

A GEOGEBRA TOOL FOR DRAWING IMMERSIVE PERSPECTIVES

António B. Araújo

DCeT

Universidade Aberta / CIAC-UAb

Lisbon Portugal

antonio.araujo@uab.pt

ABSTRACT

Spherical perspective provides a connection between traditional handmade drawings and virtual reality environments, which can be exploited to create new forms of hybrid immersive artworks. We discuss the use of GeoGebra tools as auxiliary software for the creation of immersive designs in azimuthal equidistant (360-degree fisheye) spherical perspective.

CCS CONCEPTS

• Applied computing → Arts and humanities • Computing methodologies → Computer graphics

KEYWORDS

Spherical perspective, technical drawing, perspective drawing, interactive geometry, computer algebra system, anamorphosis, virtual reality, VR panoramas, fisheye, geodesics

1 INTRODUCTION

Spherical perspective has a relatively short history since its formal inception in the work of Barre and Flocon [1], [2] in the 1960s. Recent theoretical developments have made exact technical drawing accessible and systematic in three spherical perspectives: azimuthal equidistant (or 360-fisheye) [3], equirectangular [4], and cubical [5]. The technical relation between these spherical perspectives and the source data for VR panoramas creates a link between handmade spherical perspectives, drawn with traditional tools, and digital immersive visualizations, opening the way to *hybrid models*, that is, artworks that have both a physical and a digital realization, and an artistic process that is similarly a mix between physical and digital. A recent exhibition displayed such hybrid models [6], shown in triple form; as flat perspective drawings hand-drawn in cubical or equirectangular perspectives; as physical models (painted spheres or cubes) in the tradition of Dick Termes' spherical paintings [7], [8], and as immersive visualizations, to be seen from within. These hybrid models have applications to several fields, from art to entertainment, to architectural design [9]. They can be used to register an observed scene (as in urban sketching) or for planning a full dome presentation [10]. Although the main push has been, lately, in the direction of establishing methods for technical drawing with traditional tools, there is also an obvious need for digital tools that can be used to develop such drawings efficiently. Most existing tools for immersive drawing, however, such as the software Sketch 360 [11], are designed to avoid the aspects of technical drawing altogether rather than facilitating them. This is for good reason – many artists interested in immersion, probably most, just want an interface that will allow them to explore immersive drawing intuitively within the mental framework they already have, and not force them to

learn a new and complex type of perspective drawing. But there is also a place for a “rational drawing” [12] of immersive art, that is, drawing in the tradition of classical perspective and descriptive geometry, which embraces the technical and mathematical aspects of the process, and can even make the process itself (and its inherent mathematical beauty) the subject of the artwork, especially when making use of a new type of perspective process that truly clashes with the intuition, and in a very concrete sense questions the expectations of both the spectator and the artist regarding the geometric features of the represented space. We therefore required a software that rather than avoid the inherent difficulties of this new geometric process, works to expose them clearly and to facilitate the learning of their peculiarities by the artist so inclined.

There is such a technical drawing tool for equirectangular spherical perspective drawing: a raster drawing program called *Eq A Sketch 360* described in [13]; but there was up to now no equivalent tool for fisheye perspective.

In the present paper we report on a new tool we created to satisfy this need. Originally it was conceived along the model of *Eq A Sketch 360*, as a standalone piece of software, written in Processing, but with the proviso that it should be vectorial rather than raster, and interact well with orthogonal projection auxiliary drawings, which are much needed in the fisheye case. It soon became apparent, however, that instead of programming this as a standalone application, the goal might be achieved far better by building a set of GeoGebra tools and scripts.

In this paper we describe these tools and illustrate their uses with some applications. We also describe the mathematical constructions of geodesics that we had to solve in order to construct the objects required by the software. We expect the present software to be very useful in the construction of hybrid artworks in spherical perspective. The tool may be found at the author's webpage [14].

2 SPHERICAL PERSPECTIVE

We need to provide a brief review of spherical perspective, in particular of the azimuthal equidistant spherical perspective (“360-degree fisheye”). More details may be found in [3].

Suppose you have a large transparent sphere with an observer at its center point O . If you were to project the 3D environment radially towards O , and mark where each ray hits the sphere, you would obtain a drawing on the sphere that, for an observer at O , would look exactly like the original scene. This mimetic object is called a (spherical) anamorphosis.

