ICS>pushtofit_Callback (1 call, 66.754 sec)

Generated 24-Jan-2013 12:04:43 using cpu time.
subfunction in file /home/dsk03/ugrad/oqadri/Downloads/ICS with Poisson N/ICS.m
Copy to new window for comparing multiple runs

Refresh

Show parent functions Show busy lines Show child functions
Show Code Analyzer results Show file coverage Show function listing

Parents (calling functions)

Function Name	Function Type	Calls
gui_mainfcn	function	1

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
3917	`r1(nbihlfp + v1, nbihlfp + u1,...`	400384	5.995 s	9.0%	
3937	`end`	368640	4.746 s	7.1%	
3935	`r1(nbihlfp + v1, nbihlfp + u1,...`	368640	4.715 s	7.1%	
3919	`end`	400384	4.465 s	6.7%	
3920	`end`	400384	3.997 s	6.0%	
All other lines			42.838 s	64.2%	
Totals			66.754 s	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time	% Time	Time Plot
lsqcurvefit	function	7	18.359 s	27.5%	
inputdlg	function	2	4.059 s	6.1%	
surfc	function	3	1.218 s	1.8%	
dlmwrite	function	23	0.843 s	1.3%	
optimset	function	7	0.375 s	0.6%	
newplot	function	7	0.312 s	0.5%	
imread	function	1	0.125 s	0.2%	
str2double	function	2	0.094 s	0.1%	
fft2	function	12	0.094 s	0.1%	
num2str	function	7	0.062 s	0.1%	
fftshift	function	5	0.062 s	0.1%	

axis	function	6	0.062 s	0.1%	
fullfile	function	15	0.062 s	0.1%	
ifft2	function	6	0.062 s	0.1%	
Gausshalf	function	1	0.031 s	0.0%	
colormap	function	2	0.031 s	0.0%	
Crosscorrelation	function	3	0.031 s	0.0%	
guidata	function	9	0.031 s	0.0%	
lineseries	function	7	0.031 s	0.0%	
initprintexporttemplate	function	10	0 s	0%	
Autocorrelation	function	3	0 s	0%	
isdir	function	1	0 s	0%	
Self time (built-ins, overhead, etc.)			40.808 s	61.1%	
Totals			66.754 s	100%	

Code Analyzer results

Line number	Message
2587	The function 'pushtofit_Callback' might be unused.
2587	Input argument 'eventdata' might be unused, although a later one is used. Consider replacing it by ~.
2629	The variable 'gRnew' appears to change size on every loop iteration. Consider preallocating for speed.
2629	The variable 'gRnew' appears to change size on every loop iteration. Consider preallocating for speed.
2661	The variable 'newcorredfit1' appears to change size on every loop iteration. Consider preallocating for speed.
2665	The variable 'newcorred1' appears to change size on every loop iteration. Consider preallocating for speed.
2714	The variable 'gGnew' appears to change size on every loop iteration. Consider preallocating for speed.
2714	The variable 'gGnew' appears to change size on every loop iteration. Consider preallocating for speed.
2748	The variable 'newcorgreenfit1' appears to change size on every loop iteration. Consider preallocating for speed.
2752	The variable 'newcorgreen1' appears to change size on every loop iteration. Consider preallocating for speed.
2799	The variable 'gBnew' appears to change size on every loop iteration. Consider preallocating for speed.
2799	The variable 'gBnew' appears to change size on every loop iteration. Consider preallocating for speed.
2834	The variable 'newcorbluefit1' appears to change size on every loop iteration. Consider preallocating for speed.
2838	The variable 'newcorblue1' appears to change size on every loop iteration. Consider preallocating for speed.

2887	The variable 'gRGnew' appears to change size on every loop iteration. Consider preallocating for speed.
2887	The variable 'gRGnew' appears to change size on every loop iteration. Consider preallocating for speed.
2921	The variable 'newcorrgfit1' appears to change size on every loop iteration. Consider preallocating for speed.
2925	The variable 'newcorrg1' appears to change size on every loop iteration. Consider preallocating for speed.
2972	The variable 'gRBnew' appears to change size on every loop iteration. Consider preallocating for speed.
2972	The variable 'gRBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3006	The variable 'newcorrbfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3010	The variable 'newcorrb1' appears to change size on every loop iteration. Consider preallocating for speed.
3058	The variable 'gGBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3058	The variable 'gGBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3092	The variable 'newcorgbfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3096	The variable 'newcorgb1' appears to change size on every loop iteration. Consider preallocating for speed.
3259	FOR might not be aligned with its matching END (line 3270).
3262	The variable 'testmatrix1' appears to change size on every loop iteration. Consider preallocating for speed.
3262	The variable 'testmatrix1' appears to change size on every loop iteration. Consider preallocating for speed.
3263	The variable 'testmatrix2' appears to change size on every loop iteration. Consider preallocating for speed.
3263	The variable 'testmatrix2' appears to change size on every loop iteration. Consider preallocating for speed.
3264	The variable 'testmatrix3' appears to change size on every loop iteration. Consider preallocating for speed.
3264	The variable 'testmatrix3' appears to change size on every loop iteration. Consider preallocating for speed.
3265	The variable 'testmatrix4' appears to change size on every loop iteration. Consider preallocating for speed.
3265	The variable 'testmatrix4' appears to change size on every loop iteration. Consider preallocating for speed.
3266	The variable 'testmatrix5' appears to change size on every loop iteration. Consider preallocating for speed.
3266	The variable 'testmatrix5' appears to change size on every loop iteration. Consider preallocating for speed.
3267	The variable 'testmatrix6' appears to change size on every loop iteration. Consider preallocating for speed.
3267	The variable 'testmatrix6' appears to change size on every loop iteration. Consider preallocating for speed.

3276	The variable 'newtricorre1' appears to change size on every loop iteration. Consider preallocating for speed.
3277	The variable 'newtricorre2' appears to change size on every loop iteration. Consider preallocating for speed.
3278	The variable 'newtricorre3' appears to change size on every loop iteration. Consider preallocating for speed.
3280	The variable 'newtricorre4' appears to change size on every loop iteration. Consider preallocating for speed.
3281	The variable 'newtricorre5' appears to change size on every loop iteration. Consider preallocating for speed.
3283	The variable 'newtricorre6' appears to change size on every loop iteration. Consider preallocating for speed.
3284	The variable 'newtricorre7' appears to change size on every loop iteration. Consider preallocating for speed.
3285	The variable 'newtricorre8' appears to change size on every loop iteration. Consider preallocating for speed.
3287	The variable 'newtricorre9' appears to change size on every loop iteration. Consider preallocating for speed.
3288	The variable 'newtricorre10' appears to change size on every loop iteration. Consider preallocating for speed.
3290	The variable 'newtricorre11' appears to change size on every loop iteration. Consider preallocating for speed.
3291	The variable 'newtricorre12' appears to change size on every loop iteration. Consider preallocating for speed.
3293	The variable 'newtricorre13' appears to change size on every loop iteration. Consider preallocating for speed.
3294	The variable 'newtricorre14' appears to change size on every loop iteration. Consider preallocating for speed.
3325	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
3403	The variable 'gRnew' appears to change size on every loop iteration. Consider preallocating for speed.
3403	The variable 'gRnew' appears to change size on every loop iteration. Consider preallocating for speed.
3436	The variable 'newcorredfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3440	The variable 'newcorred1' appears to change size on every loop iteration. Consider preallocating for speed.
3479	The variable 'gGnew' appears to change size on every loop iteration. Consider preallocating for speed.
3479	The variable 'gGnew' appears to change size on every loop iteration. Consider preallocating for speed.
3513	The variable 'newcorgreenfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3517	The variable 'newcorgreen1' appears to change size on every loop iteration. Consider preallocating for speed.
3555	The variable 'gBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3555	The variable 'gBnew' appears to change size on every loop iteration. Consider preallocating for speed.

3590	The variable 'newcorbluefit1' appears to change size on every loop iteration. Consider preallocating for speed.
3594	The variable 'newcorblue1' appears to change size on every loop iteration. Consider preallocating for speed.
3632	The variable 'gRGnew' appears to change size on every loop iteration. Consider preallocating for speed.
3632	The variable 'gRGnew' appears to change size on every loop iteration. Consider preallocating for speed.
3666	The variable 'newcorrgfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3670	The variable 'newcorrg1' appears to change size on every loop iteration. Consider preallocating for speed.
3712	The variable 'gRBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3712	The variable 'gRBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3745	The variable 'newcorrbfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3749	The variable 'newcorrb1' appears to change size on every loop iteration. Consider preallocating for speed.
3793	The variable 'gGBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3793	The variable 'gGBnew' appears to change size on every loop iteration. Consider preallocating for speed.
3827	The variable 'newcorgbfit1' appears to change size on every loop iteration. Consider preallocating for speed.
3831	The variable 'newcorgb1' appears to change size on every loop iteration. Consider preallocating for speed.
3998	The variable 'testmatrix1' appears to change size on every loop iteration. Consider preallocating for speed.
3998	The variable 'testmatrix1' appears to change size on every loop iteration. Consider preallocating for speed.
3999	The variable 'testmatrix2' appears to change size on every loop iteration. Consider preallocating for speed.
3999	The variable 'testmatrix2' appears to change size on every loop iteration. Consider preallocating for speed.
4000	The variable 'testmatrix3' appears to change size on every loop iteration. Consider preallocating for speed.
4000	The variable 'testmatrix3' appears to change size on every loop iteration. Consider preallocating for speed.
4001	The variable 'testmatrix4' appears to change size on every loop iteration. Consider preallocating for speed.
4001	The variable 'testmatrix4' appears to change size on every loop iteration. Consider preallocating for speed.
4002	The variable 'testmatrix5' appears to change size on every loop iteration. Consider preallocating for speed.
4002	The variable 'testmatrix5' appears to change size on every loop iteration. Consider preallocating for speed.
4003	The variable 'testmatrix6' appears to change size on every loop iteration. Consider preallocating for speed.

