REFERENCES

[1] M. D. Duncan and M. Bashkansky. Subsurface defect detection in materials using optical coherence tomography. Optics Express. 13 (1998): 540-545.

[2] E Jonathan. Non-contact and non-destructive testing of silicon V-grooves: A non-medical application of optical coherence tomography (OCT). Optics and Lasers in Engineering. 44 (2006): 1117–1131.

[3] R. Onodera, H. Wakaumi and Y. Ishii. Measurement technique for surface profiling in low-coherence interferometry. Optics Communications. 254 (2005): 52–57.

[4] P. Pavlíèek. Height profile measurement by means of white light interferometry. Proceedings of the SPIE. 5259 (2003): 139-144.

[5] A. F. Fercher, W. Drexler, C. K. Hitzenberger and T. Lasser. Optical coherence tomography-principles and applications. Rep. Prog. Phys. 66 (2003): 239-303.

[6] B. E. Bouma and G. J. Tearney. Handbook of Optical Coherence Tomography. 1st ed. USA: Informa HealthCare, 2001.

[7] S. Chang, X. Liu, X. Cai and C. P. Grover. Full-field optical coherence tomography and its application to multiple-layer 2D information retrieving. Optics Communications. 246 (2005): 579–585.

[8] S. Chang, X. Cai and C. Flueraru. An efficient algorithm used for full-field optical coherence tomography. Optics and Lasers in Engineering. 45 (2007): 1170–1176.

[9] C. Quan, C.J. Tay and M. Li. Inspection of a micro-cantilever's opened and concealed profile using integrated vertical scanning interferometry. Optics Communications. 265 (2006): 359–365.

[10] C. J. Tay, C. Quan and M. Li. Investigation of a dual-layer structure using vertical scanning interferometry. Optics and Lasers in Engineering. 45 (2007): 907–913.

[11] M. Li, C. Quan and C.J. Tay. Continuous wavelet transform for micro-component profile measurement using vertical scanning interferometry. Optics & Laser Technology. 40 (2008): 920-929.

[12] F. L. Pedrotti and L. S. Pedrotti. Introduction to Optics. 2nd ed. USA: Prentice Hall, 1993.

[13] E. Hecht, Optics. 4th ed. USA: Addison Wesley, 2001.

[14] H. Wang, M. W. Jenkins, and A. M. Rollins. A combined multiple-SLED broadband light source at 1300 nm for high resolution optical coherence tomography. Optics Communication. 281 (2008): 1896-1900.

[15] B. E. A. Saleh and M. C. Teich. Fundamentals of Photonics. 1st ed. New York: John Wiley & Sons, 1991.

[16] J. W. Goodman. Statistical Optics. 1st ed. New York: John Wiley & Sons, 2000.

[17] A. L. Graps. An introduction to wavelets. IEEE Computational Science and Engineering. 2 (1995): 50-61.

[18] M. Farge. Wavelet transforms and their applications to turbulence. Annu. Rev. Fluid Mech. 24 (1992): 395-457.

[19] L. Chongchun and Q. Zhengding. A method based Morlet wavelet for extracting vibration signal envelope. Proceeding of ICSP 2000. 1 (2000): 337-340.

[20] R. Büssow. An algorithm for the continuous Morlet wavelet transform. Mechanical Systems and Signal Processing. 21 (2007): 2970-2979.

[21] C. Torrence and G. P. Compo. A practice guide to wavelet analysis. Bulletin of the American Meteorological Society. 79 (1998): 61-78.

[22] Superlum. [online]. Available from: http://www.superlumdiodes.com[2009, March 2]

[23] United Kingdom Accreditation Service. The Expression of Uncertainty and Confidence in Measurement. 2nd ed. London: UKAS, 2007.

