

# Eq A Sketch 360, a Serious Toy for Drawing Equirectangular Spherical Perspectives

António B. Araújo  
DCeT  
Universidade Aberta and CIAC-UAb  
Lisbon, Portugal  
antonio.araujo@uab.pt

## ABSTRACT

Eq a Sketch 360 is a simple drawing program for raster sketching in equirectangular spherical perspective. It is built as a serious toy, to develop sketching intuition regarding the structure of equirectangular drawing as proper perspective drawing, with its specific constructions of vanishing points, geodesic segments, line projections, antipodes, and grids. As a drawing tool, it is useful for preliminary perspective sketches to be further rendered with other digital or traditional tools, or as a teaching aid. It is naturally adapted for the input variables adequate for observational sketches. In this paper we survey the operation and purposes of the program. We also show how it calculates the equirectangular geodesic through two given points, which enables one of its main drawing features.

## CCS CONCEPTS

• Human-centered computing~Visualization • Applied computing~Fine arts • Applied computing~Media arts

## KEYWORDS

Spherical perspective, equirectangular, drawing, digital drawing, art education, geodesics, VR panoramas, anamorphosis.

## ACM Reference format:

António B. Araújo. 2019. Eq A Sketch 360, a Serious Toy for Drawing Equirectangular Spherical Perspectives. In Proceedings of Artech 2019, 9<sup>th</sup> International Conference on Digital and Interactive Arts (Artech 2019), Braga, Portugal, 8 pages. <https://doi.org/10.1145/3359852.3359893>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

ARTECH 2019, October 23–25, 2019, Braga, Portugal © 2019 Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-7250-3/19/10...\$15.00  
<https://doi.org/10.1145/3359852.3359893>

## 1 Introduction

*Eq A Sketch 360* is a raster drawing program for sketching VR panoramas in equirectangular spherical perspective. It is aimed at bridging a gap between the practice of digital and handmade spherical perspective drawing, being both a teaching and production tool for equirectangular perspectives. Its name is a playful reference to the mechanical drawing toy *Etch A Sketch* invented by André Cassagnes and first commercialized in 1960. That toy has two knobs that allow the user to draw along horizontal or vertical lines, all other curves being obtained through careful combination of these two motions [1]. Analogously, the present program allows the user to draw any equirectangular line projection using combinations of either horizontal or vertical equirectangular lines. This is possible because these “horizontals” and “verticals” form, by translational symmetry, a generating set for the geodesics of the sphere, as shown in [2]. The program’s main feature is the exploitation of this symmetry, using a sliding equirectangular grid and snap-to commands that allows the user to draw with precision an equirectangular line selected by choosing two arbitrary points or a special apex point. This opens the way to a large class of perspective constructions [3].

The simplicity of this scheme frames *Eq A Sketch 360* as a *serious toy*, focusing the user’s attention on the properties of the perspective constructions rather than on software frills. As a learning/teaching tool, it allows the teacher to draw demonstrations with precision and ease, and for the student to solve problems with speed and clarity by sidestepping the physical difficulties of the construction but not its essentials; as a production tool, it allows the artist to use those same constructions to make precise spherical perspective drawings that can be turned into VR panoramas for artistic or technical purposes. In its present version the program is very limited as a rendering tool, but it can be used as part of a pipeline, to create the structural drawings that may then be rendered in detail by other drawing programs.

## 2 Developments in Spherical Perspective

In order to understand the purpose and uses of *Eq A Sketch 360*, one must understand at least the basics of spherical perspective and its relation to VR panoramas. We start this section with a brief review of what can be more leisurely read in [4] and [2]. Then we finish the section with a simple result that seems to be unpublished elsewhere: a calculation of an equirectangular geodesic from two given arbitrary points. This calculation enables one of the main features of *Eq A Sketch 360*.

### 2.1 Brief Review of Spherical Perspective

VR panoramas are just the latest update in a long history of the relations between perspective drawings and their immersive visualizations through the optical illusion of *anamorphosis*. This is a long narrative, from the first proto-Mixed Reality demo by Brunelleschi's mirror-and-peephole contraption [5] at the doors of Firenze's cathedral, through the magic of 19th century panoramas [6], to our present VR, AR, and MR displays, that look absolutely new only to those who do not indulge in the humbling past time of reading History.

Spherical perspectives are maximally symmetric repositories of anamorphic visual data: they are drawings that represent, in a bounded region of a plane, all the visual data accessible to an observer that is stuck to a point in space but free to look around it in any direction [4]. VR panoramas use this data to recreate the immersive visual experience of being *inside* the original drawn environment.

An example is the A4-sized graphite drawing of Figure 3. It is an equirectangular drawing wherein *both* vanishing points of a corridor can be seen simultaneously. Note that straight lines are deformed into curves. In the VR panorama the line deformations are gone, and only one vanishing point of each line may be seen at a time, but the view can be rotated freely.

Spherical perspective was initiated by Barre and Flocon in the 1960s [7] as a perspective in the classical sense of the word: a systematic method for obtaining all lines and vanishing points through elementary means, usually ruler and compass. This perspective was restricted to a 180-degrees field-of-view, for both technical and philosophical reasons [8]. Extensions to a full view were attempted through qualitative [9], ad-hoc [10] or grid-based [11] approaches. The full generalization of Barre and Flocon's method to the total sphere was obtained by the present author in [4] where a general schema for spherical perspectives is proposed, based on the separation between two steps with dual roles: *anamorphosis* and *flattening*. In the anamorphosis step we project all the visual information onto a geometrical object that is naturally visually immersive and maximally symmetric: we get exactly two vanishing points for each line. In the flattening, the same information is represented on a plane, immersion is lost, some symmetries are broken, but one gains the convenience of drawing on a flat surface. So perspective is

like cartography of the visual sphere. This is a useful schema even when dealing with conical projections onto other surfaces such as cubes [12, 13] or more general parametric surfaces [14], as deformations of the sphere can be reinterpreted as deformations of the flattening map.

