Reconstructing Incomplete 3D Models Using Singular Value Decomposition for Architectural Design with Blender3D

Barru Adi Utomo - 13523101[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523101@std.stei.itb.ac.id, barru.adi@gmail.com

***Abstract*— 3D reconstruction involves determining the three-dimensional shape or surface structure of an object, a process crucial in fields such as architectural design. However, mistakes or poorly captured point clouds during data collecting frequently result in missing data for 3D models, which can impair their correctness in processes for design and analysis. In this paper, application of Singular Value Decomposition (SVD) is used to rebuild partial 3D models to overcome this difficulty. The goal is to restore the integrity of 3D point clouds and make them usable in architectural design processes by using SVD. Root Mean Square Error (RMSE) calculations are used to assess the reconstruction process and Blender3D are used to visualize it. The experiment shows the effectiveness of SVD in successfully reconstructing missing data, highlighting its potential to enhance data-driven planning and design processes in architecture.**

***Keywords*— 3D Reconstruction, Blender3D, Point Clouds, Singular Value Decomposition**

I. Introduction

In the era of advanced computational tools and data-driven methodologies, the reconstruction of three-dimensional (3D) models has become a cornerstone for numerous applications, especially in architectural design [1][2]. Accurately capturing and representing the physical world in digital formats facilitates efficient planning, analysis, and visualization. However, the process of 3D reconstruction often encounters challenges such as missing or incomplete data, primarily due to errors or limitations during the data collection phase [3]. These gaps in 3D point clouds can significantly impact the precision and reliability of models, ultimately hindering their utility in critical design and analysis processes.

Addressing this issue requires robust and efficient techniques to restore the integrity of incomplete 3D models. Singular Value Decomposition (SVD) has emerged as a promising mathematical tool for tackling such problems, offering a systematic approach to reconstructing missing data with high accuracy [4][5]. Utilizing SVD enables the reconstruction of the basic structures of 3D point clouds, thereby improving their applicability in architectural processes [6][7].

This paper explores the application of SVD to rebuild partial 3D models, focusing on its effectiveness in restoring missing data for improved architectural design processes. Root Mean Square Error (RMSE) is employed as a quantitative metric to evaluate the accuracy of the reconstruction, while the visualization of the reconstructed models is conducted using Blender3D. The findings highlight the potential of SVD in mitigating data loss and advancing the reliability of 3D reconstructions in architectural contexts, paving the way for more robust and data-driven design methodologies.

II. Related Works

A. Singular Value Decomposition

The reconstruction process utilizes Singular Value Decomposition (SVD), a mathematical technique that decomposes a data matrix into three components, with Eq.1 [].

$$A = U \cdot S \cdot V^T \tag{1}$$

In this decomposition, U is an orthogonal matrix containing the left singular vectors, S is a diagonal matrix containing singular values that represent the strength of each component, and V^T is another orthogonal matrix whose rows are the right singular vectors. SVD is widely applied in data approximation by finding the best low-rank representation of a matrix, capturing its dominant patterns.

B. Point Clouds

A point cloud is a set of 3D points in Euclidean space. Each point has x-, y-, and z-coordinates in the Cartesian system on the sampled surfaces of 3D objects [8]. The geometry of entire 3D objects can be fully captured by merging a large number of these distinct spatial points into a single data set using a shared system frame. Point clouds have been described as the best data for 3D visualization in broader metropolitan contexts for various applications since they can offer the spatial coordinates of observed object surfaces [9].

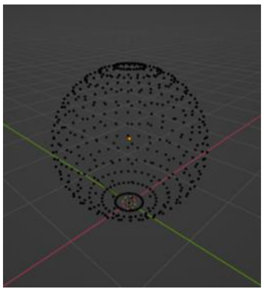

Fig 1. Point Clouds example

C. Root Mean Square Error (RMSE)

RMSE is a standard metric for quantifying the average magnitude of differences between the reconstructed and original data points [10].

