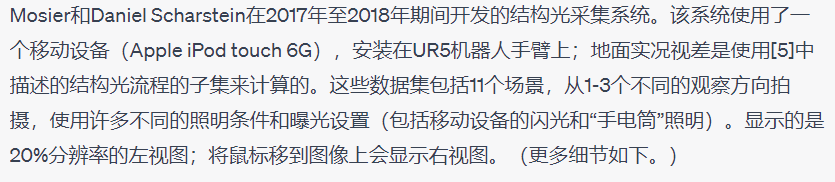
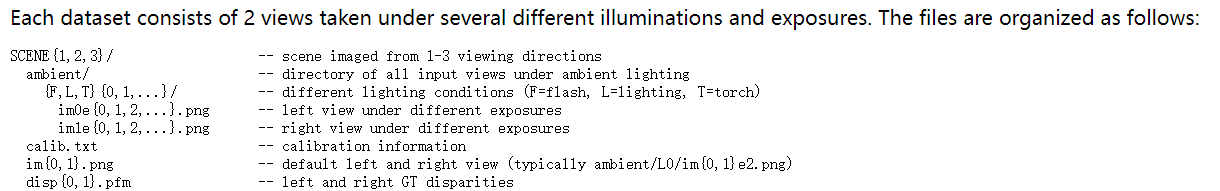
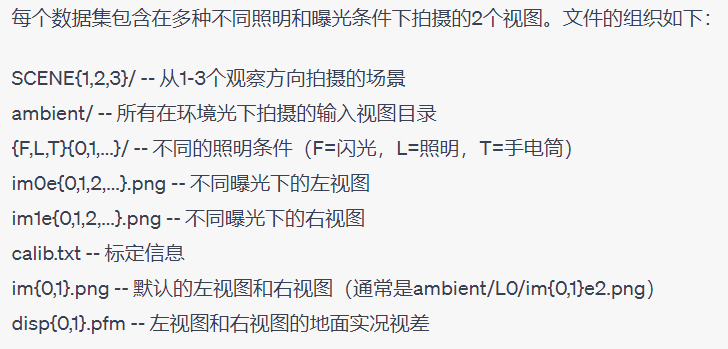
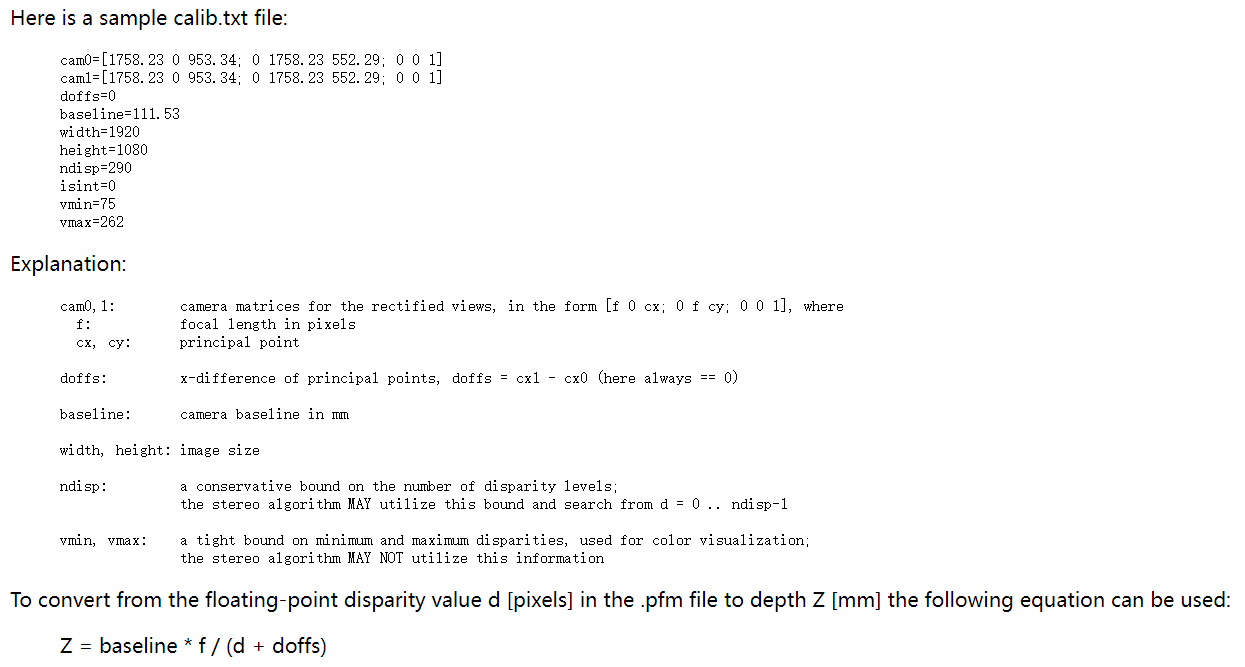
先下载图片。网站——datesets——选择（2021【5】）

