ARL-TN-0888 • June 2018

US Army Research Laboratory

Using Self-Organizing Maps for In-situ Monitoring of Melt Pool Thermal Profiles for Porosity Prediction in Laser-Based Additive Manufacturing Processes

by Mojtaba Khanzadeh, Sudipta Chowdhury, Mohammad Marufuzzaman, Mark A Tschopp, and Linkan Bian

NOTICES

Disclaimers

ARL-TN-0888 ● JUNE 2018

US Army Research Laboratory

Using Self-Organizing Maps for In-situ Monitoring of Melt Pool Thermal Profiles for Porosity Prediction in Laser-Based Additive Manufacturing Processes

by Mojtaba Khanzadeh, Sudipta Chowdhury, Mohammad Marufuzzaman, and Linkan Bian
Department of Industrial and Systems Engineering, Mississippi State University, Starkville, MS

Mark A Tschopp
Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**			
1. REPORT DATE *(DD-MM-YYYY)* June 2018	**2. REPORT TYPE** Technical Note		**3. DATES COVERED (From - To)** January 2017–December 2017
4. TITLE AND SUBTITLE Using Self-Organizing Maps for In-situ Monitoring of Melt Pool Thermal Profiles for Porosity Prediction in Laser-Based Additive Manufacturing Processes			**5a. CONTRACT NUMBER**
			5b. GRANT NUMBER
			5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Mojtaba Khanzadeh, Sudipta Chowdhury, Mohammad Marufuzzaman, Mark A Tschopp, and Linkan Bian			**5d. PROJECT NUMBER**
			5e. TASK NUMBER
			5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Army Research Laboratory ATTN: RDRL-WMM-F Aberdeen Proving Ground, MD 21005-5069			**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TN-0888
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			**10. SPONSOR/MONITOR'S ACRONYM(S)**
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			
13. SUPPLEMENTARY NOTES			
14. ABSTRACT The objective of this technical note is to use unsupervised machine learning to characterize the underlying thermophysical dynamics of laser-based additive manufacturing (LBAM) captured by melt pool signals to predict porosity during the build. Herein, a novel porosity detection method is proposed based on morphological features and the temperature distribution of the top surface of the melt pool as the LBAM part is being built. Self-organizing maps (SOMs) are then used to further analyze the 2D melt pool dataset to identify similar and dissimilar melt pools. The significance of the proposed methodology based on melt pool profile is that this may lead the way toward in-situ monitoring to minimize or eliminate pores within LBAM parts.			
15. SUBJECT TERMS MATLAB, unsupervised machine learning, self-organizing maps, additive manufacturing, in-situ process monitoring			

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Mark A Tschopp
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	31	**19b. TELEPHONE NUMBER (Include area code)** 410-306-0855

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

1. Introduction

One of the major challenges of implementing laser-based additive manufacturing (LBAM) is the lack of understanding of its underlying process-structure-property relationship.[1–3] Due to this, LBAM is still not widely accepted and is often considered inconsistent and unreliable for many industrial applications. Modeling the melt pool boundaries and correlating the extracted features to microstructure anomalies may lead to real-time, nondestructive detection of part defects. Most of the existing methods for quantifying and characterizing the time-varying melt pool can be broadly categorized into 2 groups: 1) quantifying and characterizing melt pool using morphological characteristics, and 2) quantifying and characterizing melt pool using temperature characteristics. The former method utilizes morphological data to characterize and quantify the melt pool,[4,5] whereas the later utilizes thermal distribution data to characterize melt pool.[6,7] However, these methods are developed based on simple physics-based differential equations that govern the underlying thermophysical process.[8,9] Hence, LBAM processes need a data-driven modeling scheme that is able to signal the formation of porosity through quantifying and characterizing the melt pool temperature distribution based on thermal imaging data streams during the building of additive manufacturing (AM) parts.

In this technical note, a number of different measures implemented in MATLAB are used to quantify and characterize the time-varying melt pool temperature distribution from thermal imaging data streams to identify the relationship between melt pool characteristics and pores in the fabricated part.[10–12] The MATLAB scripts are attached as appendixes and a description of the methodology behind the MATLAB script is described within this document.

