# Soft Shadow Removal and Image Evaluation Methods

*Maciej Gryka*

**Doctoral Thesis**

Department of Computer Science

University College London

December 22, 2015

*I, Maciej Gryka, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.*

# Abstract

High-level image manipulation techniques are in increasing demand as they allow users to intuitively edit photographs to achieve desired effects quickly. As opposed to low-level manipulations, which provide complete freedom, but also require specialized skills and significant effort, high-level editing operations, such as removing objects (inpainting), relighting and material editing, need to respect semantic constraints. As such they shift the burden from the user to the algorithm to only allow a subset of modifications that make sense in a given scenario.

Shadow removal is one such high-level objective: it is easy for users to understand and specify, but difficult to accomplish realistically due to the complexity of effects that contribute to the final image. Further, shadows are critical to scene understanding and play a crucial role in making images look realistic. We propose a machine learning-based algorithm that works well with soft shadows, that is shadows with wide penumbrae, outperforming previous techniques both in performance and ease of use.

We observe that evaluation of such a technique is a difficult problem in itself and one that is often not considered throughly in the computer graphics and vision communities, even when perceptual validity is the goal. To tackle this, we propose a set of standardized procedures for image evaluation as well as an authoring system for creation of image evaluation user studies. In addition to making it possible for researchers, as well as the industry, to rigorously evaluate their image manipulation techniques on large numbers of participants, we incorporate best practices from the human-computer intraction (HCI) and psychophysics communities and provide analysis tools to explore the results in depth.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

***Hypothesis:*** *Perceptually superior soft shadow removal is made possible by learning the relationship between shadowed image patches and their shadow mattes from synthetically-generated data. Simultaneously, the procedures commonly used by the computer graphics community to evaluate their algorithms can be codified and streamlined into a user-study-authoring system.*

## 1.1 Motivation

Digital image manipulation is becoming an increasingly important task in the age of ubiquitous camera sensors and vast sharing channels: *e.g.* in 2012 over 300 million photos per day were uploaded to Facebook alone[1]. Because of the sheer variety of contexts in which it is needed, there exists a great diversity of approaches to image editing with different levels of difficulty and flexibility afforded to the users.

Image manipulation techniques can be divided into two categories based on who their target is: expert or non-expert users. Non-expert operations include changing contrast and brightness, cropping and automatic red-eye correction. These are usually easily understood and straightforward to compute, but only offer limited possibilities. On the other hand, expert techniques by definition require experience and a deeper understanding of the field, while providing complete freedom. They usually rely on the combination of many atomic operations applied in the right amount to the correct parts of the image, combined with clear artistic vision of the desired result. An example of such an operation might be transforming a virtually shadow-free image taken under a cloudy sky into a more dramatic scene with a strong directional light and striking shadows, such as a sunset. Such change requires carefully adjusting the brightness of different parts of the image by different amounts, removing and adding realistic-looking shadows manually and changing the colour balance.

One possible solution to this disparity between capabilities of novices and experts is to build in some form of intelligence (or simply clever constraints) into the algorithm thus freeing the users from certain responsibilities. Ideally the only task asked from the user would be the specification of what should be changed, but without specifying how exactly. As an example, image inpainting algorithms ask for input to determine which part of the image should be replaced or filled-in and proceed to automatically determine what the result should look like. Similarly, in the case of soft shadow removal, which we

[1] https://www.sec.gov/Archives/edgar/data/1326801/000119312512235588/d287954ds1a.htm

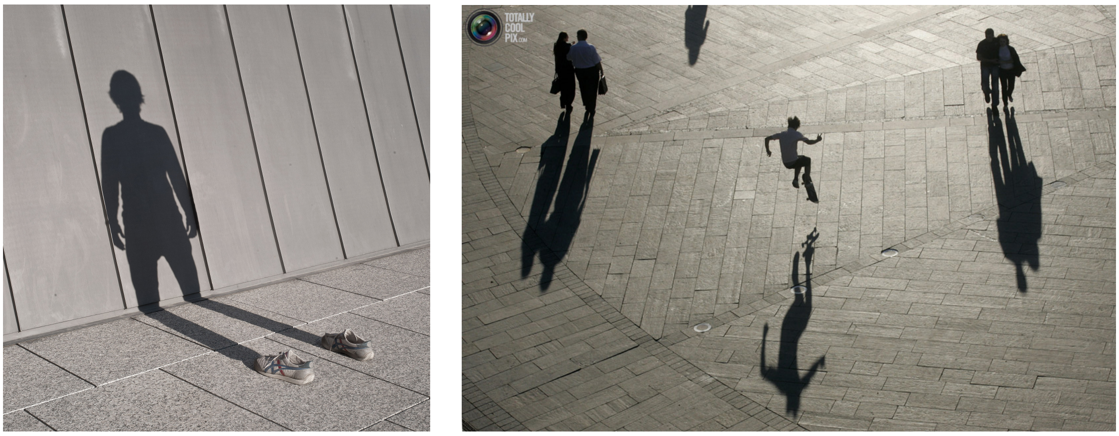


**Figure 1.1:** *Examples of images, where shadows play central roles. Image credits Pol Úbeda Hervàs (*`https://secure.flickr.com/photos/polubeda`*) and* `http://totallycoolpix.com/`*.*

focus on in Chapter 3, the user should only have to indicate which shadow to remove without thinking about how the result should be created.

Generally, these operations can be described as semantic, because their aim is to perform an image manipulation that has a semantic meaning in the real world *e.g*. "remove an object" or "change a light source". This contrasts with low-level operations that usually can only be interpreted in image space. Unfortunately, correctly performing semantically meaningful operations is often very challenging because various properties of the captured scene, such as the 3D scene structure and light characteristics, are not preserved in the image, but have to be accounted for.

Creating intelligent image manipulation methods is challenging on several levels. Firstly, creating the algorithm itself takes considerable time, effort, and creativity. Further, the problem of evaluating such techniques has to be considered as well, otherwise no principled success criteria can be established. Given the aim of plausibility, rather than physical accuracy, that many algorithms strive for, perceptual evaluation is the only reliable way to determine the degree of success. Thus, the ability to run user studies in a principled, rigorous and reliable manner is critical for the computer graphics and vision communities. This need is amplified when advances in established fields are made and comparisons to previous attempts are necessary to prove the merit and scope of a given method. Unfortunately, designing and executing a user study is difficult. While many recent graphics papers include user studies, the community is aware of the challenges involved in conducting them correctly and not satisfied with the status quo (see Section 4.1). In other words, while a few examples of high-quality evaluations exist, progress in our field is hampered by poorly conducted and irreproducible perceptual experiments.

Aiming to remedy these problems, we have built a user study authoring system specifically tailored to the needs of computer graphics researchers. Our prototype is now an online tool (`http://www.imcompadre.com`), allowing scientists to quickly create a range of principled user studies. Additionally, since published experiments should be trivially repeatable , it facilitates greater reproducibility than any existing alternative.

We have proven the utility of our tool by reproducing four published SIGGRAPH and Transactions on Graphics papers, and in each case, rapidly iterating to obtain novel, previously unreported insights into the data.

## 1.2 Overview

In the first part of this work, we focus on soft shadow removal, which is a semantic operation that can be an aim in itself, but can also be seen as a prerequisite for other operations, such as relighting. It is important to note here that we do not aim for physically accurate representation (what the scene would look like if the light really was in a different position) since this problem is massively under-constrained. Rather, we hope to present a solution that looks plausible to the human eye and does not contain jarring inconsistencies.

So far, shadow removal research either focused on a) hard shadows, where the penumbra was so narrow as to be negligible, or b) shadows that were only moderately soft and could be compensated for using simple approximations, such as linear or polynomial functions (see **Chapter 2** for in-depth discussion). However, as we explain in Section 3.5, there is a large class of images where these assumptions do not hold and where such techniques fail. In contrast, in **Chapter 3** we propose a novel, data-driven soft shadow removal aimed at wide-penumbra situations.

Having developed a novel image manipulation method, we turn to the problem of evaluating it thoroughly. In **Chapter 4** we show the current practices of the graphics and vision communities when it comes to evaluation of image manipulation methods and show how the situation could be improved. We identify two different tasks that can be used to evaluate a wide array of image manipulation methods and describe them throughly to indicate how and where it is appropriate to use them. In Section 4.7 we present an integrated system that makes it easy to create and run user studies for evaluation of image manipulation methods. Our hope is that this will allow researchers, whose focus does not lie in the area of user studies, to conduct principled, accurate and reproducible experiments in order to evaluate and characterize their methods.

Finally, in **Chapter 5** we reproduce and expand the experiments from three past, high-profile papers as well as our own work. In each case, we recreate the original experiment and find new insights into the methods by conducting even more in-depth investigation. Having reproducible user studies, where the subsequent re-runs are cheap to perform, besides being useful for evaluation of new methods, opens up new research problems. For instance a quick and reliable way to perform user studies can allow iterative refinement integrated into image manipulation algorithms.

After concluding our scientific contributions in **Chapter 6**, we explore the potential for using the new technology in an industrial context in **Chapter 7**.

### 1.2.1 Publications

Substantial portions of Chapter 3 appear in: Gryka, M., Terry M., Brostow, G.J., *Learning to Remove Soft Shadows*, Transactions on Graphics, Transactions on Graphics, 2015

# Chapter 2

# Related Work

In this section we present an overview of literature from the areas closest to the contributions we present in later chapters. We begin by exploring the area of shadow synthesis and editing, borrowing from related problems such as intrinsic images, matting and inpainting. Subsequently, we introduce the problems connected with perceptual evaluation of visual stimuli. This information informs our approach to verifying the success of our shadow removal method. It also allows us to later contribute to the current state of user study creation in the computer graphics community, by taking into account recommendations about how to perform general perceptual evaluations and effectively use microtask markets.

## 2.1 Shadow Removal

Most of the previous shadow removal work has focused on hard or nearly hard shadows. In contrast, in this work, we focus on soft shadows, mostly ignoring the specific case of hard shadows. Nevertheless, we present a review of the literature spanning the entire area as concepts introduced are useful for the discussion of the method in later sections.

Finlayson *et al.* [FDL09] proposed a method of detecting shadows by recovering a 1-dimensional illumination invariant image by entropy minimization. Given this, they were able to discriminate between shadow and non-shadow edges in the original image and subsequently perform gradient domain operations for unshadowing. The process forced image derivatives at shadow edges to 0, which worked well with hard shadows, but produced wide, noticeable bands of missing texture when applied to wide penumbrae.

Shor and Lischinski [SL08] tackled shadow detection and introduced a removal scheme using image pyramid-based processing. They dealt with non-trivial umbras by compensating for the occluder obstructing ambient light. Their method was not, however, able to deal with complex shadow intensity surfaces such as leaves or shadows without any umbra. The estimation of non-uniform inner shadow surfaces was only done at the coarsest pyramid level, and so only took into account large-scale variations. Further, it is not clear how to compute the "strips" used for parameter estimation in the case of complex shadows. The approach we present below is more generic, treating the entire shadow as potentially varying, and not limiting the variations to the coarsest scale. Further, their method was not well equipped to entirely deal with penumbrae, and inpainting around the shadow edges was still necessary

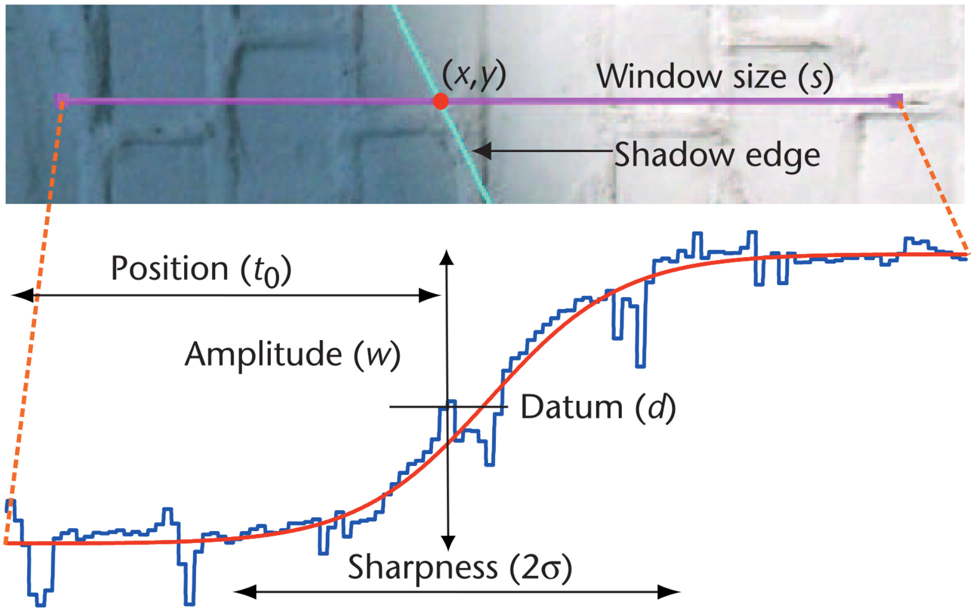


**Figure 2.1:** *Illustration from [MTC07]. The horizontal magenta line corresponds to the intensity graph (blue) through the penumbra region. The red curve is a least-squares fit of the specified piecewise quadratic function. Window size $s$, position $t_0$, amplitude $w$ and sharpness $\sigma$ are initialized by the user.*

to avoid artifacts.

Mohan *et al*. [MTC07] proposed a method for removing as well as modifying soft shadows. They modeled the penumbra by fitting a piecewise quadratic model to the image intensity in user-marked areas, therefore separating texture variations from illumination changes. This enabled them to work in the gradient domain and reintegrate the image after recognizing and removing gradients caused by shadows. The system first asked the user for input in the form of a shadow outline specified by control points along the shadow boundary. Additionally, the user was required to initialize the width of the penumbra as well as the shadow amplitude for each of the colour channels separately. The algorithm then performed iterative optimization by fitting the assumed penumbra fall-off model (shown in Figure 2.1) to either vertical or horizontal intensity slices through the penumbra, updating the parameters and optimizing again. This procedure was repeated for each segment of the shadow boundary separately (the number of boundary segments was also user-specified) and values between the boundary points were obtained by linear interpolation. The method produced convincing results, but was labor- and time-intensive for the user and required a significant amount of computation time. In our tests it took over 40 minutes per image, of which 10 were spent providing the input. Further, the algorithm was sensitive to user input and some situations required special care. For instance in places where the penumbra changed rapidly (such as corners) it was necessary to place 2 boundary points very close to each other so that the linear interpolation of parameters did not introduce artifacts. It was also difficult to correctly mark complicated shadows as they often require many boundary points.

After the penumbra parameters were optimized, the user had control over which gradients to remove from the image. Due to the nature of gradient domain operations, this method often modified the entire image noticeably, rather than just removing the shadow. Finally, this technique operated under two assumptions that did not always hold: that penumbrae could be modeled accurately using a sigmoid-shaped curve and that an umbra region existed at all.

Wu *et al.* [WTBS07] presented a matting approach to natural shadow removal. In contrast to standard matting methods, however, they treated the shadow matte as a pixel-wise fractional multiplier of the unshadowed image. While their method worked well on many shadows, it required noticeably more user input than our technique: a quad map signifying the "definitely in shadow", "penumbra", "definitely out of shadow" as well as "excluded" regions. Additionally, their matting formulation required a distance function to be optimized. While they presented one that performed well on many natural shadows, problems could occur in some scenarios (such as significant noise) since the matting cost function was not tuned for these.

Arbel and Hel-Or [AHO11] presented a critical survey of recent shadow removal literature and argued that matting approaches, such as that of Wu *et al.* described above, are not an optimal way to pose shadow removal. Instead, they decided to fit an intensity plane to the shadow-free surface and thus obtain an approximate unshadowed result and separate out the shadow. To recover the lost texture in the penumbra after fitting the intensity surface, they performed directional smoothing on the shadow matte in the direction perpendicular to the shadow edge. They demonstrated results on penumbrae up to 15 pixels wide.

Another method to detect as well as remove shadows was described by Guo *et al.* [GDH12]. The whole detection was region-based and performed by running an SVM classifier followed by GraphCuts on the regions of the image to decide whether they were in shadow or not, based on their appearance and relationship to others. Once every pixel in the image was classified as either shadowed or shadow-free, constraints for the matting step were built by skeletonizing the obtained shadow mask. Next, the matting method by Levin *et al.* [LLW08] was used to obtain penumbra reconstruction.

As noted previously, matting-based approaches are problematic for shadow removal in that they use a heuristic affinity function at the core of their energy minimization. Since engineering a shadow-specific affinity function might be challenging, our method effectively learns it from the data. Another problem, as we found in our evaluation (please see the supplementary material for examples), is that the method by Guo *et al.* is not well suited for user input since it can be difficult to specify which shadows should be removed. In the cases of wider penumbrae, the matting often "misses" the subtle gradients and does not remove the shadow at all, even with a user-provided shadow mask. While this problem could potentially be addressed by changing how the shadow mask is used to build matting constraints, the authors reported experimenting with a few (*e.g.* treating the eroded mask as definitely-in-shadow region) and choosing the most successful one.

Additionally all of the above methods suffered from the assumptions about the umbra region. Namely it was usually assumed that the attenuation due to shadow was constant in the umbra and that the correct reconstruction could be obtained by multiplying it by a constant. The first problem was that real shadows rarely exhibit characteristics of true ambient illumination in the umbra, which means that their attenuation varies. Secondly, simply multiplying the pixel values by a constant might introduce artifacts such as decreased contrast and amplified noise. Both [SL08] and [LG08] dealt with these issues by applying a method similar to previous colour transfer work [RAGS01] to recover the umbra correctly by analyzing the areas in and out of shadow and matching their characteristics. They differed, however, in how they dealt with the penumbra. [SL08] used inpainting, which had the drawback of potentially loosing valuable information, and was only applicable to limited penumbra widths that could be successfully inpainted. On the other hand [LG08] used approach similar to [MTC07] of fitting a parametric curve (in this case cubic polynomial) to the penumbra with a more robust user input mechanism. While this method produced good results for some examples, it is unclear how well it could cope with wide and spatially varying penumbrae because of its fixed parametric model.

### 2.1.1 Soft Shadow Synthesis

Computer graphics literature presents various ways of rendering soft shadows and can therefore provide valuable insights into how penumbrae can be modeled. While today it is feasible to create physically-based lights and shadows using sampling methods, it used to be a very expensive operation which led the researchers to derive certain approximations. [PSS98] observed that a diffuse spherical light and a straight occluding edge produce penumbrae, which fall off according to a sinusoidal function:

$$s(\tau) = \frac{1}{2}(1 + sin(\pi\tau - \frac{\pi}{2})), \tag{2.1}$$

which can be approximated well with the Bernstein interpolant:

$$s_B(\tau) = 3\tau^2 - 2\tau^3. \tag{2.2}$$

Other works [Hai01, CD03] drew on this and, while methods of rendering shadows based on the above improved, the fall-off model remained the same. [MTC07] showed that a different model is more appropriate for real penumbrae, while [AHO11] argued that fitting parametric models generally is not the correct approach in the case of real shadows. In this work we posit the same, and provide some evidence for it in Section 3.2.

### 2.1.2 Intrinsic Images

Intrinsic image algorithms, as defined by [BT78], separate images into the intrinsic components of reflectance and shading. This information can be used to aid other image manipulation techniques, scene understanding, *etc*. While much progress has been made in this space, many open problems remain. The work reviewed here describes approximate solutions that provide good results in specific cases. However, in general, this class of algorithms is not well equipped for dealing with cast shadows as we show in Section 3.6.2.

One approach to solving such under-constrained problems is to incorporate higher-level reasoning or additional data into the pipeline. For instance, [SA93] showed how to differentiate reflectance from illumination discontinuities in the world of painted polyhedra, which improved on previous approaches based on the Retinex theory [LM71]. Work by [LBD13] used depth data from multiview stereo to accurately recover illumination of a scene, while [Wei01] used multiple images of the same object under varying lighting conditions and a prior based on statistics of natural images to obtain convincing reflectance and shading separation.

Similarly in [BPK13], multiple images of the same scene with different illuminations were used to enable rich image relighting operations. Interestingly, this technique allowed softening of lights by blending multiple images, while our approach performs image-space operations for control over sharpness. [BPD09] got their additional data from the user, who was asked to mark areas of constant reflectance and constant shading.

[TFA05] leveraged machine learning by first classifying each gradient in the image as either shading or reflectance, and then employing Generalized Belief Propagation to extend areas of high confidence to more ambiguous regions. Our approach is related in that we also use supervised learning followed by a regularization of a Markov Random Field. What makes our solution unique is a heavily customized learning algorithm and the ability to deal with hundreds of labels at each site.

Recently, [BM12] used a set of priors over reflectance and illumination combined with a novel multi-scale optimization to obtain results on the MIT Intrinsic Images dataset [GJAF09], outperforming other methods by 60%, while also recovering shape and illumination. While this method works very well on images of single objects, we found in our experiments that its results are not as reliable when faced with complex scenes and cast shadows.

### 2.1.3 Matting

Image matting provides another type of decomposition, and can be used to separate foreground and background objects. In principle, this formulation could be used to separate soft shadows as [GDH12] do when using a method by [LLW08] to matte out small penumbra regions. [WAC07] presented an intuitive brush interface combined with a fast algorithm to interactively matte out fuzzy objects. As noted previously, the challenge with using these techniques on noticeably soft shadows lies in specifying the correct affinity function to optimize. Our method effectively learns such a shadow-specific function from the data. Additionally, our users specify only a coarse binary mask, rather than a trimap.

While [CGC$^+$03] presented an effective method for shadow matting and compositing, they required much more input and did not tackle the challenge of wide penumbrae.

### 2.1.4 Inpainting

Inpainting refers to a family of techniques capable of filling in missing image regions. The field has matured in recent years to the point of implementations being available in commercial tools. While useful in many cases, it is not a perfect solution to the problem of shadow removal as it completely discards potentially valuable information. Consequently, it often fails to reconstruct structure (see Figure 3.4). It does, however, often produce visually convincing results, and we exploit it to obtain a rough initial guess to guide our algorithm.

One of the early inpainting works [MM98] drew on [NMS93], but proposed a solution that applied to natural images. It performed inpainting by connecting geodesic curves (*i.e.* of the same gray level) that arrived at the occlusion region by straight line segments. One of the shortcomings was that it did not consider at what angle the curve arrived at the region, often resulting in an unnatural-looking fill.

[BSCB00] coined the term “inpainting” and provided a solution conceptually similar to [MM98], but slightly more robust. It was motivated by how art conservators repaired damaged paintings and as such also tried to continue lines “coming into” the inpainting region. After the user marked unwanted areas in the image, the algorithm iteratively filled them in taking into account the angle at which curves entered the hole. The direction was found by computing the dominant gradient and continuing in the direction perpendicular to it. Each iteration filled in one, outmost layer of the missing region, continuing inwards. Best results were achieved on thin regions such as text and creases in images and for structured inpainting regions. On the other hand perceptually noticeable errors were produced while filling in larger areas, since texture was not propagated.

The most significant limitation of [BSCB00] was working with structure alone and resulting inability to deal with textures, which was addressed in [BVSO03]. The idea was to perform a decomposition on the target image to produce structure and texture components and fill in missing regions in two resulting images using different techniques. This was motivated by the observation that there is no good existing technique that deals with the whole spectrum (structure and texture simultaneously), while specific solutions work well under some constraints. First, image decomposition of [VO03] was used to separate structure and texture components. Afterwards, the authors used inpainting work of [BSCB00] to fill in structure and texture synthesis of [EL99] for recovering texture. Finally, the two reconstructed images were added together to produce the output. Presented results showed improvement over both structure-only and texture-only methods.

In contrast to previous techniques that completed the missing region solely based on the boundary around it [MM98, BSCB00, BVSO03] the solution of [LZW03] tried to first learn global statistics of the image based on the known areas. Then the inpainting was done by finding a solution which maximized the probability computed based on the surrounding region as well as the learned global characteristics. The maximization was completed using loopy belief propagation and linear programming. This approach worked well for simple missing regions, but, similarly to [BSCB00, MM98], would not be able to complete large, textured areas.

The idea of working with structure and texture together was gracefully extended by Criminisi *et al*. [CPT03], where the authors proposed a unified solution that preserved texture while also following isophotes (i.e. considering structure). This was an iterative, patch-based method based on exemplars (regions in the image that were known to be correct, *i.e.* not in the inpainting region). The procedure started by evaluating and assigning priority to each patch on the inpainting front (border of the fill-in region). Separate patches were created around every pixel on the inpainting front with a constant, user-specified size slightly larger than the biggest texture unit. At every iteration patch with the highest priority was filled in, where patch priority was defined as the product of confidence and data terms. While the confidence term effectively encouraged concentric fill order (layer-by-layer as in [BSCB00]), the data term tried to first continue isophotes (lines “coming into” the inpainting region). Results shown compare favorably to previous works.

[DCOY03], similarly to [CPT03], aimed to provide a way of simultaneous structure and texture inpainting by using patches (or "fragments" as defined in the paper). To perform inpainting, the system looked for data in the input image at various scales and orientations and, after successfully selecting a fragment, performed compositing with Laplacian pyramids to produce visually pleasing results. It worked on a coarse-to-fine basis, first approximating colours on the coarsest level and then using this result as an initialization for adding details at the finer level. Each fragment was considered at a dynamic, adaptive scale to capture detail at the right level.

Special consideration for structure was shown by [SYJS05] where the authors took advantage of the fact, that users are good at completing the structure in missing regions. Therefore they asked for user input in the form of line segments extending from the known area into the inpainting region. Then the system completed the structure along the specified lines using the information from the known region. Next the remaining areas were assumed to consist mostly of texture and were completed using a patch-based texture synthesis algorithm. Finally, photometric correction in gradient-domain was performed on the inpainted areas to reduce visual inconsistencies. In case of non-overlapping lines the structure propagation was presented as a graph labeling problem and authors performed energy minimization using a custom energy function. If there were junctions in the structure lines, the problem was solved by belief propagation.

Komodakis and Tziritas [KT06] defined a discrete global optimization scheme that was presented as a unifying approach to image inpainting and texture synthesis. Similarly to some earlier works [DCOY03, CPT03, WBTC05] the approach was patch-based in that it copied small patches from the known (source) region into the unknown (target) area. The energy function minimized here was a sum over the costs of placing patches next to the known boundary and next to each other. To perform the minimization authors proposed a new method, "Priority-BP" which is an extension of belief propagation (BP) with two new elements described in the paper: "priority-based message scheduling" and "dynamic label pruning". Because of the global nature of this solution, authors argued that they avoid local minima resulting from the earlier, greedy approaches.

Copying patches from either known areas of the same image or other images was also used by Wilczkowiak *et al.* [WBTC05]. Here, the assumption was made that it was acceptable to slightly extend the inpainting region, thus losing known data, in order to obtain more visually pleasing results. Patches were copied into the hole based on similarity of the neighbouring pixels and their shape was determined by finding an optimal graph labeling with graph cuts based on similarity to already known areas. The system worked automatically after the user selected region to fill in, but it was also possible to provide addition constraints. Additionally the authors used automatic image rectification to find matches regardless of perspective at which the image was taken. This produced high-quality results, especially for photographs of buildings and other man-made structures, which were targeted in this work.

The idea that the missing regions could be completed using information from many different photographs was taken to the extreme in [HE07]. By utilizing a database of millions of outdoor images (which were pre-processed before) the authors were able to create a system that completed desired areas with data from similar images. Therefore, content was not only visually pleasing, but in majority of cases semantically correct because images of similar scenes are likely to be semantically similar. In order to find images suitable for inpainting the target, gist scene descriptor [OT06] was employed. Even though it did not explicitly describe the semantics, given large enough set of images and the fact that the final selection was performed by the user, it succeeded in finding good source image candidates.

Generally, inpainting methods need a source of data that can be used to fill in the missing parts. They can be divided into two categories based on where this data is taken from: a) bootstrapping algorithms that use the remainder of the image to be modified such as [CPT03], [BSFG09], [PKVP09], and b) methods that rely on previously created datasets such as [HE07]. Algorithms in the former category are appealing since one does not need to worry about creating an extensive training set. Yet, in practice, it is often difficult to make them scale to general scenarios. See Section 3.8 for an analogous extension of our method. Both Criminisi *et al.* [CPT03] and Barnes *et al.* [BSFG09] filled in the missing regions by finding patches in the rest of the image that "fit into" the hole. In both cases, care had to be taken to propagate the structure correctly into the missing parts. While Criminisi *et al.* achieved this in an automatic way by searching for matches along isophote lines, Barnes *et al.* opted for user guidance to indicate structure lines crossing the hole, and thus manually constraining the search space. While not originally used for inpainting, an alternative here could be a robust approach for finding non-rigid correspondences as presented by HaCohen *et al.* [HSGL11].