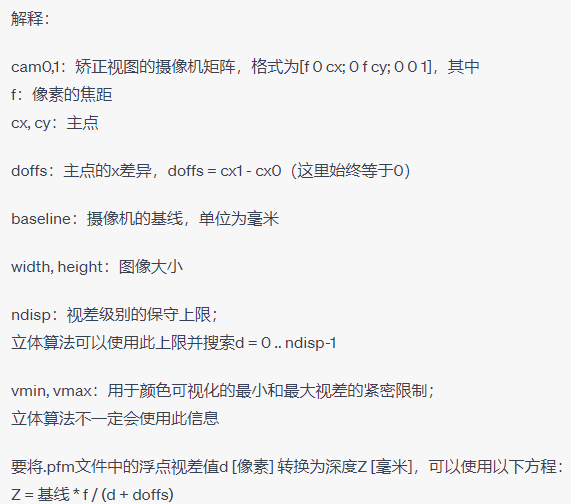




注意每个数据集的信息！







此页用于python知识点的补充：

1、PEP8的警告：PEP 8: E501 line too long (139 > 120 characters)

拆分长行代码：将一行代码拆分成多行，以确保每行不超过 120 个字符。可以使用反斜杠线 \ 来继续一行代码，或者将长表达式分成多个变量或子表达式。

你的代码之所以会收到 PEP 8 中的 E128 警告，是因为你在换行继续代码时，第二行的缩进不正确。PEP 8 要求在换行继续一行代码时，继续的那一行要比上一行的缩进多一个空格。在你的代码中，第二行的缩进与第一行相同，这是不符合 PEP 8 规范的。

要修复这个警告，你可以将第二行的缩进增加一个空格，使其与上一行的缩进不同。这将会让代码符合 PEP 8 规范，同时也更容易阅读。

PEP 8: E502 the backslash is redundant between brackets

模板代码中有一些需要根据具体需求修改的地方，以使其适应您的实验要求和数据。以下是需要根据您的具体需求进行修改的主要部分：

1. \*\*图像读取\*\*：

```python

left\_image\_path = 'left\_image.png'

right\_image\_path = 'right\_image.png'

```

修改这两行以指定左右图像的文件路径。确保您的图像文件在相应的路径上。

2. \*\*模板匹配参数\*\*：

```python

window\_size = 5

disparity\_range = 64

```

根据您的实验需求，您可以更改`window\_size`（匹配窗口大小）和`disparity\_range`（视差范围）。这些参数将影响计算的视差图的质量和性能。

3. \*\*相似度度量方法\*\*：

在示例代码中，使用了均方差（SSD）来计算相似度。如果您希望尝试其他相似度度量方法，例如标准化互相关（NCC）等，请相应地修改模板匹配部分的代码。

4. \*\*结果保存\*\*：

```python

cv2.imwrite('disparity\_map.png', disparity\_map)