4003	The variable 'testmatrix6' appears to change size on every loop iteration. Consider preallocating for speed.
4011	The variable 'newtricorre1' appears to change size on every loop iteration. Consider preallocating for speed.
4012	The variable 'newtricorre2' appears to change size on every loop iteration. Consider preallocating for speed.
4015	The variable 'newtricorre4' appears to change size on every loop iteration. Consider preallocating for speed.
4016	The variable 'newtricorre5' appears to change size on every loop iteration. Consider preallocating for speed.
4018	The variable 'newtricorre6' appears to change size on every loop iteration. Consider preallocating for speed.
4019	The variable 'newtricorre7' appears to change size on every loop iteration. Consider preallocating for speed.
4022	The variable 'newtricorre9' appears to change size on every loop iteration. Consider preallocating for speed.
4023	The variable 'newtricorre10' appears to change size on every loop iteration. Consider preallocating for speed.
4025	The variable 'newtricorre11' appears to change size on every loop iteration. Consider preallocating for speed.
4026	The variable 'newtricorre12' appears to change size on every loop iteration. Consider preallocating for speed.
4028	The variable 'newtricorre13' appears to change size on every loop iteration. Consider preallocating for speed.
4029	The variable 'newtricorre14' appears to change size on every loop iteration. Consider preallocating for speed.
4052	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4095	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4096	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4097	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4098	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4104	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4105	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4106	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4107	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4108	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4109	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4110	Use of brackets [] is unnecessary. Use parentheses to group, if needed.
4111	Use of brackets [] is unnecessary. Use parentheses to group, if needed.

Coverage results

[Show coverage for parent directory]

Total lines in function	1560
Non-code lines (comments, blank lines)	693

Code lines (lines that can run)	867
Code lines that did run	435
Code lines that did not run	432
Coverage (did run/can run)	50.17 %

