VisiFit: Structuring Iterative Improvement for Novice Designers

LYDIA B. CHILTON, Columbia University
ECENAZ JEN OZMEN, Columbia University
SAM ROSS, Barnard College
VIVIAN LIU, Columbia University

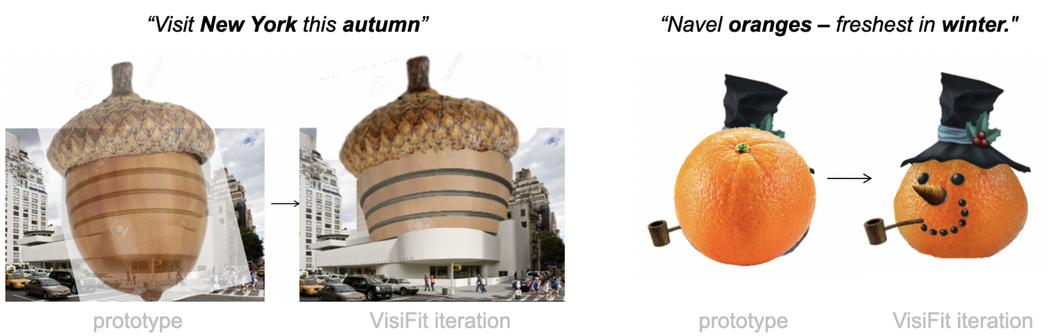

Fig. 1. Two examples of how the VisiFit system can improve a visual blend prototype in under 4 minutes. The left image blends *New York City* and *autumn*. The right image blends *navel orange* and *winter*.

Visual blends are a graphic design challenge to seamlessly integrate two objects into one. Existing tools help novices create prototypes of blends, but it is unclear how they would improve them to be higher fidelity. To help novices, we aim to add structure to the iterative improvement process. We introduce a technique for improving blends called *fundamental dimension decomposition*. It is grounded in principles of human visual object recognition. We present VisiFit - a computational design system that uses this technique to enable novice graphic designers to improve blends by exploring a structured design space with computationally generated options they can select, adjust, and chain together. Our evaluation shows novices can substantially improve 76% of blends in under 4 minutes. We discuss how the technique can be generalized to other blending problems, and how computational tools can support novices by enabling them to explore a structured design space quickly and efficiently.

CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**.

Additional Key Words and Phrases: Computational design, Design tools, Iterative design

ACM Reference Format:
Lydia B. Chilton, Ecenaz Jen Ozmen, Sam Ross, and Vivian Liu. 2018. VisiFit: Structuring Iterative Improvement for Novice Designers. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Iterative improvement is the essence of the iterative design process. No design is perfect at inception, thus iteration through prototypes is necessary to improve it. If a prototype passes an evaluation, it should become a new, higher fidelity prototype that can be tested and potentially iterated upon again. In case studies of improved software usability by the Nielsen Norman Group [41], median improvement per stage of iteration was 38%, leading to overall usability improvements of 165%. Iteration is not just an aspect of usability engineering, it is a fundamental part of the design process that generalizes across many domains. In web design, designers start with a wireframe prototype and move to a minimum viable product. In mechanical design, designers improve upon initial proofs of concept by iterating upon features and prototype reliability. In graphic design, designers sketch prototypes and then move onto higher-fidelity mockups. In each domain, iteration looks different, but the objective is the same - extend the prototype to move closer to the goal. To help novice designers in a meaningful and practical way, we need tools to support iteration.

Although there are many existing tools that support other phases of the design process - brainstorming, prototyping, evaluation, and final design execution, there is a lack of tools focusing on iteration [18]. Only 6% of 148 creativity support tools from 1999-2018 focus on iteration. Iteration tools are similar to brainstorming and prototyping tools in that they help people explore a design space. However, they are more difficult to build because they have more constraints. Unlike general prototyping tools, iterating on prototypes must be constrained further to build on ideas that were validated in the previous prototypes. Iteration still involves searching the design space, but the tools that were previously used to explore an expansive design space are not the right tools to explore a more constrained one.

Like all prototyping tools, iteration tools must be domain-specific so they can effectively operate on the materials of that domain. We focus on the difficult design challenge of making visual blends [4]. Visual blends are an advanced graphic design technique used to convey a message visually in journalism, advertising, and public service announcements. They combine two visual symbols into one object to convey a new meaning, for example *"Visit New York City in Autumn"*. Visual blends are a canonical example of a creative design challenge [26, 43] because they are open-ended enough to encapsulate all aspects of the design process, but well-defined enough to test in a short time frame. Moreover, cognitive scientists consider blending to be an important aspect of general creativity for its ability to "create new meaning out of old."[16] Currently, tools already exist to help people brainstorm and create initial prototypes [10] by finding the right images and arrangements to use for the blend. However, visual blends generally require an expert with Photoshop skills to execute the design and it would be faster, easier, and more empowering for novices to improve blends by themselves, without relying on an expert.

We perform several formative studies to learn how experts approach the iterative improvement of visual blends. From an analysis of blends created by experts and a participatory design process with graphic designers, we learned that blends do not simply blend the surface-level style of two objects, they combine the fundamental visual dimensions of both objects - silhouette, color and internal details. Based on this observation, we present a technique for structuring the iterative improvement process of blends called *fundamental dimension decomposition* (FDD). In FDD, the improvement process is first broken into stages that blend each of the dimensions separately. Then the results of each stage are combined into a single blended output. For example, in visual blends the user first blends the silhouettes of both objects, then blends the colors of the objects, then combines the internal details of both objects. The results of blending each dimension separately are then chained together to produce a seamless and aesthetic visual blend.

We present VisiFit - a computational design tool that allows novice graphic designers to improve a prototype of a visual blend. The initial prototype has the basic parts and arrangements of elements to blend, but the blend is low-fidelity with many rough edges. VisiFit uses the structure provided by *fundamental dimension decomposition* to create a pipeline of computational tools that seamlessly and aesthetically blends the two objects. Figure 1 shows two initial prototypes and the improvements made by novices using VisiFit in under 4 minutes. Our evaluation shows that novices can quickly and easily iterate on prototypes to create substantially improved blends.

This paper makes the following contributions:

- Three preliminary investigations into visual blends: a demonstration of how fully automatic systems fail, an analysis of patterns used by professionals, and a co-design process with graphic artists.
- Three design principles for a computational approach to improving visual blends.
- A technique for structuring the improvement of blends called *fundamental dimension decomposition*, which is grounded in the neuroscience of human visual object recognition.
- VisiFit, a system that applies the technique and design principles in a pipeline of computational tools.
- An evaluation of VisiFit showing that in under 4 minutes, novices can substantially improve blends in 76% of cases and create blends suitable to publish on social media in 70% of cases.

We conclude with a discussion of how *fundamental dimension decomposition* can help structure iteration in other fields and how pipelines of computational design tools can support the iterative design process.

2 RELATED WORK

2.1 Design Tools

Design tools and creativity support tools (CSTs) have a rich tradition of accelerating innovation and discovery [48] by supporting the design process. A survey of 143 papers from 1999-2018 on creativity support tools (CSTs) found that there are papers supporting all phases of the design process: ideation, exploration, prototyping, implementation, evaluation, and process/pipeline, and iteration. [18]. Many of these tools support more than one phase of the design process. However, not all phases of the design process are equally represented in the literature. In fact, a majority of these tools focused on either very early or very late phases of the design process. Of the systems in the survey, 45% support ideation [31, 49, 58], 41% support implementation, including high-fidelity tools [56] or low-fidelity tools for prototyping or sketching [11, 21, 32, 33], and 18% supported evaluation through feedback [37, 62] or expert annotation [50]. However, only 6% of the systems surveyed supported iteration, and only 4% supported the related task of design management or pipelines. More research is needed on how to support iteration more effectively — that is, how to help designers improve on an initial prototype to get closer to their final design goal. Our work in this paper focuses on this problem.

2.2 Iteration Support

Existing systems that explicitly aid iteration use a number of approaches. One class of iteration applications uses crowds to iterate towards better solutions [34]. This can be by mixing features of previous designs [65], responding to community feedback [28], hiring experts [45], or identifying weak points and fixing them [29]. All of these use the strength of multiple people's viewpoints to iterate. However, crowds can introduce errors and may be difficult to steer toward your particular vision. Therefore, it is often useful to provide designers with single user tools for iteration.

