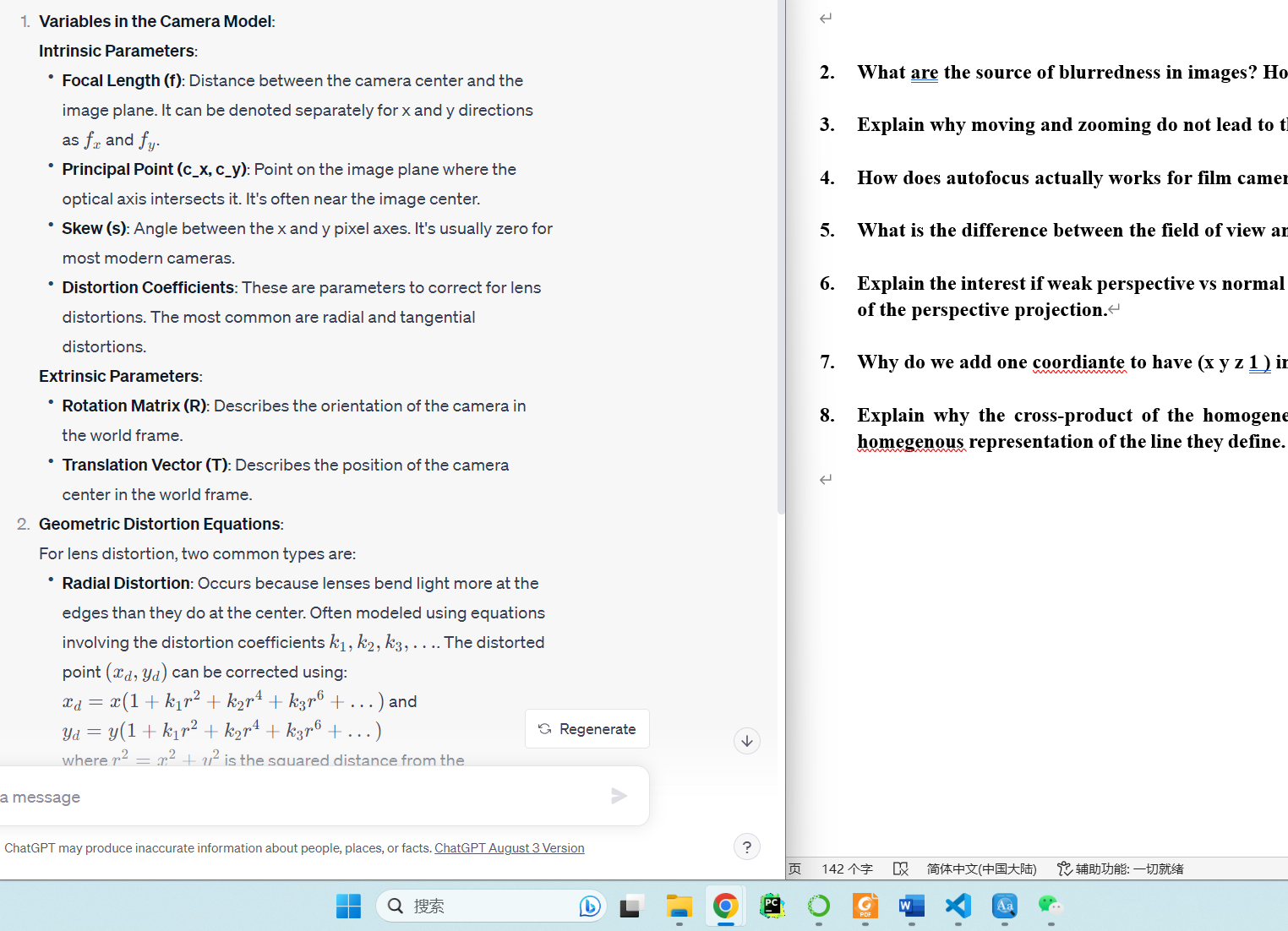
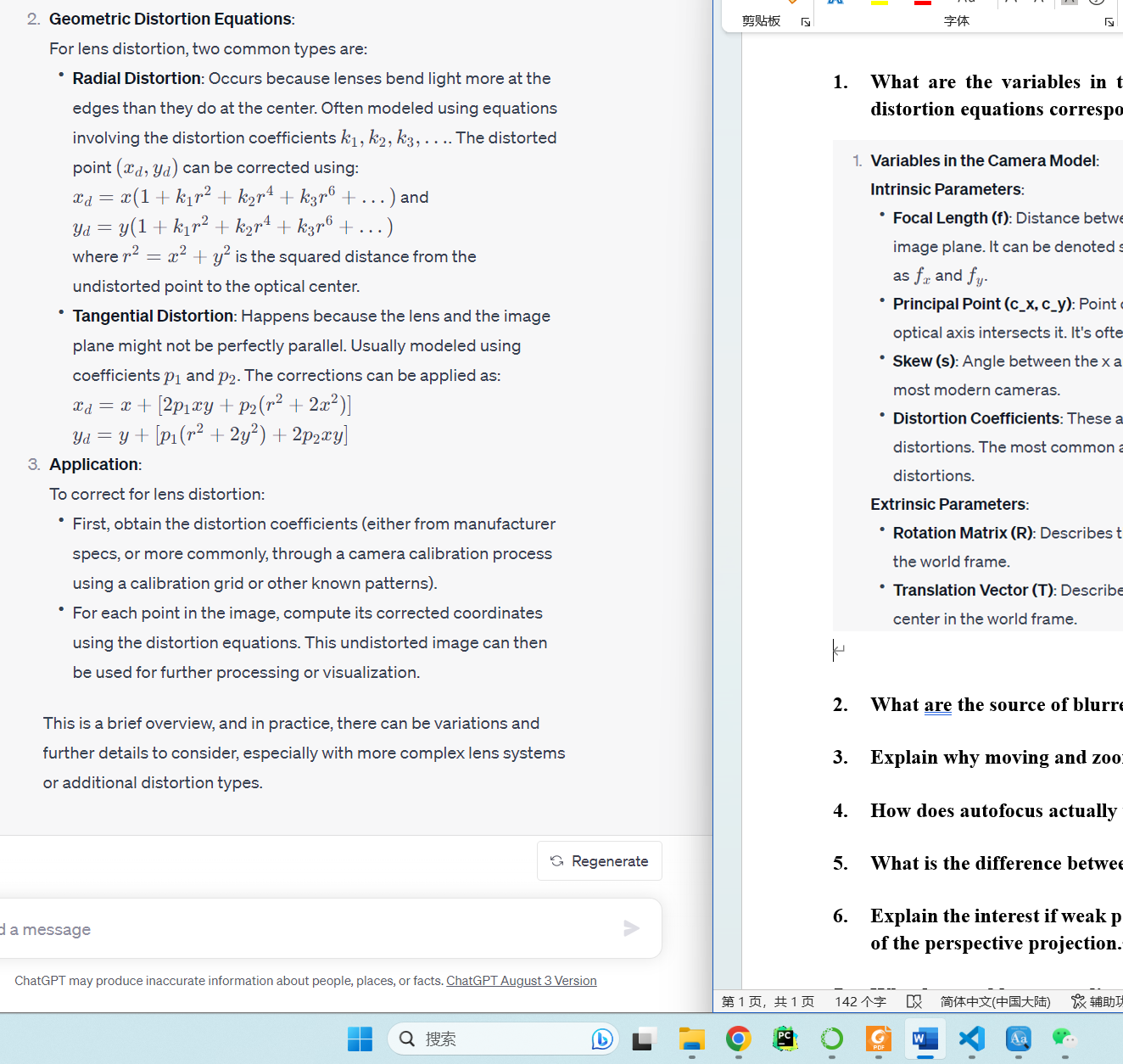
# Ch2