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_{orig,i} - P_{recon,i})^2} \quad (2$$

N = total number of data points
$P_{orig,i}$ = original coordinates
$P_{recon,i}$ = reconstructed coordinates

The RMSE metric serves as a crucial measure of the reconstruction's accuracy, with lower RMSE values indicating better alignment between the reconstructed and original datasets. This validation process is carried out by comparing the reconstructed coordinates against the original values, focusing only on the points that were not initially missing. By calculating the squared differences for each coordinate, averaging these values, and taking the square root, the RMSE provides a comprehensive measure of the reconstruction's reliability and accuracy [11].

III. Research Methodology

A. Data Preparation

The first step in this experiment is to prepare the data, which serves as the foundation for the 3D reconstruction process. The data can originate from various sources, including real-world examples of 3D scans captured using devices like LiDAR or depth cameras, 3D models created in software such as Blender, or generic geometric shapes such as sphere. The format of the data needs to be CSV with x, y, z column value.

```
def load_dataset(file_path):
    path = os.path.join(os.getcwd(), "model", file_path)
    return pd.read_csv(path).to_numpy()

def load_multiple_datasets(file_paths):
    datasets = [pd.read_csv(file_path).to_numpy() for file_path in file_paths]
    return datasets
def generate_sphere(radius=5, num_points=1000):
    theta = np.random.uniform(0, np.pi, num_points)  # Latitude
    phi = np.random.uniform(0, 2 * np.pi, num_points)  # Longitude

    x = radius * np.sin(theta) * np.cos(phi)
    y = radius * np.sin(theta) * np.sin(phi)
    z = radius * np.cos(theta)

    return np.vstack([x, y, z]).T
```

Once the dataset is prepared, the next step is to simulate missing data, which emulates real-world issues such as incomplete scans, occlusions, or sensor errors. Missing data can be generated in two primary ways: through random removal of points or by introducing alternative models. Random removal involves eliminating a percentage of points, in this study using 10%, from the original dataset to simulate gaps commonly found in 3D scan data. Alternatively, another model of the same type, such as a different design of a chair or another geometric shape, can be used to simulate diversity in missing patterns while maintaining structural consistency. A lower percentage of missing data imitate minor gaps caused by noise or small occlusions, while a higher percentage represents significant data loss, such as incomplete scans or structural damage. By preparing the dataset and simulating missing data with particular setup, the experiment ensures a realistic and rigorous testing environment for the reconstruction methodology.

```
def introduce_missing_data(points, missing_ratio=0.2):
    mask = np.random.rand(*points.shape) < missing_ratio
    points_with_missing = points.copy()
    points_with_missing[mask] = np.nan
    return points_with_missing
```

B. Data Preprocessing

Preprocessing the data by scaling and centering each data point is crucial to ensuring that the dataset is appropriate for Singular Value Decomposition (SVD). This stage guarantees that the decomposition accurately reflects the underlying patterns in the data and improves numerical stability. Subtracting the mean of each coordinate (x, y, z) from each data point is the conventional method for scaling and centering. Biases or offsets in the coordinates are eliminated by centering the dataset around the origin, which allows SVD to concentrate on the data's inherent structure.

Since incomplete datasets cannot be directly subjected to SVD, handling missing data is an essential part of preprocessing. To overcome this restriction, temporary values must be used in place of missing data prior to decomposition. Two common techniques for handling missing data are to use the mean of the associated coordinate or to replace the missing values with zeros. In this study, mean replacement is chosen because it reduces distortions by maintaining the dataset's general structure and keeps results from being skewed by extreme values. By ensuring that the dataset is complete and centered, this preprocessing method makes it possible for SVD to recover missing data points efficiently and precisely.

```
def replace_missing_with_mean(points):
    col_means = np.nanmean(points, axis=0)
    filled_points = np.where(np.isnan(points),
col_means, points)
    return filled_points
```