## 2.2 User Studies and Evaluation of Images

In this section we review literature related to the issues of perceptual image similarity and methodology for conducting image-focused user studies. We also conduct a survey to determine how feasible is crowdsourcing as a perceptual evaluation tool. Additionally we mention several published works in the area of automatic image enhancement as well as other image manipulation methods, where user studies have been used as a measure of success and opportunity for further exploration.

### 2.2.1 Psychophysics

While our community mostly focuses on creating computer graphics and vision algorithms, some members also think about the human aspect of how to evaluate these methods. The area of psychophysics, which is concerned with finding how objective changes in the outside, physical world relate to our perceived, psychological experiences, is a ripe field to borrow ideas and experience from.

Over the recent years many computer graphics papers and journal articles included psychophysical experiments as evaluation (*e.g.* [WBSS04], [WFGS07], [ČWNA08]). Further, the need for more in-depth psychophysical knowledge was confirmed by a 2008 SIGGRAPH workshop "Psychophysics 101" [Fer08]. Based on this, and other resources such as [EE99], we summarize selected concepts most relevant to our research.

As already mentioned, psychophysical experiments usually aim to establish relationships between physical phenomena and their effects on human perception. A widely used example is estimating the brightness of a patch. Brightness, *e.g.* on a screen, can be measured directly and is known to the researcher. However, knowing the absolute brightness does not necessarily mean we know how the patch appears to people, both on its own and in relation to other patches. To find the relationship, several established methods can be used: method of adjustment (MOA), method of limits (MOL), method of constant stimuli (MCS). They all aim to find the mapping between the "physical delta", *i.e.* the objectively measured change in the world, and the "perceptual delta", *i.e.* how much change people perceived, often establishing the minimum absolute change that results in a detectable change in perception.

In contrast to threshold methods, it is also possible to arrange stimuli on a specific scale using scaling methods. Scaling methods include rating, where each stimulus is assigned a numerical value based on the specified question, *e.g.* "How bright is this patch?"; pair comparisons, where stimuli with different values on the objective, measurable scale are compared to each other; ranking where stimuli are ordered on a linear scale; category scaling, where stimuli are assigned to categories, *e.g.* "very dark", "very bright".

It is important to remember, however, that not all standard, psychophysical assumptions will hold in our research. Notably, while our community is very concerned about perceptual effects, there is rarely a continuous, objectively-measurable scale to relate them to. More often we are faced with a situation, where we would like to assign perceptual properties to stimuli from different categories, *e.g.* created by different algorithms. This does not directly translate to psychophysical problems, since different image processing methods do not imply change on one, measurable axis, but usually completely different characteristics like brightness, sharpness, colour balance *etc.* coupled together. In short, while psychophysics aims to evaluate human perception given data, we mostly aim to evaluate data given perception. Consequently we do not apply the aforementioned methods directly, but instead describe derivatives that have already been used in graphics research.

### 2.2.2 Image Evaluation & Perceptual Distance

While psychophysics deals with the general case of stimulating human senses, significant work was also done focusing on estimating visual distance specifically between photographs. For instance [RFS$^+$98] describes an experiment that was ran to establish which image features influence image similarity, as understood by study participants (overall colour characteristics, "human" vs. "non-human" and "man-made" vs. "natural" were some of the most important ones). Further, the authors used a new method of collecting similarity judgments from populations, trading off some accuracy for efficiency as compared to the standard psychophysical technique of paired comparisons.

Also based on low-level image features, Cox *et al.* [CMM$^+$00] proposed an image search system that leveraged Bayesian reasoning and psychophysical theory to enable very fast and accurate retrieval of images. To validate the system, detailed experiments were ran to prove the utility of the proposed solution. Subsequently, the system was used in [PCY$^+$01] to conduct a number of psychophysical experiments evaluating image similarity distance schemes.

An argument for how low-level image features can be used to accurately predict image similarity perceived by human observers was presented in [NG06]. Several image features, based on the knowledge of the early processing stages in the human visual system, were evaluated. The authors report seeing significant correlation between image indexes based on such features and image similarity.

### 2.2.3 Aesthetics and Predicting Image Attractiveness

One of the most obvious experiments one can conduct given a set of image data and a large population of Turkers is to sort the images by their attractiveness to human observers. Over the recent years a number of algorithms emerged aiming to predict how attractive a given image is automatically, with no human input. While there are many other tasks that humans can complete (annotation, comparisons on different scale than attractiveness *etc.*), progress on even this one front is difficult. Nevertheless, a brief examination of the field of visual aesthetics is useful in order to appreciate the difficulties and the value that psychophysical user studies still hold for the graphics community.

### 2.2.4 Crowdsourcing

Since the Amazon Mechanical Turk became available, many researchers across disciplines like psychology, psychophysics and computer graphics and vision started using it to recruit study participants. At the same time, a lot of work was dedicated to evaluating the differences between studies ran using online microtask markets and more traditional setups.

In this section we present an overview of these efforts and extract several recommendation about how to maximize the effectiveness of tools like Mechanical Turk when conducting experiments. Additionally, we shortly examine how similar tools can be used for different purposes.

#### Feasibility

One of the early analyses was presented by Kittur *et al.* in [KCS08]. They explored using Mechanical Turk to conduct tasks normally done with “low numbers, high-engagement” participants using the “high numbers, low engagement” setup that microtask markets offer and how to design the studies to adapt them to this new environment. In summary, they found that using microtask markets is most useful for examining problems which have definite answers. Attempts to game the system were detected, but thoughtful task design (*e.g.* requiring subjects to answer a number of non-trivial, but verifiable questions), made it less likely for participants to cheat. It was also found that the audience seemed diverse, which is positive from the researchers’ perspective, however, not always possible to measure directly.

The main recommendations given by the authors included ensuring that some questions are explicitly verifiable to enable detection of malicious input; even better when the questions also *look like they are going to be verified*. Additionally, the effort of providing malicious input should be as close as possible to the effort of producing real answers.

In a related effort, Paolacci *et al.* [PCI10] addressed widespread concerns about data quality obtained from Mechanical Turk, aiming to show that it is possible to use it to run subjective experiments and still obtain good results. Firstly, the authors showed that the population of Turkers is at least as representative of US population as groups recruited traditionally and that more and more international workers are appearing. Secondly, three classic psychological experiments were conducted and the results obtained from Mechanical Turk agreed with what was expected (*i.e.* the results of experiments and the magnitudes of effects were similar) showing that there does not necessarily need to be a difference between traditional and online study audiences.

### Effectiveness

Multiple factors influence the quality of results obtained from any user study. Recruiting microtask workers, while having significant advantages, introduces even more areas, where care has to be taken to avoid bias. Several methods and recommendations have been proposed by researchers in psychology and human-computer interaction fields. For instance, Oppenheimer *et al.* [OMD09] proposed "Instructional Manipulation Checks" (IMCs): a way of assessing the quality of answers that survey participants provide and therefore increasing statistical power of experiments. IMCs are instructions embedded in the survey questions, which look just like other questions, but ask the participant to perform some unusual task to confirm that they have read the instructions (*e.g.* click on a small element on the bottom of a website, rather than the more prominent Likert scale used in "normal" questions). The authors demonstrated, that IMCs have twofold benefits: firstly they allow screening out of unreliable data during analysis and therefore increase signal-to-noise ratio of the dataset. Secondly, they can be used interactively to force survey participants to read the instructions before proceeding, by making sure that the next question is displayed only after the instructions are followed correctly. Interestingly in the second case, from the perspective of the data, the previously negligent subjects "turned into" good subjects after the IMC failure was pointed out by the system.

Some approaches aimed to detect negligent behavior (cheating) by workers on Mechanical Turk. Rzeszotarski and Kittur [RK11] showed how to detect whether the study participants were diligently fulfilling tasks, or not, by looking at their task fingerprint, *i.e.* the record of all interactions with the study interface. This approach was especially valuable when "gold standard" data was not available, as in the case of subjective surveys, so rejecting bad workers based on answers provided was not possible.

Similarly, Oleson *et al.* [OSL+11] proposed a way of creating gold standard data semi-automatically. This ability is useful in situations, where no gold standard is available ahead of time, but might be generated given some initial data from users and a manual inspection from study authors.

Interestingly, Ipeirotis *et al.* [IPW10] proposed a fine-grained analysis method for estimating reliability (quality) scores of Mechanical Turk users while also providing the output (*i.e.* the class labels for a set of objects). As opposed to previous approaches that simply downgraded users' scores if they were inconsistent with majority or known ground-truth answers, the authors separated unrecoverable error rates (*i.e.* noise due to negligence or other factors) from bias, which could be corrected for.

This bias measure was subsequently used to adjust users' answers accordingly and made it possible to obtain reliable data from users, who would have otherwise been considered low-quality Turkers. Conversely it also allowed detection of "strategic" spammers, who always assigned the class label with highest prior probability.

Another important step towards increasing data quality, is setting up of correct incentives for the respondents. This topic was explored by Kapelner and Chandler [KC10a], who examined three different ways of preventing online survey respondents from "satisficing", that is taking mental shortcuts and being negligent when answering questions. The first method, displaying prominent text along with the questions asking the participants to take the survey seriously, did not have any measurable effect. However, forcing people to spend at least a specified amount of time per questions (by disabling the "Continue" button for the first 10 seconds) increased the quality of answers by 10%. The third, and most effective measure consisted of slowly fading words from the question and other elements of the interface. This method increased the quality by 13%. The main conclusion we can draw from this work, is that changing the cost/benefit ratio of satisficing and answering diligently, has a significant impact on the quality of gathered data.

Another aspect of maximizing the efficiency of using Mechanical Turk is ensuring that people choose to complete our tasks without requiring expensive compensation. One way of achieving this is understanding how workers behave, especially when searching for and evaluating tasks. Chilton *et al*. [CHMA10] presented such an an examination and provided insight into how to best formulate tasks so that they are answered quickly and with minimal cost. In summary, tasks behave similarly to search results other media: the tasks on the first page get by far the most interest and the tasks on the top of the first page, more still.

Further, being on the top of results when searched by categories (cheapest, lowest price, soonest expiration, shortest time, alphabetic ordering) is also beneficial. In fact, being at either extreme is good, since sorting can be both incremental and decreasing. On the other hand, being close to the median in the search results is the worst case scenario, since the workers have to click through many pages, whichever extreme they start from.

In addition to perceptual evaluation, microtask markets offer many other possibilities. For instance, a novel application was presented by Little *et al*. [LCGM09] where, instead of running HITs "in parallel", they ran them "in series". In other words, early HITs provided input to later HITs allowing for verification, iterative improvements and verification of earlier work.

### Competition

While there is currently no system, which completely fulfills the needs of graphics researchers, some solutions exist, which contain a subset of the capabilities needed.

Firstly, if Mechanical Turk is to be used, it is possible to build tasks using Amazon's interface. This approach has some advantages (for instance, anecdotally, Turkers are slightly more willing to perform tasks when they are not required to leave the Mechanical Turk domain), however, it is too constraining to address all the needs of researchers in our field. It is not possible to define new task types with full control over the interaction and captured information, and the tasks available are not geared towards specific purpose of perceptual image evaluation. Importantly, this "locks in" the researcher to using Mechanical Turk, which is not always desirable. For the same reasons, services built to make working with Mechanical Turk easier, such as [KC10b] and similar, are not able to provide complete solutions.

Other, more comprehensive products such as [Cro] cater more to industrial customers and have prohibitively high minimum transaction thresholds.

Recently Matera *et al*. [MJCB14] presented a prototype crowdsourcing tool similar in scope to our solution. However, they aim to fill a larger niche and do not focus exclusively on graphics researchers and image evaluations. Additionally, no plans were mentioned for developing image-specific evaluation methods such as we proposed in Section 4.7.

# Chapter 3

# Soft Shadow Removal and Editing

## 3.1 Introduction

Smart image editing algorithms are increasingly needed as non-experts demand more post-processing control over their photographs. Across the wide range of techniques available in commercial tools and research prototypes developed by the graphics community, soft shadow manipulation stands out as a severely under-explored, but important problem. Since shadows are a key cue for perceiving shape, curvature, and height [Ken74], countless Internet tutorials attempt to teach users how to separate out a shadow layer using tedious masking and thresholding operations. These manual techniques are employed in part because many real scenes have soft shadows, and most shadow detection and removal algorithms developed thus far address mainly hard shadows. Because soft shadows have been shown to be correlated with people's perception that an image is real [RLCW01], easier methods for extracting soft shadows are required.

We present a data-driven shadow removal method that is pre-trained offline and can deal with shadows of widely varying penumbra widths (*i.e.* where there is no clear boundary between the shadowed and unshadowed region). In contrast to previous work, our technique does not assume the existence of a specific model for the umbra, and processes the entire shadow with a unified framework while still giving users full control over which region to modify. We can deal with complex situations that were previously impossible, such as when the entire shadow is essentially penumbra, as is often the case (*e.g.* with shadows of leaves cast on the ground). Our technique requires user interaction only to roughly indicate which area of the image should be modified. The system then initializes and applies our model. Once the shadow matte is computed, the user can interactively manipulate it, or the rest of the image, using our simple interface.

Our regression model is trained through supervised learning to cope with our underconstrained problem: given a shadowed RGB image $I_s$, we aim to find a corresponding shadow matte $I_m$ and the unshadowed image $I_u$ that satisfy $I_s = I_u \circ I_m$ ($\circ$ is an element-wise product). Similar decompositions are explored in the intrinsic images domain [LM71], but we compute $I_m$ to ignore both reflectance and shading, and only focus on cast shadows. Also, rather than aiming for physical accuracy, our practical objective is to produce a convincing-looking $I_u$ as measured subjectively.

In a user study comprising hundreds of rankings and assessments, our technique was found to be

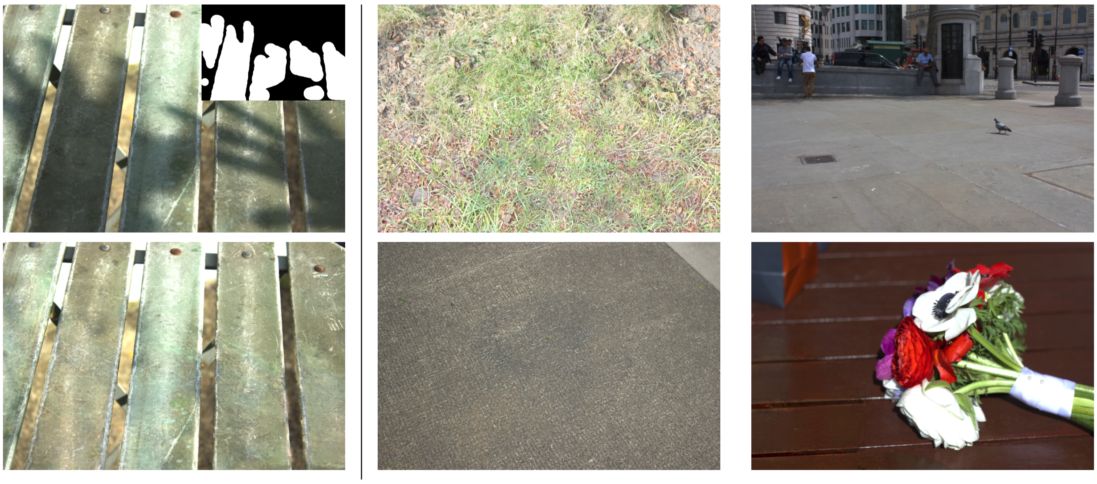


**Figure 3.1:** *The first column shows the input shadowed image (top) with a user-provided coarse shadow mask (inset) as well as the unshadowed image (below) produced by our method. The four images on the right present different unshadowed results for which the corresponding inputs can be seen in Figure 3.27. This technique could be used* e.g. *as a pre-processing step for texture extraction algorithms such as [LXDR13].*

significantly more likely to remove soft shadows successfully than the competing methods by [GDH12], [AHO11] and [MTC07]. Additionally, when shown together with results produced by these other methods, our results were most often chosen as the most natural-looking (please refer to Section 3.6.5 for more details).

Our specific contributions are:

1. A regression model that learns the relationship between shadowed image regions and their shadow mattes.

2. A system that leverages existing inpainting and adapts large-scale regularization to our field of matte patches, producing results that compare favorably with the real-world baseline.

3. A system for generating countless training examples of scenes with both soft and hard shadows.

4. A large-scale dataset of soft shadow test photographs.

5. We will make available the code for the method as well as the user study experiments for easy replication.

**System Overview**: To use our system, the user first paints the region of the image containing the shadow they wish to modify. This masked region is then processed automatically, as follows. First, the input image is divided into non-overlapping $16 \times 16$ patches, and for each patch $a_i$ a descriptive feature vector $f(a_i)$ is computed. Next, our pre-trained regressor maps each feature vector to $m_i$, a distribution of possible shadow mattes for that patch. A Markov Random Field on the grid of shadow matte patches is regularized to generate the maximum a posteriori shadow matte image $I_m$ for the red channel. A final optimization computes the corresponding shadow mattes for the green and blue channels, also yielding the unshadowed image $I_u$. With the shadow removed, our interface then allows the user to place a

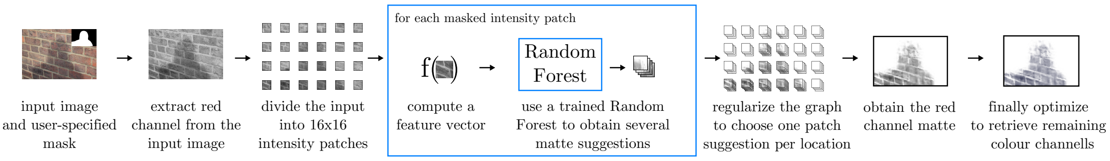

input image and user-specified mask
extract red channel from the input image
divide the input into 16x16 intensity patches
for each masked intensity patch
compute a feature vector
Random Forest
use a trained Random Forest to obtain several matte suggestions
regularize the graph to choose one patch suggestion per location
obtain the red channel matte
finally optimize to retrieve remaining colour channells

**Figure 3.2:** *System overview. Obtaining the final matte (far right) allows us to remove, as well as modify, soft shadows.*

new shadow derived from the original shadow matte. This shadow matte can be translated, scaled, and distorted as desired to allow the user to create a new shadow in the image, or use the resulting layers for compositing and other creative tasks.

## 3.2 Exploration of Parametric Penumbra Models

By penumbra fall-off we mean the way the shadow matte changes in the penumbra, along the direction perpendicular to the shadow edge, from constant multiplicative factor $c < 1$ in the in-shadow region to 1 in the out-of-shadow region (in reality, in-shadow areas rarely exhibit constant attenuation, but, for the sake of simplicity, we will make this assumption in this section); we show example penumbra profiles in Figure 3.3. The exact fall-off depends on many factors such as the shape of the light and the occluder as well their relative positioning. Since these are, for a given 2D image, unknown, we have investigated candidate functions that could be fitted to the fall-off under different conditions. The intuition behind this was that, if there was a model that could represent shadows and have sufficiently distinct characteristics from commonly-seen textures, it should be possible to fit it to intensity profiles and avoid being confused by the texture. Below we present a few models that have been evaluated.

There are three factors that make parametric approaches difficult in practice. Firstly it is difficult to find the right model to fit to the penumbrae, as we demonstrate below. Secondly the model, to be tractable, is usually treated as a 1D intensity slice through the image. This makes it difficult to deal with overlapping shadows, since 1D slices are unable to capture the added complexity. Finally, even given a good model, information about the location and orientation of the penumbra is needed. Mohan *et al.* solve this problem by asking the user to indicate shadow edges, but this is not practical in the case of complex shadows.

### 3.2.1 Second- and Third-Order Polynomials

In the computer graphics literature [Wat93, PSS98] soft shadows are often modeled with a third order polynomial as mentioned in Section 2.1.1, Equation (2.2). However, this function only holds in the case of spherical light source and a straight occluder edge. In other configurations this approximation does not hold, even when the polynomial coefficients are allowed to vary.

### 3.2.2 Logistic Function

Another model that seemed promising for the penumbrae was a sigmoid function. One advantage over aforementioned polynomial was the fact that sigmoid functions are constant everywhere, except the defined interval, making them a natural fit to shadow profiles without the need for piecewise combinations (assuming constant umbra). We have discovered, however, that the logistic functions such as

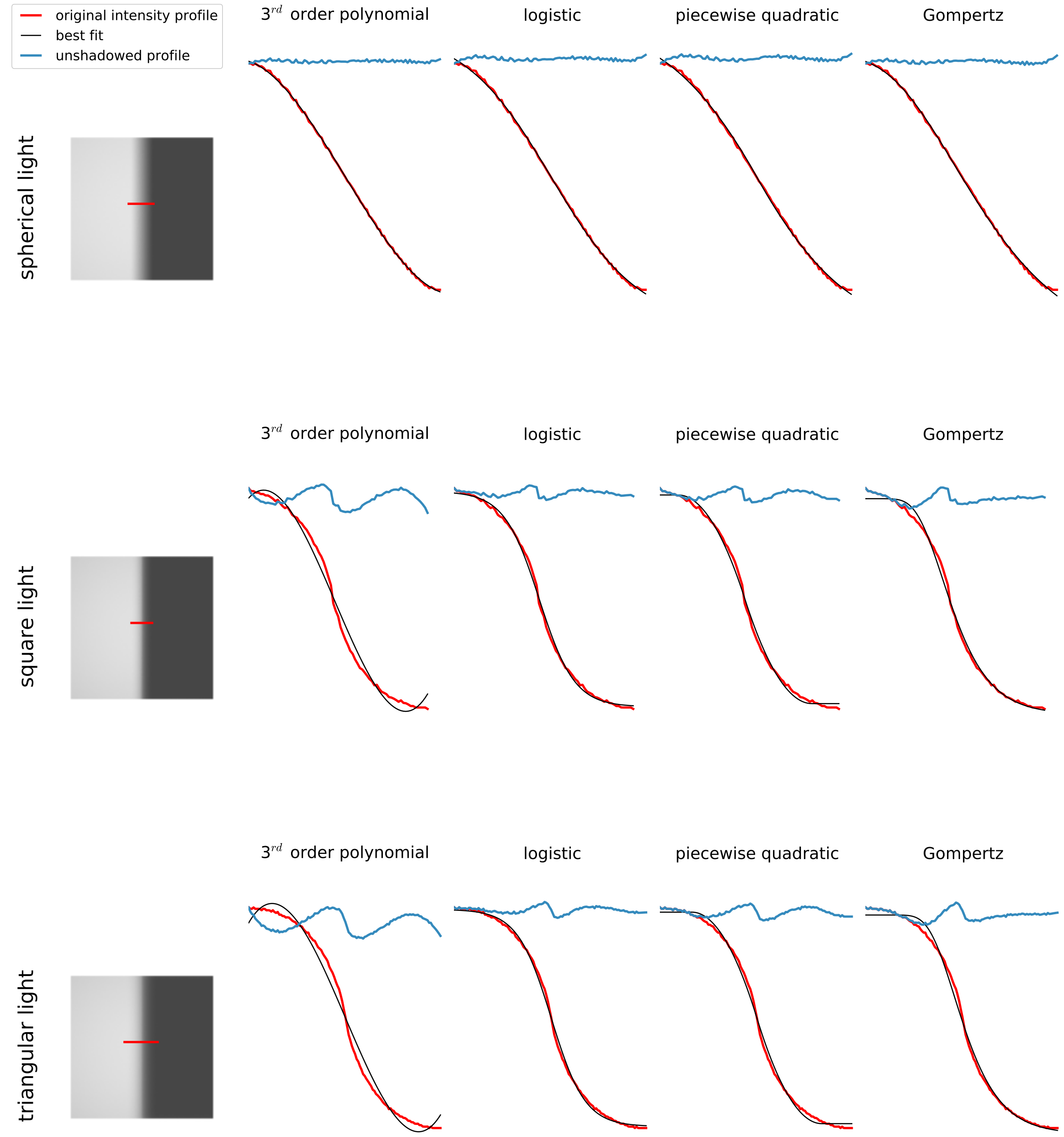


**Figure 3.3:** *Examples of shadow profiles cast by three distinct light shapes and the best-fit curves using different paramteric models. The red lines show the intensity profiles extracted from the marked region in each image, while the black lines show the corresponding best-fit curves. Finally, the blue lines demonstrate the unshadowed intensity profiles,* i.e. *the result of removing the attenuation from the red profiles using the black curves as penumbra models. Since there is no underlying texture, all blue profiles should be flat; while all models are suitable for modeling spherical lights, we have not found one, which fits either square, or triangular illuminators.*

$s_L(\tau) = \frac{1}{1+e^{-\tau}}$ are generally not flexible enough. Firstly they couple the slope parameter with the speed-of-approach. Secondly, they are symmetrical, while penumbrae often exhibit different behaviors at two ends.

### 3.2.3 Piecewise-quadratic Spline

Mohan *et al.* [MTC07] used a piecewise-quadratic function to fit soft shadows. They have performed the fitting at several points along the shadow boundary, by ensuring that the curve gradients at joining points agree and that the resulting profile is a good "explanation" for the observed pixel intensities. Their method was able to correctly unshadow previously-problematic images, however, as we show in Figure 3.3 and mention at the end of Section 3.6.5, failed in cases where their assumed shadow model did not hold.

### 3.2.4 Gompertz Function

While some shadows exhibit different characteristics at the "shallow end" and the "deep end" of the penumbrae, most sigmoid functions are symmetric and, therefore, are unable to capture such behavior. One exception is the Gompertz function, defined as

$$y(x) = ae^{be^{-cx}}. \tag{3.1}$$

Unfortunately, while this model is still not flexible enough to fit real shadows and the resulting unshadowed profiles exhibit noticeable artifacts.

### 3.2.5 Summary

While fitting a parametric model to the penumbra produces a simple solution and works well in some cases, we have not found a model flexible enough to accommodate all scenarios. Different light source and occluder shapes, and phenomena such as multiple penumbrae, and global illumination effects make modeling the penumbra fall-off challenging. In contrast, a learning-based system, which automatically captures such features in the training data, has a better chance of success and directly models 2D surfaces, instead of 1D slices.

## 3.3 Learning and Inference

While shadow mattes are generally unique, our hypothesis is that they can be constructed from a finite set of patches tiled next to each other. We exploit this property and perform learning and inference on a patch-by-patch basis. We considered alternatives to the machine learning approach we present here, such as fitting sigmoid functions to model the soft shadow fall-off. Even with complicated heuristics, these results were unconvincing. Those parametric models needed many degrees of freedom with hard-to-summarize constraints and relationships to adequately approximate different ground-truth mattes. Our learning-based approach focuses on the input/output dimensions that correlate most, and is, broadly speaking, a kind of supervised, non-Euclidean, nearest-neighbour model.

We have empirically determined that patches of $16 \times 16$ pixels work well for our purposes. Further, we assume that colour channels of a shadow matte are related to each other by a scaling factor.

