Exploring 3D Objects with Non-Linear Perspectives in Real-Time, a First User Study

Michael Ortega, David Furió

Exploring 3D Objects with Non-Linear Perspectives in Real-Time, a First User Study

Michael Ortega
Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble, France
michael.ortega@imag.fr

David Furio
Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble, France
david.furio@rplusplus.com

ABSTRACT

Non-linear perspectives have the potential to improve 3D scene perception by increasing the information bandwidth of 3D contents. As with the example of the Mercator projection of earth, they can reduce occlusions by showing more of the shape of an object than classical perspectives. However, an ill-advised construction of such "usually static" perspectives could make the original shape difficult to understand, drastically reducing the scene comprehension. Yet, despite of their potential, these perspectives are rarely used.

In this paper we aim at making non-linear perspectives more widely usable on mobile devices. We propose to solve the understanding issue by allowing the user to control the transition between linear and non-linear perspectives in real-time with bending gestures. Using this approach, we present the first user study that investigates real-time manipulation of non-linear perspectives in an exploration task. Results show significant benefits of the approach, and give insights on the best bending gestures and configurations.

CCS CONCEPTS

• **Human-centered computing** → *User studies.*

KEYWORDS

Non-linear perspectives, Flexible screen, 3D interaction

ACM Reference Format:
Michael Ortega and David Furio. 2020. Exploring 3D Objects with Non-Linear Perspectives in Real-Time, a First User Study. In *mobileHCI '20: 22nd Conference on Human-Computer Interaction with Mobile Devices and Services, October 05–08, 2020, Oldenburg, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3379503.3403533

1 INTRODUCTION

Today, most of the mobile devices are powerful enough to display complex 3D virtual environments in real-time. As a consequence, tablet-computers are used in several fields. For instance, in anatomy courses students use tablets for exploring and understanding 3D models of bones, muscles and tendons [15]. Displaying such environments is commonly done using *linear perspective* (*LP*), inspired by the pinhole camera (projection center is a single point). This projection is close to the human visual system, and provides a "realistic" view of the scene. Nonetheless, *LP* has some limitations. For example, it cannot display the whole surface of an object in a single view, and forces the user to constantly change the viewpoint

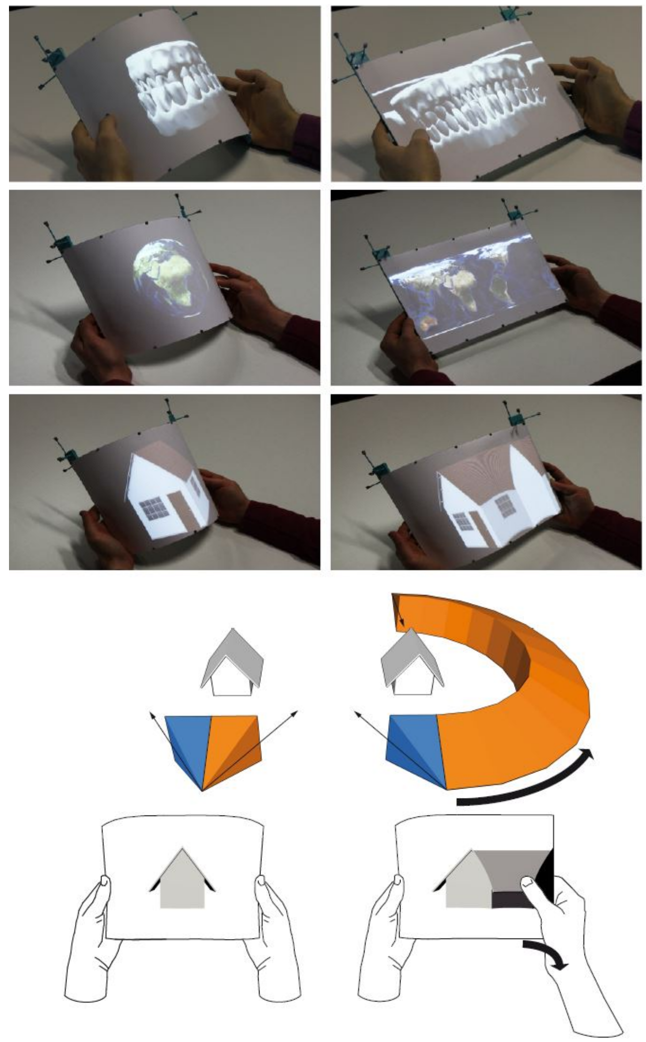

Figure 1: *Convex Flat* technique. *Photos*: models are displayed on a convex screen (*Left*), then 'unfolded' when screen is flat (*Right*). *Bottom*: Half unfolded house, right part only, with the corresponding viewpoint behaviour and gesture for simulating the None-Linear Perspective.

mobileHCI '20, October 05–08, 2020, Oldenburg, Germany

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/3379503.3403533

position [17]. Furthermore, many elementary tasks are difficult to achieve with precision, like measuring the distance or the relative size of objects. *Orthographic/Parallel projection* (a particular case of *LP*) can solve the measuring issue, but still does not give access to the back faces.