2. Methodology

We propose a modeling procedure that converts melt thermal images to continuous temperature models with the identical function support. Image processing is applied to extract temperature distribution of the top surface of each melt pool from the captured melt pool images. Extracted data points are converted to spherical coordinates. Subsequently, a nonparametric surface interpolation method is implemented on the extracted temperature distribution of the top surface of each melt pool. The resulting continuous temperature models with identical function supports will be used as input for melt pool clustering via a self-organizing map (SOM). The SOM is an unsupervised machine learning technique, which has previously been used for characterizing the geometric accuracy of additively manufactured components.[13,14]

3. Code Description and Usage

The algorithm used to predict porosity in AM parts is divided into 5 parts that are placed in Appendixes A, B, C, D, and E, respectively. Appendix A demonstrates a MATLAB script that contains all the MATLAB functions used in the code. Using these functions, a step by step implementation of the algorithm is also provided in Appendix A. There are 4 functions used in Appendix A: melt pool data acquisition, melt pool data transformation, melt pool interpolation, and SOM analysis. Implementation of melt pool data acquisition, melt pool data transformation, melt pool interpolation, and SOM analysis functions are illustrated in Appendixes B, C, D, and E, respectively. The MATLAB script in Appendix B extracts the temperature distribution of the melt pool top surface based on the melting temperature of fabricated materials. For each point of extracted contour, a corresponding temperature is assigned. Since the image data are still ill-structured due to nonfunctional form, a spherical transformation is applied, which is shown in the MATLAB script in Appendix C. Appendix D executes the bi-harmonic surface interpolation method to the extracted temperature distribution of the melt pool boundary in the spherical domain obtained from the MATLAB code in Appendix C. A 2D map is also defined to extract the same number of melt pool temperature distribution features. These homogenized features are the output of the script in Appendix D that are fed to the MATLAB script in Appendix E. Appendix E encompasses the MATLAB script of SOM that produces the SOM hit map and the correlation matrix among all clusters.

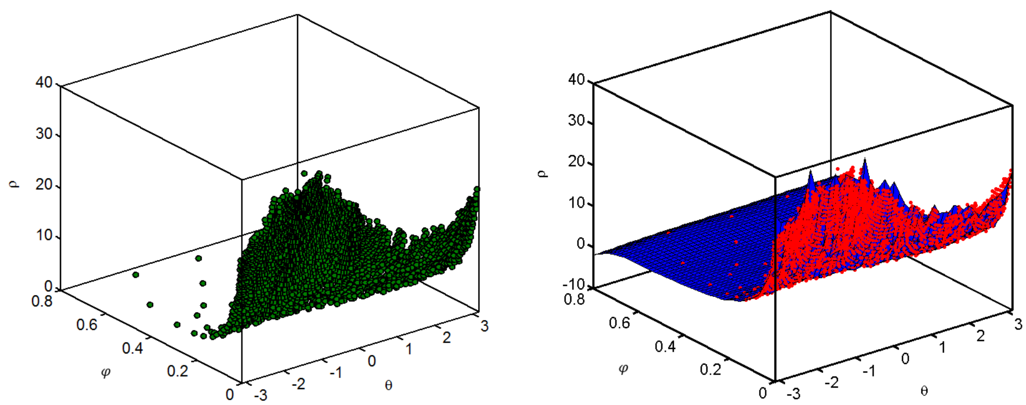
The attached scripts have been tested on MATLAB R2015a on 3 Windows operating systems. The code can be executed by the following steps:

Step 1: Download the 5 scripts from the Appendixes in the same folder and create the required datasets.

Step 2: Open the script "ARL_FIRST.m" in MATLAB and run it. It will call and read the necessary functions required to run the script and generate SOM hit maps and the correlation matrix.