## Camera Model

1. What are the variables in the camera model? What do the variables in the geometric distortion equations correspond to and how do you apply them ?





1. What are the source of blurredness in images? How can they be fixed?

Blurredness in images can arise due to various reasons. Here's a brief overview:

**1. \*\*Camera Motion\*\*:** If the camera moves during exposure, it can result in motion blur.

\*\*Fix\*\*: Use a tripod or image stabilization techniques. In post-processing, deblurring algorithms that model the camera's motion can be employed.

**2. \*\*Subject Motion\*\***: If the subject of the image moves while the shutter is open, it will appear blurred.

\*\*Fix\*\*: Faster shutter speeds can capture moving subjects without blur. Alternatively, motion deblurring techniques in post-processing can help.

**3. \*\*Out-of-focus\*\***: This happens when the camera's lens isn't focused correctly on the subject.

\*\*Fix\*\*: Adjust the camera focus. In post-processing, there are deblurring techniques that attempt to reverse out-of-focus blur, but they're limited in efficacy.

4**. \*\*Diffraction\*\***: This happens when light waves encounter an obstacle or aperture and bend, especially when using very small apertures.

\*\*Fix\*\*: Avoid using the smallest aperture settings on a camera. Post-processing can't typically address diffraction-caused blur.

**5. \*\*Lens Aberrations\*\***: Imperfections in lens manufacturing can introduce various types of blur.

\*\*Fix\*\*: High-quality lenses can reduce this effect. Some software solutions can correct specific lens aberrations if the lens profile is known.

**6. \*\*Atmospheric Disturbances\*\***: This includes effects like heat shimmer or fog, which can blur distant subjects.

\*\*Fix\*\*: Shooting at different times or conditions can help. Some dehazing algorithms can improve images taken in foggy conditions.

**7. \*\*Low Resolution\*\***: When images are enlarged beyond their native resolution, they can appear blurry.

\*\*Fix\*\*: Super-resolution techniques in computer vision can enhance image resolution, albeit with limitations.

