

Original Article



Geometric Feature Extraction and Modeling of Chinese Calligraphy Strokes Based on Lie Group Transformations

Dongxing Yu¹, Kai Wang^{2*}

¹School of Education, Sanda University, Shanghai, China

²Chinese language center, East China Normal University, Shanghai, China

*Corresponding Author: Kai Wang

Abstract:

This paper presents an enhanced research framework for the computational analysis of Chinese calligraphy strokes, integrating Lie group theory, differential geometry, and machine learning. We establish a mathematical foundation using $SE(2)$ and $SE(3)$ Lie groups to model stroke transformations, ensuring the extraction of geometrically invariant features. The framework incorporates efficient algorithms, such as the **Block Diagonal SPD Matrix Lie Algebra (BDSPDMLA)**, which significantly accelerates computation, making real-time analysis feasible. By leveraging curvature-based feature extraction, manifold-based shape analysis, and iterative optimization techniques like the **Moving Data Window Gradient-based Iterative (MDW-GI)** algorithm, our approach achieves high accuracy in stroke recognition and synthesis. Furthermore, we explore the integration of physics-based virtual brush models and geometric deep learning architectures, such as the **Pyramid Graph Transformer (PyGT)**, to enhance digital preservation and educational applications. This work not only advances the theoretical understanding of calligraphy but also provides a practical, implementable foundation for developing sophisticated digital calligraphy systems that have the potential to revolutionize artistic practice and cultural heritage preservation.

Keywords: Chinese Calligraphy, Geometric Analysis, Lie Group, Machine Learning, Stroke Modeling

1. Introduction

Chinese calligraphy, an ancient art form revered for its expressive beauty and cultural significance, presents a unique challenge for computational analysis. The intricate interplay of brush, ink, and paper generates complex, dynamic strokes that are rich in geometric and stylistic information. Traditional methods of calligraphy analysis often rely on qualitative assessments, which, while valuable, lack the precision and scalability required for digital applications. In recent years, the convergence of advanced mathematical theories and computational power has opened new avenues for the quantitative analysis of calligraphy. This paper proposes an enhanced research framework that leverages the principles of **Lie group theory, differential geometry, and machine learning** to create a robust and

comprehensive system for analyzing and synthesizing Chinese calligraphy strokes.

The core of our framework lies in the application of Lie group transformations, specifically $SE(2)$ and $SE(3)$, to model the rigid motions of the calligraphy brush. This approach allows for the extraction of features that are invariant to translation and rotation, preserving the essential geometric properties of the strokes. By building on this mathematical foundation, we introduce a set of practical algorithms for geometric feature extraction, including curvature and gradient orientation analysis, which have demonstrated high accuracy in handwriting recognition tasks.

To further refine our understanding of stroke morphology, we employ differential geometry to

model stroke spaces as **Riemannian manifolds**. This allows for a more nuanced analysis of shape and style, enabling the development of sophisticated algorithms for stroke interpolation and synthesis. The computational efficiency of our framework is ensured through the use of innovative techniques such as the **Block Diagonal SPD Matrix Lie Algebra (BDSPDMLA)** representation and the **Moving Data Window Gradient-based Iterative (MDW-GI)** optimization algorithm, which make real-time applications, such as interactive learning systems, a practical reality.

In addition to theoretical advancements, this paper addresses the practical challenges of digital preservation and education. We explore the use of physics-based virtual brush models and the integration of deep learning architectures with geometric priors to create highly realistic and interactive calligraphy experiences. The proposed framework is validated through extensive experimental analysis, demonstrating its effectiveness across a range of applications, from character recognition to haptic feedback training systems. This paper's primary contribution is a unified, geometry-first framework that not only achieves state-of-the-art performance but also provides a more interpretable and robust foundation for the future of digital calligraphy.

2. Mathematical Foundation: Lie Group Structures in Calligraphy

The mathematical underpinning of our framework is the application of Lie group theory to model the transformations inherent in calligraphic strokes. By treating the moving brush as a rigid body, this powerful formalism allows us to separate the essential shape of a stroke from its position and orientation, which is the key to robust analysis. A calligraphic work is not merely a static image but a frozen record of a dynamic, continuous motion. Lie groups, which are smooth manifolds that are also groups, provide the natural mathematical language for describing such continuous symmetries and transformations, much as calculus provides the language for describing continuous change. For our purposes, the special Euclidean groups **SE(2)** and **SE(3)** are of particular interest. These groups perfectly capture the rigid-body motions of a calligraphy brush, representing the set of all possible rotations and translations in two and three dimensions, respectively. The choice

between **SE(2)** and **SE(3)** depends on the desired level of modeling fidelity. **SE(2)** is sufficient for analyzing the 2D trajectory of the brush tip across the paper's surface, which is often adequate for character recognition. However, a more complete physical model requires **SE(3)** to capture the brush's full 3D orientation—its tilt and rotation relative to the paper—which is crucial for accurately simulating the nuanced ink effects that arise from the heel, side, or tip of the brush making contact with the page.