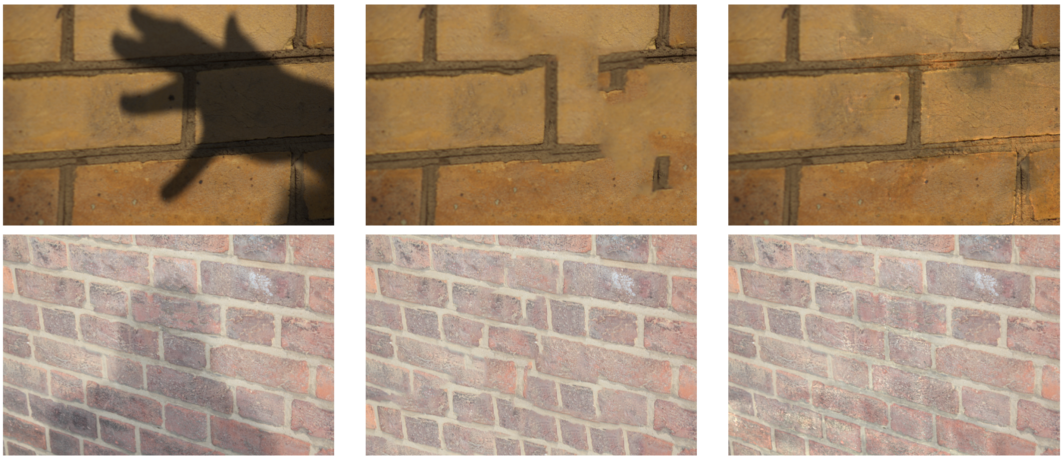


**Figure 3.4:** *Example initial guesses. From the input image (left) we use inpainting to obtain an initial guess for the unshadowed image (middle). That in turn yields an initial-guess matte that forms part of our feature vector and aids regularization. The right column shows the output of our algorithm: an unshadowed image respecting the texture present in the shadow. Note that inpainting alone is unable to recover structure in the above cases.*

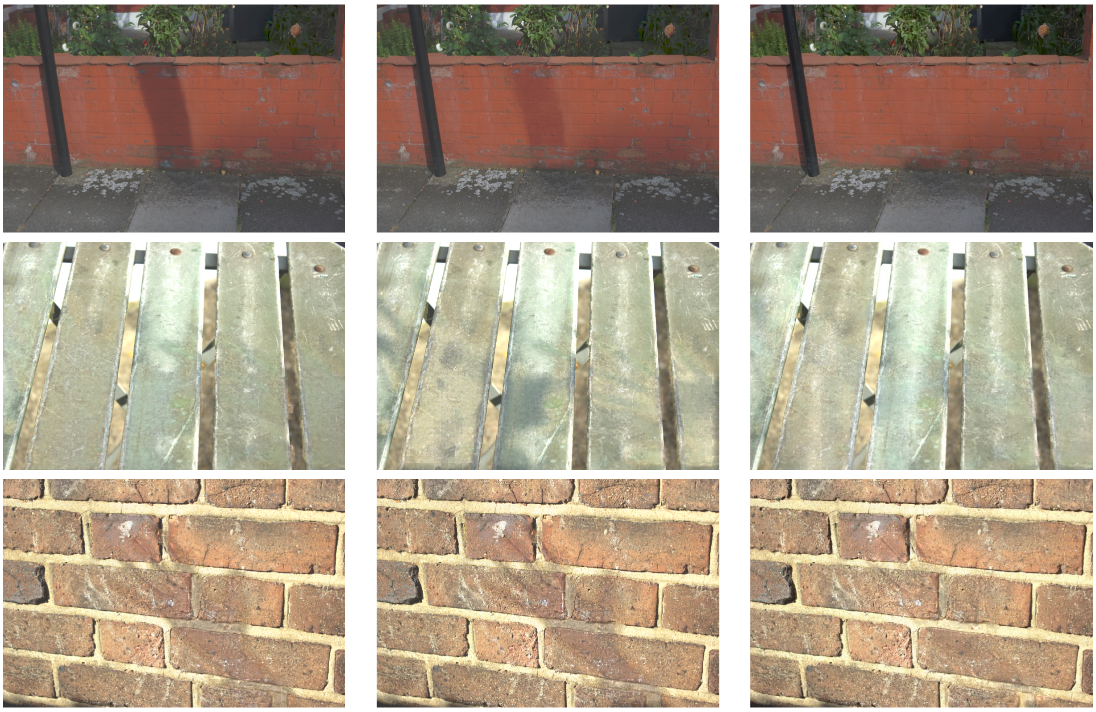
Specifically, we assume it is possible to reconstruct the blue-channel and green-channel mattes given the red-channel matte and the correct scaling factors $\sigma_b$ and $\sigma_g$ (see Section 3.4). We have chosen the red channel for performing learning and inference, and to reconstruct the colour result in the post-processing step. While this splits the problem into two optimizations, it reduces the parameter space that must be learned from data.

### 3.3.1 Preprocessing

Using an off-the-shelf inpainting method by [BSFG09], we first replace the user-specified shadow region completely with a plausible combination of pixels from the rest of the image (see middle column in Figure 3.4). We then apply Gaussian blur in the inpainted region and divide the input image by it to obtain the first approximation to the matte. The blurring step is necessary to both minimize the impact of slight inpainting errors, and to avoid producing spurious frequencies in the image after division.

Our aim is to create a mapping mechanism from RGB intensity patches to grayscale shadow matte patches, so that given a new image we are able to predict the corresponding shadow matte. For consistency with literature, we will refer to the input intensity values as the "feature vector", since we are computing a vector of informative features from the input, and to the output patches as "labels", since we are effectively assigning a label to each input.

**Alternative Inpainting Methods** We have examined two alternatives to this initialization method: a) plane-fit to the unmasked regions of the image, similar to Arbel and Hel-Or 2011 and b) guided inpainting. In guided inpainting, our aim was to replace the shadowed regions of the image with unshadowed pixels from the same image and, ideally, the same material. We have modified the PatchMatch algorithm to replace masked image areas with patches from unmasked regions in the same image. Further, we have modified the distance function used by PatchMatch aiming to make it illumination invariant. To achieve that, we have transformed the image from RGB space to a different, multi-channel space, where the new



(a) plane fitting (b) texture-guided PatchMatch (c) off-the-shelf inpainting

**Figure 3.5:** *Comparison of results using different inpainting approaches. Note that in the case of texture-guided inpainting there are often out-of-shadow parts of the image that the algorithm chooses as the source. This is a failure of the distance function used as described below, however, we were not able to find a better solution.*

channels included the illumination-invariant image from Finlayson et al. 2009, pixel chromaticity, as well as Gabor filter responses. Unfortunately, the final results obtained when using this method proved to be comparable to, but still noticeably worse than, off-the-shelf inpainting. One of the main issues was that images often contained other shadows that were not to be removed as per user input. As a consequence, despite our efforts, the most closely matching patches used to replace the in-shadow regions came from other shadows, therefore providing a poor guidance for unshadowing. A few examples comparing the three approaches are presented in Figure 3.5.

It is important to note here the potential size of our input space: since we need to learn how different shadows look like, when cast on different textures, the learning stage should ideally see examples of all possible shadows cast on all possible textures (while learning algorithms are capable of generalizing, the underlying distributions of the training- and test-set need to be similar). However, this is not practical, and instead, we have instead taken the steps listed below to allow our training set to cover a much smaller part of the entire input space (*i.e.* the "types" and sizes of shadow profiles):

- alignment of patches to make the learning algorithm invariant to Euclidean transformations (see below),
- selection of features aiming to minimize the impact of texture and maximize the information about shadows (3.3.2).

For training, our regressor expects as input a set of small intensity patches $a_i$ (or rather a feature

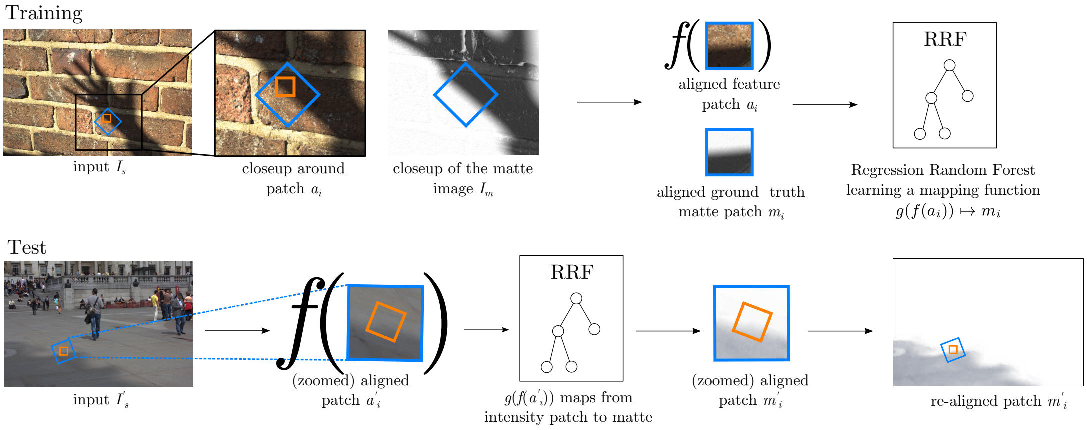
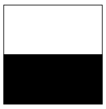

Training
input $I_s$
closeup around patch $a_i$
closeup of the matte image $I_m$
aligned feature patch $a_i$
aligned ground truth matte patch $m_i$
RRF
Regression Random Forest learning a mapping function $g(f(a_i)) \mapsto m_i$
Test
input $I'_s$
(zoomed) aligned patch $a'_i$
RRF
$g(f(a'_i))$ maps from intensity patch to matte
(zoomed) aligned patch $m'_i$
re-aligned patch $m'_i$

**Figure 3.6:** *Patch alignment.* ***At training time*** *(top) we look at each patch in the shadow area (orange square) and find a small-magnitude Euclidean transform to bring the patch as close as possible to the template. We then cut out the expanded patch $a_i$ (blue square) and use it to compute features $f(a_i)$. Additionally, we cut out the same location from the ground truth matte to obtain the label $m_i$. Finally we feed these feature/label pairs to the Regression Random Forest to learn a mapping function $g(f(a_i)) \mapsto m_i$.* ***At test time*** *(bottom), we also start with a grid patch (orange square) and, after finding an optimal offset, cut out the surrounding area $a'_i$ (blue square) from which we compute the features $f(a'_i)$. After pushing these through the forest we obtain a label $m'_i$ that we re-align and crop to paste into the original position in the output image (small, orange square).*

vectors $f(a_i)$ computed over these patches), as well as the corresponding ground truth matte patches $m_i$ as labels, so we need to extract these pairs from our large set of training images. From each image we could potentially extract many thousands of such pairs. To avoid redundancy, we chose to sub-sample each training image to extract $J$ training pairs overall (in all our experiments we have used a training set of size $J = 500\,000$). Additionally, we bias our sampling so that in each training image, half of the samples come from the penumbra region only and half are sampled evenly across the entire shadow (which also includes the penumbra). This avoids over-sampling of uninteresting regions of flat shadow profile and provides greater variety in the training set.

**Alignment** We observe that many patches, though seemingly different in their raw form, can ultimately appear very similar after aligning with an appropriate Euclidean transform. This allows us to perform inference on rotationally-invariant data, effectively multiplying the size of our training set without increasing training time.

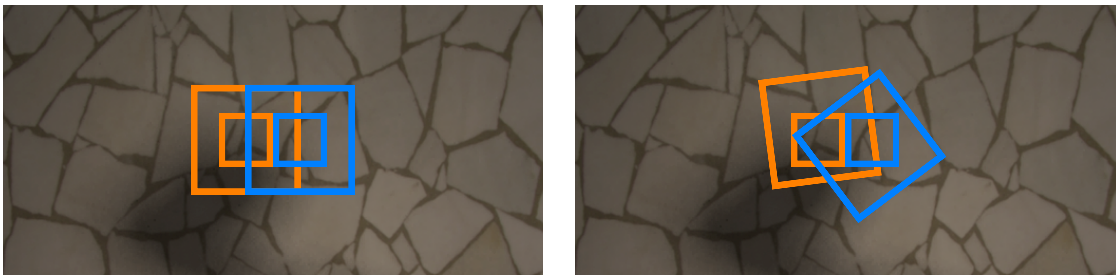
For each intensity patch that we wish to include in our training set, we search through a limited set of rotations and translations to find one that results in the smallest distance to an arbitrary template patch (we use a simple black-and-white square, as shown on the right). We then apply this transformation to both the intensity and the matte patches. At test time, we perform the same procedure before computing features and, after obtaining the estimated label, apply the inverse transformation to it (see Figure 3.6).



### 3.3.2 Learning

For each patch, we form a column *feature* vector by concatenating:

1. pixel intensity values of the patch in the range $[0.0, 1.0]$ shifted in the intensity domain so that their



(a) Before alignment

(b) After alignment

**Figure 3.7:** *Our algorithm needs to output shadow matte values inside the inner squares. However, in the interest of generalizability and smoothness, we compute the feature-label mapping over the larger, aligned patches depicted by outer squares.*

mean falls at $0.5$

2. $x$ and $y$ gradients (finite differences) of the patch
3. distance from the edge of the user-masked region normalized so that the values lie in $[0.0, 1.0]$ range
4. predicted matte for this patch from the initial guess.

The intensity values are normalized, since they are not directly correlated with the matte (given a dark shadowed intensity patch it is impossible to determine whether it is a dark shadow on a bright background, or a bright shadow on a dark background). Therefore we give the inference algorithm processed features that are likely to contribute to the decision (*i.e.* indicating the slope of the shadow), but without confusing differences in absolute intensity.

Our *label* vector contains the pixel values from the shadow matte. Even though at test time we obtain suggestions for each patch in the $16 \times 16$ grid in the shadow region (just the inner squares in Figure 3.7), both our features and labels are computed over a larger $32 \times 32$ window (outer squares). This serves two purposes: to enable smoother results by providing more context to the features, and to aid the alignment and realignment described in Section 3.3.1.

We have chosen to use Random Forest as the inference mechanism both because of its versatility and a widespread use in the literature (e.g. [RDP$^+$11], [SGF$^+$12], [CRK$^+$13]). The simplicity of individual steps in the training and testing pipelines makes it possible to subtly adjust the behavior at various stages. Further, because of the vast space of possible inputs and outputs in our problem, we have adjusted the algorithm to act both as a regressor and as a clustering algorithm. A brief introduction to the traditional Random Forest algorithm below is followed by our modifications and the reasons for introducing them.

Given a standard supervised-training data set of input/output pairs (*i.e.* the feature vectors and the corresponding label vectors), we can use Random Forests to create a set of decision trees that will allow us to predict the label vectors for new, yet unseen feature vectors (provided they resemble the statistics of the training examples). Each of the separate decision trees is imperfect, usually only trained on a random subset of the training data (a process called bagging) and with no guarantees about a globally optimal inference due to the nature of the training process described below. However, averaging the responses of a collection of trees (*i.e.* a "forest"), often results in accurate predictions.

Given a "bagged" set of training data (that is, a subset of all available feature/label pairs), a decision tree is trained as follows. First, we define an impurity, or entropy, measure for a collection of labels, a value that is low when the labels at a node are homogeneous, and high when the labels are different to each other. Then, a binary tree is generated by splitting the available data along the dimensions of the *feature* vector in a way that minimizes the impurity of the split collections. Alternatively, this can be posed as maximizing the information gain—the difference between the original impurity and the sum of child impurities. The generation process starts at the root node, with all the data available to a given tree. It tests a number of random splits along the feature dimensions and chooses the one that results in the largest information gain (an example gain could test if the 7th entry in a feature vector is greater than 0.3). It then creates two child nodes, left and right, and pushes the split data into them. The same process is repeated at each node until stopping criterion is reached (usually a predefined tree depth or a number of samples).

After training, the forest can be used for predicting the labels for new feature vectors. The feature vector in question is "pushed" down each tree depending on the values of individual features and the node thresholds. After arriving at a leaf node, the mean label of all the training samples that landed at this node is taken as the answer of this tree. Finally, answers of all trees are averaged to get a more robust prediction.

We use a modified version of Multivariate Regression Random Forests in this work. While Random Forests in general have been well-explored already, their use for practical multivariate regression has been limited [CRK$^+$13]. One of the challenges lies in computing node impurity—in classification, this can be done easily by counting samples belonging to each class, whereas in regression, one needs to evaluate the probability density function, which can be costly in high dimensions.

Our labels lie in $\mathbb{R}^{2N \times 2N}$ (where $N = 16$ and $2N$ comes from the fact that we store areas larger than the original patches). However, we observe that they can be effectively represented in lower-dimensional space, since penumbrae generally do not exhibit many high-frequency changes. Moreover, we only need the representation to be accurate enough to cluster similar labels together—we do not lose detail in the final answer because of the non-parametric nature of our inference method described below (in short we build the forest based on the low-dimensional representation, but retrieve final labels in the original, high-dimensional, space). Therefore, we use PCA to project our labels into $\mathbb{R}^{D=4}$, which provides good balance between the degrees of freedom necessary to discriminate between patches, and computational complexity while evaluating impurities. Specifically, at each tree node $n$, we fit a Gaussian distribution to the labels of all samples $\mathcal{S}_n$ falling into it, and evaluate impurity

$$H_n = \log(\det \Sigma_{\mathcal{S}_n}) \tag{3.2}$$

by taking the log of the determinant of its covariance matrix $\Sigma_{\mathcal{S}_n}$ following [CRK$^+$13].

This allows us to define the information gain

$$G_n = H_n - \sum_{c \in \{l,r\}} (|\mathcal{S}_c|/|\mathcal{S}_n|)H_c, \tag{3.3}$$

that we aim to maximize at each split node while building the trees. We weight the information gain by the proportion of samples falling into each child node ($l$ and $r$) to encourage more balanced trees as in [BFSO84].

We set the minimum sample count at a node to $K = 2D$ and grow our trees as deeply as necessary until we do not have enough samples to split. In principle, $K$ could be as low as $D$ (number of samples needed to compute a $D$-dimensional covariance matrix). However, in practice, we find that the samples are often not linearly independent, leading to degeneracies. After a leaf node is reached, instead of building a typical parametric distribution of all its labels, we save the indices of training samples falling into this node, allowing us to perform inference as described in the next section.

We have evaluated how the performance of our algorithm varies with the amount of training data supplied. First, we have kept the number of patches sampled per image constant and set the number of training images rendered to 1000, 5000, 10000 and 15000. We have empirically determined that using up to 10000 images provides a good balance between the training time and performance, while beyond the trade-off becomes less attractive. Secondly, we have used similar procedure to arrive at the appropriate number of patches sampled per image.

### 3.3.3 Inference

Our inference step acts as a constraint on the initial guess—we want to “explain” the initial-guess mattes as well as possible using samples from our training set, but only those suggested by the forest as relevant.

At test time, we compute the feature vector for each patch as before and, after pushing it through each tree, arrive at a leaf node. From here, instead of looking at the label distribution, we simply get the indices of training samples that fell into this leaf. Consequently, we obtain $L$ label suggestions, where $L \geq TK$ and the number of trees in the forest $T = 25$. We do this for each patch in the $16 \times 16$ grid in the shadow region and arrive at an optimization problem: for each image patch in our matte we want to choose one of $L$ labels that agrees both with our initial guess and any available neighbours.

In summary, the changes we have made to the original RRF algorithm are:

1. Using two representations of labels: low-dimensional used to evaluate node impurity and build the forest, and high-dimensional used for retrieving the labels at test time. This is motivated by computational constraints and enabled by the non-parametric treatment of labels.

2. Non-parametric treatment of the labels to avoid over-smoothing. Instead of one mean answer in label-space, we get a distribution of samples from the data, including extremes, which we want to preserve.

3. Treating the inference algorithm as a step in the pipeline, rather than the entire pipeline. We only get an intermediate result from the forest (several matte patch suggestions for each patch)

and use regularization later on to extract the final answer. As above, this allows us to benefit from relatively limited amounts of training data (compared to the number of theoretically possible labels in $256^{16\times16}$-dimensional space), without averaging out unusual shadow profiles.

### 3.3.4 Regularization

Finding a more specific combination of matte patches is not trivial due to the nature of our labels and the fact that there might be different numbers of label suggestions available at each node. Averaging all the candidate patches at each location would not be optimal, since any unusual shadow profiles would be lost. On the other hand, choosing best-fitting patches greedily and then trying to smooth out the edges between them would a) be extremely difficult to do for small patches that are likely to be incompatible and b) introduce an implicit, non-data-driven, shadow model in smoothed regions. Instead, at each location, we choose the best patch by regularizing the entire graph with the TRW-S message passing algorithm [Kol06]. We use the energy function

$$E = \sum_{i \in \mathcal{I}} \omega(m_i) + \lambda \sum_{i,j \in \mathcal{N}} \psi(m_i, m_j), \tag{3.4}$$

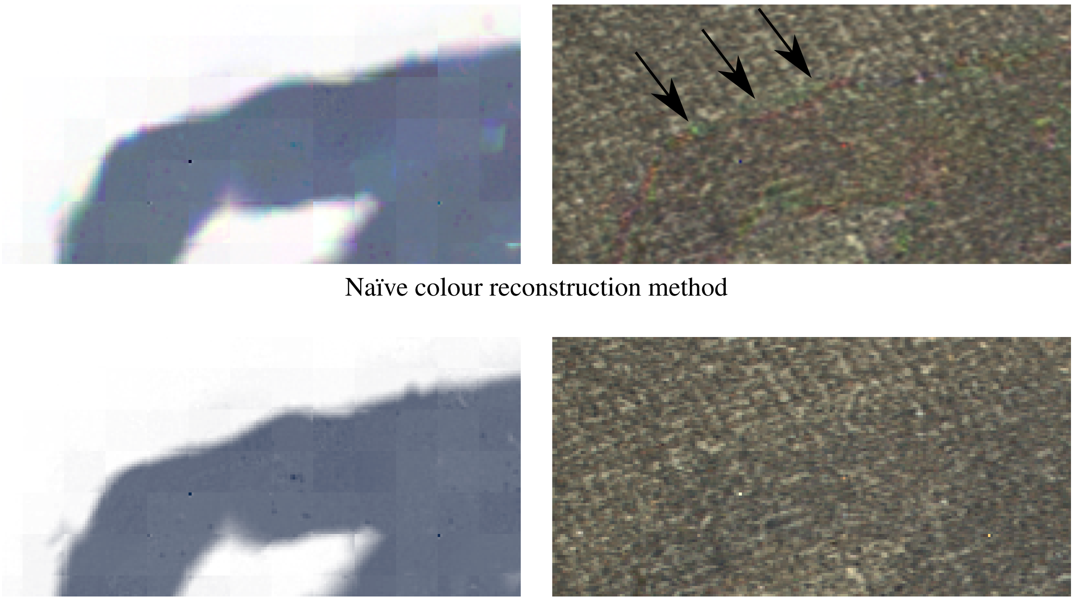
where $\mathcal{I}$ is the set of nodes in the regularization graph (*i.e.* all the masked image patches) and $\mathcal{N}$ denotes the set of neighbouring nodes in a 4-connected neighbourhood. The unary cost $\omega(m_i)$ is the SSD distance from the patch $m_i$ to the corresponding area in the initial guess, the pairwise cost $\psi(m_i, m_j)$ is the compatibility of patch $m_i$ to $m_j$, and $\lambda$ is the pairwise weight ($\lambda = 1$ in all our experiments). We define the patch compatibility $\psi(m_i, m_j)$ as the sum of squared differences between adjoining rows (or columns) of these two patches:

$$\psi(m_i, m_j) = \begin{cases} SSD\Big(row_N(m_i), row_1(m_j)\Big), & \text{if } m_i \text{ is above } m_j \\ \\ & \text{if } m_i \text{ is to the} \\ SSD\Big(col_N(m_i), col_1(m_j)\Big), & \text{right of } m_j \end{cases} \tag{3.5}$$

where $row_N(m_i)$ and $col_N(m_i)$ are the last row and column of patch $m_i$ respectively. We also create boundary constraints to ensure that the shadow disappears outside of the user-selected region by forcing patches just outside of the user mask to constant $1.0$ (meaning completely shadow-free).

## 3.4 Colour Optimization

We could repeat the above procedure twice more to obtain the remaining green and blue channel mattes. Indeed, in theory, this would be the most general solution, assuming no relationship between different frequencies of light. In practice, however, we find that this relationship is quite strong, and providing an additional constraint to enforce it makes our method more robust. The top row in Figure 3.8 shows a shadow matte and the corresponding unshadowed image estimated by the naïve way, *i.e.* each channel separately. Note the splotches of different colours revealing where the mattes of different channels do



Naïve colour reconstruction method

Our method

**Figure 3.8:** *Treating each channel independently results in inconsistent shadow mattes (top left) that manifest themselves with colourful splotches in the unshadowed output image $I_u$ (top right). Our method assumes that the three colour channels are dependent—it only regresses one of them and reconstructs the other two as explained in Section 3.4. Please see the digital version for faithful a colour representation.*

not agree. The bottom row shows our default procedure, described here.

We assume that the surface of the shadow matte has the same shape in all three channels, but that it differs in magnitude as shown in Figure 3.9. For instance, in outdoor scenes the shadow mattes are blueish. On a sunny day there are two illumination sources: the Sun (white) and the sky (blue); out-of-shadow regions are illuminated by both light sources, while the in-shadow regions, only by the sky. Consequently, in-shadow areas and, therefore, shadow mattes appear more blue. In such mattes the red channel usually has the lowest intensity, therefore, the green and blue channels can be obtained by scaling the red channel matte so that areas with no shadow remain that way, but the overall depth of the shadow changes proportionately. This assumes that, there are at most two differently-coloured light sources.

Relative to the estimated red channel shadow matte, we model each of the other channels with a single scale factor parameter, $\sigma_g$ and $\sigma_b$ respectively. To estimate them jointly, we discretize and search the 2D space to minimize the error function

$$E_{colour}(\sigma_g, \sigma_b) = \log(\det(\Sigma_{\mathcal{R}})), \tag{3.6}$$

where $\Sigma_{\mathcal{R}}$ is the covariance of a three-column matrix $\mathcal{R}$ listing all the RGB triplets in the unshadowed image after applying that colour matte. We constrain the search in each parameter to lie between $0.8$ and $1.2$ with discrete steps of $0.01$. We find that the scaling factors $\sigma_g$ and $\sigma_b$ rarely exceed $1.1$ and never fall below $1.0$ in outdoor scenes.

The intuition behind this approach is that unshadowing an image should not significantly change

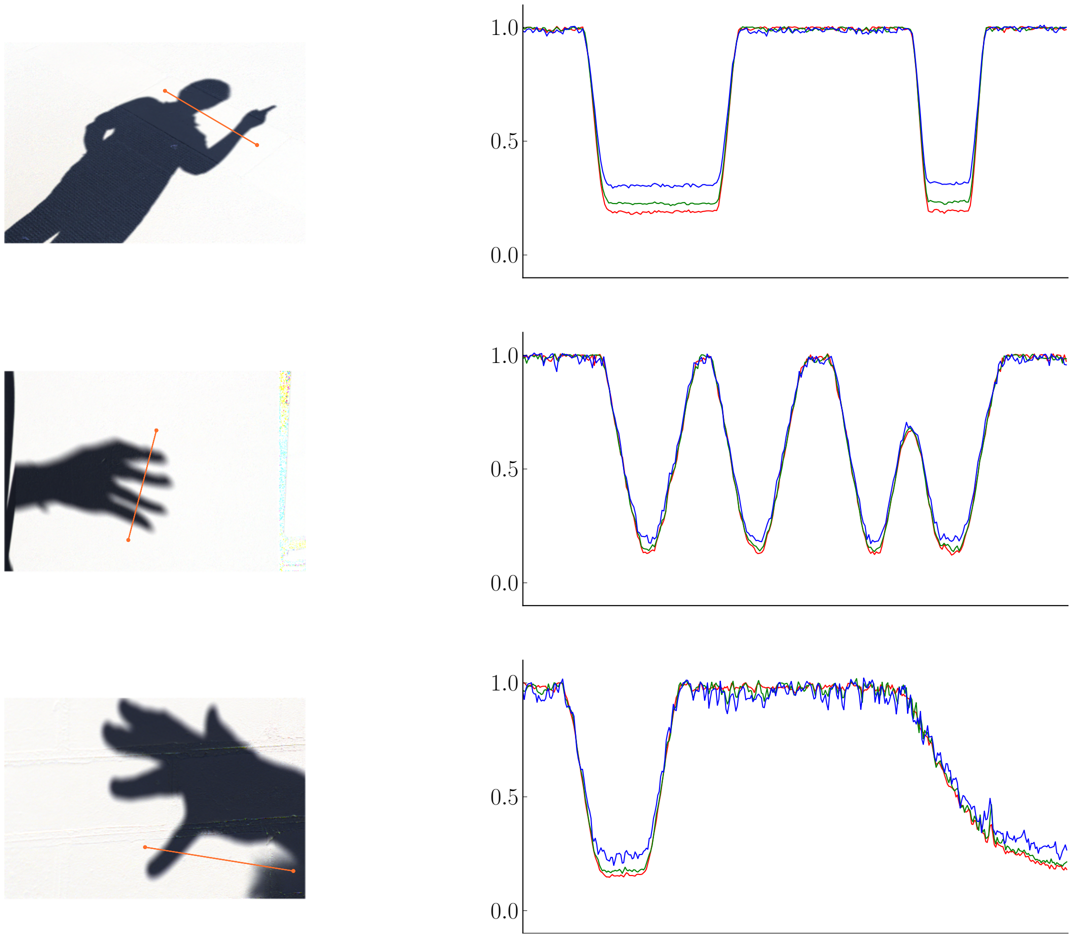


**Figure 3.9:** *Ground truth shadow matte profiles for RGB colour channels. Note that, disregarding noise, all channels share the same basic "shape" and reach* 1.0 *(no-shadow zone). Our experiments indicate that acceptable green and blue mattes can usually be reconstructed by optimizing (3.6).*

the distribution of colours in it. Since introducing new colours would increase the entropy of $\Sigma_{\mathcal{R}}$, we use this measure to find scaling factors that minimize it.

For efficiency, we run this optimization on a version of the output image downsampled to 10% of its original size. This optimization serves to prevent our unshadowing method from introducing new colours into the images. The user can override these scale parameters with our shadow-editing interface (Section 3.6.1), but all our results are shown with the automatic adjustment unless specifically stated.

