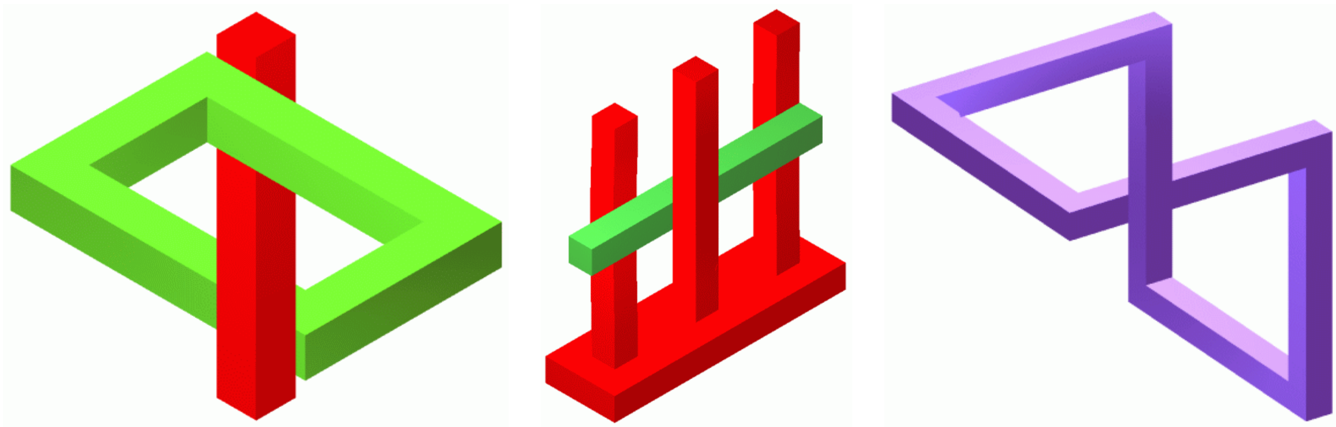
Modeling (Seemingly) Impossible Models

Gershon Elber*
Dept. of Computer Science, Technion – IIT, Haifa 32000, Israel



(a) (b) (c)

Figure 1: *Three examples of seemingly impossible models (SIMs) that are realizable as 3D tangible objects using the modeling paradigm presented in this work.*

Abstract

In recent years, there has been a growing interest in computer graphics and geometric modeling in the ability to create and manipulate Seemingly Impossible Models (SIMs). Methods to create derivative work and modify drawings and paintings of SIMs made by artists were suggested. Similarly, 3D realization of SIMs of some of these drawings were also offered.

In this work, we further explore the nature of SIMs and identify a class of SIMs that can be realized and modeled in 3D. As part of this analysis we show an invariance whose preservation allows one to model SIM artifacts that are completely realizable. We further present a mini-modeling package that allow end-users to realize whole new SIMs in two stages: modeling a regular 3D model and then converting it into a SIM using special deformations. We conclude with some examples, a discussion on the current limitations, and a layout of possible future work.

keywords: Impossible 3D Models, M. C. Escher, Non realistic modeling (NRM), Art in science.

1 Introduction and Previous Work

Drawings of impossible models are never-ending intriguing pieces of art that captivate and intrigue the beholder. The immense public interest in the drawings of Maurice Cornelius Escher [Escher] probably stems from the sensation of the impossibility that these drawings generate in the spectator. In other words, these drawings build upon the spatial training of the human mind and trick the observer into seeing impossible scenarios. The desire to create tangible 3D models that mimic these drawings leads to the following definition:

Definition 1.1 *A tangible 3D model will be considered a* Seemingly Impossible Model *(SIM), if it leads, from at least one specific viewing direction, to a seeming contradiction in the presented scene.*

The "Belvedere", "Ascending and Descending, and "Waterfall" drawings of Escher all belong to this family of SIMs, and they are all realizable as tangible 3D objects. Several authors have presented such 3D realized SIM artifacts from wood and plastic [Elber] and even Lego® pieces [Lipson]. These tangible 3D artifacts mimic, from one specific viewing direction, 2D scenes that are seemingly contradictory. Focusing on a single view, there are others who were able to (manually) realize tangible SIMs. The work of Sugihara [Sugihara 2007] stand out in this direction, work that recently was also awarded the best-illusion-of-the-year contest [1].

Kupla [Kupla 1987] and Sugihara [Sugihara 1982] before him tried to classify the SIMs into different classes of impossibilities, to possible, impossible, likely and unlikely. Object-background contradictions, depth-estimation contradictions and similar measures are used in the process. Also interesting is how the realized 3D tangible SIM is made. For instance, are we required to introduce cuts into the model? An interesting claim is made by [Kupla 1987] that "all impossible figures have possible interpretation - all impossible figures are possible". While one can argue this claim and what does 'possible' means, in this work, we aim at creating real tangible 3D models that look like the SIM from one view, the view from which they are typically painted. Having this requirement, not every drawing of a SIM is indeed realizable as a 3D model.