C. Reconstruction Using SVD

In this experiment, missing values in the dataset are replaced with the mean and normalized of their respective coordinates during the preprocessing stage. This ensures that the dataset is complete and centered, allowing SVD to operate effectively [12].

```
def normalize_data(points):
    mean = np.nanmean(points, axis=0)
    std = np.nanstd(points, axis=0)
    normalized_points = (points - mean) / (std
+ 1e-8)
    return normalized_points, mean, std
```

The SVD decomposition extracts underlying structures and patterns from the data, which are then used to reconstruct the original matrix using the formula:

$$\hat{A} = U \cdot S \cdot V^T \quad (3$$

$$\hat{A} = \text{reconstructed matri}x$$

The reconstructed matrix, A ,includes estimates for the missing values, derived based on the relationships between data points and the patterns captured by the singular vectors and singular values. This process preserves the overall structure and geometry of the original dataset while filling the gaps caused by missing data.

```
def svd_reconstruction(points):
    U, S, Vt = np.linalg.svd(points,
full_matrices=False)
    reconstructed_points = np.dot(U,
np.dot(np.diag(S), Vt))
    return reconstructed_points
```

After getting the reconstructed points, the points need to be denormalized.

```
def denormalize_data(points, mean, std):
    denormalized_points = (points * std) +
mean
denormalized_points[np.abs(denormalized_points
) < 1e-8] = 0.0
    return denormalized_points
```

D. Validation

This study employs the Root Mean Square Error (RMSE) to evaluate the accuracy of the reconstructed model.

```
def calculate_rmse(original, reconstructed):
    mask = ~np.isnan(original)
    mse = np.mean((original[mask] -
reconstructed[mask]) ** 2)
    return np.sqrt(mse)
```

The validation process is instrumental in assessing the effectiveness of SVD in handling missing data. Additionally, RMSE serves as a benchmark for evaluating different experimental conditions, such as varying the percentage of missing data or testing alternative reconstruction methods. A higher RMSE may indicate limitations in the reconstruction process, such as excessive data loss or noise in the dataset [10]. By incorporating RMSE as a validation metric, this study ensures a rigorous assessment of the proposed methodology, demonstrating its applicability and robustness in 3D reconstruction tasks.

E. Experiment Setup

The experiment investigates factors influencing the accuracy of 3D reconstruction, focusing on the percentage of missing data and dataset diversity. Missing data levels are tested at 10% to evaluate the reconstruction method's ability to estimate missing points. Dataset diversity is ensured by using both real-world 3D scans and synthetic shapes like spheres and cubes. This combination allows the methodology to be tested across different scenarios, ensuring its generalizability and adaptability.

Additionally, the experiment examines the role of initialization methods in preprocessing missing data before applying Singular Value Decomposition (SVD). Missing values are replaced either by the mean of their respective coordinates or by zeros. Mean replacement helps preserve the dataset's structure and minimizes biases, while zero substitution offers a simpler but less representative approach. By analyzing these methods, the experiment identifies best practices for handling missing data, providing insights into improving reconstruction workflows.

IV. EXPERIMENT

A. Data Preparation

This study will test two types of data. The first one is a simple shape, such as sphere and the second one is chair. The Figure 2 shows the complete sphere points and the Figure 3 represents the missing sphere points from randomly selected points.

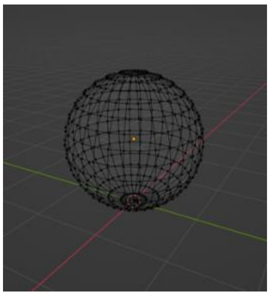

Fig 2. Complete sphere points

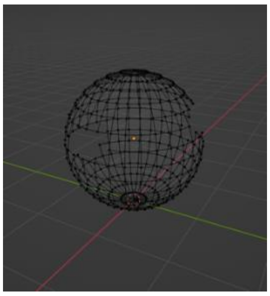

Fig 3. Missing sphere points

The same applies to the chair object. The Figure 4 shows the complete chair points and the Figure 5 shows the missing chair points from randomly selected points.

