Mathematizing Pictures:
Sketchpad's Toolkit for Digital Imagery

Nicholas Jackiw
njackiw@kcptech.com
www.dynamicgeometry.com
KCP Technologies, Inc.

1 Introduction

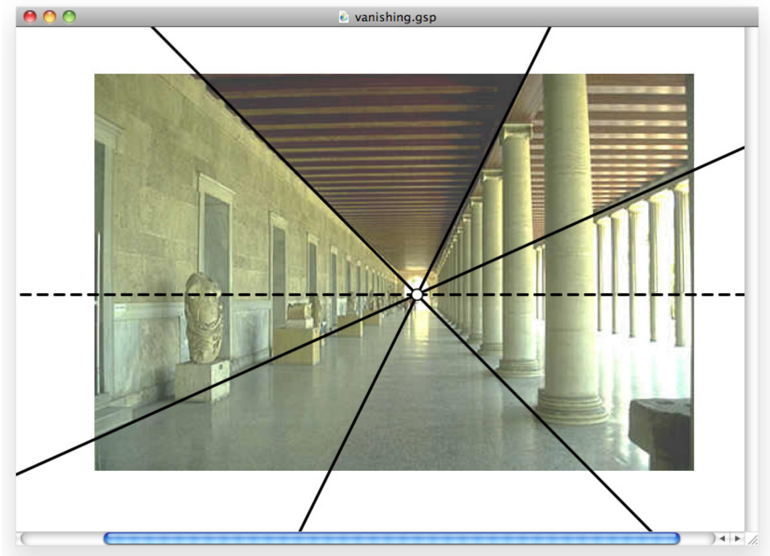
Where the traditional role of imagery in mathematics has been as mathematical *output*—with "visualizations" of mathematical and scientific processes and concepts providing elegant results, aesthetic motivation, and occasional intellectual insight to practitioners[i]—more recent developments also conceive of the image as appropriately a mathematical *input*: an object to be analyzed, measured, dissected, and (re)constructed through mathematical inquiry and technique. Computational tools for mathematical analysis and modeling have embraced this new potential in both educational and research settings. For the school learner, interactive geometry environments such as *The Geometer's Sketchpad*[ii] provide possible settings for exploring mathematical applications of photographs, line art, and other digital imagery, and for connecting those applications to curricular topics in geometry and algebra. Whether such settings actually lead to significant mathematical picture applications depends in some degree on how convenient and conducive the operations defined by the software environment are to such a content application domain. *Sketchpad's* picture functionalities—manipulation and analysis operations—have evolved over the past several major versions of the software, and with the recent Version 5 release compose what might be considered a mature set of operations for working with digital images in the context of Dynamic Geometry. Where my ATCM talk[iii] ("Images and the Geometric Imagination") will focus broadly on the mathematics of some of these applications and connections, in this paper I draw attention to the specific technology tools—the software features and operations—that permit such mathematical modeling within *Sketchpad* Version 5.

1.1 Background

The context for this work is *The Geometer's Sketchpad* computer software, a general-purpose mathematical modeling environment with an emphasis on visual forms and expressions of mathematical contents, relationships, and constructions. In its signature functionality, it enables the dynamic manipulation (with a mouse or pointer) of the position, shape, or value of component of a mathematical construction or derivation while preserving all related definitions, and while interactively visualizing these definitions' "behavior" in response to the user's manipulation. The term "Dynamic Geometry" was coined to describe this process in *Sketchpad*,[iv] and the process has been recognized as having strong impact in mathematics and mathematics education settings[v] as a concrete and accessible intellectual scaffold on which learners and practitioners can build or refine conceptions of such as mathematical ideas as *variation* ("how does this shape move?"), *generalization* ("what happens to the system as a vary this component?"), and *abstraction* ("how do I characterize the system regardless of this component's variable aspect?"). Since these ideas are in a mathematical sense profound, they occur in various forms across the curriculum, and therefore *Sketchpad* has found wide use in mathematics contexts from kindergarten through undergraduate and research mathematics. In the United States, its primary application is in Grades 8-10 geometry and algebra classes, but its spread beyond that grade range is broad.

Sketchpad began in the mid-1980's with the (American) National Science Foundation's Visual Geometry Project at Swarthmore College, Pennsylvania. School versions have been in continuous publication by Key Curriculum Press since the early 1990's. Similar technological ideas to *Sketchpad*'s were developed independently and contemporaneously in France, by Jean-Marie Laborde, where they have led to the *Cabri Géomètre* software family.[vi] *Sketchpad* and *Cabri* are among the most widely-used educational mathematics programs in the world, and have inspired an entire industry of so-called "interactive geometry" programs. These include products such as *Cinderella* and *Geometry Expressions* that significantly enhance or alter the mathematical or educational foundations of *Sketchpad* and *Cabri* with other theoretical constructs, as well as numerous open-source efforts that shamelessly take only the elements of *Sketchpad* or *Cabri* easiest to duplicate and assemble them in constructs that have little or no design integrity or theoretical cohesion.[vii]