This view of spherical perspective as a two-step process is useful in separating the anamorphosis problems, that are common to all spherical perspectives, from the flattening problems, that are specific to each spherical perspective (equirectangular, azimuthal equidistant, etc.). For example, the problem of *occlusion* (related to the hidden faces problem), being solved at the anamorphosis step, is independent of the specific perspective. By contrast, the problem of the flat rendering of line images (a sub-problem of the rendering of the sphere's geodesics) is flattening dependent and specific to each perspective.

These renderings are easily done by computers, but they are subtle when done by hand. The computer can ignore the geometric structure – there are no vanishing points explicitly coded in the usual rendering engines, they are only implicit in the renderings themselves, achieved “blindly”, point by point. To do such a rendering by hand requires judicious classifications of vanishing points, geodesics and lines, and proper use of symmetry groups [3], to do in a few steps what a computer does in millions. The purpose of the formalization of spherical perspective in [4] was to give a framework and strategy to solve any such perspective with a view to handmade drawing, through a small number of operations that a human can perform with simple tools such as ruler and compass.

In [4], the method is both presented and applied to the case of the azimuthal equidistant (fisheye) perspective of Barre and Flocon, generalizing it to a 360-degrees view. In [2], the method is applied to equirectangular spherical perspective. This is an interesting application because equirectangular projection is commonly used by VR software to store 360-degree photographic data to be viewed as VR panoramas. Therefore a drawing in equirectangular perspective can be displayed immersively by the same engines. This creates interesting artistic possibilities for handmade drawing to enhance its classical role, as both a way of designing visual environments and of experiencing the imagined environments immersively. But to make such drawings we need to know how to render lines and their vanishing points in equirectangular perspective.

### 2.2 The problem of line rendering in Equirectangular Perspective

The equirectangular flattening maps a point on the sphere to its longitude  $\lambda$  and latitude  $\varphi$ . The full sphere is rendered to a 2 by 1 rectangle, with  $\lambda$  in  $]180^\circ, 180^\circ]$ , and  $\varphi$  in  $] - 90^\circ, 90^\circ]$ . We set an orthogonal referential with points  $F$ (ront),  $L$ (eft),  $R$ (ight),  $B$ (ack) on the equator and points  $U$ (p),  $D$ (own) at the poles. We set  $\lambda = \varphi = 0^\circ$  at  $F$ .

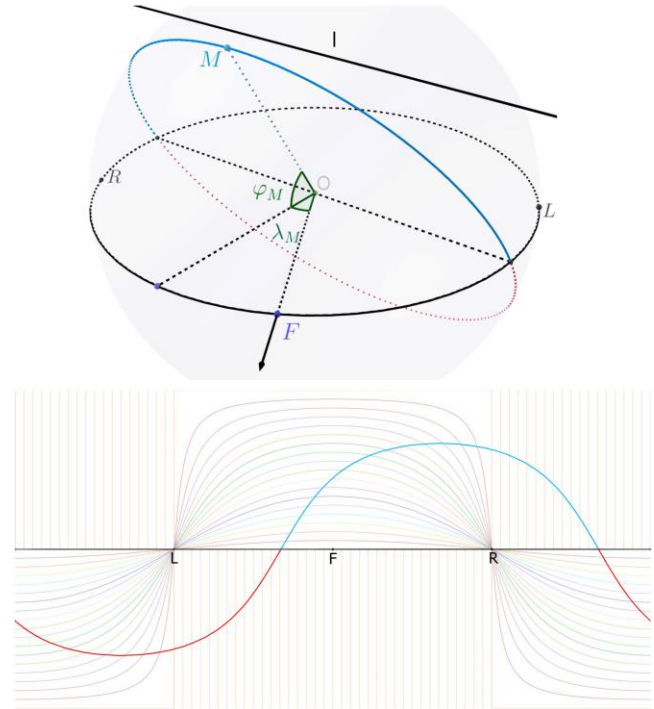
In a perspective we want to draw line projections. These are better seen as subsets of geodesics. Every spatial line defines a plane through the origin of the visual sphere, the plane intersects the sphere on a geodesic and the line projects as half of that geodesic (a meridian) ending in two *antipodal* (diametrically opposite) vanishing points. In the case of equirectangular spherical perspective, geodesics are best classified as either vertical (passing through the *U* and *D*) or non-vertical. The former are trivial to plot since equirectangular projection maps verticals to verticals, so we focus on the latter.

