VisiFit: AI Tools to Iteratively Improve Visual Blends

Lydia B. Chilton
Columbia University
New York, NY, USA
chilton@cs.columbia.edu

Ecenaz Jen Ozmen
Columbia University
New York, NY, USA
eo2419@columbia.edu

Sam Ross
Barnard College
New York, NY, USA
shr2118@barnard.edu

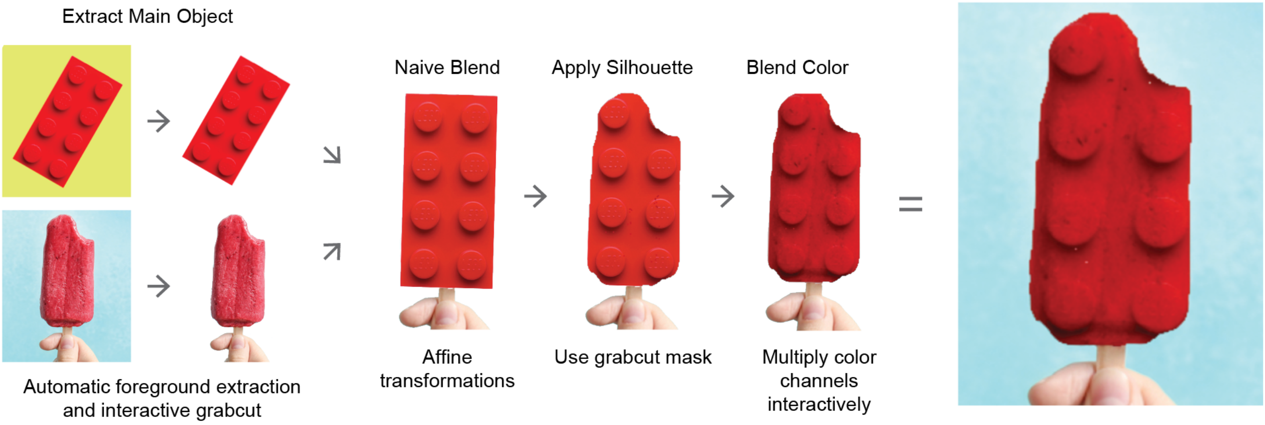


Figure 1. Iterative improvement of blend for *Lego* and *Summer* with VisiFit. AI and computer vision tools are used to 1) extract the main object from images, 2) position the images, 3) change the silhouette, 4) blend the textures and 5) extract and replace details from the hidden object (not used here).

ABSTRACT

Iterative improvement is essential to the design process. However, iterative improvement requires difficult decisions about what to iterate on and requires the time and expense of making multiple prototypes. With the current advances in AI, there is the potential that AI can reduce these expenses and augment peoples' ability to design. However, it is unclear what AI can reliably do and whether it should be fully automatic or if it needs human guidance. We explore how AI tools can assist novices in the difficult graphic design challenge of creating visual blends. First, we present four design principles for AI design tools based on co-design sessions with graphic designers. We introduce a system for iterating on visual blends by improving one visual dimension at a time. An evaluation of the tool on novices shows they can improve the blends beyond what existing novice tools can do in 97.5% of the cases and they produce publishable quality blends in 65% of the test cases. We discuss the implications for ways to combine human and computers' abilities in the design process.

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04. . . $15.00
DOI: https://doi.org/10.1145/3313831.XXXXXXX

Author Keywords

Design tools; artificial intelligence; computational design;

CCS Concepts

•Human-centered computing → Interactive systems and tools; *Participatory design;* **•Computing methodologies →** *Artificial intelligence; Computer vision problems;*

INTRODUCTION

Iterative improvement is the essence of the design process. The original spiral model of software design [2] characterizes the design process as a way to minimize the overall risk of failure; each iteration is a prototype that tests the next riskiest feature. Product design methodology is aligned with this. The Double Diamond model first explores the framing of the problem, then explores the space of solutions [23]. Each exploration uses multiple parallel prototypes, which has been shown to improve outcomes by exploring the space of solutions before picking the best one [6].

Although the iterative approach to design is generally accepted to be more successful than linear approaches, it creates major challenges such as 1) selecting what risks should be tested first, and 2) how to manage the time and expense of making multiple prototypes in parallel. With all the current advances in AI, there is the potential that AI can reduce these expenses and augment peoples' ability to design. However, it is unclear exactly how AI can be helpful. In particular, should AI be fully automatic and take all the burden of design or should it be interactive? If it takes on the full burden, this alleviates the time and attention novices need to spend to get results. But if the AI can't achieve good enough results, people cannot help give feedback to correct errors. On the other hand, if AI is of assistance, people can help guide it but they must have some design ability, taste, or knowledge in order to guide it in a good direction. When we explored fully automatic AI approaches to problem and we found that they consistently fall short in basic ways. There is a challenge to explore interactive tools that use peoples' abilities, but are powerful enough to alleviate the time and expense of the design process.

As a design challenge, we explore iteratively improving an advanced graphic design technique called *visual blends*. Visual blends blends two objects in a way that is novel and eye-catching - they are considered difficult and creative to make. Existing tools can create an initial prototype of a visual blend, given a concept pair such as *football* and *dangerous* or *Lego* and *summer vacation*. The next challenge is to iteratively improve on them, with the ultimate goal of enabling novices to produce publishable quality blends quickly and easily.

We introduce VisiFit - a system that for novice designers to iteratively improve visual blends. In each iteration, VisiFit helps users improve one visual dimension of the blend. First it improves the crop of the images, then silhouette, then the texture blend, and lastly the details of the blend. Each step is assisted by automated tools (Figure 1). The design of VisiFit is informed by: 1) formative studies of novices using existing end-user tools to identify their shortcomings and where novices need support, 2) analysis of visual blends created by professional designers 3) cognitive principles of visual object detection that underlie our ability to recognize objects, and 4) co-design with professional designers to verify the cognitive principles and incorporate their best practices in the tools.

