July 20, 2023

Computing a Partial Elastic Shape Registration of 3D Surfaces using Dynamic Programming

This README file describes the zip file by the name ESD_surf_3d.zip which contains a software package that incorporates the methods presented in the NIST Technical Note 2274

> *Partial Elastic Shape Registration of 3D Surfaces using Dynamic Programming*
> by Javier Bernal, James Lawrence, DOI: https://doi.org/10.6028/NIST.TN.2274

A copy of this manuscript is included in the zip file under the name NIST.TN.2274.pdf

The software package can be used to compute a partial elastic shape registration of two simple surfaces in 3-d space, and the elastic shape distance between them associated with the partial registration.

Javier Bernal
National Institute of Standards and Technology (NIST)
Gaithersburg, MD
Javier.Bernal@nist.gov

The software package is written in Matlab (MATLAB R2023a) with the exception of the Dynamic Programming routine which is written in Fortran but is executed as a Matlab mex file.

Since ESD_driv_surf_3d.m is the driver routine for the package, the package can be executed by simply typing in the Matlab window:

```
ESD_driv_surf_3d
```

As will be described below, values that are assigned to a couple of variables in this routine may have to be modified in the routine before the execution of the package. In addition, we note that in this routine four input files are read: x_domain.txt, y_domain.txt, surface1.txt and surface2.txt. The contents of these files are described below.

DP_MEX_WNDSTRP_ALLDIM.F is the Dynamic Programing Fortran routine which is executed as a Matlab mex file. If necessary, this routine can be processed to obtain a new mex file for the routine by typing in the Matlab window:

```
mex -compatibleArrayDims DP_MEX_WNDSTRP_ALLDIM
```

At the start of the execution of the software, we assume S1, S2 are the two simple surfaces in 3-dimensional space under consideration, with continuously differentiable functions c_1, c_2 from D = [0,T1] x [0,T2] into 3-dimensional space, T1, T2 > 0, as their parametrizations, respectively, so that S1 = c_1(D), S2 = c_2(D).

We also assume that as input to the software, for positive integers M, N, not necessarily equal, and partitions of [0,T1], [0,T2], respectively, r_i, i = 1, …, M, r_1 = 0 < r_2 < … < r_M = T1, and t_j, j = 1, …, N, t_1 = 0 < t_2 < … < t_N = T2, not necessarily uniform, discretizations of c_1, c_2 are given, each discretization in the form of a list of M x N points in the corresponding surface, namely c_1(r_i,t_j) and c_2(r_i,t_j), i = 1, …, M, j = 1, …, N, respectively, and for k = 1, 2, as specified in the Introduction section of the manuscript mentioned above, in the order c_k(r_1,t_1), c_k(r_2,t_1), …, c_k(r_M,t_1), …, c_k(r_1,t_N), c_k(r_2,t_N), …, c_k(r_M,t_N).

We note, the partitions r_i, i = 1, …, M, and t_j, j = 1, …, N, are obtained in routine ESD_driv_surf_3d.m from input files x_domain.txt and y_domain.txt, respectively, and the discretizations of the surfaces c_1(r_i,t_j) and c_2(r_i,t_j), i = 1, …, M, j = 1, …, N, are obtained in the same routine from input files surface1.txt and surface2.txt, respectively, in the order as described above. Accordingly, we think of the first partition (the r partition) as being in the x direction, and the second partition (the t partition) as being in the y direction.

Based on this input, for the purpose of computing, using Dynamic Programming, a partial elastic shape registration of S1 and S2, together with the elastic shape distance between them associated with the partial registration, the program always proceeds first to scale the partitions r_i, i = 1, …, M, t_j, j =1, …, N, so that they become partitions of [0,1], and to compute an approximation of the area of each surface.

During the execution of the software package, the former is accomplished by Matlab routine ESD_driv_surf_3d.m, while the latter by Matlab routine ESD_comp_surf_3d.m through the computation for each k, k = 1, 2, of the sum of the areas of triangles with vertices c_k(r_i,t_j), c_k(r_{i+1},t_{j+1}, c_k(r_i,t_{j+1}), and c_k(r_i,t_j), c_k(r_{i+1},t_j), c_k(r_{i+1},t_{j+1}), for i = 1, …, M - 1, j = 1, …, N - 1.

The program then proceeds to scale the discretizations of the parametrizations of the surfaces so that each surface has approximate area equal to 1 (given a surface and its approximate area, each point in the discretization of the parametrization of the surface is divided by the square root of half the approximate area of the surface).

Once routine ESD_comp_surf_3d.m is done, the actual computations of the partial registration and associated elastic shape distance are carried out by Matlab routine ESD_core_surf_3d.m in which the methods for this purpose presented in the manuscript mentioned above, mainly Procedure DP-surface-min in Section 7 of the manuscript, have been implemented.