Non-vertical geodesics are determined by the position of their *apex*, i.e., of the highest point reached by the geodesic. Their perspective images have the form [2]

$$\varphi(\lambda) = \arctan(\tan(\varphi_M) \cos(\lambda - \lambda_M)) \quad (1)$$

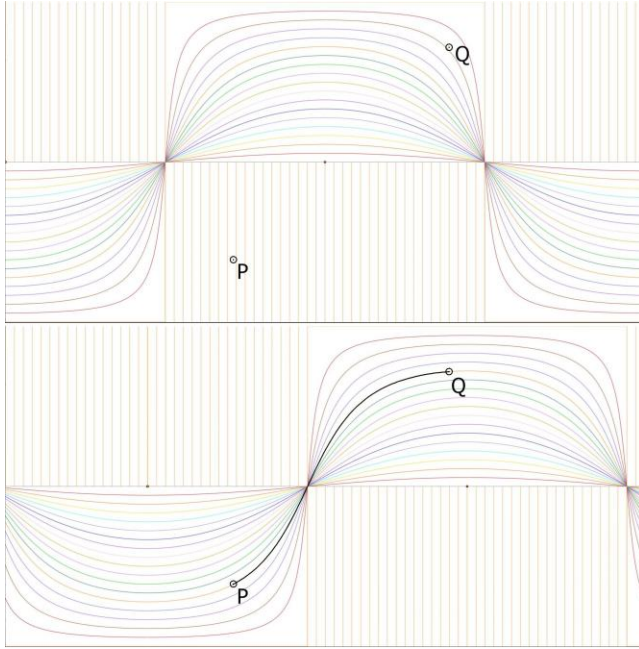
where  $\varphi$  is the latitude (or elevation) as a function of  $\lambda$ , the longitude (or azimuth), and  $(\lambda_M, \varphi_M)$  are the angular coordinates of the apex (Figure 1).

In Figure 2 (top) we see a grid generated by equation 1 when we plot all the geodesics passing the horizon through the left and right reference points (that is, with the apex at  $\lambda_M = 0^\circ$ ) with  $\varphi_M$  varying at  $5^\circ$  intervals. In Figure 2 (bottom) we see an equirectangular line segment AB. These are sigmoid curves that can be unintuitive to draw by hand. In [2] it is shown that the geodesics obtained from equation 1 are well approximated by simpler curves such as arcs of circles and sinusoids up to about 60 degrees elevation, and higher elevation ones are obtainable by descriptive geometry constructions. But since such constructions are hard to use in outdoor drawing, a practical gridding method was proposed [2, 3], based on a minimal set of geodesics that generates all others by translation. For example, in Figure 2 (top) we see that points A and B are not connected by any of the geodesics of the grid presented. But this grid generates all lines through translation. So in Figure 2 (bottom) we see that sliding the grid by  $40^\circ$  to the right the points A and B are found to be located on top of the geodesic of angular elevation  $65^\circ$ . Hence line AB is on the geodesic with apex  $(\lambda_M, \varphi_M) = (40^\circ, 65^\circ)$ .



**Figure 1: Top: The apex  $M$  of a geodesic defined by a line  $l$ . This apex is located at  $\lambda_M = 30^\circ$  and  $\varphi_M = 60^\circ$ . Bottom: The equirectangular projection of the line  $l$  (blue) and of its antipodal meridian (red) that together define the full geodesic. The faded lines are the set of geodesics passing through the reference points to the L(left) and R(right) relative to the observer. These can also be seen as the set of horizontals passing in F(front) of the observer. Verticals are for reference only.**

This allows the draughtsman, by drawing on tracing paper over a sliding grid, to join any two points by a segment (an act so trivial in classical perspective but hard to do in spherical perspective) and therefore to execute the tasks of a perspective proper, like for instance its arithmetic of line division and multiplication, and the accurate *measurement* of distances. This method was used to draw Figure 3.



**Figure 2: Top: Grid of geodesics crossing the horizon at the left and right reference points. The geodesics are generated by varying  $\varphi_M$  at  $5^\circ$  intervals, with fixed  $\lambda_M = 0^\circ$ . All other geodesics are obtained (modulo the error of the  $5^\circ$  approximation) from this family by horizontal translation. Bottom: finding a line through two points by grid translation. Image drawn in *Eq A Sketch 360*.**

This simple sliding symmetry enlarges the possibilities of both handmade and digital equirectangular drawing methods. In both cases, gridding methods were of standard use, but always with fixed grids, interpreted as grids of vertical and horizontal lines vanishing to the reference points  $F, B, L, R$ . This gives undue importance to these arbitrary points and gives no method to draw any other lines except in crude approximation, making precise perspective constructions very difficult. *Eq a Sketch 360* makes full use of these symmetries through the use a sliding grid and the ability to select a geodesic by clicking on its apex. This allows for the drawing of any equirectangular line with ease and creates an avenue for full-featured spherical perspective drawing.

### 2.3 Finding the geodesic through two points

In the previous section we saw how to obtain a geodesic graphically using a sliding grid, a method that works both digitally and with traditional drawing media. When working digitally, however, a more efficient method can be used: one can draw the exact equirectangular geodesic that joins two arbitrary (non-antipodal) points specified by the user. This in turn allows us to draw a line segment  $AB$  from given points  $A$  and  $B$ . This is one of the more practical features of *Eq a Sketch 360*. Note that two point projections *do not* determine a single line, but they do determine a single geodesic, as well as the

segment of that geodesic that joins them. We will now show how to plot this geodesic. These calculations are not complex, being a matter of cartography, but seem to be unavailable in the literature.