**8. \*\*Noise Reduction Algorithms\*\***: Over-aggressive noise reduction can result in a loss of image sharpness.

\*\*Fix\*\*: Reduce the aggressiveness of noise reduction settings or use sharpening filters afterward.

**9. \*\*Sensor limitations\*\***: A sensor's inability to accurately capture very fine details can give a perception of blurriness, especially in low-light conditions where sensor noise is pronounced.

\*\*Fix\*\*: Using cameras with better sensors, or shooting under better lighting conditions. Noise reduction and sharpening algorithms can be applied in post-processing.

To fix blurriness in post-processing, many computer vision and image processing techniques exist, including sharpening filters (like the Unsharp Mask), blind deconvolution methods, and deep learning-based super-resolution and deblurring algorithms. The efficacy of these methods depends on the blur type and the quality of the original image.

Explosion

1. Explain why moving and zooming do not lead to the same image.

When you move (translate) a camera versus zooming with it, the resultant images differ due to changes in perspective and the spatial relationships between objects. Here's a brief explanation:

**1. \*\*Perspective Changes\*\*:**

- \*\*Moving (Translating)\*\*: When you move closer to or farther from an object, the perspective changes. Objects closer to the camera appear larger relative to those farther away. This effect can be seen more dramatically in wide-angle lenses where objects in the foreground can appear exaggeratedly large compared to the background.

- \*\*Zooming\*\*: When you zoom in or out, you're changing the focal length of the lens but not its position. This magnifies the scene uniformly without introducing perspective changes. That's why zooming is often referred to as a "flat" change in scale.

**2. \*\*Spatial Relationships\*\*:**

- \*\*Moving (Translating)\*\*: As you move, the relative positions of objects within the frame change due to the change in perspective. This phenomenon is termed "parallax". For instance, when walking past a row of trees, the gaps between the nearer trees will appear to change more quickly than the gaps between distant trees.

- \*\*Zooming\*\*: Zooming only changes the size of objects in the frame and doesn't alter the relative positions of objects or the spatial relationships between them, as it lacks the parallax effect.

**3. \*\*Amount of Scene Captured\*\*:**

- \*\*Moving (Translating)\*\*: Moving closer to a subject might exclude some background elements from the frame, while moving away may introduce new elements.

- \*\*Zooming\*\*: Zooming in narrows the field of view, thereby excluding some peripheral details, while zooming out broadens the field of view to include more of the scene.

In essence, while both moving and zooming can make a subject appear larger or smaller in the frame, the effects on perspective, spatial relationships, and the content of the scene differ between the two actions. As a result, the resultant images are distinct.

1. How does autofocus actually works for film cameras

Autofocus (AF) in film cameras operates on principles that detect contrast or phase differences to achieve focus. Here's a brief overview of how autofocus works in film cameras:

**1. \*\*Active Autofocus\*\*:**

- This system emits a signal (usually infrared light) towards the subject. The camera then measures the time it takes for the signal to bounce back. Based on this time and the speed of light, the camera computes the distance to the subject and adjusts the lens accordingly.

- Active AF can struggle with transparent or highly reflective subjects.

**2. \*\*Passive Autofocus\*\*:**

- \*\*Contrast Detection\*\*: This method adjusts the lens to maximize the contrast in the image, which typically corresponds to the sharpest focus. The camera moves the lens and checks the contrast of the resultant image, repeating the process until the highest contrast (and therefore best focus) is achieved. It's more accurate than phase detection but is generally slower.

- \*\*Phase Detection\*\*: This method splits the incoming light into two images and then adjusts the lens to make the two images overlap perfectly. It does this using a beam splitter or semi-transparent mirror inside the camera. When the two split images are perfectly aligned, the subject is in focus. Phase detection is faster than contrast detection but can sometimes be less accurate in low light.

**3. \*\*Focus Motors\*\*:**

- Once the camera determines how to adjust the focus (whether closer or farther), it uses a motor to move the lens elements. Some older film cameras required the lens to have its own motor, while others had a built-in motor that engaged with the lens.

**4. \*\*Manual Override\*\*:**

- Many autofocus film cameras still allowed for manual focus adjustment, often with an AF/MF switch on the camera or lens.

It's important to note that while the principles of autofocus remain the same between film and digital cameras, advancements in technology have refined and expanded these systems, especially with the introduction of digital sensors and increased computational power in modern digital cameras.

1. What is the difference between the field of view and the focal length?

Both the field of view (FOV) and the focal length are crucial concepts in photography and optics, and they are interrelated. Here's a brief explanation of each and their differences:

**1. \*\*Focal Length\*\*:**

- \*\*Definition\*\*: *The focal length of a lens is the distance (usually measured in millimeters) between the lens's optical center (or nodal point) and the image sensor (or film) when the lens is focused at infinity.*

- \*\*Effect on Image\*\*: The focal length determines how "zoomed in" your photos appear. A shorter focal length (e.g., 18mm) provides a wider angle of view and captures more of the scene. A longer focal length (e.g., 200mm) offers a narrower angle of view, making distant subjects appear larger and closer.