Within its workspace, *Sketchpad* has offered some form of support for digital images for many years. Version 2 of the software (1993) permitted static digital images to be imported as workspace "backdrops" and short Quicktime clips embedded in documents for viewer playback. This early support was originally intended and applied primarily for "illustrational" purposes—that is, simply to enhance the cosmetic presentation of a mathematical document, or to permit some visual referent to, or explanation of, an application topic. In this view, an image (or video) is related to, but somehow separate from, the core mathematics of a problem, activity, or exploration. However, in that *Sketchpad*'s fundamental workspace is (a computer simulation of) the unbounded geometric plane, the natural affinity between that plane and digital images—which can be mathematically considered differentially-colorized $\Re \times \Re$ planar subspaces—began to assert itself, as users began to superimpose geometric constructions *on top* of digital images, using the "illustration" as some sort of dataset that could visually be virtualized as, or measured by, a mathematical construction.



In the simple example at left, a digital image of corridor has been imported into *Sketchpad*, and the user has constructed various lines "on top of the image" that follow or extend sightlines in the image (the foot of the columns, the edges of the roofbeams). By extending these geometric sitelines to their common intersection, the user locates the "vanishing point" of the one-point perspective scene, and with it, the location of the theoretical horizon line (dashed).

The set of insights and novel user-generated problem domains drawn from early *Sketchpad* picture experimentation caused iterative refinement of various software functionality supporting digital imagery, most recently and significantly in *Sketchpad* Version 5 (2010). While within the larger scope of *Sketchpad*'s overall mathematical features and functionality, the role of digital imagery is relatively small, nonetheless it forms a coherent and technologically-interesting subdomain of the software's utility. (The types of mathematical image manipulations possible with the software are simply impossible through any other reasonable means or educational technology.) The present incarnation of that evolving software functionality is the subject of this paper.

2 The Mathematical and Pedagogic Uses of Digital Imagery

While this paper's purpose is to describe in a technically-specific language that functionality that supports work with digital imagery in Sketchpad, and justify that design within the discourse of instructional design, I touch here briefly on the mathematical and pedagogic *purpose* of this functionality. In other words, setting aside the mechanics of *how* you do it, *why* do you do it?

The premise of this work is that pictures connect to geometry in a variety of mathematical topics (such as measurement, transformational geometry, coordinate systems & graphing, complex analysis, topology, etc.); in real-world applications of geometry (such as computer graphics, crime scene analysis, medical imaging, etc.); and in our educational objects (such as fostering students' sense of mathematical aesthetics). Pedagogically, topical, relevant, and—in many cases—personal imagery can significantly reframe students' perception of mathematics problems and activities from that of dry and soulless material into highly-motivating personal challenges. Mathematically, contemporary applications (such as computer graphics) not only animate a sense of engagement between the mathematical domain and the world in which they live, but also lead to the development of new and interesting mathematical material that is often (at least within computer graphics) within the scope of an interested secondary-school student.

These theoretical or rhetorical motivations appear to be borne out in practice. Studying student and teacher work collected internationally over a year-long evaluation of *Sketchpad Version 5* in many schools, I have elsewhere[viii] described real users' mathematical picture work in detail, and proposed an epistemological framework that categorizes that work according to both an increasing sense of mathematical engagement and, for students, and escalating sense of mathematical agency. As argued earlier, since the study of the plane and its mathematical "denizens" recurs (at various levels of sophistication) across the school mathematics curriculum, opportunities to engage planar-embedded digital images are numerous and broad spread. Areas of particular curricular resonance with *Sketchpad* Version 5's picture functionality—curricular areas where we see the most topical work—include:

- at the elementary/middle school border, the study of basic shapes and symmetries, and their respective properties (many of which inhere in digital images as well) and transformations;
- in secondary school algebra/trigonometry, with numerous data-motivated curve-fitting and measurement opportunities;
- at the upper secondary / lower undergraduate level, in complex analysis, wherein an image's color dimensions can "solve" the graphing problem occurred in attempting to visualize spatially the four-dimensional context of a $C \rightarrow C$ complex mapping.

- at the undergraduate / research level, in image analysis and image synthesis (where applications tend to be didactic rather than goal-directed; *Sketchpad* does not have the scalability to large problem-sets of tools like MATLAB (for analysis) or Renderman (for synthesis).