This paper makes the following contributions:

- **Four design principles** for AI design tools based on formal studies with end-user tools, analysis of professional design, cognitive principles of visual processing, and co-design sessions with graphic designers.
- **A system** for iteratively improving visual blends based on blending one visual dimension dimension at a time: silhouettes of the objects, color and texture of the objects, internal details of the objects.
- **An evaluation** of fully-automatic vs. semi-automatic AI showing that semi-automatic approaches are needed in 50% of cases.
- **An evaluation** on novice designers shows they can improve the blends beyond what current novice tools can do in 98% of the cases and they produce publishable quality blends 65% of the test cases.

We conclude with a discussion of how expert designer find VisiFit useful and general approaches for AI to aid in the design process.

RELATED WORK

Deep Learning approaches to Blending

Blending images seems like an intuitive concept, but can actually mean many things in AI. Two popular types of blends are style transfer [15] which extracts the style of an image (typically a famous paining) and applies it to another photo. This makes it fast and easy to make any painting look like Van Gogh's Starry night. One limitation is that it works best for paintings with broad abstract styles and it does not preserve the semantics of the image. The moon in Starry night may show up where it does not belong like the ocean of the image. Another approach that works on photos is GanBreeder [14] which combines two images in visually interesting ways. Although the results are typically artistic, the objects are not typically identifiable in the result. They tend to blend in abstract ways.

There are some ways of preserving semantics in an image before blending them. FaceSwap [22] is an example of this. In FaceSwap, the computer is trained to know what faces are and knows how to extract the details of one face and put it onto another person without the appearance of seams. There are many compelling examples, but this approach benefits from a vast training set of faces. Additionally, faces all have the same features. Face swapping is less of a blending task and more of a texture mapping problem to map the details of one face to the details of another. Although many results are compelling, many of the blends don't look natural and could benefit from editing.

All of these have some interesting areas of application, but are not suitable for visual blends. Visual blends require that images stay crisp, not abstract, and the objects are identifiable. They also come from a wide range of objects, not specific images (like faces) or styles (like paintings). The semantics of the image are crucial to visual blends - what parts are visible and how they are blended.

Design Tools

Design Tools have a rich tradition of helping designers rapidly prototype and iterate [11, 17, 18]. A survey of tools supporting the design process for creative tasks [8] found that computational tools have facilitated all parts of the process. However, there are many more tools that focus on the early stages of the process like brainstorming [25], ideation [32] and search for similar graphic designs [16]. There is also work on the end of the design process, like critique and layout [20, 31]. There is work on generating multiple designs by tweaking parameters. This helps users cheaply and easily explore the design space or create multiple variations of objects such as trees or airplanes that are needed to make computer generated scenes more diverse [29, 21].

There is a lack of work on the later and middle stages of design. This is where the design process can become ill defined and hard to manage. This is a challenge in supporting the design process end-to-end in a single tool and to focus on stages beyond brainstorm and more towards iteration towards the goal.

AI-assisted design

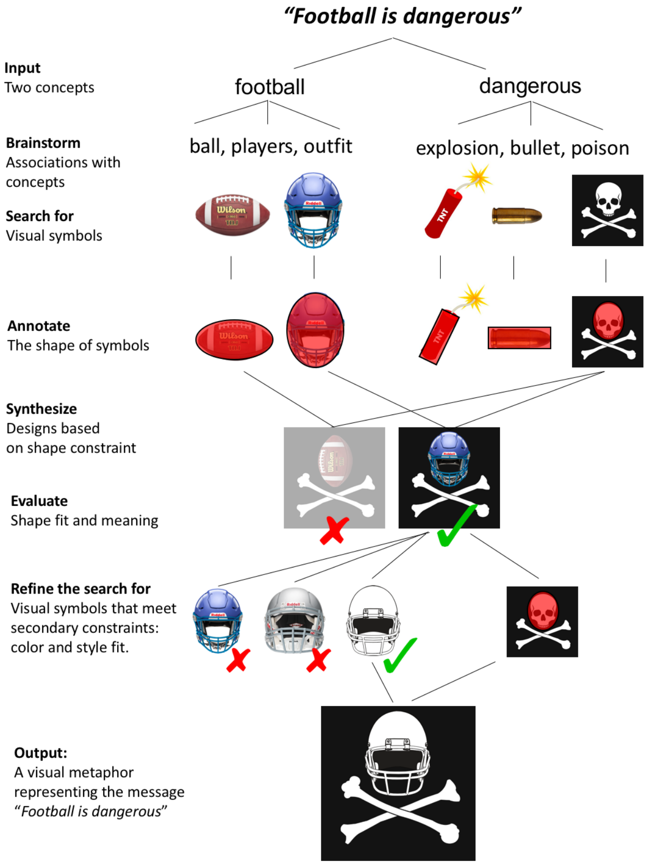
AI-assisted design has long been a promising approach in many fields in many fields even outside of graphic design such as: education [19], medicine [12], games [27], urban planning [3], and accessibility [9]. With advances we should be using advances in deep learning that can help use in design [4] but still be mindful of their potential failings when translating from evaluations on test sets to working on real problems.

BACKGROUND: VISUAL BLENDS

Visual Blends are a design challenge to fit two objects together such that they look blended. An existing VisiBlends [5] system helps novices create prototypes of visual blends by following the flare and focus design process. However, they must complete the finished design on their own, or by hiring a designer. Given two abstract concepts like *football* and *dangerous*, VisiBlends first helps users brainstorm many objects associated with both concepts, then find simple, iconic images of those concept. With the images, they identify the main shape of the object (sphere, cylinder, box, or a flat circle, or flat rectangle). It then automatically searches over pairs of object to find two that have the same basic shape. With those objects, it creates a rough 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, some editing is needed. This can be done by searching to find an object with a better shape fit, editing the objects, or paying an artist to execute a completed blend. Figure shows an illustration of the VisiBlends workflow.