4. Examples

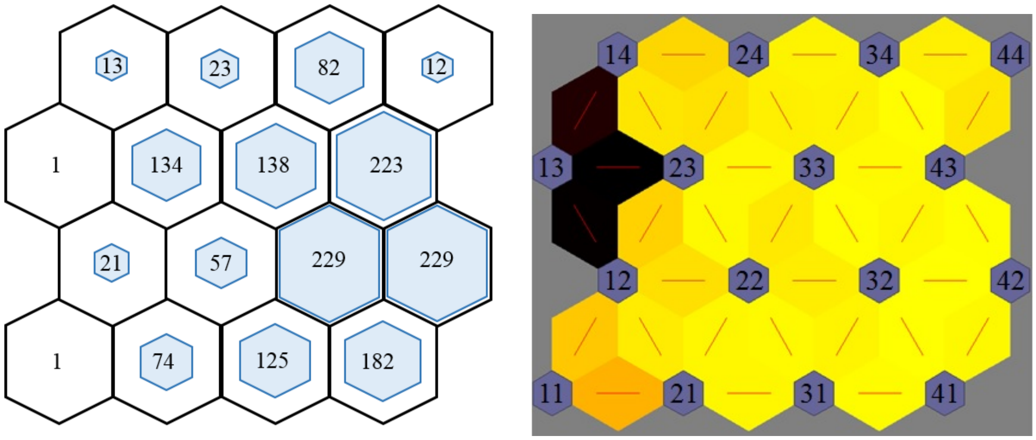
Centralized coordinates of each melt pool as well as their scaled temperature are spherically transformed to apply the tool of functional data analysis. The visualization of the temperature distribution of the top surface of the melt pool boundary with spherical coordinates is shown in Fig. 1a. The bi-harmonic surface fitting method is then applied for data interpolation and smoothing, which converts the discrete data points of spherical domain to a continuous function as shown in Fig. 1b.



(a) (b)

Fig. 1 Plots of (a) the temperature distribution at the top surface of the melt pool and (b) the interpolated temperature distribution of the melt pool using the bi-harmonic surface interpolation method. Both plots are in spherical coordinates.

Finally, as an example, a 4×4 SOM map is created leveraging the standardized vectors of temperature measurements of the same length and defined on the identical function support that is shown in Fig. 2. More specifically, Fig. 2a provides a graphical representation of an SOM hit map demonstrating the population size of each cluster. Figure 2b indicates the neighbor weight distances among these clusters. The darker color in Fig. 2b means cluster dissimilarity whereas lighter color means the opposite.



(a) (b)

Fig. 2 SOM for the proposed melt pool model, which includes (a) mapping each melt pool profile into a different cluster and (b) weighting the distances between neighboring clusters

It is assumed that 1) an abnormal melt pool has low correlation with others, or 2) the percentage of abnormal melt pool is much smaller compared with normal melt pools. Acknowledging this, an example of the correlation matrix of a 4×4 SOM map is shown in Fig. 3, which is a 16×16 symmetric matrix. The horizontal and vertical axes represent the labels of clusters. The red color represents higher correlation, whereas the blue color corresponds to lower correlation. A cluster with low correlation to the others is considered an anomaly, and porosity tends to occur at the corresponding locations.

For example, cluster $C21$ is a normal melt pool cluster, which includes a large number of melt pools highly correlated with other clusters. On the other hand, clusters $C13$ and $C44$ have low correlations with the other clusters. A melt pool in cluster $C13$ shows higher temperature and irregular shape. Similar overheating is observed in the melt pool images belonging to cluster $C44$. A visualization of these melt pools is provided in Fig. 4.

