International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0362
e ISSN: 2584-2854
Volume: 04 Issue: 07
July 2026
Page No: 2531 - 2535

Interactive Multidimensional Data Visualization Through 2d and 3d Scatter Plots Using D3. Js and Three.Js

Sourabh Yaduvanshi[1], Varsha Namdeo[2]
[1]Research Student, Sarvepalli Radhakrishnan University, Bhopal, India
[2] HOD, Sarvepalli Radhakrishnan University, Bhopal, India
Emails: *1rubh000@gmail.com[1], deanmca@srku.edu.in[2]*

Abstract
Data visualization has become an essential component of modern data analytics, enabling users to identify patterns, trends, correlations, and anomalies within large datasets. Scatter plots are among the most effective visualization techniques for representing relationships between numerical variables. This research presents the design and development of interactive 2D and 3D scatter plot visualizations using D3.js and Three.js technologies. D3.js is employed to construct dynamic two-dimensional scatter plots with interactive features such as zooming, filtering, and tooltips, while Three.js is utilized to render three-dimensional scatter plots using WebGL for enhanced spatial representation. The proposed approach demonstrates how multidimensional datasets can be effectively visualized and explored through modern web-based technologies. Experimental analysis indicates that D3.js offers superior simplicity and responsiveness for two-dimensional data representation, whereas Three.js provides greater depth perception and multidimensional exploration capabilities. The integration of these technologies creates a scalable framework for scientific, business, and analytical applications.

Keywords: *Data Visualization, Scatter Plot, D3.js, Three.js, Web Graphics, Interactive Analytics, 2D Visualization, 3D Visualization.*

1. Introduction
Data visualization transforms raw information into graphical representations that improve comprehension and decision-making. As organizations increasingly rely on data-driven strategies, the need for effective visualization tools has grown significantly. Scatter plots are widely used because they allow analysts to observe relationships between variables, identify clusters, and detect outliers within datasets [1]. Traditional two-dimensional scatter plots represent data using x and y coordinates and are suitable for analyzing relationships between two variables. However, complex datasets often contain multiple dimensions that cannot be fully represented in a two-dimensional environment. Three-dimensional scatter plots address this limitation by introducing a z-axis, enabling users to visualize an additional variable while preserving spatial relationships among data points [2]. D3.js (Data-Driven Documents) is a JavaScript library designed for creating highly customizable and interactive visualizations using SVG, HTML, and CSS technologies [3]. Three.js is a WebGL-based JavaScript framework that simplifies the development of three-dimensional graphics and interactive visualizations in web applications [4]. This study investigates the implementation of 2D and 3D scatter plot visualizations using these technologies and evaluates their effectiveness for exploratory data analysis.

1.1.Scatter Plot Visualization Principles
Each observation in a dataset is represented as a point positioned according to its corresponding variable values. The location of the point provides insight into relationships and distributions within the dataset [5]. Scatter plots assist in identifying

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0362
e ISSN: 2584-2854
Volume: 04 Issue: 07
July 2026
Page No: 2531 - 2535

positive, negative, and nonlinear relationships between variables. Trends become visually apparent through the distribution of plotted points [6]. Data points significantly distant from clusters may indicate anomalies, errors, or exceptional cases requiring further investigation [7]. Three-dimensional scatter plots introduce depth perception through the z-axis, enabling analysts to visualize additional variables and discover hidden patterns within multidimensional datasets [8].

2. Methodology

The proposed methodology consists of four major stages:

- Dataset Collection and Preprocessing
- Development of 2D Scatter Plot using D3.js
- Development of 3D Scatter Plot using Three.js
- Performance Evaluation and Comparative Analysis

2.1.Dataset Collection and Preprocessing

Numerical datasets are collected from publicly available repositories and converted into JSON format. Missing values, duplicate records, and inconsistencies are removed to improve visualization accuracy. Data normalization techniques are applied to ensure proper scaling across visualization dimensions [9].