Oriented compact 2-manifolds in $\mathbb{R}^3$ delineate space into interior and exterior sub-spaces and therefore serve as a boundaries that separate the two. The Jordan theorem [DoCarmo 1976] states that we must cross the boundary of the oriented 2-manifold, when we move from the inside out or vice versa. With that in mind, we introduce a simple test that must be satisfied in order to achieve a tangible 3D SIM:

*e-mail: gershon@cs.technion.ac.il

[1] See http://illusioncontest.neuralcorrelate.com

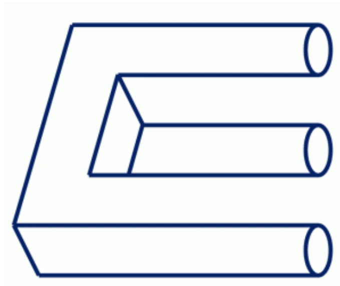


Figure 2: *A well-known impossible model that has no interior and no exterior and hence cannot be realized as a tangible 3D object.*

Condition 1.2 *A SIM that has a tangible 3D realization, must pass the Jordan curve theorem test.*

The drawings of one class of impossible models allow a connected path in the drawing plane to go from infinity (outside the model) into the model, crossing no boundary and hence violating the Jordan theorem's condition (Condition 1.2). The tribar drawing in Figure 2 is one such example. Such drawings cannot have tangible solid 3D models that realize them, not even from a single viewing direction, and are excluded from our discussion here.

In [Savransky et al. 1999], the depth misperceptions in a drawing are mapped to detections of cycles in the (depths of the objects in the) scene graph. A Z-buffer based rendering is then employed to render the elements of the scene in the desired order, a process that is quoted to require "considerable user intervention". [Wu et al. 2010] also examined the depth misperceptions. Given an initial drawing of a SIM, they are able to synthesize nearby novel views of the input.

Indeed, depth misperception is one of the most compelling features in drawings of SIMs. While a large portion of previous work synthesize derivative 2D drawing of SIMs, some efforts were directed at generating 3D models that look impossible from one specific view. Of this class of 3D SIM generators, many are manually made, and are typically based on a 2D drawing of some SIM. In contrast, in this work, we seek to go a step further and create a *modeling environment that will allow end-users to build entirely new valid tangible 3D SIMs* by deforming regular tangible 3D geometry in a special way. Consider, for example, Figure 1 that shows three SIMs that are realizable as tangible objects using the modeling environment we offer as part of this work.

The rest of this work is organized as follows. In Section 2, we present the basic ideas and portray several ways of creating SIMs, focusing on the special deformations and assuming the input 3D geometry is readily available via the means of regular 3D geometric modeling techniques. Conceptually, any regular 3D geometry can be exploited as the input to create SIMS. Specifically, we use polygonal 3D models that are refined to small enough polygons so that the deformed output continues to appear smooth. We further employ the GuIrit [GuIrit] modeling environment to create all input models. Examples of the introduced deformations are presented in Section 3, and finally, we conclude in Section 4. All the examples presented in this work, but Figures 2, 14, and 13, were created using the presented approach and implementation.

Figure 3: *A SIM of a cycle of three bars that hide parts of each other. The SIM is created by exploiting two human brain misperceptions. The first is the creation of an apparent cycle in the depth ordering and the second is that seemingly orthogonal angles and straight lines in the drawing plane are indeed orthogonal and linear in* $\mathbb{R}^3$.

2 Algorithm

While depth misperception is a major feature that is exploited in many drawings of SIMs, it is also a clear degree of freedom in creating new SIMs. When one examines artistic drawings of SIMs, the limited capability of humans to capture depth (or Z) is revealed.

Humans are excellent at detailed understanding of information in the (XY) plane. Yet, our minds aim to interpolate and complete missing details in Z. The human natural stereo-vision allows us to gather some limited depth information in real life but it is of no use when inspecting 2D drawings of SIMs. A small person is perceived as farther away compared to a larger one. Similarly, if object $\mathbf{A}$ is (even partially) hidden by object $\mathbf{B}$, our brain naturally assumes that $\mathbf{A}$ is behind $\mathbf{B}$.

Hence, it is difficult for us as humans to interpret *a loop in the depth ordering* of some objects in a drawing. Such difficulties arise when, for instance, object $\mathbf{A}$ hides object $\mathbf{B}$, object $\mathbf{B}$ hides object $\mathbf{C}$ and object $\mathbf{C}$ hides object $\mathbf{A}$ (i.e., Figure 3). To complicate the situation even more, we are trained to anticipate that a straight line in the drawing plane originates from a straight line in space and to expect that the spatial angle between two faces that look orthogonal in the 2D drawing is indeed 90 degrees in 3-space. Figure 3 shows one such example, taking advantage of these mis-interpretations.