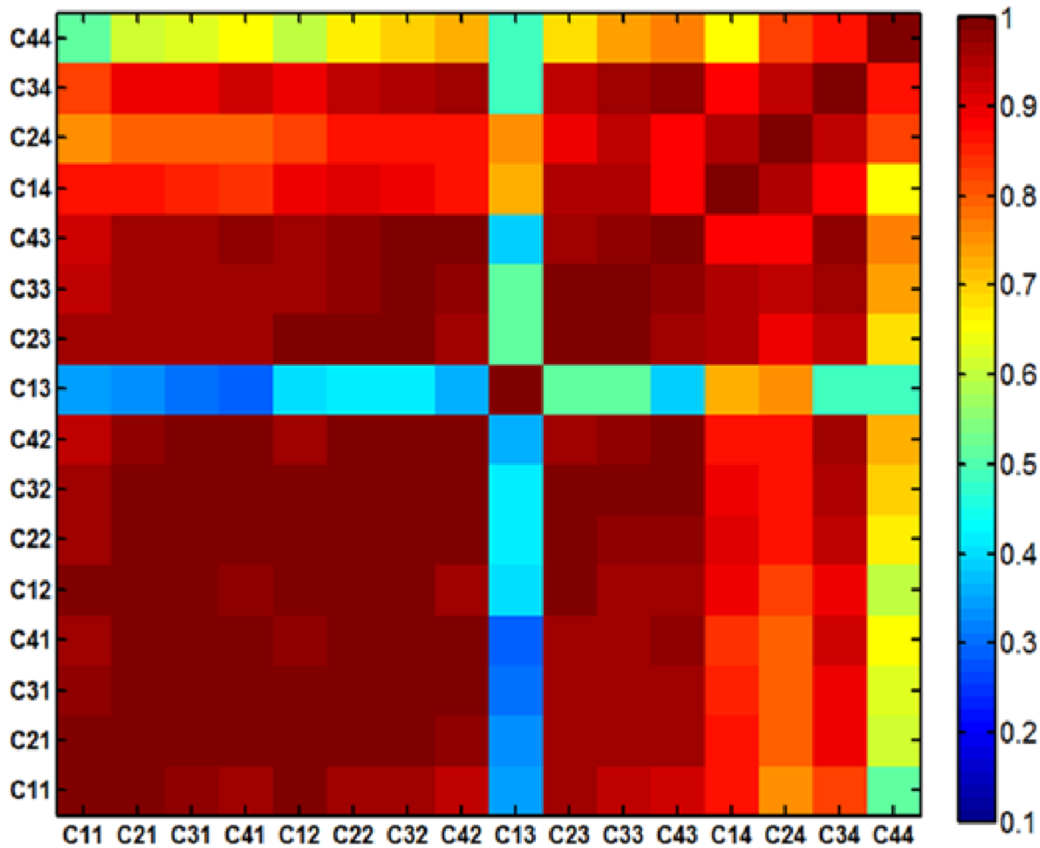
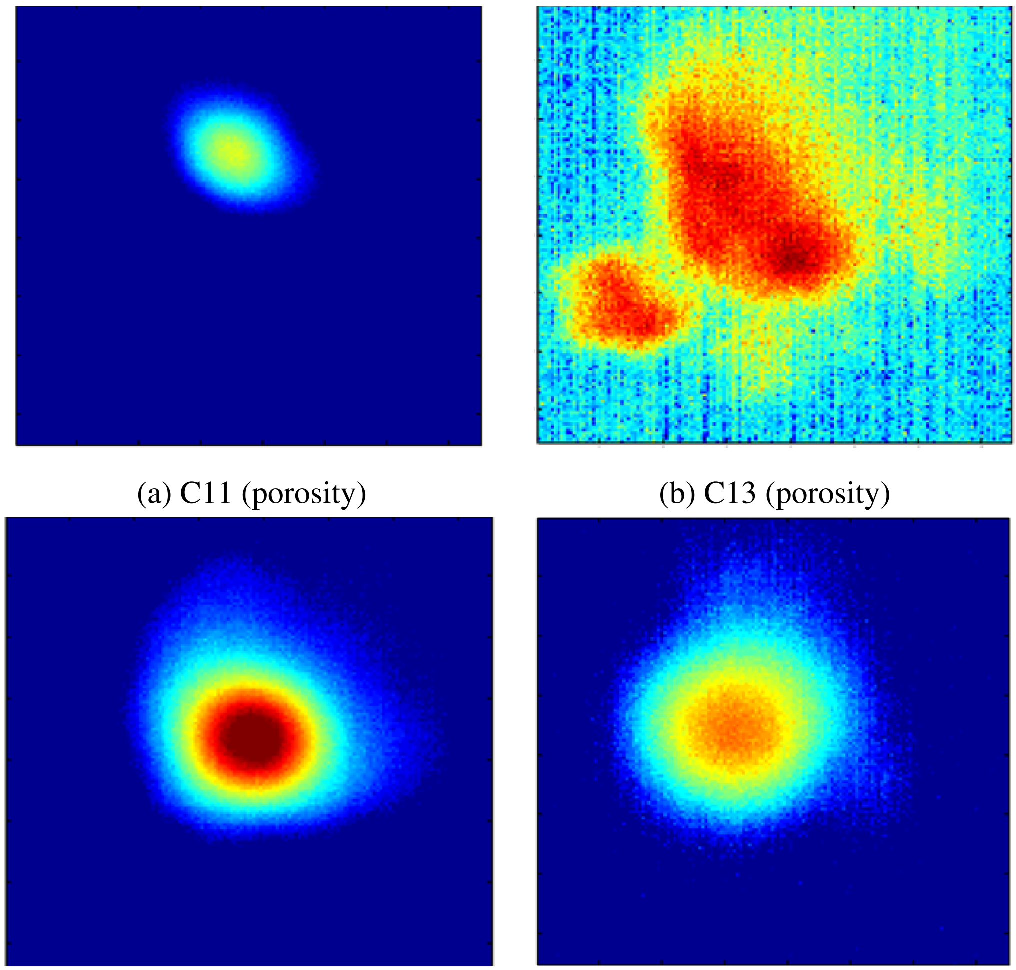


