Korea University

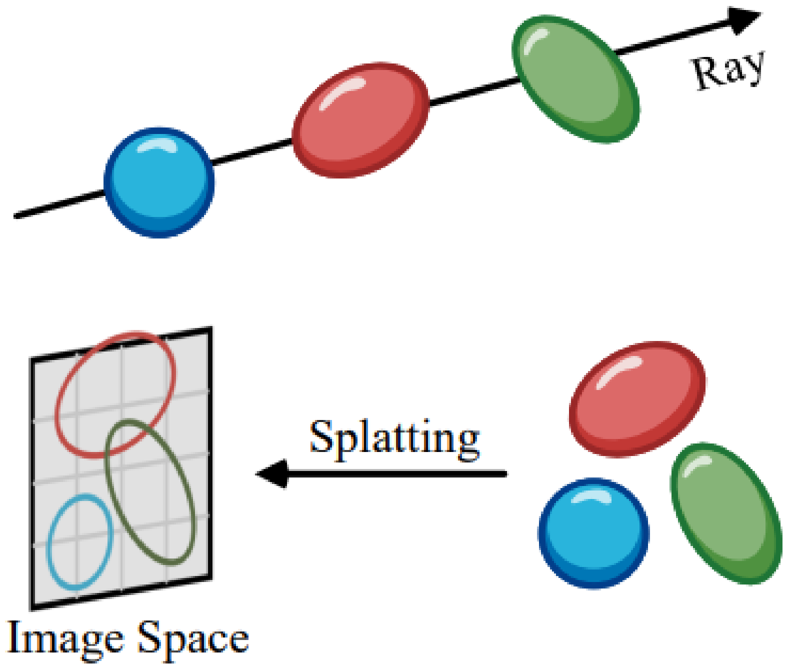
Computer Graphics
- Practice -

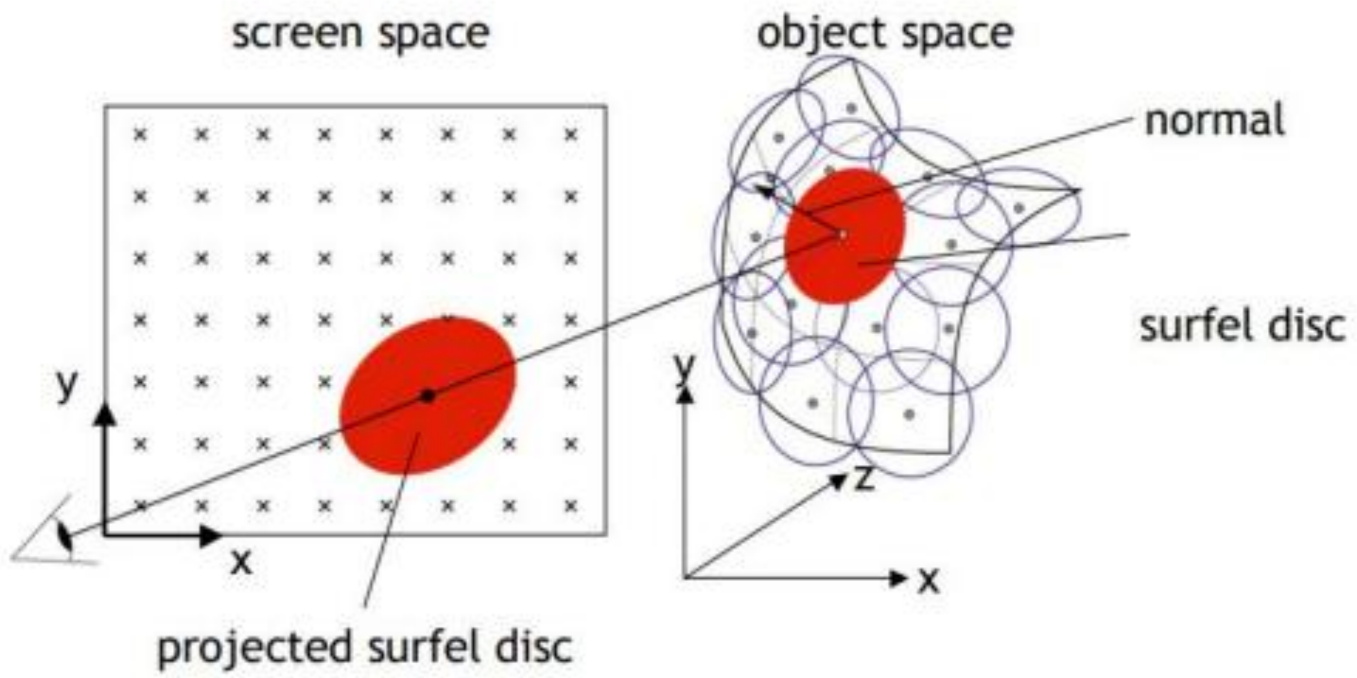
TA
IIIXR LAB
Kim Byungmin
kbmstar1@korea.ac.kr