3 Picture Functionalities

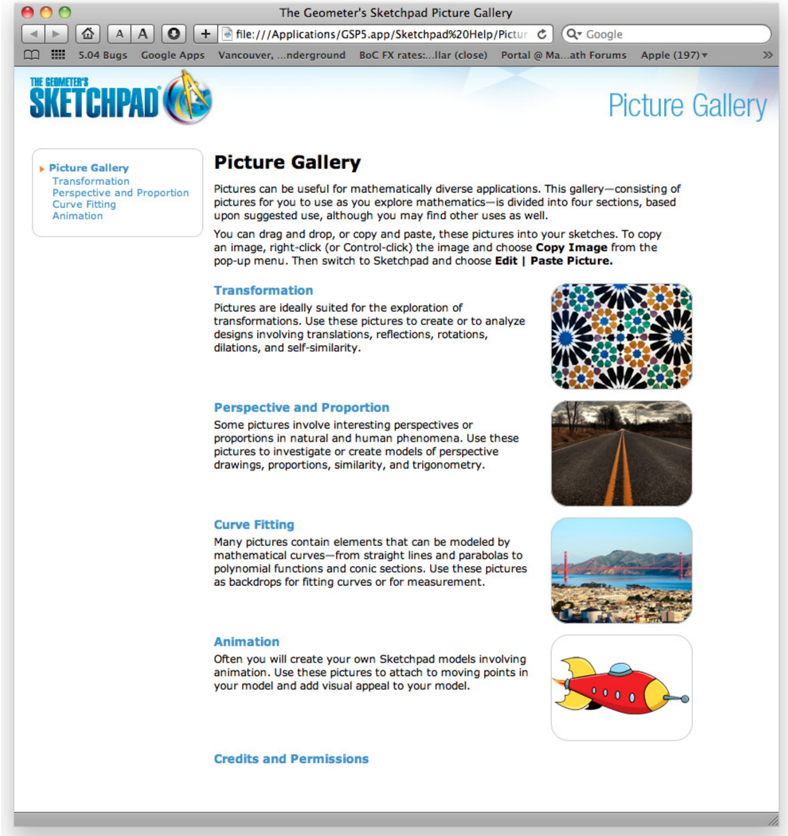
This section explains the context and considerations leading to the specific form of *Sketchpad* Version 5' functional support for digital images. While the features discussed appear to progress from more didactic considerations to more mathematical ones, in practice the earlier features wind up being more important to *all* mathematical investigations using digital imagery while the latter ones may only apply to specific types of mathematical inquiry.

3.1 Accessing Pictures: Convenience and Inconvenience

Sources of digital imagery are now ubiquitous. The camera-enhanced mobile phone has replaced the low-end "standalone" digital camera as the most popular image-capturing device. Beyond what any teacher or collection of students could reasonably produce, the internet offers a flabbergasting array of conveniently-accessible pictures, that can be searched for any conceivable topic specified by keywords[ix] or even by drawing a picture resembling the photograph you'd like to find.[x] From any of these sources, importing a digital image into *Sketchpad* is as simple as dragging a desired image directly from your photo browser or web browser into any open document.

But sometimes the accessibility of internet imagery can be *too* convenient for the school environment, where students' encounters with inappropriate and obscene internet photographs causes havoc whose repercussions routinely reach beyond the classroom. Even choosing desirable and apparently appropriate photographs from the internet for student project work can lead students unknowingly into intellectual property and copyright infringement areas, if they borrow images not explicitly licensed for such borrowing or put, by their authors, into the public domain. This is but one instance of a larger gray area (affecting not just photographs, but also text and music) prompted by the newfound ease of access to, and perfect duplicability of, digital media: if the internet delivers this content "free" to my desktop, how wrong is it to use that content, especially for educational purposes? Some educators view this as a moral gray zone, and ignore intellectual property concerns over materials they're duplicating for their own classes; others view it as an opportunity to foster dialog with students about personal and social ethics. (Our own position is that one need not enter moral "gray zones" here; international law around these rights is unambiguously black and white; and violating those laws should be strongly discouraged by social institutions such as schools.)

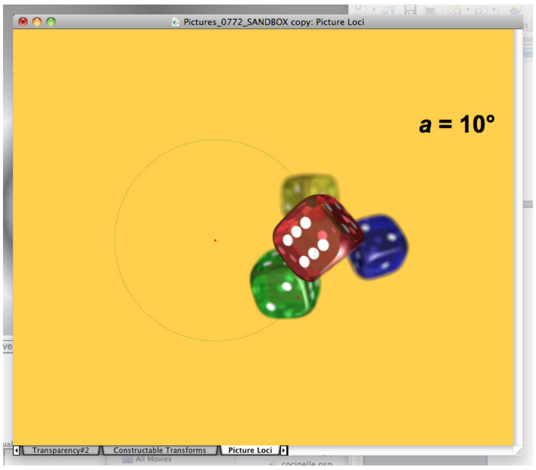
To allow teachers to sidestep both of these issues, *Sketchpad* provides a *Picture Gallery* of curated digital images that are provided with no licensing restrictions on copying or reuse. Accessed from the Help menu, the images in the Gallery are evocative of, and loosely sorted into, various categories of possible mathematical application: transformation, perspective and proportion, curve fitting, and animation. Again, drag-and-drop or copy-and-paste are sufficient to move a picture from the Gallery into Sketchpad's workspace.