Function listing

Color highlight code according to time

```
time    calls  line
             2587 function pushtofit_Callback(hObject, eventdata, handles)
             2588 % hObject    handle to pushtofit (see GCBO)
             2589 % eventdata  reserved - to be defined in a future version of MATLAB
             2590 % handles    structure with handles and user data (see GUIDATA)
             2591
             2592
             2593
           1 2594 switch handles.str{handles.val};
             2595
           1 2596     case 'Choose...' % User selection
             2597
             2598         errordlg('You need to choose a fitting model','Bad Input','modal')
             2599
           1 2600     case 'ICS red' % User selection
             2601
             2602
             2603         if isempty(handles.filename) ~= 1
             2604             if isdir(handles.filename) ~= 1
             2605                 mkdir(handles.filename);
             2606             end
             2607         end
             2608
             2609
             2610
             2611
             2612            imag = imread('Redtemp.bmp');
             2613
             2614            R = double(imag(:, :, 1));
             2615
             2616            %gets dimensions of orig matrix
             2617            SR = size(R);
             2618
             2619            %calculates average of all matrix entries
             2620            avgR = sum(sum(R))/(SR(1)*SR(2));
             2621
             2622
             2623            gR = (ifft2((fft2(R).*conj(fft2(R))))/(avgR^2*SR(1)*SR(2)))-1;
             2624
             2625            range = handles.range;
             2626
             2627            for i = 1:range
             2628                for j = 1:range
             2629                    gRnew(i, j) = gR(i, j);
             2630                end
             2631            end
             2632
             2633            %Removing the value of g(0, 0) by assigning to it the adjacent
             2634            %value g(1 ,1)
             2635            tempvalueR = gRnew(1, 1);
             2636             gRnew(1, 1) = gRnew(1, 2);
             2637
             2638             delta = [0:range-1; 0:range-1];
             2639             delta1 = [1:range-1; 1:range-1];
             2640             AutoGinitialR = [handles.g0, handles.w, handles.ginf];
             2641             lb = [0, 0, -1000];
             2642             ub = [10000000, 100000000, 1000];
             2643
             2644             options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
```

```
2645            AutoGfitR = lsqcurvefit(@Autocorrelation, AutoGinitialR, delta, gRnew, lb, ub, options);
2646
2647            gRfit = Autocorrelation(AutoGfitR, delta);
2648
2649            pathname = handles.filename;
2650            filename1 = 'ACr.txt';
2651            filename2 = 'ACrFit.txt';
2652            File1 = fullfile(pathname,filename1);
2653            File2 = fullfile(pathname,filename2);
2654
2655            gRnew(1, 1) = tempvalueR;
2656
2657            dlmwrite(File1, gRnew, 'delimiter', '\t');
2658            dlmwrite(File2, gRfit, 'delimiter', '\t');
2659
2660             for mm = 1:handles.range
2661                newcorredfit1(mm) = gRfit(mm, 1);
2662            end
2663
2664            for mm = 1:handles.range-1
2665                newcorred1(mm) = gRnew(mm+1, 1);
2666            end
2667
2668
2669
2670            figure('Name','Autocorrelation Red Y','NumberTitle','off')
2671            plot(delta1,newcorred1,'or',...
2672                delta,newcorredfit1,'--r',...
2673                'LineWidth',2)
2674            if newcorred1(1) >= newcorredfit1(1)
2675                axis([0 range 0 newcorred1(1)])
2676            else
2677                axis([0 range 0 newcorredfit1(1)])
2678            end
2679
2680
2681
2682            guidata(hObject,handles);
2683
2684
2685
2686        %     set(handles.timer, 'String', ceil(toc));
2687
1 2688    case 'ICS green' % User selection
2689
2690
2691        if isempty(handles.filename) ~= 1
2692            if isdir(handles.filename) ~= 1
2693                mkdir(handles.filename);
2694            end
2695        end
2696
2697        imag = imread('Greentemp.bmp');
2698
2699           G = double(imag(:, :, 2));
2700
2701           %gets dimensions of orig matrix
2702           SG = size(G);
2703
2704           %calculates average of all matrix entries
2705           avgG = sum(sum(G))/(SG(1)*SG(2));
2706
2707
2708           gG = (ifft2((fft2(G).*conj(fft2(G))))/(avgG^2*SG(1)*SG(2)))-1;
2709
2710           range = handles.range;
2711
2712           for i = 1:range
2713               for j = 1:range
2714                   gGnew(i, j) = gG(i, j);
2715               end
2716           end
```

```
2717
2718            %Removing the value of g(0, 0) by assigning to it the adjacent
2719            %value g(1 ,1)
2720            tempvalueG = gGnew(1, 1);
2721             gGnew(1, 1) = gGnew(1, 2);
2722
2723             delta = [0:range-1; 0:range-1];
2724             delta1 = [1:range-1; 1:range-1];
2725
2726             AutoGinitialG = [handles.g0, handles.w, handles.ginf];
2727             lb = [0, 0, -1000];
2728             ub = [10000000, 100000000, 1000];
2729
2730             options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
2731             AutoGfitG = lsqcurvefit(@Autocorrelation, AutoGinitialG, delta, gGnew, lb, ub, options);
2732
2733             gGfit = Autocorrelation(AutoGfitG, delta);
2734
2735             pathname = handles.filename;
2736             filename1 = 'ACg.txt';
2737             filename2 = 'ACgFit.txt';
2738             File1 = fullfile(pathname,filename1);
2739             File2 = fullfile(pathname,filename2);
2740
2741         gGnew(1, 1) = tempvalueG;
2742
2743             dlmwrite(File1, gGnew, 'delimiter', '\t');
2744             dlmwrite(File2, gGfit, 'delimiter', '\t');
2745
2746
2747             for mm = 1:handles.range
2748                 newcorgreenfit1(mm) = gGfit(mm, 1);
2749             end
2750
2751             for mm = 1:handles.range-1
2752                 newcorgreen1(mm) = gGnew(mm+1, 1);
2753             end
2754
2755
2756
2757             figure('Name','Autocorrelation Green Y','NumberTitle','off')
2758             plot(delta1,newcorgreen1,'og',...
2759                 delta,newcorgreenfit1,'--g',...
2760                 'LineWidth',2)
2761             if newcorgreen1(1) >= newcorgreenfit1(1)
2762                 axis([0 range 0 newcorgreen1(1)])
2763             else
2764                 axis([0 range 0 newcorgreenfit1(1)])
2765             end
2766
2767
2768
2769
2770             guidata(hObject,handles);
2771
1 2772         case 'ICS blue' % User selection
2773
2774             if isempty(handles.filename) ~= 1
2775                 if isdir(handles.filename) ~= 1
2776                     mkdir(handles.filename);
2777                 end
2778             end
2779
2780
2781
2782         imag = imread('Bluetemp.bmp');
2783
2784            B = double(imag(:, :, 3));
2785
2786            %gets dimensions of orig matrix
2787            SB = size(B);
2788
```

```
2789            %calculates average of all matrix entries
2790            avgB = sum(sum(B))/(SB(1)*SB(2));
2791
2792
2793            gB = (ifft2((fft2(B).*conj(fft2(B))))/(avgB^2*SB(1)*SB(2)))-1;
2794
2795            range = handles.range;
2796
2797            for i = 1:range
2798                for j = 1:range
2799                    gBnew(i, j) = gB(i, j);
2800                end
2801            end
2802
2803
2804            %Removing the value of g(0, 0) by assigning to it the adjacent
2805            %value g(1 ,1)
2806            tempvalueB = gBnew(1, 1);
2807             gBnew(1, 1) = gBnew(1, 2);
2808
2809             delta = [0:range-1; 0:range-1];
2810             delta1 = [1:range-1; 1:range-1];
2811
2812             AutoGinitialB = [handles.g0, handles.w, handles.ginf];
2813             lb = [0, 0, -1000];
2814             ub = [10000000, 100000000, 1000];
2815
2816             options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
2817             AutoGfitB = lsqcurvefit(@Autocorrelation, AutoGinitialB, delta, gBnew, lb, ub, options);
2818
2819             gBfit = Autocorrelation(AutoGfitB, delta);
2820
2821             pathname = handles.filename;
2822             filename1 = 'ACb.txt';
2823             filename2 = 'ACbFit.txt';
2824             File1 = fullfile(pathname,filename1);
2825             File2 = fullfile(pathname,filename2);
2826
2827        gBnew(1, 1) = tempvalueB;
2828
2829             dlmwrite(File1, gBnew, 'delimiter', '\t');
2830             dlmwrite(File2, gBfit, 'delimiter', '\t');
2831
2832
2833             for mm = 1:handles.range
2834                 newcorbluefit1(mm) = gBfit(mm, 1);
2835             end
2836
2837             for mm = 1:handles.range-1
2838                 newcorblue1(mm) = gBnew(mm+1, 1);
2839             end
2840
2841
2842
2843             figure('Name','Autocorrelation Blue Y','NumberTitle','off')
2844             plot(delta1,newcorblue1,'ob',...
2845                 delta,newcorbluefit1,'--b',...
2846                 'LineWidth',2)
2847             if newcorblue1(1) >= newcorbluefit1(1)
2848                 axis([0 range 0 newcorblue1(1)])
2849             else
2850                 axis([0 range 0 newcorbluefit1(1)])
2851             end
2852
2853             guidata(hObject,handles);
2854
2855
1 2856        case 'XCS red/green' % User selection
2857
2858             if isempty(handles.filename) ~= 1
2859                 if isdir(handles.filename) ~= 1
2860                     mkdir(handles.filename);
```

```
2861                end
2862             end
2863
2864
2865
2866
2867             imag1 = imread('Redtemp.bmp');
2868             imag2 = imread('Greentemp.bmp');
2869
2870            R = double(imag1(:, :, 1));
2871            G = double(imag2(:, :, 2));
2872
2873            %gets dimensions of orig matrix
2874            SR = size(R);
2875
2876            %calculates average of all matrix entries
2877            avgR = sum(sum(R))/(SR(1)*SR(2));
2878            avgG = sum(sum(G))/(SR(1)*SR(2));
2879
2880
2881            gRG = (ifft2((fft2(R).*conj(fft2(G))))/(avgR*avgG*SR(1)*SR(2)))-1;
2882
2883            range = handles.range;
2884
2885            for i = 1:range
2886                for j = 1:range
2887                    gRGnew(i, j) = gRG(i, j);
2888                end
2889            end
2890
2891            %Removing the value of g(0, 0) by assigning to it the adjacent
2892            %value g(1 ,1)
2893             tempvalueRG = gRGnew(1, 1);
2894             gRGnew(1, 1) = gRGnew(1, 2);
2895
2896             delta = [0:range-1; 0:range-1];
2897             delta1 = [1:range-1; 1:range-1];
2898