Chapter I

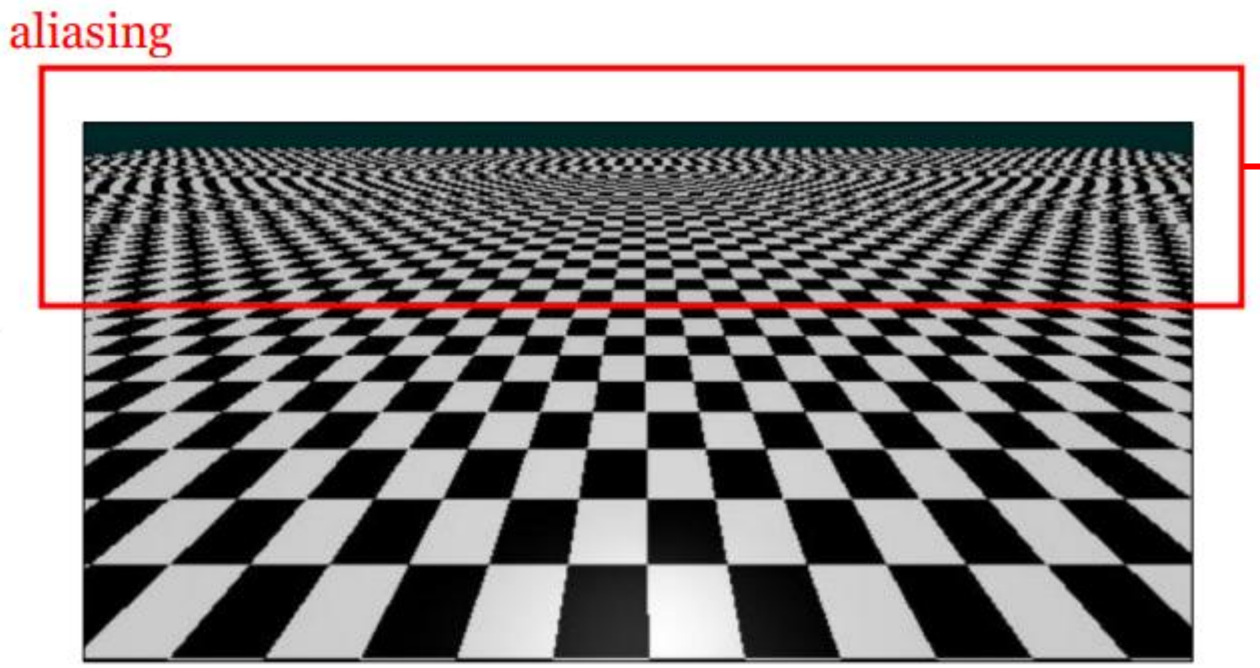
Assignment 2

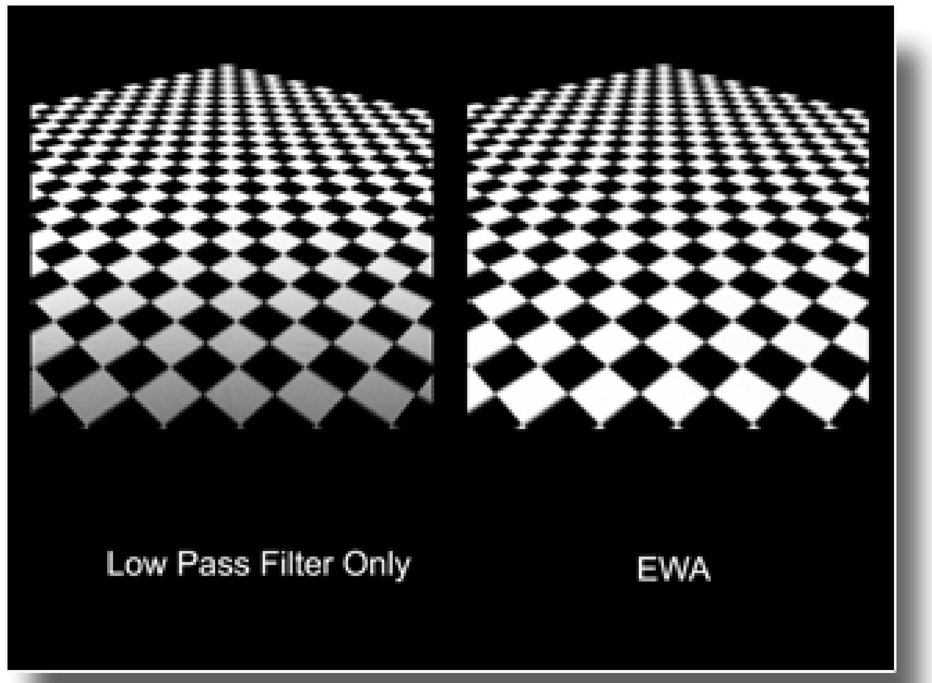
- During the theory lecture, we discussed the aliasing problem in 3D Gaussian Splatting.
- Nyquist-Shannon sampling theorem
  - It is one of the fundamental principles of digital signal processing, stating that to perfectly reconstruct a signal, it must be sampled at a rate of at least twice its highest frequency.
  - If the sample rate is constant but the frequency (e.g., the projection result of a Gaussian) is too high, the signal cannot be processed correctly, leading to aliasing issues.
  - Furthermore, if a projected 3D Gaussian becomes smaller than a single pixel, situations where the Gaussian is either 'captured' or 'missed' by the pixel can occur, leading to undesirable phenomena such as flickering.