Armed with the knowledge about genetic evolution and the outcomes of mind training, we are now ready to fool the eye, using the following crucial observation:

Observation 2.1 *Consider object* $\mathbf{A}$ *that is seen from eye position* $\mathcal{E}$*. A deformation of* $\mathbf{A}$ *along the line of sight from* $\mathcal{E}$ *(aka projection lines or* projectors*) does not change the shape of* $\mathbf{A}$ *as seen from* $\mathcal{E}$*. We denote such deformation a* Line of Sight Deformation (LoSD).

Observation 2.1 is crucial to our discussion. A change (i.e. deformation) in object $\mathbf{A}$ along the projectors does not change the drawing of $\mathbf{A}$ as seen from $\mathcal{E}$! Hence after, and for simplicity of explanation, we will assume that the parallel projectors are only along the Z axis. That is, we are dealing with an orthographic projection.

The extension of the presentation to perspective projections is simple. Given a 2-manifold object A, we expect the result of the LoSD to be a 2-manifold. Hence, the LoSD should be injective near the 2-manifold. One should note that this condition does not exclude self-intersections.

Any LoSD we can apply to the model along projector lines parallel to the Z axes will not be reflected in the way the object is drawn (and seen) in the XY plane[2]. In other words, given object $\mathbf{A}$, we seek simple and intuitive a LoSD that will move point $P(x, y, z) \in \mathbf{A}$ to a new depth $\tilde{P}(x, y, z + \delta z) \in \tilde{\mathbf{A}}$, on the deformed object $\tilde{\mathbf{A}}$. This changes $\mathbf{A}$'s relative depth with respect to other objects in the scene.

We enumerate several possible LoSDs. Because the (x, y) position is invariant in all these transformations, we specify only the transform in the depth, or Z, of input point (x_1, y_1, z_1):

1. Global Z-skew transformation, $F_{skew}(z_1) = z_1 + f_{skew}(x_1, y_1)$. A simple example can be the linear transform

$$F^1_{skew}(z_1) = z_1 + y_1. \tag{1}$$

   Figure 5 shows a simple example of a depth skewing transformation applied to one of three bars, creating a well-known illusion called the Penrose triangle, from one view. However, this SIM is disconnected, an approach we rather avoid. Figure 6 shows a simple example of skewing that is applied to an entire mode, a cube. Note that the objects, in Figures 6 (a) and (c), looks very much the same from the original line of sight. Moreover, the impression that all faces are orthogonal is preserved in Figure 6 (c). Finally, because $F^1_{skew}(z)$ is linear, colinearity and coplanarity is preserved (while orthogonality is obviously not preserved).

2. Local Z-deformation transformation, $F_{deform}(z_1) = z_1 + f_{deform}(x_1, y_1)$. Here, a main question is how to define the *area or volume of influence* of F_{deform}. One possibility is based on the XY distance from a user-selected center-of-deformation location, (x_0, y_0), as

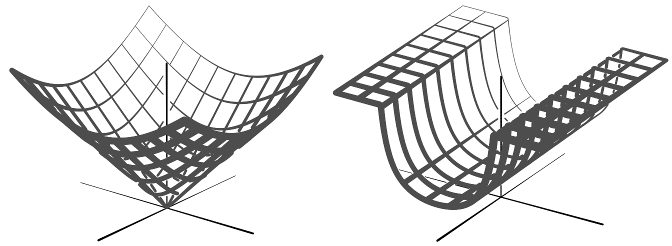
$$F^1_{deform}(z_1) = z_1 + \sqrt{(x_1 - x_0)^2, (y_1 - y_0)^2}. \tag{2}$$

   $F^1_{deform}(z_1)$ (see Figure 4 (a)) preserves the depth of the input model at (x_0, y_0) and increasingly deforms the space along the projectors as we move away from location (x_0, y_0), forming a cone of increasing deformation effect around (x_0, y_0). A second valuable possible alternative is a deformation along a *line* in the projection plane. Let $\mathcal{L}(x, y) := Ax + By + C = 0$ be some line in the XY plane. Then, given point (x_1, y_1, z_1) (see Figure 4 (b)),

$$F^2_{deform}(z_1) = z_1 + \frac{\alpha}{\max(d - |\mathcal{L}(x_1, y_1)|^k, \epsilon)}, \tag{3}$$

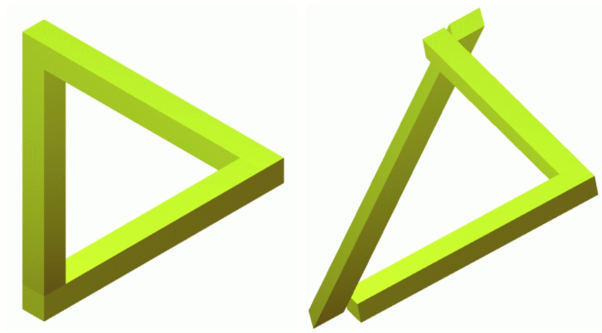
   $\epsilon,\ \alpha,\ d,\ k \in I\!R^+$, applies a depth deformation of amount α along $\mathcal{L}$ and as we move away from line $\mathcal{L}$ the amount of the deformation decays (unlike F^1_{deform}). Clearly one can control the amount and region-of-influence of the deformation and the rate of the decay using constants α, d, and k in Equation (3). Figure 7 shows a bar that underwent a few different $F^2_{deform}(z)$ deformations.

