
parsePhantomParameterFile

Version number: 1.1

Arguments: file -- .txt file containing specifications of the atelectasis phantom to be created

Return: parsedData -- Structure containing phantom creation parameters

Description: This function reads a parameters file and parses the information into a structure which is returned.

Revision / Date / Author / Description

1.0 / 04 02 17 / Chris Guy / Initial build

1.1 / 04 04 17 / Chris Guy / Add vessel-related parameters

```
function [ parsedData ] = parsePhantomParameterFile( file )

    fid = fopen(file);

    while ~feof(fid)
        currentLine = fgetl(fid);
        splitLine = regexp(currentLine, '=', 'split');

        if(strcmp(splitLine(1), 'firstName'))

            parsedData.firstName = splitLine{2};

        elseif(strcmp(splitLine(1), 'secondName'))

            parsedData.secondName = splitLine{2};

        elseif(strcmp(splitLine(1), 'nDims'))

            parsedData.dims = str2num(cell2mat(splitLine(2)));

        elseif(strcmp(splitLine(1), 'imgSpacing'))

            parsedData.spacing = str2num(cell2mat(splitLine(2)));

        elseif(strcmp(splitLine(1), 'imgOrigin'))

            parsedData.origin = str2num(cell2mat(splitLine(2)));

        elseif(strcmp(splitLine(1), 'imgSize'))

            parsedData.size = str2num(cell2mat(splitLine(2)));

        elseif(strcmp(splitLine(1), 'bgDensity'))

            parsedData.bgDensity = str2num(cell2mat(splitLine(2)));

    end
```

```

elseif(strcmp(splitLine(1),'featureDensity'))

    parsedData.featureDensity =
str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'firstObj'))

    firstObjFile = splitLine{2};

elseif(strcmp(splitLine(1),'firstDensity'))

    parsedData.firstDensity = str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'secondObj'))

    secondObjFile = splitLine{2};

elseif(strcmp(splitLine(1),'massRatio'))

    parsedData.massRatio = str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'addNoise'))

    parsedData.noise = str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'addFirstFeatures'))

    parsedData.addFirstFeatures =
str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'addSecondFeatures'))

    parsedData.addSecondFeatures =
str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'firstFeatures'))

    firstFeatureFile = splitLine{2};

elseif(strcmp(splitLine(1),'secondFeatures'))

    secondFeatureFile = splitLine{2};

elseif(strcmp(splitLine(1),'addFirstZGradient'))

    parsedData.addFirstZGradient =
str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'addSecondZGradient'))

    parsedData.addSecondZGradient = ...
        str2num(cell2mat(splitLine(2)));

elseif(strcmp(splitLine(1),'firstZGradient'))

```

```

        parsedData.firstZGradient =
str2num(cell2mat(splitLine(2)));

        elseif(strcmp(splitLine(1),'secondZGradient'))

            parsedData.secondZGradient = ...
                str2num(cell2mat(splitLine(2)));

        else

            errorMsg=['Error - Parameter not recognized: '
splitLine{1}];
            disp(errorMsg);

        end

    end % while ~feof

    fclose(fid);

    % Object files are read.
    parsedData.firstObj = parseObjectFile(firstObjFile);
    parsedData.secondObj = parseObjectFile(secondObjFile);

    % If features are to be added, feature files are read.
    if parsedData.addFirstFeatures
        parsedData.firstFeatures = parseObjectFile(firstFeatureFile);
    end

    if parsedData.addSecondFeatures
        parsedData.secondFeatures =
parseObjectFile(secondFeatureFile);
    end

    % Required fields are initialized if not provided.
    if ~isfield(parsedData, 'addFirstZGradient')
        parsedData.addFirstZGradient = 0;
    end
    if ~isfield(parsedData, 'addSecondZGradient')
        parsedData.addSecondZGradient = 0;
    end

end % parsePhantomParameterFile

```

parseObjectFile

This subfunction reads object specifications from the given file.

```

function [ objData ] = parseObjectFile( file )

    fid = fopen(file);

```

```

iObject = 1;

while ~feof(fid)
    currentLine = fgetl(fid);
    splitLine = regexp(currentLine, '=', 'split');

    % Cylinder aligned along the Z axis
    if(strcmp(splitLine(1),'cz'))

        params = strsplit(splitLine{2},',');
        params = ['1', params];
        objData(iObject,:)=params;
        iObject = iObject + 1;

    % Sphere
    elseif(strcmp(splitLine(1),'s'))

        params = strsplit(splitLine{2},',');
        params = ['2', params];
        objData(iObject,:)=params;
        iObject = iObject + 1;

    % Above plane sloping along an XZ line
    elseif(strcmp(splitLine(1),'apy'))

        params = strsplit(splitLine{2},',');
        params = ['3', params];
        objData(iObject,:)=params;
        iObject = iObject + 1;

    % Spheroid about the Z axis
    elseif(strcmp(splitLine(1),'sz'))

        params = strsplit(splitLine{2},',');
        params = ['4', params];
        objData(iObject,:)=params;
        iObject = iObject + 1;

    % Below plane sloping along an XZ line
    elseif(strcmp(splitLine(1),'bpy'))

        params = strsplit(splitLine{2},',');
        params = ['5', params];
        objData(iObject,:)=params;
        iObject = iObject + 1;

    % Cylinder aligned along the Z axis (e.g. "vessel")
    elseif(strcmp(splitLine(1),'cy'))

        params = strsplit(splitLine{2},',');
        params = ['6', params];
        objData(iObject,:)=params;
        iObject = iObject + 1;

```

```
        else

            errorMsg=['Error - Object type not recognized: '
splitLine{1}];
            disp(errorMsg);

        end

    end % while ~feof

    fclose(fid);

end % parseObjectFile
```

Published with MATLAB® R2016a