It is possible to project this spherical anamorphosis onto a plane in such a way as to get a flat drawing – a spherical *perspective* – that may look very deformed by the projection, but still contains all the visual data required to reconstruct the anamorphosis. It is convenient for our purposes to require that the projection's full image be contained in a *compact* set, that is, a bounded, closed region of the plane. Such a compact region is readily converted into a datafile,

which becomes a source for VR panoramas. These panoramas in turn work by re-converting the flat data to cover a virtual sphere, within which they place the observer to interactively recover the experience of the original visual environment.

2.1 AZIMUTHAL EQUIDISTANT SPHERICAL PERSPECTIVE

To give a spherical perspective is to give the form of its flattening. In our case we are interested in the azimuthal equidistant perspective, whose flattening has the same name. The azimuthal equidistant map projection has a very old ancestry, coming from cartography both of the Earth and of the night sky [15], [16], being especially adapted for the latter. It works by releasing the meridians at a point B (Fig. 1 - top) and straightening them without stretching at a point B's antipodal point F. This results in a flat disc (Fig. 1 - bottom) that has one-to-one correspondence with the sphere at all points except the outer circle, which maps entirely to the single point B, each point of this circle corresponding to one of the meridian directions from which B can be approached along the sphere.

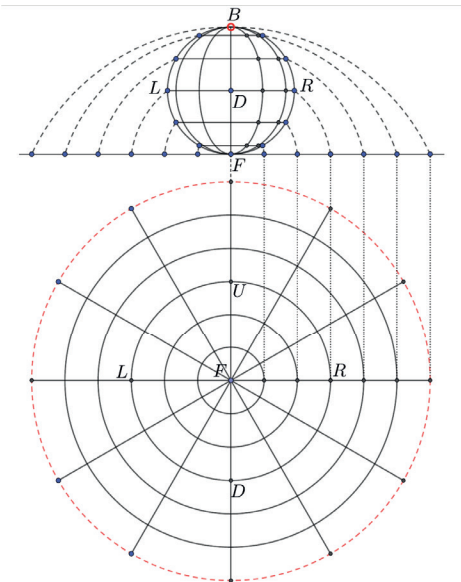


Figure 1: The Azimuthal equidistant projection from the visual sphere (above, in side view) to the perspective disc.

3 A GEOGEBRA DRAWING TOOL

In [3] it is shown that geodesics are the main objects of any spherical perspective from which all other objects of interest – mainly line and plane images – may be obtained through judicious diagrams. It is then shown how to classify and plot these geodesics from the measurement of special points. However, these special points may sometimes be hard to measure especially when drawing from observation, or sometimes pairs of points may be obtained from internal perspective constructions that are theoretically enough to define a geodesic but not enough to construct it from the methods of [3]. In [17], [18] special dynamic gridding methods are used which take advantage of the perspective's symmetry groups to plot a geodesic from two arbitrary points graphically. In [13] an analytic method is

developed to achieve the same end. We will adapt the viewpoint of [13] to our purposes, but while the former works on the equirectangular framework, we will here plot the geodesics of the azimuthal equidistant perspective.

3.1 FORMULA FOR THE GEODESIC THROUGH TWO POINTS

We want to plot the perspective image of the geodesic that joins two arbitrary points on the sphere. This is the fundamental result we need for our GeoGebra drawing script to work as intended.

We need to consider two coordinate systems and to classify our geodesics within them. We consider the observer's eye at the point $O = (0,0,0)$, and we put an orthonormal coordinate system (x, y, z) at O . For reference we place on the sphere points F(oward), R(ight), and U(p), such that FRU form an orthonormal system, and call B(ack), L(eft), and D(own) their antipode points, diametrically opposite on the sphere. We assume that, as in Figure 1, the sphere was "blown up" at point B, and therefore the perspective disc is centered on F. We set coordinate axes so that $R=(1,0,0)$, $U=(0,1,0)$, $F=(0,0,1)$.