Another class of iteration tools has the user produce a prototype, and then computationally generate the rest of the design. If the user is unhappy with the outcome, they can regenerate, alter their input, or adjust parameters. Several applications apply this method to generate multi-tracked music from a simple input melody. This can be done using rules and constraints [15, 60] or implicit patterns learned by deep learning [36]. Having the computer generate outcomes is especially usable for novices; it allows them to recognize good outcomes, even if they cannot produce them. This seems to work well in music, which has many mathematical rules, but it is unclear if it works as well in other domains.

A third way to support iteration is to provide rich undo history to allow users control and freedom while exploring the design space. This is often done in the drawing domain both for single users [40] and for multiple users who want to draw collaboratively [66]. In the creative design process, exploration is clearly important [9], and supporting that is essential. In VisiFit, we use aspects of all three of these approaches. We target key properties of the prototype that need improving and focus iteration on these properties. We provide computational tools to generate outcomes that novices could not produce themselves. We allow users to explore design alternatives and to adjust parameters so they can achieve results they are satisfied with.

2.3 Computational Approaches to Design Tools

Computational tools have long been a promising approach to aid design because they can search a design space and help meet a constraint. The power of computational or computer-aided design has been shown in many fields such as: education [35], medicine [22], games [51], urban planning [6], and accessibility [19]. The system designer must define the space and the search parameters, as well as provide design patterns for solutions that can be adapted to different inputs. [3, 63, 64]

Computational design tools have had particularly strong adoption in graphic design problems like optimizing layout [8, 12, 42, 55], making icons [5, 7], and providing inspiration through mood boards [30, 59] and relevant examples [13, 31]. This is also true in the 3D domain, where computational tools can be used to search a design space and create multiple mesh and texture variations of objects (i.e. trees or airplanes) that can make computer generated scenes more diverse [38, 53]. Deep learning has also been applied to generate new designs that fit user specifications [39, 61]. In this paper, we address a specific kind of graphic design problem of that requires blending two objects into one in order to convey a new meaning. To our knowledge, none of the existing computational design tools have addressed this problem.

Although these tools can be fully automatic, some of the most useful tools are interactive and allow users to explore and guide the process. We take much inspiration from Side Views [54], an application that allows users to preview the effect of various image editing menu options, like those in Photoshop. By providing previews, users are able to recognize rather than recall the right tool to use. This also helps users adjust parameters of key properties and chain tools together to explore an even wider section of the search space. In VisiFit, we also take the interactive approach to computational design. Like Side Views, VisiFit allows users to preview and adjust tools, as well as chain them together. However, VisiFit is not just a tool for exploration - it is targeted at achieving a specific goal; multiple tools are chained together in a pipeline that explores each of the three key visual properties needed to complete a blend. This allows the user to explore the design space and iterate in a structured fashion towards their goal.

3 BACKGROUND: VISUAL BLENDS

Visual blends are an advanced graphic design technique where two objects are blended together into one that conveys a symbolic message. They represent a canonical and very challenging design problem. When asked to define design, Charles Eames once said, *"Design is a plan for arranging* *elements to accomplish a particular purpose"* [1] In a visual blend, the objects to blend are the elements, the way they overlap is their arrangement, and the particular purpose is the seamless blend of the objects to convey the message. Visual blends can be a lens through which we view creativity and cognition [27], and this is one reason why they are considered an interesting design challenge and have been studied from a computational standpoint by several researchers. In order to achieve a visual metaphor, two objects related to the metaphor must be blended such that both objects are recognizable yet both objects appear blended into one. In visual communication, visual blends and visual metaphors are a well-studied phenomenon [16, 17, 57] and considered a difficult graphic design challenge [4, 43]. The objective is not to convey the message without words, but to create a blend related to the words that draws attention to the message. [44].

An existing system called VisiBlends [10] helps novices with the first step to the design process: creating a prototype. However, they must complete the finished design either on their own or by hiring a graphic artist. Figure 2 shows an illustration of the VisiBlends workflow to create a visual blend for the message "Starbucks is here for summer". The creator must first identify two abstract concepts to visually blend, for example, *Starbucks* and *summer*. VisiBlends helps users brainstorm many objects associated with both concepts, then find simple, iconic images of those concepts. Users identify from those images the main shape of the object (i.e. whether it is a sphere, cylinder, box, a flat circle, or a flat rectangle). It then automatically searches over pairs of objects to find two that have the same basic shape. With those objects, VisiBlends creates a mock up of the blend by cropping, scaling, positioning and rotating the objects to fit together. The user then selects the best blends. Sometimes the system produces blends that are immediately ready to use, but most often, professional editing is needed. The bottom of Figure 2 shows the editing done by an artist. However, we would like to help novice designers create such iterations on their own.

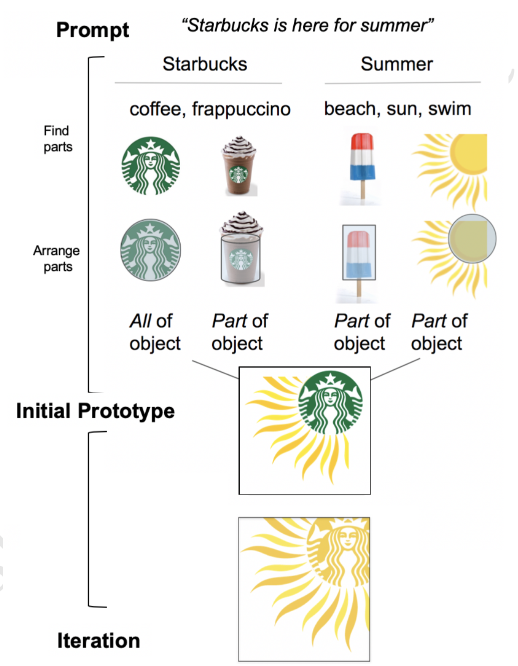

Fig. 2. An illustration of VisiBlends workflow that prototypes a blend for the prompt *"Starbucks is here for summer"*. After the initial prototype is made, an artist is needed to perform the next iteration. The goal of VisiFit is to enable novices to iterate on their own prototypes.

In VisiBlends, objects are matched if they have the same main shape. This is because shape match is the riskiest and most important aspect of a visual blend. It is hard to edit an object's basic shape (like turning a sphere into a long and thin rectangle). Thus, it is better to use the flare and focus approach to meet the shape-matching constraint. This design insight is backed up by the neuroscience of human visual object recognition, which states that 3D shape is the primary feature used by the brain to determine what an object is [52]. This is likely because 3D shape is the least mutable property of the object. Other features can change based on time or instance; for example, color changes in different lighting conditions, and identifying details have variation among individuals (hair color, eye color, etc.). By using different objects that have the same shape, you effectively interest the visual system.

The VisiBlends system primarily uses shape to make prototypes of visual blends because it is the primary feature for identifying objects. If we want to improve on the blend prototypes, we may consider combining secondary visual identifiers. The main secondary features that the brain's visual object recognition system uses are silhouette, color/texture, and internal details. It follows that when we look for expert patterns in improving blends, we should pay special attention to how these three visual properties are transformed.

4 FORMATIVE STUDIES OF BLENDING ITERATION

To explore approaches to iteration we conducted three preliminary investigations that informed the three design principles we propose for improving blends. We tie it all together into a general technique for structuring the iterative improvement of blends.

4.1 Shortcomings of Deep Style Transfer

Advances in deep learning have shown impressive results in manipulating images. An early and prominent result is deep style transfer [25] which trains a model on a visual style, such as Van Gogh's Starry Night, and applies that style on any image to make it look like Van Gogh painted it in the Starry Night style. This technique has the potential to automatically improve prototypes of visual blends by training on the style of one object and applying it to another.

To explore the potential of deep style transfer, we took four blend prototypes from the VisiBlends test set, and applied deep style transfer to them. For each pair of images in the blend, we selected which object to learn the style of and which object to apply the style to. We used an implementation of style transfer from the popular Fast Style Transfer (FST) paper [25] which only requires a single image to learn style from and has impressive results on transferring artistic style. We tried multiple combinations of hyper-parameters (epochs, batch size, and iterations) until we saw no noticeable improvements in the results. We also tried input images of the same object and different ways of cropping it, in case the algorithm was sensitive to any particular image.

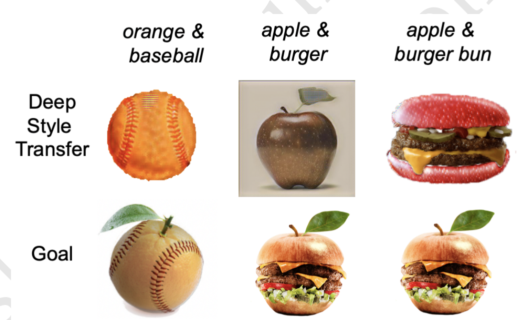

Fig. 3. Blends created by Fast Style Transfer (top) compared to blends produced by an artist (bottom). The FST blends fail because this problem cannot be solved with an indiscriminate, global application of one object's style onto another. Experts take apart and blend objects in a more nuanced way, preserving relevant characteristics of each object to keep each one identifiable in the final blend.