Fig. 3 Correlation matrix (Cor) showing how pores were identified in the present methodology for the 16 clusters (Corr) of the 4×4 SOM map



(a) C11 (porosity)

(b) C13 (porosity)

(c) C44 (porosity)

(d) C21 (normal)

Fig. 4 Thermal profile images of 3 melt pools with low correlation to other melt pools: (a) C11, (b) C13, and (c) C44. These melt pools resulted in porosity within the final part. The last melt pool, C21, is a normal melt pool profile (based on high correlation with other melt pool profiles) and is shown in (d).

5. Summary and Conclusion

A robust in-situ porosity prediction method for AM parts can be developed through characterizing and using the underlying thermophysical dynamics of LBAM captured by its melt pool signals. This technical note provides a brief demonstration of a porosity prediction methodology that uses SOMs to distinguish between normal and abnormal melt pools. Implementation of this methodology is carried out in MATLAB; scripts are provided as appendixes and a description of their execution is provided in the main body of this note.

6. References

1. Huang Y, Leu MC, Mazumder J, Donmez A. Additive manufacturing: current state, future potential, gaps and needs, and recommendations. ASME J Manuf Sci Eng. 2015; 37(1):014001.

2. Thompson SM, Bian L, Shamsaei N, Yadollahi A. An overview of direct laser deposition for additive manufacturing; Part I: transport phenomena, modeling and diagnostics. Additive Manuf. 2015; 8:36–62.

3. Shamsaei N, Yadollahi A, Bian L, Thompson SM. An overview of direct laser deposition for additive manufacturing; Part II: mechanical behavior, process parameter optimization and control. Additive Manuf. 2015; 8:12–35.

4. Pinkerton A, Lin L. Modelling the geometry of a moving laser melt pool and deposition track via energy and mass balances. J Phys D. 2004; 37(14):1885–1895.

5. Qi H, Mazumder J, Ki H. Numerical simulation of heat transfer and fluid flow in coaxial laser cladding process for direct metal deposition. J Appl Phys. 2006; 100(2): 024903.

6. Chandrasekhar N, Vasudevan M, Bhaduri AK, Jayakumar T. Intelligent modeling for estimating weld bead width and depth of penetration from infra-red thermal images of the weld pool. J Intell Manuf. 2015; 26(1):59–71.

7. Tang L, Landers RG. Melt pool temperature control for laser metal deposition processes—Part I: online temperature control. J Manuf Sci Eng. 2010; 132(1):011010.

8. Kim JD, Peng Y. Melt pool shape and dilution of laser cladding with wire feeding. J Mater Proc Tech. 2000; 104(3):284–293.

9. Picasso M, Hoadley AFA. Finite element simulation of laser surface treatments including convection in the melt pool. Int J Numer Methods Heat & Fluid Flow. 1994; 4(1):61–83.

10. Khanzadeh M, Bian L, Shamsaei N, Thompson SM. Porosity detection of laser based additive manufacturing using melt pool morphology clustering. In: Annual International Solid Freeform Fabrication Symposium (SFF); 2016 Aug 8–10; Austin, TX. p. 1487–1494.

11. Khanzadeh M, Chowdhury S, Tschopp MA, Bian L. A methodology for predicting porosity from thermal imaging of melt pools in additive manufacturing thin wall sections. In: ASME 2017 Proceedings of the 12th International Manufacturing Science and Engineering; 2017 June; Los Angeles, CA. doi:10.1115/MSEC2017-2909.

12. Khanzadeh M, Chowdhury S, Tschopp MA, Doude HR, Marufuzzaman M, Bian L. In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. IISE Trans. doi:10.1080/24725854.2017.1417656.

13. Khanzadeh M, Marandi RJ, Tootooni MS, Bian L, Smith BK, Rao P. Profiling and optimizing the geometric accuracy of additively manufactured components via self-organizing map. In: Annual International Solid Freeform Fabrication Symposium (SFF); 2016 Aug 8–10; Austin, TX. p. 1303–1313.

