ChatGPT

# Mapping Symbolic Herb Attributes to Geometric Signatures

## Overview and Objectives

We seek to formalize how **symbolic herb attributes** can be translated into distinct **geometric visual signatures**. The goal is a design that represents each herb's properties (e.g. *action, element, intensity, power, time*) as components of a visual glyph. This glyph system must work for a flat 2D UI initially, while being extensible to AR/light-field displays. Key requirements include real-time interactivity (users can tap petals, rings, nodes to filter or select herbs) and scalability from single-herb focus to multi-herb comparison overlays. Below, we survey relevant geometric encoding techniques, propose a primary mapping specification, outline design constraints for legibility (especially in overlays), and present alternative encoding designs with pros and cons.

## Survey of Geometric Encoding Techniques

To represent multi-attribute data like herb properties, various geometric visualization techniques can be leveraged:

- **Radial/Polar Plots:** Radial layouts map data onto a circle or angular coordinates. Examples include radar charts (star plots) and rose diagrams. Radial encodings naturally use **angle** and **radius** as visual channels – two of the most effective quantitative channels (comparable to position/length on Cartesian plots) [1]. A circle's inherent cyclic nature is advantageous for representing periodic data (e.g. seasonal **time** cycles) [1]. Radial plots also spread information in all directions, making efficient use of space. However, they can become cluttered if too many variables or segments are crowded in the circle, and interpreting angles or areas can be less precise than linear scales [2].

- **Phyllotaxis (Spiral) Arrangements:** Phyllotactic patterns (such as the sunflower seed spiral based on the golden angle) offer an **even, non-overlapping distribution** of points [3]. This makes them useful for laying out numerous elements without collision. In data viz, phyllotaxis has been used to place markers or sub-glyphs in a spiral that fills space optimally even if the total number of items is unknown in advance [3]. The even spacing minimizes overlap and visual clutter, a property useful for real-time overlays of many items. For example, researchers have used **fractal phyllotaxis** layouts in 3D to ensure even spacing of objects, allowing viewers to navigate dense collections without confusion [4]. The spiral layout can also encode sequences or time naturally (moving outward along the spiral for chronological data). A potential drawback is that interpreting precise values from spiral position is harder than from radial or Cartesian positions, and it may require a legend or interaction to decode positions.

- **Toroidal and 3D Grids:** Moving to three dimensions, a **toroidal** layout (doughnut shape or torus grid) provides a continuous wrap-around surface. A torus can represent two cyclical dimensions (one around the ring, another around the tube), which could map well to periodic attributes like time cycles. Moreover, using a 3D shape like a torus as a glyph allows multiple shape parameters to vary. For instance, an *ellipsoid* glyph might offer 3 degrees of freedom (e.g. radii along axes) while a *toroid* (ring) glyph offers **4+ degrees of freedom** because one can vary the torus' major radius, minor radius (tube thickness), etc. [5]. These extra parameters mean more data variables can be encoded in one glyph. Additionally, a torus glyph viewed in AR could use depth and rotation to present information (e.g. different layers or sides for different data aspects). The challenges are ensuring the 3D shape remains legible in 2D projections and not overwhelming users – decoding a 3D glyph may require interaction (rotation or stereoscopic/AR view) and careful design of color/lighting to discern depth.

- **Glyph-Based Systems:** Glyphs are composite symbols whose visual features represent multiple data attributes [6]. Common glyph designs include star glyphs (radar charts), Chernoff faces (facial features encode data), and various custom shapes. The strength of glyphs is the ability to present **multivariate patterns** at a glance – humans can often spot patterns or outliers in a collection of glyphs by holistic shape [7] [8]. Effective glyph design uses multiple **visual channels** (position, shape, size, color, orientation, texture, etc.) to encode different variables [9]. However, not all channels are equally effective perceptually. Studies have shown that, for example, an attribute encoded by **color hue** may "pop out" faster to the eye than one encoded by subtle shape changes or orientation [10]. Likewise, quantitative judgments are most accurate with position/length, followed by angle or area, with color intensity and curvature being less precise [2]. Another guideline is to leverage **semantic metaphors** – mapping data in a way that intuitively aligns with its meaning [11]. For instance, using a heat-related color (red) to encode a "fire" element, or using an upward-pointing shape to indicate a "rising" action. Glyph systems must also consider user learning: a memorable, metaphor-rich encoding is easier to learn and recall [11]. On the flip side, a highly stylized or complex glyph might require training and could overwhelm if too many variables are encoded (channel capacity and separability are limiting factors [12] [13]).