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 flare and focus to mitigate the riskiest feature first, which is shape fit. This design insight is backed up by the neuroscience of visual object recognition which that 3D shape is the primary feature used by the brain to determine what an object is [28]. This is likely because 3D shape is the least mutable propriety of the object. Other features can change based on time or instance; color changes in different lighting conditions, and identifying details have variation among individuals (hair color, eye color, etc.). By using objects that have the same shape, you effectively confuse the visual system - the overall shape makes it look like a skull and crossbones, but the details make it look like a football helmet. This is what makes visual blends novel and eye-catching.

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 the secondary visual identifiers. The main other features that the brain's visual object recognition system uses are silhouette, color, texture, and internal details. The hypothesis we follow is that we can iteratively improve blends by allowing people to choose the silhouette, color, texture, and details from the two objects to be blended.


"Football is dangerous"
Input
Two concepts
football
dangerous
Brainstorm
Associations with concepts
ball, players, outfit
explosion, bullet, poison
Search for
Visual symbols
Annotate
The shape of symbols
Synthesize
Designs based on shape constraint
Evaluate
Shape fit and meaning
Refine the search for
Visual symbols that meet secondary constraints: color and style fit.
Output:
A visual metaphor representing the message "Football is dangerous"

. An illustration of the VisiBlends workflow to find a visual blend for the concepts *football* and *dangerous* based on shape fit.

FORMATIVE STUDIES AND DESIGN PRINCIPLES

Based on four formative studies of alternative approaches for improving blend prototypes, we derive design principles for designing systems that combine the abilities of people and AI to create visual blends.

Short-comings of Fully Automatic AI

Advances in deep learning have shown impressive results in manipulating images. An early and prominent result is Deep Style Transfer [15]. It trains a model of an image style, like Van Gogh's Starry Night, and can then apply that style of 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 the style of one object and applying it to another. Even it takes lots of machine time, it takes very little human time.

To explore the potential to use this fully automatic AI technique, we took four blend prototypes from the VisiBlends test set with blends made by paid artists and compared them to automatic style transfer results. We used an implementation of style transfer from the popular Fast Style Transfer (FST) paper [15]. We tried multiple combinations of hyper-parameters (epochs, batch size, and iterations) waiting up to 12 hours to train a model. We also tried input images of the same object

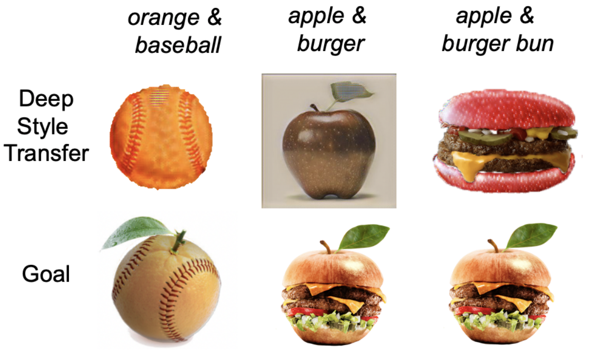


Figure 2. Blends created by fully a automatic approach compared to work by an artist.

and different ways of cropping in it in case the algorithm was sensitive to a particular image.

Although the algorithm was able to extract styles and apply them, the results fell far short of the bar. See figure 2. To blend *orange* and *baseball*, FST first learned the orange style. However, when it applied that learned style to the baseball, it preserves the characteristic red seams of a baseball, but it simply turned the white baseball a blotchy orange color that is not reminiscent of the fruit. In contrast, the artist who blended it use the texture of the orange, and the stem of the orange, in addition to the red baseball seams. This makes both objects highly identifiable. The computer used the overall look of the orange, but didn't consider it's elements separately in order to mix and match the parts.

Similarly, for the *apple* and *burger* blend, the burger style applied to the apple just turned the apple brown, because that's the predominant color of a burger. We also explored isolating a part of the image by hand and applying 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 doesn't 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 don't easily apply to visual blends. Blends are not just about applying high-level "style", they require considering 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'd have to create those thousands of blends, and even so, the results are not guaranteed. Instead we want to explore semi-automatic approaches that augment people's ability to create blends.

Design Principle 1. Instead of pursuing fully automatic approaches, break up the objects into components that can each be blended.

Analysis of artists blends

To investigate how artists use identifying elements to create blends, we analysed the thirteen input and blend images from VisiBlends to see how many needed professional editing and if so, what elements they took from each input image. We found that 2 of the 13 images needed no editing. The output from VisiBlends was a perfectly acceptable blend. The *football* and *dangerous* blend in Figure is an example of a blend that VisiBlends can execute. Here, the color and style of the helmet already matches the white color and line-drawing style of the skull and crossbones. A second iteration of the search was enough to improve the blend.

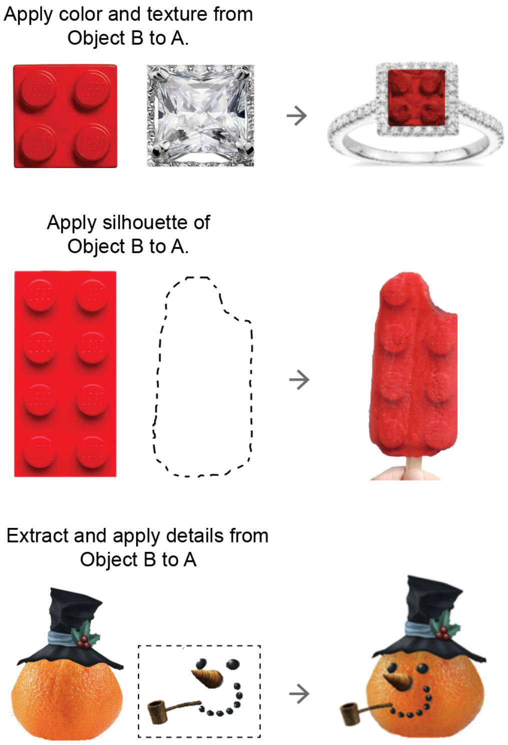

Apply color and texture from Object B to A.
Apply silhouette of Object B to A.
Extract and apply details from Object B to A

Figure 3. Three visual dimensions to iterate on when improving visual blends: color, silhouette, and details.