- \*\*Types\*\*: Lenses can be classified based on their focal lengths. "Wide-angle" lenses have short focal lengths, "standard" lenses have medium focal lengths, and "telephoto" lenses have long focal lengths.

**2. \*\*Field of View (FOV)\*\*:**

- \*\*Definition\*\*: *FOV describes the angular extent of the observable scene that is captured by the camera. It's often expressed in degrees.*

- \*\*Effect on Image\*\*: A larger FOV captures a broader scene, whereas a smaller FOV captures a more limited portion of the scene. For instance, a fish-eye lens has an extremely wide FOV, while a telescope has a very narrow one.

- \*\*Dependency\*\*: **FOV is dependent on the focal length and the size of the image sensor (or film). Two lenses with the same focal length might produce different FOVs on cameras with different sensor sizes.**

\*\*Relationship\*\*:

**The focal length and FOV have an inverse relationship. As the focal length increases, the FOV decreases, and vice versa. This means that as you use a lens with a longer focal length, you "zoom in" on your subject, resulting in a narrower field of view. Conversely, using a lens with a shorter focal length gives you a "wider" perspective with a broader field of view.**

In summary, while focal length is a physical property of the lens that indicates how "zoomed in" an image will appear, the field of view is the actual extent of the scene captured by the camera, and it's determined by both the lens's focal length and the camera's sensor size.

1. Explain the interest if weak perspective vs normal perspective. Explain the degenerate cases of the perspective projection.

Let's break down the concepts of weak perspective, normal perspective, and the degenerate cases of perspective projection:

1. \*\*Weak Perspective vs. Normal Perspective\*\*:

- \*\***Normal (or Full) Perspective Projection\*\*: In this model, the projection of a 3D point onto the image plane is dependent on its depth relative to the camera.** Different points in the scene can have different depths, causing them to be projected differently on the image plane. This provides a realistic representation of the 3D world in 2D images.

- \*\***Weak Perspective (or Scaled Orthographic) Projection\*\*: This arises when objects are far away from the camera compared to their size. Given the distance, the change in depth of the object is relatively small compared to the average distance from the camera.** So, all points of the object can be assumed to be at roughly the same distance from the camera. In weak perspective, the image of an object is essentially its orthographic projection scaled by a constant value. It's simpler than the full perspective model and is often used in computer vision tasks for distant objects or when precision isn't a top concern.

- \*\*Interest\*\*:

**- Normal Perspective offers a more accurate and realistic representation of the scene, capturing depth cues and relative sizes of objects based on their distance from the camera.**

**- Weak Perspective is computationally simpler and is a good approximation when depth variation is minimal. It's useful in scenarios like facial recognition from a distance where the depth variation in a face is negligible compared to its distance from the camera.**

2. \*\*Degenerate Cases of Perspective Projection\*\*:

**Degenerate cases in perspective projection occur when the geometry of the scene or the relative configuration of scene structures leads to ambiguities or indeterminacies in the projected image.** Common degenerate cases include:

- \*\*Collinearity\*\*: When three or more points lie on the same line. In perspective projection, this can make it challenging to determine their relative positions.

- \*\*Concurrence\*\*: When three or more lines meet at a single point.

- \*\*Coplanarity\*\*: When all points of interest lie on the same plane. This can lead to issues, especially in structure-from-motion tasks, because if all points are coplanar, there's an inherent ambiguity in recovering the 3D structure.

- \*\*Multiple Structures with Parallel Projection Lines\*\*: This can lead to ambiguities as the depth cannot be easily determined from the projections.

Understanding these degenerate cases is crucial when working with perspective transformations in computer vision because they can introduce errors or ambiguities in tasks like 3D reconstruction or camera calibration.

1. Why do we add one coordiante to have (x y z 1 ) in homogenuous coordinates ?

Homogeneous coordinates introduce an additional dimension to represent points and transformations in a unified way, especially in projective geometry. This system is frequently used in computer graphics and computer vision.

Here's why we use homogeneous coordinates:

1. **\*\*Unified Representation for Points and Vectors\*\***: In homogeneous coordinates, points and vectors can be represented using the same data structure, a 4-element tuple for 3D or a 3-element tuple for 2D.

2. **\*\*Representation of Points at Infinity\*\***: Homogeneous coordinates allow for the representation of points at infinity. For example, in 2D, the point \( (x, y, 1) \) in homogeneous coordinates corresponds to the point \( (x, y) \) in Cartesian coordinates. But, \( (x, y, 0) \) represents a point at infinity in the direction \( (x, y) \).