14. Khanzadeh M, Rao P, Jafari-Marandi R, Smith BK, Tschopp MA, Bian L. Quantifying geometric accuracy with unsupervised machine learning: using self-organizing map on fused filament fabrication additive manufacturing parts. ASME J Manuf Sci Eng. 2018; 140:031011.

Appendix A. MATLAB Function: Master File

This appendix appears in its original form, without editorial change.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Mojtaba MK Khanzadeh %%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Mississippi State University %%%%%%%%%%%%%%%%
%%%%%%% 06 15 2017 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This is a master file with functions: MeltPool, Transform,
% Interpolation, and SOM.  MeltPool function extracts the melt pool
% temperature distribution based on melting point temperature.
% Transform function applies spherical transformation to the pre
% processed data points. Interpolation function applies bi-harmonic
% interpolation  to the extracted data points in the spherical domain.
% In SOM function, interpolated features are fed to SOM clustering
% technique that results in clusters of melt pool.

% First, the melt pool files are read from the directory. The melting
% point temperature and total number of melt pool files are set.
% Afterwards, criteria to differentiate among melted melt pool,
% unmelted melt pool, and noisy melt pool are fixed. Spherical
% transformation is then applied to the pre-processed data to obtain a
% functional form. Bi-harmonic interpolation is applied to extracted
% data points in spherical domain to generate homogenized number of
% features. Finally, SOM is applied to the interpolated features to
% render hit map and correlation matrix.

clc; clear; close all;

% Reads file
IndexFileAdress='F:\New Data\IndexData.csv';
IndexData = xlsread(IndexFileAdress);

% Address of the directory
BaseAdressFile = 'F:\New Data\Data\t';

% Melting point temperature of Ti-6Al-4V
MeltingPoint = 1636;

% Total number of melt pool files
Number_of_Files = 1564;

% Indexing all the files
Empty_CsvFile.Index = [];
```

```
% Number of points for each melt pool
empty_CsvFile.NumberOfIPs = [];

% Condition zero is for Unmelted area
Empty_CsvFile.ConditionZero = 0;

% Condition one is for noisy melt pool
Empty_CsvFile.ConditionOne = 0;
DataSet_Files = repmat(empty_CsvFile,[1,Number_of_Files]);
for i=1:Number_of_Files
    DataSet_Files(i).index = i;

    % MeltPool function is used
    [Collector] = MeltPool(i,IndexData,BaseAdressFile,MeltingPoint);
    DataSet_Files(i).NumberOfIPs = max(size(Collector));

    % Melt pool file which has less than 20 data points, is assumed as
    % unmelted melt pool
    if (DataSet_Files(i).NumberOfIPs < 20)
        DataSet_Files(i).ConditionZero = 1;
        warning(['The file ' int2str(i) ' has ConditionZero.'])
    else
        DataSet_Files(i).ConditionZero = 0;

        % Melt pool file which has more than 6000 data points, is
        % assumed as noisy melt pool
        if (DataSet_Files(i).NumberOfIPs > 6000)
            DataSet_Files(i).ConditionOne = 1;
            warning(['The file ' int2str(i) ' has Condition One.']);
        else
            DataSet_Files(i).ConditionOne = 0;
        end

        % Spherical transformation is applied
        DataSet_Files(i).Sperical = Transform(Collector);

        % Interpolation function is applied
        DataSet_Files(i).interpolatedfeature = ...
            Interpolation(DataSet_Files(i).Sperical);
        disp(['Code just finished working on the file ' int2str(i)])
    end
end
```

```matlab
Interpolatedfeatures =[];
for p = 1:Number_of_Files
    if (DataSet_Files(p).ConditionZero ==1)
        datacollector =[];
    else
        datacollector =[DataSet_Files(p).interpolatedfeature];
        if (DataSet_Files(p).ConditionOne ==1)
            datacollector =[];
        else
            datacollector =[DataSet_Files(p).interpolatedfeature];
        end
    end

    % Collected continuous models for all melt pools
    Interpolatedfeatures = [Interpolatedfeatures,datacollector];
end

% SOM is applied.
% map denotes the topology of SOM.
map = 4;
[net] = SOM(map,Interpolatedfeatures);
```

Appendix B. MATLAB Function: Melt Pool Data Acquisition

This appendix appears in its original form, without editorial change.