Although the algorithm was able to extract styles and apply them, the results fell far short of the bar for creating convincing blends. Figure 3 shows Deep Style Transfer results (top) and blends made by artists who we commissioned to produce high fidelity blends. To blend *orange* and *baseball*, FST first learned the orange style. However, when it applied that learned style to the baseball, while it preserved the baseball's characteristic red seams, it simply turned its white texture into a blotchy orange color that is not reminiscent of the fruit. In contrast, the artist who blended it used the texture and stem of the orange, in addition to the red seams of the baseball. This made both objects highly identifiable. The computer used the overall look of the orange, but didn't separately consider its elements as it mixed and matched the parts.

Similarly, for the *apple* and *burger* blend, the burger style applied to the apple just turned the apple brown, because the predominant color of a burger is brown. We also explored what would happen if we isolated part of the image by hand and applied the style only within that area. To mimic the artist, we isolated the burger bun and applied the apple style to it. The results are better, but still disappointing. Although the burger has the color and texture of an apple, it does not appear as blended as the artist's version. The artist chose to mix the apple color and the bun color to give a sense of *both objects* in that element.

We conclude that these existing style transfer results do not easily apply to visual blends. Blends are not just about applying high-level "style", they require designers to consider the individual elements and how they might be fit together. If we trained a model on thousands of visual blends, we might be able to make progress on this problem, but we would need to create those thousands of visual blends, and even so, results would not be guaranteed. Instead we want to explore semi-automatic approaches that augment people's ability to create blends.

Design Principle 1. To help users achieve better results, structure the problem into subtasks and provide interactive tools specific to each subtask. Fully automatic tools do not always achieve desired results and give you little control in how to fix them.

4.2 Analysis of professional blends

To investigate potential structures for improving blends we analyzed examples of blend prototypes that were improved by professional artists. We paid 3 professional artists to make visual blends based on 13 prototypes made by novices using VisiBlends. Of those 13 images, artists told us that two did not need editing — the output from VisiBlends was a perfectly acceptable blend. However, the other 11 blends needed significant iteration.

Based on the cognitive science of human visual object recognition used to establish the shape-based matching for visual blends, our analysis focused on how artists used secondary visual dimensions (silhouette, color/texture, and internal details) to improve blends. For example, Figure 2 shows one example of a low-fidelity prototype produced by VisiBlends, as well as a higher-fidelity iteration made by an artist. In this example, the artist made two key improvements: first, they changed the color of the Starbucks logo. It was originally green, but they made it yellow to match the color of the sun. Second, the artist cropped the Starbucks logo from a perfect circle to a partially occluded one at the corner of the page, to fit a silhouette that implies the sun. By changing these two visual dimensions (color and silhouette), the blend was dramatically improved.

We performed this visual dimension-based analysis on the 11 improved blends and found that three visual properties were sufficient to explain almost all of the improvements the artists made. Figure 4 shows examples of these dimensions:

- **Color/Texture**: The Lego in *Lego* and *ring* was initially solid red, but the artist gave the Lego the faceted texture of the diamond it replaces.
- **Silhouette** - the Lego in *Lego* and *Popsicle* was originally a rectangle, but the artist gave it the silhouette of the Popsicle. (It also has the texture of the Popsicle.)
- **Internal Details**: The orange in *orange* and *snowman* has the internal face details of the snowman placed back on the orange. (It also has the silhouette of the snowman head, and a blend of color/texture between the snow and the orange.)

Each improved blend is transformed on at least one visual dimension. Some prototypes can be improved by just blending one dimension. For example, *Lego* and *ring* only blends on color. However, for other prototypes, multiple dimensions need to be blended to achieve a seamless and aesthetic blend. For example, *Lego* and *Popsicle* blends on two dimensions - silhouette and color. *Orange* and *snowman* blends on all three dimensions - color, silhouette, and internal details. Thus, we believe that the three visual dimensions can be used together to guide the process of improving prototypes.

Design Principle 2. Identify fundamental dimensions to structure the iteration process. For visual blends, the three key fundamental dimension are: color, silhouette and internal details.

4.3 Co-Design with Graphic Artists

The three visual dimensions provide high-level structure for improving blends, but we wanted to know if there are actionable activities associated with this structure that are useful when improving blends. To investigate this, we worked with two graphic artists in multiple one-hour sessions over a period of three weeks to observe and probe their process. Both designers worked in Photoshop and had created numerous print ads although neither had made visual blends before. The goal of these sessions was to introduce them to the fundamental dimensions and to see if a) they found them useful to structure their process, b) what actions they took to improve the blends based on these dimensions, and c) whether novices would be able to replicate their success.

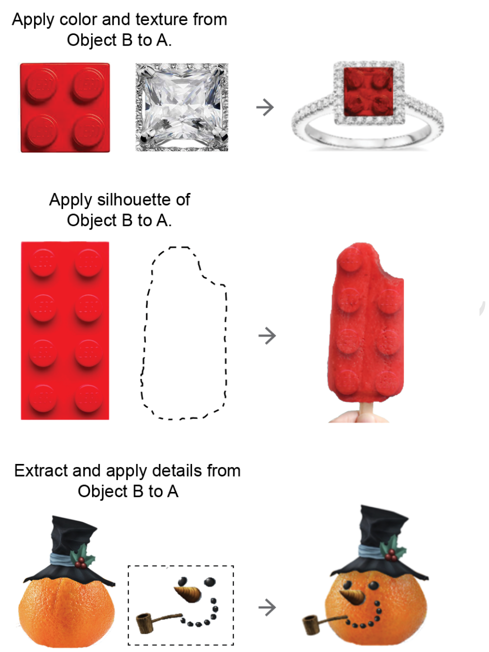

Fig. 4. Three visual properties artists change when improving visual blends: color, silhouette, and internal details.

To familiarize the artists with the concept of visual blends, we showed them examples of professionally made blends and asked them to recreate two of them in Photoshop. They found the task challenging, but through trial and error they were ultimately satisfied with their results. Next, we introduced them to the principles of blending based on color, silhouette and details. We discussed with them how we thought those principles could have been used to create the blends. Then we gave the artists prototypes of blends and asked them to improve them, referencing the visual dimensions when applicable.

The concepts of color/texture, silhouette, and internal details were intuitive to the artists, and they readily used them to improve the blends. Blending color/texture was a familiar idea to them, and it was very easy for them to do in Photoshop. An effective tool one artist used for blending was the "Multiply" feature, which preserved both the color and the texture of each object, as seen in the top panel of Figure 4. Both artists were surprised at how effectively silhouettes could be used in blends. They tried using the concept of silhouette blending in blends such as the middle panel of Figure 4 and were pleased with the results. The idea of extracting and reapplying details was natural to them, as they had employed analogous features in Photoshop (i.e. magic wand) to manipulate details before. However, even with industry tools, extraction was often tedious. In general, both designers thought that if they worked on the basis of these visual dimensions, they could recreate any visual blend.

The artists both note that there were additional techniques they would use to produce and even higher fidelity blends. One artist mentioned the addition or removal of shadows. The other mentioned making a background that would complement the blend. However, when restricted to these three visual dimensions, they could produce a second iteration with substantially reduced seams and enhanced aesthetic quality. If they were producing a pixel-perfect print ad, they would want to do a third iteration.

As we observed the artists using Photoshop to execute their improvements, we noticed two parts of their process that novice designers would struggle to replicate. First, almost all of the tools the artists used in Photoshop are not available in the typical applications novices use to quickly edit images. The simple filters, cropping, and movement afforded by Instagram, presentation software, and Mac Preview aren't enough to improve blends. Even simple the color/texture transfer operations like "Multiply" don't exist in most end-user tools. This is probably because most end-user tools focus on operations that can be applied to one image at a time. For blending, operations have to apply to two objects. Second, these tools often require multiple steps and tedious low-level manipulation. Applying the silhouette from one object to another is a process with multiple steps including positioning, object extraction, appropriate layer composition, and edge cleanup. Extracting details like the snowman face are tedious, even with the magic wand tool, which largely operates based on pixel color similarity. Instead of making users think in pixels, we want to provide higher-level abstractions, such as the separation of foreground from background or the separation of details from a base. To create operations that novices can use, we need to provide tools at a higher-level of abstraction than pixels.