2899             CrossGinitialRG = [handles.g0, handles.w, handles.ginf];
2900             lb = [0, 0, -1000];
2901             ub = [10000000, 100000000, 1000];
2902
2903             options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
2904             CrossGfitRG = lsqcurvefit(@Crosscorrelation, CrossGinitialRG, delta, gRGnew, lb, ub, opt
2905
2906             gRGfit = Crosscorrelation(CrossGfitRG, delta);
2907
2908             pathname = handles.filename;
2909             filename1 = 'XCrg.txt';
2910             filename2 = 'XCrgFit.txt';
2911             File1 = fullfile(pathname,filename1);
2912             File2 = fullfile(pathname,filename2);
2913
2914         gRGnew(1, 1) = tempvalueRG;
2915
2916             dlmwrite(File1, gRGnew, 'delimiter', '\t');
2917             dlmwrite(File2, gRGfit, 'delimiter', '\t');
2918
2919
2920             for mm = 1:handles.range
2921                 newcorrgfit1(mm) = gRGfit(mm, 1);
2922             end
2923
2924             for mm = 1:handles.range-1
2925                 newcorrg1(mm) = gRGnew(mm+1, 1);
2926             end
2927
2928
2929
2930             figure('Name','Crosscorrelation Red Green Y','NumberTitle','off')
2931             plot(delta1,newcorrg1,'oy',...
2932                 delta,newcorrgfit1,'--y',...
```

```
2933                'LineWidth',2)
2934            if newcorrg1(1) >= newcorrgfit1(1)
2935                axis([0 range 0 newcorrg1(1)])
2936            else
2937                axis([0 range 0 newcorrgfit1(1)])
2938            end
2939
2940            guidata(hObject,handles);
2941
2942
1 2943        case 'XCS red/blue' % User selection
2944
2945            if isempty(handles.filename) ~= 1
2946                if isdir(handles.filename) ~= 1
2947                    mkdir(handles.filename);
2948                end
2949            end
2950
2951
2952            imag1 = imread('Redtemp.bmp');
2953            imag2 = imread('Bluetemp.bmp');
2954
2955           R = double(imag1(:, :, 1));
2956           B = double(imag2(:, :, 3));
2957
2958           %gets dimensions of orig matrix
2959           SR = size(R);
2960
2961           %calculates average of all matrix entries
2962           avgR = sum(sum(R))/(SR(1)*SR(2));
2963           avgB = sum(sum(B))/(SR(1)*SR(2));
2964
2965
2966           gRB = (ifft2((fft2(R).*conj(fft2(B))))/(avgR*avgB*SR(1)*SR(2)))-1;
2967
2968           range = handles.range;
2969
2970           for i = 1:range
2971               for j = 1:range
2972                   gRBnew(i, j) = gRB(i, j);
2973               end
2974           end
2975
2976           %Removing the value of g(0, 0) by assigning to it the adjacent
2977           %value g(1 ,1)
2978            tempvalueRB = gRBnew(1, 1);
2979            gRBnew(1, 1) = gRBnew(1, 2);
2980
2981            delta = [0:range-1; 0:range-1];
2982            delta1 = [1:range-1; 1:range-1];
2983
2984            CrossGinitialRB = [handles.g0, handles.w, handles.ginf];
2985            lb = [0, 0, -1000];
2986            ub = [10000000, 100000000, 1000];
2987
2988            options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
2989            CrossGfitRB = lsqcurvefit(@Crosscorrelation, CrossGinitialRB, delta, gRBnew, lb, ub, opt
2990
2991            gRBfit = Crosscorrelation(CrossGfitRB, delta);
2992
2993            pathname = handles.filename;
2994            filename1 = 'XCrb.txt';
2995            filename2 = 'XCrbFit.txt';
2996            File1 = fullfile(pathname,filename1);
2997            File2 = fullfile(pathname,filename2);
2998
2999
3000             gRBnew(1, 1) = tempvalueRB;
3001
3002            dlmwrite(File1, gRBnew, 'delimiter', '\t');
3003            dlmwrite(File2, gRBfit, 'delimiter', '\t');
3004
```

```
3005            for mm = 1:handles.range
3006                newcorrbfit1(mm) = gRBfit(mm, 1);
3007            end
3008
3009            for mm = 1:handles.range-1
3010                newcorrb1(mm) = gRBnew(mm+1, 1);
3011            end
3012
3013
3014
3015            figure('Name','Crosscorrelation Red Blue Y','NumberTitle','off')
3016            plot(delta1,newcorrb1,'om',...
3017                delta,newcorrbfit1,'--m',...
3018                'LineWidth',2)
3019            if newcorrb1(1) >= newcorrbfit1(1)
3020                axis([0 range 0 newcorrb1(1)])
3021            else
3022                axis([0 range 0 newcorrbfit1(1)])
3023            end
3024
3025
3026            guidata(hObject,handles);
3027
3028
1 3029        case 'XCS green/blue' % User selection
3030
3031            if isempty(handles.filename) ~= 1
3032                if isdir(handles.filename) ~= 1
3033                    mkdir(handles.filename);
3034                end
3035            end
3036
3037
3038            imag1 = imread('Greentemp.bmp');
3039            imag2 = imread('Bluetemp.bmp');
3040
3041           G = double(imag1(:, :, 2));
3042           B = double(imag2(:, :, 3));
3043
3044           %gets dimensions of orig matrix
3045           SG = size(G);
3046
3047           %calculates average of all matrix entries
3048           avgG = sum(sum(G))/(SG(1)*SG(2));
3049           avgB = sum(sum(B))/(SG(1)*SG(2));
3050
3051
3052           gGB = (ifft2((fft2(G).*conj(fft2(B))))/(avgG*avgB*SG(1)*SG(2)))-1;
3053
3054           range = handles.range;
3055
3056           for i = 1:range
3057               for j = 1:range
3058                   gGBnew(i, j) = gGB(i, j);
3059               end
3060           end
3061
3062           %Removing the value of g(0, 0) by assigning to it the adjacent
3063           %value g(1 ,1)
3064            tempvalueGB = gGBnew(1, 1);
3065            gGBnew(1, 1) = gGBnew(1, 2);
3066
3067            delta = [0:range-1; 0:range-1];
3068            delta1 = [1:range-1; 1:range-1];
3069
3070            CrossGinitialGB = [handles.g0, handles.w, handles.ginf];
3071            lb = [0, 0, -1000];
3072            ub = [10000000, 100000000, 1000];
3073
3074            options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
3075            CrossGfitGB = lsqcurvefit(@Crosscorrelation, CrossGinitialGB, delta, gGBnew, lb, ub, opt
3076
```

```
3077             gGBfit = Crosscorrelation(CrossGfitGB, delta);
3078
3079
3080             pathname = handles.filename;
3081             filename1 = 'XCgb.txt';
3082             filename2 = 'XCgbFit.txt';
3083             File1 = fullfile(pathname,filename1);
3084             File2 = fullfile(pathname,filename2);
3085
3086         gGBnew(1, 1) = tempvalueGB;
3087
3088             dlmwrite(File1, gGBnew, 'delimiter', '\t');
3089             dlmwrite(File2, gGBfit, 'delimiter', '\t');
3090
3091             for mm = 1:handles.range
3092                 newcorgbfit1(mm) = gGBfit(mm, 1);
3093             end
3094
3095             for mm = 1:handles.range-1
3096                 newcorgb1(mm) = gGBnew(mm+1, 1);
3097             end
3098
3099
3100
3101             figure('Name','Crosscorrelation Green Blue Y','NumberTitle','off')
3102             plot(delta1,newcorgb1,'oc',...
3103                 delta,newcorgbfit1,'--c',...
3104                 'LineWidth',2)
3105             if newcorgb1(1) >= newcorgbfit1(1)
3106                 axis([0 range 0 newcorgb1(1)])
3107             else
3108                 axis([0 range 0 newcorgbfit1(1)])
3109             end
3110
3111             guidata(hObject,handles);
3112
0.03       1 3113         case 'Triple ICS' % User selection
3114
3115
3116
3117            if isempty(handles.filename) ~= 1
3118                 if isdir(handles.filename) ~= 1
3119                     mkdir(handles.filename);
3120                 end
3121            end
3122
3123
3124            imag = imread('RGBtemp.bmp');
3125
3126
3127            R = double(imag(:, :, 1));
3128            G = double(imag(:, :, 2));
3129            B = double(imag(:, :, 3));
3130
3131            %gets dimensions of orig matrix
3132            SG = size(R);
3133
3134            %calculates average of all matrix entries
3135            avgR = sum(sum(R))/(SG(1)*SG(2));
3136            avgG = sum(sum(G))/(SG(1)*SG(2));
3137            avgB = sum(sum(B))/(SG(1)*SG(2));
3138
3139
3140             deltaimagred = R - avgR;
3141             deltaimaggreen = G - avgG;
3142             deltaimagblue = B - avgB;
3143
3144             %-----------------------------------------------------------------
3145             % Calculating triple correlation function through bispectrum
3146
3147
3148             %-----------------------------------------------------------------
```

```
3149            % Calculating Fast Fourier Transform of deltasignal
3150
3151            fftdeltaimagred = fftn(deltaimagred);
3152            fftdeltaimaggreen = fftn(deltaimaggreen);
3153            fftdeltaimagblue = fftn(deltaimagblue);
3154
3155            sfftdeltaimagred = fftshift(fftdeltaimagred);
3156            sfftdeltaimaggreen = fftshift(fftdeltaimaggreen);
3157            sfftdeltaimagblue = fftshift(fftdeltaimagblue);
3158
3159            figure('Name','Fourier Transform of RED','NumberTitle','off')
3160            surfc(abs(sfftdeltaimagred))
3161
3162
3163            prompt = {'Enter starting point:'};
3164            dlg_title = 'Input for calculating bispectrum';
3165            num_lines = 1;
3166            def = {'64'};
3167            lowlim = inputdlg(prompt,dlg_title,num_lines,def);
3168            lowlim = str2double(lowlim);
3169
3170            highlim = SG(1) - lowlim;
3171
3172            lim = highlim - lowlim;
3173
3174            nhlf = SG(1)/2;
3175            nhlfp = nhlf+1;
3176            nhlfm = nhlf-1;
3177
3178            nbi = lim;
3179            nbihlfp = nbi/2+1;
3180
3181            r1 = zeros(lim, lim, lim, lim);
3182
3183            tic
3184
3185                for v1 = (-nhlf+lowlim):0
3186                    for u1 = (-nhlf+lowlim)-(v1-1):(nhlfm-lowlim)
3187                        for v2 = (-nhlf+lowlim):(nhlfm-lowlim)
3188                            for u2 = (-nhlf+lowlim):(nhlfm-lowlim)
3189                                if abs(u2+v2) <= nhlfm
3190
3191                                    r1(nbihlfp + v1, nbihlfp + u1, nbihlfp + v2, nbihlfp + u2) = sff
3192
3193                                end
3194                            end
3195                        end
3196                    end
3197                end
3198
3199
3200
3201
3202
3203                for v1 = 1:(nhlfm-lowlim)
3204                    for u1 = (-nhlf+lowlim):(nhlfm-lowlim)-v1
3205                        for v2 = (-nhlf+lowlim):(nhlfm-lowlim)
3206                            for u2 = (-nhlf+lowlim):(nhlfm-lowlim)
3207                                if abs(u2+v2) <= nhlfm
3208
3209                                    r1(nbihlfp + v1, nbihlfp + u1, nbihlfp + v2, nbihlfp + u2) = sff
3210
3211                                end
3212                            end
3213                        end
3214                    end
3215                end
3216                toc