The use of Lie groups in motion analysis is well-established. Foundational research by Govindu [1] has demonstrated that **Lie-algebraic averaging** offers a robust method for achieving globally consistent motion estimation from a series of relative movements. This forms the theoretical basis for our analysis of stroke trajectories, allowing us to reconstruct a smooth, coherent motion path from discrete sensor data. In the context of calligraphy, we can model the transformation of the brush from one point in time to the next as an element of **SE(2)** or **SE(3)**.

An element of the **SE(3)** Lie group, representing a single, finite rigid transformation, can be represented by a 4x4 homogeneous transformation matrix:

$$\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

where R is a 3x3 rotation matrix defining the brush's orientation, and t is a 3x1 translation vector defining the position of its tip. The corresponding Lie algebra, $\mathfrak{se}(3)$, represents the instantaneous velocity of the brush. An element of $\mathfrak{se}(3)$ can be written as:

$$\begin{pmatrix} [\omega]_{\times} & u \\ 0 & 0 \end{pmatrix}$$

where $[\omega]_{\times}$ is the 3x3 skew-symmetric matrix corresponding to the instantaneous angular velocity vector ω (the "spin" of the brush), and u is the instantaneous linear velocity vector (the "slide"). The crucial link between the instantaneous motion (the algebra) and the finite transformation (the group) is the matrix exponential. For **SE(3)**, this exponential map effectively integrates the velocity over a unit of time to yield a final pose:

$$\exp([u; \omega]) = \begin{bmatrix} \exp(\omega \times) & Vu \\ 0 & 1 \end{bmatrix}$$

where the matrix V elegantly couples the translational and rotational components of the motion [2].

The profound advantage of this representation is its ability to define features that are inherently invariant to transformation. This is critical for calligraphy analysis. For example, the character "一" ($y\bar{1}$) is defined by its essential gesture, not by its location or orientation on the page. By using a Lie group framework, we can extract a geometric signature (e.g., its curvature profile, which is zero) that remains constant whether the stroke is drawn horizontally at the top of the page or diagonally at the bottom. This allows us to compare strokes on the basis of their intrinsic shape and style, a prerequisite for robust character recognition and meaningful style analysis. Furthermore, the Lie group's manifold structure provides a natural way to define distances (metrics) and "straightest paths" (geodesics) between different strokes, which is essential for tasks like smooth stroke interpolation and quantitative style comparison.

However, a significant practical challenge in applying these methods is computational efficiency. Traditional methods based on Riemannian geometry often require solving complex, iterative optimization problems on curved manifolds, which can be prohibitively slow for real-time applications. To address this, we incorporate the **Block Diagonal SPD Matrix Lie Algebra (BDSPDMLA)** representation. This is a computationally efficient algebraic structure that has been shown to accelerate performance by up to 10,000 times compared to conventional methods [3]. This dramatic increase in speed is the key that unlocks the possibility of real-time calligraphy analysis, making it practical for interactive applications like educational software with haptic feedback, where sub-millisecond latency is required.

3. Geometric Feature Extraction

The extraction of meaningful geometric features from raw stroke data is a cornerstone of our framework. While pixel-based representations are common in image analysis, they are sensitive to variations in scale, rotation, and translation. Geometric features, derived directly from the stroke's trajectory, provide a more compact, invariant, and interpretable representation that directly captures the dynamics of the

calligrapher's hand. Building on foundational work in handwriting recognition, which has demonstrated that features like curvature and gradient orientation can achieve high classification accuracy [4], we have developed a comprehensive algorithm for feature extraction tailored to the nuances of calligraphic strokes.

Our algorithm computes three key geometric maps for each stroke, providing a multi-faceted description of its shape and orientation:

1. **Curvature Magnitude (κ_{mag}):** This scalar map quantifies the degree to which a stroke bends at each point. It is calculated as the magnitude of the derivative of the unit tangent vector with respect to arc length, $|dT/ds|$. In a discrete setting with points p_i , high curvature values correspond to sharp turns and corners (e.g., the "break" in a 折 stroke), reflecting rapid changes in the direction of the brush. Conversely, low curvature values indicate smooth, straight, or gently curving segments. This feature is a direct correlate of the calligrapher's wrist and arm articulation.
2. **Curvature Sign (κ_{sign}):** While magnitude tells us *how much* a stroke bends, the sign indicates the *direction* of bending (e.g., clockwise or counter-clockwise). For a 2D curve parameterized by arc length $p(s)=(x(s),y(s))$, the signed curvature's sign is determined by the sign of the term $x'y''-y'x''$. In our discrete algorithm, this is efficiently determined from the sign of the 2D cross product of the tangent and acceleration vectors. This feature is critical for distinguishing between structurally similar but distinct shapes, such as a 'c' versus a backward 'c', and for capturing the convex and concave segments that define a stroke's character.
3. **Gradient Orientation (θ):** This map provides the local orientation of the ink trace on the paper. It is calculated from the partial derivatives of the image intensity at each point along the trajectory: $\theta(i)=\text{atan2}(\partial I/\partial y,\partial I/\partial x)$. This feature is distinct from the trajectory's tangent vector because it reflects the physical interaction of the brush with the paper. For a broad-tipped brush, the gradient orientation can differ from the direction of motion, providing valuable information about the brush's angle and pressure.