2.2.Development of 2D Scatter Plot using D3.js

The D3.js framework is used to generate scalable vector graphics representing data points in a two-dimensional coordinate system. Linear scales map numerical values to screen coordinates. Interactive features including tooltips, zoom controls, brushing, and filtering mechanisms are incorporated to improve user engagement and analytical capability [3]. Source Code 2D Scatter Plot Visualization using D3.js.as shown in Figure 1 2d Scatter Plot (D3.Js), Figure 2 3D Scatter Plot (Three.js). Table 1. Software and Hardware configuration.

```
<script src="https://d3js.org/d3.v7.min.js"></script>
<script>
const data = [
  {x:10, y:20},
  {x:20, y:40},
  {x:30, y:25},
  {x:40, y:60},
  {x:50, y:45},
  {x:60, y:80}
];

const width = 800;
const height = 500;
const margin = 50;
const svg = d3.select("svg");

const xScale = d3.scaleLinear()
  .domain([0, 70])
  .range([margin, width - margin]);

const yScale = d3.scaleLinear()
  .domain([0, 100])
  .range([height - margin, margin]);

svg.append("g")
  .attr("transform", `translate(0,${height-margin})`)
  .call(d3.axisBottom(xScale));

svg.append("g")
  .attr("transform", `translate(${margin},0)`)
  .call(d3.axisLeft(yScale));

svg.selectAll("circle")
  .data(data)
  .enter()
  .append("circle")
  .attr("cx", d => xScale(d.x))
  .attr("cy", d => yScale(d.y))
  .attr("r", 6)
  .attr("fill", "steelblue");
</script>
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0362
e ISSN: 2584-2854
Volume: 04 Issue: 07
July 2026
Page No: 2531 - 2535

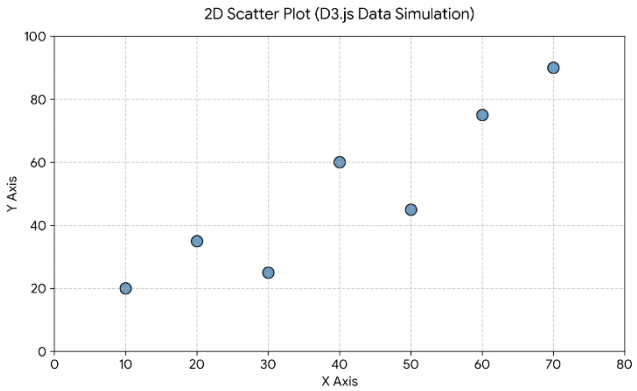

Figure 1 2d Scatter Plot (D3.Js)

2.3.Development of 3D Scatter Plot using Three.js

Three.js is utilized to render data points within a three-dimensional environment. Each observation is mapped to x, y, and z coordinates and represented using sphere geometries. Camera controls, lighting effects, and orbit navigation are implemented to provide an immersive visualization experience [4].

Source code 3D Scatter Plot using Three.js

```
<script type="importmap">
  {
   "imports": {

"three":"https://unpkg.com/three@0.165.0/build/three.module.js",

"three/addons/":"https://unpkg.com/three@0.165.0/examples/jsm/"
    }
  }
  </script>

<script type="module">

import * as THREE from 'three';
import { OrbitControls }
from 'three/addons/controls/OrbitControls.js';

const scene = new THREE.Scene();
scene.background = new THREE.Color(0xf5f5f5);

const camera = new THREE.PerspectiveCamera(
75,
window.innerWidth/window.innerHeight,
0.1,
1000
);

camera.position.set(20,20,20);

const renderer = new THREE.WebGLRenderer({
antialias:true
});

renderer.setSize(
window.innerWidth,
window.innerHeight
);

document.body.appendChild(renderer.domElement);

const controls = new OrbitControls(
camera,
renderer.domElement
);

const points = [
  [2,4,5],
  [4,8,7],
  [6,3,10],
  [8,12,6],
  [10,15,12],
  [12,7,15]
];

const geometry =
new THREE.SphereGeometry(0.3,16,16);

points.forEach(p => {

  const material =
  new THREE.MeshBasicMaterial({
    color:0x0077ff
  });
```

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0362
e ISSN: 2584-2854
Volume: 04 Issue: 07
July 2026
Page No: 2531 - 2535

```
    const sphere =
    new THREE.Mesh(geometry,material);

    sphere.position.set(
        p[0],
        p[1],
        p[2]
    );

    scene.add(sphere);
});

const axesHelper =
new THREE.AxesHelper(15);

scene.add(axesHelper);

function animate(){
    requestAnimationFrame(animate);
    controls.update();
    renderer.render(scene,camera);
}

animate();

