**Photogrammetry**

1. **Introduction to Photogrammetry:**

**Structure-from-Motion (SfM) and Multi-View Stereo (MVS) Photogrammetry: A Brief Overview:**

Structure-from-Motion (SfM) and Multi-View Stereo (MVS) are closely related photogrammetry techniques used to reconstruct 3D models from 2D images. These methods are widely applied in fields such as archaeology, architecture, and gaming due to their ability to generate accurate, high-resolution 3D representations of real-world objects or environments using standard cameras.

**Structure-from-Motion (SfM):**

SfM is a computer vision technique that estimates the 3D structure of a scene by analyzing the motion between multiple overlapping images taken from different angles. Unlike traditional photogrammetry, which requires predefined camera positions, SfM can work with unordered image collections, making it highly flexible. The technique involves identifying corresponding points in different images and using these correspondences to reconstruct the camera's movement and the 3D coordinates of the points in the scene.

**SfM works in three main stages:**

1. **Feature Detection**: SfM identifies common features in multiple images using algorithms such as SIFT (Scale-Invariant Feature Transform) or SURF (Speeded-Up Robust Features).
2. **Feature Matching**: Corresponding features across images are matched, allowing the software to estimate the relative positions of the camera for each image.
3. **3D Reconstruction**: Using the relative camera positions, SfM triangulates the 3D coordinates of the points in the scene, producing a sparse point cloud.

**Multi-View Stereo (MVS)**

MVS complements SfM by taking the sparse point cloud generated by SfM and refining it into a dense 3D model. MVS algorithms analyze multiple images from different angles to generate a highly detailed 3D surface, filling in the gaps left by SfM. The process is similar to how human binocular vision works, using multiple views to create depth perception and a detailed understanding of the scene.

**MVS typically involves:**

1. **Depth Map Estimation**: For each image, the algorithm estimates depth information by analyzing disparities between images.
2. **Point Cloud Densification**: MVS refines the sparse point cloud produced by SfM, creating a dense cloud that captures more of the object's geometry.
3. **Mesh Creation**: The final step involves converting the dense point cloud into a polygonal mesh, which represents the 3D surface of the object or scene.

**Applications:**

SfM-MVS photogrammetry has become a powerful tool in various fields:

1. **Archaeology**: In archaeology, SfM-MVS is used to create detailed 3D models of artifacts, excavation sites, and historical structures. This technique allows archaeologists to document, analyze, and preserve cultural heritage digitally. For example, SfM-MVS has been employed to model ancient ruins and create virtual tours of archaeological sites.
2. **Architecture**: SfM-MVS is widely used in architectural surveys and restoration projects. It allows architects and engineers to create precise 3D models of existing structures, which are essential for renovation, documentation, and preservation. These models are often integrated with Building Information Modeling (BIM) systems to ensure accurate restoration or planning.
3. **Gaming and Virtual Reality**: In game development, SfM-MVS is used to generate realistic 3D environments and assets. Game designers use these techniques to scan real-world objects or landscapes and import them into virtual environments. This provides a high level of realism for games and virtual reality experiences, enhancing player immersion.

**Polycam**

Polycam is a popular app for photogrammetry, which allows you to create detailed 3D models from photos. It's widely used for scanning objects, spaces, and even larger environments using mobile devices. Here's a brief overview of how Polycam works, the photogrammetry process, and how to import a 3D colored model into Blender.

**How Polycam Works:**

Polycam uses your device’s camera to capture a series of photos from different angles. These images are then processed through photogrammetry algorithms, which analyze overlapping points in the photos to reconstruct a detailed 3D model of the object or scene.

**Photogrammetry Process in Polycam:**

1. **Install Polycam**: Download Polycam from the App Store (iOS) or Play Store (Android) or use web.
2. **Create a New Capture**: Open the app and select the "Capture" option.
3. **Take Photos**:
4. Walk around the object or scene you want to scan, taking overlapping photos from different angles.
   1. Try to get all sides and details of the object, ensuring good lighting and enough coverage for accurate model generation.
5. **Process Photos**:
   1. Once you’ve taken enough photos (usually 20-200 depending on

complexity), upload them to Polycam’s server for processing.

* 1. Polycam processes the images into a 3D model using photogrammetry algorithms. This step may take a few minutes depending on the number of photos.

1. **Refining the Model**:

1. After processing, you can review the 3D model within the app, adjust

boundaries, and clean up any unnecessary parts.