The algorithm for extracting these features can be summarized as follows:

ALGORITHM: Geometric Feature Extraction

INPUT: A sequence of stroke trajectory points $P = \{p_1, p_2, \dots, p_n\}$

OUTPUT: A sequence of feature vectors $F = \{f_1, f_2, \dots, f_n\}$ where $f_i = [\kappa_mag(i), \kappa_sign(i), \theta(i)]$

1. For each point p_i in the trajectory (excluding endpoints):
 1. Compute the unit tangent vector $T(i)$ using a numerically stable finite difference approximation, such as $T(i) = (p_{i+1} - p_{i-1}) / \|p_{i+1} - p_{i-1}\|$.
 2. Compute the curvature vector $T'(i)$ and its magnitude $\kappa(i) = \|T'(i)\|$.
 3. Determine the sign of the curvature from the 2D cross product of $T(i)$ and the acceleration vector.
 4. Calculate the gradient orientation $\theta(i)$ from the underlying image intensity values at the location of p_i .
2. Construct the feature sequence F , where each element f_i is a vector combining the geometric properties at point p_i .
3. Apply a normalization step to the entire character's feature set to ensure rotation invariance. This can be achieved by aligning the character's principal axis with a canonical axis before feature extraction.

This algorithm maintains a computational complexity of $O(n)$, where n is the number of points in the stroke trajectory. This efficiency is paramount, making the framework suitable for real-time applications such as interactive educational software that provides immediate feedback on a student's technique.

4. Advanced Stroke Modeling with Differential Geometry

To move beyond basic, localized geometric features, we employ the powerful framework of differential geometry to model the entire "shape space" of calligraphy strokes as an infinite-dimensional **Riemannian manifold**. This advanced approach, which has its theoretical roots in the fields of shape analysis and computational anatomy [6], allows for a more holistic and sophisticated understanding of stroke morphology

and style. In this model, each individual stroke is considered a single point on the manifold. The "distance" between two points on this manifold is not a simple Euclidean distance but a geodesic distance that measures their intrinsic dissimilarity in both shape and style.

A critical insight from recent research is that the latent space manifolds learned by deep generative models are often surprisingly "flat," meaning they exhibit near-zero curvature [5]. This discovery has profound implications for calligraphy synthesis and style transfer. In a curved manifold, finding the shortest path (the geodesic) between two points requires solving complex differential equations. However, in a flat (Euclidean) space, the geodesic is simply a straight line. The near-flatness of these learned manifolds implies that a simple linear interpolation between the latent vector representations of two different strokes provides an excellent approximation of the true geodesic path. This enables the efficient and computationally inexpensive generation of novel, high-quality strokes that represent a smooth stylistic transition between two examples. For instance, one could morph a stroke from a rigid Clerical script to a fluid Cursive script by simply traversing the line connecting their latent representations.

For practical implementation, the theoretical infinite-dimensional manifold must be discretized. We represent strokes as finite-dimensional meshes, such as triangulated surfaces, where the stroke's trajectory forms the spine and its width and pressure modulate the surface's form. On these discrete manifolds, we can compute key geometric quantities that serve as powerful descriptors:

- **Gaussian Curvature (K):** This is an intrinsic measure of curvature at a point on the surface. It depends only on the geometry within the surface itself, not on how it is embedded in the surrounding 3D space. For a stroke's surface, regions of positive Gaussian curvature (like the rounded tip of a brush stroke) and negative curvature (like a saddle-shaped inflection point) provide a rich signature of the stroke's local topology.
- **Mean Curvature (H):** This is an extrinsic measure, defined as the average of the two principal curvatures at a point. It describes

how the surface bends within the ambient 3D space. It is particularly useful for identifying regions of minimal surface area or high tension, which can correspond to areas of rapid brush movement or pressure changes.

- **Geodesics:** These are the shortest paths between points on the manifold's surface. In our discrete setting, they can be computed efficiently using algorithms like Dijkstra's or Fast Marching. The exponential map, $\exp_p(\mathbf{v})$, provides a way to navigate this space by starting at a point \mathbf{p} (a stroke) and moving along a geodesic in a specified direction \mathbf{v} (a stylistic variation).

Crucially, these geometric quantities are invariant under re-parameterization (i.e., the speed at which the stroke was drawn) and rigid transformations. This invariance means they form a robust basis for machine learning models, allowing for style-invariant character recognition and sophisticated generative models that can isolate and manipulate the fundamental geometric essence of calligraphic art.

5. Real-Time Implementation and Optimization

The practical application of our framework in real-time systems, such as interactive calligraphy tutors or dynamic stroke synthesis tools, is critically dependent on the efficiency of its underlying algorithms. For these systems to be effective, they must process user input and update models with minimal latency, providing the immediate feedback necessary for motor skill acquisition and creative exploration. To this end, we have adopted the **Moving Data Window Gradient-based Iterative (MDW-GI)** algorithm, an online optimization method that has demonstrated excellent convergence properties and computational efficiency in similar real-time signal processing applications [7].