Suppose then that we are given two point images on the sphere,  $P_1, P_2$ , not antipodal of each other, i.e., not diametrically opposite on the sphere. We wish to find the image of the line segment that joins the two points. Let  $P_i = (x_i, y_i, z_i)$  be the cartesian coordinates of the two points and  $P_i = (\lambda_i, \varphi_i)$  their angular coordinates, for  $i \in \{1, 2\}$ . Recall that the cartesian coordinates on the unit sphere are related to the angular ones by

$$\begin{cases} x = \cos \varphi \cos \lambda \\ y = \cos \varphi \sin \lambda \\ z = \sin \varphi \end{cases} \quad (2)$$

Since  $P_1, P_2$  are not antipodal to each other, there passes through them a single geodesic  $g$ . This geodesic is the intersection of the plane  $G = P_1OP_2$  with the sphere. For simplicity set the coordinates of the center of the sphere as  $O = (0, 0, 0)$ . Since  $G$  passes through the center of the sphere, then its equation is of the form  $ax + by + cz = 0$ . If  $G$  is a vertical plane then the solution of our problem is trivial (as vertical planes project to vertical lines in equirectangular perspective), so assume  $G$  is non-vertical. Then  $c$  is non-zero and we can write the equation of  $G$  as  $a'x + b'y + z = 0$  with  $a' = a/c, b' = b/c, c \neq 0$ . Then we have a system of two linear equations for  $G$ .

$$\begin{cases} a'x_1 + b'y_1 = -z_1 \\ a'x_2 + b'y_2 = -z_2 \end{cases} \quad (3)$$

The determinant  $x_1y_2 - y_1x_2$  is non-zero since the points are not in the same vertical plane. Then by Kramer's rule on (3), we get the solution

$$\begin{aligned} a &= y_1z_2 - y_2z_1 \\ b &= x_2z_1 - x_1z_2 \\ c &= x_1y_2 - x_2y_1 \end{aligned}$$

which expressed in angular coordinates through (2) is

$$\begin{aligned} a &= \cos(\varphi_1) \sin(\lambda_1) \sin(\varphi_2) - \cos(\varphi_2) \sin(\lambda_2) \sin(\varphi_1) \\ b &= \cos(\varphi_2) \cos(\lambda_2) \sin(\varphi_1) - \cos(\varphi_1) \cos(\lambda_1) \sin(\varphi_2) \\ c &= \cos(\varphi_1) \cos(\varphi_2) \sin(\lambda_2 - \lambda_1) \end{aligned}$$

Recall from section 2.2 that a non-vertical geodesic can be expressed as

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

where  $(\lambda_M, \varphi_M)$  are the angular coordinates of the apex point  $M$  of the geodesic, where it reaches its highest angular elevation  $\varphi_M$ . The apex determines the non-vertical geodesic fully, so we will calculate it in terms of the coordinates  $a, b, c$ , and this will determine the geodesic in terms of our two given points as intended.

The apex lies on the line defined by the intersection of  $G$  with the vertical plane  $V$  defined by the normal vector to  $G$ ,  $N = \text{grad}(ax + by + cz) = (a, b, c)$ . Since the vertical plane  $V$  intersects the horizon plane  $z = 0$  on the line  $\langle (a, b, 0) \rangle$  and

since  $z = -ax + by/c$  on  $G$ , then the line that joins  $O$  with the apex is generated by the vector  $\left(a, b, -\left(\frac{a^2+b^2}{c}\right)\right)$ .

Then, since the  $z$  coordinate must be the positive at the apex, the apex  $M$  can be written as  $M = \alpha \left(-\text{sgn}(c) a, -\text{sgn}(c) b, \frac{a^2+b^2}{c}\right)$  for some  $\alpha > 0$ , where  $\text{sgn}(x) = x/|x|$  is the signal function. Hence the angular coordinates of  $M$  are

$$\lambda_M = \text{atan2}(-\text{sgn}(c) (b, a)), \quad \varphi_M = \arctan\left(\frac{\sqrt{a^2 + b^2}}{|c|}\right)$$

where  $\arctan2(y, x)$  is the angle subtended between the ray to  $(x, y)$  and the positive  $x$  axis.

### 3. Eq A Sketch 360

#### 3.1 On current methods and software

Spherical perspective is having a renaissance, due in a large part to its association with VR visualizations. Each new element in the growing bestiary of spherical perspectives is a possible new avenue of visual expression. But while these choices expand, artists still struggle with the birth pains of a creative process adequate for the medium. This is especially true where digital media is concerned, as the options are hard to pin down, given both the scope of possible interfaces but also the specificities of target media, as panoramas may be destined to such various devices as phones, tablets, VR helmets or dome projections.

The current software used for drawing in equirectangular perspective uses two basic approaches (some software using both approaches in different degrees). The first is to draw in the flat perspective, the second to draw directly in the VR panorama.

The first approach is now available in standard image editing software like *Adobe Photoshop* or *Affinity Photo* that either natively or through plug-ins support special layer modes that allow one to draw directly on the VR view, on sections of the visual sphere. Some plugins add domain specific features, like *360° Art*, aimed at spherical perspective comics, that uses the *cubemaps* rather than equirectangular projections. The strength of this approach is also a limitation. The drawing is intuitive, as it happens in *local* linear perspective sections: you draw what you see. But by bypassing spherical perspective, fracturing it into multiple local views, you lose access to *global* constructions. For instance, you can no longer see vanishing point pairs all at once. This matters both for practical reasons and because the spherical perspective drawing is itself an object of artistic study.

The other type of approach, drawing on the flat perspective view, can also be achieved in image editing programs, but there are dedicated programs such as Microsoft's *Sketch 360* [15] or *Journalist* that integrate the perspective drawing view

with a VR preview window, to offer visual feedback and create more of a sketchbook-like process adequate for sketching on location (so-called *urban sketching*). The drawing process is grid-based, following the same technique artists have been using for handmade drawings, using a fixed grid of verticals and of horizontals going to four reference points. By drawing over the grid the user may draw any objects made up of such verticals and horizontals. But all other lines are obtained by guesswork. That means that diagonals will not in general be drawn in the right way – they will look curved in VR – unless they are painfully boxed in. There is no way to draw an arbitrary segment between two given points or a line going to an arbitrary vanishing point with good precision. Hence proper perspective constructions will be very hard to achieve.