For the remaining 11 of 13 images, professional editing was needed. (The professional blends in Figure 2 are two examples of them.) There are three main visual dimensions the artists used to blend objects:

- **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)
- **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.)

Figure 3 shows examples of each of the three visual dimensions needed for blends. (The examples were made in VisiFit by experienced users).

Sometimes using one visual dimension is enough to blend on, but sometimes you use all three. Regardless, these dimensions are concepts artists seem to use. From a cognitive perspective, it makes sense. Our visual object recognition system uses several high-level features to determine what an object is. The primary feature is shape. The first pass of VisiBlends uses this as a basis for finding two objects that can be blended and creates a prototype based on that. In the terms of Boehms' spiral model, the shape fit is the primary feature and primary risk to prototype and test. However, our visual system uses secondary features to further identify an object, including its color, it's details and it's fine-grained silhouette. For example, in identifying a leaf, we might first use shape to identify it is a leaf, then use it's color and texture, details like spots, and silhouette, like the jaggedness of the leaf outline to identify what type of leaf it is. It makes sense that the second iteration of visual blends would use secondary principles of visual object recognition to blend on.

Design Principle 2. When iteratively improving the a blends consider three visual dimensions of an object: color, silhouette and details.

Co-Design with Graphic Artists

After analysing blends and identifying visual dimensions as a use abstraction to use, we worked with two graphic artists over an extended period of time to create and improve blends. We used this experience to either validate these principles or refine them. Both designers had Photoshop training and experience and had created numerous print ads, although neither had made visual blends before.

To start, the artists recreated some of the professional blends with no exposure to our tools. They didn't immediately know how to re-create the blend, but both trial and error to explore alternatives and was ultimately satisfied with the results. In their trials, thinking about the visual dimensions helped them come up with techniques they hadn't considered. Although blending colors and adding details were intuitive to them, using the silhouette of one object to crop the other was a an insight they were able to apply successfully.

In general, both designers thought that by restricting themselves to thinking based only on these tools they could recreate the most impressive visual blends in the test set. They did note that there were other techniques to improve blends like adding shadows and backgrounds, but that those could be added on top of the existing design principles, if needed.

The three visual dimensions seemed like sound principles to iterate on, and provided insight for experienced designers. However, their creation process involved a lot of trial and error. They wanted to try an idea and see if it worked. Photoshop has some tools to help with this, but the artists still spent a lot of time manipulating pixels to create each version.

Design Principle 3. When deciding how to apply each visual dimension, allow trial and error in the process, make each trial cheap and easy, so designers can judge the effect with as little pixel manipulation as possible.

Formative study with novices making visual blends

Novices often make visual aids for posters, social media, or presentations. However, they typically don't know how to use Photoshop, so they often use presentation software to do image editing. We watched 13 novices create visual blends using Google Presentations and Preview.

Some operations were useful and intuitive to them such as move, resize, rotate, re-order images, crop, adjust transparency, and search for images within a sidebar. A few people knew that you can crop to a shape like a circle. Only one of the 13 participants knew that Preview has the magic wand tool, and it can be used to remove backgrounds and delete parts of an image. Half of them were able to achieve a prototype of a blend, but none of them were happy with the quality of the blend in the result.

They spent time and effort on low level operations like moving and cropping to get objects to fit an align. They also brought images front and back to edit them, then ordered them. This is a problem that Photoshop fixes with layers, but layers are also difficult to understand. Overall it was clear that novices have an intent they are trying to express, but could benefit from more powerful tools to help them execute their intent and spend less time on low level manipulation.

Design Principle 4. To novices, translating intent into action is a barrier to achieving their desired outcome. Learning new techniques or switching between multiple applications is a burden. To assist novices, build tools that have a more direct mapping between action and intent. This is where AI can help in assisting novices.

VISIFIT SYSTEM

To help novices iteratively improve visual blends, we created a system called VisiFit that leverages AI tools to help users easily extract and combine visual dimensions of each image into a blend. The user starts with a prototype of a blend from the VisiBlend system, and first improve the cropping of the main objects in the image, then improve the three visual 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 [1]. It builds on the Fabric.js canvas element to implement interactive image manipulation. Figure 4 shows the five steps of the interface in the order users see them. The input to the system is two images. These two objects must already be determined to have a shape match. We refer to them as Object A and Object B. In Object A, the shape covers is the entire object, in Object B, the shape only covers the main body of the object - it leaves parts of the object outside the shape.

There are two main steps: extracting the main shape of both objects, which will automatically generate a blend prototype. Next, the user must improve the prototype by selecting and adjusting options for the blends color, silhouette, and internal details. The steps of the system are as follows:

Step 1.1 Automatically crop Object A When the page loads the system first shows the A Object and the results of automatic