APPENDICES

APPENDIX A

MATLAB SOURCE CODES

```
%%%%% THIS IS A PROGRAM FOR CALCULATE A HEIGHT PROFILE OF SURFACE
%%%%% 1. INPUT A START ROW (Y START) IN starow PARAMETER
%%%%% 2. INPUT A STOP ROW (Y STOP) IN endrow PARAMETER
%%%%% 3. INPUT A START COLUMN (X START) IN stacol PARAMETER
%%%%% 4. INPUT A STOP COLUMN (X STOP) IN endcol PARAMETER
%%%%% 5. INPUT THE BLOCKSIZE OF SELECTED AREA FOR CALCULATE IN
%%%%%    blocksize
%%%%%    PARAMETER NOTE: BLOCK SIZE IS SQUARE
%%%%% 6. dj AND dz PARAMETER ARE THE RESOLUTION OF SCALE AND THE
%%%%%    RESOLUTION OF EXPERIMENT, RESPECTIVELY.

clear all
clc

display('Select a file in the folder');
[fname,pname] = uigetfile('*.*','Please select a file in the folder');
filelist = dir(pname);
L = length(filelist);
FList = cell(L-2,1);
ResultList = cell(L-2,1);
filename = filelist(3).name;
fullname = [pname,filename];
rgbdata = imread(fullname);
graydata = rgb2gray(rgbdata);
[row,col] = size(graydata);

display(['The image file size is ',num2str(row),'x',num2str(col)]);

starow = input('Please input start point of row : ==> ');
endrow = input('Please input end point of row : ==> ');
stacol = input('Please input start point of column : ==> ');
endcol = input('Please input end point of column : ==> ');
blocksize = input('Please input the block size :  ==> ');
dz = input('Please input vertical resolution : ==> ');
```

```
dj = input('Please input frequency resolution : ==> ');

display('Getting data from files...please wait.')
scan = 0;

M = floor((endrow-(starow-1))/blocksize);

for rowscan = 1:M;
    selectline = ((rowscan-1)*blocksize)+starow;
    count = 0;

    for n = 1:length(filelist);
        filename = filelist(n).name;
        if isdir(filename),
        else
            count = count+1;
            FList{count} = filename;
            fullname = [pname,filename];
            rgbdata = imread(fullname);
            graydata = rgb2gray(rgbdata);
            [row,col] = size(graydata);
            A = graydata(selectline:selectline+blocksize-1,:);
            N = floor((endcol-(stacol-1))/blocksize);
            result = [];
            for j = 1:N;
                B = A(:,(j-1)*blocksize+stacol:(blocksize*j)+(stacol-
1));
                C = 255-mean(mean(B));
                result = [result;C];
            end
            ResultList{count} = result;
        end
    end
    ResultMat = zeros(length(result),count);
    for n = 1:count;
        ResultMat(:,n) = ResultList{n};
    end

    ResultTable = ResultMat';
```

```
number_of_frame = count;

omg_k = [1:floor(number_of_frame/2)];
omg_k = omg_k.*((2.*pi)/(number_of_frame*dz));
omg_k = [0., omg_k, -omg_k(floor((number_of_frame-1)/2):-1:1)];

n = 5;
Wn = [62.5 125]/500;
[b,a] = butter(n,Wn);
[Filter,tF] = impz(b,a,number_of_frame);
fouFil = fft(Filter);

timevec = clock;
yearstr_ = sprintf('%4d',timevec(1));
monthstr_ = sprintf('%02d',timevec(2));
datestr_ = sprintf('%02d',timevec(3));
hourstr_ = sprintf('%02d',timevec(4));
minstr_ = sprintf('%02d',timevec(5));
curr = cd;
folder_name = [yearstr_,monthstr_,datestr_,hourstr_,minstr_,'Line
',sprintf('%3d',selectline)];
mkdir(folder_name);
cd(folder_name);

DelCwt = zeros(length(result),1);
MAXSUM = zeros(length(result),1);
MAXSUM2 = zeros(length(result),1);
MAXSUM3 = zeros(length(result),1);
MaxFil = zeros(length(result),1);
TotalIndData = zeros(length(result),1);
TotalIndFou = zeros(length(result),1);
TotalIndCwt = zeros(length(result),1);
MaxPosData = zeros(length(result),1);
MaxPosFou = zeros(length(result),1);
MaxPosCwt = zeros(length(result),1);


for k = 1:length(result);
```

```
