MATLAB App Designer

创建了一个图形用户界面 (GUI)，用于选择和处理图像。用户可以通过这个 GUI 打开图像、在图像上定义点，并根据这些点执行某些操作。

代码的功能和结构：

属性部分 properties 块定义了应用的组件：

- figure1：主窗口
- text18：显示标题 "Tour into the Picture" 的标签
- start、resetPoint、openPicture、addPoint：各种操作的按钮
- axes2：用于显示图像的坐标轴组件

私有方法

pointCallback(app, src, ~)：

当用户点击一个已经添加的点时，删除该点并更新全局变量 pointNum

按钮回调函数

1. **openPictureButtonPushed(app, event)**：

   打开并显示用户选择的图像文件

2. **addPointButtonPushed(app, event)**：

   当用户点击"4 Eckpunkte und 1Fluchtpunkt definieren"按钮时，允许用户在图像上添加指定数量的点

3. **resetPointButtonPushed(app, event)**：

   重置点并重新显示原始图像，清除任何覆盖内容

4. **startButtonPushed(app, event)**：

   启动主要的图像处理任务

   根据控制点计算距离并执行投影矫正

   如果有目标图像，则根据控制点调整大小和位置

调用另一个函数，根据计算的点进行 3D 变换

辅助函数（在其他地方实现）

1. **CalculatePointCoordinates(I, control_pts)**：根据控制点计算非控制点
2. **ImageCropping(I, poly)**：裁剪由多边形 poly 定义的图像部分
3. **ProjectiveRectification(control_pts, non_control_pts, ...)**：执行图像部分的投影矫正
4. **ForegroundCalculation(I, control_pts, non_control_pts, point2, geo)**：计算前景对象的位置
5. **Transform3D(X, Y, img_target, img_front_rectified, ...)**：根据点将图像转换到 3D 空间

应用创建和删除

1. **构造函数 (main_App)**：

   如果应用尚未运行，则创建应用及其组件

   将应用注册到 App Designer

2. **析构函数 (delete(app))**：

   当应用关闭时删除图形。

全局变量

该应用大量依赖全局变量在函数之间共享数据：

- I：图像矩阵
- pointNum：添加的点的数量
- img_background、img_target：背景和目标图像
- non_control_pts、control_pts：存储点坐标的数组

CalculatePointCoordinates

用于计算 8 个非控制点，这些点可以用于后续图像切割操作。函数的输入是图像和控制点的坐标，输出是计算得到的 8 个非控制点。计算方法基于几何插值，使非控制点位于消失点与相应控制点的连线上