In both approaches there is an attempt at using spherical perspective while studiously avoiding treating it as a perspective as such. This is understandable from a usability standpoint, when apps are aimed at non-specialists, but it also leads to a black box mentality [16]. The developments mentioned in the previous section cannot be used, first because the software doesn't implement them, second because the average user wouldn't know what to do with them. Luciani et al. [17] reporting on the challenges faced by designers sketching for full dome presentations, note that whatever the choice of sketching interface, designers had trouble dealing with the unintuitive aspects of spherical perspective. They speculate on what new media may help to sidestep these difficulties. While agreeing that these strategies are adequate for the non-specialist, and while sharing the enthusiasm for new interfaces, the present author believes that, for any artist with more than a casual interest in immersive drawing, the strategy should be to teach the perspective as such, so that it *becomes* intuitive, using software that makes that learning as simple and rapid as possible, and that later facilitates a more sophisticated and informed artistic process based on that acquired understanding of the principles. In short, the author thinks the most exciting role of VR is to *create "software" in the mind* of the artist, not in acquiring new gadgets. Accordingly, *Eq A Sketch 360* aims at being *human developing* software, a notion we will discuss further in the concluding section.

#### 3.2 Implementation and Operation of Eq A Sketch 360

*Eq a Sketch 360* is implemented as a *Processing 3.0* sketch. The user is presented with a drawing window on which an equirectangular grid is rendered (Figure 2), which can be translated horizontally using the arrow keys. A second smaller window provides instructions and angular coordinates of currently selected reference points and of the grid's displacement.

Given two reference points in the drawing,  $N$  and  $M$ , the user can draw the geodesic that joins them in one of two ways: 1) by sliding the grid until one of its geodesics matches the

two points, then choosing the apex of that geodesic. 2) by simply selecting the two points when in snap-to mode. Then when the user draws a line with the mouse or pen, the line will snap to the exact geodesic that joins  $A$  to  $B$ . The following commands can be issued by the user through the keyboard:

$n/m$ : point selection. Two reference points  $N$  and  $M$  are kept in memory. Pressing the corresponding key stores in memory the angular coordinates of the point over which the cursor is currently hovering.

$e$ : equirectangular snapping mode. When this is set and the user draws on the screen (with pen or mouse) the drawn line will be constrained (snapped) to the equirectangular geodesic joining the two reference points  $N$ ,  $M$ . Pointwise, whenever the user presses the mouse button on point  $(x, y)$  corresponding to angular coordinates  $(\lambda, \varphi)$ , the program will plot point  $(\lambda, \varphi(\lambda))$  where  $\varphi(\lambda)$  is obtained by equation 1 for the apex  $(\lambda_M, \varphi_M)$  defined by the reference points  $N$ ,  $M$ . The calculation of this apex from the reference points is explained in section 2.3.

$f$ : freehand drawing mode. In this mode the user can freely sketch with his mouse or pen as in any drawing program.

*Left/Right arrow*: slide the geodesic grid left/right by one degree.

*Up/Down arrow*: slide the geodesic grid left/right by five degrees.

$g$ : show/hide the geodesic grid.

$x$ : Set apex at the mouse location. This is useful when training the use of the sliding grid. Defining the apex identifies the geodesic without having to place two specific points. Given two points already drawn the user slides the grid horizontally until finding the single geodesic that joins them. Then pressing  $x$  with the mouse on the apex of the geodesic selects it as a *snap-to* guide for further drawing.

$v$ : draw verticals. This will set a vertical guide on top of the point where the mouse is located. Further drawing in this mode will be constrained to this vertical.

$s$ : quicksave the display window (transparent background, with grid removed).

$l$ : load a previous sketch to continue the work.

*Ctrl-z/Ctrl-Shift-z*: undo/redo.

$c$ : select line color.

$e$ : eraser.

*Shift-n/Shift-m*: selects as point  $N/M$  the antipode of the point at the current position of the mouse.

### 3.3 Features and roles of *Eq A Sketch 360*

*Eq A Sketch 360* was born both from a personal need and as a response to needs expressed by other developers and artists to address the inadequacies of current software we discussed above.

The utility of the snap-to command to draw an exact line between two points is easily understood. Its desirability was expressed to this author in private communication by several researchers, among them the author of Microsoft's *Sketch 360*

who discussed with the present author the problem of sending lines to a fixed vanishing point. That problem is a special case of the one we have solved here. Since it is probably in the to-do list of various developers, the present paper should prove useful.

In tests with a small group of immersive drawing practitioners, with the authors own research group and with workshop students, this feature has proved a great time-saver and a way to do previously unattainable constructions. It was previously far easier to do perspective illustrations by hand with a sliding grid on tracing paper than in any existing software, if precision was required. The program has proved of great value in the author's own artistic work, especially as part of a pipeline, using ordinary sketching software for the later detailing stages once the perspective scaffolding is done.

The utility of the sliding grid/apex selection method needs a lengthier explanation. While teaching the formal perspective techniques of [3] in workshops the author found it necessary to do live drawings with speed and precision, showing the process in a projector. Existing software proved inadequate for the task, and projecting handmade drawings is awkward. The sliding grid/apex selection method corresponded exactly to the handmade process and therefore allows a speedy and precise execution of demonstrations that workshop participants can execute on paper, following along with the projection of the teacher's live drawing. This was first tested in a workshop at the Universidade Federal do Pará, Brasil, in June of 2019. The workshop's contents were deliberately exactly the same as in a workshop performed in the previous year in Bridges Stockholm [3] and it was found that the use of the program greatly eased the task for both instructor and students, in particular allowing for a far quicker instruction.