1. You can also apply textures and color data from the photos to your 3D model.

**Exporting to Blender:**

1. **Export the Model**:

* Once your model is ready, click on the “Export” button in Polycam and choose a file format that Blender supports, such as .obj, .fbx, or .glb.

1. **Download the Model**:

* Download the exported file from Polycam. It should include the 3D mesh and associated textures (color).

1. **Import into Blender**:

* Open Blender and go to File > Import.
* Select the correct file format (e.g., .obj, .fbx, .glb) and locate your exported file.
* Blender will import both the 3D model and the textures. Make sure to check the material properties to ensure the textures are correctly applied.

**Common Problems and Solutions:**

1. **Low-Quality Model or Texture**:

* **Problem**: The 3D model might have low resolution or poorly aligned textures.
* **Solution**: Ensure you take enough high-quality photos from all angles. Avoid shadows, reflections, and inconsistent lighting. Use Polycam’s

settings to choose higher quality export options if necessary.

1. **Missing or Incorrect Textures in Blender**:

* **Problem**: After importing to Blender, textures may appear missing or incorrectly applied.
* **Solution**: Check that the texture files (usually .mtl or texture maps like .jpg or .png) are in the same folder as the 3D model. In Blender, make sure the

texture paths are correct. You may need to manually apply the textures through the “Materials” tab.

1. **Alignment Issues**:

* **Problem**: The 3D model might appear misaligned or deformed in Blender.
* **Solution**: Ensure that the model and textures are correctly set to the same scale. You can use Blender's tools to adjust the orientation or scale of the model if necessary.

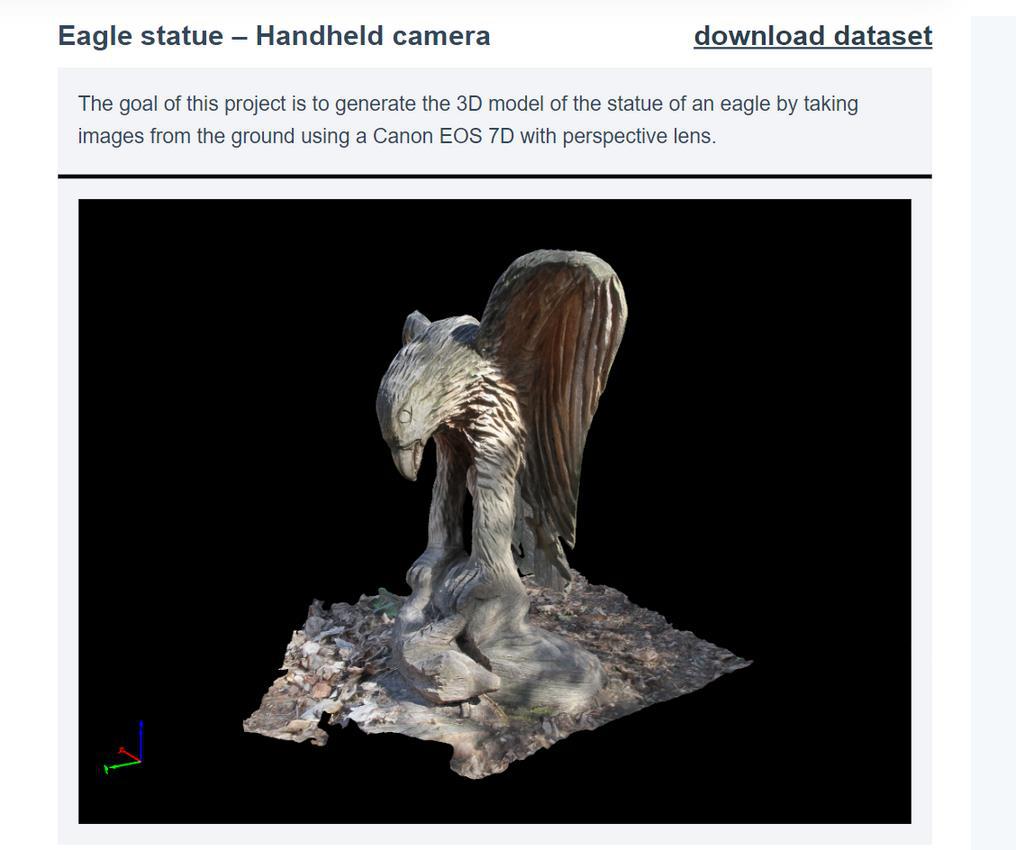
1. **Heavy File Size**:

* **Problem**: Photogrammetry models can result in large file sizes, which might slow down Blender or make it hard to work with.
* **Solution**: Use Blender’s decimate modifier to reduce the polygon count without losing too much detail. You can also simplify the model within

1. **Model Artifacts**:

* **Problem**: Extra or unwanted geometry (artifacts) might be present in the model.
* **Solution**: Use the Polycam app's cleanup tools to remove artifacts before exporting. You can also manually clean up the mesh in Blender by selecting and deleting unwanted geometry.