```matlab
% Melt Pool Temperature Distribution Collector: Function named
% Meltpool is created that extracts the melt pool temperature
% distribution.Here, i denotes the index of the file, IndexData is a
% csv file that contains the characteristics of each melt pool
% image.BaseAdressFile denotes the directory of the data folder,and
% finally MeltingPoint is the melting temperature of used material.

function [Collector] = File(i,IndexData,BaseAdressFile,MeltingPoint)
    % T,TP,X,Y,YP,Z,ZP, and Layer denote different characteristics of
    % each melt pool
    T  = IndexData(i,2);
    TP = IndexData(i,3);
    X  = IndexData(i,4);
    Y  = IndexData(i,5);
    YP = IndexData(i,6);
    Z  = IndexData(i,7);
    ZP = IndexData(i,8);
    Layer = IndexData(i,9);

    % Directory of melt pool files
    AdressFile = [BaseAdressFile int2str(T) 'p' int2str(TP) ...
        '_x' int2str(X) '_y' int2str(Y) 'p' int2str(YP) ...
        '_z' int2str(Z) 'p' int2str(ZP) '_layer' int2str(Layer) ...
        '.csv' ];
    Data = xlsread(AdressFile);
    Data = Data(4:end,:);
    Collector = [];
    [N_row,N_col] = size (Data);

    % Identifying data points based on melting point temperature
    for ii = 1:N_row
        for jj = 1:N_col
            if(Data(ii,jj)>MeltingPoint)
                Neighbors= 0;
                [a,b] = size(Data);
                if(~(ii==1 || jj==1 || ii==a || jj==b))
                    if(Data(ii-1,jj-1)>MeltingPoint)
                        Neighbors = Neighbors+1;
                    end
                    if(Data(ii-1,jj)> MeltingPoint)
                        Neighbors = Neighbors+1;
                    end
                    if(Data(ii-1,jj+1)>MeltingPoint)
```

```
                    Neighbors = Neighbors+1;
                end
                if(Data(ii,jj-1)>MeltingPoint)
                    Neighbors = Neighbors+1;
                end
                if(Data(ii,jj+1)>MeltingPoint)
                    Neighbors = Neighbors+1;
                end
                if(Data(ii+1,jj-1)>MeltingPoint)
                    Neighbors = Neighbors+1;
                end
                if(Data(ii+1,jj)>MeltingPoint)
                    Neighbors = Neighbors+1;
                end
                if(Data(ii+1,jj+1)>MeltingPoint)
                    Neighbors = Neighbors+1;
                end
                if(Neighbors>=8)
                   Collector = [Collector;[ii,jj,Data(ii,jj)]];
                end
            end
        end

        % If the number of datapoints are more than 2000, it is
        % assumed all the data points melting temperature are more
        % than 1636
        if(max(size(Collector)) > 20000)
            break;
        end
    end
end
end
```

Appendix C. MATLAB Function: Melt Pool Data Transformation

This appendix appears in its original form, without editorial change.

```matlab
% Centralizing, scaling, and spherical transformation of melt pool
% boundary: Function named Transform is created that first centralizes
% the melt pool in peak temperatures. Subsequently, the peak
% temperature is applied to scale the temperature distribution between
% 0 and 1. Finally, the spherical transformation is applied to
% preprocessed data points.

function [Spherical] = Transform(Collector)
    for i = 1: max(size(Collector))
        ThermaProfile(i).Counter = i;
        ThermaProfile(i).XX = Collector(i,1);
        ThermaProfile(i).YY = Collector(i,2);
        ThermaProfile(i).ZZ = Collector(i,3);
    end
    XC = [ThermaProfile.XX];
    YC = [ThermaProfile.YY];
    T  = [ThermaProfile.ZZ]';
    PeakTemperature = max(T);
    MinTemperature  = min(T);

    % A range for temperature is defined
    Range = PeakTemperature - MinTemperature;

    % Extracting the index of PeakTemperature
    PeakIndex = ...
        ThermaProfile([ThermaProfile.ZZ] == PeakTemperature).Counter;

    % Centralizing and scaling
    Coordinate = ...
        [[XC-XC(PeakIndex);YC-YC(PeakIndex)]',(T-MinTemperature)/Range];
    [azimuth,elevation,r] = ...
        cart2sph(Coordinate(:,1),Coordinate(:,2),Coordinate(:,3));

    % Spherical Transformation
    Spherical = [azimuth,elevation,r];
end
```

Appendix D. MATLAB Function: Melt Pool Interpolation

This appendix appears in its original form, without editorial change.