To display front and back faces in a single view, *non-linear perspective (NLP)* (also called multiperspective projection [21]) have been used in scientific representations (e.g, [13]), and also for several centuries in painting. The "unfolded" faces from Picasso are among the most famous examples. With *NLP*, the projection center is not a single point anymore, and could be any geometrical/mathematical shape (e.g. a cylinder in *Mercator* projection of Earth, Fig. 1).
In artistic projections the painter decides to make the final image still interpretable by itself or not. To take advantage of *NLP* in scientific or pedagogical contexts, like anatomy learning, the resulting image **has to be interpretable**. The user has to be able to mentally reconstruct the original shape. *NLP* deformation is therefore limited, i.e. not "far" from *LP*.

In this paper, we aim at finding a solution that takes full advantage of *NLP* to explore 3D objects. We propose to give the user the control of the projection, i.e. the control of the back-and-forth transition between *LP* and *NLP* in real-time. *NLP* can go as far as necessary from *LP*, and the amount of "deformation" can be interactively and continuously chosen according to the task. This approach takes advantage of animated/controlled transitions, that helps keeping the context between two viewpoints [10, 17].

Interactive NLP can be used as a powerful tool for exploring a wide range of environments: 3D Objects and scenes [3, 4], large landscapes and city models [8] or pedagogical multilevel representations [24]. It can also be used for creating artistic pictures and movies [7], like backdrops in cel animation [26].

In the context of mobile tablets, the control of the *NLP* transitions has to be 1) fast and continuous, 2) concurrent with the use of touch, and 3) with good degrees of indirection/integration/compatibility [2]. Organic User Interfaces (OUI) [11], Shape-Changing Interfaces (SCI) [6], and more specifically bending interactions are therefore good candidates. Indeed, as suggested by the literature (see here), and more particularly by Ahmaniemi *et al.* [1], the full potential of bending is reached when it is mapped to continuous and bipolar input. Bending gestures seem to be well adapted to control transitions.

Gummy [19], by Schwesig et al., is one of the first bendable tablet prototypes. They investigated bending gestures for 2D positioning control. Since then, while most research focuses on mapping bending gestures to GUI controls (e.g. navigation [14], gaming control [16, 20] or video exploration [22]), only a few focus on mapping bending to viewpoint deformation. For example, inspired by the cross-cut principle of PaperLens [22], FlexPad [23] implements curved cross-cuts on volumetric data. As the most closely related work to ours, Dickie et al. proposed FlexCam [9]. They use three real cameras on the back of a tablet to simulating a single camera, which Field-of-View is manipulated by bending gestures. However, FlexCam allows LP modifications only, no NLP, and has not been evaluated.

Our contributions are: a first flexible tablet-computer prototype that controls an *NLP* projection in real time, and a user study that investigates this approach in exploration tasks.

2 FLEXIBLE TABLET PROTOTYPE

Bendable tablets are around the corner, yet only foldable ones are commercially available (e.g. Galaxy Fold). As we aim at using continuous curvature for high degrees of compatibility [2], we built our own prototype. The main constraints were: a lightweight prototype (no wires around the tablet, high freedom of movement, and minimal fatigue), a minimal force to bend and keep a given curvature, and a precise detection of the surface deformations. These constraints led us to a projection system on a simple cardboard, like in [12].

We use an Optitrack tracking system, from Natural Point, and a BARCO F50 WQXGA projector which resolution is 2560×1600 pixels at 120Hz. The projector is attached to the ceiling of the experimental room, above the user's head. Except for the markers detection, we developed our own software, in Python with PyOpenGL. Using GPU fragment shaders, the *NLP* image is rendered off-screen, and stored as a texture. The texture is then mapped onto a virtual quad, computed from both position and deformation of the real one. Knowing the exact intrinsic/extrinsic parameters of the projector in the tracking frame, we use it for computing the final image. The virtual quad is projected and superimposed onto the real one, creating the flexible-screen illusion.