## 3.5 Data Generation

To train our model, we need large amounts of data to capture a variety of scene configurations and shadow-receiving surfaces. Since it would be extremely difficult to capture a large enough set of real images, we follow [MABP10] in training on a synthetically-generated training set and applying it to real data. We have carefully configured Maya with realistic lighting conditions to generate shadows cast onto various textures as illustrated in Figure 3.10. For each light-occluder-texture combination, we have rendered the image with and without shadow, implicitly obtaining the corresponding matte.

While shadows in the real world are cast by three-dimensional objects, for each shadow there also exists a 2D slice through the occluder that would produce identical results. Therefore, we have used automatically-segmented silhouettes of real objects from [GHP07] as occluders in our synthetic scenes (we have segmented them automatically by using [BVZ01]). This has the advantage over using 3D models of providing realistic silhouettes, as long as the images used are easily segmentable. Additionally, a variety of real images[1] were applied as textures for the receiving surfaces.

Finally, we varied light conditions in the scene by randomly choosing the shape and size of the light, its angle and distance from the ground, as well as the angles of the occluder and the ground plane.

In our experiments, we have trained the model on over $10{,}000$ $512 \times 512$ image pairs. Additionally, we have automatically generated binary shadow masks to only train on relevant regions. We have rendered all our images without gamma correction to make the relationship between image intensity and shadow attenuation easier to model.

## 3.6 Experiments and Results

To capture real data for evaluation, we have used a Canon DSLR camera to capture 16-bit-per-channel RAW, linear images. We provide this set of 137 photographs, 37 of which have the corresponding ground truth shadow-free images (and mattes), as a benchmark for future methods. The ground truth was obtained by placing the camera on a tripod and capturing two images: one with the shadow and one without by removing the shadow caster. For our experiments, we have converted the images to 8-bit-per-channel linear PNGs and, after processing, de-linearized them for display by applying gamma correction with $\gamma = 2.2$.

#### Feature Vector

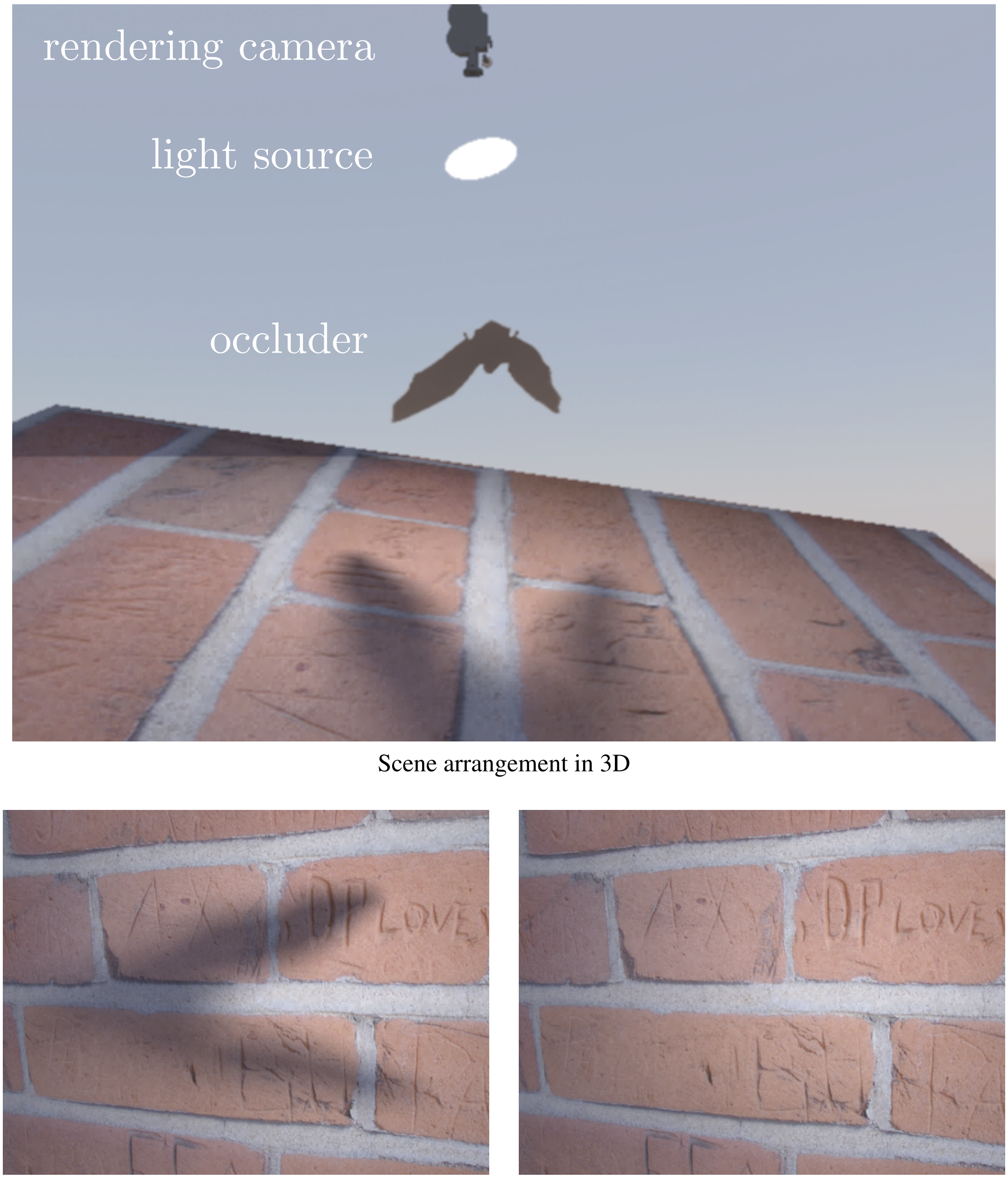
Here we present a short exploration of the relative importance of features in our feature vector. While feature importance is a very informative measure, it would not be feasible to compute using our full dataset. Instead, in Figure 3.11 we present “feature frequency”, a number of times each feature was used as a split in the entire forest.

We have also explored other features, such as explicit pixel-wise gradient orientation and magnitude, and the angular position of the patch within the masked region. However, none of them offered substantial improvements in performance.

### 3.6.1 Shadow Editing

Knowing the shadow matte allows us to not only remove the selected shadow, but also enables a range of high-level image editing techniques. We have implemented a basic interface with four different alteration methods to give artists control over how the selected shadow looks: shape transformation, changing brightness and colour, and sharpening the shadow. Colour, brightness, and sharpness are adjusted using sliders, while direct manipulation controls enable a direct homography transform, allowing users to change the position and shape of the cast shadow. Please see the supplementary video for examples.

[1] http://www.mayang.com/textures/



Scene arrangement in 3D

Shadowed $I_s$

Shadow-free $I_u$

**Figure 3.10:** *To generate the training data, we rendered* $10{,}000$ $\{$*shadowed, shadow-free*$\}$ *image pairs, each time varying the light source, the occluder, and the ground plane.*

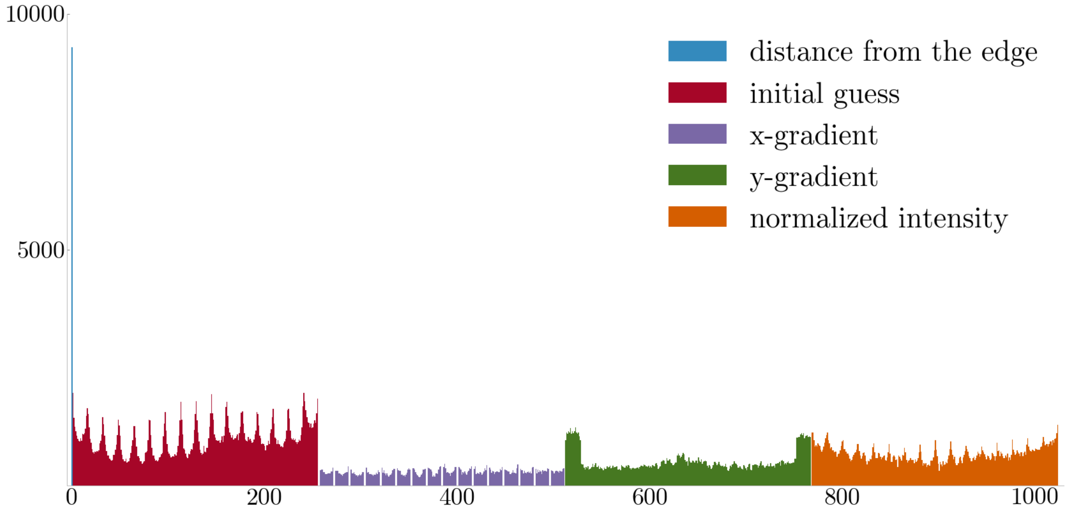
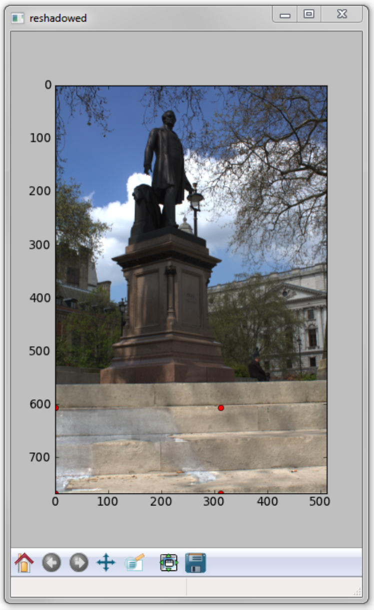
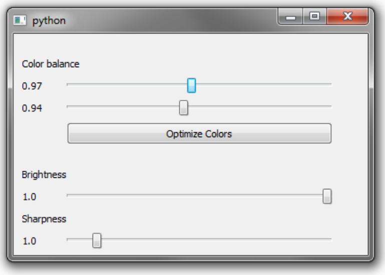


**Figure 3.11:** *Number of times each part of our feature vector was used as a split dimension at a node. Note the spike at position 0 indicating that the distance from the mask edge is an important feature.*





Color balance
0.97
0.94

Brightness
1.0

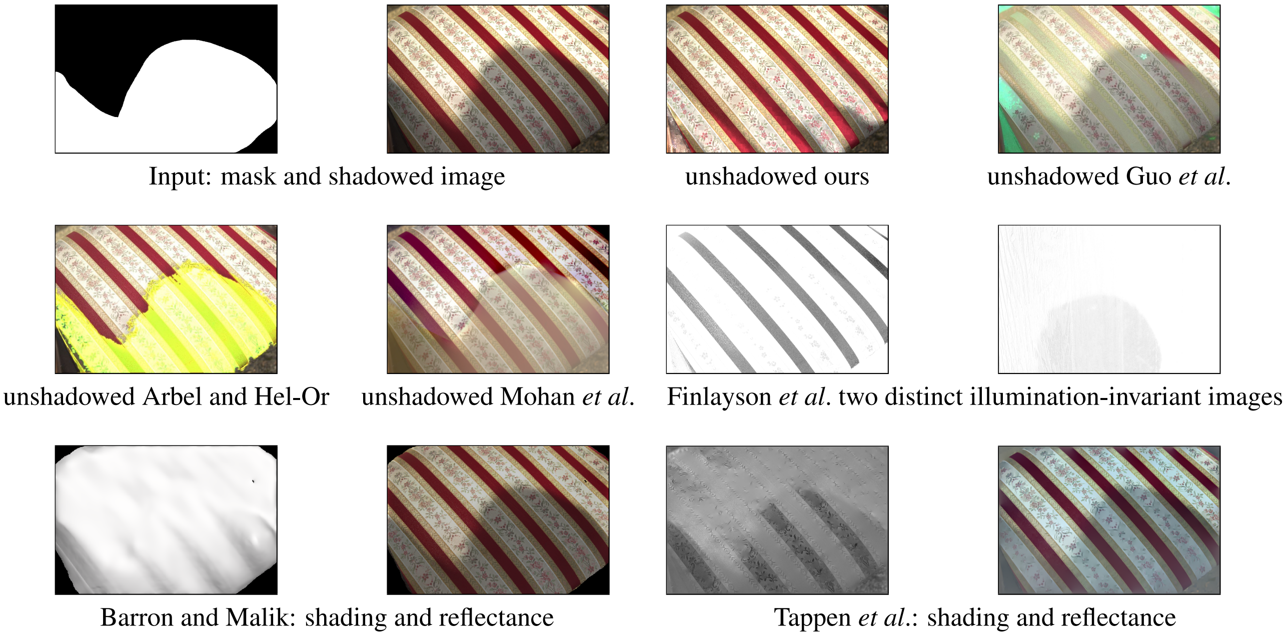
Sharpness
1.0

We can use the recovered shadow mattes to aid tasks such as compositing, texture extraction *etc.*, which are normally challenging tasks requiring substantial manual work. Both the matte and the unshadowed image can be exported to any number of generic image editing tools for further processing.

### 3.6.2 Visual Comparison With Other Methods

We have evaluated our algorithm against other related techniques and display results in Figure 3.12. We have chosen the best results we were able to obtain for this image using the original implementations of [MTC07] and [AHO11], but since the user has some freedom in the use of their systems, we cannot exclude that a more skilled person could achieve better outcomes (note that for Mohan *et al.* we have downsampled the image by 50% to speed up processing). Please see the supplementary material for many more results.

While intrinsic image algorithms could be considered an alternative to shadow matting, it is important to note that they have different goals and it would not be fair to directly compare the two, so the examples are shown just for illustration. While shadow matting usually deals with cast shadows, intrinsic image techniques generally decompose images into reflectance and shading components where, in practice, shading mostly refers to attached shadows. Most of these techniques



Input: mask and shadowed image

unshadowed ours

unshadowed Guo *et al.*

unshadowed Arbel and Hel-Or

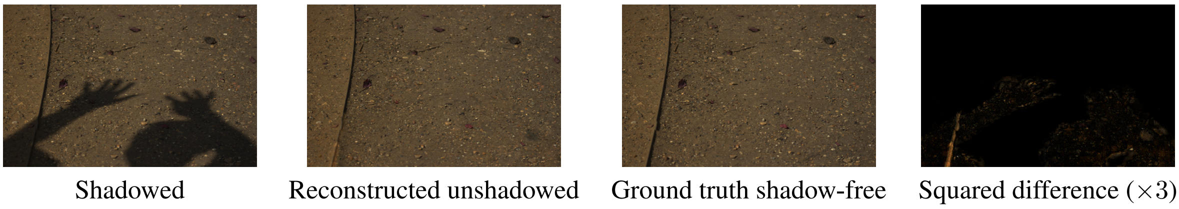
unshadowed Mohan *et al.*

Finlayson *et al.* two distinct illumination-invariant images

Barron and Malik: shading and reflectance

Tappen *et al.*: shading and reflectance

**Figure 3.12:** *Comparison with other methods. Note that both Barron and Malik and Tappen* et al. *perform slightly different image decomposition: rather than matting cast shadows, they extract shading and reflectance components. (The second illumination-invariant result for [FDL09] comes from the shadowed image in Figure 3.27.*



Shadowed

Reconstructed unshadowed

Ground truth shadow-free

Squared difference (×3)

**Figure 3.13:** *While the aim of our work is to enable perceptually-plausible results, here we show the differences between the output and the ground truth.*

are poorly equipped to recognize cast shadows as illumination changes unless given access to additional data such as in [Wei01].

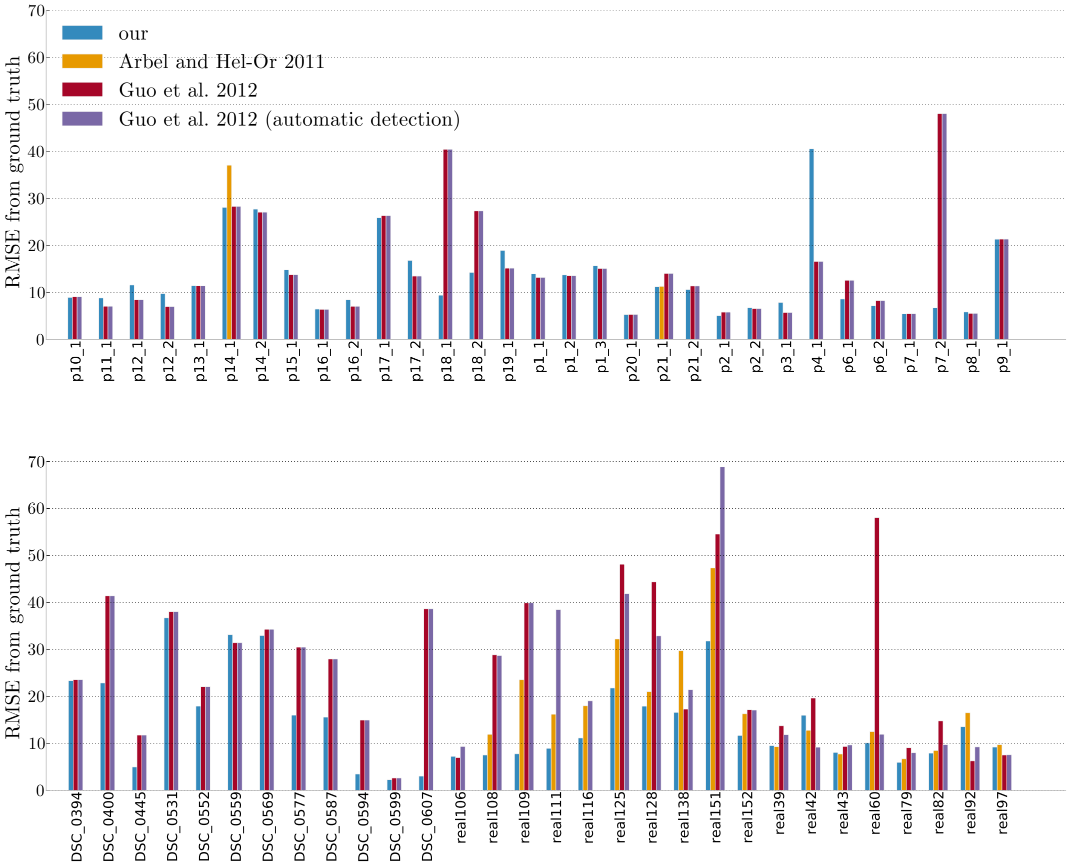
Finally, the method presented by [FDL09], does not provide perfect illumination-invariant images, as shown in Figure 3.12. In the first image, while the ilumination differences are not visible, some reflectance-induced gradients were removed as well (flower patterns in the top part of the image). For a *different* input, in the image on the right, illumination differences are still visible.

### 3.6.3 Quantitative Evaluation

Figure 3.14 (a) shows quantitative evaluation of our and related methods in terms of RMS error from ground truth (for this evaluation we have used all images, which were processed by all four methods). Additionally, in Figure 3.14 (b) we visualize per-image RMSE for different methods to give a more complete characterization. Note that quantitative comparisons are not, in general, representative of perceptual differences and are included here only for completeness.

| | **Mean RMSE** |
|---|---|
| Ours | **13.83** |
| Arbel and Hel-Or 2011 | 18.36 |
| Guo at al. 2012 | 19.85 |
| Guo at al. 2012 (automatic detection) | 19.19 |

(a)



(b)

**Figure 3.14:** *(a) RMSE between results of different shadow removal methods and the ground truth shadow-free images. While our scores are best, the most important aim is to convince subjects that the resulting images are unaltered, rather than obtaining the best numerical result. (b) Per-image RMSE between results of different shadow removal methods and the ground truth shadow-free images.*

| Scene Name | Mean Pairwise RMSE |
|---|---|
| real22 | 8.35 |
| real26 | 6.15 |
| real138 | 14.55 |
| real168 | 1.02 |
| real249 | 1.58 |

**Table 3.1:** *Differences between images unshadowed by different users. Each of the 5 scenes was unshadowed by 4 users. For each scene, RMS differences were computed between each image pair, and the mean of these errors is shown above. Please refer to the supplementary material to see all the user-provided masks and the resulting unshadowed images.*

### 3.6.4 Impact of Variations in the User Input

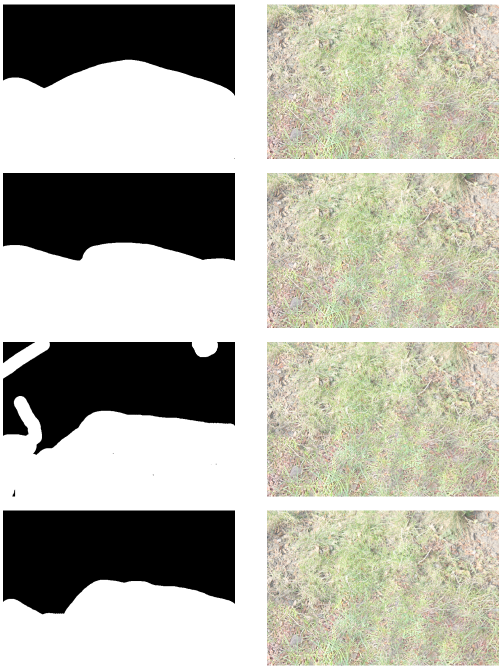
Our method does not automatically detect shadows in the image, instead giving the user control over which regions should be modified. To establish how robust it is to variation in the user input, we have asked 4 users to create masks for 5 different scenes and ran the unshadowing algorithm for each input.

Each user was instructed to paint over the areas of the image containing the shadow and to prefer over- to under-selection for consistency with our assumptions (namely, that it is difficult to exactly determine boundaries of soft shadows and that our algorithm is only allowed to modify selected regions).

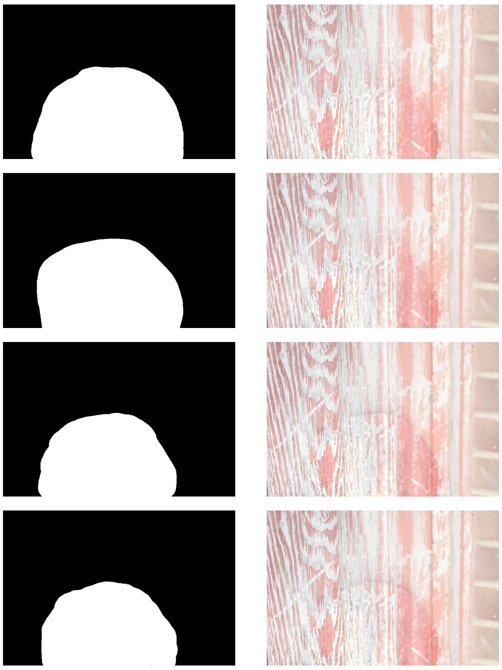
To properly investigate the impact of user input only, we have constrained the inpainting to be the same for each user. This is necessary, since the inpainting algorithm we use is non-deterministic and constitutes the main source of variation between runs. After having all the user-provided masks we have created a union of them (*i.e.* pixel-wise logical-`or`) and used it as the inpainting mask.

As Table 3.1 indicates, the final results are fairly robust to variance in user input. The largest differences are caused by users underestimating the extent of the shadow and thus not marking some regions for modification. We have included the different input mattes and corresponding unshadowed images below.

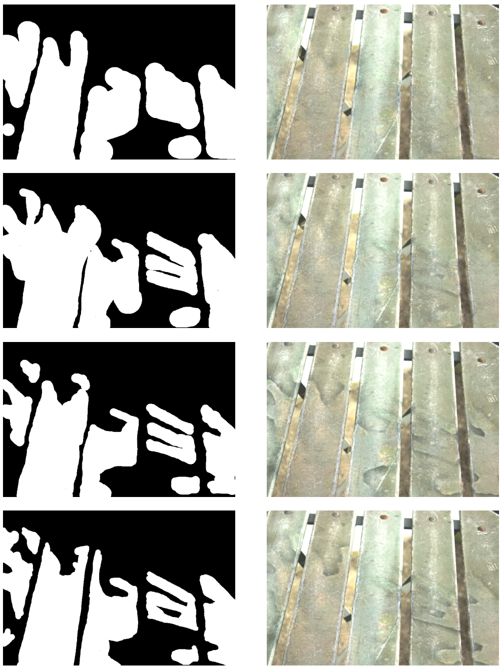
We conclude that small differences in the provided masks do not contribute significantly to the quality of final results as long as all the penumbrae are selected. The exception is the bottom row in most of the scenes, which shows penumbrae parts being left untouched, which is caused by this user's tendency to underestimate the extent of the shadow. Recall that we intentionally constrain the unselected regions to be considered out-of-shadow and therefore unquestioningly trust the users' judgment. This design decision could be revisited *e.g.* by automatically dilating the masks by a certain amount.



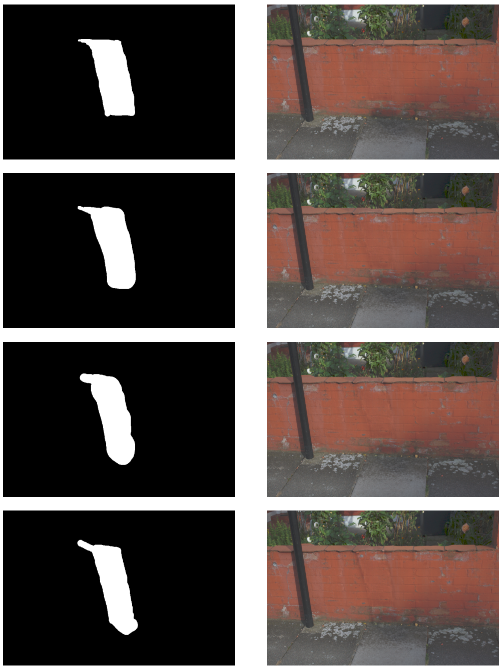
real22



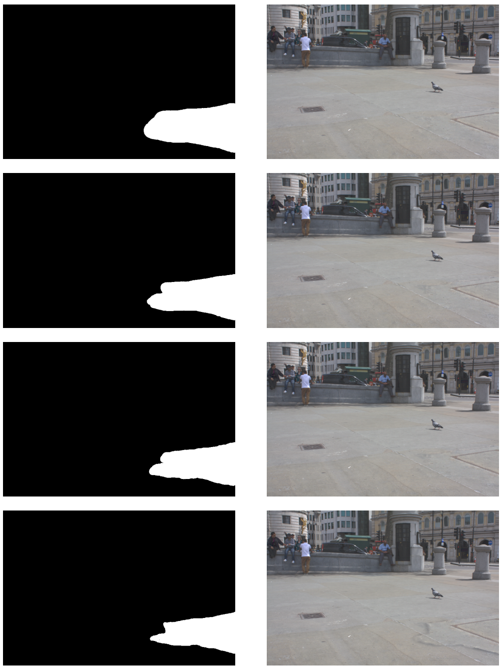
real26



real138



real168



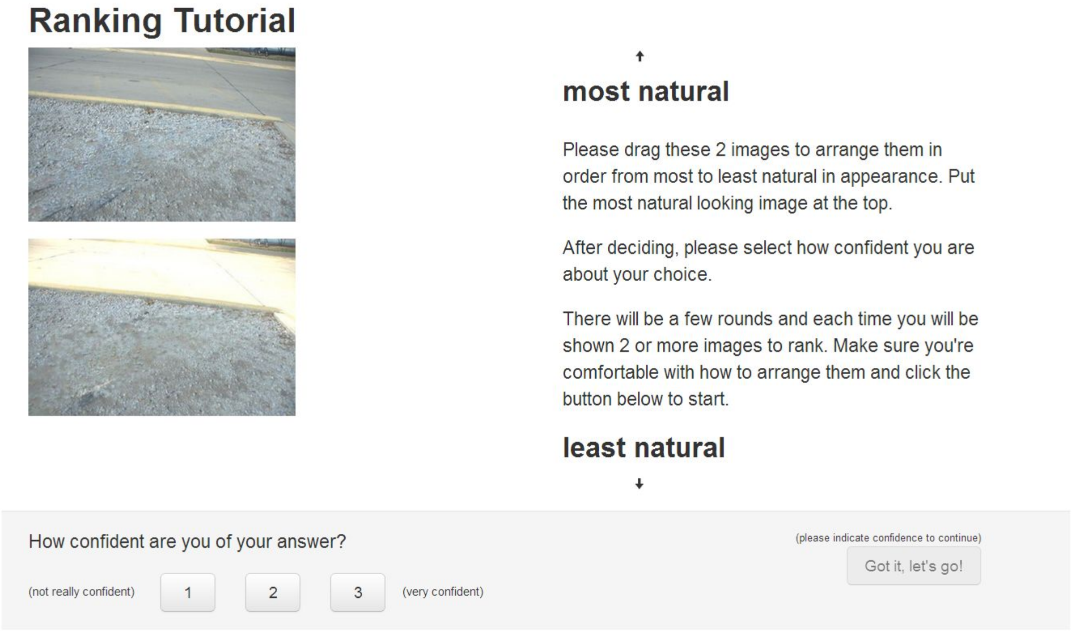
real249

### 3.6.5 User Study

To understand how our results compare to those produced by prior work, specifically the methods of [GDH12] and [AHO11], we conducted a user study similar in spirit to [KKDK12]. We have modified the method of Guo *et al*. to sidestep the automatic shadow detection and instead use the same shadow mask that was given to our algorithm (though the results from Guo *et al*.'s unmodified version are included in the supplementary material). Please also note that the method of Arbel and Hel-Or required more input that was provided manually for each image.