**Takeaway:** A combination of these techniques can inform our design. Radial "flower" layouts provide a structured, at-a-glance shape; phyllotactic patterns help avoid overlaps for many items; 3D/toroidal forms allow richer encoding for AR; and general glyph design principles guide effective mapping of variables to visual features.

## Proposed Mapping Specification: Flower Glyph Design

For the primary visualization, we propose a **flower-like radial glyph** for each herb. This design draws on the metaphor of herbs as natural objects (flowers) and uses a radial symmetry for clarity. The mapping specification is as follows:

- **Petals for Actions:** Each distinct *action* of the herb (e.g. medicinal or magical actions) is represented by a petal shape. The number of petals or their shapes correspond to the herb's action categories. For example, if an herb has four key actions, its glyph will have four petals arranged circularly. Petal *shape* or size might further indicate the type or strength of the action (e.g. a pointed petal for a stimulative action versus a rounded petal for a calming action). Using separate petals inherently groups related properties and makes them easier to distinguish than if multiple variables were overlaid on one shape [14]. The petals are evenly spaced around a circle to maintain visual balance and symmetry, aiding recognition and aesthetic appeal.

- **Hue for Element:** The herb's classical *element* association is encoded as the **color hue** of the glyph (or of each petal). For instance, an herb aligned with the Water element might be blue, Fire red, Earth green/brown, Air yellow, etc. Hue is a nominal channel well-suited to categorical data like elements. We choose highly discriminable colors for each element to ensure quick identification. Color is a powerful pre-attentive feature – viewers will quickly notice color differences [10] – so this mapping lets element stand out immediately. (In an AR context, we will ensure the color choices remain visible against varying backgrounds by adding outlines or adjusting brightness as needed.)

- **Concentric Rings for Intensity:** The *intensity* or potency level of the herb is shown by **concentric rings** at the center of the glyph (like growth rings or an inner halo). For example, a mild herb might have one ring, a very potent herb three rings. Ring count or radius is an ordinal visual feature that gives a sense of "more vs less" easily. Rings are stacked without overlap and with sufficient contrast. Using radius/size here aligns with the quantitative nature of intensity (size is an ordered channel, suitable for ordinal or ratio data [12]). One design consideration is to normalize how rings scale so that adding petals (for actions) doesn't physically enlarge the glyph and mislead the viewer; the rings can be drawn within a fixed radius regardless of petal count to keep this channel independent [13].

- **Node Markers for Power:** The herb's *power* attribute (perhaps a measure of efficacy or magical power) is represented by **node icons or dots** on the glyph. These nodes could be placed at the petal tips or around the circumference. For instance, a herb of power level 5 might show five small node circles around the flower (like sparkles around a flower). If power is linked to specific actions, the nodes could attach to the corresponding petals; otherwise, they can form a ring of dots encircling the glyph. This numeric count encoding is straightforward for users to count at a glance when numbers are small, though larger counts might need grouping. The nodes add a secondary visual layer that does not interfere with the petal shapes, preserving channel separability (since shape and count are independent cues).

- **Dash Pattern for Time:** The *time* aspect (e.g. the duration of effect, or the season/time of day the herb is used) is encoded by a **stroke or outline pattern**. For example, a short-acting herb might have a dashed outline around each petal, whereas a long-acting one has a solid outline. Alternatively, if representing seasonal availability, different dash patterns (dot, dash, dash-dot) can indicate categories like annual, perennial, etc. Line pattern is a relatively subtle channel, used here because time is an important attribute but should not overpower the more central attributes. A dashed border is easily noticeable upon inspection and can be explained via a legend. We ensure the patterns have clear differences and are thick enough to be seen in both 2D and AR displays (dashing might be augmented with slight glow in AR to remain visible against real backgrounds).

