

Geometric and algebraic machine learning methods in computer graphics and vision systems

Meena Rani [†]

*Department of Computer Science & Engineering
Chitkara University Institute of Engineering and Technology
Chitkara University
Rajpura 140401
Punjab
India*

Tarandeep Kaur ^{*}

*School of Computer Applications
Lovely Professional University
Phagwara 144411
Punjab
India*

Paramita Sarkar [§]

*Department of Computer Science & Engineering
JIS University
Kolkata 700109
West Bengal
India*

M.V.V. Prasad Kantipudi [‡]

*Department of Electronics and Telecommunication
Symbiosis Institute of Technology
Pune Campus
Symbiosis International (Deemed University)
Pune 412115
Maharashtra
India*

[†] E-mail: meena.rani@chitkara.edu.in

^{*} E-mail: tarandeep.24836@lpu.co.in (Corresponding Author)

[§] E-mail: paramita.sarkar@jisuniversity.ac.in

[‡] E-mail: mvvprasad.kantipudi@gmail.com

Priyanka Gupta [@]

*Department of Computer Science and Engineering
Chitkara University
Rajpura 140401
Punjab
India*

R. S. M. Lakshmi Patibandla [#]

*Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Guntur 522001
Andhra Pradesh
India*

Abstract

Computer vision and image processing are the subjects of this study, which investigates geometric and algebraic machine learning. Our ability to extract features and add data is facilitated by the use of vector measure spaces, manifolds, and algebraic structures such as groups and tensors. For the purpose of stabilizing and agreeing models, our technique makes use of orthogonal sets and sequences. Improvements in representations derived from high-dimensional visual input can be achieved by the application of techniques that combine geometric sensibility and mathematical rigour. The results of trials indicate that these tactics are effective in improving computer vision tasks such as object recognition and three-dimensional form analysis.

Subject Classification: 31A10, 33E30.

Keywords: *Geometric machine learning, Algebraic structures, Feature extraction, Orthogonal sequences, Data embedding.*

I. Introduction

Computer vision and visuals have altered because machine learning lets computers understand and change visual input. Old machine learning methods may not understand visual data's complex geometry due to Euclidean assumptions [1]. Geometry and algebra models can improve learning stability, comprehension, and efficacy [2]. Recent studies demonstrate its importance. Geometric models can explain for many data's shape and irregular relationships. These approaches leverage manifolds and metric spaces. This helps with jobs like recognizing objects, segmenting images, and building 3D models and makes it easier to draw broad conclusions. Figure 1 shows ML methods for feature extraction that use both math and geometry. Structures in mathematics like

[@] E-mail: gupta.priyanka@chitkara.edu.in

[#] E-mail: patibandla.lakshmi@gmail.com

rings, groups, and tensors also make it easy to find patterns, invariances, and interactions between different types of data [3, 4].

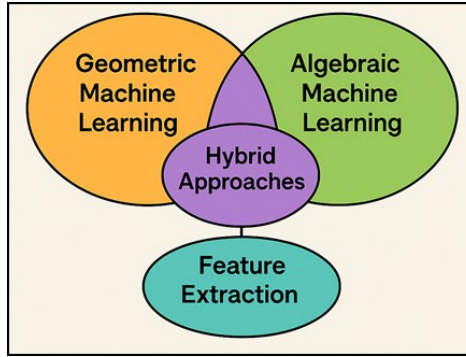


Figure 1
Integration of Geometric and Algebraic Machine Learning Approaches for Feature Extraction in Computer Graphics and Vision

Features may be retrieved and incorporated in these structures without changing throughout transformations. This provides additional stability and is easier to handle [5]. Combining precise arithmetic with geometric intuition improves machine learning approaches for computer pictures and vision systems [6]. Real-world applications of geometry and algebraic machine learning approaches are the focus of this research. These arithmetic tools may be used alone or together to speed up learning, identify features, and merge data [7,8].

II. Theoretical Foundations

A. Mathematical geometry in ML

Manifolds are a kind of smooth, curved space that can be used to model difficult data trends in machine learning. Plus, they help you understand geometry [9]. To understand measure spaces, which show how similar two things are, you need to be able to sort and group things. When computers work with data, these shapes all work together to help them figure out how the data is organised [10, 11]. This makes them more general and reliable by recording complex relationships in feature spaces with a lot of dimensions.