Considering the last improvements in graphical cards, we decided to develop *NLP* computation with Ray-Casting, on GPU [3, 4, 13, 25]. With this approach, we can compute a different viewpoint for each pixel of the resulting image, and then simulate an infinite number of perspectives. We implemented the Ray-Casting in a fragment shader, and not in a vertex shader, in order to provide visually smooth deformations regardless of the objects' mesh precision. Indeed, computing the rays for each pixel, instead of for each vertex, makes the triangle edges to curve, and therefore accurately fits to the expected deformation.
For this first experiment, and as it is widespread in the literature, we decided to limit *NLP* to horizontal deformation only, like in [18, 27]. Moreover, vertical bending while keeping the screen horizontal and the hands on lateral sides, is more demanding. Efforts for up and down bending are not symmetric, and needs the fingers more than the wrist. The combination with touch interaction is then a lot more difficult, if not impossible. Such a gesture has not been investigated in the literature and we decided not to integrate it in this very first study.

3 FROM BENDING GESTURES TO NLP CONTROL

As shown in Figure 2, for this first prototype we decided to allow biquadratic deformation (quartic polynomial degree 4). The flexible screen is therefore divided in 4 vertical areas, and we compute a bending coefficient from each one. These coefficients, C_{0-3} with values in $[-1, 1]$, are sent to the GPU, and used as inputs to the *NLP* computation. Fragment shaders are then executed for each ray/pixel of the final image, from the current viewpoint position, and from the bending coefficient associated to the area the pixel is from. As described in Figure 2, each vertical area can be bent independently for transforming the projection from a point to a line, and then creating a continuous assembly of *NLP*. In our experiment,

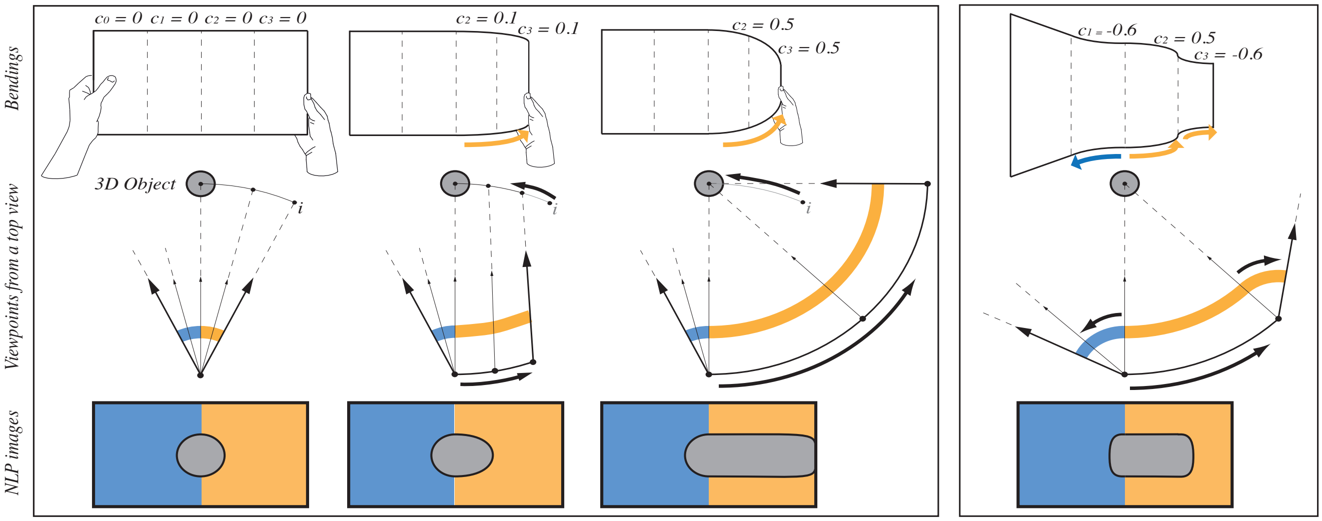

Figure 2: Bending coeficients are independently applied to the 4 vertical parts of the viewport. When $C_i > 0$, the starting positions of the rays travel along a circle centered on the 3D object position. Rays directions are interpolated from their initial position, i.e., when $C_i = 0$ (e.g., looking at *i* for the extreme right ray) to the object position. This interpolation is computed when $C_i \in [0, 0.3]$. A ray looks at the object position since $C_i > 0.3$. Negative coefficients make the rays rotate on themselves and open the viewport angle (*right*).