```matlab
% Interpolating the spherically transformed features of melt pool:
% EqualMap is the identical support defined for each melt pool

function [Interpolated] = Interpolation(Spherical)
    % Grid_theta and Grid_phi are the sequence of 20 points in the
    % respective range
    MAX_theta = max(Spherical(:,1));
    MAX_phi = max(Spherical(:,2));
    MIN_theta = min(Spherical(:,1));
    MIN_phi = min(Spherical(:,2));
    Range_phi  = MAX_phi - MIN_phi;
    Range_theta  = MAX_theta - MIN_theta;
    Grid_theta = MIN_theta:Range_theta/19: MAX_theta;
    Grid_phi = MIN_phi:Range_phi /19 :MAX_phi;
    for i = 1:20
        Column_one =  Grid_theta(i);
        for j = 1:20
            Element(j).Map = [Column_one; Grid_phi(j)];
        end
        Section(i).Map =  [Element.Map]';
    end

    % An equal map is defined
    EqualMap = [];
    for k = 1:20
        Combine = Section(k).Map;
        EqualMap = [EqualMap;Combine];
    end

    % Bi-harmonic interpolation on extracted features in spherical
    % domain
    [xData, yData, zData] = prepareSurfaceData(Spherical(:,1), ...
        Spherical(:,2), Spherical(:,3));
    ft = 'Biharmonicint';
    [fitresult, ~] = ...
        fit([xData, yData], zData, ft, 'Normalize', 'on');
    mesh_data_x = EqualMap(:,1);
    mesh_data_y = EqualMap(:,2);

    % Bi-harmonic interpolation is applied to identical function
    % support
    Interpolated = fitresult(mesh_data_x,mesh_data_y);
end
```

Appendix E. MATLAB Function: Self-Organizing Map Analysis

This appendix appears in its original form, without editorial change.

```matlab
% Predicting anomaly clusters via Self organizing map: Interpolated
% features are fed to SOM clustering technique that results in
% clusters of melt pool.

function [net]  = SOM(map,Interpolatedfeatures)
    % Topology of SOM clustering
    dimensions   = [map map];
    coverSteps   = 30;
    initNeighbor = 3;

    % This can be changed to other topologies such as 'gridtop' or
    % 'randtop'
    topologyFcn  = 'hextop';

    % This can be changed to other topologies such as 'link
    % Distance','mandist', or 'boxdist'
    distanceFcn  = 'linkdist';

    % SOM clustering starts
    net = selforgmap(dimensions, coverSteps, initNeighbor, ...
        topologyFcn, distanceFcn);
    net = train(net,Interpolatedfeatures);
    view(net);

    % SOM map
    plotsomtop(net);

    % SOM hit map
    plotsomhits(net,Interpolatedfeatures);
    plotsompos(net,Interpolatedfeatures)
    plotsompos(net,Interpolatedfeatures);
    grid on

    % plot SOM neighbor distances
    plotsomnd(net)

    % plot for each SOM neuron the number of input vectors that it
    % classifies
    figure;
    plotsomhits(net,Interpolatedfeatures)

    % Centroids of each clusters are extracted
    CENTER = net.IW;
```

```
    A = CENTER{1,1};

    % Correlation matrix among centroids of clusters are calculated
    MATRIX_CORRELATION = abs(corrcoef(A'));

    figure1 = figure('Color',[1 1 1]);
    xx = 0.5:1:map^2+0.5;
    yy = 0.5:1:map^2+0.5;
    axes1 = axes('Parent',figure1,...
        'YTick',xx,'YGrid','on',...
        'XTick',yy,'XGrid','on',...
        'Layer','top');
    xlim(axes1,[0.5 map^2+0.5]);
    ylim(axes1,[0.5 map^2+0.5]);
    box(axes1,'on');
    hold(axes1,'all');

    % Correlation matrix is visualized
    imagesc(MATRIX_CORRELATION,'Parent',axes1,'CDataMapping','scaled');
    colorbar('peer',axes1);
end
```

List of Symbols, Abbreviations, and Acronyms

2D	2-dimensional
AM	additive manufacturing
LBAM	laser-based additive manufacturing
SOM	self-organizing map

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIR ARL IMAL HRA RECORDS MGMT RDRL DCL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	RDRL-D M TSCHOPP
2 (PDF)	MISSISSIPPI STATE UNIV L BIAN M MARUFUZZAMAN