ISSN 2186-7437

NII Shonan Meeting Report

No. 226

The Power of Geometric Algebra in Modern Computer Vision

Diego Thomas
Vincent Nozick
Takuya Funatomi

May 19–22, 2025

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

The Power of Geometric Algebra in Modern Computer Vision

Organizers: Diego Thomas[1],
Vincent Nozick[2], and Takuya Funatomi[3]

[1]Kyushu University, Japan
[2]Université Gustave Eiffel, France
[3]Nara Institute of Science and Technology, Japan

May 19–22, 2025

Background and introduction

In the last decade, deep learning has become ubiquitous in Computer Vision to solve various tasks ranging from object recognition to 3D and 4D capture. A vast majority of recent scientific papers published in top level venues such as the international conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision (ICCV) or the IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI) rely on deep neural models formulated in the linear algebra. This is because of the simplicity of the basic operations in the neurons: scalar multiplication and addition, and the availability of many existing tools such as auto differentiation in pytorch. However, for some tasks such as 3D transformation, other algebras such as Clifford Algebra (also called Geometric Algebra) have proven to be more efficient. Deep neural networks formulated with geometric algebra have been proposed recently, with promising applications in Computer Graphics. However, none has been applied yet to solve Computer Vision tasks.

Overview of the meeting

The meeting is envisaged to focus on new settings and applications of AI models defined with Geometric algebra for tasks in Computer Vision such as 3D and 4D capture. The meeting will invite internationally leading and renown researchers in the fields of Geometric Algebra, Computer Vision and Computer Graphics, whose contributions are likely to be essential to the field. We anticipate that the meeting will foster discussions and new ideas to open a new research area at the frontier between Geometric Algebra, Computer Vision and Computer Graphics. The meeting will also be a wonderful opportunity to strengthen existing collaborations and create new collaborations between top-level researchers in Japan and abroad. The meeting will focus on the following two topics.

Deep learning with Geometric Algebra

Equipping deep neural networks with geometric priors has achieved several successes ([1, 2]) to model geometric transformations of 3D shapes. Some very recent Geometric Algebra based networks, such as Clifford Neural Layers, embed some powerful properties like equivariance (i.e. invariance with respect to the 3D symmetry group) that strongly simplifies some standard networks. Other networks, like Geometric Clifford Algebra Network (GCANs), provide several interesting advantages over classical linear algebra that could open new possibilities in the field of Computer Vision: (1) GCANs naturally and efficiently encode the transformations and the invariant elements of classic geometries. (2) In Geometric Algebra, objects transform covariantly with transformations of space. This means that a single function can transform multiple types of objects, including vectors, points, lines and planes. (3) geometric algebra generalizes over dimensions in the sense that transformations and objects are constructed consistently regardless of the dimensionality of the space. The objective of this seminar is to give an overview of recent successes of deep learning with geometric priors and revisit classical Computer Vision problems from a geometric algebra perspective. A first target will be to clarify formulation of deep neural networks using geometric algebra and existing tools to implement them. A second objective will be to find promising future research directions and concrete applications. For this purpose, the seminar will invite international experts in the domain of geometric algebra, computer vision and computer graphics.

3D and 4D capture with the support of Geometric AI

Research on Neural Radiance Fields (NeRF) has recently gained significant attention within the realms of 3D Vision and Computer Graphics. Notably, the combination of NeRF with Signed Distance Fields (SDF) has demonstrated impressive outcomes in the field of multi-view 3D scene reconstruction. There is a current surge in exploring the adaptation of established concepts for reconstructing dynamic scenes. In this context, the introduction of neural warp fields has been proposed to capture non-rigid 3D shape transformations across multiple images. While recent findings show promise, they are constrained by their efficacy primarily in handling minor motions and often demand extensive training periods. Furthermore, these models exhibit limitations in interpolation and extrapolation capabilities. The fact that Geometric Algebra is especially performant for such interpolation tasks is promising to find new solutions and improve current state-of-the-art. The primary objective of this seminar is to provide a comprehensive overview of existing alternatives for modeling 3D non-rigid deformations. The initial focus will involve exploring novel formulations of 3D neural warp fields capable of learning to deform 3D shapes and volumes based on input images and videos. A secondary aim is to identify exciting new applications of Geometric AI, specifically employing deep learning with Geometric Algebra, for various tasks in Computer Vision. This involves elucidating the advantages of geometric algebra over classical linear algebra and highlighting the available tools for implementing concrete solutions. To achieve these goals, the seminar aims to bring together participants from diverse backgrounds. This includes experts in Geometric Algebra with a keen interest in its applications in Computer Vision, as well as specialists in Computer Vision intrigued by alternative approaches to modeling 3D shape transformations and manipulations.

Overview of Talks

Keynote & Workshop: The Power of Geometric Algebra in Modern Computer Vision

Martin Roelfs, University of Antwerp & Flanders Make

In order to give an introduction into the modern framework of Projective Geometric Algebra (PGA) and the modern `kingdon` GA library[3], this keynote talk combines a theoretical introduction with a hands-on workshop.

The theoretical part introduces the plane-based mindset of (P)GA, and the geometric interpretation of the elements and binary operators of (P)GA. This serves as a build up to one of the most important features of plane-based (P)GA: dimension agnostic thinking. This mindset has as a consequence that (if proper care is taken) algorithms will work in any number of dimensions. The talk showed some examples, like the thin lens and the tesseract on a string.

In the workshop the participants then get some hands-on experience with `kingdon`, by working through several selected exercises in the `kingdon` teahouse [1].

Keynote: Statistical Approaches for Internal Anatomy Prediction

Marilyn Keller, Kyushu University

The talk "Statistical Approaches for Internal Anatomy Prediction" presents recent advances on research about generating personalized anatomical digital twins, which are essential for medicine, computer graphics, and biomechanics. Observing internal anatomy usually requires expensive medical imaging. Instead, we can leverage the correlation between external body shape and internal structures to predict the anatomy directly from a subject' s appearance [4, 5]. Learning this correlation raises three key challenges: building datasets with paired observations of body shapes and internal anatomy, annotating these datasets, and learning models that capture the correlation between external and internal features. In this talk, Marilyn showcased how we became able to predict skeleton geometry, bone location, and soft tissue distribution solely from external body shape

Keynote: Geometric Algebra in Medicine

Eckhard Hitzer, International Christian University

The talk "Geometric Algebra in Medicine" focused on novel computational workflows for natural and biomedical image processing based on hypercomplex algebras and on the project of the venture business company ORamaVR[2] toward Accelerating Computational Medical XR (Founded by G. Papagiannakis et al). At the beginning it was shown how within the feature-rich hypercomplex setting (in particular that of quaternion algebra applied to color image processing [6]),

[1] `https://tbuli.github.io/teahouse/`
[2] `https://oramavr.com/`

novel image processing workflows can be realized for natural and biomedical images enabling alternative visual representations, offering effective solutions to current problems in computer vision and digital pathology, and generally expanding the scope and impact of hypercomplex image processing across a wide range of applications.

Then we showed how GA leads to impactful XR computer graphics, where GA serves as single virtual human enabling simulation framework. ORamaVR provides platforms for medical XR training developed in collaboration with medical institutions, institutes and universities across the globe. Comparative analysis confirms the VR training effectiveness. ORamaVR also provides international real time shared medical surgery VR collaboration. A key component is the ability to deform, cut and tear a skinned model using CGA. The ORamaVR developed MAGES 4.0 framework allows for automation in action development, VR recorder capture and replay of VR session, realistic real-time cut, tear and drill algorithm, AR and mobile (ios) support, has a dissected edge physics engine, provides edge-cloud remote visual rendering, optimized networking layering with collaboration of AR/VR devices, convolutional neural network automatic assessment, and new template applications (open source). The integration of generative models revolutionizes ORamaVR content creation.

The background are an over 150 year old outdated medical educational residency model of master and apprentice, an expected short fall of 10 million medical professionals by 2030, and that 5 billion people lack access to affordable surgical & anesthesia care according to the WHO. ORamaVR achieves

1. Democratized XR content development and access
2. Increased medical XR curricula adoption
3. Increased trainee competency & proficiency

with its leading medical-XR authoring, training & Assessment software platform, which offers for educators: Create, Record, Publish your medical XR training simulation, for learners: See, Do, Teach to achieve competency, proficiency, expertise, and finally objective metrics, performance analytics and AI co-tutors for all.