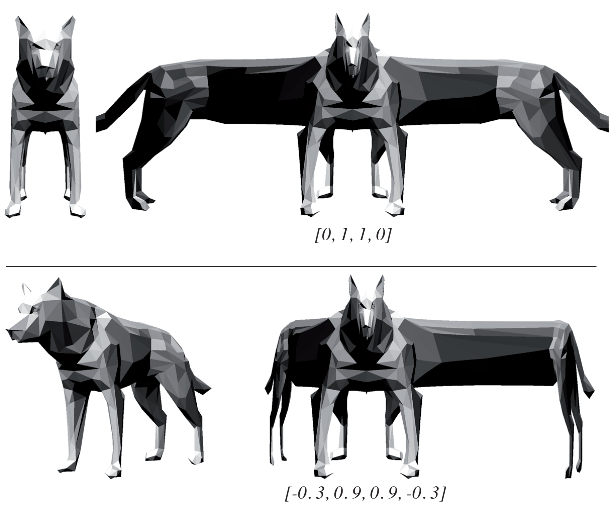

Figure 3: Examples of *NLP* deformations (rights, with their bending coeficients), from two different viewpoints (lefts)

a gain is applied on coefficients for allowing users to reach the *NLP* maximum while the whole quad surface is still visible.

When all the coefficients are equal to 0 all the rays start from the current viewpoint position, and define a classical linear perspective, with a wide-angle lens of 60° (Fig. 2-*Left*). With $C_i > 0$, the starting position of a ray moves along an horizontal circle that is centered onto the "look-at" position of the original viewpoint (Fig. 2-*Middle*). The ray direction is interpolated from the initial direction (when $C_i = 0$) to the object center. In our prototype, this interpolation is made during the beginning of the bending, while $C_i < 0.3$. With $C_i \geq 0.3$, the ray is directed to the object center. With $C_i = 1$, the ray made a quarter circle. Then, when all the coefficients are equal to 1, the rays are distributed on a complete circle around the look-at position of the original viewpoint, giving the illusion of an unfolded object.

With $C_i < 0$, a ray rotates on itself, and opens the viewing angle (Fig. 2-*Right*). Then, when all the coefficients are equal to -1, the rays simulate a 360° camera.

4 USER EVALUATION

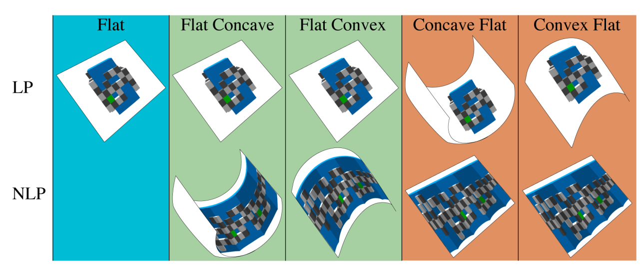

Figure 4: Interaction techniques. *Top:* initial curvatures with *LP*. *Bottom*: shapes with full *NLP*.

In this study, we aim at evaluating the benefits of controlling *NLP* via bending gestures in an exploration task. As bending interaction performance can be influenced by curvature direction and starting curvature, we explored 4 variations that combine 2 initial curvatures and 2 directions (see Fig. 4):

- ***Flat Concave***. The initial curvature is flat. Transition to *NLP* is made by bending into a concave shape: the screen center moves away from the user.

- ***Flat Convex***. The initial curvature is flat. Transition to *NLP* is made by bending the screen into a convex shape: the screen center comes closer to the user.
- ***Concave Flat***. The initial curvature is concave. Transition to *NLP* is made by flattening.
- ***Convex Flat***. The initial curvature is convex. Transition to *NLP* is made by flattening.

These techniques are compared to the status quo, called ***Flat***, which allows touch interaction only, and then rotation only. All of the techniques allow touch interaction for rotating the 3D models. Rotations are made using the arcball technique (isotonic control), and were performed with the thumbs.

4.1 Task and Design

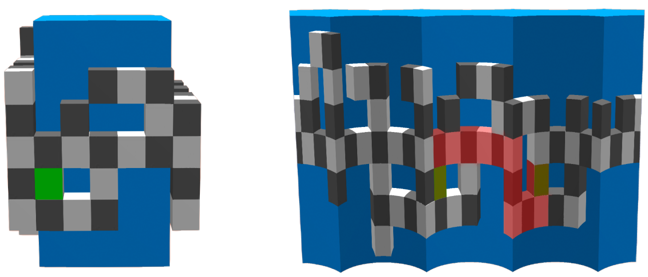

Figure 5: *Left*: One model used in the study. *Right*: full unfolded model (bend. coef. = 1) with shortest path in red.

As no standard protocol exists in the literature to evaluate exploration, we created our own. Participants were successively presented 3D models that consist of a central blue cube with several white and grey cubes around (Fig. 5). As there are several ways to explore an object: from rough shape estimation to detailed observation, and as we don't know the influence of the techniques on this continuum, the participant's mission was divided in two steps. For each model, the participant had 1) to explore the model for finding two green cubes among the surrounding cubes (rough shape estimation), and 2) to count the shortest path between these cubes (detailed observation). Depending on the interaction technique, participants could use rotation, bending or both. 18 different models were created, and presented to participants.