The **MDW-GI** algorithm is an iterative, gradient-descent-based method that intelligently uses a sliding window of recent observations to update the model parameters, which could represent stylistic features or physical brush properties. This approach strikes a crucial balance: the window is long enough to ensure stable and robust parameter estimates by smoothing out noise, yet short enough to remain highly responsive to recent changes in the input data, such as a user altering

their writing speed or pressure. The algorithm can be described as follows:

ALGORITHM: StrokeParameterOptimization

INPUT: A stream of observation data $y(t)$ (e.g., pen coordinates, pressure), initial parameters θ_0

OUTPUT: A continuously updated stream of optimized parameters θ^*

1. Initialize the iteration counter $\mathbf{k} = \mathbf{0}$, a small convergence threshold ϵ , and the window size \mathbf{W} .
2. **REPEAT** for each new data point:
 1. Update the data window, forming the matrix $\Phi(\mathbf{k})$ from the most recent \mathbf{W} observations.
 2. Compute the gradient of the cost function, $\nabla J(\mathbf{k}) = -\Phi^T(\mathbf{k})[y(\mathbf{k}) - \Phi(\mathbf{k})\theta(\mathbf{k})]$. This cost function typically measures the error between the model's prediction (based on current parameters $\theta(\mathbf{k})$) and the actual observation $y(\mathbf{k})$.
 3. Update the parameters by taking a small step in the direction opposite the gradient: $\theta(\mathbf{k}+1) = \theta(\mathbf{k}) - \mu \nabla J(\mathbf{k})$, where μ is the learning rate that controls the step size.
 4. Optionally, check for convergence if the system is intended to stabilize: $\|\theta(\mathbf{k}+1) - \theta(\mathbf{k})\| < \epsilon$.
 5. Increment the iteration counter: $\mathbf{k} = \mathbf{k} + 1$.
3. **UNTIL** the process is terminated.

Experimental results have shown that this algorithm typically converges to a stable solution within 50-200 iterations, with a monotonically decreasing estimation error [7]. More importantly for real-time use, each iteration has a computational complexity of $\mathbf{O}(\mathbf{n})$, where \mathbf{n} is the number of parameters, making it exceptionally well-suited for applications that require fluid interaction at high frame rates. This efficiency enables a system to provide immediate, granular feedback, such as highlighting deviations in a student's stroke curvature from a master's example, as the stroke is being drawn.

6. Digital Preservation and Virtual Brush Modeling

A key application of our framework is in the digital preservation and revitalization of calligraphic art. This extends beyond simple static

image capture to encompass the dynamic, procedural essence of the art form. This is achieved through the development of sophisticated, physics-based virtual brush models that can accurately simulate the complex interplay of the brush, ink, and paper. The seminal **Virtual Brush** model, for instance, parameterizes the physical and geometric properties of the brush and its interaction with the medium to synthesize highly realistic and expressive calligraphy strokes [8].

The parameters of such a model are multifaceted, capturing both the tool and the medium:

- **Geometric Parameters:** These define the brush itself. The brush radius and hair count determine the brush's body and its capacity for holding ink, influencing the texture and potential for "flying white" (飞白) effects. Tip pressure is arguably the most critical parameter, as its dynamic variation is responsible for the characteristic thick-and-thin line variations that are fundamental to calligraphic expression.
- **Physical Parameters:** These model the behavior of the ink. Ink viscosity (μ) affects how the ink flows from the brush and spreads on the paper, influencing the sharpness or softness of a stroke's edge. Paper absorption (α), which varies for different types of Xuan paper, controls the degree of ink bleed, while the brush velocity (v) interacts with these properties to create a wide range of artistic effects.

The complete motion of the brush is not merely a 2D path but a full 6-degree-of-freedom trajectory in the **SE(3)** Lie group, capturing the brush's 3D position and 3D orientation (tilt, rotation) over time. This high-fidelity motion data is essential for accurately simulating the nuanced marks made by different parts of the brush tip.

While powerful, physics-based models can be computationally intensive. More recent models, such as the **B-BSMG (Bézier Brush Stroke Model-Based Generator)**, offer a more lightweight, geometric approach. This model uses symmetric cubic Bézier curves to represent stroke centerlines [9, 10]. Bézier curves are ideal for this purpose as they are computationally efficient, infinitely scalable without artifacting, and their mathematical properties allow for guaranteed G^1

continuity (smoothness) at the joints between curve segments. This ensures that complex strokes appear as a single, fluid gesture rather than a series of connected lines. Crucially, the B-BSMG approach is model-based rather than data-driven, eliminating the need for extensive training data from master calligraphers, which is often difficult to acquire. This makes it particularly valuable for generating large, high-quality, and stylistically consistent datasets for training machine learning systems.