**Interaction:** In this design, each part of the glyph is interactive. A user can click a petal to focus on that action (filtering herbs with that action), click a ring to adjust an intensity range filter, or select node markers to see details on power. The **center of the flower** could also be a hotspot showing the herb's name or image when hovered. This design supports comparison: multiple herb-glyphs can be drawn side by side or semi-transparently overlaid. For overlay, we might draw one glyph in outline mode behind another in solid mode, allowing the user to see differences (e.g. petal count differences or color differences immediately indicate differing actions or elements). The flower glyph metaphor provides a **natural mapping** for users (herbs as flowers), which can aid memorability and intuitive interpretation [11].

## Design Constraints for Legibility and Overlap

In real-time, dynamic overlays (especially in AR or mixed reality), maintaining **legibility** is crucial. We outline key constraints and solutions:

- **Avoid Glyph Overlap:** When multiple herb glyphs are displayed (e.g. a blend of herbs or comparison of two plants), overlapping symbols can confuse users. We must ensure glyphs do not unintentionally sit on top of each other. A spacing algorithm (potentially using phyllotactic placement for multiple glyphs) can arrange multiple glyphs in a non-overlapping layout [3]. If glyphs are tied to physical objects (in AR, hovering over actual herbs), we will implement label/glyph placement rules to avoid collisions. Prior research on AR annotation emphasizes minimizing overlap and occlusion of labels [15] – similarly, our system could dynamically offset glyphs or use leader lines if many herbs in view would otherwise collide. In a flat UI, if multiple glyphs must overlap (e.g. to compare), we can use transparency or outline modes to distinguish them and perhaps interactive "explode" on selection (clicking one glyph could separate it in space for clarity).

- **Maintain Visual Separation:** Within a single glyph, each visual channel should be distinct and not mask another. For example, petal shapes should remain recognizable even with multiple rings drawn behind them. We set a **hierarchy of visual prominence**: the user's eye should catch color and petal count/shape first (high-level info), then see rings and nodes upon closer look. This aligns with known pop-out hierarchies (color > size > shape > orientation) [10]. Ensuring sufficient contrast is part of this: e.g. rings might use a lighter shade or semi-transparent fill so they don't overpower petal outlines. If a glyph has too many petals or nodes (making it busy), we might algorithmically limit certain combinations or aggregate minor attributes to preserve clarity – *less can be more* for real-time decoding.

- **Orthogonality of Channels:** We must design the glyph such that encoding one attribute minimally distorts another. This is related to **separable channels** in visualization design [12]. For instance, if a petal's size or number inadvertently changes the overall glyph size, it could mislead interpretation of intensity rings (which rely on radius). To counter this, we normalize sizes: e.g. always draw the petals within a fixed radius and scale internal features accordingly [13]. Likewise, if shape changes (like a very wide petal for a certain action) would increase the apparent area of the glyph, we adjust other elements (perhaps slightly reducing petal length) to keep the visual weight consistent [13]. The design will be tested for such interactions to fine-tune independence.

- **Real-Time Legibility (AR Considerations):** In AR, additional challenges include varying backgrounds and depth. We impose constraints such as **minimum angular size** – glyphs should not appear smaller than a certain visual degree, or they become hard to read. If the user steps back, the system might switch to a simpler representation (e.g. just a colored dot or single icon for the herb) to avoid tiny, inscrutable shapes. We also consider using **outlining and halos** around glyphs in AR to ensure they stand off from the background. Prior work suggests adding halos or contrast outlines can help differentiate overlapping symbols in dense scenes [16]. Moreover, depth ordering in AR is important: we will try to layer glyphs or labels such that important ones are not obscured. A **chroma-depth coloring** or slight drop-shadow could be used to indicate depth layering if multiple glyphs are floating in the view [17].