[2]Granted, the shading of the deformed object might be different in the drawn picture in the XY plane–a second order concern we will address at the end of the next section.



(a) (b)

Figure 4: *Plots of $F^1_{deform}(z_1)$ (Equation (2)) in (a) around the origin and of F^2_{deform} (Equations (3)) in (b) around the origin, with $\epsilon = 0.3$, $\alpha = d = 1$, and $k = 2$.*



(a) (b)

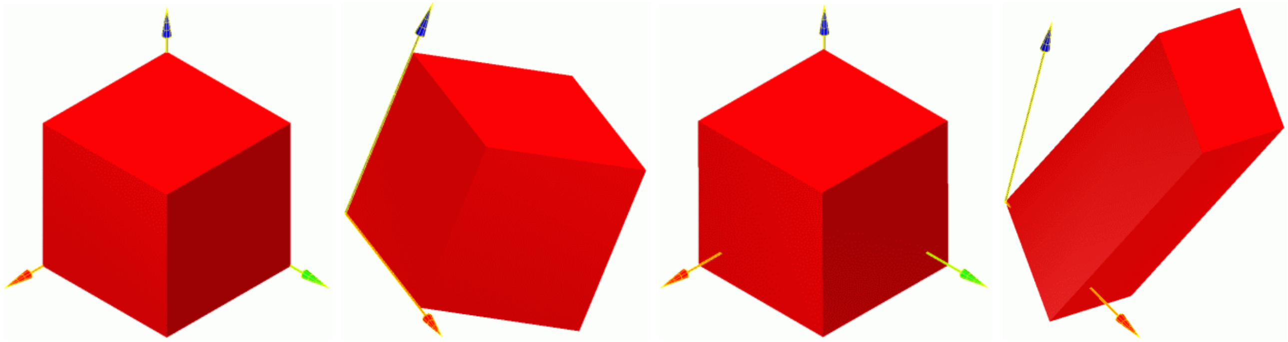
Figure 5: *Recreating the Penrose triangle (a) as a SIM using the application of a global skewing transformation, F^1_{skew}, to one of the bars. Note the evident shading discontinuity at the bottom left corner in (a). (b) shows a different view of the same scene.*

There are many variations on F_{skew} and F_{deform}. The transformation can be limited to a certain region in the XY plane or to a certain region in 3-space. It can also be limited to some sub-object parts (i.e. individual bars). The later is crucial when one bar in the object covers a second bar in the XY image plane and only the second bar is to be locally deformed. All these techniques as well as similar ones are classical geometric modeling deformation tools and are not part of the scope of this work. In the next section, we will demonstrate LoSDs and present examples of SIMs.

3 Examples and Limitations

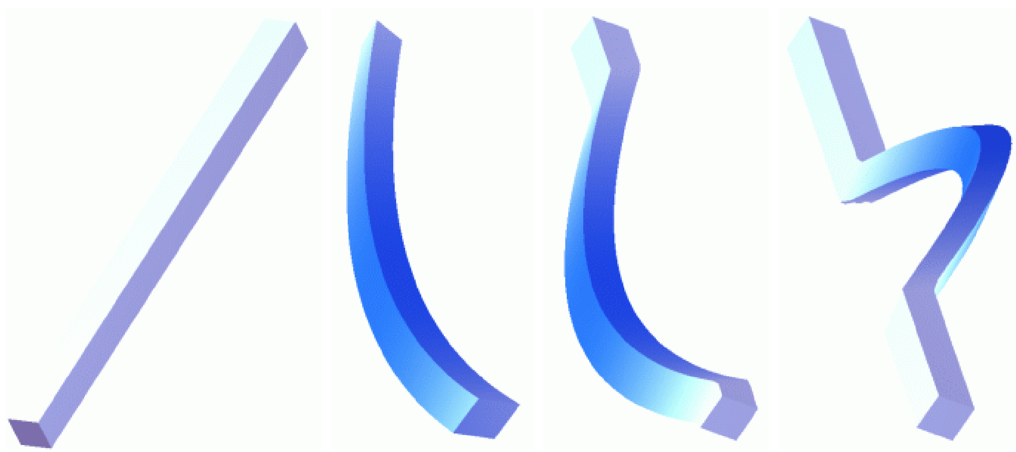
We start again with one of the most famous 3D models that is a SIM–the Penrose triangle. This 3-bars model creates the simplest cycle in depth, of three bars. Bar $\mathbf{A}$ hides parts of bar $\mathbf{B}$, which hides parts of bar $\mathbf{C}$, which, in turn, hides part of bar $\mathbf{A}$ again, creating the loop in the visibility graph. We can recreate the Penrose triangle as a SIM in several methods. Figure 8 shows two examples. In Figure 8 (a) and (b), the 3 bars are positioned parallel to the X, Y, and Z axes, creating the isometric view in Figure 8 (a). However, special cuts must be introduced to the end of the bars to complete the illusion in the SIM's drawing. Nevertheless, we can locally deform one of the bars, using F_{deform}, and create a real connected object as seen in Figure 8 (c) and (d).