Keynote: Optimisation under geometric constraints for designing Kinematic Chains

Shizuo Kaji, Kyushu University

Kinematic chains, such as those found in robotic arms, have joint states that can be represented using geometric algebra. The overall configuration of the arm, given a specified position of the end effector, corresponds to a subspace of the direct product of geometric algebras. The design problem for kinematic chains can thus be formulated as the minimisation of an energy function defined on this subspace. Focusing particularly on closed chains known as Kaleidocycles, this talk introduces two types of energy functions: (1) a discretised elastic energy and (2) total torsion. It is shown that optimising these functions yields structures with special properties, such as one degree of freedom (1-DoF) [7, 8, 9].

Keynote: Corrected curvature measures — A unified approach for the geometric analysis of discrete data

Jacques-Olivier Lachaud, University Savoie Mont Blanc

We propose a new mathematical and computational tool for inferring the geometry of shapes known only through approximations, such as triangulated or digital surfaces. The main idea is to decouple the position of the shape boundary from its normal vector field. To do so, we extend a classical tool of geometric measure theory, the normal cycle, so that it takes as input not only a surface but also a normal vector field. We formalize it as a current in the oriented Grassmann bundle. By choosing adequate differential forms, we define geometric measures like area, mean, and Gaussian curvatures. We then show the stability of these measures when both position and normal input data are approximations of the underlying continuous shape. As a byproduct, our tool is able to correctly estimate curvatures over polyhedral approximations of shapes with explicit bounds, even when their natural normals are not correct, as long as an external convergent normal vector field is provided. We show that this framework induces state-of-the-art curvature estimations on polyhedral surfaces, digital surfaces, and even point clouds with normal data [10, 11, 12]. This talks gather joint works with Pascal Romon, Boris Thibert, David Coeurjolly, and Céline Labart.

List of Participants

- Prof. Diego Thomas, Kyushu University
- Prof. Vincent Nozick, Université Gustave Eiffel
- Prof. Takuya Funatomi, Nara Institute of Science and Technology
- Prof. Shizuo Kaji, Kyushu University
- Prof. Hiroyuki Ochiai, Kyushu University
- Prof. Jacques-Olivier Lachaud, University of Savoie Mont-Blanc
- Prof. Eckhard Hitzer, International Christian University
- Dr. Martin Roelfs, University of Antwerp, Flanders Make
- Prof. Kanta Tachibana, Kogakuin University
- Prof. Zhaoyuan Yu, Nanjing Normal University
- Prof. Akihiro Sugimoto, National Institute of Informatics
- Prof. Pascal Monasse, Ecole Nationale des Ponts et Chaussées
- Prof. Hideo Saito, Keio University
- Dr. Clement Chomicki, Kyushu University
- Prof. Stephane Breuils, Universite Savoie Mont Blanc
- Prof. Shohei Hidaka, JAIST
- Prof. Takuma Torii, Tokyo Denki University
- Dr. Marilyn Keller, Kyushu University

Meeting Schedule

Check-in Day: May 18 (Sun)

19:00–21:00 Welcome Banquet

Day 1: May 19 (Mon)

09:00–09:15 Welcoming address - Diego Thomas, Vincent Nozick, Takuya Funatomi

09:15–10:00 Research introductions (5 mn by each participant)

10:00–10:30 Coffee break

10:30–12:00 Research introductions (5 mn by each participant)

12:00–14:00 Lunch

14:00–15:00 Tutorial 1 - Martin Roelfs

15:00–15:30 Coffee break

15:30–16:30 Tutorial 2 - Martin Roelfs

16:30–17:00 Break

17:00–18:00 Breakout session 1

18:00–19:30 Dinner

Day 2: May 20 (Tue)

07:30–09:00 Breakfast

09:00–09:15 Program briefing

09:15–10:00 Breakout session 2

10:00–10:30 Coffee break

10:30–12:00 Breakout session 3

12:00–13:30 Lunch

13:30–14:00 Keynote 1 - Marilyn Keller

14:00–14:30 Keynote 2 - Eckhard Hitzer

14:30–15:00 Keynote 3 - Shizuo Kaji

15:00–15:30 Coffee break

15:30–16:00 Keynote 4 - Shohei Hidaka

16:00–16:30 Keynote 5 - Jacques-Olivier Lachaud

16:30–18:00 Breakout session 4

18:00–19:30 Dinner

Day 3: May 21 (Wed)

07:30–09:00 Breakfast

09:00–09:15 Program briefing

09:15–10:00 Breakout session 5

10:00–10:30 Coffee break

10:30–12:00 Breakout session 6

12:00–13:30 Lunch

13:30–20:45 Excursion and Main Banquet

Day 4: May 22 (Thu)

07:30–09:30 Breakfast and check-out

09:30–10:00 Final presentations by each group

10:00–10:30 Coffee break

10:30–11:00 Group photo

11:00–11:30 Final presentations by each group

11:30–12:00 Conclusion and wrap-up

Figure 1: View of Mount Fuji on the last day of the meeting.

Summary of discussions

Introduction and Tutorials

The first day of the Shonan Meeting began with an engaging tutorial led by Prof. Roelf, who introduced participants to the `kingdon` GA library and the foundational principles of Projective Geometric Algebra (GA). Through a series of interactive demonstrations, the tutorial offered a hands-on approach to understanding how GA provides a powerful framework for geometric computing. Participants actively explored key operators such as the meet operator, gaining a deeper understanding of how GA can be used to represent and manipulate geometric relationships in an elegant and efficient way.

Beyond the theoretical explanations, the `kingdon` GA library was presented as both a pedagogical resource and a computational tool. On the one hand, its ability to produce dynamic, interactive geometric drawings makes it an excellent platform for teaching and conceptual visualization. On the other hand, the library also holds promise as a potential alternative to traditional linear algebra approaches in modern optimization pipelines. Participants discussed how GA's expressive geometric modeling could complement or, in some cases, replace matrix-based methods in applications ranging from robotics to computer graphics.

Collaborative Group Work

At the heart of the meeting were the collaborative group activities. Participants were divided into interdisciplinary teams, each typically consisting of four key roles: a senior researcher with deep domain knowledge, a younger researcher with strong programming and implementation skills, a participant with a real-world problem to solve, and an expert in geometric algebra. This carefully balanced composition fostered rich interdisciplinary collaboration and knowledge exchange.

Over the course of three days, these groups engaged in focused efforts to apply GA tools to real-world problems in Computer Vision and Computer Graphics. The working sessions were characterized by lively and productive discussions, both within each group and informally during breaks, where cross-group dialogues emerged. Participants explored concrete use cases ranging from 3D reconstruction and camera geometry to surface modeling and transformation analysis.

Final Presentations and Reflections

On the final day of the meeting, each group presented their findings, highlighting both the progress made and the challenges encountered. These presentations led to a broader roundtable discussion where participants reflected on the potential and limitations of Geometric Algebra in current research and industrial applications.

A key insight that emerged from these discussions was that many of the practical problems initially posed could indeed be revisited — and in some cases, significantly simplified — using existing GA tools. Geometric Algebra was praised for offering intuitive, coordinate-free representations that enhance

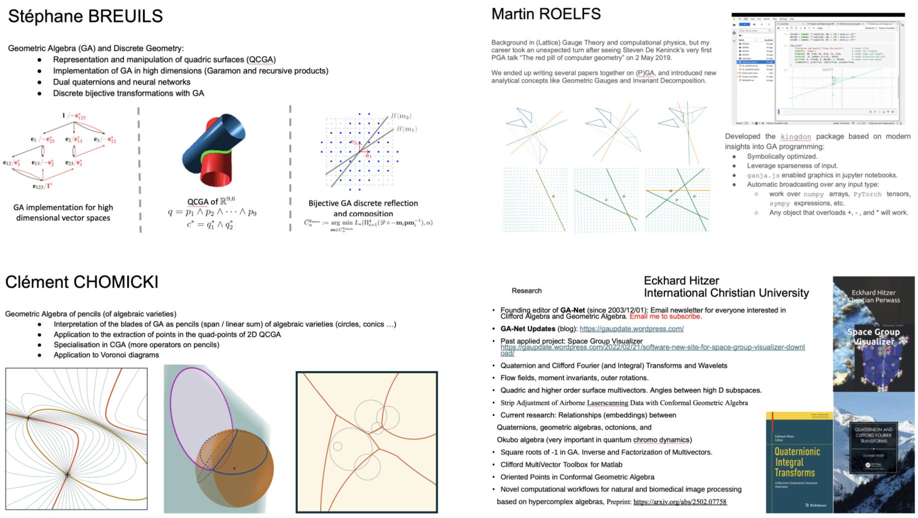

Figure 2: Representative slides of discussions about Geometric Algebra at Shonan meeting #226.

conceptual clarity. In particular, its ability to unify various geometric transformations under a single algebraic framework was seen as both educationally valuable and practically powerful.