The software package does require that values be assigned to variables iten and both_drct in the driver routine. By opening the source code of the driver routine ESD_driv_surf_3d.m, one can easily locate two lines that appear as follows:
iten = 7;
both_drct = 0;
variables iten and both_drct are described below and if the values at which they are currently set (7 and 0, respectively) are satisfactory, then nothing has to be done. Otherwise, they can be easily modified.

iten, currently set to 7, is the maximum number of times the repeat loop in Procedure DP-surface-min is allowed to be executed.

both_drct, currently set to 0, is for determining whether the first surface is to be considered in the reversed direction of rows as well (0 if no, 1 if yes). Here the reversed direction of rows means that the N rows on the surface are reversed: the 1st row becomes the Nth row, and viceversa, etc. Accordingly, it is assumed that the surfaces are oriented in such a way that only the two directions addressed in this program are possible. Partition in y direction should be uniform if yes. Actually eight directions are possible, two per corner of the unit square, but here for the sake of simplicity we assume only two are possible as pointed out above.

At the end of the execution of the driver routine ESD_driv_surf_3d.m and therefore of the software package, nine variables which we call output variables, will have values that can be used to identify the computed partial elastic shape registration of the two surfaces, the elastic shape distance between the two surfaces associated with the partial registration, and some other information. Accordingly at the end of the execution of the driver routine six output files are created that contain the values of six of these variables. We note that the values of the three other variables are simply printed on the screen as each one of them is just a single number. The nine variables are described below. The name of the output file, if any, into which each variable is saved is identified as well.

drct_srfc - if equal to 1, even if the first surface was considered in the reverse direction of the rows as well, it is the direction given on input that gives the optimal solutions; otherwise the first surface was indeed considered in the reverse direction of the rows and it is in this direction that the optimal solutions were obtained. Printed on screen.

f1, f2 – with MN equal to the product of M and N, 3xMN arrays containing the discretizations of the first and second surfaces, respectively, after being processed (scaled, etc.) in Matlab routine ESD_comp_surf_3d.m, discretized by partitions r, t, in the order as described above, neither one rotated or parametrized. Saved in ouput files f1_data.txt, f2_data.txt, respectively.

f1opt, f2opt – with MN as above, 3xMN arrays containing the discretizations of the first and second surfaces, respectively, that are interpreted to achieve the partial elastic registration of the two surfaces, both discretized by partitions r, t, in the same order as arrays f1, f2 above, and are the result of rotating the first surface and reparametrizing the second one. Saved in output files f1opt_data.txt, f2opt_data.txt, respectively.

iter - number of times the repeat loop in Procedure DP-surface-min was executed in Matlab routine ESD_core_surf_3d.m. Printed on screen.

Ropt – 3x3 array containing the rotation matrix used on first surface to obtain partial elastic shape registration. Saved in output file ropt_data.txt

gammaopt – with MN as above, 1xMN array containing x-coordinates of discretization of homeomorphism used on second surface to obtain partial elastic shape registration, discretized by partitions r, t, in the same order as arrays f1, f2, f1opt, f2opt above, the y-coordinates understood to be the corresponding values of partition t. Saved in output file gammaopt_data.txt

minE - the square of the computed elastic shape distance between the two surfaces corresponding to the computed partial elastic shape registration. Printed on screen.

The zip file also contains a few other files that are not part of the software package described above. Two of these files contain Fortran programs (helicoid.f and f12opt_edges.f) and two others Matlab programs (plot_surfaces.m and plot_op_surfaces). plot_surfaces.m plots the boundaries of the two input surfaces, and plot_op_surfaces.m plots the boundaries of the two output surfaces after a partial elastic shape registration of the two surfaces has been computed. helicoid.f generates files of partitions in the x and y directions, and of two discretized helicoid surfaces for which a partial elastic shape registration is to be computed with the software package. It also generates input files for plot_surfaces.m. Once the software package has been executed, f12opt_edges.f generates input files for plot_op_surfaces.m. All of these additional programs are short and simple and can be modified without much effort. Actually all of the Fortran and Matlab routines and programs in the zip file are essentially short and simple and can be easily read and modified. The exception is the Dynamic Programming Fortran routine which should only be modified when setting parameter nmax, the maximum number of points in the discretization of a curve. Note that because there is a limit to the maximum size an integer variable can have and there are some calculations in the routine that can exceed this limit, it has been established that nmax should not exceed 46340.

As mentioned above, the software package is executed by typing ESD_driv_surf_3d in the Matlab window. Also as mentioned above, ESD_driv_surf_3d.m, the driver routine, reads four input files. For testing purposes, four such input files are included in the zip file that contain in particular discretizations of two helicoids, one helicoid slightly perturbed in the direction of the r partition (the x direction). These files were created with program helicoid.f, each file containing 10201 points. The shapes of the two helicoids are essentially the same, so that the elastic shape distance between them should be essentially zero. See Section 8 of the manuscript for more details about the definition of a helicoid. Also the way in which the two variables described above in the driver routine are set to values can be easily understood by reading the routine itself. Again see Section 8 for more details about using these files and the results obtained.

Again it is highly recommended, with the exception of the Dynamic Programming Fortran routine which happens to be extremely intricate, to open any file and look at its contents for gaining understanding and clarification of what the file is about. Also it is recommended to study Section 8 of the manuscript.