The astronomer's coordinate system is the natural one for the azimuthal equidistant flattening: a point P is located on the sphere by two angles, the zenith angle or polar angle $\zeta = \angle POF$ and the azimuth $\theta = \angle P_F OR$ where P_F is the orthogonal projection of P onto the plane UOR . With our choice of coordinates above we have

$$\begin{aligned} x &= \sin(\pi/2 - \zeta) \cos(\theta) \\ y &= \sin(\pi/2 - \zeta) \sin(\theta) \\ z &= \sin(\pi/2 - \zeta) \end{aligned}$$

In the perspective plane a point P of coordinates (θ, ζ) will project according to

$$\mu: S_0^2 \rightarrow R^2, (\zeta, \theta) \mapsto (x', y') = \zeta(\cos(\theta), \sin(\theta))$$

Hence the flattening is naturally in polar coordinates with radius $\zeta = \sqrt{(x')^2 + (y')^2}$ and angle $\theta = \text{atan2}(y', x')$. Given two points on the sphere, of coordinates $\theta_1 = \text{atan2}(y'_1, x'_1)$, $\zeta_1 = \sqrt{x'^2_1 + y'^2_1}$ and $\theta_2 = \text{atan2}(y'_2, x'_2)$, $\zeta_2 = \sqrt{x'^2_2 + y'^2_2}$, we want to find the geodesic that joins them.

We can obtain this by piggybacking on the methods of [13] by noticing the relationship between our problem and the formula therein obtained for the analogous equirectangular problem. In that paper the geodesics are characterized by the value of their apex (λ_m, φ_m) where φ_m is the maximum value of the angular elevation (in the horizontal coordinate system) that the geodesic reaches, and λ_m is the bearing at which it is reached. In the present case if we can relate the horizontal coordinate system (λ, φ) of [13] with the polar angle defined above by the identifications $\lambda = \theta$ and $\varphi = \pi/2 - \zeta$. Then, the variables a, b, c of [13] become (since $\cos(\pi/2 - \zeta) = \sin(\zeta)$ and $\sin(\pi/2 - \zeta) = \cos(\zeta)$),

$$\begin{aligned} a &= \sin(\zeta_1) \sin(\theta_1) \cos(\zeta_2) - \sin(\zeta_2) \sin(\theta_2) \cos(\zeta_1) \\ b &= \sin(\zeta_2) \cos(\theta_2) \cos(\zeta_1) - \sin(\zeta_1) \cos(\theta_1) \cos(\zeta_2) \\ c &= \sin(\zeta_1) \sin(\zeta_2) \sin(\theta_2 - \theta_1) \end{aligned}$$

and the formulas for the apex become

$$\zeta_m = \pi/2 - \arctan(\sqrt{a^2 + b^2}/|c|)$$

$$\theta_m = \text{atan2}(-\text{sgn}(c)(b, a))$$

Note that in this case the apex (θ_m, ζ_m) is not the point of the geodesic where ζ hits its maximum (as φ does in the equirectangular case) but rather its minimum.

By [13] we know that, for a given apex, a geodesic is parametrized by

$$\varphi(\lambda) = \arctan(\tan(\varphi_M) \cos(\lambda - \lambda_M))$$

in the vertical coordinate system. This implies

$$\zeta(\theta|\eta_m, \zeta_m) = \pi/2 - \arctan(\tan(\pi/2 - \zeta_m) \cos(\theta - \theta_m))$$

Hence the geodesic image can be parametrized by θ as

$$\eta \mapsto (x', y') = \zeta(\theta|\eta_m, \zeta_m)(\cos(\theta), \sin(\theta))$$

3.2 GEOGEBRA IMPLEMENTATION: PROBE POINTS AND DYNAMICAL GEODESICS

The result of the previous section is what we need to construct our drawing tool. We did it as a script in the GeoGebra environment.

GeoGebra [19] is a free software interactive geometry environment (2D and 3D), with integrated computer algebra system as well as scripting hooks to JavaScript and to its own scripting language. It started out as a master's thesis project by Markus Hohenwarter [20] and is now used worldwide by teachers of all levels, from elementary school to university, and is available in many platforms such as Windows, macOS, Linux, Android and also as a web app. By making our tools in GeoGebra we make them part of an already established ecosystem for teachers of mathematics, perspective, and descriptive geometry. We also get immediate access to the integrated Euclidean geometry tools which are very useful for the auxiliary constructions that are the input for our own perspective constructions.