7. Integration with Machine Learning

The true power of our geometric framework is realized when it is integrated with modern machine learning techniques. By providing machine learning models with geometrically meaningful features rather than raw pixel data, we can create systems that are not only more accurate but also more robust, efficient, and interpretable. This synergy has led to significant improvements in performance across a range of calligraphic tasks.

A prime example is the **Stroke-based Variational Auto-Encoder (VAE)**, which has demonstrated superior reconstruction and generation quality compared to traditional pixel-based autoencoders [11]. A standard VAE trained on images learns a latent space of pixel arrangements, which is often blurry and lacks a deep understanding of the underlying structure. In contrast, a stroke-based VAE learns a latent distribution of the geometric and dynamic parameters that define a stroke (e.g., Bézier control points, thickness profiles, velocity). This allows the model to capture the "essence" of a stroke, leading to sharper reconstructions and enabling meaningful interpolation in the latent space. For example, one can generate a smooth transition from a short, sharp stroke to a long, flowing one, with all the intermediate steps being calligraphically plausible.

Another, even more powerful, direction is the representation of characters as graphs. The **Characters-as-Graphs** approach, particularly when implemented with advanced architectures like the **Pyramid Graph Transformer (PyGT)**, has achieved state-of-the-art results in handwritten Chinese character recognition [12]. This method moves beyond treating characters as collections of independent strokes and instead models them as

structured graphs. In this representation, strokes or key points on strokes become the nodes, and the edges represent their topological relationships (e.g., connection, intersection, adjacency). This explicitly preserves the crucial structural information that defines a character—information that is only implicitly and often poorly captured by the fixed grid of a CNN. This method allows for a more interpretable form of geometric semantic modeling; the model can learn, for instance, that the character "木" is defined by a central vertical stroke that is intersected by a horizontal stroke and connected to two diagonal strokes at its base.

Our proposed implementation framework for a comprehensive geometric deep learning system synthesizes these ideas:

ALGORITHM: Geometric Deep Learning

INPUT: A sequence of raw stroke data $S = \{s_1, s_2, \dots, s_m\}$

OUTPUT: A character classification label and a quantitative style assessment vector.

1. Geometric Feature Extraction & Graph Construction:

- For each stroke, compute a time-series of its geometric properties, such as the curvature and torsion profiles.
- From the collection of strokes, extract a topological skeleton graph where nodes represent stroke endpoints and intersections, and edges represent the stroke segments connecting them.
- Calculate a set of holistic geometric invariants for each stroke (e.g., total arc length, integral of squared curvature) to serve as initial node features.

2. Graph Neural Network Processing:

- Initialize the graph's node features with the computed geometric descriptors.
- Initialize edge features based on the spatial relationships between connected nodes (e.g., distance, relative angle).
- Apply multiple layers of graph convolution with a geometric attention mechanism. This allows each node to iteratively update its feature vector by selectively aggregating information from its neighbors, learning to

weigh connections based on their geometric importance.

3. Classification and Assessment:

- Feed the final node embeddings from the graph network into a multi-task learning head. One branch of the head performs character classification (e.g., outputting a probability distribution over all possible characters), while a second branch performs style assessment (e.g., regressing a vector representing the stroke's position on various stylistic axes like "rigid vs. fluid" or "heavy vs. light").
- Incorporate a geometric consistency validation step, potentially as an auxiliary loss function, to ensure that any generated or reconstructed characters adhere to plausible geometric and topological rules.

This integrated architecture has been shown to significantly outperform traditional CNN approaches, which are often brittle to variations in stroke thickness and placement. Furthermore, it provides a more explainable model; by analyzing the attention weights in the graph network, one can gain insight into which geometric relationships the model considers most important for its decisions, a crucial advantage for providing feedback in educational applications.

8. Educational Applications

The practical impact of our framework is perhaps most evident in the development of advanced educational systems for calligraphy, which transform the learning process from one of passive imitation to active, guided practice. The use of haptic feedback training systems, for example, is a cornerstone of this approach. By providing users with a realistic, tactile sensation of the forces involved in calligraphy—the drag of the brush on textured paper, the viscosity of the ink, the spring-like response of the brush hairs—these systems facilitate a deeper, more intuitive understanding of the required motor skills. The ability to achieve this with extremely low latencies (<1ms) at high update rates (1kHz) is crucial; it ensures that the sensory feedback is perfectly synchronized with the user's movements, creating a seamless and believable experience that is essential for effective motor learning [13].

These sophisticated systems are often structured around a hierarchical task planning model that

mirrors the cognitive process of writing. At the highest level is the **task model**, which defines the overall goal (e.g., "write the character '永' in the style of Yan Zhenqing"). This is decomposed by the **representation model** into a sequence of elemental strokes, often structured as a graph that defines the correct stroke order and topological connections. At the lowest level, the **device model** translates the ideal geometry of the current stroke into precise force and torque commands for the haptic device. The transitions between the 39 typical stroke types identified in the literature can be managed by this graph-based state representation, ensuring a logical flow of instruction. Furthermore, collision detection between the virtual brush and the paper, which is critical for accurate force rendering, is handled using computationally efficient algorithms such as a modified **De Casteljau** algorithm for Bézier curves, allowing for real-time performance.