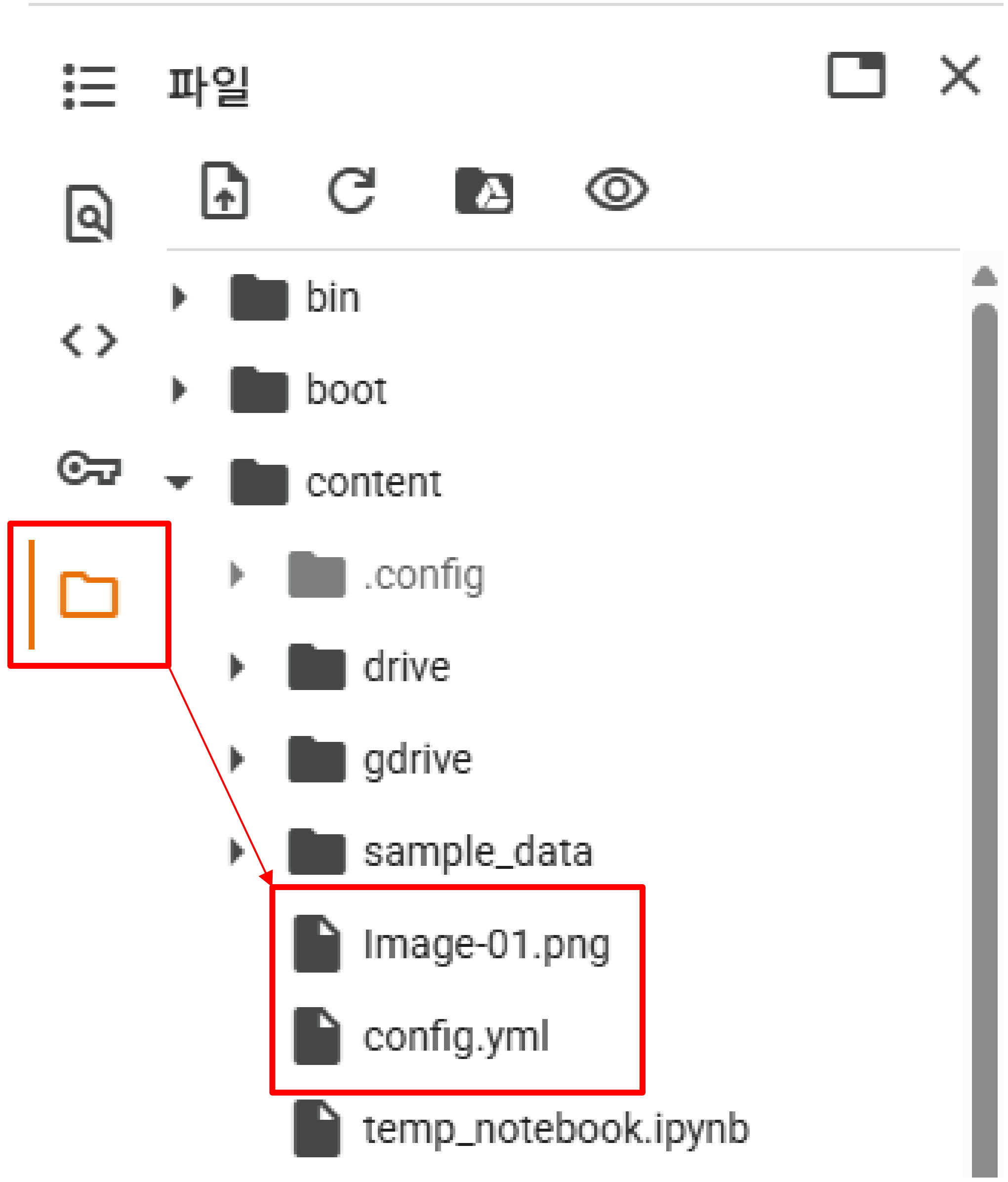
- To address this, we perform a smoothing operation using either a Low-pass filter or an EWA (Elliptically Weighted Average) filter.
- Low-pass filter
  - This method forcibly blurs Gaussians to prevent them from becoming too small, thereby removing high-frequency components.
- EWA filter
  - This method involves applying convolution, giving higher weights to Gaussians closer to a texture pixel (texel) and lower weights to distant ones, thereby achieving blending.
- For more detail, please refer [lecture 10 – Gaussian splatting].





- In this assignment, you will implement Low-pass and EWA filters and compare their results.
- The base code (.ipynb) will be provided. After implementing the Low-pass and EWA filters, you will compare their results against the Baseline and then convert your ipynb file to HTML for submission.
- First, I will guide you on how to run the base code.