GeoGebra is inherently vectorial and scale-independent, and holds objects defined by relations to and constructions from other objects. It handles parametrized curves, and allows these great numbers of objects to be worked on at any scale, within the limitations of the hardware. This is very useful as spherical perspective constructions can entail hundreds of auxiliary points even for relatively modest drawings (see Fig. 5 for instance).

Further, GeoGebra can handle a large number of objects that depend dynamically on each other, thus enabling us to construct not only complex drawings, but drawings that can themselves be animated through scripts and sliders. So, for instance, a tiled floor can be drawn by hand and then parametrized for its vanishing points to shift position or for the tile to become larger or smaller. The user of our software can very easily extend it to his own uses by altering or adding to our scripts. Since we intend for an open source artistic and didactic tool, this is very much preferable to a more monolithic, standalone solution.

Our GeoGebra script works through the concept of *probe points*. The main thing is that there are two dynamically controlled points

A and B that the user may move with the mouse. These are called our probe points as they serve to both construct and analyze a drawing by investigating, for instance, what are the coordinates where some pair of geodesics intersect, and then pass a line through such intersection. In Figure 2 the probe points A and B can be seen, marked in blue. Whenever we move them, GeoGebra traces the geodesic through A and B (the green curve in Figure 2) according to the formula from the previous section. This way it is easy to adjust the resulting line dynamically to any required points already present in the drawing. These are called *internal* constructions. By contrast, *external constructions* generate new lines and points from data in external diagrams or from external numerical data.

Beyond points A and B, there are two additional point I and J that live on this probe geodesic and move along with it. When A and B are fixed in the intended place, thus setting a geodesic, the user may then move I and J freely along the geodesic to select any segment within it. Then, GeoGebra buttons may be pressed to execute a script that draws a copy of the geodesic segment between points A and B or between I and J. The shortest segment of the geodesic is always selected automatically.

The basic commands of the script are as follows:

Set A and **Set B**: these set the values of A and B numerically, which is useful since often we want exact locations for our vanishing points and it is not efficient to place the points there with the mouse. These commands set the values of the azimuth, theta, and of the zenith angle, zeta, defined in the previous section, for each point.

Line AB and **Line IJ**: These draw the above mentioned geodesic segments.

The curve through the probe points is defined dynamically as a GeoGebra *Curve* command, as:

$$\text{Curve}((180/\pi)*\cos(t)*\text{fish}(t), (180/\pi)*\sin(t)*\text{fish}(t), t, -180^\circ, 180^\circ)$$

(where angles are being converted to degrees for convenience of the user) and *fish()* is the function

$$\text{fish}(x) = 90^\circ - \text{atan}(\tan(90^\circ - \text{zeta}_m) * \cos(x - \text{theta}_m))$$

that defines a geodesic curve of given apex $(\text{theta}_m, \text{zeta}_m)$, and these coordinates are in turn calculated dynamically by GeoGebra anytime A or B moves, using the formulas we established in the previous section. When the draw button is pressed we use the GeoGebra command *CopyFreeObject()* to make static copies of the variables and of the resulting curve, and leave it at the screen as a vectorial curve, which can be edited at will.

Draw horizontal/Draw Vertical: Draw the image of a vertical/horizontal plane through probe point A. These are useful as very common operations in both auxiliary diagrams and drawings as such.

Draw Elevation Line: This draws a line of constant elevation φ . Also a common auxiliary operation for the constructions in [3].

Draw antipodes: Antipodes are a very useful construction in spherical perspective, so it is convenient to be able to obtain them at the click of a button. Following [3], the antipode of a point *P* is the point *P** on the ray from *P* through *F*, such that $|PP^*|$ equals half a diameter of the perspective disc.

These are the necessary commands to draw the ordinary constructions of spherical perspective. Apart from these, we need all the apparatus of Euclidean Geometry and in particular of orthogonal projections. The advantage of using GeoGebra is that this part comes free with the main program itself. Further, everything is dy-

dynamic (dependent objects change as we move free objects) and everything is vectorial, so that the end result may be saved as an svg format file and imported into vectorial illustration software such as inkscape so that more elaborate illustration tools may be applied to the diagrams constructed with our script. This way we have a functional pipeline for making spherical drawings that may end up both as dynamic, animated scripts, or as static illustrations with a more elaborate finish.