For each task, **the initial position of the models did not allow to see any of the green cubes**, and the participant had to rotate the model and/or modify the projection to find them. In half of the models, the green cubes were close enough to be seen together with a single linear perspective. We considered exploring these models as an ***Easy task***. In the other models, the green cubes could not be seen at the same time in any single linear perspective. Only *NLP* projections allow it. We considered exploring these models as an ***Hard task***. With the *Flat* technique, participants were forced to use successive rotations to count the path length. As touch was allowed with all the techniques, this strategy be used with all techniques.

4.1.1 Quantitative. The study used a within-subject design, all participant experimented all 5 interaction techniques. Participants were asked to complete the 18 exploration tasks with one technique before changing to another. Participants completed the tasks as fast as possible.

The dependent variables were ***Movement time*** (the time to complete the task), ***Searching time*** (the time to find the two green cubes), ***Counting time*** (the time to find and measure the shortest path between the green cubes) and ***Error*** (whether the path length was correct or not).

4.1.2 Qualitative. Once all the exploration tasks were completed, we asked each participant to rate the techniques, from 0 to 20 (20 is the highest score), by ***Global preference*** (how much the participant liked each technique), ***Perceived time*** (which technique he thought he was faster with), ***Estimated error*** (how confident he was about the estimated path length), and ***Ease of use*** (how easy-to-use the technique was). To force participants to classify all the techniques, two techniques could not have the same score. Combined with the large scale of 20, participants could clearly express differences, and also create groups, e.g. several techniques could all have very good rates while being ordered and classified.

4.2 Hypotheses

As *NLP* enables access to the sides and back of a 3D object in a single image, we first hypothesise that the exploration task will be completed faster with techniques that allow *NLP* (***H1***). This hypothesis can be divided in two. Finding the green cubes should be faster with *NLP* techniques ***H1a***. Indeed, *Flat* forces the participant to make several rotating gestures for exploring the 3D model while *NLP* enables displaying the whole model surface in one 'unfolding' gesture. Counting the path length should also be faster with *NLP* techniques ***H1b***. Indeed, after knowing the cubes position, *Flat* still forces the participant to rotate the object during the *Counting Time*. Also, for the same reason as in *H1b*, we hypothesise that there will be less errors with techniques that allow *NLP* than with *Flat* (***H2***).

4.3 Participants and Procedure

A total of 27 participants, 22 to 39 years old, were recruited. The experiment lasted about 30*min*. It started with a global explanation, and a calibration for touch interaction. Then, before each technique participants trained with a house model (Fig. 1), as long as they needed to. After the trial session, the 18 (randomly presented) exploration tasks were chained.

To log the switch between Searching and Counting phases, we could not use any physical nor virtual button. As participants held the tablet with two hands, any additional hand or finger movement would have changed the current bending configuration. In a pilot study, we used a foot pedal, and several participants were not comfortable with it. They reported that they were losing time on switching between these two interaction modes (hand interaction vs foot interaction), and therefore losing time in completing the task. As this interaction could potentially have created a high variability in *Movement Time* (beside driving, most of the people are not used to combine hand and foot movements), we asked participants to explicitly notify the switch out loud to a unique person. We observed that participants were more concentrated on the main task when they just had to say a short 'now!'. This method cumulates the reaction times of both the participant and the experimenter, but we believe this has a lower impact than asking participants to combine both hand and foot interactions.

Participants could rest at any time between the tasks.
We performed ANOVA tests on the four dependent variables (Movement Time, Searching Time, Counting Time and Error). Independent variables were Technique (the five techniques) and Difficulty (Easy, Hard). Since more than two factors were involved, we performed post-hoc tests (Tukey's tests) to find the statistical differences (Figure 6).