Future Directions and Challenges

While the benefits of GA were enthusiastically acknowledged, participants also recognized the current technological and computational gaps that need to be addressed for broader adoption. One significant challenge lies in the performance optimization of GA libraries like `kingdon` GA library. At present, many state-of-the-art optimizations—such as GPU acceleration, vectorized operations, and parallel computing—are more mature and accessible in linear algebra-based toolkits.

Bridging this performance gap represents a promising area for future development. Enhancing the computational efficiency of GA libraries would not only make them more competitive with established linear algebra frameworks but also open the door to their use in high-performance applications, including real-time rendering, simulation, and deep learning pipelines.

Figures and Illustrations

- Figure 2 illustrates the range of theoretical and methodological topics discussed by experts in Geometric Algebra during the meeting.
- Figure 3 presents a selection of application domains where GA shows significant promise, based on the concrete problem-solving efforts of the working groups.

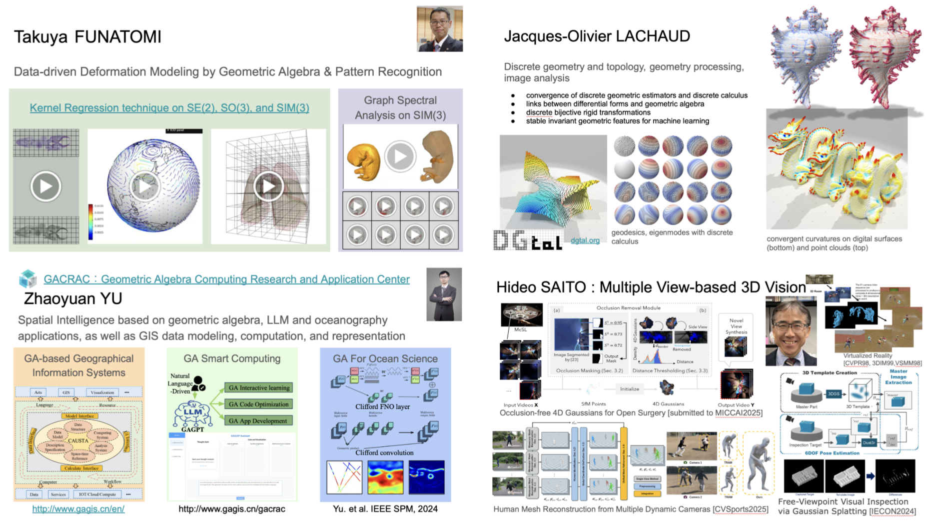

Figure 3: Representative slides of discussions about Concrete applications of Geometric Algebra in Computer Vision and Graphics at Shonan meeting #226.

Summary of new findings

Project A: Fitting faces of Voronoi diagram to a point cloud

Motivation and Background

Voronoi diagrams defined over discretized 3D volumes provide an alternative method for extracting 3D meshes from volumetric data, offering notable advantages such as improved regularity and a reduced number of cells needed to capture fine details. When 3D shape observations are available as point clouds, there is considerable interest in aligning the faces of the Voronoi diagram with these observations to enhance the quality of 3D mesh reconstruction. Since the vertices of a Voronoi diagram can be described as intersections of bisectors, Geometric Algebra (GA) holds promise for developing efficient representations and more effective optimization strategies.

Goal

PyTorch offers a user-friendly optimization framework with built-in automatic differentiation through its autograd system. Inspired by recent approaches that formulate differentiable Voronoi diagram optimization using standard linear algebra, the goal of this project is to reformulate the problem using Geometric Algebra (GA). Specifically, we aim to implement a PyTorch-based pipeline that directly optimizes the positions of Voronoi sites using only GA operations, as provided by the `kingdon` GA library [3].

Outcome

We transcribed the algorithm proposed in VoroMesh [13] onto the board and reformulated all its operations using Geometric Algebra (GA) operators. We then implemented the algorithm using the `kingdon` GA library and conducted basic 2D experiments, where we fit the faces of a Voronoi diagram to a set of points randomly sampled along a circle.

(Centroidal) Voronoi Tessellation. A tessellation of a 3D space is a disjoint set of polyhedron that fills the 3D space of interest. Centroidal Voronoi Tesselation (CVT) are used in a wide range of applications in Computer Graphics as it provides an elegant tool to compute a regular and optimized discretization of the 3D space [14]. Voronoi tessellation is the dual of the Delaunay triangulation and is defined by its sites: a set of 3D points in space.

A Voronoi cell V_i is associated with its site x_i and consists of all points in space that are closest to x_i:

$$\{p \in \ R^3 \ / \ ||p - x_i|| < ||p - x_j||, \ j \in [1, K], j \ ! = \ i\}$$

The bisector $B(i, j)$ of two sites x_i and x_j is the set of points $\{p \in R^3 / \ d(p, x_i) = d(p, x_j)\}$. In 2D it is a line, in 3D it is a plane. The bisector of multiple sites is the intersection between the bisectors of the pairs of sites. In 2D this creates segments. In 3D this creates polygons that can be subdivided into triangles. Each site i is associated with a set of bisectors B_i that represents a polygon and is composed of triangles. This also can be understood as the flat membrane that separates two sites. When each site of a Voronoi tessellation coincides with

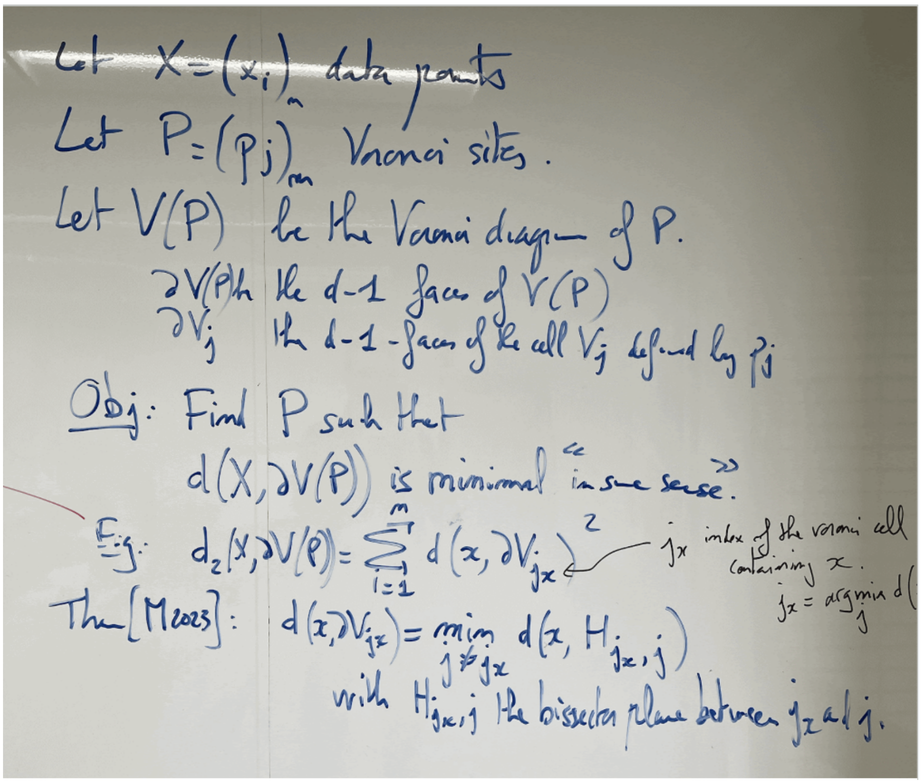

Figure 4: The original Voromesh algorithm.

the centroid of its associated cell, the tessellation is called a Centroidal Voronoi Tessellation (CVT).