We have assembled a set of 117 images for user evaluation by combining a subset (soft shadows only) of the dataset released by Guo *et al*. with a subset of our own images: 65 from Guo's and 52 images from our dataset. The complete set of images used can be found in the supplementary material. Our study consisted of two phases: a) a series of *ranking* tasks in which participants ordered a set of two or three images, and b) a series of *evaluation* tasks in which participants indicated the success of shadow removal on a particular image using a 4-point Likert scale. Both phases started with a tutorial example and displayed instructions on the right side of the screen throughout all trials. Additionally, in each phase, we have asked the participants to estimate their confidence in their choice on a 3-point Likert scale. The interfaces used are shown in Figure 3.15 and Figure 3.15 below.

In the ranking phase, participants were shown 15 random image tuples (from a set of 117), where each tuple consisted of images of the same scene modified with one of three methods: ours, Guo *et al*.'s and Arbel and Hel-Or's. Using a drag-and-drop interface, participants were asked to order the images according to how natural they looked (*i.e*. from most to least natural). The aim of this part of the study was to establish how believable the produced results were, without the participants being aware that any shadows were removed. Roughly half of the tuples showed results from each of the three methods, and half paired our result with one of either Guo *et al*. or Arbel and Hel-Or. For all participants, the order of



Ranking Tutorial

most natural

Please drag these 2 images to arrange them in order from most to least natural in appearance. Put the most natural looking image at the top.

After deciding, please select how confident you are about your choice.

There will be a few rounds and each time you will be shown 2 or more images to rank. Make sure you're comfortable with how to arrange them and click the button below to start.

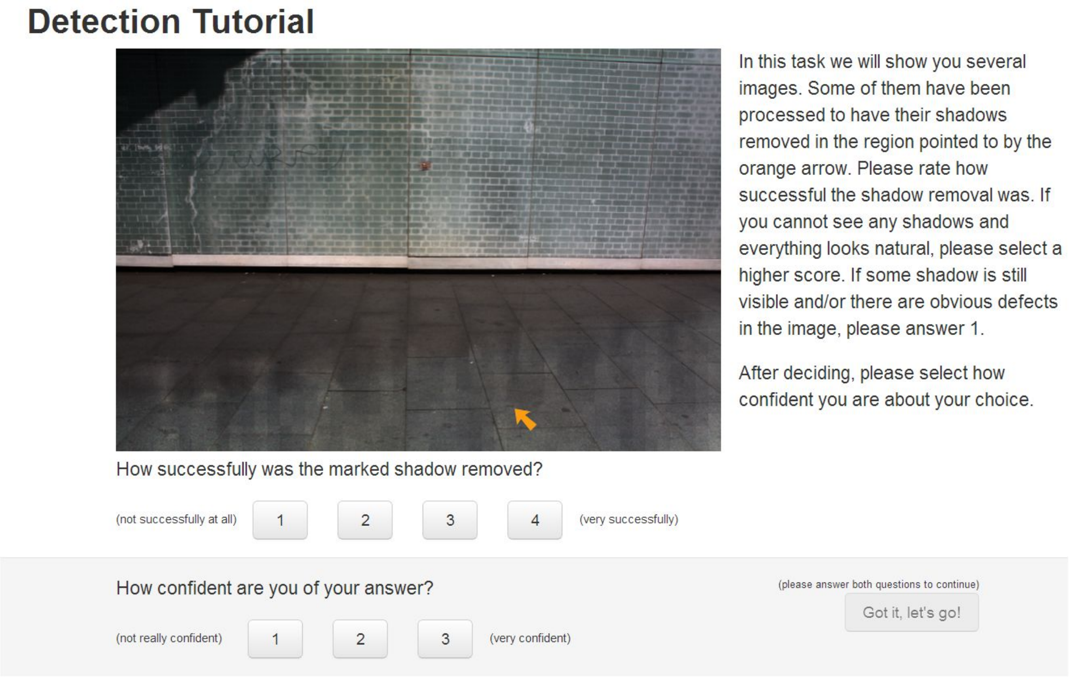
least natural

How confident are you of your answer?

(not really confident) 1 2 3 (very confident)

(please indicate confidence to continue)

**Figure 3.15:** *Screen capture of Task 1 of the user study. The participants were asked to decide which of the presented images seemed more natural and arrange the presented results by click-and-drag.*



Detection Tutorial

In this task we will show you several images. Some of them have been processed to have their shadows removed in the region pointed to by the orange arrow. Please rate how successful the shadow removal was. If you cannot see any shadows and everything looks natural, please select a higher score. If some shadow is still visible and/or there are obvious defects in the image, please answer 1.

After deciding, please select how confident you are about your choice.

How successfully was the marked shadow removed?

(not successfully at all) 1 2 3 4 (very successfully)

How confident are you of your answer?

(not really confident) 1 2 3 (very confident)

(please answer both questions to continue)

**Figure 3.16:** *Screen capture of Task 2 in the user study. The participants were presented with results of shadow removal using either our method, Guo* et al.*'s or Arbel and Hel-Or's and asked to indicate how successful the shadow removal was in the marked region.*

the tuples was randomly chosen, along with the order of the images within the tuple.

In the second phase, 15 images were randomly drawn for each user from a pool of 282 images. These were the same set of images as in the first phase, however, now each image was shown separately rather than in a tuple. Of these images, 118 were processed using our technique, 114 were processed using Guo *et al.*'s, and 50 using Arbel and Hel-Or's. The set of 15 images was randomly chosen subject to the constraint that the images seen during the first phase could not be used in the second. Each of these images was augmented with a bright arrow pointing to the centroid point of the shadow that was to be removed, and participants were asked to assign a score $\{1, 4\}$ based on how successfully they thought the shadow was removed. Low values corresponded to cases where the shadow was not removed or where the removal introduced artifacts, while high scores indicated successful shadow removal and no visible defects. The centroid was computed automatically by finding a point on the skeleton of the binary shadow mask closest to the mean position of the masked pixels. The order of the two phases (ranking then evaluation) was fixed for all participants.

### Results

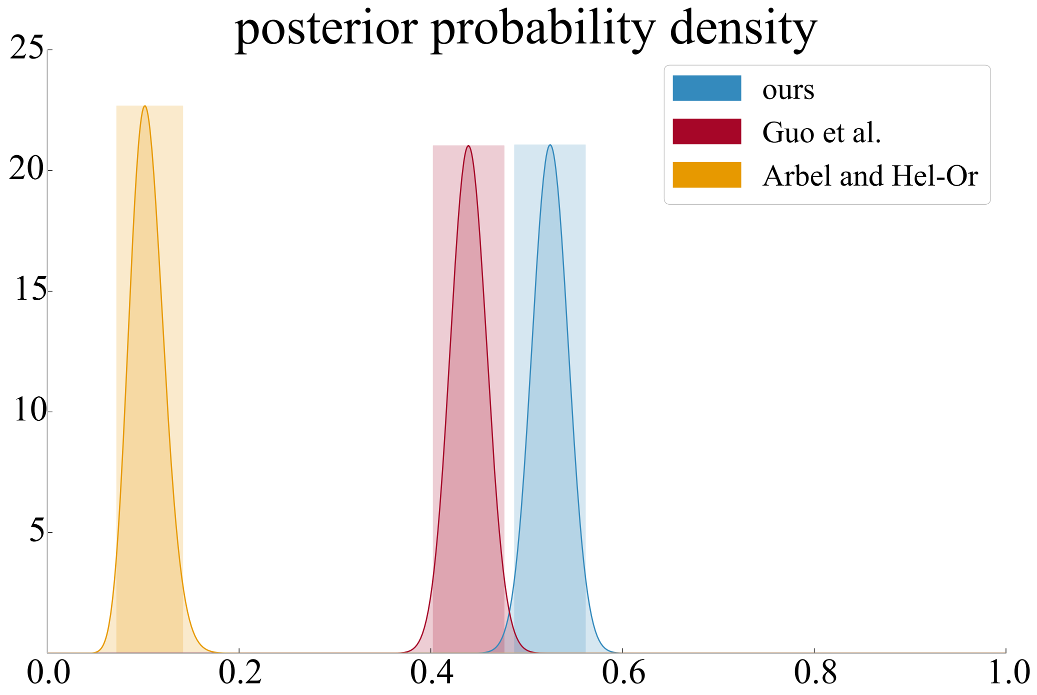
The study was deployed on a website. We recruited 51 participants through email lists, word-of-mouth, and Facebook. Individuals could participate by visiting a link using their own computing device. Of the 51 participants, 39 completed the whole experiment, 7 quit before finishing the first phase, and 5 quit during phase two. Because of the randomized design of the study, we include all participant data, including data from participants who did not complete the entire study. Additionally, 28 people visited the experiment, but did not answer any questions.

We analyze our study data using methods described in [Kru11]. Unless otherwise stated, reported results represent the posterior mean, and the confidence interval (CI) represents the range encompassing 95% of the posterior probability.

Participants rated a total of 694 image tuples in the ranking phase and analyzed 605 images in the evaluation phase. In the ranking phase, we calculate the posterior probability of each method being ranked first (*i.e.* appearing the most natural). We model this as a Bernoulli random variable with a uniform Beta prior. As shown in Figure 3.17, results produced by our method were significantly more likely to be ranked first than the competing methods.

In the second phase, participants ranked the success of shadow removal with a score $\{1, 4\}$ for each image. Figure 3.18 shows the normalized histograms of scores assigned to results produced with each of the three methods. As can be seen, our method obtained high scores in the evaluation task more often than the other methods. Additionally, we have evaluated how likely each method was to unshadow images perfectly (we define "perfect" shadow removal as one with mean evaluation score across all participants $\mu_{eval} > 3.5$). Figure 3.19 (left) shows the posterior probabilities for each method to produce a perfect result (as before we have modeled this using a Bernoulli random variable with a uniform Beta prior). The results show that our algorithm is significantly more likely than others to succeed in this scenario.

We have also characterized the results by considering the data from both user study phases together. The right part of Figure 3.19 shows the probability of a given image winning both the ranking phase and



| Arbel and Hel-Or | Guo *et al.* | Our technique |
|---|---|---|
| 10% (7-14% CI) | 43% (40-48% CI) | 52% (49-56% CI) |

**Figure 3.17:** *Posterior probabilities of each method winning a ranking round. The shaded, rectangular regions signify the 95% Confidence Interval (CI).*

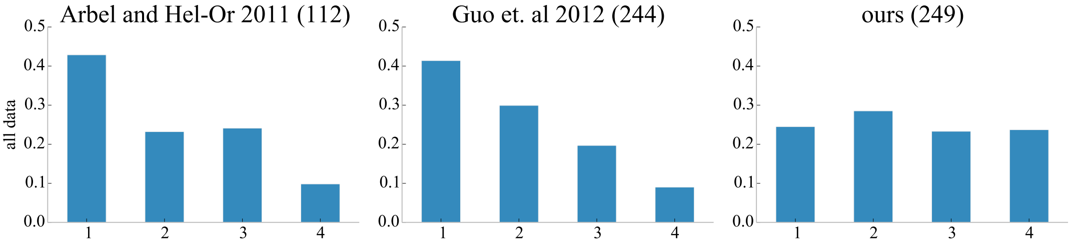


**Figure 3.18:** *Normalized histograms of image evaluation scores for different methods. Higher scores correspond to images where the shadow was removed successfully and no artifacts were introduced, while low scores mean that the shadow removal failed and/or there were visible artifacts introduced. Numbers in brackets above each plot show how many evaluations contributed to it. Overall, our method has the highest chance of obtaining high scores and therefore removing shadows successfully.*

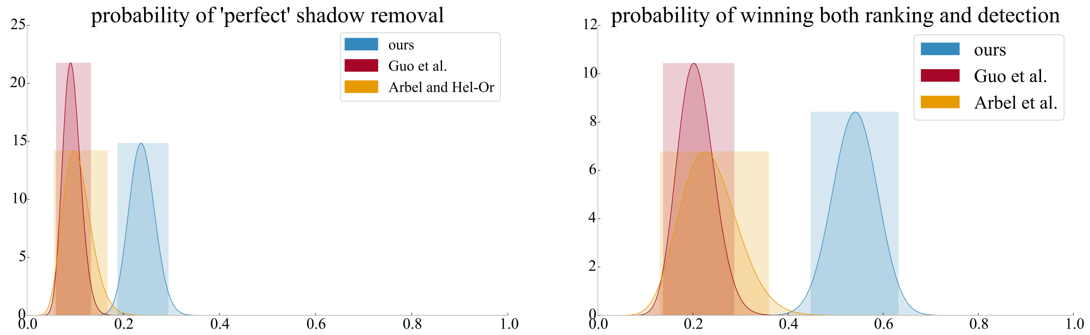


**Figure 3.19:** *Left: posterior probability of image having a shadow removed perfectly (mean score $\mu_{eval} > 3.5$). Right: posterior probability of image modified with a given method winning both in the ranking and evaluation phases.*

the evaluation phase.

Additionally, in Figure 3.20 we show the probability of our method or Guo *et al.*'s method winning the ranking task while simultaneously having different evaluation scores. We show that when images unshadowed by Guo *et al.*'s method win, they are likely to have low scores, while our method winning likely means high scores. This can be explained by the fact that in their case, the method sometimes "misses" softer shadows and does not modify them at all. In these cases, the image is likely to rank high on the naturalness scale, but still fail the shadow removal evaluation.

Closer inspection of this combined test set revealed that the mean maximum penumbra width of images from Guo *et al.* is 32 pixels, while for the test images we have introduced it is 55, as shown in Section 3.6.5. We have therefore analyzed how the performance of different methods varies on different subsets of the data. As shown in Figure 3.22 in the case of testing on Guo's data only, no significant difference between our and Guo *et al.*'s method was observed (while the maximum a posteriori (MAP) estimate of our method is lower, the confidence intervals overlap significantly). On the other hand, our method performs much better on the dataset with higher mean penumbra width (*i.e.* softer shadows).

Figure 3.23 shows similar trends as in the ranking case: when using the dataset with moderately soft shadows our method is indistinguishable from Guo *et al.*'s, but as the penumbra size increases our performance becomes relatively higher.

Further, we have measured the standard deviations of scores that the study participants assigned to individual images. As Figure 3.24 illustrates, different participants were mostly consistent in score assignments.

Finally, we have conducted a separate user study comparing our method to that of [MTC07]. We found that participants preferred our results 65% of the time when shown against Mohan *et al.*'s, and were significantly more likely to highlight artifacts in their results than in ours.

## 3.7 Limitations

Though our method is somewhat robust to inpainting errors, it is often unable to recover when the initialization fails significantly. More explration into structure-guided inpainting methods, similar to the

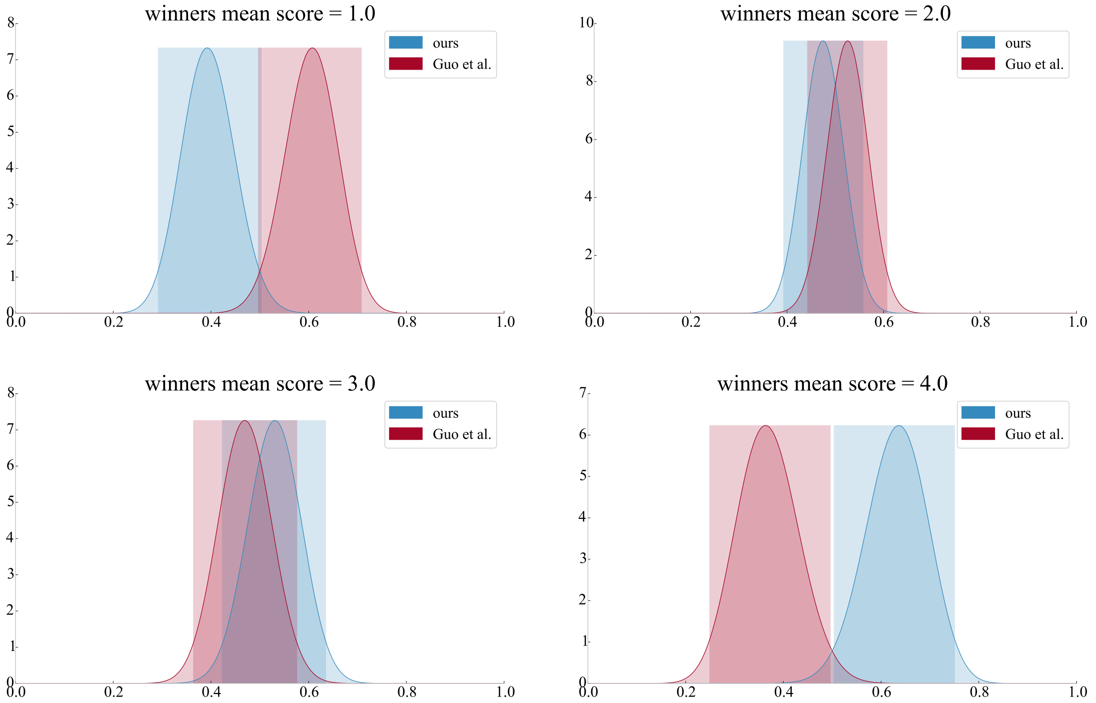


**Figure 3.20:** *Probability of winning the ranking task conditioned on the winner's mean score in the evaluation task. Note that when Guo* et al. *win, their scores are likely to be low, while the opposite is true for our method.*

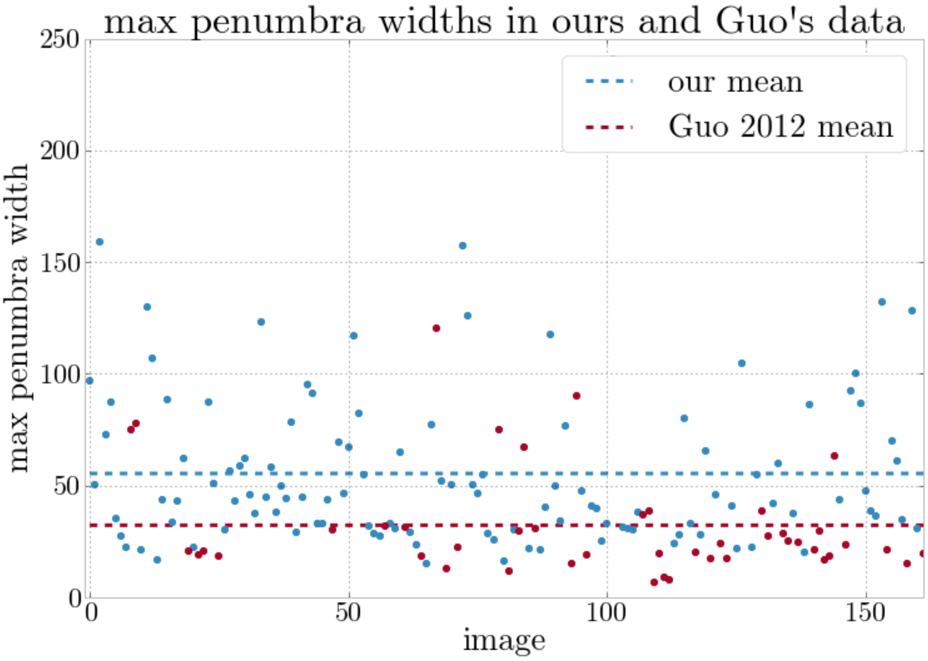


**Figure 3.21:** *Maximum penumbra widths for images in ours and Guo* et al.*'s datasets. Note that the average penumbra in our dataset is significantly wider than in Guo* et al.*'s (meaning softer shadows).*

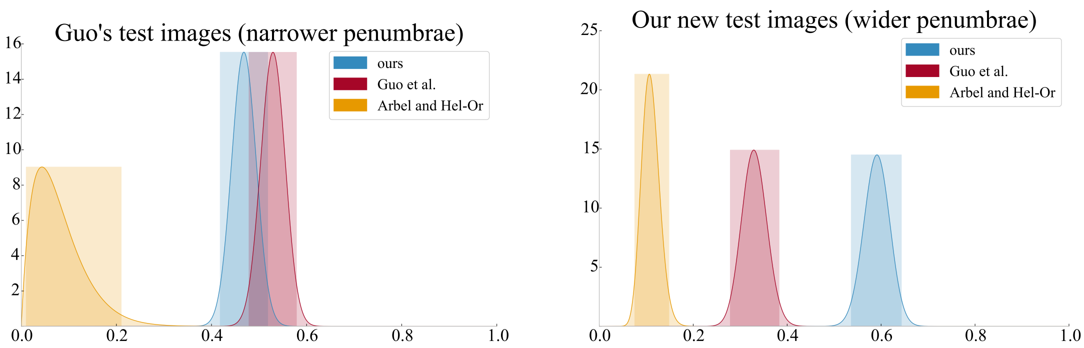


**Figure 3.22:** *The different performance characteristics on different slices through the dataset seem to be correlated with the softness of the shadows: our technique has the biggest advantage on images with softer shadows.*

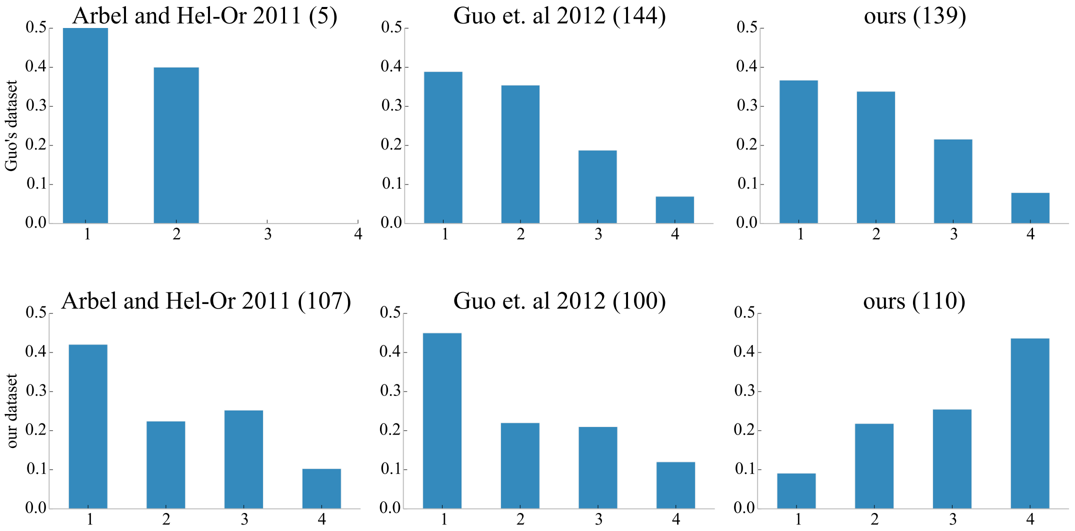


**Figure 3.23:** *Histograms of evaluation scores conditioned on the dataset used. Our method is either significantly better or comparable to the competition.*

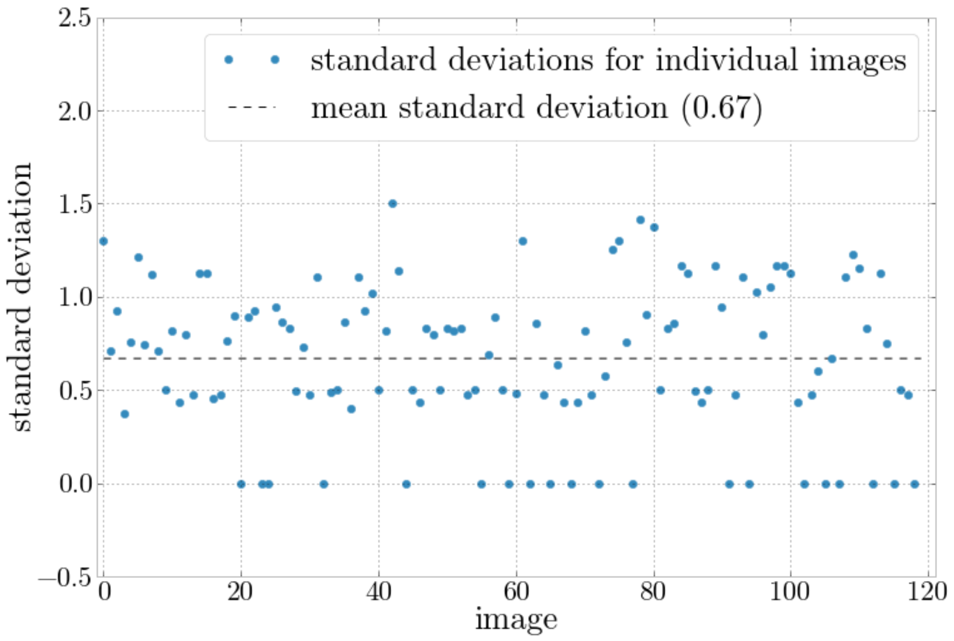


**Figure 3.24:** *The users seemed to mostly agree in their evaluations of the results.*

one we tested in Section 3.3.1 could produce better results.

Another limitation is that the technique is not able to deal with most hard shadows. An interesting area of future work would be to either extend it to perform better in this regard, or combine it with one of the hard-shadow specific methods and automatically choose the correct implementation.

Additionally, since our training set does not contain scenes with mutliple light sources casting overlapping shadows, we do not expect to handle all such cases correctly (however, some complex shadows such as ones cast by leaves, are removed successfully, despite not being represented in the training set, *e.g.* “real276” in the supplementary material). Further, the colour optimization step makes assumptions, which do not hold in the presence of multi-coloured light sources (*e.g.* stained glass or mutliple light sources of different colours), or when the in-shadow region is not similar to any out-of-shadow parts. Specifically, we allow for the illumination colour balance to change (interpolate) between in-shadow and out-of-shadow regions, which is enough for most scenarios, such as sun-and-sky illumination. It does not allow, however, to capture the shadow profiles in the presence of more than two distinct, differently-coloured light sources (since then one has to interpolate between more than two target illumination colours). Regressing each channel separately would remove this limitation at the cost of intriducing slight chromatic artifacts, as demonstrated in Section 3.4.

While we have not tested it, this technique should also work for shadows cast by translucent objects, since the colour optimization aims to make the unshadowed region similiar (colour-balance-wise) to the out-of-shadow parts. To handle such situations, we would need to relax our hard constrains on assumed shadow matte channel scaling, which were put in place to speed up computation in the most common cases. On the other hand, we would not expect our algorithm to be effective in removing attached (as opposed to cast) shadows, unless a different training set was used. Since the fall-off profiles of attached shadows are dictated by the surface shapes, they do not, in general, exhibit the same behaviour as the penumbrae of cast shadows.

Finally, in some cases the inference stage is unable to produce compatible matte suggestions at neighbouring regions in the image, which results in visible grid-like artifacts (see *e.g.* the top-right image for Figure 3.25). While the effects of this limitation are often hidden by the texture underneath the shadow, a possible solution could be to increase the feature patch overlap or to create a stronger pairwise distance constraint. Both of these solutions are challenging, however, as they require more training data and therefore computational effort.

## 3.8 Discussion and Future Work

We have presented a model for removing soft shadows that is not based on heuristics, but instead draws from the experience of the graphics community to learn the relationship between shadowed images and their mattes, from synthetically generated, but realistic, data. Our approach can deal with soft and complex shadows, and produces results faster than the most related techniques in the literature. It requires little time and input from the end-user, and our study showed that our method is significantly better than existing methods in successfully removing soft shadows.