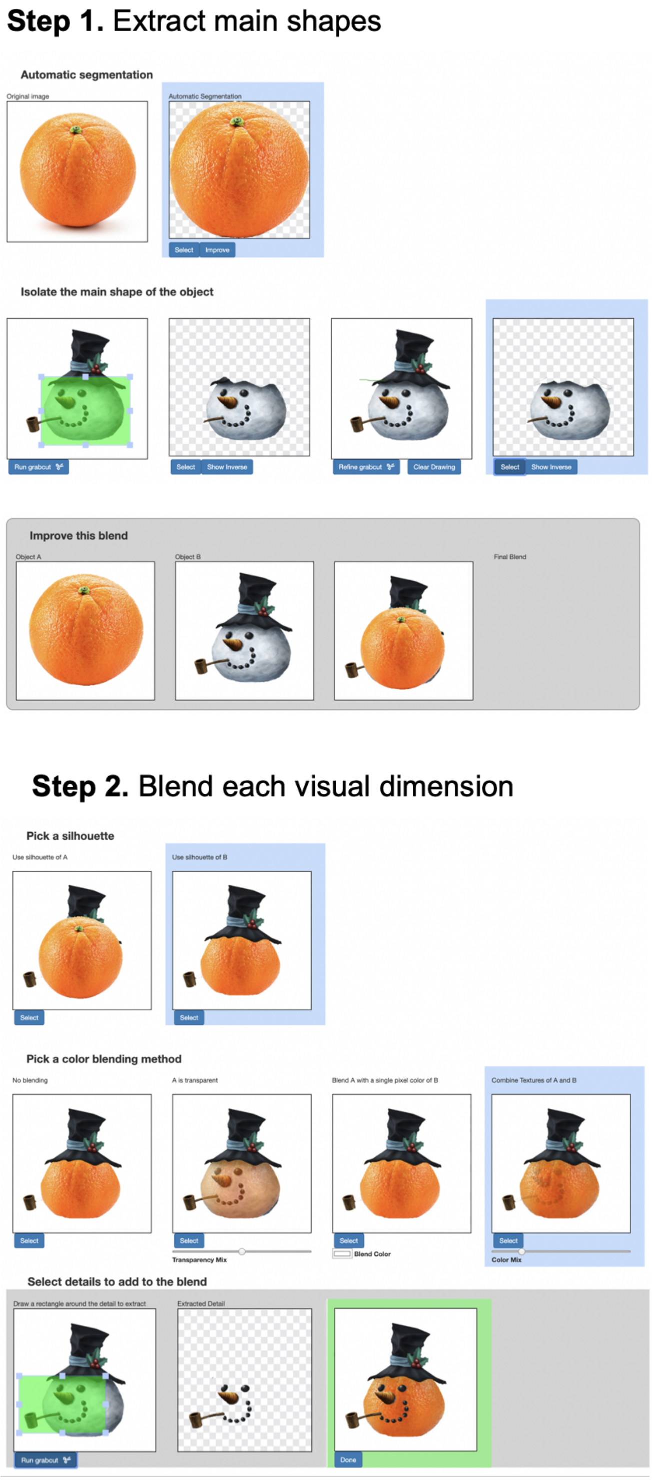


Figure 4. System steps

cropping. Object A is an image of a single object that we want removed from it's background. This is a classic computer vision problem of segmenting the salient objects in an image. Deep learning approaches are reported to be fast and accurate on test tests for this task. To leverage this automatic object extraction, we use the Tensorflow implementation of a pre-trained model for deeply supervised Salient object detection [13], and use the mask it provides to crop the images.

The user sees the output 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, and they will see an interface for Interactive Grabcut [24] that they can use to give indications of how to extract the object. Interactive Grabcut is explained more in the next section.

Step 1.2 Interactively Crop the main shape of Object B. The user sees object B and must interactively extract the main shape from the image. To do this, we use a Python implementation of Interactive Grabcut [24] - a traditional computer vision algorithm for foreground extract. Users first draw a rectangle that encloses the entire object to extract. Grabcut uses this to produce a foreground extraction. We show the result to users, then they can mark any extraneous pieces for removal by drawing on the image and running Grabcut again.

We used a classic interactive approach rather than trying fully automatic approaches because identifying parts or shapes within an image automatically is very difficult. Traditional automatic approaches like Hough Transforms [7] do not work well on most images. Deep learning approaches are fairly good at segmenting multiple objects from and image [10] but not yet at identifying the internal parts.

After both objects have had their main shape cropped, the system automatically produces a new prototype using simple affine transformations to move, scale, position, and rotate the objects to fit. Now they start improving the blend one visual dimension at a time

Step 2.1 Select a silhouette. When blending two objects, you can apply the silhouette of either object. 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 they think will make the better blend. This is the first iterative improvement to the blend.

To create the two silhouetted prototypes, the system uses the inverses of the cropped images from steps 1 and 2 and layers them on top of the current image to give an effect of having the silhouette of the object.

Step 2.3 Blend color and texture Color is the next visual dimension to include in the blend. Blends place one object on part of another, so users can decide if they want the color of one object, the other object, or a blend of both. There are many ways to blend color and texture. We present four automatic but adjustable tools tools for doing this:

- *Transparency*. We layer Object A onto 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 B image, and we blend image A with that color. The user can chose to blend the image with any color, including by selecting colors from the other images by using the eye dropper tool. This is especially useful for taking light-colored objects and giving them a tint of another color to signal a coherency to Object B.
- *Multiply colors*. Multiplying two images is a way to combine both their color and texture in a way that preserves both. Whereas transparency will always balance between the two, multiplication can get both the textures simultaneously. All three examples in Figure 3 use multiply to blend colors. It allows the Lego take on the red color, but have textures of both objects - the facets of the the diamond and the bumps on the Lego. The same effect works well on the Lego and the Popsicle example. It also combines the orange color with the shading of the snow man head in the third image.
- *Remove color*.If the colors of Object A are overwhelming, you may want to remove some of them to reveal the color the Object B beneath it, and bring those colors back into the body of the blend. Using the same K-means clustering in "Color Blend", we now detect the most commonly used color in Object A, and remove it with a default threshold of 0.2. The user can adjust this to remove more or less of the color. (not shown in Figure 4).

Step 2.3. Select details to add back to the blend. The last visual dimension to include is internal details and marking that help to identify the object. In the *snowman* and *orange* blend, the snowman is not as iconic without his facial details. Thus, we want to extract those from the original Object B and place them back on Object A. Again, we use Interactive Grabcut to allow the user to use a rectangle to select and refine what details to extract. We could have used other tools such context-aware select, but Grabcut worked well on our test set and it was a method users had already used above, so it was one less tool to learn. Both tools have strengths and weaknesses, and we should explore implementing both of them so users can apply whichever one works best for their image.

VisiFit encourages users to follow a linear workflow through each of the tools so that they can at least see the each of the effects on their image, even if they choose not to use it. However, users can take multiple paths, or explore both options. Additionally, the number of steps is not fixed. They can infinitely add edits to visual dimensions, if they choose to. However, the linear workflow allows them to have simple default path though all the visual dimensions they can iterate over.

At the end, the user selects the blend they are most satisfied with, and finish by seeing the initial prototype and their improved blend side by side to confirm that they like their improvement.