Differentiable fitting of the faces of a Voronoi Diagram (VoroMesh). As illustrated in Fig. 4, the core idea behind VoroMesh is to compute the distance from a point within a Voronoi cell to its boundary by finding the minimum distance to the bisectors formed between the cell's site and its K-nearest neighbors. This formulation is fully differentiable, enabling optimization through PyTorch's Adam optimizer.

We express the computation of minimum distances and bisectors using Geometric Algebra (GA). To integrate the `kingdon` GA library with PyTorch and leverage its autograd capabilities, each site is represented as a bivector with parameters defined as PyTorch tensors. The corresponding implementation is shown in Fig. 5.

The VoroMesh algorithm begins by identifying the Voronoi cell that contains each target point, which is equivalent to finding the closest site for each point. This is achieved using the meet operator (&) followed by an argmin operation. As shown in Fig. 6, this step can be implemented in a single line of Python code, where *Pts* represents the target points and *Sites* denotes the Voronoi sites.

The distance from each target point to the boundary of its corresponding

```python
x = torch.linspace(-2.0, 2.0, 11, requires_grad=True)
y = torch.linspace(-2.0, 2.0, 11, requires_grad=True)
X, Y = torch.meshgrid(x, y, indexing='ij')
X = X.reshape(-1)
Y = Y.reshape(-1)
X.retain_grad()
Y.retain_grad()
print(X.shape)

optimizer = torch.optim.Adam([{"params": X, "lr": 0.01},
                              {"params": Y, "lr": 0.01}])

optimizer_ready = True

Sites = alg.bivector(e01=-X, e02=Y, e12=torch.ones(X.shape[0]))
```

Figure 5: Connecting the `kingdon| GA library bivectors with Pytorch to access autograd.`

```python
Sites = alg.bivector(e01=-X, e02=Y, e12=torch.ones(X.shape[0]))

with torch.no_grad():
    closest_points = [(Pts[i] & Sites).norm().map(torch.argmin).e for i in range(Pts.shape[1])]
```

Figure 6: Identifying closest points with the meet operator.

Voronoi cell is computed as the minimum distance to the bisectors formed between the closest site and all other sites.[3] This is implemented using the meet operator (&) and the reject operator (|). The corresponding code is shown in Fig. 7.

By leveraging the Pytorch autograd functions we repeat the optimization step several times (about 60 times) to converge to the final solution.

Experiments and preliminary results. As shown in Fig.8, we conducted a simple experiment by initializing a uniform grid of points to define the initial Voronoi diagram and randomly sampling points along a circle of radius 1. We then applied the GA-based VoroMesh implementation to fit the Voronoi faces to the target point cloud. The resulting fits are presented in Fig.9.

These results demonstrate that the `kingdon` GA library, with its GA operators, integrates smoothly with the PyTorch optimization framework. The Voronoi faces adapt well to the structure of the target point cloud.

However, the optimization process was relatively slow, despite involving only 100 target points.

[3]This process can be accelerated using a K-nearest neighbors approach, but for simplicity and proof of concept, we use a naive algorithm here.

```python
optimizer.zero_grad()

all_pi = Sites[closest_points]
distances = 1.0e32*torch.ones(Pts.shape[1], Sites.shape[1])

loss = 0.0
for i in range(Pts.shape[1]):
    for j in range(Sites.shape[1]):
        if closest_points[i] == j:
            continue
        p_j = Sites[j]
        b = ((all_pi[i] & p_j) | (all_pi[i] + p_j)).normalized()
        dist_curr = (((Pts[i] | b) * b) & Pts[i]).norm()
        distances[i,j] = dist_curr.e

min_dist, _ = torch.min(distances, dim = 1)
loss = (torch.square(min_dist)).sum()

print("loss: ", loss)

loss.backward()

optimizer.step()
```

Figure 7: Compute the distance of target points to Voronoi cells.

Conclusions. In this project, we demonstrated that Geometric Algebra (GA) libraries can be effectively integrated into modern PyTorch optimization frameworks. We also showed that classical computer vision tasks can be re-expressed with just a few lines of code using GA. However, our experiments indicate that more sophisticated optimization techniques are necessary for practical applications that aim to compete with standard, highly optimized linear algebra implementations. How such optimizations might be achieved remains an open question. Nevertheless, GA libraries like `kingdon` GA library serve as excellent educational tools and offer a promising new perspective for rethinking computer vision problems.

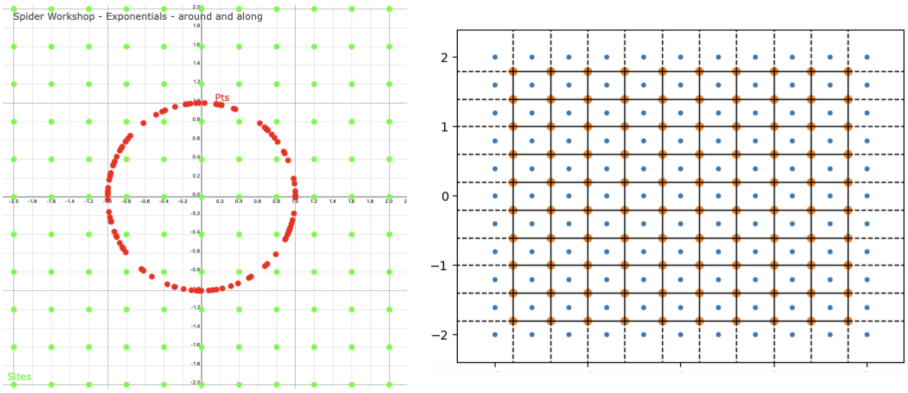

Figure 8: Initial state.

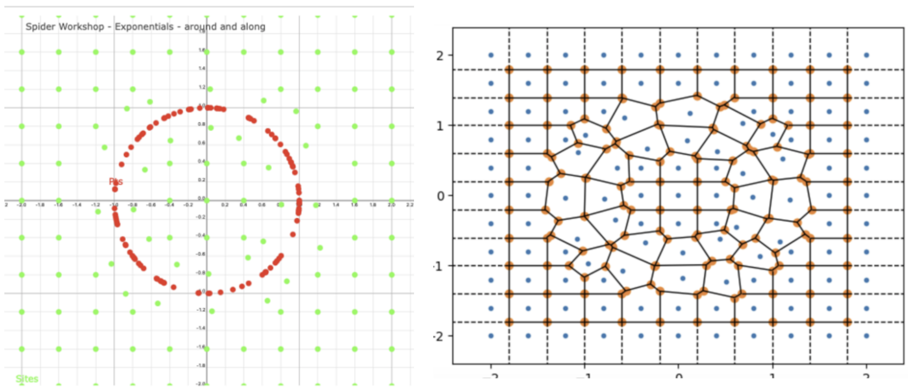

Figure 9: Final state after optimization.

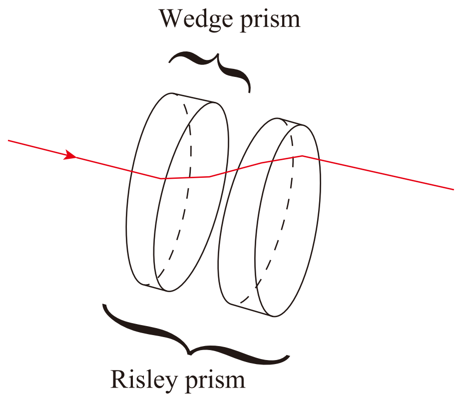

Figure 10: Wedge prism and Risley prism.

Project B: Snell's law and its application to Risley prism

Motivation and Background

A *wedge prism* (Fig. 10) is an optical element with a trapezoidal cross-section, formed by two slightly tilted (non-parallel) flat surfaces. The angle between these surfaces is referred to as the "wedge angle." When light passes through a wedge prism, its propagation direction is slightly refracted and deviated. The deflection angle is determined during fabrication based on the wedge angle, and wedge prisms are typically used for fine image displacement or precise laser beam steering.