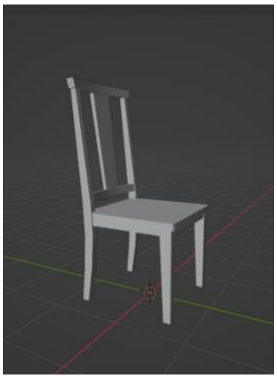

Fig 4. Complete chair points

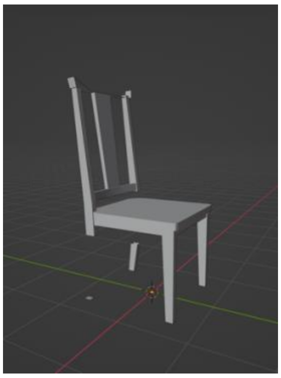

Fig 5. Missing chair points

B. Data Preprocessing

In the data preprocessing stage, an initial guess is made by replacing NaN values with the mean of all vertices as explained in Chapter III.

The orange dots in Figure 6 shows the initial guess of the missing points in sphere object compared with the missing sphere points in Figure 3.

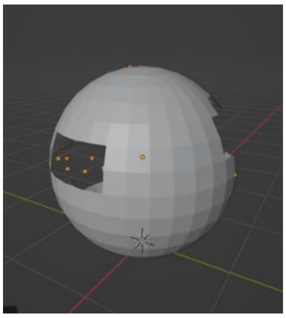

Fig 6. Temporary value sphere

The same implies in chair, the orange dots in Figure 7 shows the initial guess of the missing points.

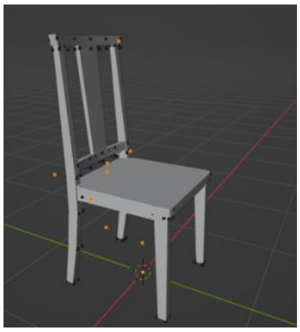

Fig 7. Temporary value sphere

C. Reconstruction Using SVD

After using SVD from the initial guess, explained in Chapter III, the orange dots in Figure 8 and 9 shows the results being compared with the missing points.

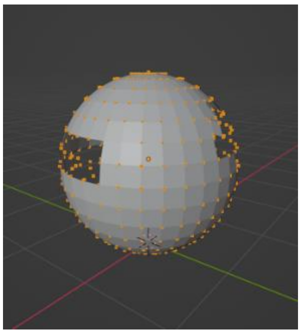

Fig 8. Reconstructed sphere

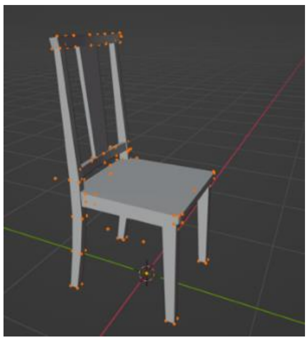

Fig 9. Reconstructed chair

D. Validation

After getting the result points from reconstruction using SVD, RMSE is being used as explained in Chapter II.

```
[3] def calculate_rmse(original, reconstructed):
        mask = ~np.isnan(original)
        mse = np.mean((original[mask] - reconstructed[mask]) ** 2)
        return np.sqrt(mse)

RMSE: 0.11372785585260507
```

Figure 10. RMSE from sphere reconstruction

Figure 10 shows the RMSE from sphere reconstruction is 0.113.

```
[1] def calculate_rmse(original, reconstructed):
        mask = ~np.isnan(original)
        mse = np.mean((original[mask] - reconstructed[mask]) ** 2)
        return np.sqrt(mse)

RMSE: 0.039146735407557864
```

Figure 11. RMSE from chair reconstruction

Figure 11 shows the RMSE value from chair reconstruction. The obtained value is 0.039, which is close to zero, and as explained in Chapter II, the reconstructed points and the actual points are fairly close.