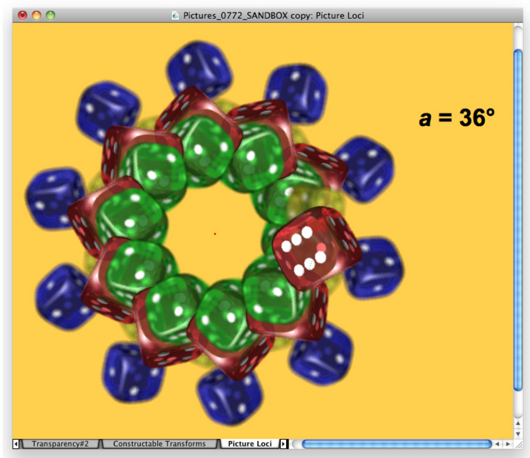


3.2 Breaking Out of "The Box:" Transparency and Cropping

Working with digital pictures as mathematically-significant shapes or regions within the geometric plane rapidly reveals a non-obvious shortcoming of the medium, which is that by default, *most pictures*—regardless of their visual content—*are rectangles*. Since the computer screen itself is rectangular, this attribute is hardly a limitation when pictures are used as backgrounds for "draw-over"-type mathematical investigations (such as calculating distances or trip routings on top of the digital image of a map), but becomes more significant in contexts where the image is intended as an arbitrary region or area. Sketchpad offers two alternatives to the traditional rectangular digital photograph in its support for *alpha transparency* and *image cropping*.

Alpha transparency is a digital image effect applied algorithmically or manually or image editing software, in which certain pixels within an image's traditional rectangular border are defined as completely transparent or partially translucent. (Essentially, the picture is digitally reformatted to specify not only color but also opacity information for each pixel; then some subset of pixels are declared non-opaque.) When portions of a picture abutting its rectangular frame are declared transparent, the resulting image loses its rectangular appearance with respect to the background or to any other images which it overlaps or on which it is stacked. Sketchpad supports full-alpha transparency where it exists in any imported digital picture, allowing non-rectangular applications and explorations. (In the example below, a mandala *right* is created from a partially-transparent image *left* through rotations of 36°.)

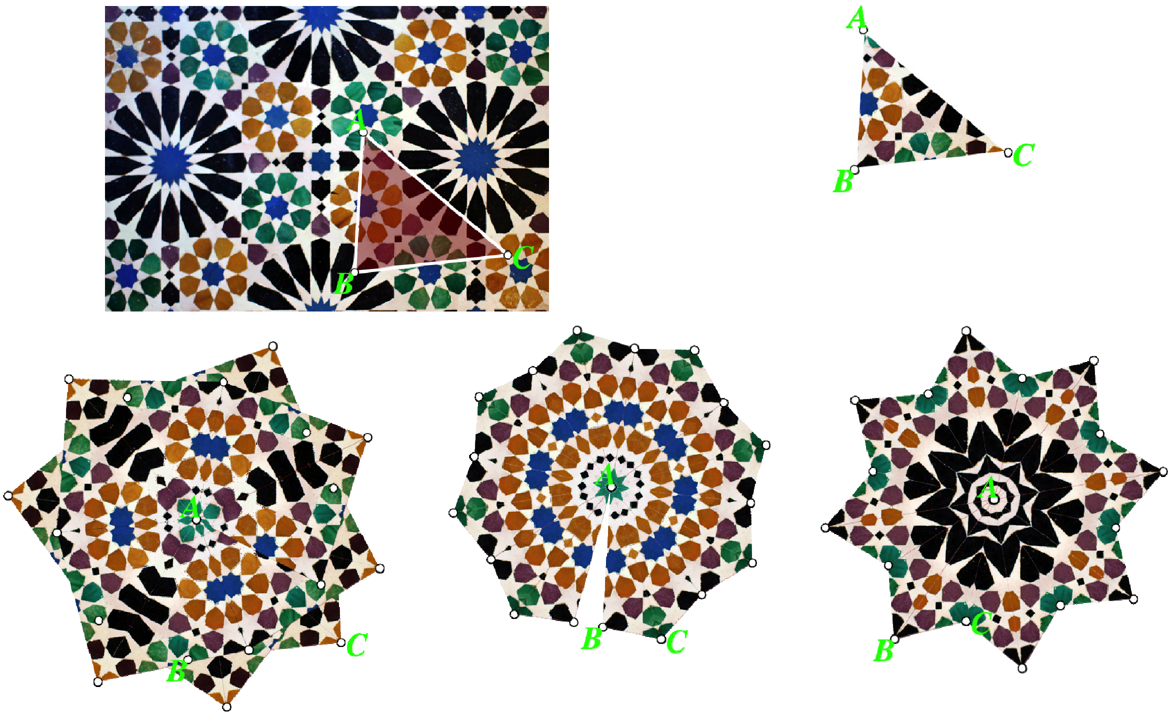




While alpha-transparency is never present in "raw" digital photographs—since it must be applied by hand or by algorithm in a post-processing step—nonetheless such partially-transparent images are common both on the web and where digital photographs have been run through various image effects software (such as tools to isolate a human face or head from its photographic backdrop).