Design Principle 3. Provide novices with high-level tools related to the fundamental dimensions that can preview results without requiring expert knowledge or tedious, low-level manipulation. In VisiFit, we provide high-level tools for (1) extracting and applying silhouettes, (2) blending color/texture between two objects, and (3) extracting and replacing internal details from one object to another.

4.4 Technique: Fundamental Dimension Decomposition

From these formative studies we proposed a technique that structures iterative improvement for novice designers. We believe this can be generally useful for many kinds of blending and remixing problems, not just visual blends. We call this technique *Fundamental Dimension Decomposition (FDD).*

The process of applying FDD is to first combine knowledge from cognitive science with expert domain knowledge to identify the fundamental dimensions of the problem space. Using those dimensions, structure the improvement process into stages that blend on the dimensions one at a time. For each dimension, provide novices with computational tools they can use to explore the different ways that dimension can be blended. These computational tools should be based on the high-level abstractions provided by each dimension, thus making them easy to apply while avoiding tedious, low-level manipulation. After decomposing the iterative process into stages based on the dimensions, the results can be recomposed by chaining the results of the stages together. In the discussion section, we provide examples of other domains where FDD can help iteratively improve blends - animated blends and creating hybrid styles in fashion and furniture design.

5 VISIFIT SYSTEM

To help novices iteratively improve visual blends, we created a system called VisiFit that leverages computational tools to help users easily extract and combine visual properties of each image into a blend. First the user improves the cropping of each image, then improves the three fundamental dimensions one at a time. At each step, they are presented with blend options that are automatically created by the system. However, they are free to interactively edit them. VisiFit is implemented as a Flask-based web application. It uses Numpy, OpenCV, and Tensorflow [2]. It builds on the Fabric.js canvas element to implement interactive image manipulation. Figure 5 shows the five steps of the interface in the order that users see them.

Inputs. VisiFit takes in two inputs that are both outputs from VisiBlends:

(1) *An ordered pair of images* that have a shape match. We refer to them as Object A and Object B. In Object A, the shape covers the entire object. In Object B, the shape covers only the main body of the object, leaving out parts of the object outside the shape. When blending the images, Object A will be mapped onto Object B.
(2) *The positioning parameters* to align Object A to the shape in Object B: x-scale factor, y-scale factor, angle of rotation, and center position. In the prototype of the blend, Object A is cropped, scaled, and positioned to fit into the shape of Object B.

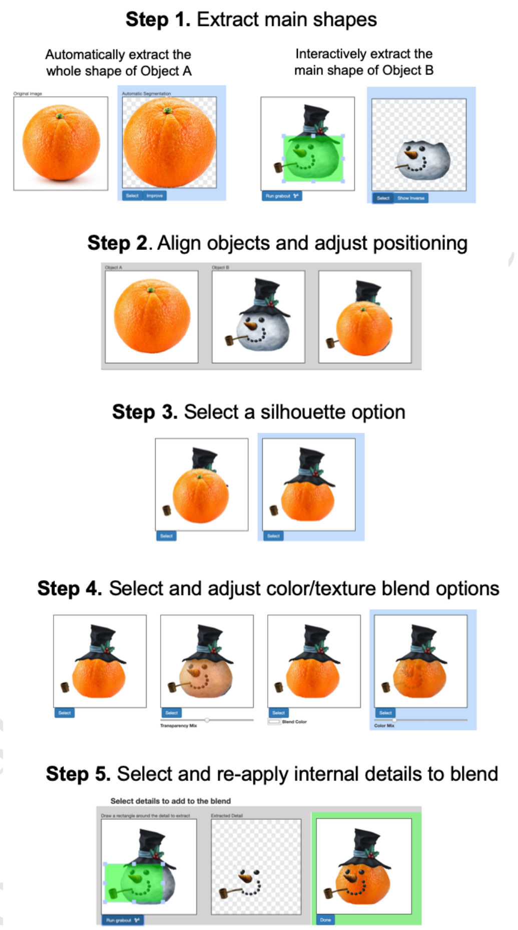

Fig. 5. The five steps of the VisiFit pipeline for improving blends. There are two options for silhouette, 5 options for color blending (only 4 are shown), and a tool to select and re-apply internal details. Each step builds on the selected output from the previous step (the blue border indicated the option has been selected.) The end of the iteration is highlighted by the green border.

Step 1. Extract main shapes When the page loads, the system shows Object A and the results of automatic cropping. Object A is an image of a single object that we want removed from its background. This is a classic computer vision problem: segmenting the salient object in an image. Deep learning approaches have been reported to be a fast and accurate approach to automatic object extraction, so we use the Tensorflow implementation of a pre-trained model for deeply supervised salient object detection [24] and use the mask it provides to crop the images.

The user sees the output for Object A and decides if it is acceptable. If it is, they select it and move to the next step. If not, they can decide to improve the object using Interactive Grabcut [46], a traditional computer vision algorithm for foreground extraction.

For Object B, users must use Interactive Grabcut to extract the main shape from the image. Our provided interface for Interactive Grabcut has users first draw a rectangle that encloses the entire object to extract. Then it produces a foreground extraction shown to users, who can mark extraneous pieces for removal by drawing on the image and running Grabcut again.

We used a classic interactive approach rather than a fully automatic approach because identifying parts or shapes within an image is very difficult. Traditional automatic approaches like Hough Transforms [14] do not work well on most images. Deep learning approaches are fairly good at segmenting objects within images [20] but are not yet capable enough at identifying the internal parts of objects.

Step 2. Automatically align objects and adjust position. After both objects have had their main shape cropped, the system automatically produces a new prototype using simple affine transformations that move, scale, position, and rotate the objects. Users are free to adjust the alignment with direct manipulation on the Fabric.js HTML5 canvas, just as they would in any image editing application.

Step 3 Select a silhouette option. When blending two objects, the blend can use the silhouette of either Object A or B, because they are very close in shape and size. The system automatically creates two versions of the blend - one with the silhouette of Object A and one with the silhouette of Object B. The user must select which silhouette looks better.

To create the two silhouetted prototypes, the system uses the inverses of the cropped images from Step 1, layers one inverse on top of the other original image, and positions them according to the coordinates in Step 2. This effectively creates a mask to produce the silhouette of the object.

Step 4. Select and adjust color and texture blend options. Color is the next fundamental dimension to include in the blend. There are 5 options for color blending. The user can keep the original colors, or use one of four adjustable tools to blend on color and texture:

- *Transparency*. We layer Object A onto Object B with 50% transparency to allow both colors and textures to come through, although somewhat weakly. The user can adjust the transparency level with a slider.
- *Color Blend*. We use K-means clustering to determine the most common color in the main shape of Object B. We then do an additive color blend with the color of Object A. This only works well when one object is very light - otherwise the color turns very dark.
- *Multiply colors*. Multiplying two images is a way to combine colors and textures in a way that preserves characteristics from both. Whereas transparency will always balance between the two, multiplication can surface both of the textures simultaneously. All three examples in Figure 4 use Multiply to blend colors. For example, in the Lego and ring example, multiplying colors allowed the Lego to take on the red color but keep the textures of both objects - the facets of the the diamond and the bumps on the Lego.
- *Replace color*. We use K-means clustering to determine the most common colors in the main shapes of Object A and B. We replace Object A's most common color with Object B's most common color and provide users with an adjustable threshold controlling the degree of color replacement. They can also choose to blend the image with colors they select from an eye dropper tool (not shown in Figure 5).

Step 5. Select and re-apply internal details to blend. The last visual dimension to include is internal details - these are smaller objects or salient features that help identify the object. In the *snowman* and *orange* blend, the snowman is not as iconic without his facial details. Thus, we want to extract them from the original Object B and place them back on Object A. Again, we use Interactive Grabcut to allow the user to select and refine what details to extract. While we could have used other tools such as context-aware select, Grabcut worked well on our test set and was a method users had already become familiar with in earlier stages of the pipeline.

VisiFit encourages users to follow a linear workflow through each of the tools. They can see effects previewed on their iteration and choose whether or not to include them. But users are not constrained to one path through the pipeline; they can take multiple paths and add an unrestricted number of edits to the fundamental dimensions if they so choose. The linear workflow is the default because it allows users to start on a simple path through their structured iteration. At the end, the user selects the blend they are most satisfied with and the system finishes by showing them the initial blend and the improved blend side by side.