4.4 Results

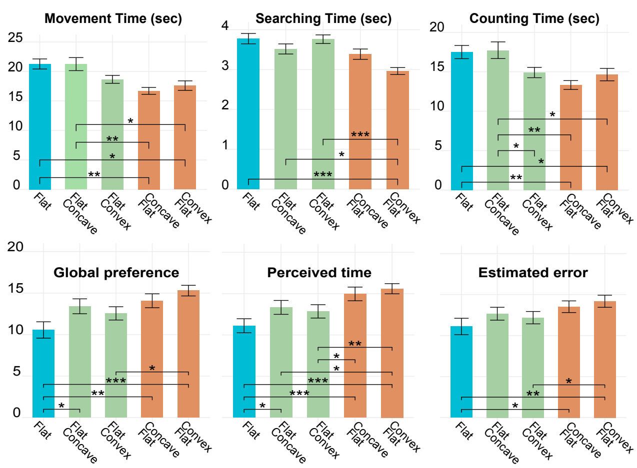

Figure 6: Mean, standard error, and significant difference from Tukey's tests ('*': $p \leq 0.05$, '': $p \leq 0.01$, '***': $p \leq 0.001$). Times (first row) are in seconds, and rankings (second row) are all in-between 0 and 20.**

As easy tasks results are not significant, this section focuses on hard tasks results only (illustrated in Figure 6).

4.4.1 Quantitative. *Technique* has a significant effect on *Movement time* ($F_{4,104} = 3.378, p = 0.012, \eta^2 = 0.039$), *Searching time* ($F_{4,104} = 4.447, p = 0.002, \eta^2 = 0.089$), and *Counting time* ($F_{4,104} = 3.776, p = 0.007, \eta^2 = 0.044$). However, none of the results showed significant effects of *Technique* on *Error*.

4.4.2 Qualitative. *Convex Flat* obtained the highest rank, followed closely by *Concave Flat* and *Flat Concave*. It is also interesting to note that *Convex Flat* was ranked as the best, the second best, and the third best technique more often than any other technique. *Perceived time* and *Perceived error* followed a similar trend. However, *Technique* had no significant effect on *Ease of use*. To go further, we also grouped the 4 *NLP* variations and compared them to *LP*. With a t-test on these results, we found significant differences for all the rankings: ***NLP* is preferred, perceived faster and less error prone than *LP*** ($p < 0.0001$). ***NLP* is also easier to use** ($p < 0.0077$).

5 DISCUSSION

First, the lack of significance for the easy tasks suggests that there is limited benefit of using NLP when LP is sufficient for completing the task. However, LP is currently a widespread standard while NLP is not. As LP does not outperform NLP, a longer evaluation could reveal the real difference between the two projections on easy tasks.

Next, as no significant differences have been found between the techniques on *Error*, ***H2* is not validated**. We observed that participants made more rotations with flat than with *NLP* techniques during the counting phase, but this seems to influence time only, and not *Error*. With *Flat*, some participants stated that they counted the path two times to be sure they got the correct number. It seems that they tried to compensate the lack of confidence by taking a bit more time counting.

This experiment does not validate H1 either, nor its variations (H1a, H1b). Indeed, H1 is about NLP in general, and while Convex-Flat and Concave-Flat are the fastest techniques for every dependent variables (times), the other two NLP techniques, Flat-Concave and Flat Convex, are not significantly faster than the baseline. However, the initial configuration of the screen, as well as the bending gesture paradigm, seem to be critical factors to explore for performance with NLP techniques. Some reasons can be found in the comments that participants made after the experience: 5 participants said that **it was easier to count when the surface was flat than when it was bent**. Indeed, when the flexible screen is bent, participants' gaze direction is not orthogonal to every part of the screen, which may disturb counting. They used touch interaction to rotate the object, as in linear perspective, and they waste some time and comfort. Also, for almost half of the participants, it felt more natural, fast and immediate to **"flatten the physical object to flatten the virtual one"**. Here they expressed the need of a high *degree of compatibility*, i.e., the similarity between a physical action of the user and the response of the object [2].

Beside the fact that users clearly prefer *NLP* over *LP*, ***Convex Flat* and *Concave Flat* were the preferred techniques**. They were also perceived as the fastest and the less error prone. For many participants, *Flat* made the exploration task more difficult, and implied to constantly rotate the 3D model to look for/at the green cubes. Sometimes these rotations got the participants lost while counting the number of cubes in the path forcing them to start counting again. They also said that "*Flat* offers only one vision of the object".

Concerning *NLP* complexity, even by showing to the participants that the screen was divided in 4 vertical areas, none of the participants used the full potential of this feature. Only three of them bent the screen in an asymmetric way, i.e., right or left screen only, and almost all of the participants bent in a symmetrical way, i.e., making a 'tubular' shape. We hypothesise that unfolding only one part of the model requires more effort, not useful for the task. This will be further investigated, in relationship with both model and *NLP* complexities.