1. Definition of a Metric Space:

$$d : X \times X \rightarrow \mathbb{R}, d(x, y) \geq 0 \quad (1)$$

2. Metric Properties:

$$(i) d(x, y) = 0 \Leftrightarrow x = y \quad (2)$$

$$(ii) d(x, y) = d(y, x) \quad (3)$$

$$(iii) d(x, z) \leq d(x, y) + d(y, z), \forall x, y, z \in X \quad (4)$$

3. Open Ball:

$$B_r(p) = \{x \in X \mid d(p, x) < r\}, r > 0 \quad (5)$$

4. Topological Space from Metric:

The collection of all unions of open balls forms a topology τ on X .

5. Definition of a Manifold:

A topological space M is a d -dimensional manifold if $\forall p \in M$,
 $\exists$ neighborhood U and homeomorphism:

$$\varphi : U \rightarrow \mathbb{R}^d \quad (6)$$

6. Charts and Atlases:

A set of charts $\{(U_\alpha, \varphi_\alpha)\}$ covering M defines an atlas.

7. Riemannian Metric:

A smooth assignment of inner product on tangent spaces:

$$g_p : T_p M \times T_p M \rightarrow \mathbb{R} \quad (7)$$

8. Distance on Manifold:

Geodesic distance d_g between $p, q \in M$ is defined as:

$$d_{g(p,q)} = \inf_{\gamma} \int_0^1 \text{sqrt} \left(g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t)) \right) dt \quad (8)$$

9. Embedding of Data on Manifolds:

$$\text{Data } X = \{x_i\} \text{ mapped to } M \text{ by an embedding:} \quad (9)$$

$\iota : X \rightarrow M$, smooth and injective

B. Algebraic structures relevant to ML: groups, rings, tensors

Algebraic structures, like rings and groups, make actions on data more formal. They support symmetry and invariance, which are very important in machine learning [12]. Tensors take vectors and matrices to higher dimensions, which makes it easier to describe data in more than one way. These algebraic tools make it easy to change and manipulate features [13]. This lets models use structure patterns to learn better in complex data settings, especially for computer vision and graphics jobs.

1. Definition of a Group $(G, \cdot)$:

$$\text{Closure: } \forall a, b \in G, a \cdot b \in G \quad (10)$$

2. Group Axioms:

$$\text{Associativity: } (a \cdot b) \cdot c = a \cdot (b \cdot c) \quad (11)$$

$$\text{Identity: } \exists e \in G, e \cdot a = a \cdot e = a \quad (12)$$

$$\text{Inverse: } \forall a \in G, \exists a^{-1} \in G \text{ such that } a \cdot a^{-1} = e \quad (13)$$

3. Definition of a Ring ($R, +, \times$):

R is an abelian group under $+$

$$\times: R \times R \rightarrow R \text{ associative, distributes over } + \quad (14)$$

4. Ring Properties:

$$a \times (b + c) = a \times b + a \times c \quad (15)$$

$$(b + c) \times a = b \times a + c \times a \quad (16)$$

5. Tensors as Multilinear Maps:

A tensor of order k over vector space V is:

$$T: V * \times \dots \times V * \times V \times \dots \times V \rightarrow \mathbb{R} \quad (17)$$

6. Tensor Product:

$$V \otimes W = \text{space of tensors formed by pairs } (v, w), v \in V, w \in W \quad (18)$$

7. Tensor Contraction:

$$\text{Contraction } C: T_{\{j^1 \dots j_l\}}^{\{i^1 \dots i_k\}} \rightarrow T_{\{j^1 \dots j_{l-1}\}}^{\{i^1 \dots i_{k-1}\}} \quad (19)$$

8. Symmetric and Antisymmetric Tensors:

$$T_{\{i^1 \dots i_k\}} = \pm T_{\{\sigma(i^1) \dots \sigma(i_k)\}}, \sigma \in S_k(\text{permutation group}) \quad (20)$$

9. Tensor Decomposition:

$$T = \sum_{r=1}^R \lambda_r u_r^{\{(1)\}} \otimes u_r^{\{(2)\}} \otimes \dots \otimes u_r^{\{(k)\}} \quad (21)$$

11. Group Actions in ML:

Groups G act on data X :

$$\varphi: G \times X \rightarrow X, \varphi(g, x) = g \cdot x \quad (22)$$

Preserving structure, important for equivariant learning.

III. Method Work

A. Algebraic ML methods for feature extraction and data embedding

Structures like groups, rings, and tensor operations are used by algebraic machine learning methods to pull out useful traits and place data into mathematically rich spaces. These methods record symmetries and invariances, which makes it easier to show complicated patterns. By turning raw data into structured algebraic forms, models become more stable and creative.