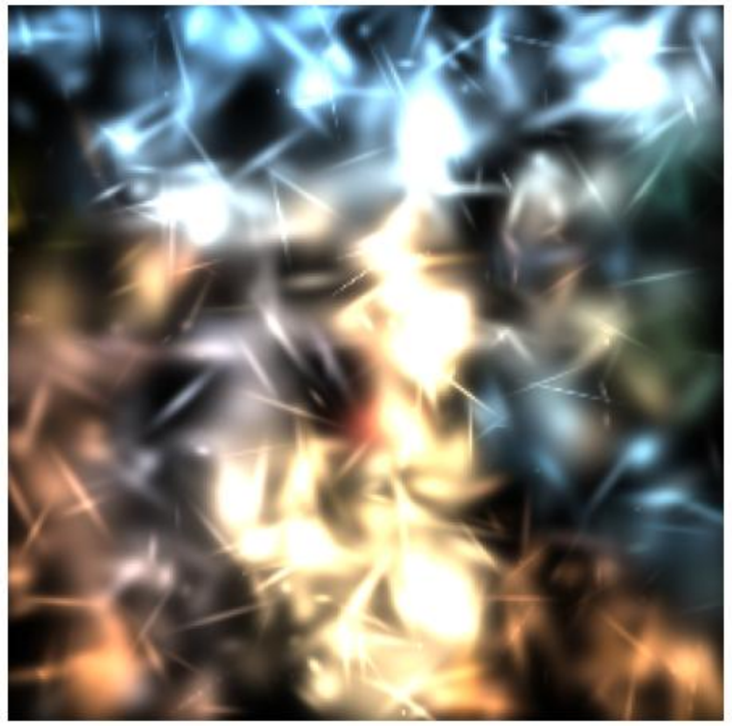
- 1. Upload your desired PNG image file and Config.yml to the Content/ folder.
- Uploading can be done via drag-and-drop.



- 2. Execute the code cells sequentially and verify that the Main training loop runs correctly.

```python
# Note: This main training loop uses generate_2D_gaussian_splatting internally.
# If you want filters to apply during training, you'd pass filter_type
# to the generate_2D_gaussian_splatting call within this loop.
# For this exercise, we are focusing on filter comparison in a separate cell.

for epoch in range(num_epochs):

    #find indices to remove and update the persistent mask
    if epoch % (densification_interval + 1) == 0 and epoch > 0:
        indices_to_remove = (torch.sigmoid(W[:, 3]) < 0.01).nonzero(as_tuple=True)[0]

        if len(indices_to_remove) > 0:
          print(f"number of pruned points: {len(indices_to_remove)}")

        persistent_mask[indices_to_remove] = False

        # Zero-out parameters and their gradients at every epoch using the persistent mask
        W.data[~persistent_mask] = 0.0
```