We continue with the presentation of the three models shown in Figure 1, this time from general and different views. Figure 1 (a) is presented again in Figure 9 (a) from a different view where the global skewing transformation, F^1_{skew}, that is applied to both the green and the red parts is apparent, and where the orthogonality is



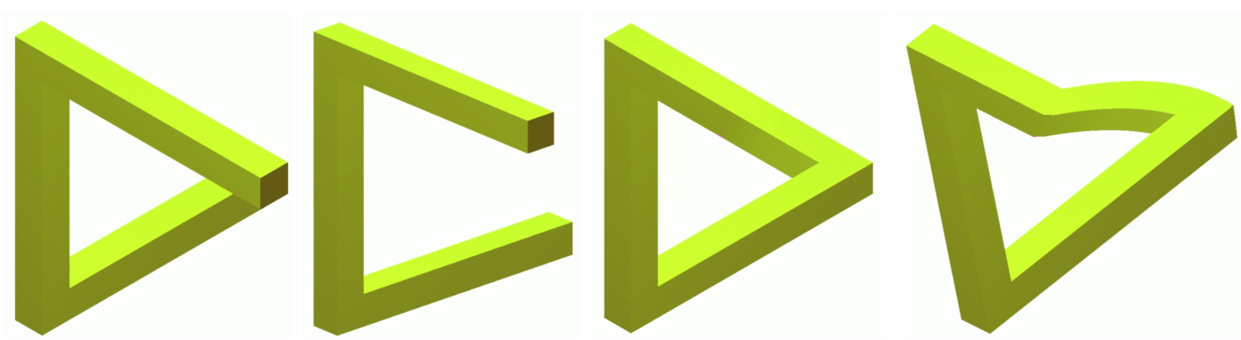
(a) (b) (c) (d)

Figure 6: *An example of global Z skewing transformation that preserves the (x, y) locations invariant, from one viewing direction ((a) and (c)). Note that (a) and (c) look almost identical whereas (c) underwent Z skewing transformation, F^1_{skew} (see Equation (1)). (b) and (d) show (a) and (c) from different views.*



(a) (b) (c) (d)

Figure 7: *An example of a local Z deformation transformation, F^2_{deform}, that is applied to a straight bar (a) and preserves the (x, y) projected image of the bar. (b), (c), and (d) shows three different local deformations (from a different view), all looking the same from the original line of sight, of (a). The dark blue color in (b), (c), and (d) is a user-interface cue to the region-of-influence.*

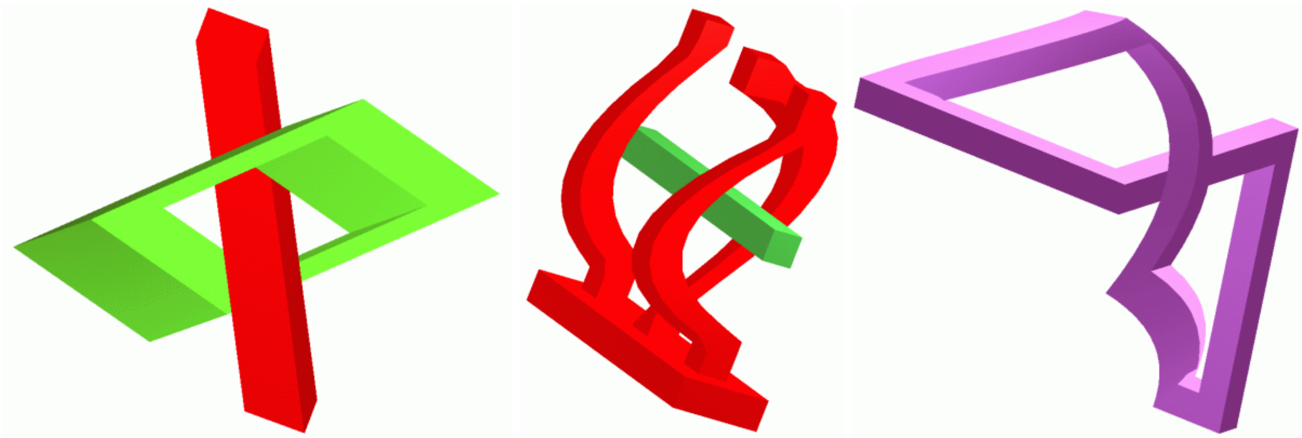


(a) (b) (c) (d)