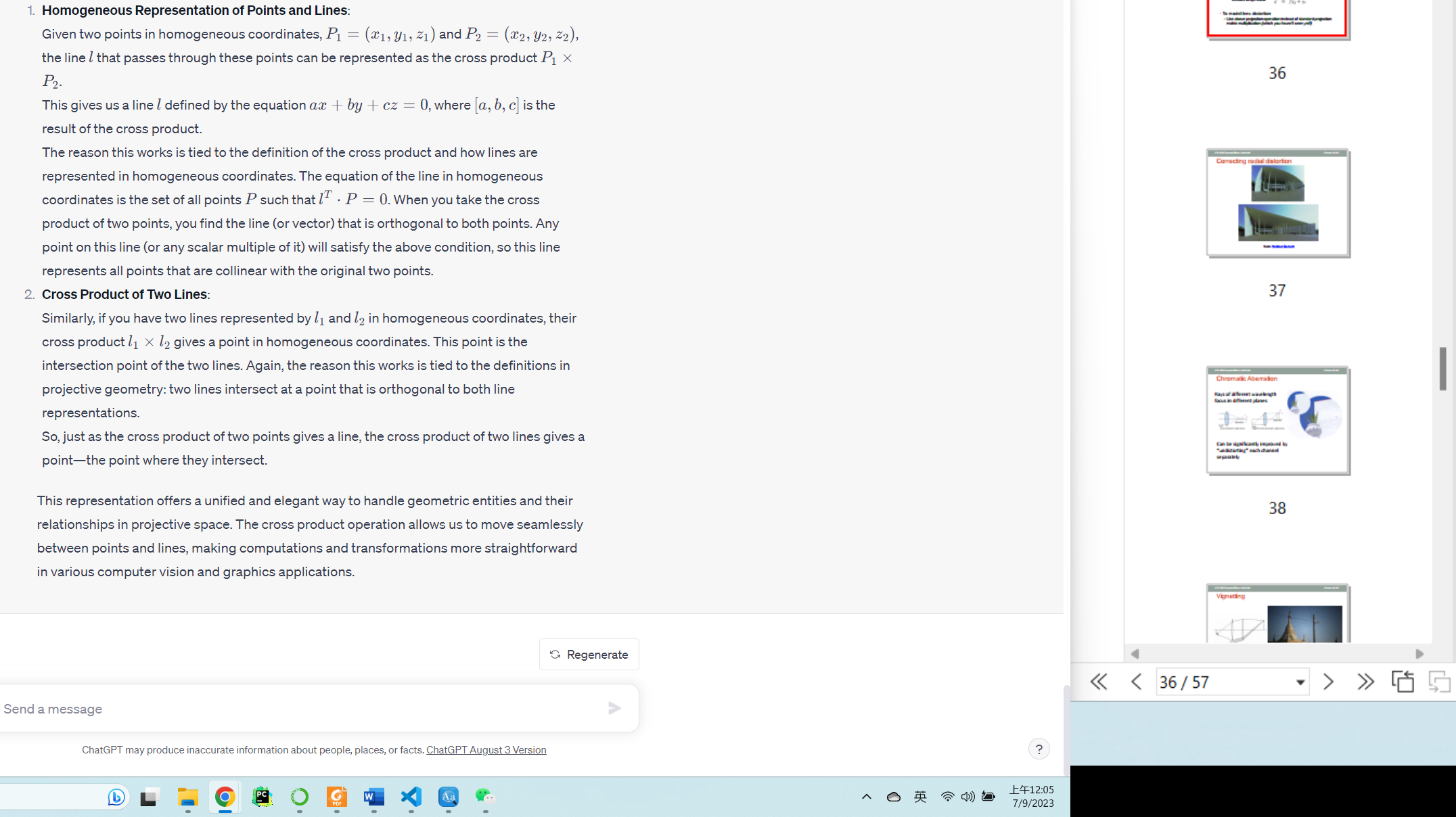
3. **\*\*Simplifying Transformations\*\*:** Homogeneous coordinates simplify the representation and application of geometric transformations, especially translations. Rotations, scalings, and translations can all be represented using a single matrix multiplication if points are in homogeneous coordinates.

4. **\*\*Projective Transformations\*\***: Homogeneous coordinates are essential for projective transformations, which are more general than affine transformations. In computer vision, especially in camera projection, using homogeneous coordinates simplifies the mathematical formulations.

5. **\*\*Avoiding Special Cases\*\***: By using homogeneous coordinates, certain operations can be carried out without having to consider special cases. For instance, parallel lines, which never meet in Euclidean space, will meet at a point at infinity in projective space.

In summary, adding an extra coordinate to represent 3D points in 4D homogeneous coordinates offers computational convenience, simplifies mathematical formulations, and provides a unified framework for handling a range of geometric operations and scenarios.

1. Explain why the cross-product of the homogeneous coordinates of two points gives the homegenous representation of the line they define. What is the "cross-product of two lines"?



## Stereo

1. How to choose the size of the window for correlation-based window matching? Why do we want window small enough to have pixels with the same disparity ?

The basic idea is to slide a window over one image and search for a matching window in the other image to find the disparity between matching points. The size of the window plays a crucial role in the accuracy and reliability of the matching.