4 EXAMPLES OF CONSTRUCTIONS

The GeoGebra tools we have described are enough to make both internal and external constructions. We here present an example that uses both such constructions.

In the bottom of Figure 3 we see a square tile that we have constructed in top orthographic view. This tile shares the screen with the spherical perspective disc (upper part of the Figure) and can be repositioned relative to it along a vertical. Point D of the sphere is seen in both the spherical projection disc and in the orthographic view and is a dynamic reference point that when moved with the mouse will move the entire construction with it.

In this figure we can see the construction of the square tile in spherical perspective view. This is done by an *external* construction described in [3]; it is “external” precisely because it relies on the top orthographic view to construct the perspective view. This makes the Euclidean environment of GeoGebra very useful; we can take advantage of having that environment readymade, so we just have to program explicitly the spherical perspective tools on top of it.

In Figure 2, that we have discussed before, we use the tile constructed in Figure 3 to make a full uniform grid of tiles. Now it would be very difficult to do this externally as the top view would soon become very large, so we instead do it through an internal process: that is, we use two diagonals to multiply the square tile repeatedly through a zigzagging process; a line ping-pongs from one diagonal to the other, creating new lines and columns of tiles as it goes (we do not go into details here; this is the process mentioned in the workshop paper [18] as *perspective multiplication* – it suffices here to say that one of the uses of the present tool is to aid in the teaching of these techniques in such a workshop environment, in which the instructor may project his GeoGebra work to a class that follows it by working with pencil and paper techniques).

This multiplication is an *internal* construction, that is, at each step it only uses points and lines constructed in the perspective disc itself. Internal constructions like these are much easier to do with the present software than with the ruler and compass methods of [3], which tend to be external or require special points. In this we are closer to the dynamic gridding methods presented in [18].

In Figure 4 (bottom) we see the same grid, now shaded (in inkscape) to resemble more clearly a tiled floor, and used as a basis for the construction of other objects: three cubes and a ramp.

The whole construction (Figure 4, top) hinges on line c, in red, a line of constant elevation (in this case of 45-degree elevation). This is the line your gaze describes if you tilt your head at a 45 degree angle to the floor and spin around your vertical axis. Hence any line on a vertical plane that has a 45 degree slope to the ground will have its vanishing point on line c. This is used in the figure in two different ways. First, on the right, it is used to draw a ramp with 45 degree slope. The base of the ramp is defined by the grid, and then the ramp must go (see blue line on the Figure) to vanishing point V where its vertical plane intersects the (red) line of 45-degree elevation.

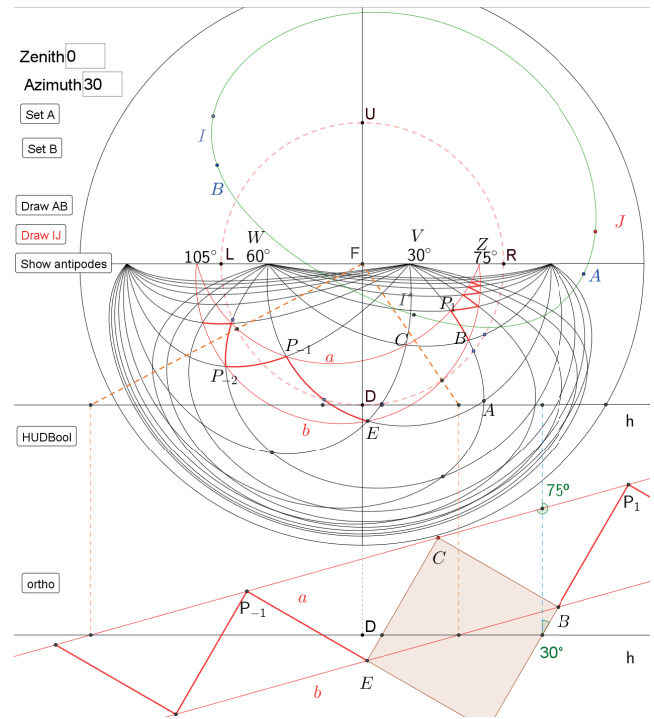


Figure 2: A spherical grid construction in the azimuthal equidistant GeoGebra Script.