6 EVALUATION

To evaluate whether VisiFit helps novice designers substantially improve prototypes of visual blends, we conducted a user study where participants used VisiFit to improve 11 VisiBlends prototypes. Two experts then rated those blends to judge whether they were substantially improved over the initial prototype.

To choose the prototypes to improve, we first listed all the blends mentioned in VisiBlends and found 15 candidates. Of these, 2 were already good blends and did not need improvement. Two others had significant similarities to blends used in the analysis and formative studies, having blended upon the same or similar objects. Hence, they would not have been fair to use in the evaluation and were thus excluded. This left an evaluation set of 11 diverse blends for different objects.

We recruited 11 novice designers (7 female, average age = 21.5) for a 1-hour long study who were paid $20 for their time. First, they were introduced to the concept of visual blends and shown examples of initial prototypes with their improved versions. Then, they had two blends to practice using the tools on. During this practice session, the experimenter answered questions, demonstrated features, and gave suggestions on how to use the tool.

In the next 44 minutes, participants used the segmentation tools to extract the main objects from all 22 images (System Steps 1 and 2) and blend the pairs into 11 improved blends (System Steps 3, 4, and 5). They had two minutes for Steps 1 and 2 and another two minutes for Steps 3 and 4, for a total of 4 minutes to create each blend. All results were saved by the system.

After the data was collected, we paid two expert graphic designers $60 per hour to look at every iterated blend and answer two questions for each of them:

- Does the iterated blend present substantial improvement over the prototype?
- Is the iterated blend of sufficient quality to post on social media?

The most important question to answer was the first one: does the tool help with substantial improvements? Small flaws in the execution were allowed, but the objects had to be seamlessly and aesthetically blended to count as an improvement. Our second question was how often these iterated blends were good enough for social media publication (i.e. a student club announcement post). Publication would mean that both objects were clearly identifiable and blended with no pronounced flaws.

Social media is much more forgiving than print publication. Print publications must be pixel-perfect, well-lit, and high definition. To meet this bar, a graphic designer should still use a professional tool like Photoshop. However, on social media, the images are often smaller, lower resolution, published more frequently, and for a smaller audience (such as student clubs, classes or majors) - so perfection is not as important. Additionally, the prevalence of low-fidelity user-generated content like memes and self-shot videos lowers the expectation of precision on social media, placing the emphasis on the message.

6.1 Results

During the study, the 11 participants attempted to improve a total of 121 blends. Six data points were lost due to errors in the saving process, leaving 115 blends as data points. The judges were introduced to their task with examples of prototypes and their VisiFit-improved counterparts, like the pairs seen in Figure 4 (which were done by the authors with graphic design background). For calibration, judges were shown blends of varying quality, to demonstrate what was considered "substantial improvement" and what was considered "suitable for publication on social media".

After studying the blends resulting from each participant, the judges answered our two questions for all Visifit-improved blends. Both questions on "improvement" and "suitability for publication" were highly subjective; however, the raters had "fair agreement" on both questions. They agreed on "substantial improvement" 71.3% of the time (κ = .23) and agreed on "suitability for publication" 73.9% of the time (κ = .37). In particular, there was one blend which they disagreed on every time. Both raters had well-reasoned answers for their differences and rather than forcing them to agree or introducing another rater, we split the difference and looked at the overall average rates of "substantial improvement" and "suitability for publication" to report the success of the tool.

Overall, people using the tool made substantial improvements to the blend 76.1% of the time. Additionally, those blends were of publishable quality 70.4% of the time. These metrics demonstrate how Visifit enables novices to quickly and easily complete a difficult iteration task.

Judges reported that blends were substantially improved when the parts of the objects looked correctly layered. This effect was achieved in a number of ways through Visifit: when the silhouette tool was used to mask one object and produce clean borders, when the internal detail extraction tool foregrounded important parts of the bottom image, (i.e. the acorn hat detail in the Guggenheim-acorn blend of Figure 1), or when the colors were blended compositely (i.e. the corn and McDonald's blend in Figure 6.)

For 10 of the 11 images, it was possible for at least one of the 11 participants to create an improved and publishable blend. There are several possible reasons why there was variability in user performance. One was subjectivity; some novice users were able to create high quality blends but chose versions that the judges did not rate as improvements. Judging one's own work is hard, because creators grow attached to their work and struggle to see it objectively.

A second and more important reason is the limitation of some of the tools. Cropping entire objects, applying a silhouette, and all four methods of blending colors worked as expected every time. However, the Interactive Grabcut tools for extracting parts of objects was sometimes problematic, since some details were too small to extract properly. While Grabcut is fast and easy, it does not have pixel-level precision. It often helped to improve the blend, but it sometimes weakened their suitability for publication. The Visifit-improved blend could still be used as a guide when creating a pixel-perfect version in a professional image editing tool. For example, for the blend of the orange slice and the barbeque grill featured in Figure 1, the idea of the blend is clear and improved, but the execution had enough flaws for it to be not suitable for publication.

There was one prototype that no user was able to improve. The burger and light bulb blend (Figure 6) left a seam between the burger and the light bulb every time. Similarly, a blend in the test set had a same problem - in the earth + ice cream blend, the melting part of the ice cream was not totally colored with an earth texture. These two examples pointed out a limitation of our tool and a potential feature we could implement. For both examples, a fill tool could have reduced the appearance of seams.

Overall, given the speed of the tool, participants thought that the results were well worth the effort they put into it [9]. During the study, several participants mentioned that the tool was fast and produced results they would not otherwise know how to achieve.

7 DISCUSSION

The two main contributions of this paper are the *fundamental dimension decomposition (FDD)* technique that structures iteration and the VisiFit system that helps novices iteratively improve blends with a pipeline of computational tools. In this discussion we want to explore how the

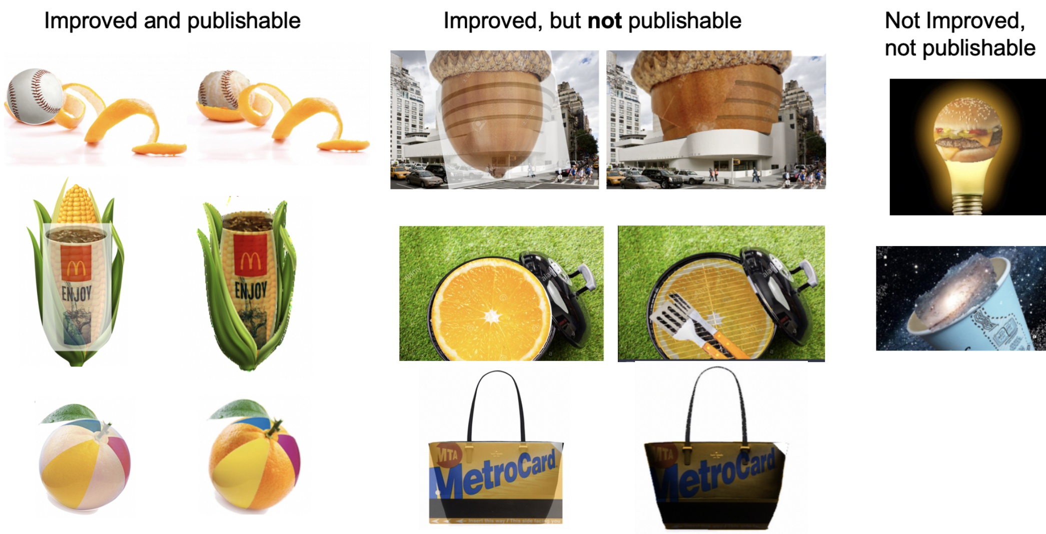

Fig. 6. Pairs of initial prototypes and Visifit-improved prototypes from the VisiFit user study. Three blends were evaluated "improved and publishable"; three blends were evaluated "improved but not publishable" and two blends were evaluated "not improved and not publishable".

computational tools could generalize to the needs of expert designers and how FDD can be applied to domains beyond visual blends. Additionally, we discuss the intellectual and engineering challenges that come with applying FDD to new domains, as well as relevant limitations.

7.1 Professional designers' impressions of VisiFit

Although VisiFit is meant to help novices, we had co-designed it with 2 graphic artists who were eager to use it as a rapid prototyping tool despite their prior domain knowledge. Thus, we wanted to see what impressions experts would have on the system and showed the tool to two professional designers (D1 and D2). D1 is a media communications director at a medium-sized organization with over twenty years of experience. D2 is a freelance graphic design with over 10 years of experience. Both expressed a need to efficiently create novel and eye-catching visuals for social media that are beyond the quality produced by tools such as Canva. Both designers had used visual blends in their professional work before, but did not know the name for the concept and did not have a strategy for producing them.