Figure 8: *Recreating the Penrose triangle as a SIM. (a) and (b) show three bars parallel to the X, Y, and Z axes, in two different views, so they almost look like the Penrose triangle from the isometric view of (a), up to one open end. In (c) and (d), a local deformation, F^2_{deform}, is applied to one of the bar. (c) is the same isometric view as in (a), creating a SIM that looks perfect, whereas (d) again shows the model in (c) from a general view.*

not really preserved. Figure 1 (b) is presented again in Figure 9 (b) from a different view where local LoSD transformation, F^2_{deform}, is applied to the different bars. Finally, Figure 1 (c) is presented again in Figure 9 (c) from a different view where local LoSD transformation, F^2_{deform}, is applied to the original back side, bringing it to the front.

The best-illusion-of-the-year for 2010, by K. Sugihara (see footnote on page 1), is modeled in Figure 10. Herein, a conical deformation is applied around the center of the model, following F^1_{deform} in Equation (2). Figure 10 (a) shows the scene from the illusion's



(a) (b) (c)

Figure 9: *The three SIMs shown in Figure 1 are presented here from a somewhat different view that reveals their real nature.*

(a) (b) (c)

Figure 10: *The model of K. Sugihara that was awarded the best-illusion-of-the-year in 2010 (see footnote on page 1) is modeled here using the conical deformation function F^1_{deform}. (a) shows the original illusion view that presents descending slides, moving away from the center, whereas (b) and (c) show that the slides are actually going upward, away from the center.*

intended direction where all four slides look as if they go downward away from the center, whereas Figures 10 (b) and (c) show two other, general, views where the slides are seen to, in fact, go upward away from the center.

We complete our set of examples with a short description of the modeling environment that allows the creation of all these SIMs. Figure 11 shows a capture of the environment [GuIrit] that serves as a graphical user interface to the Irit geometric modeling system [Irit]. The modeling package of SIMs is implemented as an external shared library in GuIrit, like many other modeling features in this system, and its interface is seen on the far right side of Figure 11. A regular tangible 3D model is created using traditional geometric modeling techniques offered in the GuIrit system, only to be fed to the LoSDs' module. The different LoSDs (i.e. F_{skew}, F^1_{deform}, F^2_{deform}, etc.) can be selected and applied to the specific geometric model/part, interactively. A simultaneous view of the model from a different direction, aside from the view selected for the illusion, provides an excellent indications about the effect of the deformation, in real time. Because many of these deformations are non linear and flat polygons can actually bend and twist due to the application of the deformation, models with only triangles are preferred so no non-planar polygons will result. An option of ensuring the existence of only triangles in the input model is, therefore, required and offered. Further, a polygonal refinement scheme to limit the size of the maximal edge length in the model is also available. This improves the accuracy and hence quality of the (non-linear) deformation's output.

The constructed SIM undergoes major deformations. These deformations affect the shading and consequently reveal the 'mystery' behind the SIM. For example, at the bottom left corner of the Penrose triangle in Figure 5(a), an hard-to-accept discontinuity in the shading is easy to spot. A simple remedy to circumvent the problem is to keep the original vertices normals of the object before the deformation. Much like Gouraud shading [Foley et al. 1990] that assigns shading intensities based on interpolated normals that are not of the piecewise linear approximated model but, typically, of the original smooth object, we will supply here the deceptive normals of the (original, pre-deformed) model that will fool the eye into accepting a more believable drawing of a SIM.

4 Conclusions and Future Work

In this work, we presented a simple modeling environment that can affect the depth of the geometry in the scene, applying line of sight deformations (LoSDs). While 2D drawn figures are not affected by this scheme, we can affect the depth order of any selected part, creating entirely new SIMs. Any model that has crossing bars or parts, in Z, in some view, can be employed and deformed using the LoSD to create an impossible model. One additional example is the stool shown in Figure 12. Similarly, objects that mimic expected shapes and behaviors can be modified into shapes that contradict the expected behavior, as shown in the slopes in Figure 10.