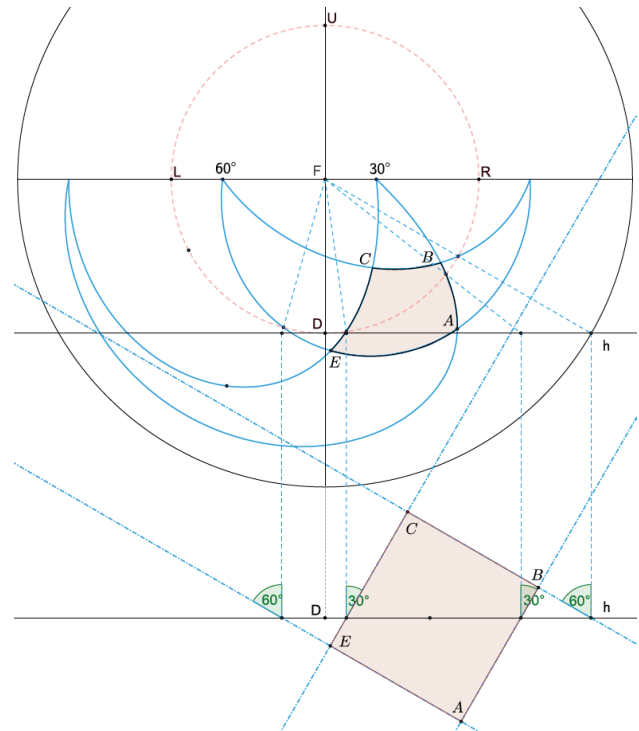


Figure 3: Construction of a square tile from its top view.

In the case of the cubes, we define their base using the tiles, and then to find their height just send a line (blue) to vanishing point U , where the vertical plane of the face touches line c . Since the cube has equal sides, where this 45-degree line touches the vertical from the base is where the top of the cube must lie. We do this for the various cubes in the figure (and for the equally sized spaces between them). In this way we can very simply measure lengths and construct objects internally by reference to each other. It is also clear from the figure why such techniques are needed to draw in spherical perspectives and why intuition must be trained. One would never say, from mere intuition, that the three cubes in this picture are equal, and yet we can reason they are.

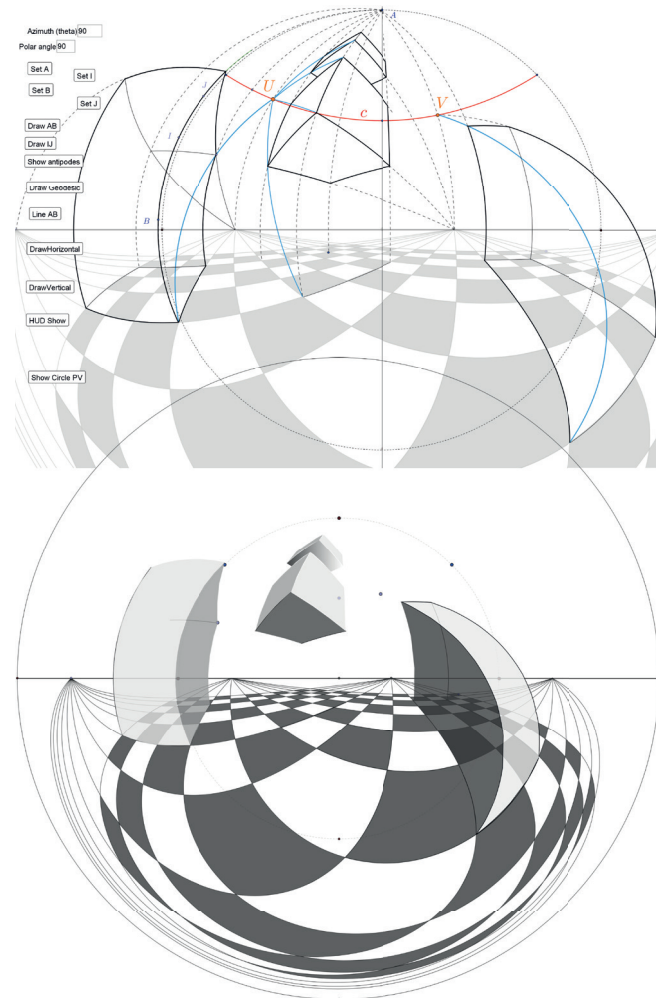


Figure 4: Ramps and boxes – a construction of objects on top of a grid. Top: construction in GeoGebra. Blue lines going to vanishing points U and V on line c have a slope of 45°. Bottom: end result, shaded in inkscape.

