by using the crater library to search the set of known blanks). The sensitivity and false positive rate will depend on the library size (100, 250, 500, 1000, 1500,

Table 1. Timing, sensitivity, and false-positive rate of the crater libraries using Crater Finder v2.1. Search times are for a 2010 MacBook Pro (Intel Core i5, 4 GB RAM) and a 3 megapixel image. Libraries are only listed for threshold = 1 until the threshold where sensitivity achieves 100% (all craters in the master library are fit above threshold by the sublibrary) although thresholds of 1, 2, 3, 5, 7, and 10 are available for use with all library sizes.

Library size	Threshold	Min/image	Sensitivity (%)	False-positive rate
100	1	1.3	74	1.7
100	2	1.3	79	3.5
100	3	1.3	83	5.2
100	5	1.3	87	8.7
100	7	1.3	90	12
100	10	1.3	92	18
250	1	3.1	85	4.3
250	2	3.1	89	8.7
250	3	3.1	93	13
250	5	3.1	95	22
250	7	3.1	98	30
250	10	3.1	99	43
500	1	6.0	96	8.7
500	2	6.0	100	17
1000	1	11.7	100	17
1500	1	17.4	100	26
2156	1	25.0	100	37

and 2156) and the threshold level (1, 2, 3, 5, 7, and 10). These results are reported in Table 1.

We can also test the performance of the algorithm by searching foil images before and after analog impacts created with a Van de Graaff accelerator. Such a test has not yet been performed.

In addition, the performance of the algorithm depends on how quickly it is able to perform a search. Although computers are becoming cheaper and faster, the algorithm must be able to search for craters in a reasonable time frame on affordable computer resources. Timing tests were performed on ten 2048×1536 SE images on a 2010 MacBook Pro (2.4 GHz Intel Core i5, 4 GB RAM) for a given library size (threshold level does not affect the search time). The timing results are also given in Table 1.

Secondary electron images to be searched should be either TIF or very low-compression JPEG, as compression artifacts could compromise the template-matching procedure.

The crater-finding algorithm typically finds >90% of the cometary craters identified by humans when used on craters that have not been incorporated into the library. Figure 5 shows an example. The algorithm will also find interesting features not identified by humans, and a few false-positive hits per three megapixel image. The false positives almost always have low scores (circled by blue in the output image), which makes it easy to screen them out using the cumulative data file and summary document written by Crater Finder. Further examples of Crater Finder versus human searching of Stardust foils are posted on: [http://jake.ssl.berkeley.](http://jake.ssl.berkeley.edu:8000/groups/westphalgroup/wiki/14e52/ISPE_SEM_Crater_Search.html)

[edu:8000/groups/westphalgroup/wiki/14e52/ISPE_SEM_Crater_Search.html](http://jake.ssl.berkeley.edu:8000/groups/westphalgroup/wiki/14e52/ISPE_SEM_Crater_Search.html)

The code, as currently compiled, does not take advantage of multiple processor cores; however, multiple instances of Crater Finder can run with the searchable images broken up into multiple subdirectories.

The total number of Stardust interstellar foil images to be searched is more than two million. Using the 500-crater library, it would take approximately 23 yr per core of computation time to search the entire Stardust Interstellar foil collection. This sounds daunting, but with the accessibility of large Linux clusters, the computation time could be reduced to a few months on a 100-core cluster. (Compiling a Linux version of Crater Finder to be used on a Linux cluster is relatively straightforward.) In addition, rapid improvements in computing power over the next several years will decrease the needed computation time substantially. Scanning the foils by SEM is itself a time-consuming process, and this may be the rate-limiting step. The searching can be optimized if images from one foil can be searched while another foil is being scanned. This could make the Herculean task of searching SE images of the entire Stardust ISPE collector at 50 nm per pixel resolution a tractable problem.