**Dataset Selected:**



**Link:** [**https://data.pix4d.com/misc/example\_datasets/example\_eagle.zip**](https://data.pix4d.com/misc/example_datasets/example_eagle.zip)

**Output:**

**Dataset to Polycam:**





**Polycam to blender Import:**





**Colmap:**

COLMAP is an open-source photogrammetry software widely used for creating 3D models from images. It is a more advanced and customizable tool compared to mobile apps like Polycam and is often used in academic research and professional projects. Here’s a breakdown of how COLMAP works, the process for using it, and how to import a 3D colored model into Blender, along with common problems and solutions.

**How COLMAP Works:**

COLMAP uses Structure-from-Motion (SfM) and Multi-View Stereo (MVS) techniques to reconstruct 3D models from a series of 2D images. SfM identifies key points and matches them across multiple photos, while MVS refines the surface details and generates dense 3D meshes.

**Photogrammetry Process in COLMAP:**

1. **Install COLMAP**: Download and install COLMAP from its official website (available for Windows, Mac, and Linux).
2. **Capture Photos**:

* Use a camera to take high-quality photos from multiple angles around the object or scene you want to model. Ensure significant overlap between the images (at least 60-70%) and consistent lighting.
* Avoid reflections, shadows, and motion blur.

1. **Launch COLMAP**:

* Open COLMAP and select the **Reconstruction** workspace. You’ll primarily

work in two stages: Feature Extraction and Dense Reconstruction.

1. **Step 1: Feature Extraction and Matching**:

* Import your photos into COLMAP by selecting the folder containing them.
* Go to the **Feature Extraction** tab. COLMAP will detect key points in each photo (e.g., corners, edges) and store this data for further processing.
* After extraction, switch to **Feature Matching**, where COLMAP finds

correspondences between these key points across different images. This step is crucial for building the camera poses and structure of the scene.

1. **Step 2: Sparse Reconstruction (Structure-from-Motion)**:

* Once features are matched, switch to the **Structure-from-Motion** tab and run the **Sparse Reconstruction** process. This will generate a sparse point cloud of the scene and calculate the camera positions relative to each

image.

1. **Step 3: Dense Reconstruction**:

* After sparse reconstruction, go to the **Dense Reconstruction** tab. Here, COLMAP performs Multi-View Stereo to create a dense point cloud by interpolating the surface between matched points.
* You can refine the point cloud by adjusting parameters like depth maps

and fusion settings.

1. **Step 4: Mesh Creation**:

* Once the dense point cloud is ready, you can create a 3D mesh by exporting the point cloud and using external tools like **Poisson Reconstruction** in COLMAP or meshing software like MeshLab or Cloud

Compare.

1. **Step 5: Texturing**:

* COLMAP can generate texture maps for the 3D model by projecting the original images onto the mesh. This step is performed after meshing, where you’ll extract texture information from the photos and map it onto the model’s surface.

**Prepare the Model in CloudCompare:**

Before exporting the model from Cloud Compare, ensure that it includes color or texture information.

* **Point Cloud**: If you’re working with a point cloud, make sure it has RGB color values assigned to each point. You can view this in the **"Scalar** **Fields"** dropdown or by enabling color display in Cloud Compare.
* **Mesh**: If your model is already a mesh (created via the Poisson surface reconstruction method or other meshing techniques in Cloud Compare), ensure that textures are correctly applied to the surface.

Steps in Cloud Compare:

1. **Open the Model**: Load your point cloud or mesh into Cloud Compare.
2. **Verify Colors/Textures**:
3. If it's a point cloud, confirm that each point has RGB color values. Use

the **"Edit > Scalar Fields > Convert to RGB"** option if needed.

* 1. If you’re working with a textured mesh, ensure the texture is visible.

1. **Generate a Mesh (if needed)**:
   1. For point clouds, generate a mesh by selecting **"Tools > Mesh > Delaunay (2D/3D)"** or the **Poisson Surface Reconstruction** under **"Tools > Mesh"**.