But the software's didactical use goes beyond its role as a demonstration tool. In order to perform the handmade perspective constructions, one requires two types of training. One, of a conceptual nature: it is necessary to understand the geometry and the constructions themselves. One, of a mechanical, physical nature: one must learn to use the grid to identify geodesics. This last act is non-trivial with the grid and tracing paper method. In teaching experience it was found that this was perhaps the hardest part to teach in a limited time. The eye loses itself in the forest of geodesics and it is hard to keep the drawing grid and tracing paper properly aligned and to not get lost in what at first seems like an optical torture. It takes days of training to do this naturally. To do both types of tasks at the same time is not very efficient.

*Eq A Sketch* separates the two tasks by abstracting the physical one, allowing the student to concentrate on the constructions. It assumes the ability to find the geodesic and trace it. Then the artist who wishes to draw by hand can train the physical task separately, already with a sound foundation on the constructions themselves, speedily acquired through the use of the software.



Further, the sliding grid/apex method is useful even for users who will draw exclusively in digital form. It is necessary for the user to understand how the more efficient two-point method works, and to develop the intuition to *predict* what result it will give in each case. Drawing lines through arbitrary points and then finding them with the sliding grid develops intuition and prevents a black box mentality. Otherwise the user wouldn't have a clue of why a line has the form it does as the calculations of the two-point method are *opaque* to the user.

In fact the program can be used as a feedback loop for training the memorization and intuition regarding the shape of equirectangular lines. If a draughtsman is to draw in equirectangular perspective intuitively enough to sketch without the grid of geodesics, it will be necessary to gain an intuition for the shape of the general geodesic joining two given points. This is a task that at first requires slow reasoning and then progressively becomes faster, using a mix of reasoning and memory (including muscle memory). The best way to acquire this habit is to isolate it into a feedback loop. The user is given two random points and tries to join them by sketching freely. Then the correct line is revealed, and the attempt is compared to it, leading the eye and brain to correct the guess. The quicker and cleaner the feedback loop can be repeated the best for such a learning task. Eq a Sketch 360 optimizes this process by removing the physical obstacles that are not essential to it. These physical aspects have also to be dealt with, but the two aspects are *separable* and gain by the separation. The aim here is to help develop, as efficiently as possible, abilities that will remain with the artist even when the software is discarded.

### 3.4 Next Steps

Since the software is currently an early development it hasn't yet been widely presented to the general public, but only to limited groups. First because it still has many bugs and unstable features, but also because its nature implies that it gains from being presented with careful tutorials if an instructor is not available. First steps have been made with the university of Pará, Brasil, do co-develop a didactic tool and interactive book incorporating a javascript version of Eq a Sketch. We believe this will allow the useful dissemination of the program.

Feedback from the limited number of test subjects has been positive, especially among the few artists who are already familiar with the principles of perspective, and therefore immediately see the use of the features. The main feature suggestions from these tests have been:

- 1) to incorporate a VR preview in the program itself. Currently we use a separate program to do visualize the results.
- 2) to add the usual features available in ordinary drawing programs, such as a proper layer system, good brushes, etc.
- 3) to make the program vectorial rather than raster based.

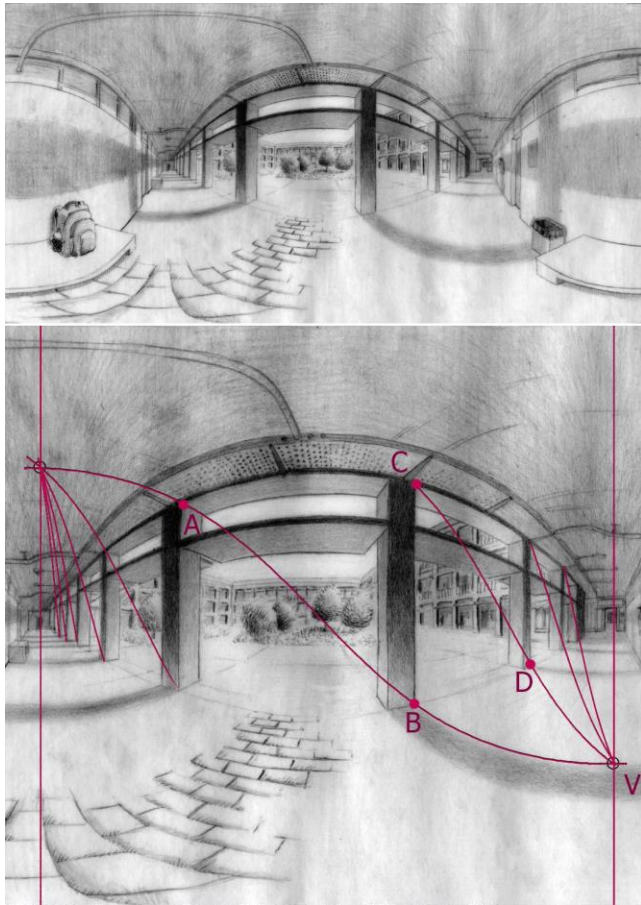
Request 1 was already in the to-do list. Request 3 is intended but only as an option, as raster seems more in line with a sketcher's sensibilities. The many features implied in 2) will be tested as we go along. Efficiency has to be balanced with simplicity if *Eq A Sketch 360* is to keep its essence as a *serious toy* rather than lose it under feature overload.