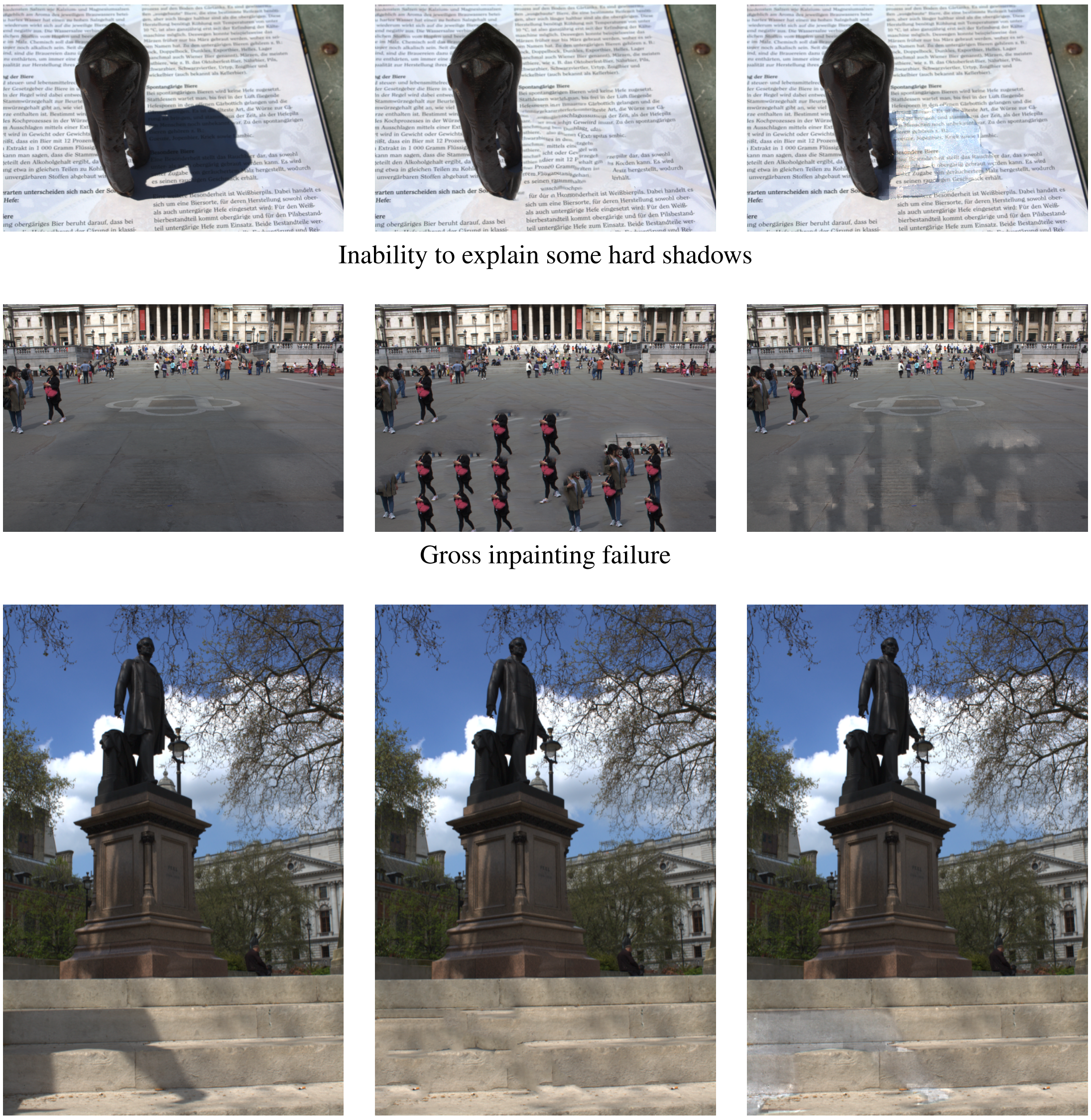
More generally, we have presented a unique use of Regression Random Forests for supervised clustering of high-dimensional data, coupled with a regularization step that is adaptable to general scenarios. Similar ideas could be applied to video *e.g.* by performing regularization across frames.

There are several ways this technique could be extended in future work. One of the most obvious additions could be some understanding of the scene and its contents. With more information about *e.g.* normals and depth discontinuities, our technique might be able to better identify and composite shadows. This information could also feed into the following bootstrapping extension to sample the unshadowed texture more efficiently.

Another interesting problem would be the creation of guided inpainting systems that could be used for initializing our method. For example, a method similar to [HSGL11] could help find correct out-of-shadow correspondences, while more user input could provide constraints for the initialization (*e.g.* structure cues as in [SYJS05]). As better guided inpainting algorithms emerge, our framework will be increasingly effective.

**Possible extension** As mentioned previously, inpainting algorithms can be divided into those that use a pre-built dataset and those that use the remainder of the image being modified. Similarly, some super-resolution techniques (*e.g.* [GBI09]) use parts of the image to be modified as exemplars for synthesis. Using the same reasoning, we can adapt our method so that it bootstraps the training set from the input image. For this variant, we prepared a set of prerendered shadow mattes and applied a random subset of them to different positions in the shadow-free areas of the input image. This results in pairs of [shadowed, shadow-free] images that we use to train the forest, which is then used for inference in the same process as previously.

The advantage of this extension is that it builds a finely-tuned regressor for this particular image

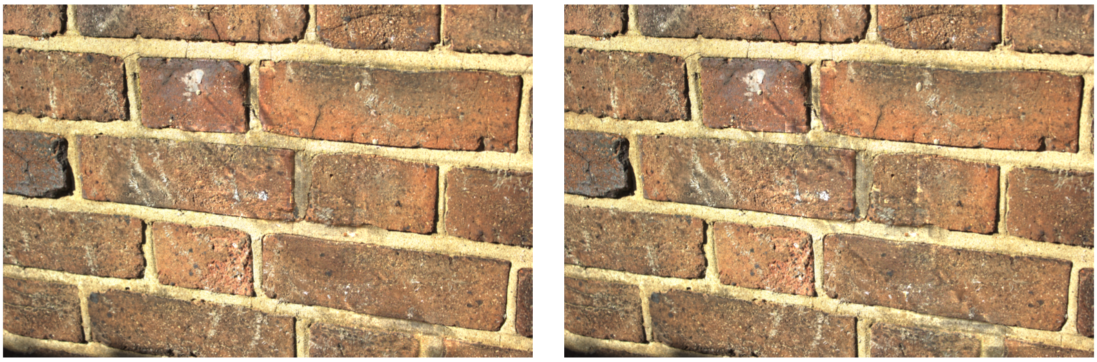


Inability to explain some hard shadows

Gross inpainting failure

Incorrect colour adjustment (see video)

**Figure 3.25:** *The left column shows input images, the inpainted initializations are in the center, and the outputs can be seen on the right. Please note that in the case of incorrect colour optimization, the user can easily rectify any mistakes by using our simple shadow editing interface as shown in the supplementary video.*



Standard

Bootstrapping

**Figure 3.26:** *Comparison with bootstrapping extension. While for some images bootstrapping allows us to obtain comparable results with a much smaller training set (the image on the right used a training set of $J = 50$ images) it makes much stronger assumptions and is therefore not as generally applicable.*

which yields high performance given a smaller training set. On the other hand, it is critically reliant on the assumption that the image has enough out-of-shadow areas with similar texture as the shadowed parts, which limits the number of images suitable for this method. Nevertheless, in the right circumstances this method can produce good results—see Figure 3.26. More work in this area could lead to more robust solutions.

Further, it might be possible to automatically detect hard and soft shadows in the given image and to selectively apply our method for soft shadows only, and a hard shadow-specific method otherwise.

Additionally, techniques for detecting forgeries, such as [KOF13], may gain more power given the ability to explain soft shadows. Finally, works such as [SPDF13], addressing the problem of image relighting from a single photograph belong to an exciting area that could benefit from the ability to seamlessly remove and modify shadows.

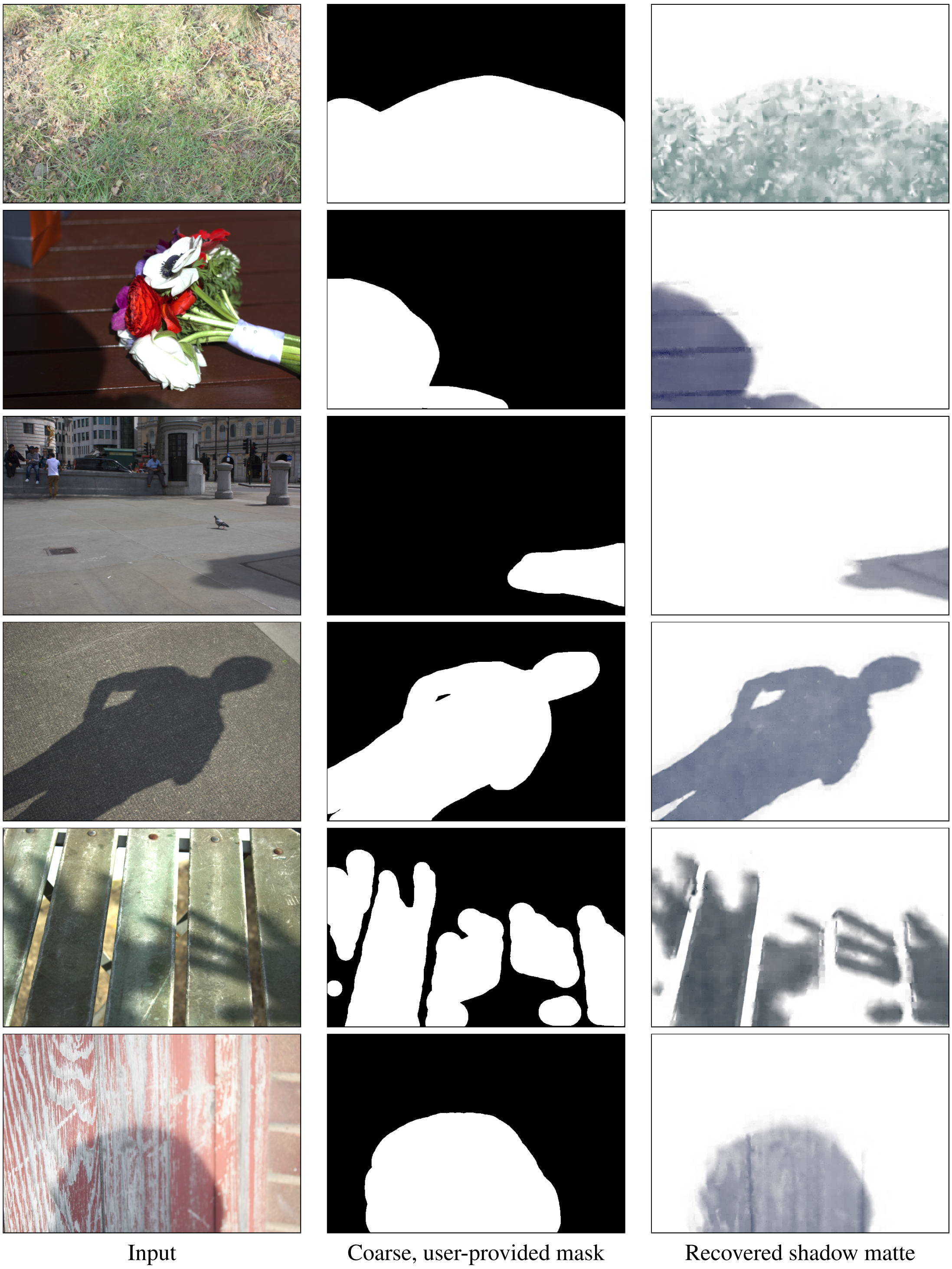


**Figure 3.27:** *Results of shadow removal using our method on a variety of real photographs. The left column shows original images, the middle column shows user input, and the right column shows the obtained shadow mattes (the resulting unshadowed images are presented in Figure 3.1). The mattes could also be used to modify the properties of the shadows as we show in Section 3.6.1.*

# Chapter 4

# Image Evaluation User Studies

The graphics community presented many impressive technical contributions over the recent years. Despite that, high-quality perceptual user studies are still difficult and expensive to conduct. Symptoms of this include small numbers of participants, poor generalizability of studies, non-consistent analyses and lack of thorough examination of the results. In this chapter we examine how big the problem is, why it is important to address and present our attempt at resolving it.

In short, we had created an authoring system that makes it easy to create image evaluation user studies with consistently high quality and requiring relatively little effort. First, however, we show that the problem is already visible to influential researchers and explain why and when user studies are useful. Next, we enumerate desirable features a user study should have and contrast this with what is usually done in the graphics community to evaluate image manipulation techniques. Based on this, we propose a set of tasks that fit the use cases described and can serve as a template for creating user studies, describe each task in detail, discuss when each is useful, how to carry them out, and how to analyze and draw conclusions from their results. Finally, equipped with all the above information, we present our user study creation system that facilitates evaluation of image manipulation techniques.

## 4.1 Motivation: The Need for User Studies

A big portion of the work done in the graphics community aims be perceptually convincing: inpainting ([BSCB00], [CPT03], [LZW03], [BSFG09], [PKVP09], [KKDK12]), shadow removal ([MTC07], [WTBS07], [FDL09], [AHO11]), automatic image enhancement ([CKK11], [HKK12], [KLW12], [YS12]) and rendering are just some examples, where the success is defined by perceptual criteria. Clearly we have a need for a reliable way to evaluate perception-based characteristics, for which numerical measures are not a good approximation. While there is a body of work examining the relationship between perception and numerical measures (for instance Hwang *et al.* [HKK12] show that in their studies, L2 distance had poor correlation with perceptual differences expressed by the users) and even numerical measures were created specifically to reflect perceptually-based criteria ([RFS+98]) there are many dimensions, for which no numerical metrics exist (*e.g*. “scaryness” or “funiness” of an image), but which can be easily evaluated by people.

In the academia, as well as in the industry, running user studies and asking humans to answer questions or perform specific tasks is so far the only reliable way to carry out such an evaluation. We believe that making the creation and refinement of perceptual user studies easier will contribute greatly to the quality and depth of evaluation of a large part of research in our area. To establish, whether this opinion is held by other researchers, we have created a short questionnaire to find out what is the current state of perceptual user studies in the graphics community. The questionnaire was distributed to the SIGGRAPH Program Committee members during the meeting in Boston in March 2014. The summarized results can be found in Appendix C[1], while below we present a summary of the findings. Our aim was to find out:

- Whether user studies are considered an important part of graphics papers.
- What kinds of publications can benefit most from having a user study.
- What other methods can be used to convince reviewers of the merit of a new method.
- How the analysis of the results should be carried out, given the choice between Null Hypothesis Significance Testing (NHST) and Bayesian methods.

Overall, 27 Program Committee members answered the questionnaire. To obtain a rough characterization of the audience, we have asked how many years they have been reviewing graphics papers (average of 11 years with a standard deviation of 4.66) and how many times they have been on SIGGRAPH or Eurographics Program Committee (average of 4.96 times with a standard deviation of 3.57).

Next, we have asked whether there are circumstances, under which presence or absence of a perceptual user study can decide about the acceptance or rejection of a paper. Out of 26 answers, 23 were “yes” and 3 were “no”. When asked to explain the circumstances in more detail, the respondents indicated that user studies are especially important when

- the paper presents visual, perceptual results,
- there are no obvious objective measuerments,
- there is an aesthetic aspect to the task, or
- users are in the loop.

Further, we have asked the respondents to assess the current state of user studies in the graphics publications. Out of 22 answers, nobody concluded that the studies are generally “well carried out”, 8 people said that they are “sufficient”, 13 chose “not thorough enough” and one person indicated that they couldn’t remember any publication, which included a user study.

We had also included a short description of a sample experiment taken from a recent SIGGRAPH paper as reported by the authors. In addition to the original analysis (based on Null Hypothesis Testing) we had analyzed the results using a Bayesian method and asked the participants to indicate which analysis method they prefer and why. Out of 18 answers the majority (10 people) indicated that they do not

[1] raw responses are also available under http://www0.cs.ucl.ac.uk/staff/M.Gryka/download/SIGGRAPH-PC-UserStudy-Questionnaire.pdf

mind which method is chosen, as long as it is carried out well. Of the rest, 4 chose the classical analysis and 4 chose Bayesian. The reasons for classical were mostly connected to it being more familiar, while the proponents of the Bayesian method felt they were more honest and thorough. Additionally, a flaw in the classical analysis (*i.e.* the one copied from the original paper) was pointed out by two participants: the effect sizes were missing. While this means that the questionnaire results might be biased (since one of the analyses was less than perfect), it highlights the fact that analyzing user study results correctly is something the graphics community struggles with.

Finally, we had asked our respondents about any other methods, besides perceptual user studies, that they feel are useful when presenting new graphics research that cannot be easily examined using objective measures alone. The most prominent were visual comparisons by reviewers themselves, abundance of results, comparisons to previous work and to ground truth, and perceptual quality metrics. One person additionally suggested that opinions of experts might be convincing.

In conclusion, it seems that influential reviewers of graphics papers feel that user studies are important when presenting perceptual results and that our community, currently, has difficulties ensuring these studies are carried out and analyzed well. We believe that the authoring system presented in this chapter can solve these problems, while also making it easier and faster to draw conclusions about new methods.

Further, enabling a cheap, but reliable way of creating and running user studies as well as automating the analysis opens up possible areas of further research. Iteratively optimizing image manipulation methods and incorporating user feedback as a parameter is just one such possibility.

## 4.2 Current Trends in Evaluation of Image Manipulation Methods

Armed with the knowledge of what the community members think about the current state of user studies in graphics papers, we now present a short survey of several relevant papers to see concrete examples. This section aims to show what the people in our field are already doing and will examine the following characteristics of each study:

- What was the aim of the study?
- What was the task that the participants were asked to perform?
- Was there a a pilot study to tune the study setup?
- How many participants were recruited and how? Were they from a specific demographic?
- How was the data analyzed?

We have created a collection of papers published in SIGGRAPH and Transactions on Graphics over the last 5 years. The set was collected by first searching paper abstracts for relevant keywords, adding the articles we have already known about and then following citation trail. While this is not an exhaustive list, we aim to show that there is a significant subset of works that share important characteristics of how they evaluate results. In addition to the summaries below, see Chapter 5, where we have recreated experiments from four more recent papers.

- Cole *et al*. [CSD+09] investigated how well line drawings facilitate understanding of shapes of 3D objects. To gather the data, they have conducted an interactive study on Mechanical Turk, where they were asking participants to interactively orient gauges in 3D based on the underlying shapes represented as 2D images. Regardless of the interface, their detection and treatment of negligent (or malicious) respondents was interesting. In short, in the absence of hard distinctions between right and wrong answers, they performed consistency checks and rejected data that fell below a consistency threshold (which was determined during a pilot study). The authors recruited 560 participants from Mechanical Turk. The analysis was specific to the task at hand: the authors reported simple statistics of angular errors for different 2D representations, scatter plots for a more qualitative comparison and Null Hypothesis Significance Testing-based analysis to establish when two conditions differed significantly.

- An exhaustive set of experiments was conducted by Cadik *et al*. [ČWNA08] to compare different HDR tone mapping methods. Their experiment had two phases: scoring and ranking. In the scoring phase the participants were asked to score how well given image represented a real-world scene. In contrast, during the ranking phase the task was to rank multiple images based on defined image quality criteria. The pilot study was performed before the full experiment to test the setup. Overall 20 individuals were recruited for the study, but the recruitment method was not mentioned. The results of the experiments were very thoroughly investigated using null hypothesis significance testing (NHST).

- Rubinstein *et al*. [RGSS10] conducted a large-scale survey to thoroughly evaluate 8 different image retargeting methods. They have recruited 420 participants for their experiments, of which some were volunteers and some recruited through Mechanical Turk. They have also asked the subjects about their gender, age and familiarity with computer graphics and found that around 40% of participants were female and around 60% male, the average reported age was 30 and that their graphics expertise varied. Two types of paired comparison tasks were presented: one with the reference image visible and one without. Before the main experiment a pilot study was ran to assign labels to the evaluated images based on their contents (the labels were “lines/edges”, “faces/people”, “texture”, “foreground objects”, “geometric structures” and “symmetry”). After collecting the data, the methods were ranked based on the frequency they were preferred over alternatives and the statistical significance demonstrated using NHST.

- A paper by Kang *et al*. [KKL10] used pairwise comparisons (a non-forced-choice variant, *i.e*. also allowing people to answer “no preference”) to evaluate the effectiveness of their personalized image enhancement technique. The authors recruited 14 friends and colleagues to participate in the study and no pilot experiments were reported. NHST was used for analysis.

- Another example of using ranking experiments, was the work by Kong *et al*. [KHA10], where they used a series of Mechanical Turk tasks to create a set of best practices for creating perceptually-based treemaps. Over multiple experiments (including a pilot study) the authors ecruited over 200

Mechanical Turk workers. Again NHST was used during analysis.

- Caicedo *et al.* [CKK11] hired 336 respondents on Mechanical Turk to find clusters in personalized image preference space. They have created a simple, interactive interface allowing image modifications and asked people to modify a set of photographs to their liking.

- A large-scale experiment, including a scoring-based task was described in the work by O'Donovan *et al.* [OAH11], where colour themes were evaluated based on their subjective attractiveness. Overall 1301 participants were polled for their colour theme preferences. The data from the experiment was used to train a regressor to rate the colour themes and a classifier to distibguish between "good" and "bad" ones.

- Hwang *et al.* [HKK12] used non-forced-choice, pairwise ranking to evaluate different image enhancement methods. No information about the pilot study, number of participants or recruitment methods was given and NHST was used for analysis.

- Yuan and Sun [YS12] created an automatic exposure adjustment algorithm and demonstrated its performance using pairwise comparison (*i.e.* binary ranking) experiments. For the evaluation, 12 volunteers with different expertise levels in photography were recruited. Their task was to choose the better image out of a pair (or indicate a tie), where each was processed with a different method. No pilot study or statistical analysis was reported, besides proportions of preferences of different methods together with corresponding standard deviations.

- Similarly, non-forced-choice pairwise comparisons were used by Kaufman *et al.* [KLW12]. To gather data about their method, two experiments were carried out: one with 71 students (21 women and 50 men) and one with 22; it is unclear whether there was overlap in participants between the two runs. Each time, the subjects were asked to complete the study online, using their own devices. No data about the pilot study was reported and the authors reported only the percentages of times each method was preferred.

- Kopf *et al.* [KKDK12] evaluated their optimized cropping algorithm for inpainted panoramas using a pairwise comparison task. The study aimed to establish, which method generated more appealing panorama images. Thirteen (13) participants took part in two experiments, but no pilot was reported. Interestingly the authors used Mechanical Turk to gather opinions from a larger population (117 individuals), however, this data was used for the purpose of optimizing parameters for their algorithm. NHST was used to establish the statistical significance of collected data.

Based on the above data, we have decided to refine and implement two task types: ranking and scoring. While ranking can be used to directly compare different results and express relative preferences, scoring tasks allow assigning absolute values (on the author-defined spectrum) to single images. These two modalities cover a large part of what graphics researchers need to evaluate their methods. Even though we mostly use binary ranking and therefore the term "paired comparison" might be more accurate, we have decided to keep the more general description.

Before specifying the tasks in more detail, however, we also briefly look at how user studies in general should be built.

## 4.3 Desirable Characteristics of a User Study

In the general case, user studies vary: they can be employed to evaluate new interfaces, copy writing, photographs, gather opinions about current events or products *etc*. Consequently, different constraints apply in each scenario and different rules should be followed to make the study useful. However, as we have seen in the previous section, focusing on image evaluation narrows the scope and leads to many commonalities between different experiment. Additionally, there are a few general rules that should apply to any high-quality perceptual experiment. Specifically, when designing a user study the following goals should be kept in mind:

1. **Accuracy and Objectivity:** the study should be as accurate as possible and avoid bias; it should aim to find the true properties of the studied phenomena. In the case of image evaluation, the study should accurately answer questions such as “Which image manipulation method is superior under given circumstances?” or “How does algorithm A perform in comparison to algorithm B?”. One important tool that allows drawing objective conclusions from the collected data is statistical analysis. While traditionally it was often carried out using null hypothesis significance testing [Fis66], recently this approach received significant criticisms ([Kru11]) and Bayesian methods with confidence estimates have been proposed as an alternative. See Section 4.4.2 and Section 4.5.1 for more details on how we tackle this problem.

2. **Generalizability:** the results should be representative of the general population, unless a specific target audience is defined. As an example, in the case of image manipulation, the results evaluated should ideally be equally convincing to experts as well as novices. It is worth noting that this point was often a weakness of studies in our community, where the participants are recruited from the easiest to reach, but a highly biased population: colleagues and computer science students. A partial solution to this problem might be recruiting comparatively large numbers of microtask workers *e.g.* on Mechanical Turk.

3. **Repeatability:** given the data and the procedure the study should be repeatable, *i.e.* the results should be easily reproduced by others. This is important not only because of “trust, but verify” phenomenon, but also because it should be easy for new research to build upon earlier findings. The answer we present below is twofold: firstly we define two common tasks, how they can be carried out and analyzed. Secondly, we present our implementation of a system that includes these tasks as well as the ability to export and import the exact setups.

4. **Creation time and cost / efficiency:** traditionally running user studies implies some form of compensation to the participants. Ideally a study would use the resources efficiently and not have other significant overhead; this implies making the most of the feedback participants provide and ensuring that their tasks are simple and well-defined. Additionally, since the focus of graphics

researchers is creating interesting algorithms, the preparation of the study should be convenient for the researcher and take minimal time and effort.

Particularly the last point is a key obstacle leading researchers in our community to suboptimal treatment of studies. Their creation and thorough evaluation traditionally take large amounts of effort and often occur shortly before submission deadlines. Alleviating this problem is a major motivation for our efforts and, consequently, our main measure of success. We have early indications, summarized in Section 4.8, that we have taken a correct approach to maximizing the quality/effort ratio. However, the real proof can only be apparent after more researchers use, and provide feedback on, our methods.

## 4.4 Ranking Task

Many novel algorithms for editing images aim to be perceptually convincing and have to compare themselves to previous attempts to establish their performance. In such cases a user study with a ranking task can provide indication about people's preferences between images produced with different methods.

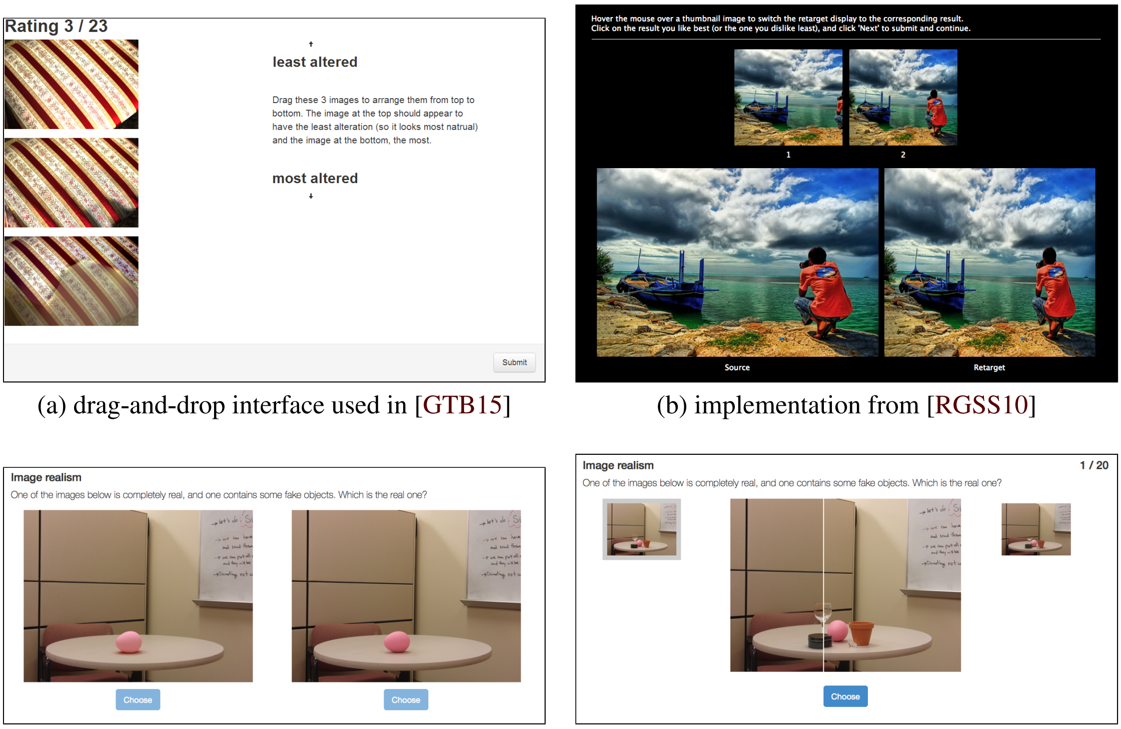
While a ranking task in general can mean "order M images based on criterion X", we mostly focus on the simplified case, where $M = 2$. This simplification can be used even when more than two methods have to be compared, by selecting 2 out of $M$ images for each round and always presenting pairwise choices. This setup has a number of advantages: most importantly it makes the task simpler for the participant, enabling a faster and more efficient data gathering process. In most of our experiments people took less than 10s to pick a better one of two images, while Cadik *et al.* [ČWNA08] report participants taking 35m to order 14 (printed) images. Secondly, when presenting the stimulus digitally it is often impractical to have more than two images on a screen at one time without losing valuable detail. Finally, analysis of the data is made easier, especially when different numbers of results are available from different methods.