3217
3218
3219                figure('Name','Bispectrum','NumberTitle','off')
3220
```

```
3221                colormap hsv
3222
3223
3224                bispectdisp = real(r1(:, :, nbihlfp, nbihlfp));
3225                surfc(bispectdisp)
3226
3227
3228    % End of calculating bispectrum
3229    %-------------------------------------------------------------------------
3230
3231
3232
3233    %-------------------------------------------------------------------------
3234    % Calculating triple correlation function
3235
3236
3237    bispectorig = fftshift(r1);
3238
3239    tricorr = ifftn(bispectorig).*((lim^4)/(SG(1)^2*SG(2)^2));   %compute the triple correlation (in
3240
3241    tricorrnorm = tricorr/(avgR*avgG*avgB);
3242
3243    figure('Name','Triple Correlation Function 3D','NumberTitle','off')
3244    tricorrdisp = fftshift(real(tricorrnorm(:,:,1, 1)));
3245    surfc(tricorrdisp)
3246    colormap hsv
3247
3248
3249
3250
3251            prompt = {'Enter number of pixels:'};
3252            dlg_title = 'Fitting range';
3253            num_lines = 1;
3254            def = {'20'};
3255            range = inputdlg(prompt,dlg_title,num_lines,def);
3256            range = str2double(range);
3257
3258
3259            for ii = 1:range
3260        for jj = 1:range
3261
3262            testmatrix1(ii, jj) = real(tricorrnorm(ii, jj, 1, 1));
3263            testmatrix2(ii, jj) = real(tricorrnorm(1, ii, jj, 1));
3264            testmatrix3(ii, jj) = real(tricorrnorm(1, 1, ii, jj));
3265            testmatrix4(ii, jj) = real(tricorrnorm(1, ii, 1, jj));
3266            testmatrix5(ii, jj) = real(tricorrnorm(ii, 1, jj, 1));
3267            testmatrix6(ii, jj) = real(tricorrnorm(ii, 1, 1, jj));
3268
3269        end
3270    end
3271
3272
3273
3274    for mm = 1:range
3275
3276         newtricorre1(mm) = testmatrix1(1, mm);
3277         newtricorre2(mm) = testmatrix1(mm, 1);
3278         newtricorre3(mm) = testmatrix1(mm, mm);
3279
3280         newtricorre4(mm) = testmatrix2(1, mm);
3281         newtricorre5(mm) = testmatrix2(mm, 1);
3282
3283         newtricorre6(mm) = testmatrix3(1, mm);
3284         newtricorre7(mm) = testmatrix3(mm, 1);
3285         newtricorre8(mm) = testmatrix3(mm, mm);
3286
3287         newtricorre9(mm) = testmatrix4(1, mm);
3288         newtricorre10(mm) = testmatrix4(mm, 1);
3289
3290         newtricorre11(mm) = testmatrix5(1, mm);
3291         newtricorre12(mm) = testmatrix5(mm, 1);
3292
```

```
3293          newtricorre13(mm) = testmatrix6(1, mm);
3294          newtricorre14(mm) = testmatrix6(mm, 1);
3295
3296      end
3297
3298  %   disp(newtricorre1)
3299  %   disp(newtricorre2)
3300  %   disp(newtricorre3)
3301  %   disp(newtricorre4)
3302  %   disp(newtricorre5)
3303  %   disp(newtricorre6)
3304  %   disp(newtricorre7)
3305  %   disp(newtricorre8)
3306  %   disp(newtricorre9)
3307  %   disp(newtricorre10)
3308  %   disp(newtricorre11)
3309  %   disp(newtricorre12)
3310  %   disp(newtricorre13)
3311  %   disp(newtricorre14)
3312
3313
3314      newt = ((newtricorre1+newtricorre2+newtricorre3+newtricorre4+newtricorre5+newtricorre6+newtricor
3315
3316
3317
3318      % End of calculating triple correlation function
3319      %-------------------------------------------------------------------------
3320
3321
3322  % End of calculating triple correlation function through bispectrum
3323  %-------------------------------------------------------------------------
3324              correctionw = SG(1)/lim;
3325              delta = [0:range-1];
3326              TripleGinitialRGB = [handles.g0, handles.w, handles.ginf];
3327              lb = [0, 0, -1000];
3328              ub = [10000000, 100000000, 1000];
3329
3330              options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
3331              TripleGfitRGB = lsqcurvefit(@Gausshalf, TripleGinitialRGB, delta, newt, lb, ub, options)
3332
3333              TripleRGBfit = Gausshalf(TripleGfitRGB, delta);
3334
3335              TripleGfitRGB(2) = round(TripleGfitRGB(2)*correctionw*10)/10;
3336              set(handles.wfit, 'String', TripleGfitRGB(2));
3337
3338
3339              pathname = handles.filename;
3340              filename1 = 'TripleCrgb.txt';
3341              filename2 = 'TripleCrgbFit.txt';
3342              File1 = fullfile(pathname,filename1);
3343              File2 = fullfile(pathname,filename2);
3344
3345
3346
3347              dlmwrite(File1, newt, 'delimiter', '\t');
3348              dlmwrite(File2, TripleRGBfit, 'delimiter', '\t');
3349
3350
3351              figure('Name','Triple Correlation Function','NumberTitle','off')
3352              plot(delta,newt,'ok',...
3353                  delta,TripleRGBfit,'--k',...
3354                  'LineWidth',2)
3355
3356
3357              guidata(hObject,handles);
3358
3359
3360
0.03        1 3361              case 'All' % User selection
3362
3363
            1 3364                  if isempty(handles.filename) ~= 1
```

```
                1 3365                  if isdir(handles.filename) ~= 1
                  3366                      mkdir(handles.filename);
                  3367                  end
0.03              1 3368              end
                  3369
                  3370
0.12              1 3371               imag = imread('RGBtemp.bmp');
                  3372
                  3373
                1 3374              R = double(imag(:, :, 1));
                1 3375              G = double(imag(:, :, 2));
                1 3376              B = double(imag(:, :, 3));
                  3377
                  3378
                1 3379            AutoGfitR=0;
                1 3380            AutoGfitG=0;
                1 3381            AutoGfitB=0;
                1 3382            CrossGfitRG=0;
                1 3383            CrossGfitRB=0;
                1 3384            CrossGfitGB=0;
                1 3385            TripleGfitRGBforfile=0;
                  3386
                  3387
                1 3388              if sum(sum(R)) ~= 0
                  3389
                  3390                  %gets dimensions of orig matrix
                1 3391                  SR = size(R);
                  3392
                  3393                  %calculates average of all matrix entries
                1 3394                  avgR = sum(sum(R))/(SR(1)*SR(2));
                  3395
                  3396
0.12              1 3397                  gR = (ifft2((fft2(R).*conj(fft2(R))))/(avgR^2*SR(1)*SR(2)))-1;
                  3398
                1 3399                  range = handles.range;
                  3400
                1 3401                  for i = 1:range
               20 3402                      for j = 1:range
              400 3403                          gRnew(i, j) = gR(i, j);
              400 3404                      end
               20 3405                  end
                  3406
                  3407                  %Removing the value of g(0, 0) by assigning to it the adjacent
                  3408                %value g(1 ,1)
                1 3409                tempvalueR = gRnew(1, 1);
                1 3410                 gRnew(1, 1) = gRnew(1, 2);
                  3411
                1 3412                 delta = [0:range-1; 0:range-1];
                1 3413                 delta1 = [1:range-1; 1:range-1];
                  3414
                1 3415                     AutoGinitialR = [handles.g0, handles.w, handles.ginf];
                1 3416                     lb = [0, 0, -1000];
                1 3417                     ub = [10000000, 100000000, 1000];
                  3418
0.16              1 3419                     options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals
3.09              1 3420                     AutoGfitR = lsqcurvefit(@Autocorrelation, AutoGinitialR, delta, gRnew, lb, ub, optio
                  3421
                1 3422                     gRfit = Autocorrelation(AutoGfitR, delta);
                  3423
                1 3424                     pathname = handles.filename;
                1 3425                     filename1 = 'ACr.txt';
                1 3426                     filename2 = 'ACrFit.txt';
0.03            1 3427                     File1 = fullfile(pathname,filename1);
                1 3428                     File2 = fullfile(pathname,filename2);
                  3429
                1 3430                 gRnew(1, 1) = tempvalueR;
                  3431
0.37            1 3432                 dlmwrite(File1, gRnew, 'delimiter', '\t');
                1 3433                 dlmwrite(File2, gRfit, 'delimiter', '\t');
                  3434
                1 3435                 for mm = 1:handles.range
               20 3436                     newcorredfit1(mm) = gRfit(mm, 1);
```

```
        20 3437            end
           3438
         1 3439            for mm = 1:handles.range-1
        19 3440                newcorred1(mm) = gRnew(mm+1, 1);
        19 3441            end
           3442
           3443
           3444
0.37     1 3445            figure('Name','Autocorrelation Red Y','NumberTitle','off')
0.19     1 3446            plot(delta1,newcorred1,'or',...
           3447                delta,newcorredfit1,'--r',...
           3448                'LineWidth',2)
         1 3449            if newcorred1(1) >= newcorredfit1(1)
           3450                axis([0 range 0 newcorred1(1)])
         1 3451            else
0.03     1 3452                axis([0 range 0 newcorredfit1(1)])
         1 3453            end
           3454
           3455
           3456
         1 3457            guidata(hObject,handles);
0.03     1 3458           end
           3459
           3460
           3461
           3462
           3463
         1 3464           if sum(sum(G)) ~= 0
           3465
           3466               %gets dimensions of orig matrix
         1 3467           SG = size(G);
           3468
           3469           %calculates average of all matrix entries
         1 3470           avgG = sum(sum(G))/(SG(1)*SG(2));
           3471
           3472
0.03     1 3473           gG = (ifft2((fft2(G).*conj(fft2(G))))/(avgG^2*SG(1)*SG(2)))-1;
           3474
         1 3475           range = handles.range;
           3476
         1 3477           for i = 1:range
        20 3478               for j = 1:range
       400 3479                   gGnew(i, j) = gG(i, j);
       400 3480               end
        20 3481           end
           3482
           3483           %Removing the value of g(0, 0) by assigning to it the adjacent
           3484           %value g(1 ,1)
         1 3485           tempvalueG = gGnew(1, 1);
         1 3486            gGnew(1, 1) = gGnew(1, 2);
           3487
         1 3488            delta = [0:range-1; 0:range-1];
         1 3489            delta1 = [1:range-1; 1:range-1];