Here's how to choose the window size and the importance of having pixels with the same disparity within the window:

1. **Resolving Ambiguity**: Smaller windows are less likely to contain multiple objects at different depths. This means that the pixels within that window are likely to have the same disparity, reducing the chance of ambiguous matches.
2. **Handling Occlusions**: In real-world scenes, occlusions happen when an object in one image is covered by another object in the other image. A smaller window reduces the risk of including occluded pixels, which would otherwise distort the correlation score and lead to incorrect matches.
3. **Noise and Texture**: Smaller windows are more sensitive to noise and less textured regions. In regions with little or no texture, small windows might not provide enough information to determine a reliable match. This is why sometimes adaptive window sizes or methods that factor in the texture content of the region can be beneficial.
4. **Computational Cost**: Smaller windows are faster to compute since there are fewer pixels to correlate, but if the window is too small, there might not be enough information for a reliable match. Conversely, larger windows increase computational costs but can be more robust in low-texture regions.
5. **Disparity Gradient**: If the window is too large, it might span areas with varying disparities. If the disparity changes rapidly within the window (due to the presence of objects at varying depths), the matching algorithm might not produce accurate results.
6. **Resolution and Detail**: Smaller windows can capture finer details and nuances in the image. Larger windows might smooth out these details, which can be problematic when trying to match intricate patterns or when the images have high resolution.

To choose the optimal window size:

1. **Analyze Your Dataset**: It's a good idea to start by visualizing a few examples from your dataset. Look for areas with rapid depth changes, occlusions, and varying textures. This will give you insights into potential challenges and guide your choice.
2. **Experimentation**: It's often beneficial to try out several window sizes and evaluate their performance. Performance can be measured in terms of matching accuracy, computational speed, and the number of false matches.
3. **Adaptive Methods**: Consider methods that adapt the window size based on the content. For instance, in regions with high texture, a smaller window might suffice, while in smoother areas, a larger window might be necessary.

In conclusion, choosing the right window size for correlation-based matching is a balance between various factors. It's essential to ensure that the window is small enough to contain pixels with the same disparity to reduce ambiguity and improve the accuracy of the matches.

1. Why do we want to solve the correspondance problem? How do you choose the corresponding points ?

The correspondence problem in stereo matching refers to the challenge of finding the same point in two (or more) images taken from slightly different viewpoints. **Solving this problem allows us to determine the disparity (the difference in position) of that point between the two images. By knowing the disparity and the geometry of the stereo cameras, we can calculate the depth or distance of that point from the cameras, enabling 3D reconstruction of the scene.**

Depth map is useful to enable 3D reconstruction of the scene.

Reason to solve the correspondence problem:

- \*\*Depth Estimation\*\*: Once we know which points in one image correspond to points in the other image, we can determine the depth or 3D structure of the scene.

Choosing the corresponding points:

1. \*\***Feature-based Matching**\*\*: Identify unique features (like corners or other distinct patterns) in one image and find the same features in the other image.

2. \*\***Area-based Matching (or Block Matching)\***\*: Use small windows (or blocks) and search for the most similar window in the other image using a similarity measure, such as Sum of Squared Differences (SSD) or Normalized Cross-Correlation.

3. \*\*Constraint-based Matching\*\*: Use constraints like epipolar geometry, which dictates that the corresponding point must lie on a known line (the epipolar line) in the other image.

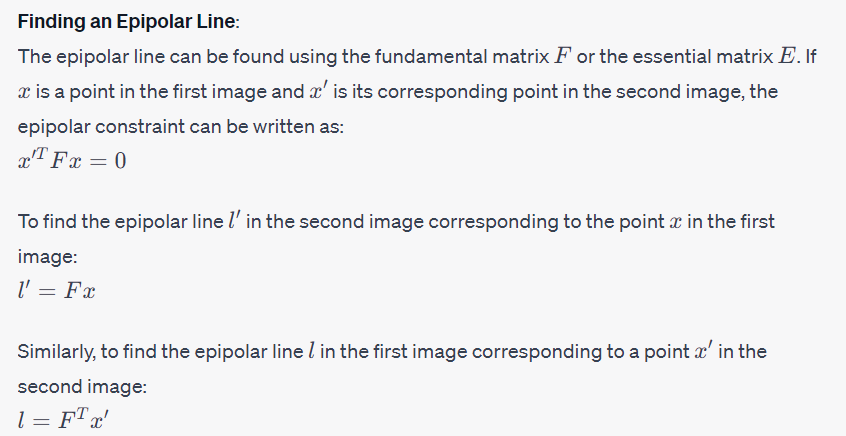
4. \*\*Global Optimization Methods\*\*: Techniques like graph cuts or dynamic programming to ensure that the selected correspondences are globally consistent and respect smoothness constraints in the disparity map.

Choosing the right method or combination of methods depends on the application, image characteristics, and desired accuracy.

1. What is the epipolar constraint used for ? How to find an epipolar line? Can assuptions we make on epipolar problems induce false results?

Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view. The epipolar constraint can reduce the correspondence problem to a 1D search along an epipolar line.

The epipolar constraint is a geometric constraint that arises in stereo vision due to the coplanarity of the camera centers and a scene point in both images. This constraint states that the corresponding point of a point in one image must lie on a specific line in the other image, known as the epipolar line.



**Assumptions and Potential for False Results**:

1. **Camera Calibration Errors**: If the cameras are not perfectly calibrated, the computed fundamental or essential matrix might be inaccurate, leading to incorrect epipolar lines and thus false matches.
2. **Occlusions**: The epipolar constraint assumes the visibility of corresponding points in both images. If a point is visible in one image but occluded in the other, the epipolar constraint might lead to incorrect correspondences.
3. **Repetitive Patterns**: In scenes with repetitive textures or patterns, the epipolar constraint might not be enough to disambiguate the correct correspondence since many points along the epipolar line might look similar.
4. **Non-Lambertian Surfaces**: Surfaces that don't exhibit Lambertian reflection (like shiny or reflective surfaces) can violate the brightness constancy assumption, leading to potential mismatches.
5. **Assumption of Pinhole Camera Model**: The epipolar geometry is derived from the pinhole camera model. If the cameras have significant lens distortions that aren't accounted for, the derived epipolar lines might be inaccurate.

In practice, while the epipolar constraint is a powerful tool, it's often combined with other constraints or matching techniques to ensure robust correspondence in a variety of scenes and scenarios.

1. What is SSD? And what is it useful for? Why are the advandages/disadvantages of using SSD over normalized correlation for window matching over the epipolar line?

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

**SSD stands for Sum of Squared Differences**. It is a similarity measure used in image processing and computer vision, especially for tasks like template matching or stereo matching. Given two image patches or windows, \(A\) and \(B\), the SSD between them is computed as:

\[ SSD(A, B) = \sum\_{i,j} (A(i,j) - B(i,j))^2 \]

Here, \(i,j\) index the pixels in the windows \(A\) and \(B\).

\*\*Uses of SSD\*\*:

1. \*\***Stereo Matching\*\*: To find correspondences between two images based on similarity of windows.**

2. \*\*Template Matching\*\*: To locate a given template or patch within a larger image.

3. \*\*Motion Estimation\*\*: In video processing, to estimate motion vectors between consecutive frames.

\*\***Advantages** of using SSD over Normalized Correlation for Window Matching\*\*:

1. \*\***Simplicity**\*\*: SSD is computationally simple and straightforward to implement.

2. \*\***Faster Computation**\*\*: Depending on the implementation, SSD can be computed faster than normalized correlation.

3. \*\***Sensitivity to Differences**\*\*: SSD is sensitive to even small differences between windows, which can be useful if you want exact matching.

\*\***Disadvantages** of using SSD over Normalized Correlation\*\*:

1. \*\***Brightness and Contrast Variations**\*\*: SSD is sensitive to brightness and contrast variations. If two windows have the same content but one is brighter or has higher contrast than the other, their SSD will be high. Normalized correlation accounts for such variations and can give a high similarity value for such cases.

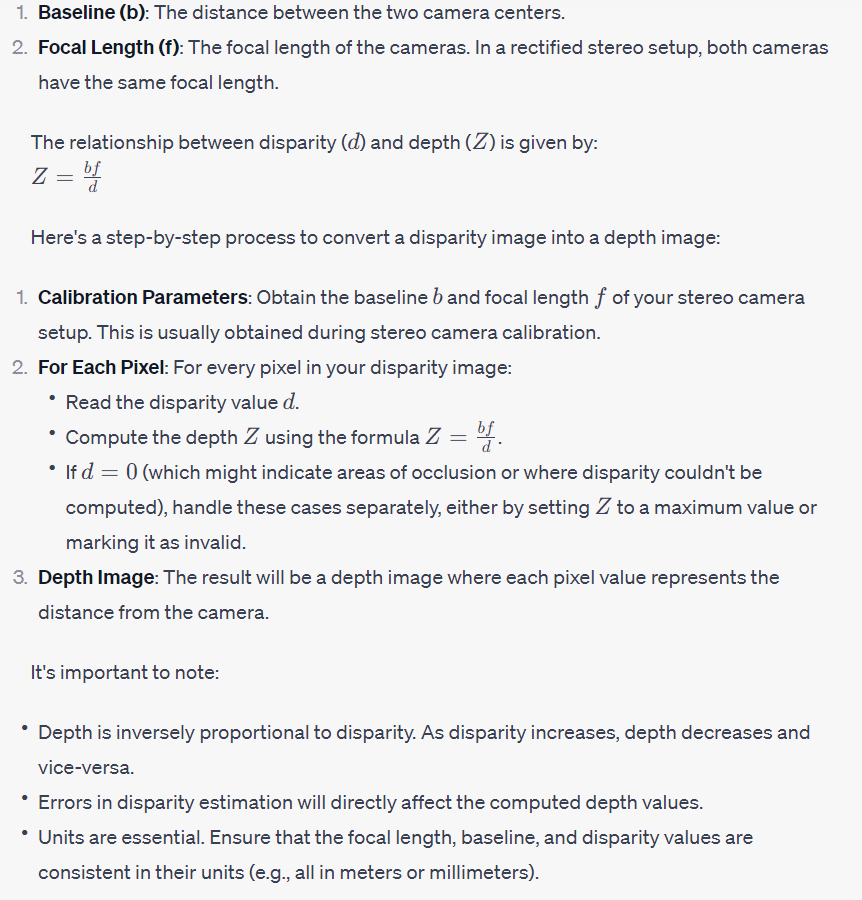
2. \*\***Value Range**\*\*: The value of SSD can range widely based on the content of the windows and their differences. In contrast, normalized correlation values lie between -1 and 1, making them easier to threshold or interpret.