We presented them with the same blend examples from the user study and asked them to perform the same task: use the tool to iterate on the blend prototypes and create seamless and aesthetic blends. Both were impressed by how quickly and easily the blending tools helped them explore the design space. All of the basic operations were familiar to them from their experience with Photoshop, but they expressed surprise and relief to see results generated so quickly. D2 said *"Sometimes I spend hours pixel pushing just to test an idea. I love being able to test an idea quickly."*. D1 likened it to filter previews on Instagram which she loves to use to make photos more interesting on social media. Even for professional designers who are adept at using pixel-perfect tools, there is a need to provide high-level tools that can preview results without low-level manipulation (Design Principle 3).

When using VisiFit, both made blend improvements in a manner different from novice designers. D1 especially liked to push the boundaries, to try extracting and blending the less non-obvious options within the fundamental dimensions. D1 almost always started by looking at the inputs and formulating a plan. However, as D1 proceeded through the workflow, she found better and surprising ideas from the flare and focus nature of VisiFit. The system helped D1 explore the design space while keeping multiple threads open at a time. From this interaction, we believe that structuring blend improvement around fundamental dimensions has value even for professional designers (Design Principle 2).

D2 was impressed by the way the computational tools worked and particularly so for object extraction. He found Interactive Grabcut impressive in how effective it was on shape extraction but unimpressive in how unsuccessful it could be when selecting internal details. After multiple attempts with the tool, he noted that he would have preferred either better precision during user interaction or a better automatic approach. This raised an important limitation - VisiFit only provided one tool to extract internal details. Having a back-up tool (such as shaped-based cropping) could have relieved user frustration. This reinforces Design Principle 1 - that automatic tools don't always achieve desired results - and stresses that the system must provide *multiple* interactive tools specific for each subtask so that users have control over the creative process.

Overall, we believe that computational design tools for structured iteration can be as useful to professional designers as they are to novices. Both groups need to explore design spaces quickly and easily. Although experts have the ability to do this with existing tools, a pipeline of computational design tools could make this more efficient and attainable for designers of all experience levels.

7.2 Generalization to other blending problems

While the VisiFit system is tailored to the domain of visual blends, we believe the technique of *fundamental dimension decomposition* and the design principles behind it can be used to help novices structure iteration for other blending domains. We discuss three domains that FDD could be generalized to: animated visual blends, furniture design, and fashion design.

7.2.1 Animated Visual Blends. One way to further enhance visual blends is to add motion to them. Although it would be easy to add arbitrary motion, it would be ideal if the motion complemented the message. The top panel of Figure 7 shows a visual blend for *condom* and *action* that implies the message "condoms are for action." (The clapperboard is a symbol of *action*.) This blend is already effective at conveying the message, but to enhance it, we could add motion from the clapperboard onto the condom wrapper. We call this type of motion graphic an *animated blend*.

We propose to structure iteration during the creation of animated blends with fundamental dimension decomposition. Instead of decomposing the dimensions of object recognition, we can decompose the dimensions of motion. To create one, start with a static visual blend, and find a reference video of the motion made by one or both of the objects. Next, decompose the reference motion into the following fundamental dimensions: the pattern of motion (i.e. path segments, circular motion, appearance/disappearance, expansion/contraction, or gradient changes), the speed of motion, acceleration, and timing of pauses. Add and adjust these dimensions of motion on the static blend to create an aesthetic and seamless animation.

Figure 7 shows three visual blends, reference videos of one or both objects in motion (which have the motion annotated in red), and an example of how these dimensions of motion can be added to the visual blend and adjusted for a seamless aesthetic animated blend.

***Condom* and *action* animation**. This is the simplest case, where the path and speed of the reference video can be mapped to the visual blend with almost no adjustments needed. The top of clapperboard goes up slowly and accelerates down quickly, in a circular motion hinged around the red dot. These key points can be mapped to key points in the visual blend to transfer the action of the motion. However, when the mapping between the reference motion and the blend is not as close, we need to decompose and blend the dimensions more.

***Astronaut* and *food* animation**. In the reference video, the astronaut travels in a linear path at a fairly low, constant speed until he floats out of screen and disappears. While the speed of motion can be applied to the blend, the path should be changed to create a smooth loop. Thus, the path is changed so the astronaut moves in a square that can loop, but the speed stays the same.

***Tea* and *sunrise* animation**. In the reference video, the teabag is moved in a dipping motion, with slow downward and quick upward acceleration. The downward motion can be directly applied to the sun, but the upward motion looks better when it is slowed and with little acceleration, like the gradual motion of a sunrise. Additionally, the sun rising in the morning has a gradient change effect on the background which makes the sky look lighter as the sun rises. This gradient change can be applied to the background of the animated blend and amplified for larger visual impact. Thus, multiple dimensions of motion (paths/speeds, gradient changes) from two reference videos were blended to accentuate the visual blend.

Fig. 7. Three examples of visual blends that could be turned into animated blends. Each row shows the base visual blend on the left, the reference video in the middle, and the concept of the animated blend on the right. We annotate the motion on the reference videos and the animated blend in red.

Computational tools would be needed to extract and reapply the aforementioned fundamental dimensions of motion. Parameters for each fundamental dimension would become points of interactions for users. These tools could then be chained together into a pipeline similar to Visifit, to structure the iterative improvement of animated blends.

7.2.2 Fashion and Furniture Design. In furniture and fashion design, one type of problem is to combine different styles to achieve a new purpose. One way to do this is arguably a type of blending - to borrow from the functional and stylistic elements of both styles to create a new hybrid style. Two examples of hybrid styles include athleisure clothing and "updated classics" in furniture design.

- Athleisure is a clothing style that takes the fabrics and styles of athletic clothing and adapts them to non-athletic environments such as work, school, or other social environments [23, 47]. It provides the comfort and sleek look of athletic clothing while being acceptable to wear in social settings.
- Updated Classics is a style of furniture design that takes the rich feel of classic furniture (often dating to 1700's Europe) and adapts it to modern life, which requires products to be

easy to clean, space-saving, and compatible with modern electronic usage (i.e. accounting for power outlets and charging cables).

We propose that creating items within these hybrid styles could be structured using *fundamental dimension decomposition*. A tool implementing FDD would identify what type of item a user would want to create and what dimensions users would want to pull from the two style-distinct objects. The tool would guide users to blend the appropriate elements of each dimension into a single new product. For example, a fashion designer could operate on the fundamental dimensions of material, silhouette (neckline, hemline, leg width, etc.), color/pattern, fabrication (seam placement, grain direction, etc.), and details (closures, stitching, etc.). When combining these dimensions to create a hybrid style such as athleisure, designers often use the stretchy material of athletic clothing, the details and colors of street clothing, and a mix of silhouettes found in the gym, street, or workplace. This combination of traits helps designers achieve both comfort and socially appropriate styles.

A similar set of dimensions could be used for furniture design to achieve a blend of classic sophistication with modern convenience. For example, an "updated classic" chair could use the classic shape of a Louis XIV chair, but fabricate it out of plastic (as is common in modern chairs) to make it easier to move and clean. It could also reduce some of the ornamentation on the silhouette to take on aspects of a minimal modern look.

We believe this blending process can also be structured with a suite of chained computational tools. This process would certainly have to be interactive, using human judgment not only to guide the search, but also to constantly consider aspects outside the dimensions such as the social acceptability of the design, the appeal to the target market, and whether its construction is feasible within desired price points.

7.3 Limitations

The major intellectual challenge of applying FDD to a new domain is discovering what its fundamental dimensions are. For VisiFit, we were able to observe the fundamental dimensions from examples and from co-design sessions. Additionally, we were guided by what is known about the neuroscience of human visual object recognition. If one or more of those approaches is not available in a new domain, significant trial and error may be required to identify those dimensions. An exciting challenge would be to use computational tools to automatically (or semi-automatically) discover the fundamental dimensions of a new domain.

The major engineering challenge of applying the design principles behind FDD to a new domain is to find or build computational tools that can help explore each dimension with high-level tools rather than low-level manipulation. Deep learning has provided new hope for such tools, but there are still limitations to what deep learning systems can do, especially with limited data. This is an open challenge: to quickly create new computational tools for the fundamental dimensions of new domains.

For any new blending domain, there is also the possibility that some blends are too complex to be structured around fundamental dimensions due to complex interactions between dimensions. For example, when the DNA of two parents are combined to make a offspring, the offspring certainly has a blend of the parents features, but there are so many features that the combinations become too complex to choose from. There may be too many dependencies between fundamental dimensions that make designing at a high level impossible. When considering the fundamental dimensions of a new domain, one should look out for such dependencies.