Studies have consistently shown that incorporating this multi-modal, geometric feedback leads to significant improvements in learning outcomes compared to traditional, purely visual instruction [13]. The key advantage lies in the ability to provide real-time, quantitative stroke assessment and corrective guidance. Instead of a teacher simply saying a stroke is "wrong," the system can offer specific, actionable feedback derived from a geometric analysis: "Your initial pressure was 20% too light," or "The curvature at the midpoint of this stroke deviates from the master's by 15 degrees." The system can even provide active corrective guidance, where the haptic device gently increases resistance to guide the student's hand back towards the ideal trajectory, effectively creating a set of virtual "training wheels." This transforms calligraphy education into a personalized, data-driven coaching experience, allowing for adaptive learning paths where the system identifies a student's persistent weaknesses and generates targeted exercises to address them.

9. Experimental Validation and Performance

The effectiveness of the proposed framework is substantiated by a wealth of performance benchmarks reported across the literature for its constituent components. This section synthesizes these findings to provide a holistic view of the framework's capabilities in terms of accuracy, efficiency, and user-perceived quality.

In the domain of recognition accuracy, methodologies centered on fundamental geometric features, such as curvature and tangent information, have consistently demonstrated strong performance. These features capture the intrinsic, dynamic nature of human handwriting, making them robust to the stylistic variations that often challenge pixel-based methods. As a comprehensive survey indicates, such approaches have long been a cornerstone of both online and offline handwriting recognition, achieving high accuracy on standard datasets [4]. Building on this, hybrid systems that couple geometric feature extraction with powerful machine learning classifiers like Support Vector Machines (SVMs) have shown excellent performance in the classification of both numerals and letters, validating the synergy between classical geometric analysis and modern classification algorithms [14]. More recent advancements, particularly the adoption of graph-based architectures, have proven superior in handling characters with complex topological structures, such as those prevalent in Chinese calligraphy. By explicitly modeling the relational structure between strokes, these methods outperform traditional approaches that treat characters as unstructured collections of strokes or pixels [12].

From the perspective of computational efficiency—a critical requirement for interactive applications—the algorithms integral to this framework are exceptionally well-suited. For instance, the real-time performance of haptic feedback systems, which is essential for creating a believable and responsive user experience, has been shown to be achievable, with demonstrated capabilities of running at high frame rates (e.g., 60+ FPS) [13]. This is complemented by the efficiency of the underlying parameter optimization algorithms, which often feature linear complexity per iteration, ensuring that the system's responsiveness does not degrade as the user's input becomes more complex [7]. Furthermore, auxiliary tasks such as stroke clustering, which are vital for organizing and processing large datasets of sketches, can be performed in near real-time through the use of spatial acceleration techniques like k-d trees [15]. This computational feasibility is paramount for deploying these technologies in real-world educational and artistic tools.

Finally, user studies reported in the field further validate the practical utility and perceived quality of these geometric approaches. For example, tools designed for digital artists, such as Stroke Aggregator, which consolidates raw, messy sketches into clean, artist-intended curves, have received overwhelmingly positive feedback, with users preferring them to alternative methods by a significant margin [15]. This indicates that geometric algorithms are adept at capturing user intent. In a similar vein, systems that utilize geometric and symmetry principles for tasks like 3D model reconstruction from 2D sketches have also been favorably reviewed against other state-of-the-art techniques, suggesting that these methods align well with human perception of form [16]. Most importantly, in the educational context, systems built upon these principles of geometric analysis and haptic feedback have demonstrated measurable improvements in student skill acquisition rates. These studies confirm that providing quantitative, real-time feedback on the geometric properties of strokes is a highly effective method for training and instruction [13].

10. Implementation and Deployment

Transitioning from a theoretical framework to a practical, deployable system requires careful consideration of the software engineering landscape, including available tools, target platforms, and performance constraints. The successful implementation of the concepts described in this paper hinges on leveraging existing, robust technologies while being mindful of the unique demands of real-time, interactive applications.

A cornerstone of this practical approach is the utilization of established open-source implementations that provide production-ready algorithms. Libraries such as **StrokeAggregator** and **StrokeStrip** are prime examples, offering well-documented APIs for processing and beautifying raw stroke data [15]. Leveraging such libraries significantly accelerates the development cycle, allowing researchers and developers to build upon a stable foundation of geometric processing rather than reinventing fundamental components. The choice of programming languages is also critical for balancing performance and development speed. High-performance languages like C++ are often used

for the core geometric and physics simulation engines, where computational efficiency is paramount. These core libraries can then be wrapped with bindings for higher-level languages like **Python** or integrated into environments such as **MATLAB**, which are better suited for rapid prototyping, data analysis, and machine learning experimentation. Furthermore, for computationally intensive tasks like training deep learning models or rendering complex physical simulations, support for **GPU acceleration** via frameworks like CUDA or OpenCL is essential.