           3490
         1 3491            AutoGinitialG = [handles.g0, handles.w, handles.ginf];
         1 3492            lb = [0, 0, -1000];
         1 3493            ub = [10000000, 100000000, 1000];
           3494
0.03     1 3495            options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
2.25     1 3496            AutoGfitG = lsqcurvefit(@Autocorrelation, AutoGinitialG, delta, gGnew, lb, ub, options);
           3497
         1 3498            gGfit = Autocorrelation(AutoGfitG, delta);
           3499
         1 3500            pathname = handles.filename;
         1 3501            filename1 = 'ACg.txt';
         1 3502            filename2 = 'ACgFit.txt';
         1 3503            File1 = fullfile(pathname,filename1);
         1 3504            File2 = fullfile(pathname,filename2);
           3505
         1 3506            gGnew(1, 1) = tempvalueG;
           3507
0.06     1 3508            dlmwrite(File1, gGnew, 'delimiter', '\t');
```

```
0.03      1 3509            dlmwrite(File2, gGfit, 'delimiter', '\t');
          3510
          3511
          1 3512            for mm = 1:handles.range
         20 3513                newcorgreenfit1(mm) = gGfit(mm, 1);
         20 3514            end
          3515
          1 3516            for mm = 1:handles.range-1
         19 3517                newcorgreen1(mm) = gGnew(mm+1, 1);
         19 3518            end
          3519
          3520
          3521
0.41      1 3522            figure('Name','Autocorrelation Green Y','NumberTitle','off')
0.03      1 3523            plot(delta1,newcorgreen1,'og',...
          3524                delta,newcorgreenfit1,'--g',...
          3525                'LineWidth',2)
          1 3526            if newcorgreen1(1) >= newcorgreenfit1(1)
          3527                axis([0 range 0 newcorgreen1(1)])
          1 3528            else
          1 3529                axis([0 range 0 newcorgreenfit1(1)])
          1 3530            end
          3531
          3532
          1 3533            guidata(hObject,handles);
          3534
0.03      1 3535           end
          3536
          3537
          3538
          3539
          1 3540           if sum(sum(B)) ~= 0
          3541
          3542               %gets dimensions of orig matrix
          1 3543           SB = size(B);
          3544
          3545           %calculates average of all matrix entries
          1 3546           avgB = sum(sum(B))/(SB(1)*SB(2));
          3547
          3548
          1 3549           gB = (ifft2((fft2(B).*conj(fft2(B))))/(avgB^2*SB(1)*SB(2)))-1;
          3550
          1 3551           range = handles.range;
          3552
          1 3553           for i = 1:range
         20 3554               for j = 1:range
        400 3555                   gBnew(i, j) = gB(i, j);
        400 3556               end
         20 3557           end
          3558
          3559           %Removing the value of g(0, 0) by assigning to it the adjacent
          3560           %value g(1 ,1)
          1 3561           tempvalueB = gBnew(1, 1);
          1 3562            gBnew(1, 1) = gBnew(1, 2);
          3563
          1 3564            delta = [0:range-1; 0:range-1];
          1 3565            delta1 = [1:range-1; 1:range-1];
          3566
          1 3567            AutoGinitialB = [handles.g0, handles.w, handles.ginf];
          1 3568            lb = [0, 0, -1000];
          1 3569            ub = [10000000, 100000000, 1000];
          3570
0.06      1 3571            options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals', 1
3.03      1 3572            AutoGfitB = lsqcurvefit(@Autocorrelation, AutoGinitialB, delta, gBnew, lb, ub, options);
          3573
          1 3574            gBfit = Autocorrelation(AutoGfitB, delta);
          3575
          1 3576            pathname = handles.filename;
          1 3577            filename1 = 'ACb.txt';
          1 3578            filename2 = 'ACbFit.txt';
0.03      1 3579            File1 = fullfile(pathname,filename1);
          1 3580            File2 = fullfile(pathname,filename2);
```

```
                3581
                3582
              1 3583              gBnew(1, 1) = tempvalueB;
                3584
0.03          1 3585             dlmwrite(File1, gBnew, 'delimiter', '\t');
0.03          1 3586             dlmwrite(File2, gBfit, 'delimiter', '\t');
                3587
                3588
              1 3589             for mm = 1:handles.range
             20 3590                 newcorbluefit1(mm) = gBfit(mm, 1);
             20 3591             end
                3592
              1 3593             for mm = 1:handles.range-1
             19 3594                 newcorblue1(mm) = gBnew(mm+1, 1);
             19 3595             end
                3596
                3597
                3598
0.28          1 3599             figure('Name','Autocorrelation Blue Y','NumberTitle','off')
0.03          1 3600             plot(delta1,newcorblue1,'ob',...
                3601                 delta,newcorbluefit1,'--b',...
                3602                 'LineWidth',2)
              1 3603             if newcorblue1(1) >= newcorbluefit1(1)
                3604                 axis([0 range 0 newcorblue1(1)])
              1 3605             else
0.03          1 3606                 axis([0 range 0 newcorbluefit1(1)])
              1 3607             end
                3608
              1 3609             guidata(hObject,handles);
                3610
                3611
              1 3612            end
                3613
                3614
                3615
              1 3616            if sum(sum(R)) ~= 0  && sum(sum(G)) ~= 0
                3617
                3618                    %gets dimensions of orig matrix
              1 3619                SR = size(R);
                3620
                3621                %calculates average of all matrix entries
              1 3622                avgR = sum(sum(R))/(SR(1)*SR(2));
              1 3623                avgG = sum(sum(G))/(SR(1)*SR(2));
                3624
                3625
0.03          1 3626                gRG = (ifft2((fft2(R).*conj(fft2(G))))/(avgR*avgG*SR(1)*SR(2)))-1;
                3627
              1 3628                range = handles.range;
                3629
              1 3630                for i = 1:range
             20 3631                    for j = 1:range
            400 3632                        gRGnew(i, j) = gRG(i, j);
            400 3633                    end
             20 3634                end
                3635
                3636             %Removing the value of g(0, 0) by assigning to it the adjacent
                3637            %value g(1 ,1)
              1 3638             tempvalueRG = gRGnew(1, 1);
              1 3639             gRGnew(1, 1) = gRGnew(1, 2);
                3640
              1 3641             delta = [0:range-1; 0:range-1];
              1 3642             delta1 = [1:range-1; 1:range-1];
                3643
              1 3644                 CrossGinitialRG = [handles.g0, handles.w, handles.ginf];
              1 3645                 lb = [0, 0, -1000];
              1 3646                 ub = [10000000, 100000000, 1000];
                3647
0.03          1 3648                 options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals
2.25          1 3649                 CrossGfitRG = lsqcurvefit(@Crosscorrelation, CrossGinitialRG, delta, gRGnew, lb, ub,
                3650
0.06          1 3651                 gRGfit = Crosscorrelation(CrossGfitRG, delta);
                3652
```

```
                1 3653                  pathname = handles.filename;
                1 3654                  filename1 = 'XCrg.txt';
                1 3655                  filename2 = 'XCrgFit.txt';
                1 3656                  File1 = fullfile(pathname,filename1);
                1 3657                  File2 = fullfile(pathname,filename2);
                  3658
                1 3659              gRGnew(1, 1) = tempvalueRG;
                  3660
     0.03       1 3661              dlmwrite(File1, gRGnew, 'delimiter', '\t');
                1 3662              dlmwrite(File2, gRGfit, 'delimiter', '\t');
                  3663
                  3664
                1 3665              for mm = 1:handles.range
               20 3666                  newcorrgfit1(mm) = gRGfit(mm, 1);
               20 3667              end
                  3668
                1 3669              for mm = 1:handles.range-1
               19 3670                  newcorrg1(mm) = gRGnew(mm+1, 1);
               19 3671              end
                  3672
                  3673
                  3674
     0.25       1 3675              figure('Name','Crosscorrelation Red Green Y','NumberTitle','off')
     0.03       1 3676              plot(delta1,newcorrg1,'oy',...
                  3677                  delta,newcorrgfit1,'--y',...
                  3678                  'LineWidth',2)
                1 3679              if newcorrg1(1) >= newcorrgfit1(1)
                  3680                  axis([0 range 0 newcorrg1(1)])
                1 3681              else
                1 3682                  axis([0 range 0 newcorrgfit1(1)])
                1 3683              end
                  3684
                  3685
                  3686
                1 3687                  guidata(hObject,handles);
                  3688
                  3689
                1 3690             end
                  3691
                  3692
                  3693
                1 3694             if sum(sum(R)) ~= 0  && sum(sum(B)) ~= 0
                  3695
                  3696
                  3697
                  3698                 %gets dimensions of orig matrix
                1 3699                 SR = size(R);
                  3700
                  3701                 %calculates average of all matrix entries
                1 3702                 avgR = sum(sum(R))/(SR(1)*SR(2));
                1 3703                 avgB = sum(sum(B))/(SR(1)*SR(2));
                  3704
                  3705
     0.03       1 3706                 gRB = (ifft2((fft2(R).*conj(fft2(B))))/(avgR*avgB*SR(1)*SR(2)))-1;
                  3707
                1 3708                 range = handles.range;
                  3709
                1 3710                 for i = 1:range
               20 3711                     for j = 1:range
              400 3712                         gRBnew(i, j) = gRB(i, j);
              400 3713                     end
               20 3714                 end
                  3715
                  3716              %Removing the value of g(0, 0) by assigning to it the adjacent
                  3717             %value g(1 ,1)