8 CONCLUSION

Iterative improvement is the essence of the iterative design process. Although there are many existing tools that support other phases of the design process - brainstorming, prototyping, evaluation, and final design execution, there are a lack of tools focusing on iteration [18]. We present a tool that helps novices iteratively improve on the graphic design challenge of creating visual blends. Visual blends combine two visual symbols into one object to convey a new meaning. Tools already exist to help novices create initial prototypes of visual blends, however, novices do not have tools or strategies to support them through iteration from low-fidelity to high-fidelity blends.

We conducted three preliminary investigations on how to iteratively improve visual blends. This included an exploration of automatic tools, analysis of expert examples, and a co-design process with graphic designers. From these studies we derived three design principles that can be employed in the creation of iteration tools, as well as a general design technique called *fundamental dimension decomposition*. This technique structures iterative improvement into a sequence of computational tools that helps novices explore the design space evolving during iteration. For visual blends, the fundamental dimensions are color/texture, silhouette, and internal details. The computational tools we implemented to explore each of these dimensions used a combination of deep learning, computer vision techniques, and parametric control for fine-tuning.

The principles and technique are demonstrated through our system VisiFit - a pipeline of computational design tools to iterate upon visual blends. Our evaluation shows that when using VisiFit, novices substantially improve blends 76% of the time. Their blends were of sufficient quality for publication on social media 70% of the time.

Although creating visual blends is a domain-specific problem, it is emblematic of many design challenges which involve the blending or remixing of existing things to produce novel meaning or purpose. We discuss how these principles could be reapplied in three other blending domains: animated blends and hybrid styles of furniture and clothing. These domains have their own fundamental dimensions which can be used to structure the iterative improvement process.

ACKNOWLEDGMENTS

Removed for anonymity

REFERENCES

[1] 1972. Design Q A: Charles and Ray Eames. https://www.hermanmiller.com/stories/why-magazine/design-q-and-a-charles-and-ray-eames/

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[3] Maneesh Agrawala, Wilmot Li, and Floraine Berthouzoz. 2011. Design Principles for Visual Communication. *Commun. ACM* 54, 4 (April 2011), 60–69. https://doi.org/10.1145/1924421.1924439

[4] Pete Barry. 2016. *The Advertising Concept Book: Think Now, Design Later (Third).* Thames & Hudson, London, UK. 296 pages.

[5] Gilbert Louis Bernstein and Wilmot Li. 2015. Lillicon: Using Transient Widgets to Create Scale Variations of Icons. *ACM Trans. Graph.* 34, 4, Article 144 (July 2015), 11 pages. https://doi.org/10.1145/2766980

[6] Dino Borri and Domenico Camarda. 2009. The Cooperative Conceptualization of Urban Spaces in AI-assisted Environmental Planning. In *Proceedings of the 6th International Conference on Cooperative Design, Visualization, and Engineering* (Luxembourg, Luxembourg) *(CDVE'09)*. Springer-Verlag, Berlin, Heidelberg, 197–207. http://dl.acm.org/citation.cfm?id=1812983.1813012

[7] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. 2017. Learning Visual Importance for Graphic Designs and Data Visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. ACM, New York, NY, USA, 57–69. https://doi.org/10.1145/3126594.3126653

[8] Zoya Bylinskii, Nam Wook Kim, Peter O'Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. 2017. Learning Visual Importance for Graphic Designs and Data Visualizations. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 57–69. https://doi.org/10.1145/3126594.3126653

[9] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (June 2014), 25 pages. https://doi.org/10.1145/2617588

[10] Lydia B. Chilton, Savvas Petridis, and Maneesh Agrawala. 2019. VisiBlends: A Flexible Workflow for Visual Blends. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 172, 14 pages. https://doi.org/10.1145/3290605.3300402

[11] Nicholas Davis, Chih-PIn Hsiao, Kunwar Yashraj Singh, Lisa Li, Sanat Moningi, and Brian Magerko. 2015. Drawing Apprentice: An Enactive Co-Creative Agent for Artistic Collaboration. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition* (Glasgow, United Kingdom) *(Camp;C '15)*. Association for Computing Machinery, New York, NY, USA, 185–186. https://doi.org/10.1145/2757226.2764555

[12] Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. GRIDS: Interactive Layout Design with Integer Programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376553

[13] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 845–854. https://doi.org/10.1145/3126594.3126651

[14] Richard O. Duda and Peter E. Hart. 1972. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* 15, 1 (Jan. 1972), 11–15. https://doi.org/10.1145/361237.361242

[15] Morwaread M. Farbood, Egon Pasztor, and Kevin Jennings. 2004. Hyperscore: A Graphical Sketchpad for Novice Composers. *IEEE Comput. Graph. Appl.* 24, 1 (Jan. 2004), 50–54. https://doi.org/10.1109/MCG.2004.1255809

[16] G. Fauconnier and M. Turner. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.

[17] Charles Forceville. 1994. Pictorial Metaphor in Advertisements. *Metaphor and Symbolic Activity* 9, 1 (1994), 1–29. https://doi.org/10.1207/s15327868ms0901_1 arXiv:http://dx.doi.org/10.1207/s15327868ms0901$_1$

[18] Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. 2019. Mapping the Landscape of Creativity Support Tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 389, 18 pages. https://doi.org/10.1145/3290605.3300619

[19] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. 2010. Automatically Generating Personalized User Interfaces with Supple. *Artif. Intell.* 174, 12-13 (Aug. 2010), 910–950. https://doi.org/10.1016/j.artint.2010.05.005

[20] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. 2018. Detectron. https://github.com/facebookresearch/detectron.

[21] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective Physical Prototyping Through Integrated Design, Test, and Analysis. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. ACM, New York, NY, USA, 299–308. https://doi.org/10.1145/1166253.1166300

[22] Narayan Hegde, Jason D Hipp, Yun Liu, Michael Emmert-Buck, Emily Reif, Daniel Smilkov, Michael Terry, Carrie J Cai, Mahul B Amin, Craig H Mermel, Phil Q Nelson, Lily H Peng, Greg S Corrado, and Martin C Stumpe. 2019. Similar image search for histopathology: SMILY. *npj Digital Medicine* 2, 1 (2019), 56. https://doi.org/10.1038/s41746-019-0131-z

[23] Elizabeth Holmes. 2015. Athleisure: A Workout Look for Every Occasion. https://www.wsj.com/video/athleisure-a-workout-look-for-every-occasion/D0174829-3288-40F1-9E12-0F420E38AA9A.html

[24] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. 2017. Deeply Supervised Salient Object Detection with Short Connections. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5300–5309.

[25] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*.

[26] P. Karimi, M. L. Maher, K. Grace, and N. Davis. 2019. A computational model for visual conceptual blends. *IBM Journal of Research and Development* 63, 1 (2019), 5:1–5:10.

[27] P. Karimi, M. L. Maher, k. Grace, and N. Davis. 2019. A Computational Model for Visual Conceptual Blends. *IBM J. Res. Dev.* 63, 1, Article 1 (Jan. 2019), 10 pages. https://doi.org/10.1147/JRD.2018.2881736

[28] Joy Kim, Maneesh Agrawala, and Michael S. Bernstein. 2017. Mosaic: Designing online creative communities for sharing works-in-progress. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*. Association for Computing Machinery, New York, New York, USA, 246–258. https://doi.org/10.1145/2998181.2998195 arXiv:1611.02666

[29] Joy Kim, Sarah Sterman, Allegra Argent Beal Cohen, and Michael S. Bernstein. 2017. Mechanical novel: Crowdsourcing complex work through reflection and revision. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*. Association for Computing Machinery, 233–245. https://doi.org/10.1145/2998181.2998196 arXiv:1611.02682

[30] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. 2019. May AI? Design ideation with cooperative contextual bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[31] Ranjitha Kumar, Arvind Satyanarayan, Cesar Torres, Maxine Lim, Salman Ahmad, Scott R. Klemmer, and Jerry O. Talton. 2013. Webzeitgeist: Design Mining the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, New York, NY, USA, 3083–3092. https://doi.org/10.1145/2470654.2466420

[32] James A. Landay. 1996. SILK: Sketching Interfaces Like Krazy. In *Conference Companion on Human Factors in Computing Systems* (Vancouver, British Columbia, Canada) *(CHI '96)*. ACM, New York, NY, USA, 398–399. https://doi.org/10.1145/257089.257396