</script>
```

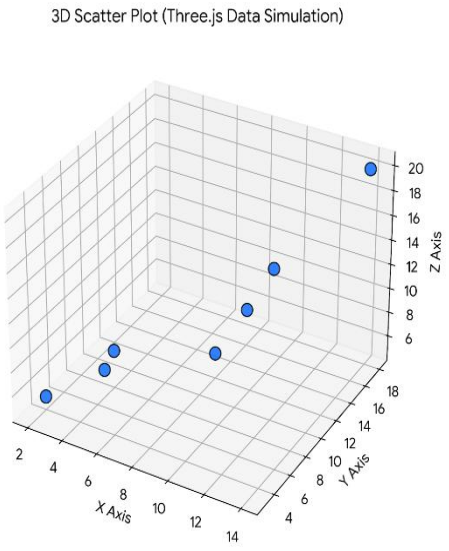

Figure 2 3D Scatter Plot (Three.js)

Table 1 Software and Hardware configuration

Component	Specification
Processor	Intel Core i5
RAM	16 GB
Operating System	Windows 11
Visualization Library	D3.js
Graphics Framework	Three.js
Browser	Google Chrome

The D3.js implementation provided smooth rendering and rapid interaction for small and medium-sized datasets. Interactive tooltips and zooming functionalities enhanced user experience. The Three.js implementation enabled multidimensional exploration and spatial analysis of datasets that could not be effectively represented in two dimensions.

3. Results And Discussion

The results demonstrate that D3.js and Three.js serve complementary purposes in modern visualization systems. D3.js excels in producing lightweight and highly customizable visualizations suitable for dashboards, reporting systems, and business intelligence applications [10][11]. Three.js provides enhanced visual depth and supports exploration of multidimensional datasets. The additional spatial dimension allows analysts to identify clusters and relationships that may not be visible within traditional 2D visualizations [12][13]. However, increased computational requirements and visual complexity may affect usability when datasets become extremely dense.

The comparison indicates that D3.js is more appropriate for routine analytical tasks involving two variables, while Three.js is advantageous for scientific visualization, machine learning analytics, and multidimensional exploratory analysis [14][15].

Conclusion

This research presented the design and implementation of interactive 2D and 3D scatter plot visualizations using D3.js and Three.js technologies. The study demonstrated that D3.js

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0362
e ISSN: 2584-2854
Volume: 04 Issue: 07
July 2026
Page No: 2531 - 2535

provides an efficient and flexible framework for creating responsive two-dimensional scatter plots, whereas Three.js enables immersive three-dimensional exploration of complex datasets. Experimental evaluation revealed that each technology offers distinct advantages depending on the visualization objective and dataset characteristics. Future work may focus on integrating real-time streaming data, virtual reality interfaces, and machine learning techniques to further enhance analytical capabilities and user experience.

Acknowledgements

The authors express their gratitude to the open-source development communities of D3.js and Three.js for providing robust visualization frameworks that facilitated the implementation of this research. The authors also acknowledge the availability of publicly accessible datasets used for experimental evaluation.

References

[1]. Few, S. (2013). *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring* (2nd ed.). Analytics Press.
[2]. Ware, C. (2021). *Information Visualization: Perception for Design* (4th ed.). Morgan Kaufmann.
[3]. Bostock, M., Ogievetsky, V., & Heer, J. (2011). D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2301–2309. https://doi.org/10.1109/TVCG.2011.185
[4]. Cabello, R. (2025). *Three.js Documentation and WebGL Framework*. Three.js Foundation. Available at: https://threejs.org
[5]. Munzner, T. (2014). *Visualization Analysis and Design*. CRC Press.
[6]. Wilkinson, L. (2005). *The Grammar of Graphics* (2nd ed.). Springer.
[7]. Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
[8]. Keim, D. A., Kohlhammer, J., Ellis, G., & Mansmann, F. (2010). *Mastering the Information Age: Solving Problems with Visual Analytics*. Eurographics Association.
[9]. Han, J., Kamber, M., & Pei, J. (2021). *Data Mining: Concepts and Techniques* (4th ed.). Morgan Kaufmann.
[10]. Heer, J., & Shneiderman, B. (2012). Interactive Dynamics for Visual Analysis. *Communications of the ACM*, 55(4), 45–54.
[11]. Ward, M. O., Grinstein, G., & Keim, D. (2015). *Interactive Data Visualization: Foundations, Techniques, and Applications* (2nd ed.). CRC Press.
[12]. Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.
[13]. R. P. Duquia, J. L. Bastos, R. R. Bonamigo, D. A. González-Chica, and J. Martínez-Mesa, “Presenting data in tables and charts,” An. Bras. Dermatol., vol. 89, no. 2, pp. 280–285, 2014, doi: 10.1590/abd1806-4841.20143388.
[14]. L. Chapman and E. Roberts, “(Point of contact).,” Infirm. Can., vol. 20, no. 10, pp. 26– 8, 1978 “Qualitative Variable"
[15]. A. Blackler and V. Popovic, "Towards Intuitive Interaction Theory, "Interacting with Computers, vol. 27,(3), pp. 203-209, 2015.