6 LIMITATIONS AND FUTURE WORK

Our NLP approach simulates unfolding objects. Exploration complexity and performance are therefore linked to the topology of the shapes. Exploring every part of 'convex' objects like a balloon might be 'easier' than more complex objects like a torus or a cup with handles. However, as our interactive approach creates NLP from a viewpoint that the user explicitly controls (by touch), one

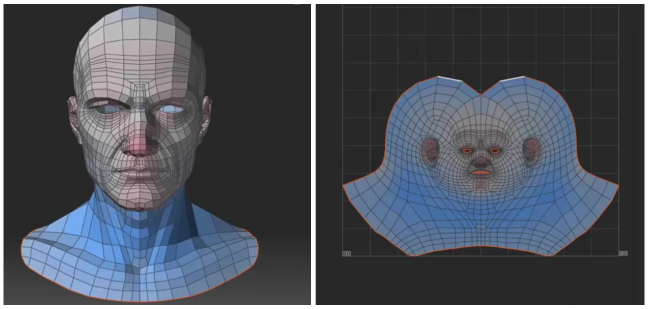

Figure 7: A 3D model (*left*) and its "unfolded" representation (*right*), from *UNFOLD3D*®.

can imagine interaction strategies for accessing all the shape. Investigating complex shapes ,and therefore more complex paths, would test our hypothesis and could reveal these strategies and new limitations of our approach.

Next, to precisely outline the benefits of using NLP in exploring a 3D object, other tasks, like 'relative positions estimation' or 'global shape estimation' should be investigated. New deformations could also be investigated, combined with new gestures, in order to propose '2D unfolding'.

In regards to the hardware, using projection is efficient enough and did not limitate our ability to precisely prototype and evaluate new bending interactions and paradigms. However, the stiffness, thickness and weight of the prototype are far from a real/future flexible tablet. As these properties influence interaction [5], they should be investigated in a new experiment with more realistic material.

7 CONCLUSION

In this paper we present the first user study that investigates the benefits of using Non-Linear Perspectives in an exploration task. Among the tested interaction techniques, *Concave Flat* and *Convex Flat* look promising. They provide good performance, they are the preferred techniques and the ones that users feel more confident with. These first results are a good starting point for widespreading Non-Linear Perspectives in everyday tablet-computer applications. However, more complex models and deformations should be investigated. We plan a longitudinal evaluation with anatomy learners, and also 3D designers who frequently "unfold" 3D models to paint on them (Figure 7). Our system is a strong candidate to help them decreasing the high cognitive load of making the correspondence between an unfolded model and its folded counterpart.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of ANR: the ISAR project (ANR-14-CE24-0013), and the ANATOMY2020 project (ANR-16-CE38-0011).

REFERENCES

[1] Teemu T Ahmaniemi, Johan Kildal, and Merja Haveri. 2014. What is a Device Bend Gesture Really Good for?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3503–3512. https://doi.org/10.1145/2556288.2557306

[2] Michel Beaudouin-Lafon. 2000. Instrumental Interaction: An Interaction Model for Designing post-WIMP User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 446–453. https://doi.org/10.1145/332040.332473

[3] John Brosz, Faramarz F Samavati, M Sheelagh T Carpendale, and Mario Costa Sousa. 2007. Single Camera Flexible Projection. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering (NPAR '07)*. ACM, New York, NY, USA, 33–42. https://doi.org/10.1145/1274871.1274876

[4] John Brosz, Faramarz F. Samavati, M. Sheelagh T. Carpendale, and Mario Costa Sousa. 2007. Single Camera Flexible Projection. In *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering* (San Diego, California) *(NPAR '07)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1274871.1274876

[5] Jesse Burstyn, Juan Pablo Carrascal, and Roel Vertegaal. 2016. Fitts' Law and the Effects of Input Mapping and Stiffness on Flexible Display Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3649–3658. https://doi.org/10.1145/2858036.2858383

[6] Marcelo Coelho and Jamie Zigelbaum. 2011. Shape-changing Interfaces. *Personal Ubiquitous Comput.* 15, 2 (Feb. 2011), 161–173. https://doi.org/10.1007/s00779-010-0311-y

[7] Patrick Coleman and Karan Singh. 2004. Ryan: Rendering Your Animation Nonlinearly Projected. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '04)*. Association for Computing Machinery, New York, NY, USA, 129–156. https://doi.org/10.1145/987657.987678

[8] Hao Deng, Liqiang Zhang, Jingtao Ma, and Zhizhong Kang. 2011. Interactive Panoramic Map-like Views for 3D Mountain Navigation. *Comput. Geosci.* 37, 11 (Nov. 2011), 1816–1824. https://doi.org/10.1016/j.cageo.2011.04.002

[9] Connor Dickie, Nicholas Fellion, and Roel Vertegaal. 2012. FlexCam: Using Thin-Film Flexible OLED Color Prints as a Camera Array. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI EA '12)*. Association for Computing Machinery, New York, NY, USA, 1051–1054. https://doi.org/10.1145/2212776.2212383