3. \*\***Noise Sensitivity**\*\*: SSD is sensitive to noise. Even a small amount of noise can significantly increase the SSD value, potentially leading to incorrect matches.

4. \*\***Local Optima**\*\*: Especially in stereo matching, SSD can sometimes lead to local optima where a wrong correspondence might have a lower SSD than the correct one, especially in regions with repetitive or similar textures.

In conclusion, the choice between SSD and normalized correlation depends on the specific application and the nature of the images being processed. It's often beneficial to experiment with both methods and even consider hybrid methods that combine the strengths of both.

1. How do you transform a disparity image into a depth image?



1. How to proceed to compute disparity when epipolar lines are not horizontal and at the same y location in both images?

When epipolar lines are not horizontal, it implies that the stereo images are not rectified. **Rectification is the process of transforming the images such that the epipolar lines are horizontal and correspondences lie on the same horizontal line in both images, making disparity computation straightforward.**

To compute disparity when epipolar lines are not horizontal, follow these steps:

**1. \*\*Stereo Calibration\*\***: Before anything, you need to calibrate the stereo camera setup. This process will give you intrinsic parameters (like focal length and principal point) for each camera and extrinsic parameters (like rotation and translation between the two cameras).

2. \***\*Rectification\***\*: Use the calibration results to rectify the stereo images. Rectification will warp the images such that epipolar lines become horizontal and lie at the same y-coordinates in both images. Most modern computer vision libraries like OpenCV provide functions to rectify stereo images using the results from stereo calibration.

3. \*\***Disparity Computation**\*\*: Once the images are rectified, disparity computation becomes much easier:

- For each pixel in the left image, you now only need to search along the same horizontal line in the right image (since epipolar lines are now horizontal).

- Use a matching cost (like SSD, normalized correlation, etc.) to find the best match for each pixel or window.

4. \*\***Post-processing**\*\* (optional but often beneficial):

- \*\*Consistency Check\*\*: Compare the disparity maps obtained by matching left-to-right and right-to-left. If they are inconsistent for a pixel, it might be an error.

- \*\*Filtering\*\*: Use median filtering or bilateral filtering to smooth the disparity map while preserving edges.

- \*\*Interpolation\*\*: Fill in any gaps or areas where disparity couldn't be computed (e.g., due to occlusions).

5. \*\***Depth Calculation**\*\*: If required, you can now compute the depth from the disparity using the formula discussed in the previous answer.

The crucial step here is rectification. Once the images are rectified, disparity computation becomes a one-dimensional search problem, simplifying the process and reducing computational requirements.

1. What is the difference between 3D TV (no glasses) and 3D cinema (glasses) ?

From a computer vision perspective, both methods leverage the principle of binocular stereo vision (how our eyes perceive depth based on two slightly different views). However, the technologies focus on different strategies to ensure each eye receives its unique view. The difference lies in the technology used to separate these images for each eye.

**3D TV (No Glasses) - Autostereoscopic Displays**:

**Technology:** Uses lenticular lenses or parallax barriers to direct light in different directions.

**How It Works:** Different parts of the image are directed towards the left and right eyes without the need for glasses. The viewer needs to be in specific 'sweet spots' to get the 3D effect.

**Pros:** No glasses needed, making the viewing experience more comfortable for some.

**Cons:** Limited viewing angles, potential loss of resolution, and not as effective as glasses-based systems for producing a strong 3D effect.

**3D Cinema (With Glasses):**

**Technology:** There are various technologies, but the most common ones are polarized glasses and active shutter glasses.

**How It Works:**

**Polarized Glasses:** Two synchronized projectors display the left and right images using different polarizations. The glasses ensure that each eye only sees the image with the matching polarization.

**Active Shutter Glasses:** A single projector alternates quickly between the left and right images. The glasses sync with the projector, "shuttering" each lens in alternation to match the displayed image.

**Pros**: Typically better 3D depth perception, wider viewing angles, and better image quality.

**Cons:** Requires wearing glasses, which some viewers might find uncomfortable or distracting.