EVALUATION

Designing AI tools to assist novices is challenging because the AI has to perform well enough to be useful, and the interaction has to be simple enough for novices to master. We evaluate VisiFit by investigating the following research questions:

- Are the tools comprehensive enough to create high quality blends for a wide range of inputs?
- Do fully automatic AI tools work or do we need interactive techniques as a back up?
- To what degree does it elevate novices' ability to improve visual blends.
- What do professional designers think of VisiFit?

We developed these tools based on the analysis of 12 visual blends from the VisiBlends paper. We evaluated the tool based on its performance on 15 other visual blends mentioned in the VisiBlends paper. We refer to this as the test set.

Comprehensiveness of VisiFit

The VisiFit system uses a small set of tools and techniques to create blends. The first question is how comprehensive that set of tools is to improve blends of images. We asked our co-design graphic designers to judge the blends into three categories: 1) prototypes that do not need blending (VisiBlends is sufficient), blends that VisiFit can improve to a degree that they would publish it on social media, and 3) blends that still need sufficient improvement that they would not publish them. They were free to discuss and debate their judgements until they came to agreement. The blends were created by members of the team using VisiFit in under 5 minutes each. This group is expert at using the tool, the evaluation shows the upper bound of what the comprehensiveness of the tool.

Of the total set of 27 prototypes, 4 did not need blending (14.8%). 20 were deemed successful enough to post on social media (74.1%), and 3 were judged as needing improvement (11%). Of the blends in the test set, 2 of 15 did not need blends, 12 of 15 were good enough to print and only 1 needed improvement. Overall, this indicates that the design principles as implemented by VisiFit are capable of improving 86% (20/23) of prototypes into publishable blends.

Automatic Vs. Interactive AI tools

AI research promises high quality results on benchmarks and test sets, but it is unclear how well these fully automated approaches work in general, and to what degree we should still invest in interactive tools that make it easier for people to do the work. To evaluate this we focus on how often novice users selected the automatic object segmentation and how often they needed to interactively improve it.

We recruited 10 novice designers (7 female, avg age of 21.5) for a 1-hour long study. In the first half hour, they used the segmentation tools to extract the main objects from 12 images in the test set (we removed 2 images that did not need editing and one had images reused in other prototypes.) In the second half hour, they blended the segmented images. For each of the 12 tasks, they had 2 minutes to complete them. Their time was capped for three reasons. First, this tool is meant to aid rapid prototyping. The time limit also ensured users didn't waste hours on a task that was impossible to do with the tools.

Across all the sessions, the participants extracted 110 Object A's. Of those trials, only 43 of them (39%) accepted the fully automated Deep Salient Object Extraction result. Interactive Grabcut allow users to extract all but three of the remaining images (64 of 67 images). This indicates that fully automated approaches can be helpful, but an interactive back-up is necessary in case of failure. In the 3 cases of failure at least one most of the users were able to extract the object. The failures were either due to bad luck (Grabcut has stochastic elements to it and doesn't perform well every time), or user oversight.

Novice Ability with VisiFit

Although experts can use VisiFit to improve all the prototypes and can achieve publication-quality blends for 86% of the prototypes, the critical question is how well it enables novices to improve visual blends. Novices are new to the interactive tools and may not be able to use them as well, they are also new to the concepts and may not be able to apply them as well. In particular, VisiFit's design requires novices to iteratively improve on three visual dimensions. This decomposition of the task requires them to evaluate intermediate stages of the design. Novices are able to evaluate finished designs well enough using their gut instinct, but they may not be able to evaluate intermediate stages that focus on singular visual aspects of the image.

For the 11 novice designers in the study, improving the same 11 visual blend prototypes, we find that novices are able to improve the blends beyond what existing novice tools can do in 97.5% of the cases. Existing tools novices used in our formative study were only able to crop images, remove background, and perform a few color blending techniques like transparency and blending with a color. These operations are laborious for novices. In contrast, novices were limited to working on and improving an image in VisiFit for two minutes. The improvements made by novices used techniques extremely difficult to mimic in those tools such as silhouette, multiplying images, and extracting and applying details. Through a combination of novel tools and easy application of them, VisiFit was able to have a dramatic effect on novices's ability to do iterative improvement. Figure 5 shows examples of before and after blends for nine successful blend and three blends that need improvement.

No design is ever perfect. The best one can hope to do is satisfice [26] for the task at hand. For this task, we define satisficing as being judged by graphic artists to be good enough to publish on social media. For the judges, there were two major criteria for this:

1. *The objects must both be identifiable*. The definition of visual blends is that both objects are integrated and both are identifiable. If a blend does not have enough characteristics of one object to recognize it, it will not pass judgement.
2. *The objects must look blended.* It cannot be an obvious overlay of one image over another, or have transparency layers that expose parts of the layer underneath that clearly are not intended to be seen.

Overall, our designers judged them as successful with publishable quality in 65.3% of the cases. Of the problems evaluations found, 40% were due to not being identifiable, and 60.0% of the errors were due to not looking blended. The errors of not looking blended are rooted in users abilities to judge the final output and decide if it's a good blend. This could possibly be improved by more training, or from getting feedback from other users who can provide fresh critique from other novices, which has been implemented successfully with real-time crowdsourcing [20]. The other 40% of errors were mostly due to poor image quality because of the failures of Grabcut to extract details precisely in the given amount of time. If we built a better detail extraction tool (perhaps using the Magic Wand Tool), the success rate of users could go as high as 87.1%, which is competitive with experts.

There was one prototype that no novice or expert could improve to publishable quality using VisiFit: the *hamburger* and *lightbulb* blend. VisiFit does not have a tool to take the bottom bun color and change the lightbulb into the bun color. The system could easily be extended to allow color blending outside the Object A area. In PhotoShop, this is done with Context Aware Fill, but it could also be done with other fill or texture tools. This change does fit within the design principles that guide VisiFit - it only requires adding a new type of color blending to the list of options.

What do professional designers think of VisiFit?

Although VisiFit is meant to help novices, we co-designed it with 2 graphic artists who were eager to use it as a rapid prototyping tool to explore the space of blends very quickly. They found the visual dimensions natural and helpful to reason about their tools. We also tested the tool on one designer who has made visual blends professionally. He had never had any input to or knowledge of the tool before our session.