It is possible to tackle some of these problems and Berthouzoz *et al.* [BLDA11] proposed an interesting alternative where users were presented with 4 images, A, B, C and D and were asked to specify the distance from each of B, C and D to image A on a scale from 1 to 5. It is unclear, however, whether this particular experiment design is superior to other methods and it still does not scale to large values of $M$.

Some examples of how the ranking task can look are found in Figure 4.1. There are a few parameters that can be adjusted depending on the scenario. For instance the user can be put under different constraints such as limited time (either minimum or maximum), ability to revise decisions *etc*. Further, there are different interfaces for performing the selection: drag-and-drop, buttons, sliders, or thumbnails with hover events as used in [RGSS10].

### 4.4.1 Interface Variations

We have implemented five different ranking interfaces to ensure a wide variety of modalities are available for study authors. While we have developed each of them based on some assumptions for when they can be useful, we have not yet gathered enough data to establish the validity of our assumptions. Each interface can optionally also show a "reference image" if needed. Finally, variety of other parameters,



Rating 3 / 23

↑ least altered

Drag these 3 images to arrange them from top to bottom. The image at the top should appear to have the least alteration (so it looks most natrual) and the image at the bottom, the most.

most altered ↓

(a) drag-and-drop interface used in [GTB15]

Hover the mouse over a thumbnail image to switch the retarget display to the corresponding result.
Click on the result you like best (or the one you dislike least), and click 'Next' to submit and continue.

(b) implementation from [RGSS10]

Image realism

One of the images below is completely real, and one contains some fake objects. Which is the real one?

(c) side-by-side, non-forced-choice interface

Image realism 1 / 20

One of the images below is completely real, and one contains some fake objects. Which is the real one?

(d) "curtain" interface

**Figure 4.1:** *Different interface implementations for a binary ranking (paired comparison) task. Note that (a) can also be used for ranking more than two variations, but its effectiveness is then reduced because of small image sizes.*

such as minimum and maximum time constraints, text displayed, layout and presentation options can be adjusted per-task by study authors. The interface variations we have implemented include:

- Horizontal side-by-side display, where two images are displayed next to each other together with buttons to make a choice; see Figure 4.1 (c).
- Vertical side-by-side display, similar to above, but with images and their corresponding buttons arranged vertically on the left and the textual description on the right.
- A "curtain" interface, allowing the participants to interactively reveal either of the two images using the mouse cursor; see Figure 4.1 (d).
- A "swap on keypress" interface, similar to "curtain", but where the images can be alternately displayed by pressing arrows on the keyboard.
- Drag-and-drop ordering, shown in Figure 4.1 (a), similar to the vertical side-by-side variant, but which allows drag-and-drop ordering, rather than button-based choice.

We have based these variations on the two main assumptions. Firstly, we expect that the study authors want to maximize the image size shown, while keeping the entire interface on one screen and avoid the need for scrolling. This is why having vertical and horizontal variants can be useful.

Secondly, based on different magnitudes of differences between evaluated images, some interfaces might seem more suitable. For instance, when the differences between images are large, seeing them side-by-side is often enough to detect the changes. However, in other cases, the differences might only be detectable after participants are allowed to "swap" two images in place.

We believe that gathering of data about which interface variations are suitable for which situations is an important task with the potential to make the creation of accurate perceptual experiments event easier.

### 4.4.2 Analysis

To analyze the results obtained from a ranking task, we have used methods described in [Kru11]. While null hypothesis significance testing (NHST)-based analysis has been historically more common in graphics papers, it has a number of problems that recent works point out ([Wag07], [Kru13], [dWD14]). While both methods are useful when applied correctly, Bayesian approaches have the advantage of directly answering the question posed (*e.g.* estimating the chance of method A being chosen over method B given the data) as well as providing confidence estimates. The potential added difficulty of choosing the correct prior is not an issue in practice for the problems we have looked at.

In 2-way ranking, or pairwise comparison, we aim to infer a Bernoulli distribution that reflects the subjects' responses we have obtained. Bernoulli distribution describes the probability of obtaining one of two possible outcomes and has the form

$$p(y|\theta) = \theta^{y}(1-\theta)^{(1-y)}, \tag{4.1}$$

where $\theta = [0, 1]$ is the parameter of the distribution and $y = \{0, 1\}$ is the outcome.

Following Kruschke [Kru11] we choose to use a uniform beta distribution

$$p(\theta|\alpha,\beta) = beta(\theta|\alpha,\beta) = \frac{1}{B(\alpha,\beta)}\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}, \tag{4.2}$$

with the normalization factor

$$B(\alpha,\beta) = \int_0^1 \theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}d\theta, \tag{4.3}$$

for the prior over the Bernoulli parameter, *i.e.* $beta(1, 1)$. A uniform beta distribution is a reasonable default choice if no additional information is available. On the other hand, when prior knowledge is present, it is possible to use a more informative distribution. The choice of a specific prior depends on individual circumstances of a given study and has to be stated explicitly and be defensible to a sophisticated audience.

The $beta(\alpha, \beta)$ prior can be interpreted as the experimenter explicitly incorporating previous data in which the outcome of "1" was seen $\alpha$ times and "0" $\beta$ times. Most experiments in graphics research claim no such prior knowledge and thus $beta(1, 1)$ is the most suitable choice, since it indicates that each value of $\theta$ is equally likely (Figure 4.2a). On the other hand, if there is a reason to believe that the probabilities of both outcomes are balanced, and therefore that $\theta$ is likely to be close to $0.5$, $beta(3, 3)$ could be used instead (Figure 4.2b).

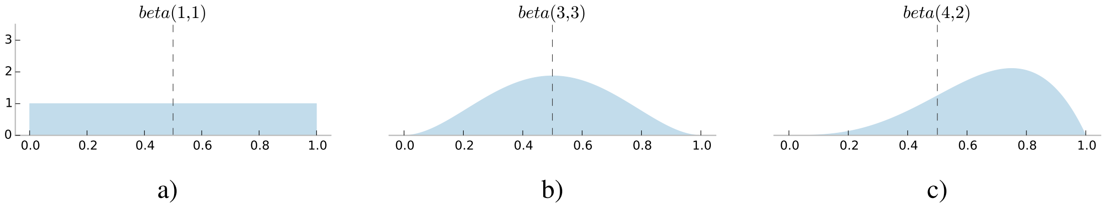


**Figure 4.2:** *Probability density plots of three different beta priors. The priors can be used to formally incorporate existing knowledge into the experiment:* e.g. *if there were reasons to believe that $\theta$ is close to 0.5, that is that the conditions are likely balanced, the prior of $beta(3,3)$ shown in b) would be a reasonable choice. On the other hand, if no prior information is available, the uniform prior $beta(1,1)$ can be used to assign equal probabilities to all values of $\theta$. Finally, if previous data points to a biased outcome, the prior could be skewed accordingly (c)).*

To illustrate the inference method, we present a hypothetical scenario similar to the experiments described in Chapter 5. Let A and B represent two different image processing methods that were used to process 10 images. Having the resulting set of 20 images, we want to discover which method produces more aesthetically appealing results. After asking 100 participants 20 questions each, we found that method A was chosen $c_A = 1040$ times, while method B $c_B = 960$ times.

To establish whether there is a statistically significant difference between the two methods, we use the procedure described by Kruschke: assuming a uniform prior of $beta(1,1)$, and taking advantage of the fact that the beta distribution is conjugate to Bernoulli, we can compute the posterior distribution over the probability of method A being chosen with

$$p(\theta|c_A, c_B) = beta(\theta|1 + c_A, 1 + c_B). \tag{4.4}$$

More precisely we obtain the posterior distribution over the value of the parameter of the Bernoulli distribution describing how often method A is chosen, but in this simple case the numbers translate directly.

We can now plot the posterior density, as shown in Figure 4.3 (blue). Finally, we need to find the extent of the Highest Density Interval (HDI) of the posterior distribution, to see which values fall within the credible region. Conventionally HDI is set to encompass 95% of the probability mass centered around the distribution's expected value. In our visualizations, we signify this by a shaded rectangular region and an annotation in the legend. Since the HDI of the posterior does not include the "null" value of 0.5, we can conclude that method A is chosen more often than method B with a probability higher than chance.

Note that Figure 4.3 contains redundant information: since the probabilities of choosing both methods must sum to 1, knowing the probability of choosing A is enough to infer the corresponding probability of choosing B. However, to make the visualization more obvious we can also calculate and plot the posterior distribution $beta(1 + c_B, 1 + c_A)$, shown in red. Now, instead of checking whether 0.5 falls within the HDI, we can look for HDI overlaps between the two distribution. This approach is especially useful when more than two methods are evaluated, as described below.

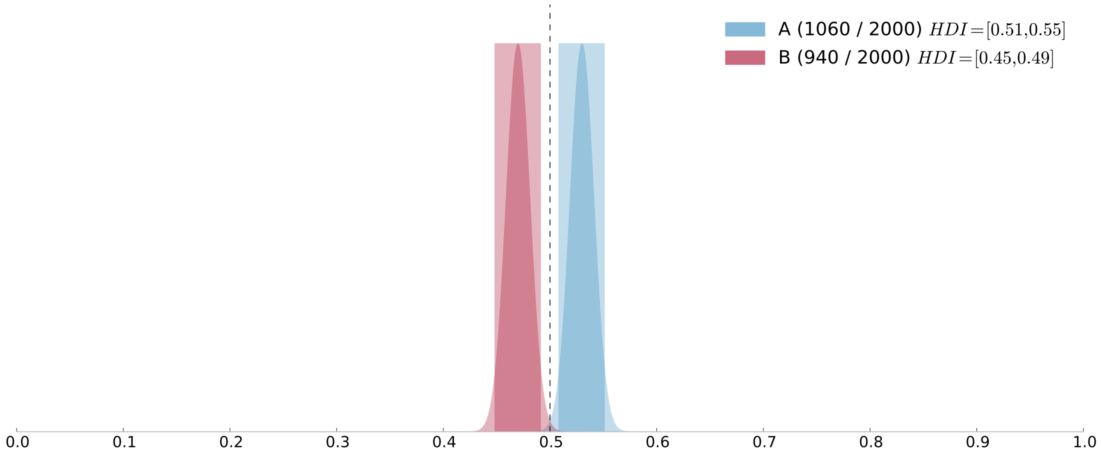


**Figure 4.3:** *Visualization of the posterior distributions estimated from the hypothetical data described above. The legend shows the proportion of trials won over the total trials as well as the HDI boundaries.*

While the above procedure only deals with two conditions (or methods), it can also be applied to more diverse data. In such cases, each method in turn can be evaluated against all other methods. That is, all trials when method A was compared with any other method can be collected and the number of “wins” and “losses” used as input to the analysis.

Similarly to before, let A, B, C and D be four different image manipulation methods. Again, we process the same dataset with each of the methods to obtain a set of 40 images. We have simulated 100 participants, again answering 20 questions each. For each simulated “question”, we randomly chose two methods to compare and made a selection based on a biased random number generator. The data obtained and the corresponding posterior plots are shown in Figure 4.4.

From the plots and HDI values, we can conclude that methods A and B perform equivalently (since their HDIs overlap) and are followed by D and C, which perform differently to A and B, and to each other. It is important to note that in this case, the probabilities of wins for each method do not sum to 1, since we are estimating four separate distributions (even though some of them share the same data).

### Comparing Posterior Distributions

In the sections above, we have presented a way of drawing conclusions from experimental data by comparing posterior probability distributions. One can look at three levels of detail when comparing probability distributions: heuristic-based binary decisions, distance measures and full distributions. Heuristics can produce very concise ways of summarizing results as long as an agreeable criterion is established (*e.g.* null-value falling within the 95% HDI in above sections). On the other hand, full distributions visualized as plots are more verbose, but present the complete, unprocessed output and lend themselves to visual inspection. Finally, multiple distance measures exist to compare the distributions to each other: Chernoff distance, Bhattacharyya distance, and KL-divergence are among the most popular. Each of these methods provides a real-valued output specifying the degree of similarity of two distributions. They are, however, not sufficient by themselves to answer the question “Are these distributions equivalent?”. Prior knowledge or explicit thresholds are needed to make them meaningful.

| method | wins | losses |
|---|---|---|
| A | 605 | 418 |
| B | 621 | 376 |
| C | 316 | 674 |
| D | 458 | 532 |

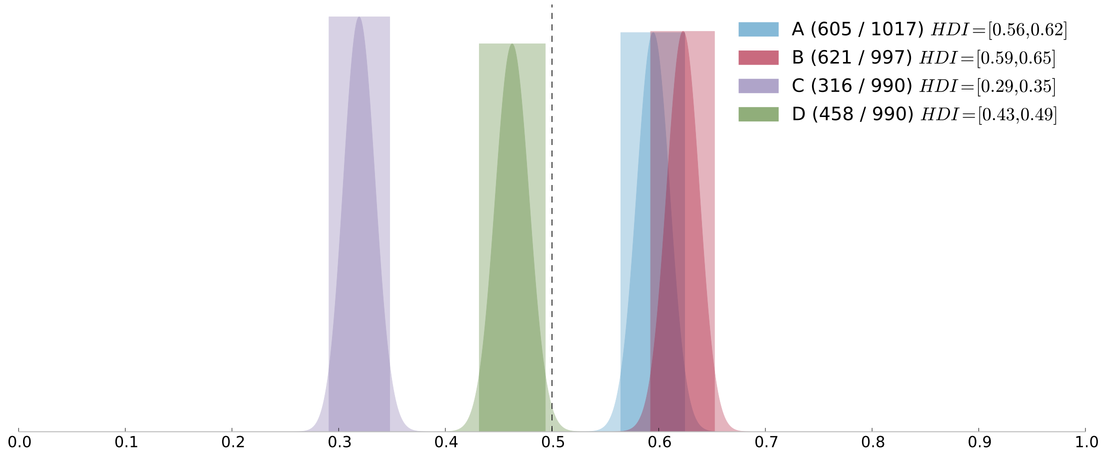


**Figure 4.4:** *Simulated data (top) and four posterior distributions estimated from it (bottom). The legend shows the proportion of trials won over the total trials that each method was evaluated in as well as the HDI boundaries.*

In this work we have decided to use the HDI overlap as the tool for interpreting posteriors. Despite not being strictly “better” than other methods, it has the advantage of intuitive interpretation and conciseness. While our system does not output other distance measures, they can be computed by study designers using raw data provided.

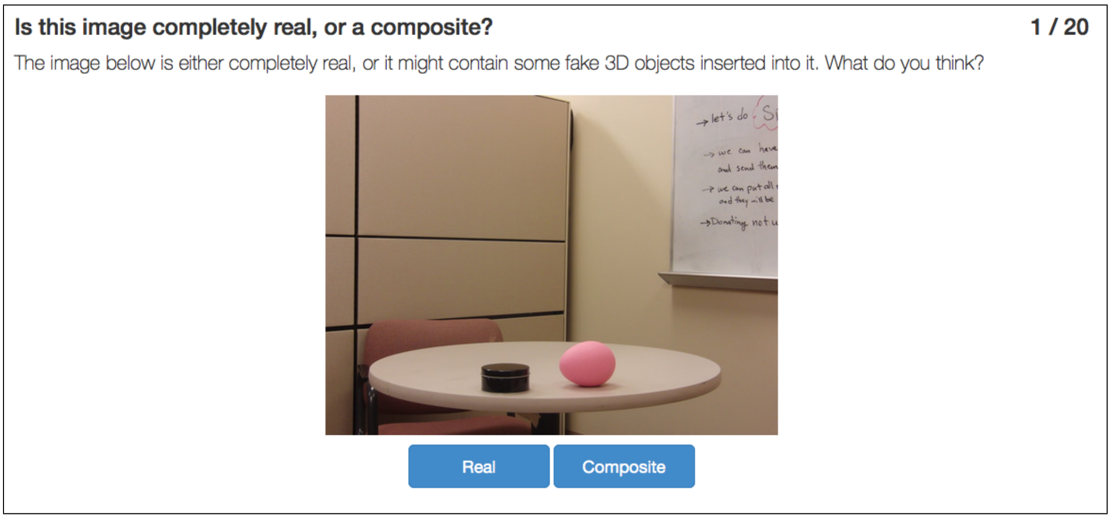
### 4.4.3 Evaluating Confidence

After the results are obtained, it is common for researchers to report confidence in their findings. In the analysis presented above, we do not usually pose the problem in terms of null hypothesis, since it is possible to directly estimate (distributions over) the parameters of the underlying sample distributions. In such situations the width of the posterior distribution, or more precisely, the width of the Highest Density Interval, is used as the confidence measure.

Further, visualizing posterior distributions allows more in-depth understanding of the data: besides the HDI, the researcher can also see the mode of the posterior and its relationship to other distributions, as was shown in Figure 4.4. If the HDIs of two distributions do not overlap, it can be concluded that the intervals space of credible parameters does not include the null values.

## 4.5 Classification Task

While ranking tasks described above focus on characterizing methods in contrast to each other, sometimes it is also informative to investigate how a given dataset fares on its own given a specific scale. For this purpose, classification tasks can be used, where images are to be assigned a category from a predefined set. Note that, while we use the general term “category”, the specific implementations can differ:



**Is this image completely real, or a composite?** **1 / 20**

The image below is either completely real, or it might contain some fake 3D objects inserted into it. What do you think?

**Figure 4.5:** *Sample image classification task. The number and descriptions of available labels ("Real" and "Composite" above) are adjustable.*

- scoring with "categories" being numbers, *e.g.* 1–5,
- yes/no tasks where users answer a binary-choice question about each image,
- labeling, with different choices being labels defined by the researcher (*e.g.* "sky", "ground" and "buildings").

In contrast to the ranking task, we have only implemented one interface for classification, shown in Figure 4.5

### 4.5.1 Analysis

To analyze the results of the classification task, we generalize the methods used for the ranking experiments above. In ranking, each image was essentially assigned one of two categories ("win" or "lose"). Correspondingly in this case we use a categorical distribution (of which Bernoulli is a special case) for the random variable:

$$p(y = i|\boldsymbol{\theta}) = \theta_i, \tag{4.5}$$

where $\boldsymbol{\theta} = [\theta_1, ..., \theta_k]$, $k$ is the number of categories and $i \in \{1, ..., k\}$. Similarly, as the prior over the parameters $\boldsymbol{\theta}$, we use the Dirichlet distribution (generalization of beta):

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = Dir(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{B'(\boldsymbol{\alpha})} \prod_{i=0}^{k} \theta_i^{\alpha_i - 1}, \tag{4.6}$$

where $B'$ is also a normalization factor, which does not need to be directly computed in our case.

Since the Dirichlet distribution is a conjugate prior of the categorical distribution, the inference is just as simple as in the previous case. Again, we use a uniform prior distribution: $Dir(\boldsymbol{\theta}|\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_k]$ and $\alpha_i = 1$ for $i = 1...k$. In other words, this prior "predicts" with low confidence that a given method is equally likely to be assigned to any of the available categories. If additional prior information is available, it can be easily incorporated by increasing values of some of the parameters of the Dirichlet prior distribution and thus biasing the posterior.

After gathering responses from study participants, we calculate $\boldsymbol{c} = [c_1, ..., c_k]$, where $c_i$ is the number times category $i$ was chosen, and use it to compute the posterior distribution

$$Dir(\boldsymbol{\theta}|\boldsymbol{\alpha} + \boldsymbol{c}) = Dir(\boldsymbol{\theta}|1 + c_i, ..., 1 + c_k). \tag{4.7}$$

Intuitively, assuming a uniform prior, the MAP estimate after the analysis will roughly correspond to the histogram of times each category was assigned to the given method. There are however, two important differences: firstly, because the prior acts somewhat as a regularizer, no category will have an a posteriori estimate of exactly 0, even when no respondents chose it. This is important especially because, depending on the sample size, the fact that certain outcome does not occur in the dataset does not necessarily imply that its probability is 0. A simple example could be 10 subsequent tosses of a six-sided die where, even with no occurence of a 6, we would not conclude that obtaining a 6 is impossible.

Secondly, because the analysis gives us the full posterior distribution, we can also infer the confidence in the estimates obtained and measure the similarity between the two methods in a more precise manner.

The remaining challenge after the analysis is the presentation and visualization the posterior distribution to draw conclusions. When $k = 3$ it is still possible to depict the entire posterior distribution graphically by drawing a heightmap over a triangle, where each vertex represents the probability of each category being chosen; this is shown in Figure 4.6. Note that attempting a similar visualization in higher dimensions would result in incorrect coupling of probabilities “on” neighbouring vertices and decoupling of vertices that are not drawn near to each other.

To avoid this problem, we have decided to marginalize the posterior and show the distribution separately for each dimension (*i.e.* for each category independently of others); in practice this means plotting $k$ separate Beta distributions, since the marginal of a Dirichlet distribution for dimension $i$ is $beta(\alpha_i, N - \alpha_i)$, where $N$ is the total number of trials. Intuitively this is similar to the analysis of ranking, however, this time we count “wins” and “losses” for each class (label) instead of each method. This view conveys enough information to achieve two important goals: depicting the characteristics of a single distribution, and making it possible to compare distributions to each other (*e.g.* by summing their 2D overlaps). The latter can be achieved in the same way as it was in the ranking task: by plotting multiple Beta distributions on the same axis and measuring the overlap between their HDIs.

To illustrate our approach, we have again simulated data from a hypothetical task. Given a set of 20 images taken by a certain photographer and 5 participants, we construct a task where, for each trial, the subject chooses one of three categories based on how the image appears: “sad”, “neutral” or “funny”. Given that “sad” was chosen 26, “neutral” 59 and “funny” 15 times, the marginalized posteriors are shown in Figure 4.7.

The same analysis can be used to compare how different photographers’ scores relate to each other: see Figure 4.8. In addition to individual confidences we can see that the highest density intervals overlap

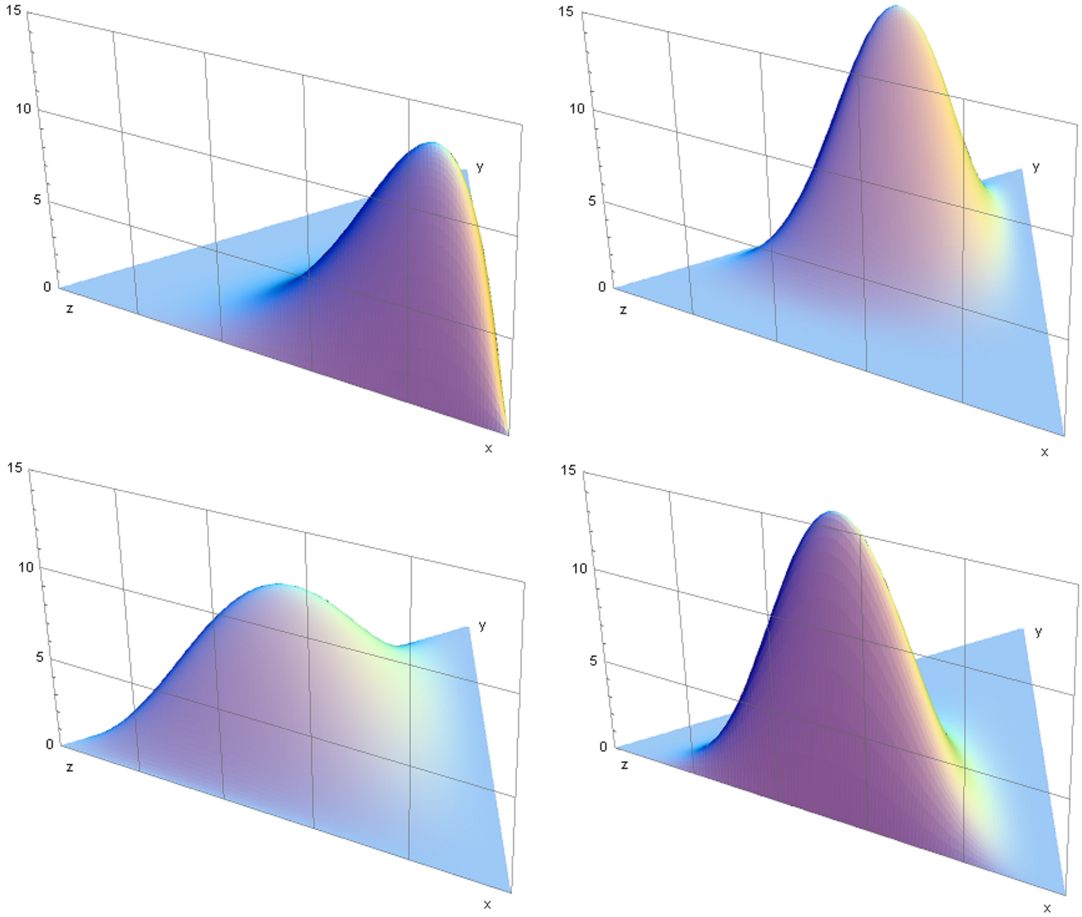
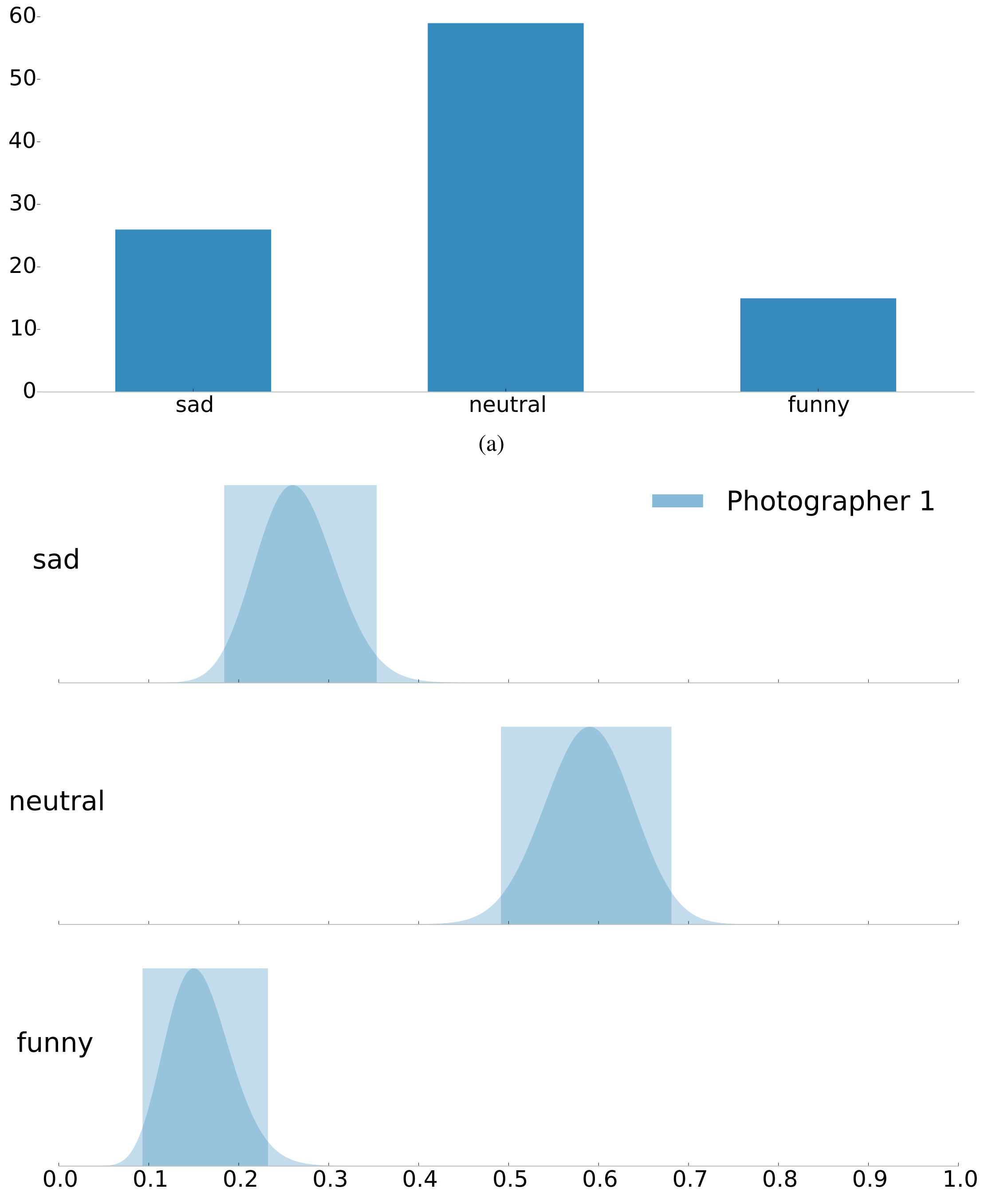


**Figure 4.6:** *Graphical representation of four different, 3-dimensional Dirichlet distributions. In our case, this distribution shows how likely each combination of 3 parameters of the categorical distribution is. Image courtesy of Wikimedia Commons,* `https://commons.wikimedia.org/wiki/File:Dirichlet_distributions.png`



(a)

(b)

**Figure 4.7:** *A histogram of how many times each label was chosen (a) and visualization of the posterior (Dirichlet) distribution over the parameters of the categorical distribution (b) for the hypothetical task described above. In (b) each row shows the distribution marginalized to a single dimension to make visualization possible even with many categories. While both visualizations present similar conclusions, the confidences (widths of shaded HDIs) are only available in the presence of full posteriors.*

for two of the classes. This allows us to conclude that these two conditions had equivalent chances of being assigned “neutral” and “funny” labels; we would not be able to say the same only by looking at the histograms.