By combining two wedge prisms and allowing each to rotate independently, a configuration known as a *Risley prism* (Fig. 10) can be constructed. The rotation of each prism enables continuous control over both the direction and magnitude of beam deflection. This allows a single device to scan a beam over a wide angular range in two dimensions.

Both wedge and Risley prisms lack rotational symmetry with respect to the optical axis, causing light to be refracted asymmetrically when passing through. As a result, describing the behavior of light rays in these systems using conventional coordinate-based methods can become geometrically complex.

However, Projective Geometric Algebra (PGA) enables a compact and unified representation of Snell's law that does not require any explicit trigonometry. Since the surfaces composing wedge and Risley prisms are planar, they can also be expressed very simply within the PGA framework. Therefore, PGA is a highly suitable and efficient mathematical tool for modeling light-ray behavior in these optical systems.

Goal

To derive a general expression for Snell's law using PGA. A wedge prism is described by two planar surfaces, and it will be shown that the behavior of a light ray passing through it can be concisely expressed by applying Snell's law. Similarly, a Risley prism, composed of two independently rotating wedge prisms, will be described using PGA, and it will be demonstrated that the propagation of a light ray through this system can also be efficiently described.

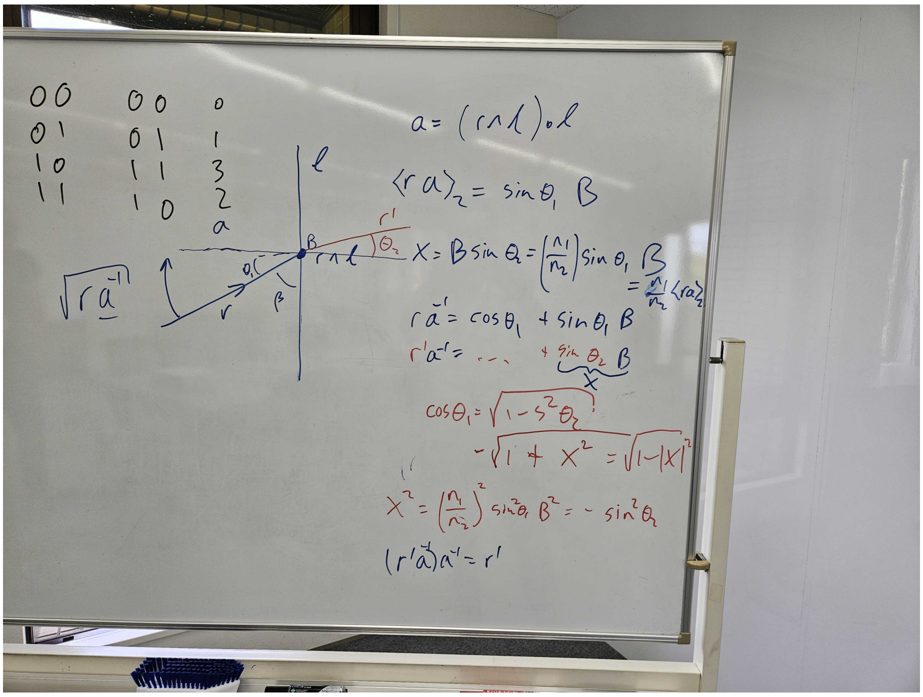

Figure 11: Snell's law solved on a whiteboard.

Outcome

Snell's law was solved on a whiteboard as presented in Fig. 11, then light-ray behavior was demonstrated using `kingdon` GA library [3].

Snell's law in Projective Geometric Algebra. We denote the incident ray as $\mathbf{r}$ and the interface as $\mathbf{l}$ (see Fig. 12). The intersection point $\mathbf{x}$ of them is determined by the outer product.

$$\mathbf{x} = \hat{\mathbf{r}} \wedge \mathbf{l}, \tag{1}$$

where $\hat{\mathbf{r}}$ is the normalized $\mathbf{r}$. This formula works in both 2D and 3D, where $\mathbf{l}$ is a line or a plane respectively, while $\mathbf{r}$ is always a line.

The normal $\mathbf{a}$ perpendicular to $\mathbf{l}$ at the intersection point $\mathbf{x}$ is computed via the inner product.

$$\begin{aligned} \mathbf{a} &= \mathbf{x} \cdot \mathbf{l} && (2) \\ \hat{\mathbf{a}} &= \frac{\mathbf{a}}{\sqrt{\mathbf{a}\tilde{\mathbf{a}}}}, && (3) \end{aligned}$$

where $\tilde{\mathbf{a}}$ is the reverse of $\mathbf{a}$, *i.e.*, $\hat{\mathbf{a}}$ is the normalized $\mathbf{a}$.

To analyze light refraction according to Snell' s law, ordinarily it is essential to determine the angle between the incident ray $\mathbf{r}$ and the interface normal $\mathbf{a}$. In geometric algebra however, this angular relationship can be captured by

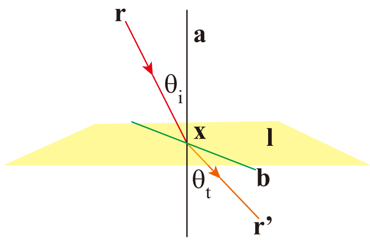

Figure 12: Snell's law

the geometric product of the normalized incident ray $\hat{\mathbf{r}}$ and the reverse of the interface normal $\tilde{\hat{\mathbf{a}}}$, thereby eliminating the need to explicitly calculate angles.

The geometric product encapsulates both the inner and outer products. Thus, taking the geometric product of the normalized incident ray and the reverse of the normalized interface normal yields a scalar part that corresponds to the cosine of the incident angle and a bivector part that corresponds to the sine.

$$\hat{\mathbf{r}}\tilde{\hat{\mathbf{a}}} = \cos\theta_\mathrm{i} + \hat{\mathbf{b}}\sin\theta_\mathrm{i}, \tag{4}$$

where $\hat{\mathbf{b}}$ is the normalized binormal, which is the perpendicular direction to the plane of incidence. In geometric terms, the reverse of the normal vector $\tilde{\hat{\mathbf{a}}}$ serves to express the incident ray in the frame defined by the interface normal. The geometric product $\hat{\mathbf{r}}\tilde{\hat{\mathbf{a}}}$ can thus be interpreted as expressing the incident ray in the coordinate frame defined by the normal vector, allowing geometric extraction of angular information. By extracting the grade-2 component, we obtain a quantity proportional to $\sin\theta_\mathrm{i}$, where θ_i is the angle of incidence. This is used to determine whether refraction or total internal reflection occurs.

$$\hat{\mathbf{b}}\sin\theta_\mathrm{i} = \left\langle \hat{\mathbf{r}}\tilde{\hat{\mathbf{a}}} \right\rangle_2 \tag{5}$$

The refracted ray can be computed by following the process in reverse. First, scaling the bivector component by the ratio of the refractive indexes n_1/n_2 leads to obtain the sine of the transmitted angle θ_t as

$$\mathbf{s}_\mathrm{t} = \hat{\mathbf{b}}\sin\theta_\mathrm{t} = \frac{n_1}{n_2}\left\langle \hat{\mathbf{r}}\tilde{\hat{\mathbf{a}}} \right\rangle_2. \tag{6}$$

The scalar component can be computed by

$$\cos\theta_\mathrm{t} = \sqrt{1 - \mathbf{s}_\mathrm{t}\tilde{\mathbf{s}_\mathrm{t}}}. \tag{7}$$

By adding them, the bireflection $\mathbf{r}'\tilde{\hat{\mathbf{a}}}$ on the other side of the interface is formed:

$$\mathbf{r}'\tilde{\hat{\mathbf{a}}} = \cos\theta_\mathrm{t} + \hat{\mathbf{b}}\sin\theta_\mathrm{t}, \tag{8}$$

where $\mathbf{r}'$ denotes the refracted ray. Finally, the refracted ray can be computed as

$$\begin{aligned} \mathbf{r}' = (\mathbf{r}'\tilde{\hat{\mathbf{a}}})\hat{\mathbf{a}} &= \left(\cos\theta_\mathrm{t} + \hat{\mathbf{b}}\sin\theta_\mathrm{t}\right)\hat{\mathbf{a}} \\ &= \left(\sqrt{1 - \mathbf{s}_\mathrm{t}\tilde{\mathbf{s}_\mathrm{t}}} + \mathbf{s}_\mathrm{t}\right)\hat{\mathbf{a}}. \end{aligned} \tag{9}$$