输入

- image：输入的图像

- control_pts：控制点坐标，是一个 2x5 的矩阵，每列表示一个控制点（分别是左上、右上、右下、左下以及中心消失点）

输出

- pts：计算得到的 8 个非控制点坐标，是一个 2x8 的矩阵

定义控制点

从输入的 control_pts 矩阵中提取四个控制点（左上、右上、右下、左下）和消失点的坐标

计算非控制点的坐标

非控制点的坐标通过控制点和消失点的坐标计算得到，计算方法如下：

1. **顶部的非控制点**

   ncpt_1 和 ncpt_2：位于图像顶部（y 坐标为 0）的点

   ncpt_1 和 ncpt_2 的 x 坐标通过插值计算得到，使它们位于消失点与顶部控制点的连线上

2. **底部的非控制点**

   ncpt_3 和 ncpt_4：位于图像底部（y 坐标为图像高度 max_row）的点

   ncpt_3 和 ncpt_4 的 x 坐标通过插值计算得到，使它们位于消失点与底部控制点的连线上

3. **左侧和右侧的非控制点**

   ncpt_5 和 ncpt_8：位于图像左侧（x 坐标为 0）的点

   ncpt_6 和 ncpt_7：位于图像右侧（x 坐标为图像宽度 max_colomn）的点

   这些点的 y 坐标通过插值计算得到，使它们位于消失点与左/右侧控制点的连线上

ImageCropping

实现了一个图像裁剪，它通过指定的多边形区域裁剪输入图像，生成并显示一个只包含多边形内区域的裁剪图像

函数定义

输入参数为 image 和 polygon。image 是要裁剪的图像，polygon 是一个包含多边形顶点的矩阵

提取多边形的顶点

从输入的多边形矩阵 polygon 中提取出两行，分别存储在 row1 和 row2 中。这两行分别代表多边形的 x 坐标和 y 坐标

创建二值掩码

使用 roipoly 函数创建一个二值掩码 BW。这个函数根据多边形的顶点在输入图像上生成一个二值掩码，掩码的区域为多边形内的区域

裁剪图像

将图像和掩码转换为 double 类型，并进行逐元素相乘。这一步将图像中位于多边形内的区域保留，其他区域设置为零

转换为 uint8 类型

将裁剪后的图像转换回 uint8 类型，以便与输入图像的类型一致

ProjectiveRectification

用于进行二维投影变换，从而实现图像的透视矫正，以便用于三维重建，该函数的主要功能是将输入的多个图像进行透视矫正，输出矫正后的图像以及计算的几何信息

输入参数:

control_pts: 控制点的坐标，用于定义前景和其他边界的四边形

non_control_pts: 非控制点的坐标，用于定义其他边界的四边形

img_front, img_left, img_right, img_top, img_bottom: 分别为前景和四个边界的图像

f: 相机的焦距或相关参数

- **输出参数:**

  img_front_rectified, img_left_rectified,
  img_right_rectified, img_top_rectified,
  img_bottom_rectified: 经过透视矫正后的图像

  geo: 包含三维图像的深度、高度和宽度信息的数组

函数实现解析

1. **计算深度、高度和宽度:**

   m1 和 m2 是根据控制点和非控制点计算的参数，用于确定三维图像的深度

   vh 是非控制点和控制点计算的另一个参数，用于计算图像的高度

   通过这些参数，计算出了三维图像的深度、高度和宽度，并存储在 geo 变量中

2. **定义四边形:**

   使用控制点和非控制点的坐标，定义了用于透视矫正的四边形 quad_front, quad_left, quad_right, quad_top, quad_bottom

3. **创建投影变换对象:**

   使用 fitgeotrans 函数根据定义的四边形和固定的点（fixed_points_horizontal, fixed_points_vertical, fixed_points_front），创建投影变换对象 trf_front, trf_left, trf_right, trf_top, trf_bottom

4. **应用投影变换并进行旋转:**

   使用 imwarp 函数将输入图像应用于投影变换

   使用 imrotate 函数对变换后的图像进行适当的旋转和镜像翻转，以使其正确方向

5. **输出结果:**

将处理后的图像赋值给输出变量 img_front_rectified, img_left_rectified, img_right_rectified, img_top_rectified, img_bottom_rectified

Transform3D

用于在三维空间中显示多个图像（前、左、右、上、下），并允许用户通过交互来调整视角和缩放范围。下面逐步解释代码的功能和实现细节：

函数 Transform3D

它接受以下参数：

- x 和 y：目标图像放置的位置坐标。
- img_target、img_front、img_left、img_right、img_top、img_bottom：分别是目标图像以及前、左、右、上、下各个方向的图像

设置图像尺寸

获取各个图像的高度和宽度。这些尺寸用于后续的坐标计算和图像显示

创建图形窗口和轴

创建一个白色背景的图形窗口，并在窗口上创建一个轴 Axes，用于显示三维图像

图像放置到三维场景中

将各个图像放置到三维场景中：warp 函数用于将图像放置到指定的三维坐标网格 (X, Y, Z) 中。每个 warp 函数调用后的 f1、f2、f3、f4、f5 是返回的图形对象句柄，用于后续的交互操作

设置轴属性和交互功能

设置轴的属性，包括使坐标轴比例相等、显示网格，并设置坐标轴标签

鼠标滚轮交互功能

定义了一个响应鼠标滚轮事件的函数 ScrollWheel，用于调整显示范围 range

更新视角和范围

function update(~)定义了一个更新函数 update，用于更新轴的范围和视角