                1 3718              tempvalueRB = gRBnew(1, 1);
                1 3719              gRBnew(1, 1) = gRBnew(1, 2);
                  3720
                1 3721              delta = [0:range-1; 0:range-1];
                1 3722              delta1 = [1:range-1; 1:range-1];
                  3723
                1 3724                  CrossGinitialRB = [handles.g0, handles.w, handles.ginf];
```

```
                1 3725                  lb = [0, 0, -1000];
                1 3726                  ub = [10000000, 100000000, 1000];
                  3727
     0.03       1 3728                  options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals
     2.59       1 3729                  CrossGfitRB = lsqcurvefit(@Crosscorrelation, CrossGinitialRB, delta, gRBnew, lb, ub,
                  3730
                1 3731                  gRBfit = Crosscorrelation(CrossGfitRB, delta);
                  3732
                1 3733                  pathname = handles.filename;
                1 3734                  filename1 = 'XCrb.txt';
                1 3735                  filename2 = 'XCrbFit.txt';
                1 3736                  File1 = fullfile(pathname,filename1);
                1 3737                  File2 = fullfile(pathname,filename2);
                  3738
                1 3739              gRBnew(1, 1) = tempvalueRB;
                  3740
     0.06       1 3741              dlmwrite(File1, gRBnew, 'delimiter', '\t');
     0.03       1 3742              dlmwrite(File2, gRBfit, 'delimiter', '\t');
                  3743
                1 3744              for mm = 1:handles.range
               20 3745                  newcorrbfit1(mm) = gRBfit(mm, 1);
               20 3746              end
                  3747
                1 3748              for mm = 1:handles.range-1
               19 3749                  newcorrb1(mm) = gRBnew(mm+1, 1);
               19 3750              end
                  3751
                  3752
                  3753
     0.19       1 3754              figure('Name','Crosscorrelation Red Blue Y','NumberTitle','off')
     0.03       1 3755              plot(delta1,newcorrb1,'om',...
                  3756                  delta,newcorrbfit1,'--m',...
                  3757                  'LineWidth',2)
                1 3758              if newcorrb1(1) >= newcorrbfit1(1)
                  3759                  axis([0 range 0 newcorrb1(1)])
                1 3760              else
                1 3761                  axis([0 range 0 newcorrbfit1(1)])
                1 3762              end
                  3763
                  3764
                  3765
                1 3766                  guidata(hObject,handles);
                  3767
                  3768
                  3769
                  3770
     0.03       1 3771          end
                  3772
                  3773
                  3774
                1 3775          if sum(sum(G)) ~= 0  && sum(sum(B)) ~= 0
                  3776
                  3777
                  3778
                  3779              %gets dimensions of orig matrix
                1 3780              SG = size(G);
                  3781
                  3782              %calculates average of all matrix entries
                1 3783              avgG = sum(sum(G))/(SG(1)*SG(2));
                1 3784              avgB = sum(sum(B))/(SG(1)*SG(2));
                  3785
                  3786
                1 3787              gGB = (ifft2((fft2(G).*conj(fft2(B))))/(avgG*avgB*SG(1)*SG(2)))-1;
                  3788
                1 3789              range = handles.range;
                  3790
                1 3791              for i = 1:range
               20 3792                  for j = 1:range
              400 3793                      gGBnew(i, j) = gGB(i, j);
              400 3794                  end
               20 3795              end
                  3796
```

```
                3797            %Removing the value of g(0, 0) by assigning to it the adjacent
                3798           %value g(1 ,1)
              1 3799            tempvalueGB = gGBnew(1, 1);
              1 3800            gGBnew(1, 1) = gGBnew(1, 2);
                3801
              1 3802            delta = [0:range-1; 0:range-1];
              1 3803            delta1 = [1:range-1; 1:range-1];
                3804
              1 3805                CrossGinitialGB = [handles.g0, handles.w, handles.ginf];
              1 3806                lb = [0, 0, -1000];
              1 3807                ub = [10000000, 100000000, 1000];
                3808
 0.03         1 3809                options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunEvals
 2.90         1 3810                CrossGfitGB = lsqcurvefit(@Crosscorrelation, CrossGinitialGB, delta, gGBnew, lb, ub,
                3811
              1 3812                gGBfit = Crosscorrelation(CrossGfitGB, delta);
                3813
                3814
              1 3815                pathname = handles.filename;
              1 3816                filename1 = 'XCgb.txt';
              1 3817                filename2 = 'XCgbFit.txt';
              1 3818                File1 = fullfile(pathname,filename1);
              1 3819                File2 = fullfile(pathname,filename2);
                3820
              1 3821            gGBnew(1, 1) = tempvalueGB;
                3822
 0.03         1 3823            dlmwrite(File1, gGBnew, 'delimiter', '\t');
 0.03         1 3824            dlmwrite(File2, gGBfit, 'delimiter', '\t');
                3825
              1 3826            for mm = 1:handles.range
             20 3827                newcorgbfit1(mm) = gGBfit(mm, 1);
             20 3828            end
                3829
              1 3830            for mm = 1:handles.range-1
             19 3831                newcorgb1(mm) = gGBnew(mm+1, 1);
             19 3832            end
                3833
                3834
                3835
 0.22         1 3836            figure('Name','Crosscorrelation Green Blue Y','NumberTitle','off')
              1 3837            plot(delta1,newcorgb1,'oc',...
                3838                delta,newcorgbfit1,'--c',...
                3839                'LineWidth',2)
              1 3840            if newcorgb1(1) >= newcorgbfit1(1)
                3841                axis([0 range 0 newcorgb1(1)])
              1 3842            else
              1 3843                axis([0 range 0 newcorgbfit1(1)])
              1 3844            end
                3845
 0.03         1 3846                guidata(hObject,handles);
                3847
                3848
                3849
                3850
              1 3851           end
                3852
                3853
                3854
              1 3855           if sum(sum(R)) ~= 0  && sum(sum(G)) ~= 0 && sum(sum(B)) ~= 0
                3856
                3857
                3858                    %gets dimensions of orig matrix
              1 3859                SG = size(R);
                3860
                3861                %calculates average of all matrix entries
              1 3862                avgR = sum(sum(R))/(SG(1)*SG(2));
              1 3863                avgG = sum(sum(G))/(SG(1)*SG(2));
              1 3864                avgB = sum(sum(B))/(SG(1)*SG(2));
                3865
                3866
              1 3867                 deltaimagred = R - avgR;
              1 3868                 deltaimaggreen = G - avgG;
```

```
                1 3869                  deltaimagblue = B - avgB;
                  3870
                  3871                  %-----------------------------------------------------------------
                  3872                  % Calculating triple correlation function through bispectrum
                  3873
                  3874
                  3875                  %-----------------------------------------------------------------
                  3876                  % Calculating Fast Fourier Transform of deltasignal
                  3877
                1 3878                  fftdeltaimagred = fftn(deltaimagred);
                1 3879                  fftdeltaimaggreen = fftn(deltaimaggreen);
                1 3880                  fftdeltaimagblue = fftn(deltaimagblue);
                  3881
                1 3882                  sfftdeltaimagred = fftshift(fftdeltaimagred);
                1 3883                  sfftdeltaimaggreen = fftshift(fftdeltaimaggreen);
0.03            1 3884                  sfftdeltaimagblue = fftshift(fftdeltaimagblue);
                  3885
0.34            1 3886                  figure('Name','Fourier Transform of RED','NumberTitle','off')
0.84            1 3887                  surfc(abs(sfftdeltaimagred))
                  3888
                1 3889                  prompt = {'Enter starting point:'};
                1 3890                  dlg_title = 'Input for calculating bispectrum';
                1 3891                  num_lines = 1;
                1 3892                  def = {'96'};
2.40            1 3893                  lowlim = inputdlg(prompt,dlg_title,num_lines,def);
0.03            1 3894                  lowlim = str2double(lowlim);
                  3895
                1 3896                  highlim = SG(1) - lowlim;
                  3897
                1 3898                  lim = highlim - lowlim;
                  3899
                1 3900                  nhlf = SG(1)/2;
                1 3901                  nhlfp = nhlf+1;
                1 3902                  nhlfm = nhlf-1;
                  3903
                1 3904                  nbi = lim;
                1 3905                  nbihlfp = nbi/2+1;
                  3906
                1 3907                  r1 = zeros(lim, lim, lim, lim);
                  3908
                1 3909                  tic
                  3910
                1 3911                      for v1 = (-nhlf+lowlim):0
               17 3912                          for u1 = (-nhlf+lowlim)-(v1-1):(nhlfm-lowlim)
              391 3913                              for v2 = (-nhlf+lowlim):(nhlfm-lowlim)
0.19        12512 3914                                  for u2 = (-nhlf+lowlim):(nhlfm-lowlim)
3.90       400384 3915                                      if abs(u2+v2) <= nhlfm
                  3916
5.99       400384 3917                                          r1(nbihlfp + v1, nbihlfp + u1, nbihlfp + v2, nbihlfp + u
                  3918
4.46       400384 3919                                      end
4.00       400384 3920                                  end
0.22        12512 3921                              end
              391 3922                          end
               17 3923                      end