- **Frame Rate and Dynamism:** Since this is real-time, the glyph rendering must be efficient. We avoid overly intricate shapes (no excessively high polygon count or continuous animation that could distract or drop frame rates). Simple petal shapes and circles are computationally cheap. If herbs or user's view move, the glyphs should update positions smoothly to remain next to their herb (in AR) – this means an anchoring system with damping to avoid jitter.

By adhering to these constraints, we ensure each herb glyph remains clear and informative on its own, and multiple glyphs can co-exist on screen or in AR without degenerating into clutter.

## Alternative Visual Encodings and Pros/Cons

In addition to the primary "flower glyph" mapping, we considered several alternative encoding designs. Each offers a different approach to mapping herb data to geometry, with its own benefits and trade-offs:

### Alternative 1: Radial Star Plot (Petal-Polygon) Glyph

Instead of discrete petal shapes, this approach uses a continuous **star/radar chart** style. Each herb is drawn as a closed polygon on polar axes: - We assign each attribute a fixed radial axis around a circle (e.g. one axis for *intensity*, one for *power*, etc.). Categorical attributes (like *element* or various *actions*) could be represented as multiple axes or converted to numeric scores (e.g. a binary 0/1 if an action is present, or a scaled score for each action's strength). - The herb's values are plotted as points on each axis, and connected to form a star-shaped polygon. For instance, intensity might map to radius on one axis, power on another; element could be split into multiple axes (one per possible element, with value 1 or 0). The polygon's shape thus encodes the herb's profile.

**Pros:** This encoding is highly **quantitative** – it directly shows magnitudes along each axis, making it easier to read exact values or compare proportions. Users can compare two herbs by the shapes of their radar plots (e.g. one herb might have a "spikier" shape indicating an extreme value in one attribute). Radar glyphs are a familiar visualization for multivariate data, potentially easing user learning. Additionally, the continuous shape avoids the potential overlap of separate petal icons; everything is within one unified form.

**Cons:** Interpretation can be harder if many dimensions (axes) are used – the polygon can become irregular or star-shaped in confusing ways if there isn't a logical ordering of axes [18] [19]. Also, categorical data is awkward to show on numeric axes (the user might misread a "0.5" value on an element axis as meaningful when it might just mean absence/presence). If the axes are not carefully ordered, the glyph's shape may vary wildly with small reordering, hurting consistency [20]. Another drawback is that filled radar charts can obscure gridlines or labels; in overlays, multiple radar polygons would overlap on the same axes which is visually complex. In summary, while star plots convey precise numeric comparisons well, they sacrifice some of the intuitive metaphor (the flower-like imagery) and can become cluttered with many variables.

### Alternative 2: Spiral Phyllotactic Glyph

This design uses a **spiral (phyllotaxis) layout** as the basis for the glyph, rather than a circular petal arrangement. The herb's attributes are represented by symbols or marks placed along a golden-angle spiral: - Starting from the center, we place small icons or shapes for each attribute outward along a spiral path. For example, an icon for each *action* the herb has, colored by *element*, sized by *intensity*, etc., in a sequence. - The spiral could be oriented such that one full revolution corresponds to a time cycle (mapping the *time* attribute directly onto angular rotation), making it naturally show temporal aspects (e.g. an annual herb might have markers that line up every 360° on the spiral). The use of the golden angle (≈137.5°) between successive items ensures an even, non-clumping distribution [3]. - If an herb has many attributes or data points (say each measurable compound could be a point), they would fill out a sunflower-like pattern – denser in the center and radiating outward. For our simpler case (5 core attributes), we might replicate some attributes along the spiral for visual balance (or leave gaps where not applicable).

**Pros:** The spiral glyph is **aesthetically pleasing and space-filling**, which could be useful for an overview of many herbs at once. Its non-overlapping packing means if we show multiple herb glyphs, we can pack more info in a given area without collision [3]. The spiral also inherently encodes sequence – if the *time* attribute is something like a chronological stage, this is an ideal way to show it (e.g. a dashed spiral line could grow through the glyph indicating progression over time). This design might shine in an AR scenario where herbs are pinned in space: a spiral glyph could surround the herb like an aura, with its spiral extending outwards to indicate its profile.