When using the extraction tools, he was impressed when the fully automatic tools worked, but disappointed in the oddly bad ways it failed (it consistently fails at removing white objects from white backgrounds). He was impressed with Interactive Grabcut both for extracting the whole shape and the main shape, but not for extracting details. For extracting details he would have preferred something that either worked better automatically or had more precision in its response to interactivity.

He was most impressed by the quick and easy way the blending tools helped him explore the design space. All of the basic operations were familiar to him, but he said it was such a relief to see a result so quickly. *"Sometimes I spend hours pixel pushing just to test an idea. I love being able to test an idea quickly."* He had two requests for more image blending options which can be achieved through several steps in Photoshop, but would be useful to test quickly such as only removing and blending the luminosity channel of an object. Earlier we experimented with implementing the Luminosity and Color blend tools in Photoshop, but we found that by themselves the didn't produce good results on the test sets. However, by combining them into one tool, he thinks it would be a useful blend technique for this task.

With VisiFit, he made blends that none of the novice users did. He liked to push the boundaries and try non-obvious features. He almost always started by looking at the inputs and formulating a plan. However, as the tool walk him through

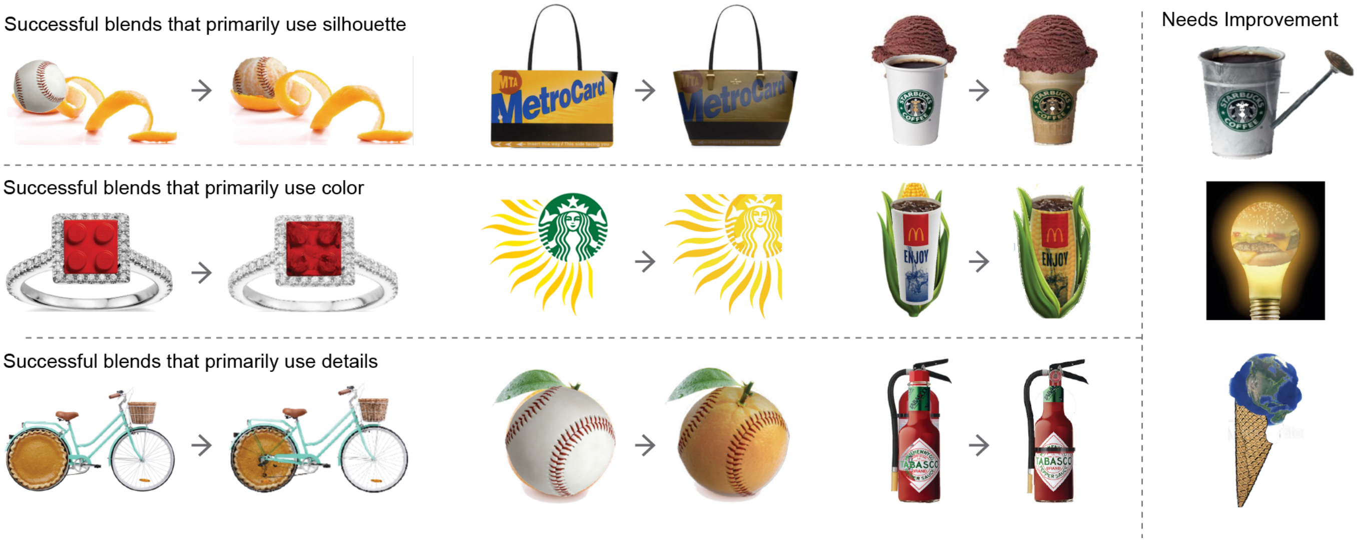

Successful blends that primarily use silhouette
Needs Improvement
Successful blends that primarily use color
Successful blends that primarily use details

Figure 5. Examples of improvements made to blends using VisiFit. It includes before and after images for 9 of the 27 cases and all 3 blends that still need improvement.

the workflow, he found some better ideas that surprised him. The flare and focus nature of the tool helped him explore the design space and keep multiple threads open at a time. From this interaction, we believe that VisiFit has value as a rapid prototyping tool even for professional graphic designers who work in visual blends.

DISCUSSION AND LIMITATIONS

AI Tools for Design

The key to making design tools for improving visual blends was to decompose the problem into visual dimensions and be able to iterate on them individually. Although VisiFit is highly specific to one creative task, we argue that many tasks can be decomposed along these lines for editing. Writing can be decomposed into style and substance. For example, a verbal argument has both its points and the convincing manner of saying it. Moreover, given a thesis statement like “gender equality is important to society”, there are multiple techniques for arguing it: appeal to authority, reducio ad absurdum. By separating thesis from execution technique, we can further support the interation in the process of writing and editing. In music, there are both chord progressions and melody. Both can be iterated on at different levels.

The idea for decomposing design into visual dimensions originally came from fashion, where designer strive to make novel garments, that are still relatable items that people want to wear [30]. To do so, they think about all the dimensions of a garment (color, fabric weight, texture, volume, print, silhouette, length, proportion, occasion, cultural associations) and innovate on some dimensions, but keep others familiar. However, not all combinations can be interchanged. There are sometimes dependencies between dimensions such as volume and fabric weight. Thin, flowy fabric cannot structurally support a large garment with volume or architecturally structured details. Such dependencies are also apparent in writing tasks. For example, it is sometimes impossible to change the style without also affecting the substance of the text at least a little. There are many research challenges in helping users balances these trade-offs.

Limitations

VisiFit is certainly not capable of improving all possible visual blends. The professional designs that co-design with us and give us feedback have listed some additional ways of blending silhouette, color, and details. Beyond improving the quality of the blend is the challenge of improving how well the ultimate message is conveyed. Some messages have a positive tone, like buying your kids *Legos* to keep them engaged during *summer* vacation. Hopefully the symbols, like a Popsicle, help convey this, but it can also reflect in the color, details, shading, textures chosen - particularly for which color Lego you pick and what color background you select. We have only addressed conveying the message through the symbols in the object, there is another challenge of conveying the tone of the message using the visual style.