Inside *Sketchpad*, users can easily manipulate alpha-transparency for the picture as a whole, causing a highly-saturated image to appear less bright, or to allow a stacked set of images to become sufficiently translucent that one can see through the "foremost" image to one of the "background" images. These operations are cosmetic in nature, although they can be useful in foregrounding a superimposed geometric construction in danger of being "lost" against a noisy picture backdrop.

If alpha-transparency is an image processing technique, *picture cropping* offers a more geometric technique for creating non-rectangular images. Picture cropping as an operation defines a new picture as the intersection of an existing picture with an arbitrary planar polygon. (Cropping is available in *Sketchpad* through **Edit | Crop Picture To Polygon**.) The result is a picture shaped like this intersection, containing the subset of the original digital image visible within it. The sequence below illustrates the process of isolating a non-rectangular subregion of a picture through cropping, and then manipulating that new subregion geometrically. (*Top-left*: a rectangular photograph and a triangular polygon, before cropping; *top-right:* the new triangular image resulting from their intersection; *bottom row:* a kaleidoscopic set of reflections of the triangular image, dynamically visualized for several shapes of the original triangle.) Note that the image contents of the cropped result vary dynamically as the cropping polygon moves dynamically with respect to the rectangular image it crops.

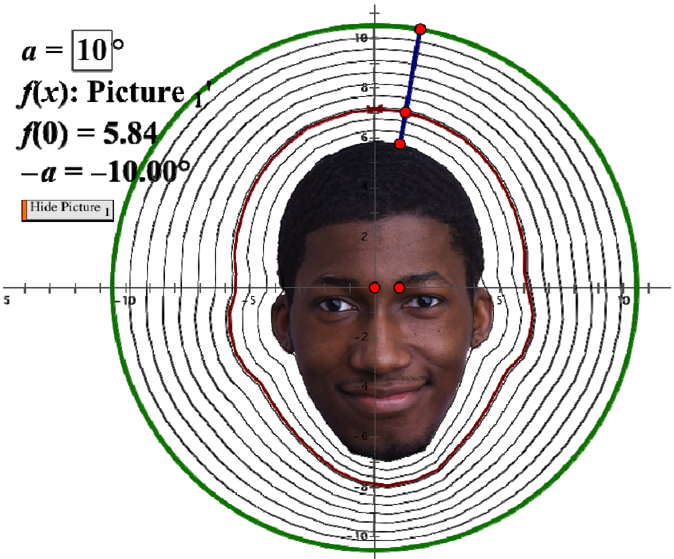


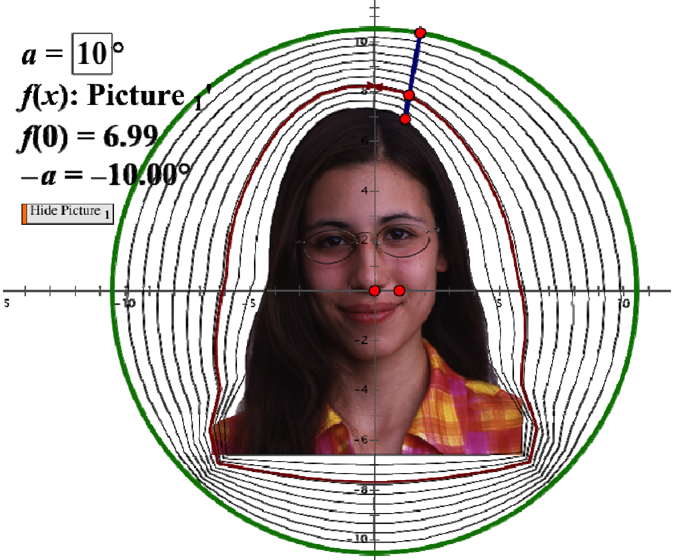
In addition to providing non-rectangular tiles for various symmetry tilings, picture cropping allows users to isolate specific forms of interest within an image (faces, buildings, etc.) based on a bounding contour.