When $\mathbf{s}_t\tilde{\mathbf{s}_t} > 1$, indicating that no real solution exists for the refracted direction, total internal reflection occurs and the reflected ray can be computed by reflecting the incident ray about the normal.

$$\mathbf{r}' = \hat{\mathbf{a}}\hat{\mathbf{r}}\tilde{\hat{\mathbf{a}}} \tag{10}$$

Snell's law was expressed using PGA. A function was implemented with the `kingdon` GA library [3], which took as input the incident ray, the refractive interface, and the relative refractive index, and returned the refracted ray as output.

```
def snells_law(interface, ray_i, refractive_index):
    r = ray_i.normalized()
    intersection = r^interface
    normal = (intersection|interface).normalized()
    binormal_sin_i = (r/normal).grade(2)
    binormal_sin_o = (refractive_index * binormal_sin_i)
    cos_o = (1-binormal_sin_o.normsq())**.5
    rp_a = (cos_o + binormal_sin_o)
    ray_o = rp_a*normal
    return ray_o, intersection

def prism_refraction(prism, ray_i):
    ray_g, i_g = snells_law(prism[0], ray_i, r_air/r_glass)
    ray_o, i_o = snells_law(prism[1], ray_g, r_glass/r_air)
    return ray_o, (i_g,i_o)
```

Listing 1: The functions of Snell's law and refraction through wedge prism.

Figure 13 illustrates the result of applying Snell's law to a single planar interface. The visualization confirms the correctness of the geometric algebra-based refraction computation, showing both the incident and refracted rays and their interaction with the refractive plane.

A wedge prism was defined by its refractive index and three parameters for each of its two planar surfaces. It was demonstrated that the behavior of a light ray passing through the prism could be precisely described for any incident ray. For the Risley prism, two wedge prisms were defined, and their respective rotations around the optical axis were represented using rotors in PGA. It was shown that the behavior of any incident ray passing through the Risley prism could be explicitly described.

Building upon this, Fig. 14 demonstrates the behavior of a light ray passing through a Risley prism, composed of two independently rotating wedge prisms. The figure visualizes how each prism contributes to the overall beam deviation. This simulation confirms that the combined effect of the two prisms can be effectively modeled using rotors in PGA.

It is known that when two wedge prisms are rotated at a fixed angular velocity ratio, the refracted light ray forms an *epitrochoid*. An epitrochoid is the curve traced by a point on a circle rolling around another. The epitrochoid pattern arises because each rotating prism imparts a direction-dependent angular shift to the beam. When these shifts are applied in a coupled, periodic manner, the cumulative effect leads to the characteristic looping curve.

In our demonstration (Fig. 15), two sliders were used to define the angular velocity of the prisms. Based on these input, rotors were constructed and applied to each wedge prism to rotate them independently about the optical axis.

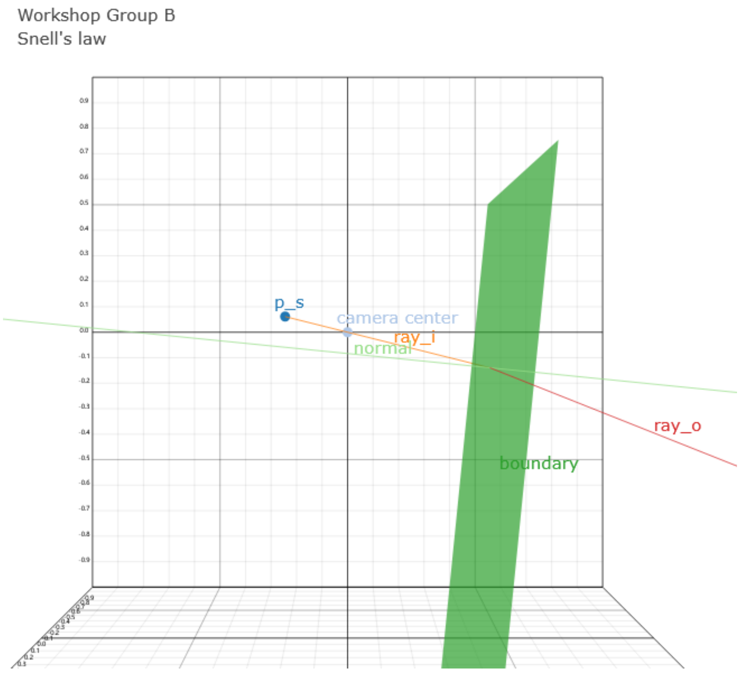

Figure 13: Visualization of Snell' s Law using Projective Geometric Algebra (PGA) in the kingdon GA library. The figure shows the incident ray, refracted ray, and the interface plane, demonstrating the PGA-based computation.

As a light ray passed through both rotating prisms, its direction was dynamically altered. By plotting the endpoint of the refracted ray at each time step, we visualized the resulting epitrochoid trajectory. Each prism shifts the beam direction, and their combined effect produces a smooth periodic curve. This demonstration shows how Risley prisms steer light and how PGA can simulate it accurately.

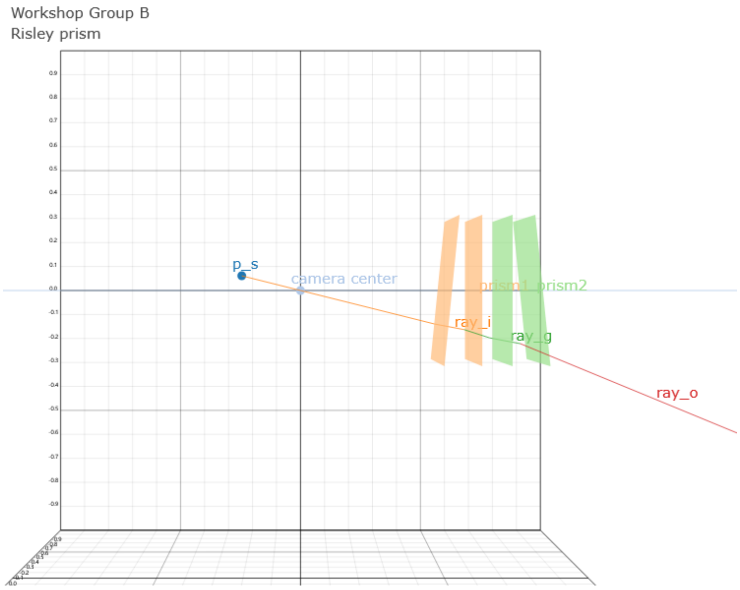

Figure 14: Simulation of light propagation through a Risley prism using kingdon GA library. Two wedge prisms rotate independently, and their combined refractions are modeled using rotors in PGA, allowing accurate and compact representation of the ray path.

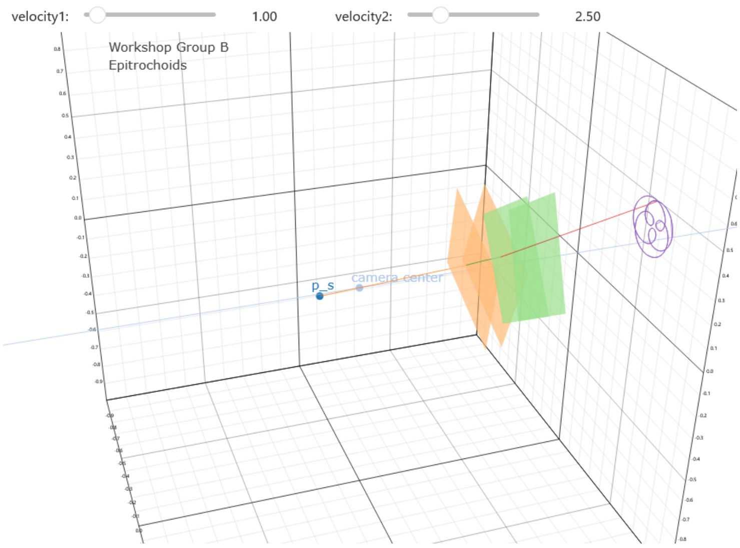

Figure 15: Visualization of the epitrochoid trajectory formed by the endpoint of a refracted light ray passing through two rotating wedge prisms. As the prisms rotate with a fixed speed ratio, the beam traces a smooth periodic curve, accurately modeled via PGA.