```

这一行代码将视差图像保存为`disparity\_map.png`。您可以更改文件名和路径，以适应您的需求。

5. \*\*异常处理和参数验证\*\*：

在实际应用中，您可能需要添加异常处理和参数验证的代码，以确保程序的稳定性。例如，检查图像是否成功读取，确保输入参数的合理性等。

6. \*\*注释和文档\*\*：

最重要的是，您需要添加清晰的注释和文档，以便他人能够理解您的代码。解释函数的作用，参数的含义，以及实现的方法。

根据您的具体实验要求和数据，您可能还需要进行更多的修改和优化，以满足特定的需求和性能要求。这个示例代码是一个起点，您可以根据需要进行自定义和扩展。

---2023.10.19---

此外，关于技术报告的内容，以下是一些主要部分的简要解释：

双目立体视觉原理：

解释了双目立体视觉的基本原理，包括使用两个相机获取不同视角的图像以估计深度信息。

讨论了视差的概念，即左右图像中像素位置的差异与物体深度之间的关系。

实现解决方案： √

详细解释了上述Python程序的工作原理，包括读取图像、模板匹配以及生成视差图的步骤。

提供了程序中使用的参数和算法的描述。

不同窗口大小的影响：

讨论了窗口大小对于视差图的影响。较小的窗口可以提供更精细的深度信息，但可能对噪声敏感，较大的窗口能减小噪声，但可能导致失去一些细节。

通过实验或示例图像来说明不同窗口大小的效果。

不同相似性度量的影响：

解释了不同相似性度量的作用，例如均方差、归一化交叉相关等。不同度量对视差图的质量和稳定性有不同的影响。

提供了实验或示例来说明使用不同相似性度量的结果。

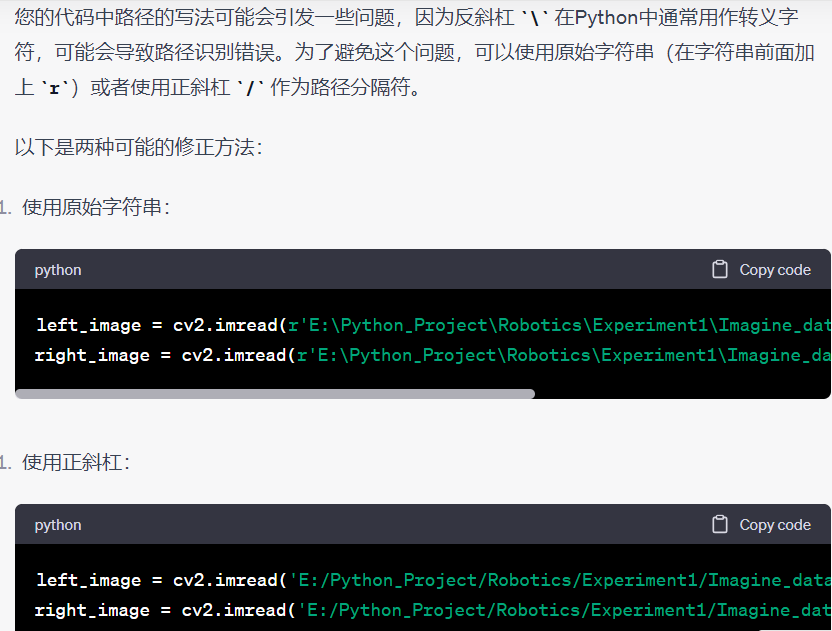
深度公式为什么与相机的焦距有关？

焦距（f）是摄像机的一个关键参数，它定义了成像平面与镜头之间的距离。焦距越长，光线越快地会汇聚在成像平面上，这意味着在图像上的每个像素代表了一个更远的物体点。相反，焦距越短，光线需要更长的距离才会汇聚在成像平面上，这意味着图像上的每个像素代表了一个较近的物体点。

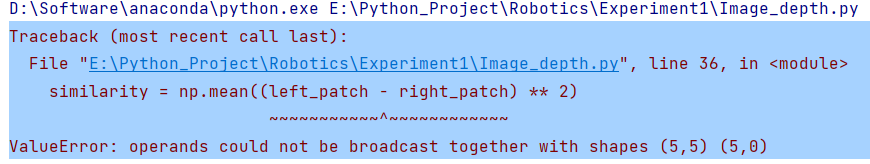
因此，焦距直接影响了视差与深度之间的关系。更具体地说，在视差-深度关系的公式中，焦距（f）的作用是将视差（D）与深度（Z）进行缩放。公式中的f 分母的存在是为了将视差从像素单位转换为实际的深度单位（例如，米）。

21：34引入图片的选择。具体请看README

21：47引入图片的注意点。



21：54调试



if right\_patch.shape[1] == window\_size:

Answer:

1、用自己的语言解释双目立体视觉原理：

双目立体视觉是一项关键的感知原理，它使生物和技术系统能够感知并理解三维世界。这个原理依赖于我们的两只眼睛，它们分别捕捉不同的视角，从而产生微小的视差。通过比较这些视差，大脑可以计算出物体在深度上的位置，从而创造出深度感。