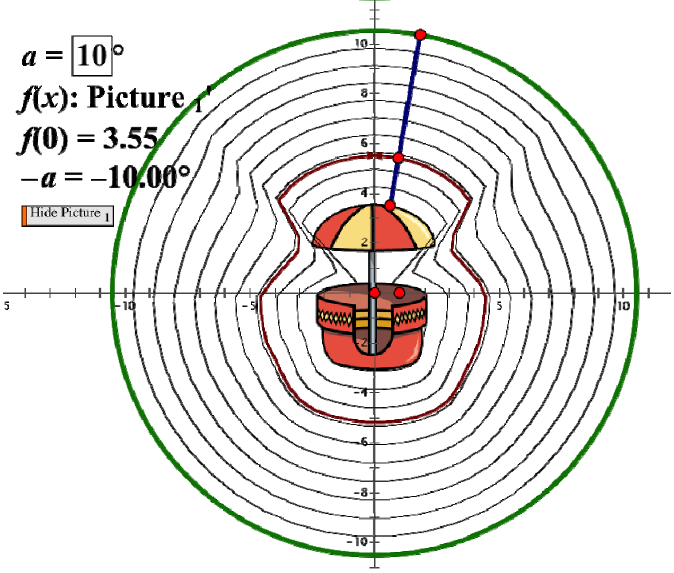
3.3 Picture Replacement

If—as the premise of this paper asserts—pictures have inherently mathematical properties and therefore, potential broad mathematical interest from the Dynamic Geometry perspective, a natural question arises: how can we think of pictures "dynamically?"

Two obvious answers to this question involve image synthesis and image replacement. If a dynamic software environment can synthesize digital pictures, then dynamic elements of a mathematical construction (e.g. animating points or parameters, dragged objects) can in turn have dynamic impact on the synthesized image. I discuss this possibility later (see *Transformations*). Similarly, consider for a moment the quintessential Dynamic Geometry operation—of dragging a vertex, or changing a parameter—as a generalized "object replacement" operation (replacing one triangle with the next, or with a sequence of next triangles; replacing one parameter value with the next, or with a sequence of nexts). In this formulation, the pedagogic dividend of Dynamic Geometry comes from exploring the behavior of a set of mathematical dependencies under the dynamically variation—or systematic replacement—of some originary object or objects.







Applied to digital pictures, the idea of dynamic replacement suggests the benefit of exploring a mathematical dependence constructed on one image with a replaced new image or set of images. This potential perhaps reaches its fullest in the idea of replacing a single image with a video feed, while observing the effects of a constructed analysis on the ever-changing imagery of the video stream. Unfortunately, *Sketchpad* Version 5—and the hardware on which it turns—is not yet sufficiently fast to apply mathematical constructions of arbitrary complexity on all the pixels of a digital image at the data-throughput required by real-time video. As a slower—and perhaps more intentional—mechanism for changing an image while holding its analysis constant, however, Sketchpad supports the **Paste Replacement Picture** operation (in the **Edit** menu), which allows any selected image to be replaced by another from the clipboard, while dynamically updating any mathematical dependencies of the original picture.

The examples at left model by geometric construction an edge-finding algorithm from image processing. (The central images are digital photographs; the curves surrounding it are closed planar curves single successively approximating the photograph's radial contour based of sampling the transparency—or lack thereof—in the picture itself.) By updating the central image with pasted replacement photographs, the second and third examples explore the construction's behavior on different problem sets—ultimately demonstrating problems with the mathematical approach when treating non-convex images (*bottom*).

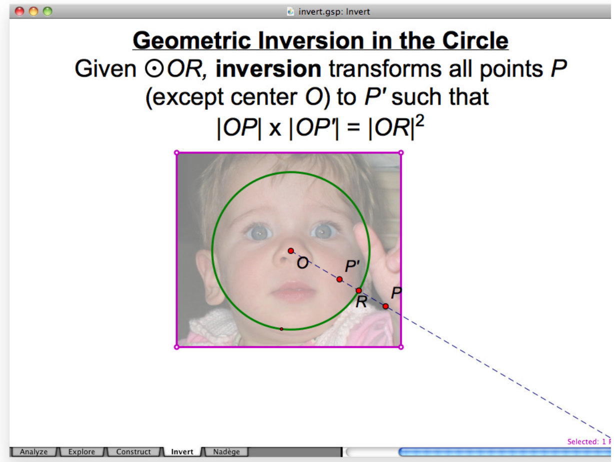
3.4 Transformations

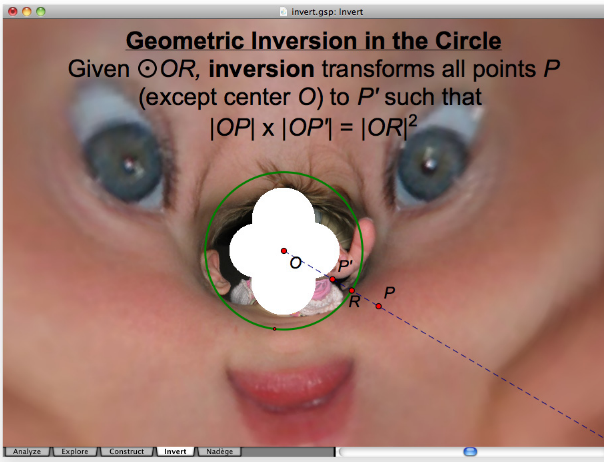
The heart of *Sketchpad*'s picture operations are defined in transformational terms, and address how images are positioned or replicated in the plane. Here the software defines an entire hierarchy of different transformational vocabularies intended to suit the needs of different levels of user, or of single users at different times in their interaction trajectory with the software.