2D Gaussian Splatting



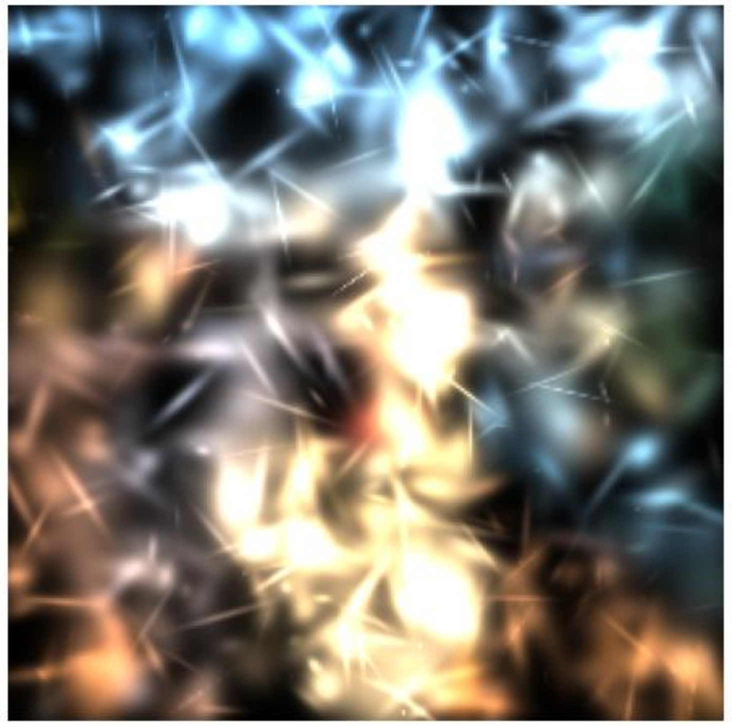
Ground Truth

Epoch 1/2001, Loss: 0.22992849349975586, on 1000 points

- 3. Implement the Low-pass / EWA filters and generate results.

```python
if filter_type == 'low_pass':
    # Implement Low-Pass Filter:
    # Goal: If a Gaussian is too small (causing aliasing), make it slightly larger (blurrier).
    # 1. Determine a 'threshold' for the Gaussian's size (e.g., related to pixel dimensions).
    # 2. Calculate the eigenvalues of the 'covariance' matrix to determine its spread.
    # 3. Identify Gaussians whose spread (eigenvalues) falls below the threshold.
    # 4. For these identified Gaussians, add a small isotropic (uniform) blur amount
    #    to their 'covariance' matrix (e.g., add 'blur_amount_square * torch.eye(2)' to it).

    # YOUR CODE HERE FOR LOW-PASS FILTER
    pass # Placeholder, remove this line when implementing

elif filter_type == 'ewa':
    # Implement EWA Filter:
    # Goal: Convolve the Gaussian with a pixel filter Gaussian.
    # This conceptually means the final rendered Gaussian's covariance is the sum
    # of the original Gaussian's covariance and the pixel filter's covariance.
    # 1. Define a 'pixel_radius' (e.g., 0.5 for a standard pixel).
    # 2. Create the covariance matrix for the pixel filter (Sigma_pixel) based on 'pixel_radius'.
    #    It should be an isotropic (circular) blur.
    # 3. Add Sigma_pixel to the original 'covariance' matrix. This applies to all Gaussians.

    # YOUR CODE HERE FOR EWA FILTER
    pass # Placeholder, remove this line when implementing
```

2D Gaussian Splatting



Ground Truth

Epoch 1/2001, Loss: 0.22992849349975586, on 1000 points

- 4. After answering Discussion Q1 and Q2, execute the final code cell.