Many classes of models that pass the Jordan test (i.e., Condition 1.2) can be modeled using the presented LoSD scheme. The

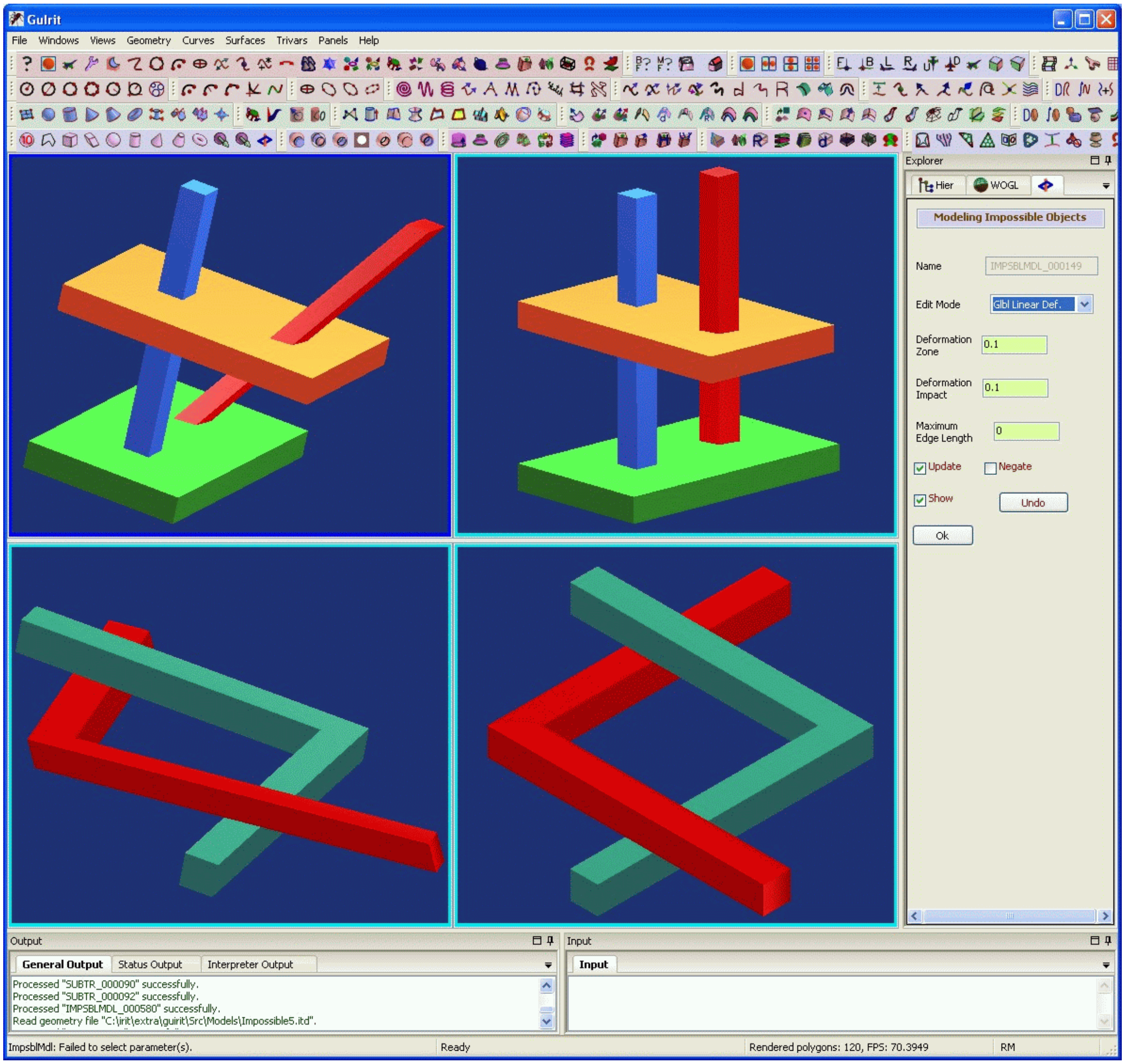


Figure 11: *The modeling environment of Seemingly Impossible Models' (SIMs) inside the GuIrit geometric modeling system [GuIrit]. Two additional SIMs are presented from a general view (left) and from the illusion, supposedly impossible, direction (right). The interface of the SIMs, modeling extension, which is modeled as a shared library in GuIrit, is shown on the far right.*

"Waterfall" drawing of M. C. Escher is merely formed out of a chain of three Penrose triangles and hence falls into the LoSD class. Another nice example that belongs to the LoSD class is "The Garden Fence" drawing (see Figure 14) of Sandro Del-Prete [Del-Prete]. In summary, any computer graphics and/or geometric modeling deformation techniques that is invariant of the line of sight can be employed here.