- **Informal Freehand Transformations**. Through these mouse-mediated actions, pictures can be dragged and resized (or stretched) by dragging their corners. *Sketchpad* specifically avoids mathematizing these operations, which are familiar to most users from other programs (such as in common word processing and image editing programs).
- **Formal Freehand Transformations**. Using *Sketchpad's* various Arrow Tools, users can translate, rotate, or dilate existing images, changing their position, orientation, and size. The transformations are still "freehand" in the sense that they are applied dynamically by the mouse, and so transformational parameters such as "vector of translation" and "angle of rotation" are specified visually by mouse gestures rather than by precise numeric parameters. At the same time, these operations are given a formal mathematical name and definition within the software, and require the user engage with explicit mathematical formulations and operations to indicate supplemental transformation parameters (such as the center point of rotation or dilation), or to constrain the mouse gesture mathematically (e.g. to limit its translation to horizontal or vertical vectors; or to limit its rotation to increments of 15°). Note that *all* geometric objects in *Sketchpad* may be manipulated by formal freehand transformations; pictures inherit like behavior simply by virtue of their status within the environment as geometric objects themselves.
- **Constructions by Formal Transformation.** Using the **Transform** menu, users can construct transformed images of selected pre-image pictures (and other objects) under precisely-specified isometries of translation, rotation, and reflection; as well as by (non-isometric) dilation. In each case, transformational parameters are specified either by numeric or symbolic quantities ("rotate by 15°", "translate by *m* cm," "scale by a factor of 1/*p*," etc.), or by indicating equivalent geometric relationships in or between existing objects within one's construction ("rotate by this angle," "translate by the length of this segment," "scale by the ratio of these two lengths."). The result (as illustrated in Section 3.2) is a constructed copy of the original pre-image under the specified transformation, which is dynamic to changes in the transformation's underlying parameters (e.g. dilation factor 1/*p* when *p* changes from 2 to 3), to changes to the pre-image picture's position (via dragging), and to changes to its digital contents (e.g. via **Paste Replacement Picture**). Transformed pictures can themselves be transformed, allowing sequences, series, tessellations, and similar pattern-making constructions.
  Again, *Sketchpad* defines these operations for *all* geometric objects—including digital pictures. Facility with the most elementary of these formal constructions (reflection, which has the fewest supplemental parameters and in some senses the simplest definition) seems to be within the reach of average kindergarten students, and the more advanced formal constructions (which require greater reading and reasoning skills) within the abilities of typical 4^{th} and 5^{th} graders.
- **Formal Constructions by Affine Transformation.** *Sketchpad* Version 5 also supports a general-purpose affine transformation of digital images in which three of the corner vertices of a rectangular digital picture can be transformed to three arbitrary distinct geometric points in one's sketch. While not a common subject of secondary-level

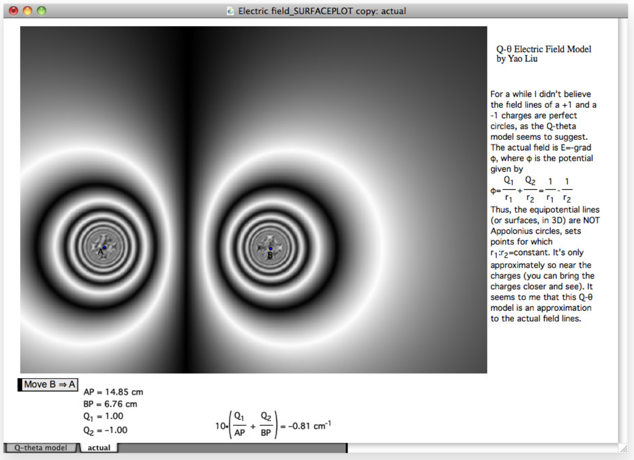
geometry investigations, exposing an affinity transformation beyond the earlier transformations permits shearing, and—more importantly—neatly permits conceptual bridge building between the various component transformations (reflection, dilation, rotation, shearing) considered discretely, their compositions, and the vector and matrix objects of linear algebra.

- **From Transformations by Constructed Example.** At the high end, *Sketchpad* Version 5 permits any construction in which a (specifically indicated) dependent point somehow depends on a (specifically indicated) free point to serve as an *example-based* definition of a *custom transformation*. (In the user-constructed example, the free point serves as an "example" pre-image and the point on which it depends as the corresponding "image" of that point under the demonstrated transformation.) Such transformations may be arbitrarily complex. If the example construction is limited to lines, their endpoints, and intersections, the types of transformations one can exemplify are generally projective; but if one permits full compass-and-straightedge constructions, there is no restriction to linearity. Indeed, since *Sketchpad*'s fundamental construction offer many non-Euclidean possibilities (including access to a fully dynamic, numeric calculator), these transformations may well be non-linear, non-conformal, and even non-analytic. Unlike earlier transformational offerings within this hierarchy, therefore, "custom transformations" need not preserve the overall measurements or even rough proportions of an image, as demonstrated in the example below, in which *inversion from the circle*, defined by the example of a simple geometric construction, is applied (at *right*) to a pre-image photograph (at *left*).