        dispstr = ['analyzing line ',num2str(selectline),'  block#
    ',num2str(k),' of ',num2str(length(result))];
        display(dispstr);
        filename = ['Line- ',sprintf('%3d',selectline),' Block-
',sprintf('%3d',k)];

        x = [1:1:count];
        y = ResultMat(k,:);
        yleft = mean(y(1:5));
        yright = mean(y(length(y)-4:length(y)));
        xleft = 1;
        xright = length(y);
        slope = (yleft-yright)/(xleft-xright);
        intercept = yleft-slope*xleft;
        offset = slope*x+intercept;
        y_c = y-offset;

        variance = std(y_c)^2;
        y_cn = (y_c - mean(y_c))/sqrt(variance);

        x_k = fft(y_cn);
        xFil = fouFil.*x_k;
        ifxFil = ifft(xFil);
        realyFil = real(ifxFil);
        [MaxFil(k),index_3] = max(realyFil);

        %%%%%%%%%%%% MAX INDEX OF DFT %%%%%%%%%
        TotalIndFou(k,1) = index_3;
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        x_k2 = fft(realyFil);

        omg_o = 2*pi/0.83;
        log_2 = floor(log2(number_of_frame*dz/2/dz));
        Jreal = (1/dj)*log_2;
        J = floor(Jreal);
        fou_para = 4*pi/(omg_o+sqrt(2+(omg_o^2)));
        s = 0;
        t = 0;
```

```
Sum1 = zeros(J+1,number_of_frame);
Sum2 = zeros(J+1,number_of_frame);
for scale = 0:1:J;
    s = s+1;
    s_j(s) = 2*dz*(2^(scale*dj));
    for time = 0:1:(number_of_frame-1);
        t = t+1;
        g = [0:1:(number_of_frame-1)];
        expomg = (s_j(s).*omg_k(g+1))-omg_o;
        Lam(s,t) = fou_para*s_j(s);
        Lam2(s,t) = 2*Lam(s,t);
        H = sqrt(2*pi.*s_j(s)./dz).*(1/(pi^(1/4))).*
            exp(-((expomg./sqrt(10)).^2));
        W = x_k2(g+1).*H.*exp(i*omg_k(g+1).*time*dz);
        Sum1(s,t) = sum(W);
        Sum2(s,t) = abs(Sum1(s,t));
    end
    t = 0;
end
[r,c] = size(Sum1);
Sum1 = reshape(Sum1,r*c,1);
Sum2 = reshape(Sum2,r*c,1);
Lam2 = reshape(Lam2,r*c,1);
[MAXSUM(k),index] = max(Sum2);
Sum1Max = Sum1(index);
Lamda = Lam2(index);
R = imag(Sum1Max)/real(Sum1Max);
PHI = atan(R);

%%%%%%%%%%%% MAX INDEX OF CWT %%%%%%%%%%%%%
TotalIndCwt(k,1) = fix((index/r))+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%% DEL OF CWT %%%%%%%%%%%%%%%%%%
DelCwt(k,1) = (Lamda*PHI)/(4*pi);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    %%%%%%%%%%% MAX INDEX OF DATA %%%%%%%%%%%%%%
    [MAXSUM3(k,1),TotalIndData(k,1)] = max(y_c);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    Sum1 = reshape(Sum1,r,c);
    Sum2 = reshape(Sum2,r,c);
    Lam2 = reshape(Lam2,r,c);
end

Block_Number=[1:1:length(result)]';

%%%%%%%%%% MAX POSITION OF DATA %%%%%%%%%%
MaxPosData = TotalIndData.*dz;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%% MAX POSITION OF DFT %%%%%%%%%%
MaxPosFou = TotalFouData.*dz;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% MAX POSITION OF CWT %%%%%%%%%%
MaxPosCwt = TotalIndCwt.*dz;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% HEIGHT POSITION OF DFT %%%%%%%%
hDFT = MaxPosFou;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%% HEIGHT POSITION OF CWT %%%%%%%%
hCwt = MaxPosCwt - DelCwt;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Report_ = [Block_Number,MaxPosData,hDFT,MaxPosCwt,hCwt];
basename = ['Line- ',sprintf('%3d',selectline),'_'];
fname1 = [basename,'raw_report.csv'];
fname2 = [basename,'summary_report.csv'];
```

```
    fid = fopen(fname2,'a+');
    [r,c] = size(Report_);
    str2print = ['Line#,',sprintf('%3d',selectline)];
    fprintf(fid,'%s\n',str2print);
    fprintf(fid,'%s\n','HBlock#,MaxData,MaxFourier,MaxCwt,HeightCwt');
    for n = 1:r;
        fprintf(fid,'%04d,%04d,%04d,%04d,%5.4e\n',Report_(n,:));
    end