Classical problems such as drawing cubes or tiled floors become new again in these immersive perspectives, carrying not only a very concrete meaning through their immersive properties when seen in VR, but also an inherent aesthetic when seen – in their strange, deformed shapes – as forms in themselves, with their own inherent

qualities and charms, their own aesthetic quality.

Again, in the example of Figure 5, a cubical room with a triangular roof, with the observer at the center, a familiar object takes new aspect. The whole construction is inherently symmetric and simple; it is defined by 45 degree geodesics that first go through the vertices of the cube, then subdivide it further and further, as far as the artist may wish (and hardware allows). Again the simple cubical form takes novel, exciting qualities, inviting the artist to experiment.

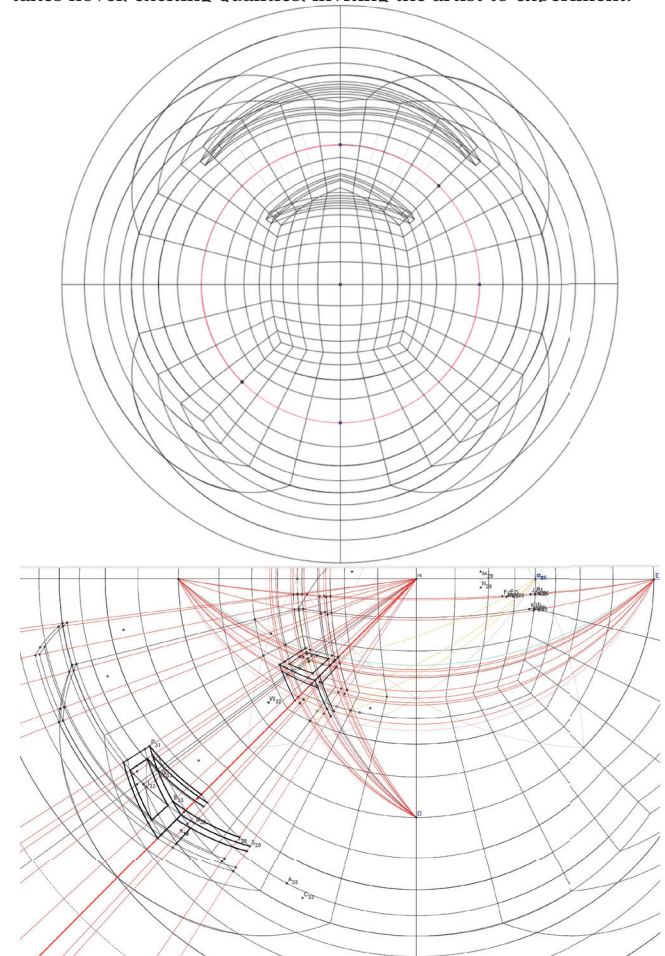


Figure 5: Top: A cubical room. Bottom: Detailing its contents, in this case adding two chairs against the back and front walls, the back chair extremely deformed by the perspective.

5 CONCLUSIONS

We presented a GeoGebra tool that works as a drawing app for azimuthal equidistant spherical perspective, using GeoGebra's interactive geometry framework. We are unaware of any other drawing tool that performs this function. We obtained a formula to determine a geodesic's apex from two given points, a formulation that may be useful to other programmers. We expect this tool to be useful for researchers, teachers, and artists. The latter may use it both to learn spherical perspective methods, and as a production tool, in a pipeline that may feed into vectorial programs such as inkscape, for further rendering.

ACKNOWLEDGMENTS

Author A. B. Araújo was funded by FCT national funds through project UIDB/Multi/04019/2020.