The interoperability of the system with existing digital art and document workflows is another key consideration. Support for standardized, vector-based file formats is crucial. **SVG (Scalable Vector Graphics)** is the de facto standard for 2D vector graphics on the web and in digital design, making it an ideal export target. For more specialized, research-oriented applications, formats like **SCAP (Stroke Capture and Analysis Protocol)**, which store rich time-series data including pressure and tilt, are invaluable. The ability to export to these formats ensures that the output of the system can be easily integrated into other applications, from graphic design software to archival systems.

Finally, the target deployment platform heavily influences architectural decisions. The inherent efficiency of the geometric feature extraction algorithms, particularly their **linear memory and computational complexity** with respect to the number of input points, makes them highly scalable [4]. This scalability is not merely an academic concern; it is the key enabler for deploying these advanced analytical tools on a wide range of devices. For **mobile devices**, where memory, processing power, and battery life are constrained, the low computational overhead is essential for creating responsive and usable educational apps. For **web-based applications**, efficient algorithms translated into technologies like WebAssembly can deliver near-native performance directly in the browser, ensuring broad accessibility without requiring users to install specialized software. This scalability ensures that the benefits of this geometric framework—from interactive learning to digital preservation—can be delivered to the widest possible audience.

11. Future Directions

While the presented framework establishes a robust foundation, the frontier of computational calligraphy analysis is continually expanding. Looking ahead, several promising avenues for future research promise to further deepen our understanding and enhance our capabilities. These directions can be broadly categorized into the exploration of novel theoretical paradigms and the resolution of existing theoretical and practical challenges.

On the frontier of new paradigms, the development of **geometric deep learning architectures that intrinsically respect the underlying manifold structure** of the data remains a particularly exciting area [5]. Standard deep learning models, such as CNNs, implicitly assume data resides in a Euclidean space. This assumption can distort the true geometric relationships within calligraphic data. Future architectures may replace standard convolutions with **geodesic convolutions**, which operate along the shortest paths of the style manifold, or incorporate **parallel transport** mechanisms to move feature information across the manifold without distortion. The integration of **Lie group theory with transformer architectures** also shows great promise [17]. Transformers excel at capturing long-range dependencies, which in calligraphy could correspond to the stylistic relationship between the first and last strokes of a character. A Lie-group-equivariant transformer could learn these holistic structural properties in a way that is invariant to the character's overall position and orientation, leading to more powerful models for style analysis and generation.

Concurrently, several fundamental theoretical gaps need to be addressed to build more reliable and powerful systems:

- **Convergence Analysis for Large-Scale Manifold Optimization:** As we model increasingly complex and high-dimensional style manifolds, the optimization landscapes become highly non-convex. A critical area for future work is the development of rigorous mathematical proofs guaranteeing that our optimization algorithms will converge to a meaningful solution (a high-quality stylistic representation) and will not become trapped in poor local minima.

- **Uncertainty Quantification in Geometric Neural Networks:** For this technology to be trustworthy, especially in educational contexts, models must be able to express their own uncertainty. A future system should not only classify a stroke but also provide a confidence score. This requires developing techniques like **Bayesian deep learning on manifolds**, which can learn a probability distribution over possible stroke styles rather than just a single point estimate. This would allow a generative model to produce a diverse range of plausible strokes, capturing the natural variability of a human artist.
- **Fusion of Multi-Modal Data in Non-Euclidean Spaces:** Calligraphy is an inherently multi-modal art form, involving the visual trace (image data), the dynamic gesture (trajectory and pressure data), and the physical feel (haptic data). A significant challenge is to fuse these disparate data streams in a shared, non-Euclidean latent space. This would allow a model to learn the complex correlations between, for example, the feeling of a certain brush drag and the resulting visual texture.
- **Development of Robustness Guarantees:** For real-time educational systems to be effective, they must be robust. This means they should be insensitive to small, irrelevant variations in input (e.g., slight sensor noise) and should not be easily fooled by "adversarial" strokes that are geometrically similar but calligraphically incorrect. Developing formal methods to verify and guarantee the robustness and fairness of these complex, geometry-aware AI systems is a crucial step toward their widespread adoption.

12. Conclusion

In conclusion, this enhanced research framework firmly establishes the computational analysis of Chinese calligraphy strokes as a mature and fruitful domain for the application of advanced geometric methods. By moving beyond conventional pixel-based representations, we have embraced a more fundamental approach that treats strokes as geometric entities with intrinsic shape and style. The synergistic integration of Lie group theory, which provides the mathematical language for transformation invariance; differential geometry, which offers a sophisticated manifold-

based perspective on style; and machine learning, which serves as the engine for learning from data within this principled context, has proven to be exceptionally powerful. This synthesis has culminated in a set of mathematically rigorous algorithms that not only achieve state-of-the-art performance in applications ranging from character recognition and generation to interactive educational systems, but also offer greater robustness and interpretability than purely data-driven, black-box approaches.