1. **Apply Texture to the Mesh**:
   1. If your point cloud doesn’t have textures, you can project color information onto the mesh during meshing (if available). However, for precise textures like photos, it’s better to use external software like COLMAP or Blender for detailed texturing.
2. **Save/Export**the Model:
   1. For point clouds with RGB color: Export as .ply (Polygon File Format),

which supports point colors.

1. For textured meshes: Export as .obj (Wavefront OBJ) or .ply. The .obj format is preferable for meshes with textures, as it generates associated .mtl (material) and texture image files (like .jpg or .png)

**Importing into Blender:**

**Now that your model is ready, the next step is to import it into Blender with colors or textures.**

**Steps in Blender:**

1. **Open Blender and start a new project or load an existing one.**
2. **Import the Model:**
3. Go to the File > Import menu and select the appropriate format based on what you exported from Cloud Compare.
   * For point clouds or simple colored meshes, select PLY (.ply).
   * For textured meshes, select Wavefront (.obj).
   1. Navigate to the folder where your model is saved and select the file.
   2. If you’re importing a .obj file with textures, make sure the .mtl and texture image files (e.g., .jpg or .png) are in the same folder as the .obj file.
4. **Check Colors/Textures:**
   1. After importing, check if the colors or textures are applied correctly to the model.

* 1. For point clouds imported from a .plyfile, Blender will display the colors based on vertex colors. To see these colors:
     + Switch to Material Preview or Rendered view in the viewport.
     + In the Material Properties tab, ensure that the "Viewport Display" is set to use vertex colors.

1. **Apply Textures to Meshes:**
   1. If you’re working with a textured mesh (from an .obj file), Blender should automatically assign the textures through the .mtlfile. However, if the textures don’t show up:
      * Go to the Material Properties tab in Blender.
      * Under the "Base Color" setting, make sure that the texture map is correctly linked. If not, click the folder icon next to it and manually select the texture image file from your folder.
      * Ensure UV mapping is correct: Go to the UV Editor to verify that the texture coordinates are properly applied to the mesh.
2. **Verify Material Settings:**
   1. If you imported a mesh with textures and the materials appear incorrect (e.g., too shiny or dull), you can adjust the material settings:
      * In the Material Properties tab, adjust the roughness, metallic, and other shader properties to fine-tune how the textures are displayed.
3. **Fine-Tune the Model:**
   1. For point clouds, you may need to convert the point cloud to a mesh inside Blender if needed (using Blender's modifiers), or leave it as is for

visualization purposes.

1. For textured meshes, adjust any material, lighting, or texture properties as needed

**Common Problems and Solutions:**

1. **Textures Not Appearing**:

* **Problem**: The model imports, but the textures don’t show up.
* **Solution**: Ensure the texture image files (.jpg/.png) and the .mtl file are in the same folder as the .obj file. Manually apply textures through the

1. **Vertex Colors Not Showing** (Point Cloud):

* **Problem**: After importing a point cloud, vertex colors are not visible.
* **Solution**: Switch to the **Material Preview** or **Rendered** view, then check the

**Material Properties** tab to ensure vertex colors are being used.

1. **Misaligned Textures**:

* **Problem**: Textures appear stretched or misaligned.
* **Solution**: Go to **UV Editing** in Blender and check the UV map coordinates. You might need to manually adjust the UV map to align the texture

properly.

1. **Large File Sizes**:

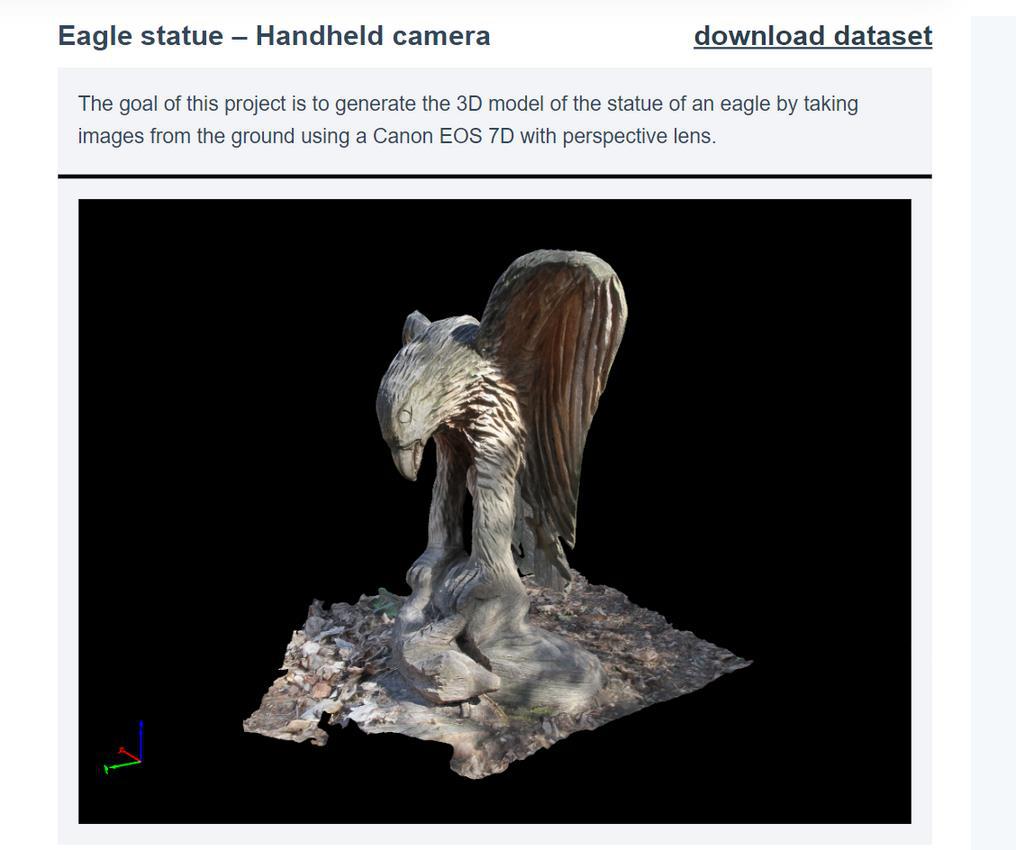
* **Problem**: The model is too large, causing slow performance in Blender.
* **Solution**: In Blender, use the **Decimate Modifier** to reduce the polygon count or clean up the model in CloudCompare before exporting it.

1. **Mesh Doesn’t Appear**:

* **Problem**: After importing, the model might not appear in the Blender viewport.
* **Solution**: Check if the model is extremely small or large by using **View >**

**Frame Selected** (hotkey: Numpad .) to zoom in. You may need to scale the model appropriately in Blender.

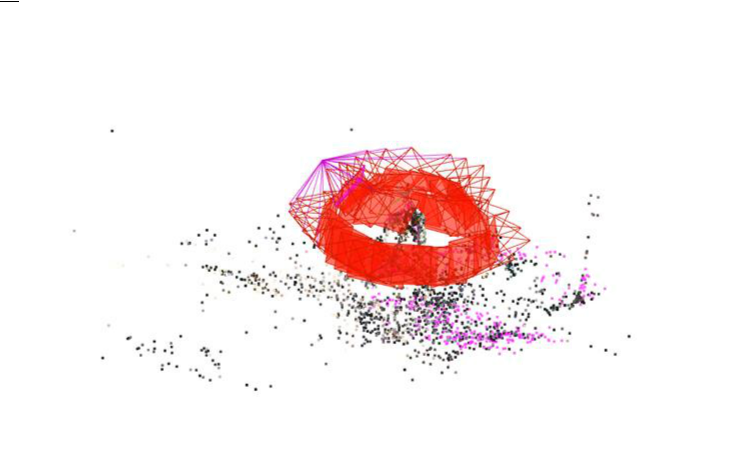
**Dataset Selection:**



**Link:** [**https://data.pix4d.com/misc/example\_datasets/example\_eagle.zip**](https://data.pix4d.com/misc/example_datasets/example_eagle.zip)