CONCLUSIONS

We have described a computer program for the identification of crater-like features on Stardust interstellar foils. The algorithm has proven to be effective so far in re-identifying human-identified craters on the

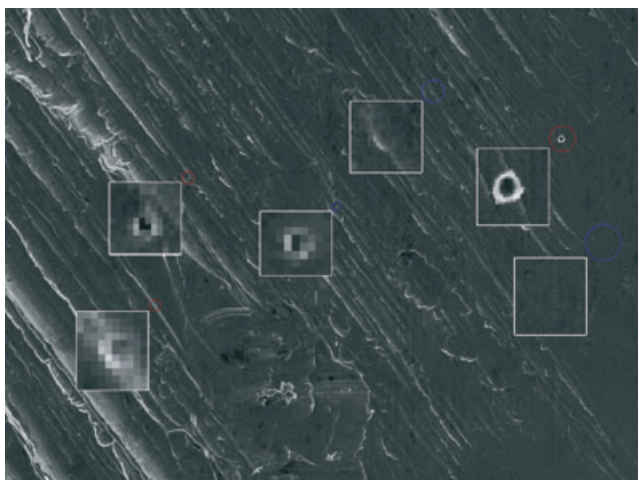


Fig. 5. Secondary electron image of a portion of a Stardust cometary foil (55 μm wide) showing five craters identified by Crater Finder v2.1, using the 500 template sublibrary, with threshold level 1. Enlargements of the identified features are shown in adjacent white rectangles. The program identified three features with high scores (red circles) and three features with low scores (blue circles). The three red-circled features were identified as craters by human inspection. The feature in the smallest blue circle appears to be a very small crater; the other two blue circles are certainly not craters.

Stardust cometary foils, and identifying craters in foils during the Stardust Interstellar Preliminary Examination. The user can select a balance between computation time, sensitivity, and false-positive rate by choosing the appropriate library size and detection threshold. The program is fast enough to search a small number of foils on reasonable time scales, and would be able to search foils from a large number of foils efficiently if deployed on a Linux cluster.

The Crater Finder software is only as good as its library, so we will continually update the library with more craters from the Stardust cometary collector foils and interstellar foils as they are discovered. In addition, verified blanks will help determine the appropriate threshold levels more accurately.

The algorithm used here is general enough to be used in other applications where template-matching is an appropriate search routine. We have shown that the genetic algorithm is a useful routine to find the optimal library, a critical requirement for an efficient template-matching algorithm.

It is important to remember that this routine, and any computational method, will not be as good as a human at identifying unusual but interesting features. Template-matching will only identify features when we know what we are looking for, but in any field, it is often the unexpected that is the most exciting discovery. Human searching is slow, inexact, and tedious—but a

human has the capability to identify a feature that is not a part of the template library, whereas a computer is fast, exact, and does not suffer from boredom or exhaustion, but will miss an interesting feature that is not a part of its library. Perhaps a human-computer hybrid approach will work best—use a computer to identify the obvious crater-like features, and use humans to search a subset of the foil images for interesting features that we may not know we are looking for. These human-identified features then can be added to the library, and the search can be iterated in this method to identify all of the interesting features on the Stardust interstellar foils.