[33] James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. 2000. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. ACM, New York, NY, USA, 510–517. https://doi.org/10.1145/332040.332486

[34] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. 2010. TurKit: Human Computation Algorithms on Mechanical Turk. In *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. ACM, New York, NY, USA, 57–66. https://doi.org/10.1145/1866029.1866040

[35] J. Derek Lomas, Jodi Forlizzi, Nikhil Poonwala, Nirmal Patel, Sharan Shodhan, Kishan Patel, Ken Koedinger, and Emma Brunskill. 2016. Interface Design Optimization As a Multi-Armed Bandit Problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 4142–4153. https://doi.org/10.1145/2858036.2858425

[36] Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. 2020. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376739

[37] Kurt Luther, Amy Pavel, Wei Wu, Jari-lee Tolentino, Maneesh Agrawala, Björn Hartmann, and Steven P. Dow. 2014. CrowdCrit: Crowdsourcing and Aggregating Visual Design Critique. In *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA) *(CSCW Companion '14)*. ACM, New York, NY, USA, 21–24. https://doi.org/10.1145/2556420.2556788

[38] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. 1997. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 389–400. https://doi.org/10.1145/258734.258887

[39] Justin Matejka, Michael Glueck, Erin Bradner, Ali Hashemi, Tovi Grossman, and George Fitzmaurice. 2018. Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 369, 12 pages. https://doi.org/10.1145/3173574.3173943

[40] Brad A. Myers, Ashley Lai, Tam Minh Le, Young Seok Yoon, Andrew Faulring, and Joel Brandt. 2015. Selective undo support for painting applications. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 2015-April. Association for Computing Machinery, New York, New York, USA, 4227–4236. https://doi.org/10.1145/2702123.2702543

[41] J. Nielsen. 1993. Iterative user-interface design. *Computer* 26, 11 (1993), 32–41.

[42] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. [n.d.]. Learning Layouts for Single-Page Graphic Designs. *IEEE Transactions on Visualization and Computer Graphics* 20 ([n. d.]).

[43] Francisco Pereira. 2007. *Creativity and AI: A Conceptual Blending Approach.*

[44] Savvas Petridis and Lydia B. Chilton. 2019. Human Errors in Interpreting Visual Metaphor. In *Proceedings of the 2019 on Creativity and Cognition* (San Diego, CA, USA) *(Camp;C '19)*. Association for Computing Machinery, New York, NY, USA, 187–197. https://doi.org/10.1145/3325480.3325503

[45] Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S. Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S. Bernstein. 2014. Expert Crowdsourcing with Flash Teams. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. ACM, New York, NY, USA, 75–85. https://doi.org/10.1145/2642918.2647409

[46] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM SIGGRAPH 2004 Papers* (Los Angeles, California) *(SIGGRAPH '04)*. ACM, New York, NY, USA, 309–314. https://doi.org/10.1145/1186562.1015720

[47] Sam Sanders. 2015. For The Modern Man, The Sweatpant Moves Out Of The Gym. https://www.npr.org/2015/04/08/397138654/for-the-modern-man-the-sweatpant-moves-out-of-the-gym

[48] Ben Shneiderman. 2007. Creativity Support Tools: Accelerating Discovery and Innovation. *Commun. ACM* 50, 12 (Dec. 2007), 20–32. https://doi.org/10.1145/1323688.1323689

[49] Pao Siangliulue, Joel Chan, Steven P. Dow, and Krzysztof Z. Gajos. 2016. IdeaHound: Improving Large-scale Collaborative Ideation with Crowd-Powered Real-time Semantic Modeling. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. ACM, New York, NY, USA, 609–624. https://doi.org/10.1145/2984511.2984578

[50] Vikash Singh, Celine Latulipe, Erin Carroll, and Danielle Lottridge. 2011. The Choreographer's Notebook: A Video Annotation System for Dancers and Choreographers. In *Proceedings of the 8th ACM Conference on Creativity and Cognition* (Atlanta, Georgia, USA) *(Camp;C '11)*. Association for Computing Machinery, New York, NY, USA, 197–206. https://doi.org/10.1145/2069618.2069653

[51] Gillian Smith, Jim Whitehead, and Michael Mateas. 2010. Tanagra: A Mixed-initiative Level Design Tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (Monterey, California) *(FDG '10)*. ACM, New York, NY, USA, 209–216. https://doi.org/10.1145/1822348.1822376

[52] Robert J Sternberg. 2011. *Cognitive Psychology*.

[53] Sou Tabata, Hiroki Yoshihara, Haruka Maeda, and Kei Yokoyama. 2019. Automatic Layout Generation for Graphical Design Magazines. In *ACM SIGGRAPH 2019 Posters* (Los Angeles, California) *(SIGGRAPH '19)*. ACM, New York, NY, USA, Article 9, 2 pages. https://doi.org/10.1145/3306214.3338574

[54] Michael Terry and Elizabeth D. Mynatt. 2002. Side Views: Persistent, on-Demand Previews for Open-Ended Tasks. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France) *(UIST '02)*. Association for Computing Machinery, New York, NY, USA, 71–80. https://doi.org/10.1145/571985.571996

[55] Kashyap Todi, Jussi Jokinen, Kris Luyten, and Antti Oulasvirta. 2019. Individualising Graphical Layouts with Predictive Visual Search Models. *ACM Trans. Interact. Intell. Syst.* 10, 1, Article 9 (Aug. 2019), 24 pages. https://doi.org/10.1145/3241381

[56] Rajan Vaish, Shirish Goyal, Amin Saberi, and Sharad Goel. 2018. Creating Crowdsourced Research Talks at Scale. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1–11. https://doi.org/10.1145/3178876.3186031

[57] Margot van Mulken, Rob le Pair, and Charles Forceville. 2010. The impact of perceived complexity, deviation and comprehension on the appreciation of visual metaphor in advertising across three European countries. *Journal of Pragmatics* 42, 12 (2010), 3418 – 3430. https://doi.org/10.1016/j.pragma.2010.04.030

[58] Hao-Chuan Wang, Dan Cosley, and Susan R. Fussell. 2010. Idea Expander: Supporting Group Brainstorming with Conversationally Triggered Visual Thinking Stimuli. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work* (Savannah, Georgia, USA) *(CSCW '10)*. Association for Computing Machinery, New York, NY, USA, 103–106. https://doi.org/10.1145/1718918.1718938

[59] Jingdong Wang and Xian-Sheng Hua. 2011. Interactive image search by color map. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 1 (2011), 1–23.

[60] Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Tomoyasu Nakano, Satoru Fukayama, and Masataka Goto. 2017. LyriSys: An interactive support system for writing lyrics based on topic transition. In *International Conference on Intelligent User Interfaces, Proceedings IUI*. Association for Computing Machinery, New York, New York, USA, 559–563. https://doi.org/10.1145/3025171.3025194

[61] Ariel Weingarten, Ben Lafreniere, George Fitzmaurice, and Tovi Grossman. 2019. DreamRooms: Prototyping Rooms in Collaboration with a Generative Process. In *Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface 2019* (Kingston, Canada) *(GI'19)*. Canadian Human-Computer Communications Society, Waterloo, CAN, Article 19, 9 pages. https://doi.org/10.20380/GI2019.19

[62] Anbang Xu, Shih-Wen Huang, and Brian Bailey. 2014. Voyant: Generating Structured Feedback on Visual Designs Using a Crowd of Non-experts. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA) *(CSCW '14)*. ACM, New York, NY, USA, 1433–1444. https://doi.org/10.1145/2531602.2531604

[63] Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2014. Distributed Analogical Idea Generation: Inventing with Crowds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 1245–1254. https://doi.org/10.1145/2556288.2557371

[64] Lixiu Yu, Aniket Kittur, and Robert E. Kraut. 2014. Searching for Analogical Ideas with Crowds. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 1225–1234. https://doi.org/10.1145/2556288.2557378

[65] Lixiu Yu and Jeffrey V. Nickerson. 2011. Cooks or Cobblers?: Crowd Creativity Through Combination. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. ACM, New York, NY, USA, 1393–1402. https://doi.org/10.1145/1978942.1979147

[66] Zhenpeng Zhao, Sriram Karthik Badam, Senthil Chandrasegaran, Deok Gun Park, Niklas Elmqvist, Lorraine Kisselburgh, and Karthik Ramani. 2014. SkWiki: A multimedia sketching system for collaborative creativity. In *Conference on Human Factors in Computing Systems - Proceedings*. Association for Computing Machinery, New York, New York, USA, 1235–1244. https://doi.org/10.1145/2556288.2557394