**Output:**

 **Dataset to Colmap(Mesh Creation):**



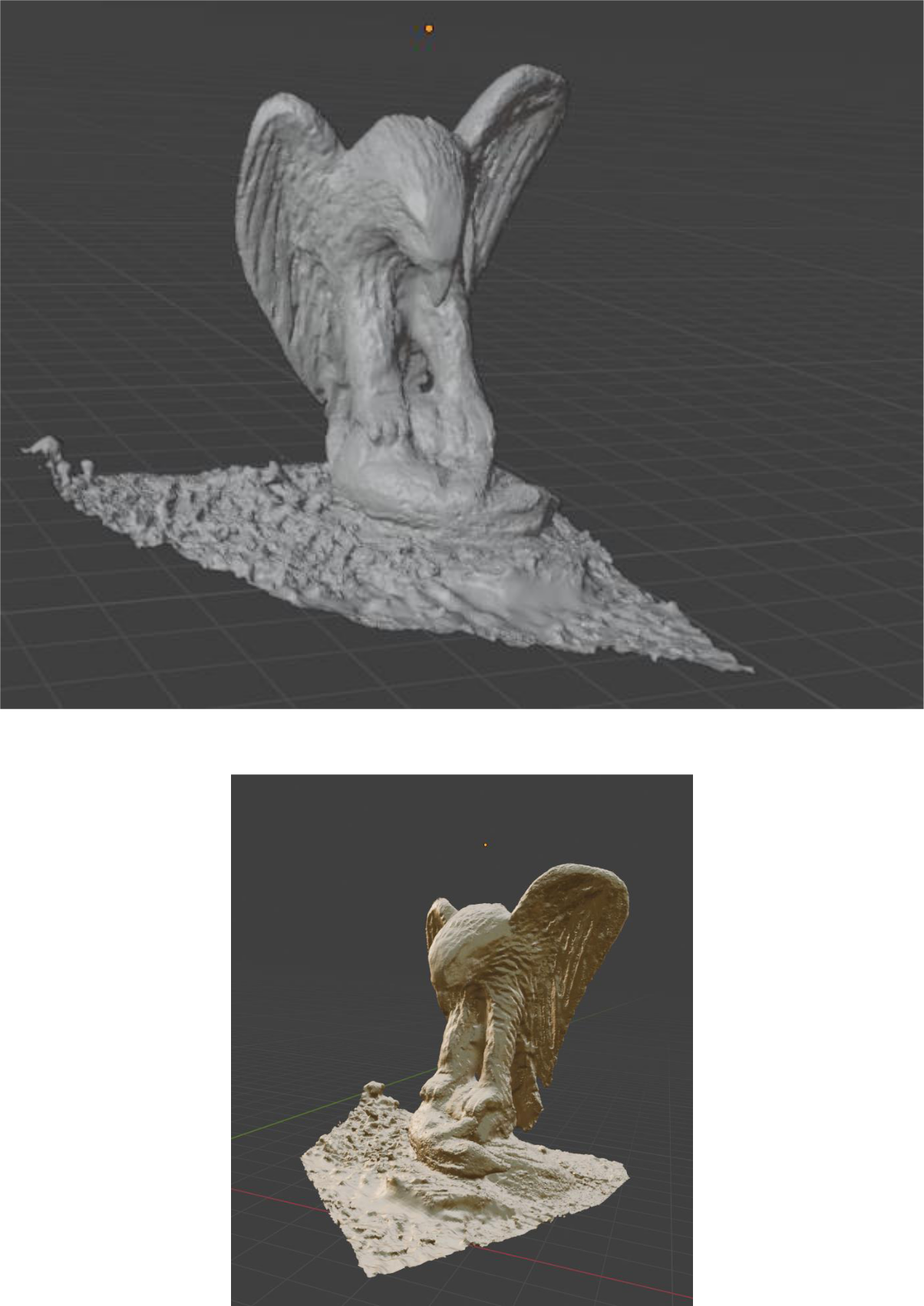
 **Colmap to Cloud Compare(Model Creation):**

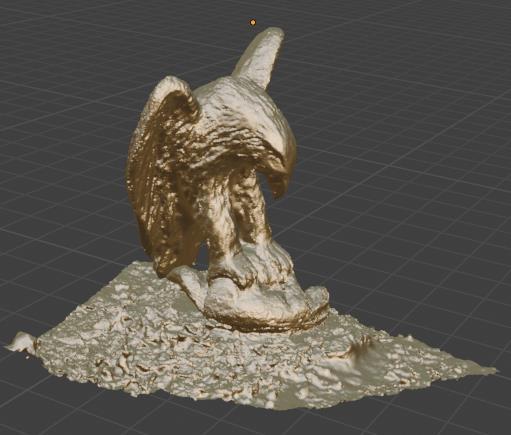




 **Cloud Compare to blender import:**







**Reference:**

<https://youtu.be/9RDT4tkj4h0?si=dKiRjj7cteGGUyZV>

<https://youtu.be/A7l2oAyLB-0?si=igYGT4yC_YBl_pGr>