Project C: Man in the mirror — What is the curve that makes you see yourself embedded in the mirror?

Motivation and Background

There is a type of visual illusion called *anamorphosis*, that an image perceived through reflection by a cylindrical mirror look natural, but the original image is distorted. As it is natural to consider mirror reflection as one of the basic concepts in Geometric Algebra (GA), as explained by Martin Roelfs, GA is expected to be useful to design some visual illusion like anamorphosis.

Goal

Being inspired by anamorphosis, we consider to device a double-mirror setting with which one can see some three dimensional object in the mirror. Specifically, we pose the question if it is possible to construct a natural image of the viewer him/herself through the reflection by the cylindrical mirror.

We set the basic setting, which can be varied upon request, as follows. Suppose that there is a cylinder, whose surface is orthogonal to the horizontal plane, and the circle parallel to the horizontal plane has its center at the origin and its radius being 1 (Figure 1 below). A viewer is supposed at another circle with its center at origin and its radius being larger than 1 (outside of the cylinder depicted as the blue broken circle in Figure 1), and the one gazes at the some point, called target point, inside the cylinder, with the coordinates $(0, z)$, $-1 < z < 1$ on the horizontal plane. Then our goal is to design some curve (depicted as the red curve in Figure), that reflects the light from the viewer to the viewer' s view point via the cylindrical mirror. In this way, one can walk around the cylindrical mirror, and see oneself is in the cylindrical mirror. Moreover, changing his/her view point would change does not change the target point, where the image of viewer inside the mirror is located. This invariance of the target location would let the viewer perceive some three dimensional inside the mirror. In this sense, this setting is considered to be a new visual illusion, which is neither just anamorphosis nor double mirror (which gives only reflect planer image from one view point).

Outcome

We first consider a simplified problem ask what is the point of reflection on the cylinder for any two given points in the horizontal plane. Prof. Hidaka has solved this problem, and find the coordinate of the points of reflection on the cylinder can be roots of a sixth-degree polynomial equation. However, Prof. Ochiai reformulate the same problem and found that it can be reduced to a fourth-degree polynomial equation, not the six-degree one. We confirmed that Hidaka' s six-degree polynomial has essentially Ochiai' s fourth-degree polynomial as its factor, by resorting symbolic computation performed by Mathematica. Martin Roelfs reformulated the same problem in terms of GA, and visualize and solve it numerically using `kingdon`, his GA library developed for python.

Next, we worked the original problem to design the curve. Prof. Kaji proposes another variation of the problem that fixes the curve and let the walk path of the viewer as variable. Clément Chomicki worked to formulate and solve the

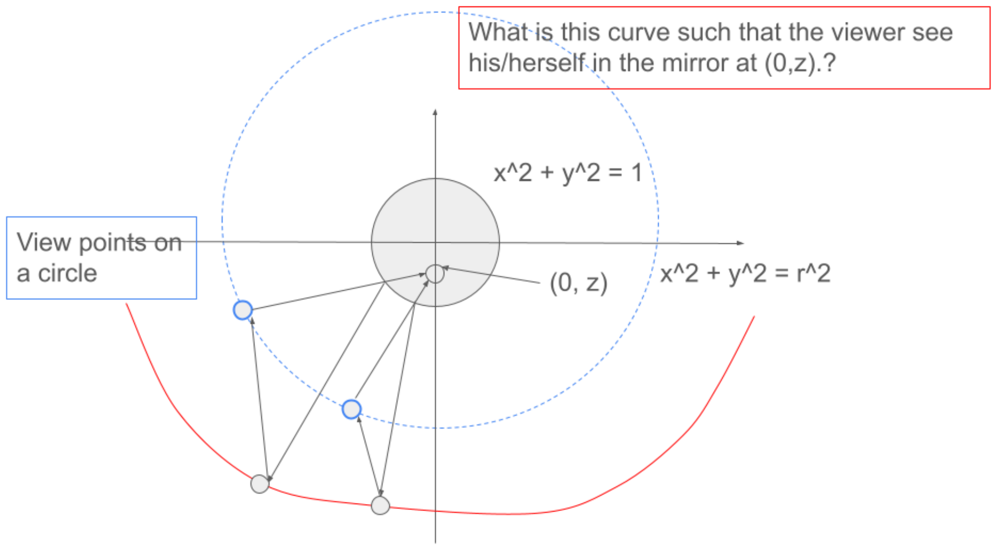

Figure 16: Cylinder-mirror anamorphosis problems.

problem by (conformal) GA. Later, Prof. Ochiai has integrates the two variations of the problem, fixed walk path vs fixed second mirror, and reformulate the problem. Further Prof. Ochiai drew a sketch describing a general solution for the integrated problem by those geometric objects represented on the complex plane (another document attached). The complex plane is equivalent to a geometric algebra $\mathbb{R}^{1,1}$, which is effective approach to the integrated problem. Although we could not reach the specific result for the original problem, we got a solid approach to the solution formulated by GA. Though this project, we have experienced and learn GA, and how it worked for specific problems.

Project D: Exploring Geometric Algebra in Machine Learning

Our group explored the possibilities of using Geometric Algebras (GA) in machine learning.

Key Property We first considered an important property:

- A geometric algebra A represents a certain symmetry group G, which is generated by reflections in all its geometric objects (e.g., planes in PGA, spheres in CGA).
- The polynomials resulting from the geometric product in this algebra are equivariant under the transformation group G.

This can be demonstrated algbraicaly. Consider A polynomial P of n variables $X_1, \ldots, X_n$ and V a versor:

$$P\left(X_1, \ldots, X_n\right) = \sum_i \prod_{j=1}^{n} X_j^{d_{i,j}} \tag{11}$$

$$P\left(VX_1V^{-1}, \ldots, VX_nV^{-1}\right) = \sum_i \prod_{j=1}^{n} \left(VX_jV^{-1}\right)^{d_{i,j}} \tag{12}$$

$$= \sum_i \prod_{j=1}^{n} \underbrace{\left(VX_jV^{-1}\right)\left(VX_jV^{-1}\right)\ldots\left(VX_jV^{-1}\right)}_{d_{i,j}\text{times}} \tag{13}$$

$$= \sum_i \prod_{j=1}^{n} \underbrace{VX_jX_j\ldots X_jV^{-1}}_{d_{i,j}\text{times}} \tag{14}$$

$$= V\left(\sum_i \prod_{j=1}^{n} X_j^{d_{i,j}}\right)V^{-1} \tag{15}$$

$$\tag{16}$$

Hence $P\left(VX_1V^{-1}, \ldots, VX_nV^{-1}\right) = VP\left(X_1, \ldots, X_n\right)V^{-1}$ which means that the polynoms of geometric a algebra are equivariant for the transformation group that this algebra is able to represent through its versors.

For more details, see [15]

This can be illustrated by the Fig. 17 . If given an equivariant function computing a contour from a point cloud. The contour can be computed after or before any rigid transformation as this transformation will be "ignored" by the function.

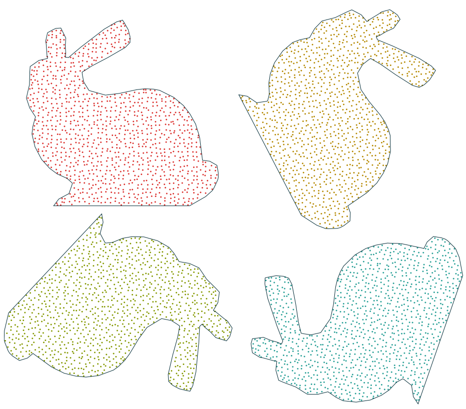

Figure 17: A neural network equivariant for the group of rigid transformation will treat these four object as identical.