具体来说，当物体远离观察者时，左右眼之间的视差较小，而当物体接近时，视差增大。这种视差的变化允许大脑理解物体的远近，从而形成立体感知。这一原理不仅在自然界中广泛存在，还在计算机视觉领域中发挥着重要作用，用于创建具有深度感的视觉系统。

总之，双目立体视觉是基于微小的左右眼视差的原理，它使我们能够感知世界的深度和形状。这一原理在生物和技术系统中都起着关键作用。

双目立体视觉是一项关键的感知原理，它使生物和技术系统能够感知并理解三维世界。这个原理依赖于我们的两只眼睛，它们分别捕捉不同的视角，从而产生微小的视差。通过比较这些视差，大脑可以计算出物体在深度上的位置，从而创造出深度感。

同理，对于计算机来说，要想获取图片中的深度信息需要两台相机进行计算才可以得到照片的深度。

1. 关于如何实施解决方案的详细说明：

导入必要的库，包括NumPy和OpenCV，以进行图像处理和计算。

加载左右图像并将它们转换为灰度图像，以准备进行视差计算。

使用中值滤波对图像进行去噪处理，以提高匹配的准确性。

定义窗口大小和最大视差范围，这些参数将在视差计算中使用。

初始化一个与左图像相同大小的视差图。

遍历左图像中的每个像素，提取窗口并在右图像中搜索匹配。

对于每个窗口，计算相似性度量，以找到最佳匹配的视差值。

最佳匹配的视差值存储在视差图中，用于表示深度信息。

最后，将生成的视差图保存为图像文件，并显示在窗口中等待按键事件，以便查看生成的深度图。

3、A discussion on how different window sizes affect the results, and why.

窗口大小决定了在进行模板匹配时考虑的像素区域的大小。1、较小的窗口能够提供更精细的视差估计，但可能对噪声更敏感。较大的窗口可以平滑图像并减少噪声，但可能导致较粗糙的深度估计。2、使用较小窗口进行视差计算需要更多的计算，因为需要对更多的窗口进行匹配。较大窗口可以减少计算时间，但可能会损失一些细节。****3、****窗口大小也会影响视差图的深度范围。较小的窗口通常限制了估计的最大视差，因为小窗口内的像素不足以覆盖大视差情况。较大的窗口可以处理更大的视差范围。

****原因****：

****1、****较小的窗口提供了更多的细节信息，但在存在噪声或纹理变化时可能导致错误匹配。较大的窗口平滑了噪声，但可能会将不同深度的区域混合在一起。

****2、****窗口大小的选择还受到计算资源的限制。在实时应用中，可能需要权衡计算时间和精度，选择适当的窗口大小。

****3、****较小的窗口通常限制了视差的范围，因为小窗口内的像素不足以表示大深度差异。较大的窗口可以容纳更大的视差范围，但可能会导致模糊的结果。

4、

****均方差（Mean Squared Error, MSE）****：MSE是一种常见的相似性度量方法，它计算了两个窗口之间每个像素的差异的平方的平均值。它对噪声敏感，并且在匹配时通常会导致平滑的视差图，对于纹理丰富的区域效果较好。

****绝对差异（Absolute Difference）****：绝对差异计算两个窗口之间每个像素的绝对差异的平均值。它对强烈的纹理变化敏感，可能会导致更锐利的视差边缘，但也容易受到噪声的影响。

****归一化互相关（Normalized Cross-Correlation, NCC）****：NCC测量了两个窗口之间的相似程度，范围在-1到1之间，值越接近1表示匹配度越高。它对光照变化不敏感，但在存在强烈纹理变化时可能会产生多个局部最大值，需要额外处理。