**Cons:** A spiral layout is **less immediately structured** than the radial flower – users might not intuitively know how to read it without a legend or animation. Individual attributes might be harder to isolate: e.g. an action icon on the spiral is one point among five, not as clearly separated as a dedicated petal. If not carefully done, the spiral could also suffer from perspective issues (in AR, parts of the spiral farther from the user might appear smaller or occluded). The spiral's density might become a problem for interaction – clicking a small icon on a tightly wound spiral could be fiddly if icons are close together. In essence, while the phyllotactic glyph excels at avoiding overlaps and can encode an extra dimension (order/sequence), it may trade off some **clarity and simplicity** in reading specific values or categories.

## Alternative 3: 3D Layered Toroid Glyph

This encoding takes advantage of 3D form, imagining the herb's data mapped onto a **toroidal (donut-shaped) glyph** or a set of concentric 3D rings: - The torus itself could represent *time* as the circular direction (one full loop around the torus = one cycle of time, making time a naturally looping dimension). The *element* might map to the torus's color, as before. *Intensity* could control the torus's **tube radius** (thicker torus = more intense herb), and *power* could control the **major radius** or overall size of the torus. *Actions* might be depicted by engraved symbols or protrusions on the torus surface – e.g. small bumps or petal-like extrusions positioned at various angles on the torus. - Another approach is multiple rings: imagine a stack of rings (like a planet with multiple rings) where each ring encodes one aspect. For example, an inner ring colored by element, an outer ring segmented into petals for actions, etc., forming a 3D composite.

**Pros:** This design is inherently **rich in encoding capacity** – as noted, a toroidal shape offers multiple geometric parameters (inner radius, outer radius, etc.) to map variables [5]. In an AR or light-field display, a 3D glyph can be inspected from different angles, potentially allowing the user to "peek" at different attributes (e.g. look at the cross-section for intensity, the face of the torus for actions). Depth can also help separate information layers; for example, action symbols could hover above or below the main ring to avoid overlapping with the color-coded surface. A 3D glyph could be very engaging and provide a futuristic UI, and it naturally supports **stereoscopic highlighting** (pop-outs) for emphasis. If multiple herbs are compared, they could be arranged in 3D space (perhaps on a rotating carousel), leveraging depth to reduce screen-space overlap.

**Cons:** The complexity of a 3D glyph is its biggest downside. Users might find it **harder to read quantitative values** from shapes – e.g. judging the exact thickness of a torus or the exact angle of a bump without careful manipulation. It might require interactive rotation or tooltip readouts for precision. In 2D fallback (normal screens), the 3D glyph has to be projected, which could obscure some data (one side of the torus is hidden). Occlusion of parts of the glyph by itself is a concern (solvable by partial transparency or cutaway views, but that adds more design overhead). Additionally, designing effective interactive cues in 3D is challenging – how to indicate that a user should click a floating bump on a torus? Performance is another issue: rendering many 3D objects with lighting in real time (especially in AR) can be heavy, though modern AR systems can handle it with optimization. Finally, consistency in view: in AR, the user's viewpoint could drastically change the appearance (the glyph might appear as just a circle from a certain angle, losing the info on thickness), so we'd need to possibly lock the glyph's orientation facing the user (billboarding), which in turn might break the 3D illusion. In summary, the toroid glyph offers **multi-dimensional encoding and AR novelty** at the cost of increased complexity and potential usability hurdles.