REFERENCES

- [1] A. Barre and A. Flocon, *La perspective curviligne*. Paris: Flammarion, 1968.
- [2] A. Barre, A. Flocon, and G. Bouligand, "Etude comparée de différentes méthodes de perspective, une perspective curviligne," *Bulletin de la Classe des Sciences de La Académie Royale de Belgique*, vol. 5, no. L, 1964.
- [3] A. B. Araújo, "Ruler, compass, and nail: constructing a total spherical perspective," *Journal of Mathematics and the Arts*, vol. 12, no. 2-3, pp. 144-169, 2018, doi: 10.1080/17513472.2018.1469378.
- [4] A. B. Araújo, "Drawing equirectangular VR panoramas with ruler, compass, and protractor," *Journal of Science and Technology of the Arts*, vol. 10, no. 1, pp. 2-15, 2018, doi: 10.7559/citarj.v10i1.471.
- [5] A. B. Araújo, L. F. Olivero, and A. Rossi, "A Descriptive Geometry Construction of VR panoramas in Cubical Spherical Perspective," *disegno*, no. 6, pp. 35-46, Jun. 2020, doi: <https://doi.org/10.26375/diseño.6.2020.06>.
- [6] A. B. Araújo, L. F. Olivero, and S. Antinozzi, "HIMmaterial: Exploring new hybrid media for immersive drawing and collage," in *Proceedings of the 9th International Conference on Digital and Interactive Arts*, Braga Portugal, Oct. 2019, pp. 1-4, doi: 10.1145/3359852.3359950.
- [7] D. A. Termes, "Six-point perspective on the sphere: The termesphere," *Leonardo*, vol. 24, no. 3, pp. 289-292, 1991.
- [8] D. Termes, *New Perspective Systems*. self-published, 1998.
- [9] L. F. Olivero and B. Sucurado, "Inmersividad analógica: descubriendo el dibujo esférico entre subjetividad y objetividad," *Estoa. Revista de la Facultad de Arquitectura y Urbanismo de la Universidad de Cuenca*, vol. 8, no. 16, pp. 80-109, 2019.
- [10] D. Tran Luciani and J. Lundberg, "Enabling Designers to Sketch Immersive Full-dome Presentations," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, San Jose, California, USA, 2016, pp. 1490-1496, doi: 10.1145/2851581.2892343.
- [11] M. S. Scherrotter, "Sketch 360," *Microsoft Garage*, 2018. <https://www.microsoft.com/en-us/garage/profiles/sketch-360>.
- [12] A. B. Araújo, "Explorations in Rational Drawing," *Journal of Mathematics and the Arts*, vol. 14, no. 1-2, pp. 4-7, Apr. 2020, doi: 10.1080/17513472.2020.1734437.
- [13] A. B. Araújo, "Eq A Sketch 360, a Serious Toy for Drawing Equirectangular Spherical Perspectives," in *Proceedings of the 9th International Conference on Digital and Interactive Arts*, Braga Portugal, Oct. 2019, pp. 1-8, doi: 10.1145/3359852.3359893.
- [14] A. B. Araújo, "Notes on Spherical Perspective." <http://www.univ-ab.pt/~aaraujo/full360.html> (accessed Aug. 25, 2020).
- [15] J. B. Harley, D. Woodward, and G. M. Lewis, *The history of cartography*. University of Chicago Press, 2015.
- [16] E. Savage-Smith, "Celestial Mapping,," in *The history of cartography*, J. B. Harley, D. Woodward, and G. M. Lewis, Eds. University of Chicago Press, 2015, pp. 12-70.
- [17] A. Araújo, "Let's Sketch in 360°: Spherical Perspectives for Virtual Reality Panoramas," in *Bridges 2018 Conference Proceedings*, 2018, pp. 637-644, [Online]. Available: <https://archive.bridgesmathart.org/2018/bridges2018-637.html>.
- [18] A. B. Araújo, "A fisheye gyrograph: Taking spherical perspective for a spin," in *Proceedings of bridges 2019: Mathematics, art, music, architecture, education, culture*, Phoenix, Arizona, 2019, pp. 659-664.
- [19] "GeoGebra," www.geogebra.org.
- [20] M. Hohenwarter, "GeoGebra: Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene," Paris Lodron University, Salzburg, Austria, 2002.