    fclose(fid);
    fid = fopen(fname1,'a+');
    [r,c] = size(ResultTable);
    fprintf(fid,'%s\n',str2print);
    str='Block#001';
    fmtstr='%03d';
    for m = 1:c-1;
        str=[str,',Block#',sprintf('%03d',m+1)];
        fmtstr=[fmtstr,',%03d'];
    end
    str=[str];
    fprintf(fid,'%s\n',str);
    for n=1:r;
        fprintf(fid,fmtstr,ResultTable(n,:));
        fprintf(fid,'%s\n','');
    end
    fclose(fid);
    cd(curr);

    scan = scan+1;
    HData{scan} = MaxPosData;
    HFou{scan} = hDFT;
    HCwt{scan} = hCwt;
end
%%% HMatData IS THE SURFACE PROFILE OF EXPERIMENTAL DATA
%%% HMatFou IS THE SURFACE PROFILE OF DFT DATA BY BUTTERWORTH FILTER
%%% HMatCwt IS THE SURFACE PROFILE OF CWT DATA

HMatData = zeros(length(MaxPosData),scan);
HMatFou = zeros(length(hDFT),scan);
HMatCwt = zeros(length(hCwt),scan);
```

```
for r = 1:scan;
    HMatData(:,r) = HData{r};
    HMatFou(:,r) = HFou{r};
    HMatCwt(:,r) = HCwt{r};
end

%%%%%%%%%%%%%%% RMS CALCULATION %%%%%%%%%%%%%%%
RmsData = mean(std(HMatData));
fprintf('RMS_Data = %5.4f\n',RmsData);

RmsFou = mean(std(HMatFou));
fprintf('RMS_Fourier = %5.4f\n',RmsFou);

RmsCwt = mean(std(HMatCwt));
fprintf('RMS_Cwt = %5.4f\n',RmsCwt);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%% SURFACE PROFILE OF SAMPLE %%%%%%%%%%
[rowSur,colSur] = size(HMatData);
xn = [1:1:colSur];
yn = [1:1:rowSur];
xSur = xn*blocksize*6.24;
ySur = yn*blocksize*6.24;

figure('color','white');
surf(xSur,ySur,HMatData);
title('Surface Profile of Experimental Data');
xlabel('x (micron)');
ylabel('y (micron)');
zlabel('z (micron)');
filename=['surface profile of experimental data'];
saveas(gcf,filename,'jpg');
saveas(gcf,filename,'fig');
close(gcf)

figure('color','white');
surf(xSur,ySur,HMatFou);
title('Surface Profile of DFT Data');
xlabel('x (micron)');
```

```
ylabel('y (micron)');
zlabel('z (micron)');
filename=['surface profile of dft data'];
saveas(gcf,filename,'jpg');
saveas(gcf,filename,'fig');
close(gcf)

figure('color','white');
surf(xSur,ySur,HMatCwt);
title('Surface Profile of CWT Data');
xlabel('x (micron)');
ylabel('y (micron)');
zlabel('z (micron)');
filename=['surface profile of cwt data'];
saveas(gcf,filename,'jpg');
saveas(gcf,filename,'fig');
close(gcf)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B

SPECIFICATIONS OF 3-D NON-CONTACT SURFACE PROFILER SP-500 SERIES

Specifications

Model	SP-500	SP-500DS
Measurement technique	Narrow-band white-light interferometry	
Algorithm	SB algorithm; jointly invented by Tokyo Institute of Technology and Toray Engineering Co., Ltd; Patent-pending	
Range of height measurement	0-100 μm (Optional: 0-350 μm)	
Vertical scan speed	Selectable; max. 50 μm/sec	Selectable; max. 100 μm/sec
Objectives	Selectable; 2.5X, 5X, 10X, 20X, 50X (Selectable upon determination of specifications)	
Measurement speed	[Example] In case of 25 μm range & 128*120 pixel array: 0.8 sec (Dependent on the various parameters)	
Repeatability	[Example] σ (Average) = 20 nm for standard step height	
Measurement array	Selectable; 512 x 480, 256 x 240, 128 x 120	
Dimensions of unit	Microscope: W400 x D600 x H500 mm PC rack: W700 x D800 x H1400 mm	
Utilities	90-110 VAC ; 50/60 Hz ; 1 KVA	

VITAE

Jakkapol Visessamit was born on December 29, 1983 in Chiang Rai. He received his Bachelor Degree of Science in Physics from Silpakorn University in 2004.

Poster Presentation:

J. Visessamit and M. Tianprateep. Thickness measurement technique of low reflected transparent media using low coherence light. 3rd SIAM PHYSICS CONGRESS. The Mandarin Golden Valley Hotel, Nakhon Ratchasima. (20-22 March 2008)