Color Transformations. Moving outside the realm of classical geometry entirely, and into the more modern field of image processing and computer graphics, Sketchpad also permits users to construct and apply color transformations rather than positional (or "planar") transformations. If a transformation—such as inversion, demonstrated above—that affects only position can arbitrarily distort an existing image, a transformation that controls both and color can selectively redefine, or entirely replace, an existing image with a newly synthesized one. Functionality to manipulate color is not explicitly presented within a transformational vocabulary in the program, but instead is located elsewhere in the interface. Specifically, for color output, the software's Parametric Color feature allows a (potentially calculated and potentially dynamic) numeric value to specify the color of an object. For color input, in prototype, the Color of a Point operation measures the color of a given digital image at a specific (and potentially dynamic) point's location as a numeric value. Operationally, this gives users a closed loop in which the color of an existing image can be measured as a number, then fed into calculations or derivations or constructions or algorithms that produce new numbers, which in turn can be reprojected back into the actual color domain of the original (or positionally transformed picture). This combination opens up the full application domain of image analysis, as well as the simpler task of image synthesis, as in the example at right, in which a student has constructed a dynamic planar surface plot using parametric color to animate equipotential behavior under the Q-θ model of electric fields.



This hierarchy of types of transformations is deep, and certainly most users do not progress all the way through it in terms of the Sketchpad operations and activities that they find productive and engaging in classrooms or research labs. Anecdotally, it appears that almost all users can—and do—successfully navigate the software to the level of constructions by formal transformation while very few carry on to affine transformations or transformations by custom example—even though the latter offers some of the richest potential for open-ended modeling and diverse types of output within the entire software system. In addition, this discussion should make clear (by example!) the clash between meanings of the word "image:" does it reference a visual image (as opposed to, say, audible music), or does it refer to a transformational image (as opposed to, specifically, a transformational pre-image)? For clarity, unlike this paper, Sketchpad itself reserves the term "image" to only, and always, to have transformational connotations, and refers to all digital images, instead, consistently as "pictures."

3.6 Other Functionalities

Certain other functionalities exist in *Sketchpad* that affect digital images and what users can accomplish with them. You can construct the locus of an image across some constructed motion, and you can iterate a picture whose position is defined by construction. Again, these are behaviors *Sketchpad* defines for all mathematical objects, and their application to digital images is not particularly dinstictive. In a more novel direction, you can define a function by a picture that represents its graph on some coordinate system—and then evaluate, differentiate, and iterate that function. However, this functionality targets "line art" digital image—especially line art created with the Marker tool, directly in Sketchpad—rather than the (usually imported, usually full-color, often photographic) bitmapped conception of "digital image" used elsewhere throughout this paper.

4 Conclusion

This paper has surveyed the entire complex of software support for mathematizing digital imagery in *Sketchpad* Version 5, and each function has been described in terms not only of its user operation, but from the software design perspective, in terms of the original motivation to, and practical application (and occasional unintended consequence) of, that technical operation. While some of these features are more directly or intrinsically mathematical than others—e.g. the ladder of transformational vocabularies, compared to the capacity for alpha transparency—the latter often address necessary preconditions critical to any successful mathematical engagement with the visual image in a computer environment.

5 Notes and References

[i] For example, see representative mathematical image galleries hosted online by the American Mathematical Society (www.ams.org/mathimagery/) and the Math Forum (mathforum.org/mathimages/index.php/Main_Page).

[ii] Jackiw, N. (1991, 2010) *The Geometer's Sketchpad* (computer software). Emeryville, California: Key Curriculum Press (Version 1, 1991—Version 5, 2010).

[iii] Jackiw, N. *Imagery and The Geometric Imagination* (plenary address), 16th Asian Technology Conference in Mathematics (ATCM 2011), Abant İzzet Baysal University, Bolu, Turkey.

[iv] "Dynamic Geometry," U.S. Registered Trademark No. 1997870.

[v] J. R. King & D. Schattschneider (1997) (Eds.), *Geometry turned on!: Dynamic software in learning, teaching and research.* Washington, DC: The Mathematical Association of America.

[vi] Laborde, Jean-Marie et al. (1989, 2007), *Cabri Géomètre* (computer software, multiple versions), Grenoble: CabriLog.

[vii] For a long list, see http://en.wikipedia.org/wiki/List_of_interactive_geometry_software.

[viii] Jackiw, N. and Sinclair, N. (2009) "Sounds and pictures: dynamism and duality in Dynamic Geometry, " *ZDM*, Volume 41, No. 4, 413-426.

[ix] For example, by Google Image Search (www.google.com/images).

[x] For example, using tools from the Princeton Shape Retrieval and Analysis Group (shape.cs.princeton.edu/search.html).