Another dimension to add to the system is the ability to search for multiple possible versions and colors of the images and to see if different variations of the object make the blend better. While iterating on color, silhouette, and details, users may get ideas for slightly different images to use as the starting point. Our professional designer says he has a hacky way of doing this in Photoshop by importing all the various images and toggling their view off, but he would love to be able to directly connect it to image search and see and direct many mock ups. There may also be value in using parts from multiple images in order to make the best blend. Maybe we want the texture of one object, the color from another, and the shape of a third.

Now that we can quickly produce blends that are of publishable quality, a next challenge is to animate them to better delight audiences, draw attention, and convey the message.

The Lego diamond ring could sparkle like a diamond, the pumpkin-pie bike could either speed away, or people could start cutting and eating its pie-tires. There are endless opportunities to add simple motion related to the objects that will enhance the meaning to viewers. AI design tools could hopefully support the process of novices pulling animations from existing videos and apply them to their blends.

CONCLUSION

Iterative improvement is essential to the design process. However, iterative improvement requires difficult decisions about what to iterate on and requires the time and expense of making multiple prototypes. With the current advances in AI, there is the potential that AI can reduce these expenses and augment peoples' ability to design. However, we find that fully automatic AI tools are not yet able to produce high quality images that blend two images yet preserve their meaning and recognizability. Through co-design session with graphic artists, analysis of professional blends, and formative studies of novices making blends, we derived four design principles for interactive AI tools to support the iterative design process. The most important of these was to break up the problem such that users can improve each visual dimension of the image: color, silhouette, and details. Iterating on each of these dimensions is supported by a set of AI techniques which are known to be reliable for that subtask.

Our evaluation shows that novices can improve blends beyond what existing novice tools can do in 97.5% of the cases and they produce publishable quality blends in 65% of the test cases. With simple improvements to the tool, we can easily increase this number to 87%. We also find that professional designers find the tool useful for rapid prototyping of visual blends. Its easy and direct mapping of intent to action allows them to try more options than they would if they had to manipulate each image at the pixel level. Based on these results, we discuss the potential for AI to assist in other design subtasks by breaking down these problems into their core dimensions allowing people to use these tools to put together the pieces into novel and useful forms.

REFERENCES

[1] 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. (2015). http://tensorflow.org/ Software available from tensorflow.org.

[2] Barry W. Boehm. 1988. A Spiral Model of Software Development and Enhancement. *Computer* 21, 5 (May 1988), 61–72. DOI:http://dx.doi.org/10.1109/2.59

[3] 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 (CDVE'09)*. Springer-Verlag, Berlin, Heidelberg, 197–207. http://dl.acm.org/citation.cfm?id=1812983.1813012

[4] 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 (UIST '17)*. ACM, New York, NY, USA, 57–69. DOI:http://dx.doi.org/10.1145/3126594.3126653

[5] 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 (CHI '19)*. ACM, New York, NY, USA, Article 172, 14 pages. DOI:http://dx.doi.org/10.1145/3290605.3300402

[6] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. 2010. Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-efficacy. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 18 (Dec. 2010), 24 pages. DOI: http://dx.doi.org/10.1145/1879831.1879836

[7] 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. DOI: http://dx.doi.org/10.1145/361237.361242

[8] 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 (CHI '19)*. ACM, New York, NY, USA, Article 389, 18 pages. DOI:http://dx.doi.org/10.1145/3290605.3300619

[9] 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. DOI: http://dx.doi.org/10.1016/j.artint.2010.05.005

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

[11] 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 (UIST '06)*. ACM, New York, NY, USA, 299–308. DOI: http://dx.doi.org/10.1145/1166253.1166300

[12] 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. DOI: http://dx.doi.org/10.1038/s41746-019-0131-z

[13] 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.

[14] Joel. Ganbreeder. https://ganbreeder.app/. (????). Accessed: 2019-09-18.

[15] 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*.

[16] 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 (CHI '13)*. ACM, New York, NY, USA, 3083–3092. DOI: http://dx.doi.org/10.1145/2470654.2466420

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

[18] 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 (CHI '00)*. ACM, New York, NY, USA, 510–517. DOI: http://dx.doi.org/10.1145/332040.332486

[19] 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 (CHI '16)*. ACM, New York, NY, USA, 4142–4153. DOI: http://dx.doi.org/10.1145/2858036.2858425

[20] 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 &#38; Social Computing (CSCW Companion '14)*. ACM, New York, NY, USA, 21–24. DOI: http://dx.doi.org/10.1145/2556420.2556788

[21] 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. DOI: http://dx.doi.org/10.1145/258734.258887

[22] Yuval Nirkin, Iacopo Masi, Anh Tu an Trãn, Tal Hassner, and Gérard Medioni. 2017. On Face Segmentation, Face Swapping, and Face Perception. *arXiv preprint arXiv:1704.06729* (April 2017).

[23] Donald A. Norman. 2002. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA.

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

[25] 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 (UIST '16)*. ACM, New York, NY, USA, 609–624. DOI: http://dx.doi.org/10.1145/2984511.2984578

[26] Herbert A. Simon. 1956. Rational choice and the structure of the environment. *Psychological Review* 63, 2 (March 1956), 129–138. DOI: http://dx.doi.org/10.1037/h0042769

[27] 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 (FDG '10)*. ACM, New York, NY, USA, 209–216. DOI: http://dx.doi.org/10.1145/1822348.1822376

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

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

[30] Simon Travers-Spencer. 2008. *The Fashion Designer's Directory of Shape and Style: Over 500 Mix-and-Match Elements for Creative Clothing Design*. B.E.S. Publishing, Los Angeles, CA. 144 pages.

[31] 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 &#38; Social Computing (CSCW '14)*. ACM, New York, NY, USA, 1433–1444. DOI: http://dx.doi.org/10.1145/2531602.2531604

[32] 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 (CHI '11)*. ACM, New York, NY, USA, 1393–1402. DOI: http://dx.doi.org/10.1145/1978942.1979147