## References (Data-to-Geometry Encodings)


1. Borgo et al., *"Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications,"* Eurographics STAR Report, 2013 – **Comprehensive survey of glyph design** (visual channels, separability, etc.) [10] [11] [2] [13] .
2. Cleveland & McGill, *"Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods,"* 1984 – **Ranks accuracy of visual encoding channels** (e.g. position, length best; color, area less precise) [2] .
3. Neumann et al., *"PhylloTrees: Phyllotactic Patterns for Tree Layout,"* EuroVis 2006 – **Introduces phyllotaxis for non-overlapping layouts**, leveraging golden-angle spirals for optimal packing [3] . Demonstrates even spacing for unknown number of items (inspiration for spiral glyphs).
4. Ivanov et al., *"Exploration & Anthropomorphism in Immersive Unit Visualizations,"* IEEE VR 2018 – **Uses fractal phyllotaxis in VR** to space out data points evenly in 3D, improving navigation in dense scenes [4] . Highlights benefits of phyllotactic spacing in immersive environments.
5. Ropinski et al., *"Survey of Glyph-Based Visualization Techniques for Spatial Multivariate Data,"* Computers & Graphics 2011 – **Glyph design in practice**, including 3D glyphs (superquadric shapes like ellipsoids and tori) and their data-mapping capacity [5] . Useful for understanding how a torus glyph can encode multiple variables with geometric parameters.
6. Onzenoodt et al., *"Out of the Plane: Flower vs. Star Glyphs…,"* IEEE TVCG 2023 – **User study comparing flower-like glyphs to star (radar) glyphs** for high-dimensional data [21] . Found that different glyph shapes affect task performance, underscoring design trade-offs (salient "spikes" vs. homogeneous shapes for clustering tasks).
7. AR Labeling Research (e.g. Tatzgern et al. 2014, Bell et al. 2001) – **Guidelines for AR visual overlays**, recommending avoidance of label overlap and occlusion in cluttered scenes [15] . Informs our overlap avoidance and grouping strategy for herb glyphs in AR.
8. Ward, *"A Taxonomy of Glyph Placement Strategies,"* InfoVis 2002 – Discusses **placement of multivariate glyphs** (data-driven, structure-driven, etc.) and notes the importance of jitter or adjustment to prevent visual clustering [22] . Relevant for ensuring our glyphs are well spaced and misinterpretation of overlapping glyphs is avoided [22] .
9. Zhou & Hansen, *"A Survey of Radial Visualization for Multidimensional Data,"* 2018 – Reviews radial techniques and their benefits. Notes that radial layouts exploit two strong channels (angle and radius) and naturally handle periodic data [1] , supporting our use of a circular design.

10. **Miscellaneous:** Additional principles from visualization classics (e.g. Bertin's retinal variables, Tufte's design rules) implicitly underlie our design – such as using **color hues for categories**, sizes for quantitative differences, and maintaining a **clear visual hierarchy** to guide the viewer's attention [10]. These principles ensure the geometric mapping is both **informative and perceptually effective**.

---


[1] Sample Paper Title
https://www.cs.ubc.ca/~tmm/courses/547-19/projects/gabriel/finalreport.pdf

[2] [8] [9] [10] [11] [12] [13] [14] [16] [17] [22] (PDF) Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications
https://www.researchgate.net/publication/235928194_Glyph-based_Visualization_Foundations_Design_Guidelines_Techniques_and_Applications

[3] [4] PhylloTrees: Phyllotactic Patterns for Tree Layout. | Request PDF
https://www.researchgate.net/publication/220778416_PhylloTrees_Phyllotactic_Patterns_for_Tree_Layout

[5] [6] Survey of glyph-based visualization techniques for spatial multivariate medical data
https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.100/institut/Papers/viscom/2011/1-s2.0-S0097849311000203-main.pdf

[7] Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications | TU Wien – Research Unit of Computer Graphics
https://www.cg.tuwien.ac.at/research/publications/2013/borgo-2013-gly/

[15] Exploring AR Label Placements in Visually Cluttered Scenarios
https://arxiv.org/html/2507.00198

[18] [19] [20] [21] (Open Access) Evaluating Ordering Strategies of Star Glyph Axes (2019) | Matthias Miller | 8 Citations
https://scispace.com/papers/evaluating-ordering-strategies-of-star-glyph-axes-39m216o7la