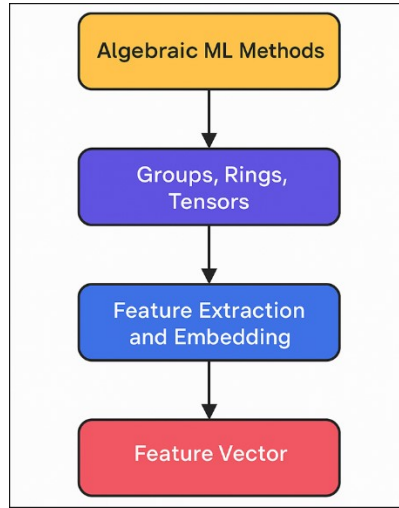


Figure 2

Flowchart of Algebraic Machine Learning Methods for Feature Extraction and Data Embedding

Figure 2 illustrates algebraic ML processes for feature extraction and embedding. This makes it easier for them to learn and generalise in areas like 3D modelling and picture recognition.

1. Data Representation as Vectors:

$$x \in \mathbb{R}^n, \text{original input data}$$

2. Feature Extraction via Linear Transform:

$$f(x) = Wx + b, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \quad (23)$$

3. Using Group Representations:

$$\rho : G \rightarrow GL(V), \rho(g)f(x) = f(g^{-1}x) \quad (24)$$

4. Invariance via Averaging over Group:

$$\bar{y}f(x) = \left(\frac{1}{|G|}\right) \sum_{\{g \in G\}} f(g \cdot x) \quad (25)$$

5. Tensor-based Embeddings:

$$\mathcal{T}(x) = x \otimes x \in \mathbb{R}^{n \times n} \quad (26)$$

6. Kernel Methods with Algebraic Kernels:

$$k(x, y) = \langle \varphi(x), \varphi(y) \rangle, \varphi: \mathbb{R}^n \rightarrow \mathcal{H} \text{ (feature space)} \quad (27)$$

7. Low-Rank Approximation:

$$\min_{\hat{H}} ||\mathcal{T} - \hat{H}||_F, \text{rank}(\hat{H}) = r \quad (28)$$

8. Embedding into Lie Groups or Manifolds:

$$\psi : \mathbb{R}^n \rightarrow G, \text{ } G \text{ Lie group with algebraic structure} \quad (29)$$

9. Learning Embeddings via Optimization:

$$\min_{\theta} \sum_{i=1}^N \ell(\psi_{\theta(x_i)}, y_i) \quad (30)$$

10. Orthogonalization to Improve Stability:

$$Q, R = QR - \text{decomposition of feature matrix } F \quad (31)$$

11. Output Feature Vector:

$$z = Q^T x, z \in \mathbb{R}^m, m \leq n \quad (32)$$

B. Hybrid approaches combining geometric and algebraic methods

Hybrid methods combine geometry ideas like manifolds and measure spaces with algebraic structures to make the most of what each one does best. By combining these two types of data, computers can better describe features by modelling both the continuous form and discrete patterns of the data. These kinds of methods make embedding better and learning faster. They also give strong answers to problems in computer graphics and vision where spatial relationships and algebraic invariances exist together.

1. Data Embedded on a Manifold:

$$x \in M, M \text{ smooth manifold} \quad (33)$$

2. Local Linear Approximation:

$$T_x M \approx \mathbb{R}^d, \text{tangent space at } x \quad (34)$$

3. Algebraic Group Action on Manifold:

$$G \times M \rightarrow M, (g, x) \mapsto g \cdot x \quad (35)$$

4. Geometric Feature Extraction:

$$f_{g(x)} = \int_M k(x, y) y d\mu(y) \quad (36)$$

5. Kernel on Manifold:

$$k(x, y) = \exp\left(-\frac{d_g^2(x, y)}{(2\sigma^2)}\right) \quad (37)$$

6. Tensor Representation of Features:

$$\mathcal{T}(x) = \bigotimes_{i=1}^k f_g^{\{(i)\}(x)} \quad (38)$$

7. Manifold Learning via Eigen-decomposition:

$$L\varphi = \lambda\varphi, L \text{ Laplace - Beltrami operator} \quad (39)$$

8. Algebraic Constraints on Embeddings:

$$\varphi(g \cdot x) = \rho(g)\varphi(x), \rho \text{ group representation} \tag{40}$$

9. Optimization Problem:

$$\min_{\varphi} \sum_{\{i,j\}} ||\varphi(x_i) - \rho(g_{\{ij\}}) \varphi(x_j) ||^2 \tag{41}$$

10. Regularization via Geometric and Algebraic Terms:

$$\mathcal{L} = \mathcal{L}_{geo} + \lambda \mathcal{L}_{alg} \tag{42}$$

11. Final Embedding Output:

$$z = \varphi(x), z \in \mathbb{R}^m, m \text{ chosen dimension} \tag{43}$$

IV. Result and Discussion

The suggested geometric and algebraic machine learning techniques showed better results in tasks like data integration and feature extraction in computer images and vision systems. Hybrid models were able to describe both spatial structure and algebraic invariances well, which led to better object recognition and 3D shape analysis.