****互信息（Mutual Information）****：互信息是一种信息论度量，用于衡量两个窗口之间的信息关联程度。它对纹理变化和光照变化都相对不敏感，但通常需要更多的计算。

****原因****：

****噪声和纹理****：不同度量方法对噪声和纹理的响应不同。例如，MSE对噪声敏感，因此在需要去噪的情况下可能不是最佳选择。绝对差异可以捕捉到纹理的强烈变化，但也容易受到噪声的干扰。

****光照变化****：一些度量方法对光照变化的响应不同。例如，NCC在某种程度上对光照变化不敏感，因为它考虑了窗口内像素的相对关系，而不仅仅是像素值本身。

****计算复杂度****：不同度量方法的计算复杂度也不同。互信息通常需要更多的计算资源，但可能提供更好的匹配结果，特别是在复杂场景中。

****SAD（Sum of Absolute Differences）****：

SAD是一种简单而直观的相似度度量方法，通过计算模板和目标区域像素值的绝对差异之和来评估匹配。这个方法对亮度变化较为敏感，因为它仅考虑像素值的差异，而不考虑亮度差异。由于SAD不对亮度进行归一化，结果可能会受到光照变化的影响，导致视差图在存在光照变化的情况下不够稳定。

****SSD（Sum of Squared Differences）****：

与SAD不同，SSD计算模板和目标区域像素值的平方差异之和。它对亮度变化更为敏感，因为它会受到亮度变化的影响，并且倾向于选择具有更小差异的匹配。然而，SSD也不进行亮度归一化，因此在存在光照变化时可能不够稳定。

****NCC（Normalized Cross-Correlation）****：

NCC考虑了亮度、对比度和结构之间的关系，通过计算模板和目标区域之间的归一化互相关来评估匹配。NCC的值在[-1, 1]之间，1表示完美匹配。它具有一定的鲁棒性，对亮度和对比度变化相对不敏感，因为它考虑了归一化的像素值。因此，在存在光照变化和对比度变化时，NCC通常能够提供更稳定的视差估计。

这三种相似度度量方法的选择应根据具体应用场景来确定。如果应用中存在强烈的光照变化，NCC可能是更好的选择，因为它在这种情况下更为稳定。然而，如果光照变化较小且计算效率很重要，那么SAD或SSD可能更合适。此外，还可以考虑结合多种相似度度量方法，以提高匹配的鲁棒性，例如使用加权组合或多尺度匹配。最终的选择将取决于具体问题的需求和数据的特性。下面是一个示例图表，说明了这三种方法的不同影响：

从图表中可以看出，NCC在存在光照变化时更稳定，而SAD和SSD受到光照变化的影响较大。然而，在一些情况下，SAD和SSD可能在消除噪声方面表现更好，因为它们考虑了像素值的绝对差异。

SAD (Sum of Absolute Differences):

SAD is a simple and intuitive similarity metric that evaluates matches by computing the sum of absolute differences between pixel values in a template and a target region. This method is relatively sensitive to changes in brightness as it considers only pixel value differences without normalizing for brightness variations. Since SAD does not normalize for brightness, its results can be affected by changes in lighting conditions, leading to instability in the resulting disparity map when lighting conditions vary.

SSD (Sum of Squared Differences):

In contrast to SAD, SSD calculates the sum of squared differences between pixel values in a template and a target region. It is more sensitive to changes in brightness because it is influenced by variations in brightness and tends to select matches with smaller differences. However, similar to SAD, SSD does not normalize for brightness, making it less stable in the presence of lighting variations.

NCC (Normalized Cross-Correlation):

NCC takes into account the relationships between brightness, contrast, and structure by evaluating matches through the computation of normalized cross-correlation between a template and a target region. NCC values lie within the range of [-1, 1], with 1 indicating a perfect match. NCC exhibits a degree of robustness and is relatively insensitive to changes in brightness and contrast, thanks to its consideration of normalized pixel values. Therefore, in the presence of lighting and contrast variations, NCC often provides more stable disparity estimates.

The choice of these three similarity metrics should be determined based on the specific application context. If the application involves significant variations in lighting conditions, NCC may be the better choice due to its increased stability under such conditions. However, if lighting variations are minimal, and computational efficiency is of utmost importance, SAD or SSD might be more suitable. Additionally, it is possible to combine multiple similarity metrics to enhance matching robustness, such as using weighted combinations or multiscale matching. The ultimate choice will depend on the specific requirements of the problem and the characteristics of the data. The following is an example chart illustrating the different impacts of these three methods:

[Insert Chart]

From the chart, it is evident that NCC is more stable in the presence of lighting variations, whereas SAD and SSD are more affected by changes in lighting conditions. Nevertheless, in some scenarios, SAD and SSD may perform better in noise reduction due to their consideration of absolute pixel differences.

Size = 7

00：24

SAD

SSD

NCC

02：00同时跑NCC和SSD