                  3924
                  3925
                  3926
                  3927
                  3928
                1 3929                      for v1 = 1:(nhlfm-lowlim)
               15 3930                          for u1 = (-nhlf+lowlim):(nhlfm-lowlim)-v1
              360 3931                              for v2 = (-nhlf+lowlim):(nhlfm-lowlim)
0.16        11520 3932                                  for u2 = (-nhlf+lowlim):(nhlfm-lowlim)
3.97       368640 3933                                      if abs(u2+v2) <= nhlfm
                  3934
4.71       368640 3935                                          r1(nbihlfp + v1, nbihlfp + u1, nbihlfp + v2, nbihlfp + u
                  3936
4.75       368640 3937                                      end
3.59       368640 3938                                  end
0.25        11520 3939                              end
              360 3940                          end
```

```
        15 3941                         end
         1 3942                         toc
           3943
           3944
      0.34  1 3945                       figure('Name','Bispectrum','NumberTitle','off')
           3946
      0.06  1 3947                       colormap hsv
           3948
           3949
         1 3950                         bispectdisp = real(r1(:, :, nbihlfp, nbihlfp));
      0.34  1 3951                       surfc(bispectdisp)
           3952
           3953
           3954             % End of calculating bispectrum
           3955             %---------------------------------------------------------------------
           3956
           3957
           3958
           3959             %---------------------------------------------------------------------
           3960             % Calculating triple correlation function
           3961
           3962
      0.03  1 3963             bispectorig = fftshift(r1);
           3964
      0.50  1 3965             tricorr = ifftn(bispectorig).*((lim^4)/(SG(1)^2*SG(2)^2));   %compute the triple correla
           3966
      0.09  1 3967             tricorrnorm = tricorr/(avgR*avgG*avgB);
           3968
      0.62  1 3969             figure('Name','Triple Correlation Function 3D','NumberTitle','off')
         1 3970             tricorrdisp = fftshift(real(tricorrnorm(:,:,1, 1)));
      0.06  1 3971             surfc(tricorrdisp)
         1 3972             colormap hsv
           3973
           3974
         1 3975                     prompt = {'Input G0:','Input w:', 'Input Ginf:', 'Input fitting range:', 'Input
         1 3976                     dlg_title = 'Parameters for fitting';
         1 3977                     num_lines = 1;
         1 3978                     def = {'50','2', '0', '15', '30'};
      1.75  1 3979                     parforfit = str2double(inputdlg(prompt,dlg_title,num_lines,def));
         1 3980                     handles.g0 = parforfit(1);
         1 3981                     handles.w = parforfit(2);
         1 3982                     handles.ginf = parforfit(3);
         1 3983                     range = parforfit(4);
         1 3984                     handles.niter = parforfit(5);
           3985
           3986
           3987
           3988
           3989
           3990
           3991
           3992
           3993
           3994
         1 3995             for ii = 1:range
        15 3996                 for jj = 1:range
           3997
       225 3998                     testmatrix1(ii, jj) = real(tricorrnorm(ii, jj, 1, 1));
       225 3999                     testmatrix2(ii, jj) = real(tricorrnorm(1, ii, jj, 1));
       225 4000                     testmatrix3(ii, jj) = real(tricorrnorm(1, 1, ii, jj));
       225 4001                     testmatrix4(ii, jj) = real(tricorrnorm(1, ii, 1, jj));
       225 4002                     testmatrix5(ii, jj) = real(tricorrnorm(ii, 1, jj, 1));
       225 4003                     testmatrix6(ii, jj) = real(tricorrnorm(ii, 1, 1, jj));
           4004
       225 4005                 end
        15 4006             end
           4007
           4008
         1 4009             for mm = 1:range
           4010
        15 4011                  newtricorre1(mm) = testmatrix1(1, mm);
        15 4012                  newtricorre2(mm) = testmatrix1(mm, 1);
```

```
                4013                % newtricorre3(mm) = testmatrix1(mm, mm);
                4014
               15 4015                newtricorre4(mm) = testmatrix2(1, mm);
               15 4016                newtricorre5(mm) = testmatrix2(mm, 1);
                4017
               15 4018                newtricorre6(mm) = testmatrix3(1, mm);
               15 4019                newtricorre7(mm) = testmatrix3(mm, 1);
                4020               %  newtricorre8(mm) = testmatrix3(mm, mm);
                4021
               15 4022                newtricorre9(mm) = testmatrix4(1, mm);
               15 4023                newtricorre10(mm) = testmatrix4(mm, 1);
                4024
               15 4025                newtricorre11(mm) = testmatrix5(1, mm);
               15 4026                newtricorre12(mm) = testmatrix5(mm, 1);
                4027
               15 4028                newtricorre13(mm) = testmatrix6(1, mm);
               15 4029                newtricorre14(mm) = testmatrix6(mm, 1);
                4030
               15 4031            end
                4032
                4033
                4034
                1 4035            newt = ((newtricorre1+newtricorre2+newtricorre4+newtricorre5+newtricorre6+newtricorre7+n
                4036
                4037
                4038
                4039            % End of calculating triple correlation function
                4040            %-------------------------------------------------------------------------
                4041
                4042
                4043        % End of calculating triple correlation function through bispectrum
                4044        %-------------------------------------------------------------------------
                4045
                4046
                4047
                4048
                4049
                4050
                1 4051                    correctionw = SG(1)/lim;
                1 4052                    delta = [0:range-1];
                1 4053                    TripleGinitialRGB = [handles.g0, handles.w, handles.ginf];
                1 4054                    lb = [0, 0, -1000];
                1 4055                    ub = [10000000, 100000000, 1000];
                4056
0.09            1 4057                    options = optimset('outputfcn',@outfun1, 'TolFun',1e-50, 'TolX', 1e-50, 'MaxFunE
2.34            1 4058                    TripleGfitRGB = lsqcurvefit(@Gausshalf, TripleGinitialRGB, delta, newt, lb, ub,
                4059
0.03            1 4060                    TripleRGBfit = Gausshalf(TripleGfitRGB, delta);
                4061
                1 4062                    TripleGfitRGB(2) = round(TripleGfitRGB(2)*correctionw*10)/10;
                1 4063                    set(handles.wfit, 'String', TripleGfitRGB(2));
                4064
                1 4065                    TripleGfitRGBforfile = [TripleGfitRGB(1), TripleGfitRGB(2), TripleGfitRGB(3)];
                4066
                4067
                1 4068                    pathname = handles.filename;
                1 4069                    filename1 = 'TripleCrgb.txt';
                1 4070                    filename2 = 'TripleCrgbFit.txt';
                1 4071                    File1 = fullfile(pathname,filename1);
                1 4072                    File2 = fullfile(pathname,filename2);
                4073
                4074
                4075
0.03            1 4076                    dlmwrite(File1, newt, 'delimiter', '\t');
0.03            1 4077                    dlmwrite(File2, TripleRGBfit, 'delimiter', '\t');
                4078
                4079
0.22            1 4080                    figure('Name','Triple Correlation Function','NumberTitle','off')
0.03            1 4081                    plot(delta,newt,'ok',...
                4082                        delta,TripleRGBfit,'--k',...
                4083                        'LineWidth',2)
                4084
```

```
          1                     guidata(hObject,handles);

          1             end

            %------------------------------------------------------------------------
            %Saving fitting parameters into a file



          1        Title1 = ['parameters:        '];
          1        Title2 = ['G(0)       '];
          1        Title3 = ['w       '];
          1        Title4 = ['Ginf'];

          1         Title = cat(2, Title1, Title2, Title3, Title4);

          1         fname = cat(2, handles.filename, '.txt');

          1         empty1= [' '];
          1         head1 = ['RED                 '];
          1         head2 = ['GREEN               '];
          1         head3 = ['BLUE                '];
          1         head4 = ['RED-GREEN           '];
          1         head5 = ['RED-BLUE            '];
          1         head6 = ['GREEN-BLUE          '];
          1         head7 = ['RED-GREEN-BLUE      '];

0.03      1         red = cat(2, head1, num2str(AutoGfitR, '%-10.5f'));
0.03      1         green = cat(2, head2, num2str(AutoGfitG, '%-10.5f'));
          1         blue = cat(2, head3, num2str(AutoGfitB, '%-10.5f'));
          1         redgreen = cat(2, head4, num2str(CrossGfitRG, '%-10.5f'));
          1         redblue = cat(2, head5, num2str(CrossGfitRB, '%-10.5f'));
          1         greenblue = cat(2, head6, num2str(CrossGfitGB, '%-10.5f'));
          1         redgreenblue = cat(2, head7, num2str(TripleGfitRGBforfile, '%-10.5f'));

          1         File = fullfile(handles.filename,fname);
0.06      1         dlmwrite(File, Title, '');
0.03      1         dlmwrite(File, empty1, '-append', 'delimiter', '');

0.03      1         dlmwrite(File, red, '-append', 'delimiter', '');
          1         dlmwrite(File, green, '-append', 'delimiter', '');
          1         dlmwrite(File, blue, '-append', 'delimiter', '');
0.03      1         dlmwrite(File, redgreen, '-append', 'delimiter', '');
          1         dlmwrite(File, redblue, '-append', 'delimiter', '');
          1         dlmwrite(File, greenblue, '-append', 'delimiter', '');
0.03      1         dlmwrite(File, redgreenblue, '-append', 'delimiter', '');




          1         guidata(hObject, handles)


            %End saving fitting parameters into a file
            %------------------------------------------------------------------------



          1 end

          1        guidata(hObject,handles);
```

Other subfunctions in this file are not included in this listing.