Relevant Literature Several papers already explore the application of GA in neural networks. We were particularly interested in the following:

- **GATr**: a transformer architecture using PGA for its layers [1]
- **Clifford-Steerable Neural Networks** [16]
- **Clifford Group Equivariant Neural Networks** [15]

Our Project: Learning Shapes from Oriented Point Clouds
We focused on an optimization problem involving oriented point clouds: **learning a shape by learning how to displace points from a sphere to match an input point cloud.**

We explored several approaches in parallel:

- Starting with the simpler $\mathbb{G}_2$ algebra as a baseline.
- Using PGA, which naturally includes rigid transformations and is commonly used in this context.
  - Using the **Flag method**: adding a 3-vector to represent the point and a 1-vector for a plane to capture orientation. See Fig. 1 of this introduction: zenodo.org/record/15030773
- Using CGA, which supports all conformal transformations at the cost of one extra coordinate (or doubling for general multivectors).
  - CGA' s point pairs allow the representation of oriented points. See: zenodo.org/record/15043692

We started doing some experiments to test the two first items. The chosen geometric algebra model is a motion neural network since the learned elements are versors. The neural layers include a multivector layer with a multivector valued activation function, a normalisation layer that simply consists in dividing by the squared norm of the input multivector, please refer to the code as shown in Listing 2

```
class CGEBlock(nn.Module):
    def __init__(self, algebra, in_features, out_features):
        super().__init__()
        self.layers = nn.Sequential(
            MVLinear(algebra, in_features, out_features),
            MVSiLU(algebra, out_features),
            SteerableGeometricProductLayer(algebra, out_features),
            MVLayerNorm(algebra, out_features)
        )
    def forward(self, input):
        # [batch_size, in_features, 2**d] -> [batch_size,
        #                                    out_features, 2**d]
        return self.layers(input)

class CGEBlockFull(nn.Module):
    def __init__(self, algebra, in_features, hidden_features,
    out_features, num_layers=2):
        super().__init__()
        net = [
            FullyConnectedSteerableGeometricProductLayer(
                algebra, in_features, hidden_features
            ),
        ]
```

```
        for _ in range(num_layers - 1):
            net.append(MVSiLU(algebra, hidden_features,
            invariant='norm'))

        net.append(
            FullyConnectedSteerableGeometricProductLayer(
                algebra, hidden_features, out_features
            )
        )

        self.net = nn.Sequential(*net)
    def forward(self, input):
        return self.net(input)
```

Listing 2: Geometric Algebra neural block.

The forward pass merely consists in computing the sandwiching product between the input point cloud and the trained versors. The loss is computed as the MSE between the resulting output point cloud and the target point cloud as follows

$$loss = \frac{1}{N}\|(Vp_iV \quad - p_i')\|^2 \tag{17}$$

where p_i refers to the ith input point cloud and p_i' to the ith target point cloud. Briefly, the resulting training loss is 15 lower than a MLP for the same number of iteration (300). Please refer to Listing 3 for the forward pass code.

```
model = CGEMLP(ca, in_features=1, hidden_features=16, out_features
    =1, n_layers=2)

adam = optim.Adam(model.parameters(), lr=1e-3)

for i in range(1000):
    adam.zero_grad()
    transformationMultivector = model(input_cl)
    reverse_signs = torch.tensor([1.0, 1.0, 1.0, -1.0],
                                 device=transformationMultivector.
                                     device,
                                 dtype=transformationMultivector.
                                     dtype)

    reversed_transformationMultivector = transformationMultivector
        * reverse_signs
    outputTrain = ca.sandwich(transformationMultivector, input_cl,
        reversed_transformationMultivector)
    loss = F.mse_loss(outputTrain, output_cl)
    loss.backward()
    adam.step()
    if i %100== 0:
        print(f"Step: {i}. Loss: {loss.item():.12f}")
```

Listing 3: Our forward pass with a CGENN block.

Conclusion We concluded that all these methods are worth testing and comparing. We note however that the results using the simple $\mathbb{G}_2$ algebra were promising enough to suggest that Geometric Algebra can be effectively applied in neural networks. The potential computational overhead is likely to be offset by the advantages it brings.

Identified points of collaborations and future directions

- The passionate discussions during the meeting made it clear that Geometric Algebra holds significant potential to offer fresh insights and novel perspectives on a wide range of tasks in Computer Vision and Computer Graphics. We believe it can play a key role in fostering new ideas, driving innovation in both fields, and opening up new avenues of research in the future.
- In Dr. Roelfs' tutorial, we learned that efficient tools are already available and ready to be applied to real-world problems. These tools can also be integrated into deep neural network frameworks. Beyond serving as intuitive educational resources, they open a new path toward the development of future Geometric AI.

Participants of this Shonan meeting will continue their collaboration through the established communication channel (Slack). Some have already submitted proposals for international collaboration aimed at enhancing research outcomes, driving innovation, and nurturing the next generation of top-tier researchers.

References

[1] J. Brehmer, P. De Haan, S. Behrends, and T. S. Cohen, "Geometric algebra transformer," *Advances in Neural Information Processing Systems*, vol. 36, pp. 35 472–35 496, 2023.

[2] J. Brandstetter, R. v. d. Berg, M. Welling, and J. K. Gupta, "Clifford neural layers for pde modeling," *arXiv preprint arXiv:2209.04934*, 2022.

[3] M. Roelfs, "The willing kingdon clifford algebra library," 2025. [Online]. Available: https://arxiv.org/abs/2503.10451

[4] M. Keller, K. Werling, S. Shin, S. Delp, S. Pujades, C. K. Liu, and M. J. Black, "From skin to skeleton: Towards biomechanically accurate 3d digital humans," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–12, 2023.

[5] M. Keller, V. Arora, A. Dakri, S. Chandhok, J. Machann, A. Fritsche, M. J. Black, and S. Pujades, "Hit: Estimating internal human implicit tissues from the body surface," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 3480–3490.

[6] N. A. Valous, E. Hitzer, D. Duşe, R. R. Moraleda, F. Popp, M. Suarez-Carmona, A. Berthel, I. Papageorgiou, C. Fremd, A. Rölle *et al.*, "Novel computational workflows for natural and biomedical image processing based on hypercomplex algebras," *arXiv preprint arXiv:2502.07758*, 2025.

[7] S. Kaji, K. Kajiwara, and H. Park, "Linkage mechanisms governed by integrable deformations of discrete space curves," in *Nonlinear systems and their remarkable mathematical structures*. Chapman and Hall/CRC, 2019, pp. 356–381.

[8] S. Kaji, K. Kajiwara, and S. Shigetomi, "An explicit construction of kaleidocycles," *arXiv e-prints*, pp. arXiv–2308, 2023.

[9] 鍛冶静雄, "環状リンク機構・メッシュ曲面の離散微分幾何学," 日本ロボット学会誌, vol. 41, no. 6, pp. 536–540, 2023.

[10] J.-O. Lachaud, P. Romon, and B. Thibert, "Corrected curvature measures," *Discrete & Computational Geometry*, vol. 68, no. 2, pp. 477–524, 2022.

[11] J.-O. Lachaud, D. Coeurjolly, C. Labart, P. Romon, and B. Thibert, "Lightweight curvature estimation on point clouds with randomized corrected curvature measures," in *Computer Graphics Forum*, vol. 42, no. 5. Wiley Online Library, 2023, p. e14910.

[12] C. Weill-Duflos, D. Coeurjolly, and J.-O. Lachaud, "Corrected laplace–beltrami operators for digital surfaces," *Journal of Mathematical Imaging and Vision*, vol. 67, no. 2, pp. 1–16, 2025.

[13] N. Maruani, R. Klokov, M. Ovsjanikov, P. Alliez, and M. Desbrun, "Voromesh: Learning watertight surface meshes with voronoi diagrams," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 565–14 574.

[14] L. Wang, F. Hétroy-Wheeler, and E. Boyer, "A hierarchical approach for regular centroidal voronoi tessellations," *Computer Graphics Forum*, vol. 35, no. 1, pp. 152–165, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12716

[15] D. Ruhe, J. Brandstetter, and P. Forré, "Clifford group equivariant neural networks," *Advances in Neural Information Processing Systems*, vol. 36, pp. 62 922–62 990, 2023.

[16] M. Zhdanov, D. Ruhe, M. Weiler, A. Lucic, J. Brandstetter, and P. Forré, "Clifford-steerable convolutional neural networks," *arXiv preprint arXiv:2402.14730*, 2024.