[10] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. 2008. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1141–1148. https://doi.org/10.1109/TVCG.2008.153

[11] David Holman and Roel Vertegaal. 2008. Organic User Interfaces: Designing Computers in Any Way, Shape, or Form. *Commun. ACM* 51, 6 (June 2008), 48–55. https://doi.org/10.1145/1349026.1349037

[12] David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. 2005. Paper Windows: Interaction Techniques for Digital Paper. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. ACM, New York, NY, USA, 591–599. https://doi.org/10.1145/1054972.1055054

[13] Wei-Hsien Hsu, Kwan-Liu Ma, and Carlos Correa. 2011. A Rendering Framework for Multiscale Views of 3D Models. In *Proceedings of the 2011 SIGGRAPH Asia Conference (SA '11)*. ACM, New York, NY, USA, 131:1—-131:10. https://doi.org/10.1145/2024156.2024165

[14] Meagan Leflar and Audrey Girouard. 2014. Navigating in 3D space with a handheld flexible device. *Entertainment Computing* 5, 4 (2014), 205 – 209. https://doi.org/10.1016/j.entcom.2014.07.002

[15] T L Lewis, B Burnett, R G Tunstall, and P H Abrahams. 2014. Complementing anatomy education using three-dimensional anatomy mobile software applications on tablet computers. *Clinical Anatomy* 27, 3 (2014), 313–320. https://doi.org/10.1002/ca.22256

[16] Jessica Lo and Audrey Girouard. 2017. Bendy: Exploring Mobile Gaming with Flexible Devices. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction* (Yokohama, Japan) *(TEI '17)*. Association for Computing Machinery, New York, NY, USA, 163–172. https://doi.org/10.1145/3024969.3024970

[17] Michael Ortega. 2013. 3D Object Position Using Automatic Viewpoint Transitions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, New York, NY, USA, 193–196. https://doi.org/10.1145/2470654.2470681

[18] Paul Rademacher and Gary Bishop. 1998. Multiple-center-of-projection Images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. ACM, New York, NY, USA, 199–206. https://doi.org/10.1145/280814.280871

[19] Carsten Schwesig, Ivan Poupyrev, and Eijiro Mori. 2004. Gummi: A Bendable Computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 263–270. https://doi.org/10.1145/985692.985726

[20] Paden Shorey and Audrey Girouard. 2017. Bendtroller:: An Exploration of In-Game Action Mappings with a Deformable Game Controller. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA,

1447–1458. https://doi.org/10.1145/3025453.3025463

[21] Karan Singh. 2002. A Fresh Perspective. In *Proceedings of the Graphics Interface 2002 Conference, May 27-29, 2002, Calgary, Alberta, Canada*. 17–24. http://graphicsinterface.org/wp-content/uploads/gi2002-3.pdf

[22] Martin Spindler, Sophie Stellmach, and Raimund Dachselt. 2009. PaperLens: Advanced magic lens interaction above the tabletop. *ITS 2009 - The ACM International Conference on Interactive Tabletops and Surfaces, Proceedings*, 69–76. https://doi.org/10.1145/1731903.1731920

[23] Jürgen Steimle, Andreas Jordt, and Pattie Maes. 2013. Flexpad: A Highly Flexible Handheld Display. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (Paris, France) *(CHI EA '13)*. Association for Computing Machinery, New York, NY, USA, 2873–2874. https://doi.org/10.1145/2468356.2479555

[24] Nisha Sudarsanam, Cindy Grimm, and Karan Singh. 2008. Non-Linear Perspective Widgets for Creating Multiple-View Images. In *Proceedings of the 6th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '08)*. Association for Computing Machinery, New York, NY, USA, 69–77. https://doi.org/10.1145/1377980.1377995

[25] Daniel Weiskopf, Tobias Schafhitzel, and Thomas Ertl. 2004. GPU-Based Nonlinear Ray Tracing. *Computer Graphics Forum* 23, 3 (2004), 625–633. https://doi.org/10.1111/j.1467-8659.2004.00794.x

[26] Daniel N. Wood, Adam Finkelstein, John F. Hughes, Craig E. Thayer, and David H. Salesin. 1997. Multiperspective Panoramas for Cel Animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 243–250. https://doi.org/10.1145/258734.258859

[27] M. Wu and V Popescu. 2016. Multiperspective Focus+Context Visualization. *EEE Transactions on Visualization and Computer Graphics (TVCG)* 22, 5 (may 2016), 1555–1567. https://doi.org/10.1109/TVCG.2015.2443804