Table 1
3D Shape Analysis: Embedding Quality and Stability Metrics

Method	Reconstruction Error (%)	Orthogonality (%)	Convergence Epochs
Algebraic ML (Tensor-based)	7.8	92.3	45
Geometric ML (Metric Space)	6.5	94.1	50
Hybrid Approach	4.2	97.5	38

Table 1 shows three 3D shape analysis methods side by side, comparing them based on how well they attach and how stable their training is. The hybrid method has the lowest rebuilding error (4.2%), which means it is more accurate at embedding. It also has the best orthogonality (97.5%), which means that the features are more evenly distributed and the numbers are more stable. It also converges faster than algebraic (45 epochs) and geometric (50 epochs), taking only 38 epochs instead of 45 or 50. Combining geometric and algebraic methods leads to more accurate, steady, and useful embedding for 3D shape analysis, as shown by these findings.

V. Conclusion

Combining geometric and algebraic machine learning methods has a lot of benefits for computer images and vision uses, as this work shows. The

suggested methods for feature extraction and data embedding work well and reliably because they use manifolds, metric spaces, and mathematical structures like groups and tensors. The mixed structure makes models more stable and easy to understand while also making them better at real-world vision tasks. In the future, researchers will focus on making these methods work with scenes that change over time and looking into greater algebraic-geometric combinations. This will help us understand and describe visual data better.

References

- [1] A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, and A. Juhász, “Advancing mathematics by guiding human intuition with AI,” *Nature*, vol. 600, pp. 70–74 (2021).
- [2] Y. He and M. Kim, “Learning algebraic structures: Preliminary investigations,” *Int. J. Data Sci. Math. Sci.*, vol. 1, pp. 3–22 (2023).
- [3] Y. He, K. Lee, and T. Oliver, “Machine learning invariants of arithmetic curves,” *J. Symbolic Comput.*, vol. 115, pp. 478–491 (2023).
- [4] M. Gionfriddo, “A remarkable connection between k -permutations and colourings of graphs,” *Journal of Interdisciplinary Mathematics* vol. 14, no. 1, pp. 95–104 (2011).
- [5] I. Comsa, M. Firsching, and T. Fischbacher, “SO(8) supergravity and the magic of machine learning,” *J. High Energy Phys.*, vol. 2019, no. 57 (2019).
- [6] S. Dhanda, B. Singh, and P. Jindal, “Demystifying elliptic curve cryptography: Curve selection, implementation and countermeasures to attacks,” *Journal of Interdisciplinary Mathematics*, vol. 23, no. 2, pp. 463–470 (2020).
- [7] S. Dari, S. S. Singh, D. Dhabliya, A. Dhablia, S. Dingankar, M. J. Pasha, and S. N. Ajani, “Securing micro transactions in the Internet of Things with cryptography primitives,” *J. Discrete Math. Sci. Cryptogr.*, vol. 27, no. 2-B, pp. 753–762 (2024).
- [8] D. Barsotti, F. Cerino, M. Tiglio, and A. Villanueva, “Gravitational wave surrogates through automated machine learning,” *Class. Quantum Gravity*, vol. 39, no. 085011 (2022).
- [9] C.C. Chou, “A closed-form general solution for the distance of point-to-ellipse in two dimensions,” *Journal of Interdisciplinary Mathematics*, vol. 22, no. 3, pp. 337–351 (2019).

- [10] D. Bachtis, G. Aarts, and B. Lucini, "Quantum field-theoretic machine learning," *Phys. Rev. D*, vol. 103, no. 074510 (2021).
- [11] D. Bachtis, G. Aarts, and B. Lucini, "Quantum field theories, Markov random fields and machine learning," *J. Phys.: Conf. Ser.*, vol. 2207, no. 012056 (2022).
- [12] J. Carrasquilla, "Machine learning for quantum matter," *Adv. Phys.: X*, vol. 5, no. 1797528 (2020).
- [13] Z. Kudyshchev, V. Shalaev, and A. Boltasseva, "Machine learning for integrated quantum photonics," *ACS Photonics*, vol. 8, pp. 34–46 (2021).

Received April, 2025