Editorial Handling—Dr. John Bradley

REFERENCES

- Baguhl M., Hamilton D., Grün E., Dermott S. F., Fechtig H., Hanner M. S., Kissel J., Lindbland B.-A., Linkert D., Linkert G., Mann I., McDonnell J. A. M., Morfill G. E., Polansky C., Riemann R., Schwehm G., Staubach P., and Zook H. A. 1995. Dust measurements at high ecliptic latitudes. *Science* 268:1016–1019.
- Burchell M. J., Fairey S. A. J., Wozniakiewicz P., Brownlee D. E., Hörz F., Kearsley A. T., See T. H., Tsou P., Westphal A., Green S. F., Trigo-Rodríguez J. M., and Dominguez G. 2008. Characteristics of cometary dust tracks in Stardust aerogel and laboratory calibrations. *Meteoritics & Planetary Science* 43:23–40.
- Duda R. O. and Hart P. E. 1973. *Pattern classification and scene analysis*. New York: Wiley. 339 p.
- Goldberg D. E. 1989. *Genetic algorithms in search, optimization & machine learning*. Reading, MA: Addison-Wesley. 432 p.
- Grün E., Zook H. A., Baguhl M., Balogh A., Bame S. J., Fechtig H., Forsyth R., Hanner M. S., Horanyi M., Kissel J., Lindbland B.-A., Linkert D., Linkert G., Mann I., McDonnell J. A. M., Morfill G. E., Phillips J. L., Polansky C., Schwehm G., Siddique N., Staubach P., Svetska J., and Taylor A. 1993. Discovery of jovian dust streams and interstellar grains by the Ulysses spacecraft. *Nature* 362:428–430.
- Illingworth J. and Kittler J. 1988. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing* 44:87–116.
- Kearsley A. T., Borg J., Graham G. A., Burchell M. J., Cole M. J., Leroux H., Bridges J. C., Hörz F., Wozniakiewicz P. J., Bland P. A., Bradley J. P., Dai Z. R., Teslich N., See T., Hoppe P., Heck P. R., Huth J., Stadermann F. J., Floss C., Marhas K., Stephan T., and Leitner J. 2008. Dust from comet Wild 2: Interpreting particle size, shape, structure, and composition from impact features on the Stardust aluminum foils. *Meteoritics & Planetary Science* 43:41–73.
- Landgraf M., Müller M., and Grün E. 1999. Prediction of the in-situ dust measurements of the Stardust mission to comet 81P/Wild 2. *Planetary and Space Science* 47:1029–1050.
- Pierrazo E. and Melosh H. J. 2000. Understanding oblique impacts from experiments, observations, and modeling. *Annual Review of Earth and Planetary Sciences* 28:141–167.
- Postberg F., Srama R., Trieloff M., Hillier J., Gainsforth Z., Westphal A., Grün E., Armes S., and Kearsley A. 2010. High velocity Van de Graaff shots of mineral dust:

- Application to STARDUST and other in situ space missions. *Geophysical Research Abstracts: EGU General Assembly* 2010:10935.
- Westphal A. J., von Korff J., Anderson D. P., Alexander A., Betts B., Brownlee D. E., Butterworth A. L., Craig N., Gainsforth Z., Mendez B., See T., Snead C. J., Srama R., Tsitrin S., Warren J., and Zolensky M. 2006. Stardust@home: Virtual microscope validation and first results (abstract #2225). 37th Lunar and Planetary Science Conference. CD-ROM.
- Westphal A. J., Allen C., Anderson D., Bajt S., Bechtel H. A., Borg J., Brenker F., Bridges J., Brownlee D. E., Burchell M., Burghammer M., Butterworth A. L., Cloetens P., Davis A. M., Floss C., Flynn G. J., Frank D., Gainsforth Z., Grün E., Heck P. R., Hillier J. K., Hoppe P., Howard L., Huss G. R., Huth J., Kearsley A., King A. J., Lai B., Leitner J., Lemelle L., Leroux H., Lettieri R., Lyverse P., Marchant W., Nittler L. R., Ogliore R. C., Postberg F., Price M. C., Sandford S. A., Sans Tresseras J. A., Schmitz S., Schoonjans T., Silversmit G., Simionovici A., Srama R., Stadermann F. J., Stephan T., Stodolna J., Stroud R. M., Sutton S. R., Toucoulou R., Trierloff M., Tsou P., Tsuchiyama A., Tyliczszak T., Vekemans B., Vincze L., von Korff J., Zevin D., and Zolensky M. E. 2011. 29,000 Stardust@home dusters 2011 constraints on the interstellar dust flux based on Stardust@home search results (abstract #2059). 42nd Lunar and Planetary Science Conference. CD-ROM.
-