### 3.5 An Example of a Construction

Let us consider an example of an exercise using *Eq A Sketch 360*. It is the fundamental construction for such drawings as the one of Figure 3 (top), where there are repeating elements such as equally spaced columns. This construction was demonstrated by hand in a Bridges workshop in Stockholm [3] in 2018, using the sliding grid on tracing paper method, and later using *Eq A Sketch 360* at the University of Pará, Brasil, for comparison under similar conditions. The original image of Figure 3 was made by hand using the tracing paper method, with graphite on a cropped A4 sheet. The drawing was sketched on location, but only the closest arch was measured from nature, the others being obtained through segment multiplication, assuming the columns to be uniformly spaced. This is analogous to what is sometimes called the *telephone pole* problem in classical perspective. This construction can be executed in *Eq a Sketch 360* through the following operations.

Consider on Figure 3 (bottom) the two adjacent columns closest to the observer. Assume we have measured the angular coordinates of two points *A* and *B*, respectively located at the top of the left column and at the bottom of the right column. We obtain the position of the next column to the right by the following multiplication method [3]: Pass a geodesic through *A* and *B* and prolong it until it finds the vanishing point *V* of *AB*, which must be on the vertical line  $90^\circ$  to the right since *A* and *B* are on a frontal plane. Now pass a line from point *C* (at the top of the second column) to *V*. Since the columns are equally spaced, then the lines *AB* and *CV* are parallel, and both vanish at *V*. Then line *CV* must hit the ground line at the bottom of the next column. Thus we find point *D*. The process may be repeated recursively, thus finding all the columns. The same can be done for the columns on the left side of the drawing.

Note that besides the construction itself, the software is also useful for analysis of already existing constructions such as the present drawing. For instance, placing the mouse on top of the *V* point already present in the drawing of Figure 3 will give us the angular measurements of that point, from which we get the true angle of the original spatial diagonal *AB* with the horizon plane.



**Figure 3: Top: Equirectangular perspective drawing of a corridor. Graphite on A4 paper, sketched on location by the author. ©A. B. Araújo. Bottom: Geometric analysis of the arch constructions using *Eq A Sketch 360*. All arches are found from the central one through geometric multiplication of lengths. The horizontal space between the verticals through A and B is multiplied by finding the vanishing point V of the diagonal AB.**

#### 4 A digression in place of a conclusion: What is good software?

What is good software? Is it the same as efficient software? In many respects, *Eq A Sketch 360* is not efficient software. It lacks many of the typical features of a modern drawing program, such as layer effects and brushes. Like the *Etch A Sketch* toy, it forces you to think in the simplest of terms. This is deliberate. The point is to force you (while helping you) to learn spherical perspective, its geometric properties and specific constructions. Rather than hiding these constructions behind friendly interfaces and menus the program forces you to either learn them or do anything at all. It is a call to understand spherical perspective rather than having it done for you at the press of a button. In this sense it fits with a *cardboarding* philosophy [16] of deliberate technical

simplification as means of exposing the theoretical innards of software black boxes.

So the program is (I hope) good software in the sense that it aims to develop the inherent abilities of a person rather than merely enhancing the ability of the person to produce work through immediate use of the software. I will call *human developing software* to software that pushes people towards developing abilities that would remain present *if the software should cease to exist*. In short: good software should leave you stronger when you're done using it. Much software that we use has the opposite effect - it is degrading of abilities we already have. It enhances our immediate productive abilities *while in use*, but If we should stop using it, we would find ourselves weaker than before, our former abilities blunted. Much software indeed aims at replacing humans at their tasks, which is good when they are the boring, soul-crushing ones. But every day we hear that a computer can now play chess (or some other pleasurable, interesting activity) better than us, and yet we can't help noticing that little progress has been made on robots that do the dishes or pick up the trash. Guess then what jobs will be left to humans! Of course, most software can be human developing or human degrading according to the use we give it, though design tends to at least strongly lean one way.

For instance, P. Wozniak's *SuperMemo* (and free competitors, such as *Anki*) is an example of software that is strongly *human developing* by nature. Based on the *spaced repetition* studies of the 1930s regarding long-term memory [18, 19], its only purpose is to enhance the brain's long term memory, a benefit that remains if the software should later vanish.

*Wikipedia* is mostly human developing: it is so when used to learn and relate new facts quickly; it can be human degrading only if used as an excuse to memorize nothing since it is always there when needed. Some theorists of education argue that data availability makes memorizing useless. This is a mistake, as the mind cannot work on empty; creative connections cannot happen in a mind devoid of facts.

*Google Maps*, though potentially human developing if used to learn and memorize new routes, is mostly used in a human degrading way, when we lazily allow it to take us from A to B without paying attention, letting our memory and orientation skills devolve from lack of use. The most intellectual aspect of a taxi driver's work was the encyclopedic knowledge of the streets. A GPS dependent Uber driver, reduced to a machine-like job of merely operating a vehicle, is a step away from eventual replacement by self-driving cars. Human degrading software is often a spearhead for human *displacing* software. Consider the tendency of turning office jobs into more and more mechanical bureaucracies. Making the job more machine-like makes the public more accepting of a lack of nuance and flexibility in their interactions with functionaries. The functionary, bereft of decision power, becomes machine-like, spitting out a "does not compute" to any form that isn't



filled out precisely by the rules. There's the lazy road to AI: it is easier to make humans think like machines than it is to make machines think like humans. Take this far enough, and none will complain when the human is replaced by a computer barely more rigid – at least the machine has an excuse! The industrial age did not start with machines but with division of labor: making each artisan machine-like. Only then could he be replaced by actual machines.