Nevertheless, a close inspection of Escher's "Belvedere" drawing shows that indeed the LoSD is not powerful enough to model it. The vertical pillars that connect the first level's balcony to the next level must be cut and stitched again, changing the topology of the model. Interestingly enough, a smaller such example is revealed on a careful inspection of the "Belvedere", exposing a person holding a Necker cube (see Figure 13). This SIM is also not part of the LosD class, while it passes the Jordan curve Condition 1.2, as again, its creation involves cutting and stitching of the model. See [Elber] for more on these models and others. Another group of SIMs that cannot be modeled using the LoSD techniques are models that are twisted in the plane. For instance, consider two lines in space that are not parallel in $\mathbb{R}^3$, and yet they are parallel when projected onto the drawing plane. The human interpretation of a ruled surface between the two lines will be of a planar region, while the ruled surface is clearly hyperbolic. Figuring out intuitive ways to model these non-LoSD objects and/or perform the deformation in some optimal way, whatever optimality means (possibly minimizing the surface normals' changes), is still ahead of us.

Acknowledgments

This research was supported in part by the Fund for Promotion of Research at The Technion, Haifa, Israel.

(a) (b)

Figure 12: *The model of a stool with crossing bars is modified into a SIM via the LoDS. (a) shows the seemingly impossible direction where as (b) shows a side view of the same model.*

The author would like to thank the anonymous referees for their invaluable comments that greatly improved the quality of this paper.

References

CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. 2009. *Introduction to Algorithms*. The MIT Press.

DEL-PRETE, S. http://www.del-prete.ch.

DOCARMO, M. P. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

ELBER, G. The "escher for real" project. http://www.cs.technion.ac.il/∼gershon/EscherForReal.

ESCHER, M. C. http://www.mcescher.com.

FOLEY, J. D., VAN DAM, A., FEINER, S. K., AND HUGHES, J. F. 1990. *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Publishing Company, second edition.

GUIRIT. The guirit geometric modeling environment. http://www.cs.technion.ac.il/∼gershon/GuIrit.

IRIT. The irit geometric modeling environment. http://www.cs.technion.ac.il/∼irit.

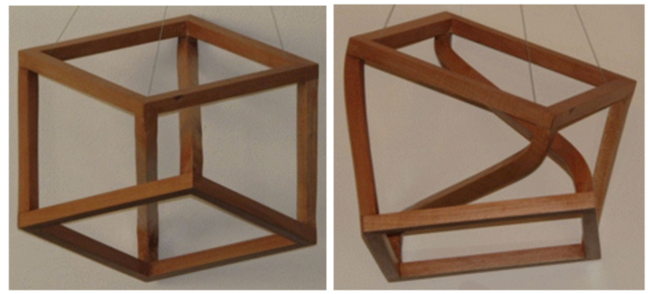
KUPLA, Z. 1987. Putting order in the impossible. *Peceptions 16*, 201–214.

LIPSON, A. Andrew lipson's lego page project. http://www.andrewlipson.com/lego.htm.

SAVRANSKY, G., DIMERMAN, D., , AND GOTSMAN, C. 1999. Modeling and rendering escher-like impossible scenes. *Computer Graphics Forum 18*, 173–179.

SUGIHARA, K. 1982. Classification of impossible objects. *Peceptions 11*, 65–74.

SUGIHARA, K. 2007. Computer-aided creation of impossible objects and impossible motions. In *Computational Geometry and Graph Theory: International Conference, KyotoCGGT*, Springer, vol. 11, 201–212.



(a) (b)

Figure 13: *The Necker cube: a SIM that is not in the class of LosD. Here the model is deformed as well as cut and stitched, affecting its topology, and is hence more general. (a) shows the model from the same view as in the "Belvedere" drawing of M. C. Escher while (b) presents a general view.*

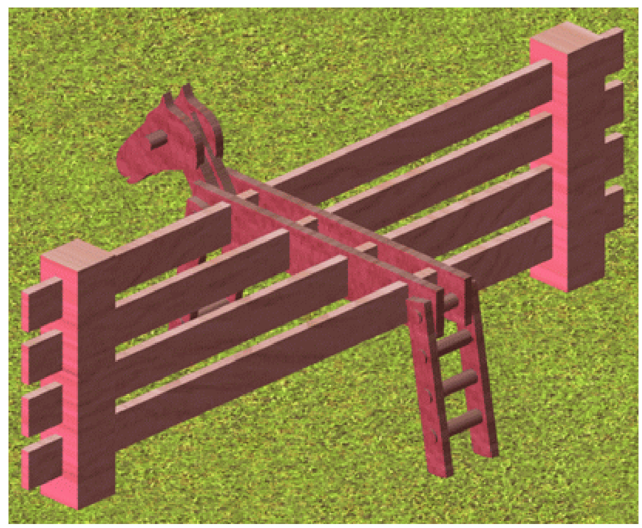


Figure 14: *Another SIM that can be modeled using a LoSD. After "The Garden Fence" of Sandro Del-Prete [Del-Prete].*

WU, T.-P., FU, C.-W., YEUNG, S.-K., JIA, J., AND TANG, C.-K. 2010. Modeling and rendering of impossible figures. *ACM Transactions on Graphics (TOG) 29*, 13:1–13:15.