V. CONCLUSION

The experiment shows that reconstruction accuracy depends on dataset complexity. The SVD-based reconstruction method performs well for datasets with predictable and simpler geometries but faces challenges with datasets involving continuous and intricate surfaces. For the sphere dataset, an RMSE of 0.113 was observed for 10% missing data, highlighting the challenges of reconstructing smooth, continuous surfaces sensitive to small deviations. In contrast, the chair dataset achieved a lower RMSE of 0.039, reflecting the simpler geometric structures of flat surfaces and edges, which are easier to reconstruct.

These results emphasize the importance of considering dataset geometry when applying SVD for 3D reconstruction. For complex datasets with curved surfaces, advanced techniques like weighted local mean or parametric surface fitting may improve accuracy. The insights from this study can guide future improvements in reconstruction methodologies for diverse 3D datasets. Future work could explore varying missing data levels and combining SVD with other methods, such as machine learning or geometric fitting, to enhance performance across diverse datasets.

VI. APPENDIX

1. Source code: https://github.com/barruadi/reconstructing-incomplete-3d-model

2. Explanation Video: https://youtu.be/DyLr00TQLXk

VII. ACKNOWLEDGMENT

I would like to express my gratitude to the individuals and institution whose support and contributions have been instrumental in the completion of this research:

1. God Almighty, thanks to His grace and guidance, this paper can be completed.
2. Both parents who have been supportive for this research.
3. Dr. Ir. Rinaldi Munir, M.T. as the lecturer of the IF2120 Discrete Mathematics.
4. All of my friends, especially Azfa Radhiyya Hakim and Rafif Farras who also support this research.

REFERENCES

[1] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, “Applications of 3D city models: State of the art review,” ISPRS International Journal of Geo-Information, vol. 4, no. 4, pp. 2842–2889, 2015.

[2] V. Sequeira, K. Ng, E. Wolfart, J. G. M. Gonçalves, and D. Hogg, “Automated reconstruction of 3D models from real environments,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54, no. 1, pp. 1–22, 1999.

[3] A. Farshian, M. Götz, G. Cavallaro, C. Debus, M. Nießner, J. A. Benediktsson, and A. Streit, “Deep-Learning-Based 3-D Surface Reconstruction—A Survey,” Proceedings of the IEEE, 2023.

[4] S. Intawichai and S. Chaturantabut, “A missing data reconstruction method using an accelerated least-squares approximation with randomized SVD,” Algorithms, vol. 15, no. 6, pp. 190, 2022.

[5] X. Chai, G. Tang, S. Wang, K. Lin, and R. Peng, “Deep learning for irregularly and regularly missing 3-D data reconstruction,” IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 7, pp. 6244–6265, 2020.

[6] D. Bassu, R. Izmailov, A. McIntosh, L. Ness, and D. Shallcross, “Centralized multi-scale singular value decomposition for feature construction in LIDAR image classification problems,” in Proc. IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2012, pp. 1–6.

[7] D. Galanakis, E. Maravelakis, D. P. Pocobelli, N. Vidakis, M. Petousis, A. Konstantaras, and M. Tsakoumaki, “SVD-based point cloud 3D stone by stone segmentation for cultural heritage structural analysis—The case of the Apollo Temple at Delphi,” Journal of Cultural Heritage, vol. 61, pp. 177–187, 2023.

[8] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, “Learning representations and generative models for 3d point clouds,” in Proc. International Conference on Machine Learning, 2018, pp. 40–49.

[9] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, “Deep learning on 3D point clouds,” Remote Sensing, vol. 12, no. 11, pp. 1729, 2020.

[10] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?--Arguments against avoiding RMSE in the literature,” Geoscientific Model Development, vol. 7, no. 3, pp. 1247–1250, 2014.

[11] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)?--Arguments against avoiding RMSE in the literature,” Geoscientific Model Development, vol. 7, no. 3, pp. 1247–1250, 2014.

[12] K. Baker, “Singular value decomposition tutorial,” The Ohio State University, vol. 24, pp. 22, 2005.

STATEMENT

I declare that this paper is my own writing, not an adaptation, or translation of someone else's paper, and not plagiarized.

Bandung, 31 Desember 2024

Barru Adi Utomo
13523101