The author's hope is that *Eq A Sketch 360* will prove a piece of software that leans in the way of human development, improving the artist's understanding of perspective rather than just another app for making pretty things whilst unmaking people. The author's greatest hope for the current renaissance of spherical perspective is that it serves as an incentive not just to buy new apps and gadgets, but to learn new ways of thinking and drawing.

*Eq A Sketch 360* can be found at [20].

## ACKNOWLEDGMENTS

I was sitting for quite a while on the result of section 2.3 and its application. My thanks to Michael Scherotter of *Microsoft*, the author of *Journalist* and *Sketch 360*, and a fellow urban sketcher, whose questions pushed me to get on with it and write it down. I hope Michael and other developers will find the result useful for their own software and art.

I was financially supported by Portuguese national funds through FCT project UID/Multi/04019/2013.

## Version Notes and Corrigenda

This author's version of the present paper dates from 31 May 2020 and differs from the published version by the following correction:

On page 4, the published version gave the following formula for  $c$ :

$$c = \cos(\varphi_2) \cos(\lambda_2) \sin(\varphi_1) - \cos(\varphi_1) \cos(\lambda_1) \sin(\varphi_1)$$

There was a typo in the last factor of the second term, which should be  $\sin(\varphi_2)$  and not  $\sin(\varphi_1)$ . This is corrected in the present version, which now reads, as intended,

$$c = \cos(\varphi_2) \cos(\lambda_2) \sin(\varphi_1) - \cos(\varphi_1) \cos(\lambda_1) \sin(\varphi_2).$$

## REFERENCES

- [1] W. S. Ed Sobey. 2008. *The Way Toys Work: The Science Behind the Magic 8 Ball, Etch A Sketch, Boomerang, and More*, Chicago Review Press.
- [2] A. B. Araújo. 2018. Drawing Equirectangular VR Panoramas with Ruler, Compass, and Protractor. *Journal Of Science And Technology Of The Arts* 10, 1 (April 2018), 15-27. DOI: <https://doi.org/10.7559/citarj.v10i1.471>
- [3] A. B. Araújo. 2018. Let's Sketch in 360°: Spherical Perspectives for Virtual Reality Panoramas. In *Proceedings of Bridges 2018: Mathematics, Art, Music, Architecture, Education, Culture*, 637-644. Retrieved September 2018 from <http://archive.bridgesmathart.org/2018/bridges2018-637.pdf>
- [4] A. B. Araújo. 2018. Ruler, compass, and nail: constructing a total spherical perspective. *Journal of Mathematics and the Arts*, 12, 2-3 (June 2018) 144-169. DOI: <https://doi.org/10.1080/17513472.2018.1469378>
- [5] M. Kemp. 1990. *The Science of Art*. Yale Univ. Press, New Haven and London.
- [6] E. Huhtamo. 2013. *Illusions in motion - media archaeology of the moving panorama and related spectacles*. (1st ed.). The MIT Press, London.
- [7] A. Barre and A. Flocon. 1964. *La perspective Curviligne*. Flammarion, Paris.
- [8] A. B. Araújo. 2016. Topologia, Anamorfoses, e o bestiário das Perspectivas curvilíneas. *Convocarte*, 2 (September 2016), 51-69.
- [9] F. R. Casas. 1983. Flat-Sphere Perspective. *Leonardo* 16, 1 (1983), 1-9.
- [10] M. Moose. 1986. Guidelines for Constructing a Fisheye Perspective. *Leonardo* 19, 1 (1986), 61-64.
- [11] G. Michel. 2013. L'oeil, au Centre de la Sphere Visuelle. *Boletim da Aproved*, 30 (April 2013), 3-14.
- [12] A. Rossi, S. Barba and L. F. Olivero. 'CubeME', a Variation for an Immaterial Rebuilding. In *Rappresentazione / Materiale / Immateriale. Drawing as (in)Tangible Representation*, 31-36. Rome, 2018. Retrieved September 2018 from <http://hdl.handle.net/11591/392282>
- [13] L. F. Olivero, A. Rossi, and S. Barba. 2019. A Codification of the Cubic Projection to Generate Immersive Models. *disegno* 4 (June 2019), 53-63. DOI: <https://doi.org/10.26375/disegno.4.2019.07>
- [14] J. V. Correia and L. Romão. 2007. Extended Perspective System. In *Proceedings of the 25th eCAADe International Conference*. Frankfurt, 185-192.
- [15] M. S. Scherotter. 2018. Sketch 360. (Nov. 2018). Retrieved September 2019 from <https://www.microsoft.com/en-us/garage/profiles/sketch-360/>
- [16] A. Araújo. 2017. Cardboarding Mixed Reality with Dürer Machines. In *5th Conference on Computation, Communication, Aesthetics & X*. Univ. do Porto, Lisbon, 102-113. <http://2017.xcoax.org/xCoAx2017.pdf>
- [17] D. T. Luciani and J. Lundberg. 2016. Enabling Designers to Sketch Immersive Fulldome Presentations. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, California, 1490-1496. DOI: <https://doi.org/10.1145/2851581.2892343>
- [18] C. Mace. 1932. *The Psychology of Study*. McBride, Oxford.
- [19] H. F. Spitzer. 1939. Studies in retention. *Journal of Educational Psychology*, 30 (1939), 641-657.
- [20] A. B. Araújo. *Eq A Sketch 360*. Retrieved September 2019 from <http://www.univ-ab.pt/~aaraujo/eqasketch360.html>