- For discussion, **you can answer either in English or Korean.**
- Executing the final code cell will prompt you to connect with Google Drive.
- In Google Colab, saving as an HTML file requires temporary authorization to access Google Drive.
- After successful connection, you can navigate to the saved HTML file's location and download it.

```
[14] drive.mount('/content/drive')
     notebook_name = 'Test.ipynb'
     !cp "/content/drive/MyDrive/Colab Notebooks/{notebook_name}" "/content/temp_notebook.ipynb"

     output_dir = '/content/gdrive/MyDrive/Colab Notebooks/'
     output_html_name = 'MyNotebook_output.html'

     !jupyter nbconvert --to html "/content/temp_notebook.ipynb" --output-dir="{output_dir}" --output="{output_html_name}"

     print(f"HTML file saved to: {output_dir}{output_html_name}")
```

```
Mounted at /content/drive
[NbConvertApp] Making directory /content/gdrive/MyDrive/Colab Notebooks/
[NbConvertApp] Converting notebook /content/temp_notebook.ipynb to html
[NbConvertApp] WARNING | Alternative text is missing on 1 image(s).
[NbConvertApp] Writing 1106532 bytes to /content/gdrive/MyDrive/Colab Notebooks/MyNotebook_output.html
HTML file saved to: /content/gdrive/MyDrive/Colab Notebooks/MyNotebook_output.html
```

- TO-DO LIST
  1. Set up the environment to run the base code.
  2. Implement the Low-pass filter.
  3. Implement the EWA filter.
  4. Verify successful execution, then answer Discussion Q1 and Q2.
  5. Convert to an HTML file and submit.

- Scoring criteria (10 points)
  - Run the base code (2 points)
  - Implement the Low-pass filter (2 points)
  - Implement the EWA filter (2 points)
  - Discussion answering (2 points each)

- You have to submit .html file instead of .ipynb file. Violation about this will be scored to zero.