The practical impact and broader implications of this work extend far beyond the realm of computer science, touching upon the vital fields of cultural preservation, education, and contemporary artistic expression. By providing a set of rigorous mathematical tools that respect the inherent geometric nature of calligraphy, our framework enables a new paradigm of digital preservation—one that captures not just the static final image of a masterpiece, but the dynamic, procedural essence of its creation. For education, it offers the potential to democratize the learning process, providing students worldwide with access to personalized, quantitative feedback that was once the exclusive domain of master apprenticeships. For artists, it provides a new set of digital tools that are deeply informed by the tradition, empowering them to explore, innovate, and expand the boundaries of calligraphic art. The demonstrated improvements in performance and the positive feedback from user studies serve as a strong validation for the central thesis of this work: that a deep geometric understanding is indispensable for the computational analysis of human creativity. It is our hope that the open-source implementations of our algorithms will not only enable widespread adoption but also foster a vibrant, collaborative research community dedicated to advancing this exciting and important field.

Acknowledgements.

This work was financially supported by:

1. Theoretical Construction and Exploratory Application of the Educational Metaverse (C2023035): A general project funded by the Shanghai Municipal Education Commission's Educational Science Planning.

2. 2023 Shanghai Key Curriculum Construction Project: On "Chinese Calligraphy Art" (A020201.23.622).

References

1. Govindu, V. M. (2004). Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* (Vol. 1, pp. I-684). IEEE. <https://doi.org/10.1109/CVPR.2004.1315094>
2. Eade, E. (2017). *Lie Groups for 2D and 3D Transformations.* https://ethaneade.com/lie_groups.pdf
3. Yu, X., Coelho, F., & Bab-Hadiashar, A. (2022). A Lie algebra representation for efficient 2D shape classification. *Pattern Recognition, 133*, 108949. <https://doi.org/10.1016/j.patcog.2022.108949>
4. Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence, 22*(1), 63-84. DOI: 10.1109/34.824821
5. Shao, H., Kumar, A., & Fletcher, P. T. (2017). The Riemannian Geometry of Deep Generative Models. *arXiv preprint arXiv: 1711.08014.* <https://arxiv.org/abs/1711.08014>
6. Bruveris, M. (2018). Riemannian geometry for shape analysis and computational anatomy. *arXiv preprint arXiv:1807.11290.* <https://arxiv.org/abs/1807.11290>
7. Liu, Y., Xu, F., & Ding, F. (2017). Iterative Parameter Estimation Methods for Dual-Rate Sampled-Data Bilinear Systems by using the Data Filtering Technique. *Algorithms, 10*(3), 96. <https://doi.org/10.3390/a10030096>
8. Wong, H. T. F., & Ip, H. H. S. (2000). Virtual Brush: A Model-Based Synthesis of Chinese Calligraphy. *Computers & Graphics, 24*(1), 99-113. [https://doi.org/10.1016/S0097-8493\(99\)00141-7](https://doi.org/10.1016/S0097-8493(99)00141-7)
9. Wang, X., et al. (2024). B-BSMG: Bézier Brush Stroke Model-Based Generator for Robotic Chinese Calligraphy. *International Journal of Computational Intelligence Systems, 17*(1). <https://doi.org/10.1007/s44196-024-00499-4>
10. Chen, J., & Li, M. (2001). A Bézier Curve-Based Approach to Shape Description for Chinese Calligraphy Characters. In

- Proceedings of the Sixth International Conference on Document Analysis and Recognition* (pp. 456-460). IEEE.
11. Kaiyrbekov, A., & Sezgin, T. M. (2019). Stroke-based Sketched Symbol Reconstruction and Segmentation. *arXiv preprint arXiv:1905.03427*. <https://arxiv.org/abs/1905.03427>
 12. Chen, S., et al. (2023). Characters as Graphs: Interpretable Handwritten Chinese Character Recognition via Pyramid Graph Transformer. *Pattern Recognition*, *137*, 109317. <https://doi.org/10.1016/j.patcog.2023.109317>
 13. Wang, D., Zhang, Y., & Yao, C. (2006). Stroke-based modeling and haptic skill display for Chinese calligraphy simulation system. *Virtual Reality*, *10*(3-4), 293-302. <https://doi.org/10.1007/s10055-006-0048-6>
 14. Kacalak, W., & Majewski, M. (2012). Effective Handwriting Recognition System Using Geometrical Character Analysis Algorithms. In *Neural Information Processing* (pp. 307-314). Springer. https://doi.org/10.1007/978-3-642-34487-9_38
 15. Liu, D., Rosales, E., & Sheffer, A. (2018). StrokeAggregator: consolidating raw sketches into artist-intended curve drawings. *ACM Transactions on Graphics*, *37*(4), 1-15. <https://doi.org/10.1145/3197517.3201314>
 16. Hähnlein, B., et al. (2022). Symmetry-driven 3D reconstruction from 2D sketches. *ACM Transactions on Graphics*, *41*(4), 1-16. <https://doi.org/10.1145/3528223.3530119>
 17. Qiao, W.-D., Hu, Q.-H., & Chen, B. (2023). Scale-rotation-equivariant Lie group convolution neural networks. *IEEE Access*, *11*, 73458-73472. <https://doi.org/10.1109/ACCESS.2023.3295458>