## 4.6 Additional Task Types

While the ranking and classification tasks seem to cover a large part of graphics researchers’ needs, based on our literature survey, there are other activities that might also be useful. Below we describe several additional task types, that could also be integrated into our system. While we have not fully explored these tasks and their analysis methods, they might nevertheless illustrate additional interesting possibilities.

### Area Markup Task

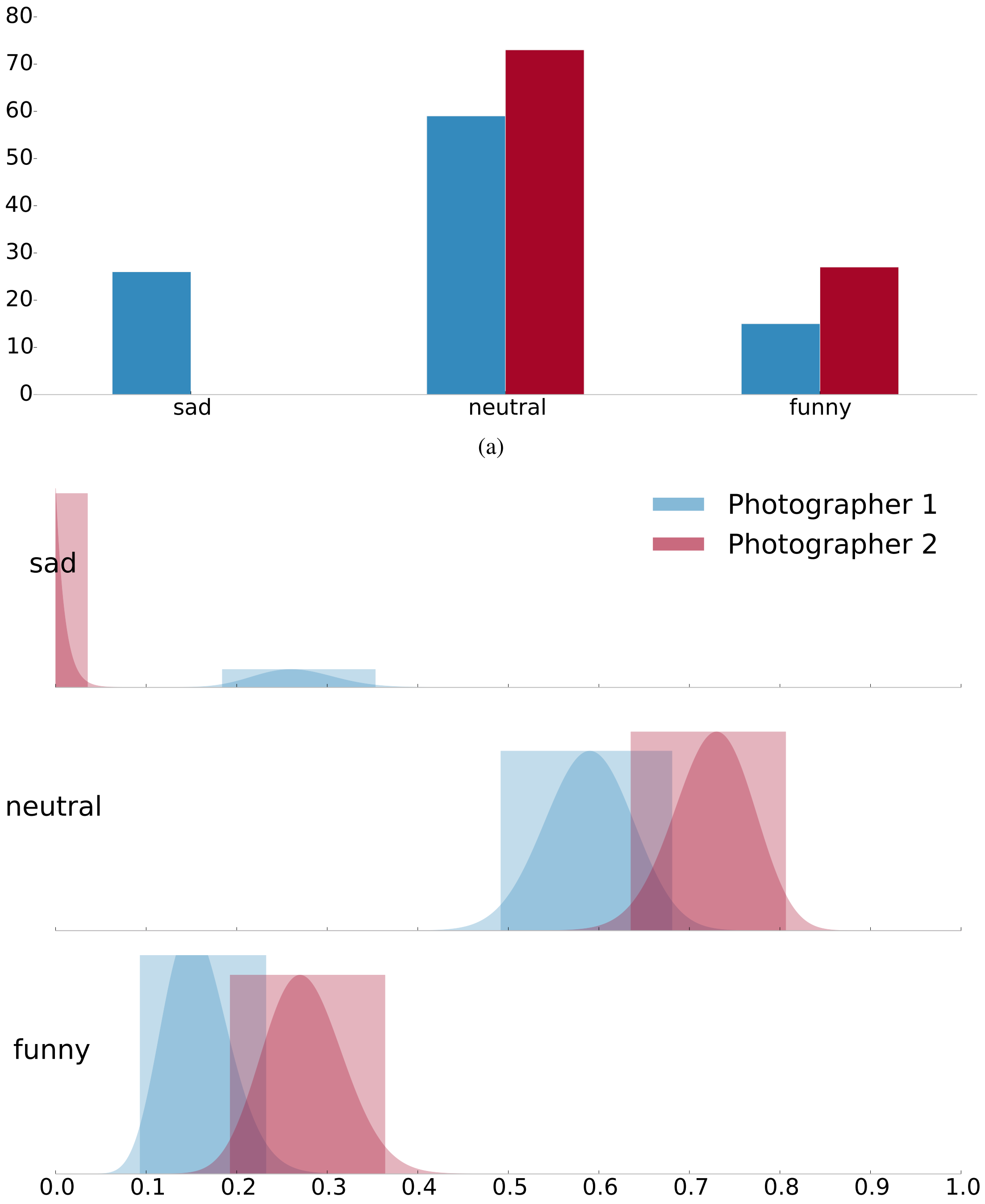
Knowing the performance of relevant methods is a first step, but it is also important to understand where and why different techniques perform better or worse. While this information is lost in the rankings, it can be obtained in a separate task by asking users to mark up relevant areas of the given image. This method can be used to find artifacts and compare them across different results. Similar task was used by Chu *et al*. [CHM+10] to check whether participants could find animals camouflaged into images.

While this task type is not yet integrated into our system, we have created a standalone implementation and conducted a short study on the subset of the shadow removal data (our method contrasted with Mohan *et al*. [MTC07] and ground truth). During the experiment, the participants were shown 15 different images and asked to use a paintbrush tool to indicate regions of the image that they believe were altered. Participants were instructed that an image may or may not contain alterations, and that they should only mark the image if they thought it was modified.

For each participant, the 15 images to mark were randomly drawn from a pool of 71 images. Forty-five of these images were processed using our technique, nine were processed using Mohan *et al*.’s, and 17 were ground truth images that showed a scene with no shadow present. We included the 17 unmodified images to provide a baseline for comparison purposes. The set of 15 images displayed to the participant was randomly chosen subject to the constraint that the participant could not see a ground truth scene without shadows prior to seeing the same scene with a shadow removed using either technique.

For analysis, if any ink was placed on the image, we considered this an instance of the participant believing they have detected at least one artifact. We modeled artifact detection as a Bernoulli random variable with a uniform beta prior. Our results, shown in Table 4.1, show that it is significantly more difficult for individuals to detect artifacts in our images compared to results produced by Mohan *et al*., however, there is also a difference between our technique and ground truth indicating that there is still room for improvement.

There are multiple variations of this task which could be explored and which have the potential to impact the results. Firstly, different interaction paradigms could be used for marking instead of a brush: lassos, scribbles, polygons *etc*. Secondly there are multiple constraints that can be enforced in addition to time, *e.g*. number of clicks or a limited amount of “ink”. Finally more sophisticated analysis



(a)

(b)

**Figure 4.8:** *Classification result visualization of different conditions on the same graph. Note that, while the histograms of the two "photographers"' images might appear different, the posterior HDIs overlap and we have to conclude that no statistically significant difference was found in the frequency of "neutral" and "funny" images.*

| Our Technique | Mohan *et al.* | Ground Truth |
|---|---|---|
| 71% (67-74% HDI) | 85% (78-91% HDI) | 48% (41-54% HDI) |

**Table 4.1:** *Chances of detecting an artifact in images processed with our method, one of Mohan* et al. *and ground truth. The first number gives the MAP estimation, while the numbers in brackets describe the HDI boundaries.*

methods might produce more informative conclusions. For instance, in the above example it might have been interesting to compute an overlap between the marked regions and the known shadow positions. Similarly, image features "under" marked regions could be analyzed in an attempt to find common features of artifacts.

### Image Tagging

Another useful addition to the system might be an image tagging task, where single images are to be tagged with textual labels (either free-form or chosen from a predefined set). While there are similarities to the classification task, tagging assumes that multiple tags can be assigned to each image during each trial. An example of a similar exercise can be found in [RGSS10], where a pilot study was used to determine roughly the contents of each image (*e.g.* lines, faces, texture).

In addition to perceptual image comparisons, this task could also be used for data gathering, *e.g.* to feed machine learning algorithms.

### Paired Classification and Tagging

Finally, it might be useful to either classify or tag pairs of images, as opposed to single ones. This would allow discovery of relationships of images to each other, *e.g.* whether they are similar, whether their colour schemes are compatible *etc*.

## 4.7 User Study Authoring System

We have created imCompadre (`http://www.imcompadre.com`), a user study authoring system fine-tuned for image comparison and evaluation. It allows anyone to perform a rigorous, perceptual user studies and reason about the outcomes, backed by a set of statistical reports. The system is entirely web-based, enabling wide distribution of the study without the usual constraints of organizing face-to-face sessions with participants (however, it is still entirely possible to hold face-to-face sessions if the study author decides to do so). Further, facilities to integrate with Amazon Mechanical Turk are integrated into the system, making it possible to easily recruit large numbers of participants for the study.

While we borrow experience and draw inspiration from psychophysics research, it is important to note that our goal is almost the opposite: the evaluation of algorithms, rather than human perception. Consequently we do not aim to compete with standard psychophysical tools such as the Psychophysics Toolbox [Bra97] and PsychoPy [Pei08]. While there is some overlap between the experiments that those and our system can create, their objectives and constraints differ. Our aim is to create a system useful for compute graphics researchers to evaluate the results of novel algorithms. The built-in tasks, therefore, are tailored to this specific set of constraints.

Further, while interpreting and utilizing the user feedback interactively, using the user-in-the-loop concept, is a very interesting area of research, we currently consider it out of scope for imCompadre, and focus solely on statically evaluating relative merits of image manipulation methods.

Our solution brings several advantages such as convenience, increased reproducibility, heuristic study setup verification, encouragement of running pilot studies, and facilities for examining the results thoroughly. We describe each of the important features below and proceed to specify details about implementation.

### Convenience

The most obvious argument for why the system we have built is useful, comes from the fact that creating user studies in imCompadre is orders of magnitude faster than building them by hand (we have explored this more in Section 4.8). Multiple mundane challenges, including designing the layout for the tasks, writing of HTML and JavaScript, hosting and participant recruitment are abstracted away so that the study authors can focus on creating algorithms.

While most of the real challenges, like ensuring that the study is measuring phenomena relevant to the problem, and avoiding bias in data, are not explicitly addressed, we hope to allow researchers to focus on them by solving the easy, but time-consuming parts.

### Reproducibility

The first way that reproducibility is increased is standardization: our system has a set of predefined tasks that are defined above and can be referred to unambiguously.

Secondly, after creating and running a study, the author can obtain a detailed description of the protocol that the experiment followed, in a human-readable format. This description contains all the variables necessary to repeat the experiment: which tasks were present with what kinds of variations, what was recorded, how many participants were recruited, how they were compensated *etc*. Optionally the evaluated data and results can be bundled together with this description to form a complete experiment package that can be analyzed and repeated again later. This data can also be used to directly “clone” the study in our system for verification or extension.

### Freedom to Choose the Participant Pool

Depending on the aim of the study, selecting a particular segment of the general population might be critical to ensure valid results. For instance, for a company creating software for professional photographers a user study ran solely on computer science students does not carry as much useful information, as one where photographers took part.

Having an online system that can be accessed from anywhere makes the recruitment task slightly easier, especially when using crowdsourcing platforms like Mechanical Turk is feasible. This also opens the possibilities of involving large numbers of subjects. However, it does not mean that the study has to be ran online necessarily, since it is also possible to create a controlled study environment and invite participants to attend physically. Depending on their specific needs, the study authors have the freedom to choose the right setup.

### Rapid Iterations

While repeatability is usually considered a necessity for good academic practice, it can also be helpful in performing research. The most obvious case is running a pilot study, gathering qualitative feedback, examining the results and adjusting parameters before running the actual study. When re-running the experiment is trivial and fast, as it is in imCompadre, this process can be repeated more than once to get more and more detailed information about the data. An example where we have performed such deep analysis post-publication and found additional interesting properties of described methods is given in Chapter 5.

This is equally true in the industry, where such a rapid, progressive discovery could be a significant competitive advantage.

### Analysis of Results

Analysis of results, described in earlier chapters, is included in our system. After running the study, the researchers can obtain reports with all the data from their experiments compiled and analyzed. After investigating past computer graphics research, we have only included analysis methods, which are likely to be useful for this purpose. Additionally, raw response data from participants together with analysis recipes are available so any additional processing can be done by the study creators.

## 4.7.1 Implementation

We have chosen to implement the system as a web application, for several reasons. Firstly this makes it straightforward to make the experiments available on the Internet, which in turn greatly simplifies participant recruitment. Secondly, there should be no need for researchers to be forced to install (and keep up-to-date) another software package just to create user studies. However, since the critical parts of our system will be released under open source license, the researchers will be able to run the experiments locally if such need arises.

Further, because of the open source nature of parts of the system, we are hoping to create an active development community dedicated to improving the state of perceptual user studies. Managed correctly, such setup will enable rapid development and iterations, which in turn bring biggest advantages when updates can be applied immediately for all users.

### System Overview

We have implemented the system using the Python[2] programming language and the Django[3] web framework. The choice was dictated both by our familiarity with these technologies and by the widespread use of Python and its scientific packages in the community. The tasks themselves use JavaScript (with jQuery[4]) for user interface and interaction, while a PostgreSQL[5] database serves as the back-end storage. Because deployment is managed by the Ansible[6] automation system, it is possible to automatically set up a server in several minutes.

The scientific Python stack (`numpy` [VDWCV11], `scipy` [JOP$^+$] and `matplotlib` [Hun07]) is used for processing and reporting the results.

### Workflow

Here, we briefly describe the workflow that is used, when working with imCompadre.

1. Create a study.

[2] https://www.python.org
[3] https://www.djangoproject.com/
[4] https://jquery.com/
[5] http://www.postgresql.org/
[6] http://www.ansible.com/

2. Define tasks and upload data.

3. Launch the study.

4. Inspect the results.

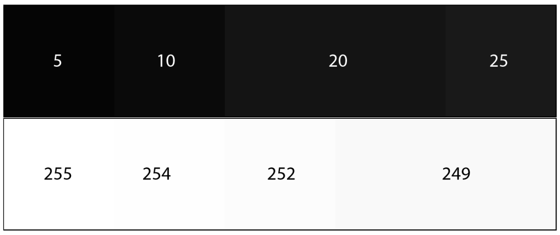
5. If more information is necessary, clone and repeat. Since one of our major goals is to enable quick iterations through different versions of the study, out system makes it very easy to duplicate, export and import studies. This means that not only researchers can adjust the study parameters quickly, but also share their studies with others simply by sharing a `zip` file.

### Analysis

Because we have, so far, only used closed-form solutions to the problems encountered, the combination of `numpy` and `scipy` sufficed to compute the posterior distributions. If, however, we extend the system to include problems where no analytical solution exists we will turn to one of Python's MCMC packages, such as `pymc` or `emcee`.

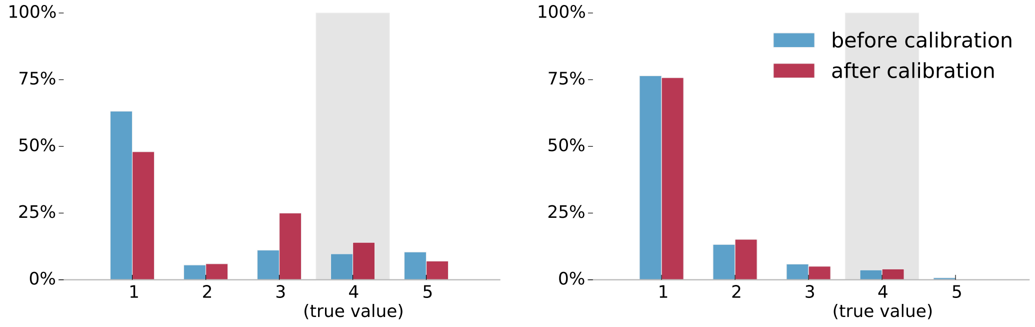
## 4.7.2 Display Calibration

Accurate colour representation is a major argument in favor of running perceptual user studies in controlled conditions. While not all research areas require exceptionally well-calibrated equipment, for some it might be crucial. We cannot hope to compensate for different colour representations completely, however, we have developed two additional tasks: a) "passive calibration", enabling identification of users with well-calibrated displays, and b) "active calibration", which modifies images for individual users based on their initial contrast adjustment.



Our system is flexible enough to allow building of simple calibration-state discovery tasks using the elements described in previous sections. To carry out the passive calibration we created a classification task, asking users how many distinct pixel intensities they perceived on presented images. We asked the same question twice, first showing a dark image containing 4 different, but similar intensities and then a bright one, also with 4 different intensities (both images are illustrated on the inset figure on this page; the participants did not see the intensity numbers and both images we shown separately on a neutral gray background). Users with perfectly calibrated monitors should have indicated that they were able to discern 4 intensities in both cases. We found that, out of 134 respondents, only one person correctly identified 4 intensities in both test images. The distribution of answers is shown as a histogram in Figure 4.9 (blue bars). The fact that the majority of people did not perceive any difference in intensities confirms the suspicions that using Mechanical Turk is difficult when accurate colour representation is important.

We also implemented a separate tasks to explicitly perform simple contrast calibration. This task can be added to the beginning of any study and asks the participants to first maximize their display



(a) Low intensities

(b) High intensities

**Figure 4.9:** *Contrast perception results before and after calibration for low (a) and high (b) intensities. The calibration increased the subjects' ability to distinguish between the dark intensities, since the values of 3 and 4 received relatively more votes.*

contrast, if possible, set the brightness to a comfortable level, and then move the contrast slider to adjust an interactively-updated test image. This image shows a horizontal, linear gradient of intensities from 0 to 255 and, on a perfect monitor, would exhibit no visible steps in intensity (*i.e.* vertical stripes). After the participant adjusts the slider to minimize the steps, we save the corresponding contrast correction and apply it to every image shown to them afterwards. Thus, rather than only asking the participants to calibrate their displays, which may not be possible *e.g.* on tablets, we additionally calibrate the images for the their displays. After re-running the first experiment on 100 new participants, but with the active calibration step included, we found that people were able to better distinguish between the low intensity patches (red bars in Figure 4.9 (a)). The perception of bright intensities, however, was not affected.

While from this data it is not possible to isolate the effects of hardware display adjustments and our software calibration, we have shown that including the calibration step can, at least partially, compensate for varying display characteristics of study subjects' displays. Further, the passive calibration task can be modified by study authors *e.g.* to find the JND (Just Noticeable Difference) in intensity levels for each participant and adjust accordingly. Additionally, Mechanical Turk allows for "qualifications", that is pre-selecting of respondents, who satisfy certain criteria. This functionality could be used to run a study only on people, who succeed in discerning the given intensities.

Finally, the above steps can be replaced by standard display calibration procedures when it is practical to conduct the study in controlled conditions and recruit the participant cohort through conventional means.

### 4.7.3 Limitations

In the current state, imCompadre has two important limitations. Firstly, it is not possible, at the moment, to specify when and in which combinations the stimuli are to be presented. While this is a desirable feature, we have so far mitigated it by recruiting large numbers of participants and showing them randomly-selected data, based on simple rules describing which images should be displayed together and which separately.

Secondly, we have not fully explored the reliability of results obtained via Mechanical Turk. Even though works like [KCS08] found that it is reasonable to trust microtask workers given the right study setup, more exploration of our specific situation is needed. In addition to the good practices of study design described in 2.2.4 and meant to minimize cheating, there are several heuristics, which can be used to filter out unwanted data:

- Result consistency checks: users' answers to the same questions over multiple trials should be reasonably consistent. The "consistency threshold" used for rejection depends on the data and should be found during a pilot study (see *e.g.* [CSD+09]).
- Interaction consistency checks: whether a user always performs the same action regardless of the data (*e.g.* always selecting the image on the right-hand side).
- "Honeypot" tasks: presenting input for which the answer is obvious and filtering out users, who provide obviously wrong answers.
- Interaction-based filtering: recording different actions the user might take while making a decision (including time taken) and only accepting answers that are accompanied by expected patterns. See [RK11] for an implementation of a learning algorithm that is trained to discriminate between malicious and honest feedback based on interaction alone.

However, our system currently has no cheating detection performed by default. We have experimented with different ways of filtering (*e.g.* only taking into account trials with more than 3s spent on the decision), but so far have not found significant differences between the conditions.

In the case of ranking tasks, cheating would most likely skew the results towards the "null" value of 0.5, if we assume a "cheating model" of random selection. Consequently, our system currently might report results slightly biased towards no difference between conditions. This should be kept in mind when interpreting the results, as we demonstrate in Chapter 5.

While an effort was made to include broadly-applicable data analysis methods, it is expected that researchers' needs will continue evolving in difficult to predict ways. For this reason, we have included a facility to download the raw results, as well as recipes for common workflows, to enable study designers to carry out their own analysis.

Finally, the system was built with the use case of evaluating graphics research in mind. Consequently, we have not investigated its suitability for the general case of psychophysical research. For instance, while the facilities to show stimulus for a limited time are included, the timing precision was not investigated. While we are confident that using imCompadre is a good choice when one wishes to compare results of different image synthesis or manipulation methods, we would not recommend it when the aim is the evaluation of human perception.

## 4.8 Meta User Studies

Our user study authoring system only achieves its goals if it is easier and faster to create a user study with its help than without. To evaluate the usability and value that imCompadre brings, we have conducted a small "meta user study" on several researchers.

The procedure of the meta study is not as rigorous as we are advocating for the case of image evaluations. This is mostly caused by the fact that the audience for our system is much more narrow and consequently it is more difficult to assemble a large group of participants. We believe, however, that it is enough to prove that our system is better than any alternative, including building of user studies manually.

### 4.8.1 Experiment Design

The aim of the experiment was to find out how quickly a graphics researcher will be able to create user studies in our system after being introduced to it for the first time. We also wanted to make sure that the results were scalable, *i.e.* that the participants were able to learn what they needed without any in-person assistance.

For this purpose we have created two exercises for the users to complete, each one recreating a different study from two past SIGGRAPH papers ([RGSS10] and [KHFH11]). The first exercise included a 15-minute screencast explaining how the authoring system works and what it is capable of and walking the user through the exercise. The second exercise was to be completed by the participants with no help, using only the description of the study and the knowledge gained previously. The exercises are available online (`http://www.imcompadre.com/docs/meta_study.html`) as well as included in Appendix B.

The experiments were set up to measure only the study creation time; design of the user studies, question formulation and the test data in the right format were all provided in advance. Consequently, the times reported below do not represent all the effort that needs to go into creating high-quality perceptual experiments, instead, focusing on only the parts that are within the scope of what imCompadre solves.

### 4.8.2 Results

On average it has taken our 6 participants 6m 37s to complete Exercise 2 (building the study). While we did not measure the entire experiment for every participant, three individuals reported time under 30 minutes, including the registration and learning about the system.

We conclude that, at the cost of roughly half an hour of learning, the researchers gained the ability to create publicly available experiments in under 10 minutes. While the most difficult aspect of creating user studies is the definition and decision about what to investigate, the fact that they can be set up and iterated on so rapidly is a significant improvement in the research process, especially when combined with the power of crowdsourcing, which enables one to obtain first results in minutes.

# Chapter 5

# Case Studies

Having built a user study authoring system, we have conducted four experiments to prove that it can satisfy the user-study-authoring needs of graphics researchers. To that end we have chosen three past SIGGRAPH publications and our own Transactions on Graphics article and have recreated their perceptual experiments using our new system. The aim of this exercise was two fold: to prove that imCompadre is capable of creating the kinds of experiments our community needs, and that it can be a tool in improving the reproducibility of research in our community by making user study re-runs very easy.

To emphasize the second goal, we have included the files describing the study setups together with the necessary data in the supplementary material[1]. These files are in human-readable JSON format, but can also be directly used to reproduce studies on imCompadre.

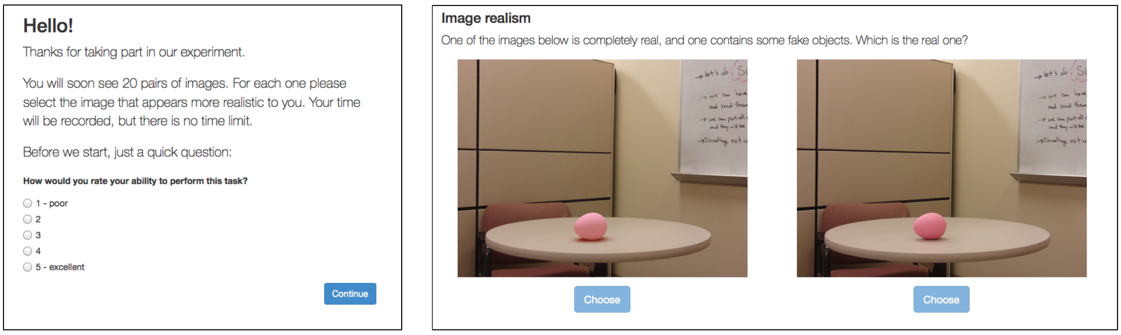
## 5.1 Rendering Synthetic Objects into Legacy Photographs

Karsch *et al*. [KHFH11] presented a method of compositing synthetic, 3D objects into real photographs. Since the method's aim was to produce images realistic to human observers, a perceptual user study was carried out to verify the results.

### 5.1.1 Our Implementation

The study described in the paper consisted of two kinds of questions: a) with a real image shown next to synthetic one and b) with two synthetic images rendered with different methods shown next to each other. Every time the participants were asked to choose the image that appeared more realistic to them. While there were three rendering methods evaluated (referred to as baseline, light probe, and the proposed method), we were only able to obtain a set of images containing the images rendered with the authors' proposed method in addition to the corresponding real images. We believe, however, that this dataset is representative enough to recreate the most interesting findings of the study and even provide additional insights. Further, we chose not to include the "variants" of images in the study as the authors did. Variants, which were: converting images to monochrome, adding clutter, cropping, and changing illumination, hindered the participants' ability to distinguish between real and synthetic images. Therefore, by not including them we have tested the most adverse scenario for the algorithm, *i.e.* one where it should be easiest for participants to answer correctly. This also means that we have asked more questions

[1] also under `http://www0.cs.ucl.ac.uk/staff/M.Gryka/download/thesis-data/`



Hello!

Thanks for taking part in our experiment.

You will soon see 20 pairs of images. For each one please select the image that appears more realistic to you. Your time will be recorded, but there is no time limit.

Before we start, just a quick question:

**How would you rate your ability to perform this task?**

○ 1 - poor
○ 2
○ 3
○ 4
○ 5 - excellent

Image realism

One of the images below is completely real, and one contains some fake objects. Which is the real one?

(a) (b)

**Figure 5.1:** *The introductory screen (a) and the task view (b) for the side-by-side interface in the "Rendering Synthetic Objects into Legacy Photographs" study re-run.*

under the same conditions, potentially resulting in even more exhaustive exploration of the problem.

To summarize, we have created a user study, where the subjects were shown two images side-by-side, one real and one synthetic, and asked them choose the one they thought was real. The order of the pairs was randomized, as was the placement of the images on the left or right. At the start of the study, the participants were shown an explanation of the task ahead (telling apart real and synthetic images), told that their time will be recorded, but there willbe no time limit and asked for their confidence in being able to answer the questions correctly. Subsequently, they were asked 20 questions using the interface demonstrated in Figure 5.1 (b).

After creating the experiment, we have used Amazon Mechanical Turk to recruit 51 participants to complete it. As described in Section 4.4.2 we have modeled the probability of people mistakenly choosing the synthetic image using a Bernoulli random variable and a uniform beta prior ($\alpha = 1$ and $\beta = 1$). Figure 5.2 shows the posterior distribution of the probability of the synthetic image being chosen after including all 1017 data points we have collected from our participants. While the "chance" value of 0.5 does not fall within the Highest Density Interval (HDI), people had between 36% and 42% chance of choosing the synthetic image, so they were not able to recognize it reliably. These results are an even stronger confirmation of the findings reported in [KHFH11]: in their experiments, when not using any variants, the subjects had only about 20% chance of being mistaken.

One possible explanation for the difference is that the original study had only 30 data points exploring this exact condition (*i.e.* probability of choosing their synthetic image over a real one, when no variants were applied). Consequently, the confidence intervals after analysis of this data should be quite wide; we have simulated this by randomly choosing 30 answers from our dataset and plotting the posterior as before. This implicitly assumes that our data samples are i.i.d. (independent and identically distributed), which is not necessarily true since all answers by a given individual might be biased in a similar way and therefore not independent. However, given the large number of data points and a small number sub-sampled, we feel comfortable making this assumption for the purposes of illustrating the confidence interval width given different number of estimates. Figure 5.3 shows the distribution over the inferred chance of choosing wrongly, and confirms that the estimate with this small number of data points is visibly uncertain, with the HDI ranging from 21% to 54%.

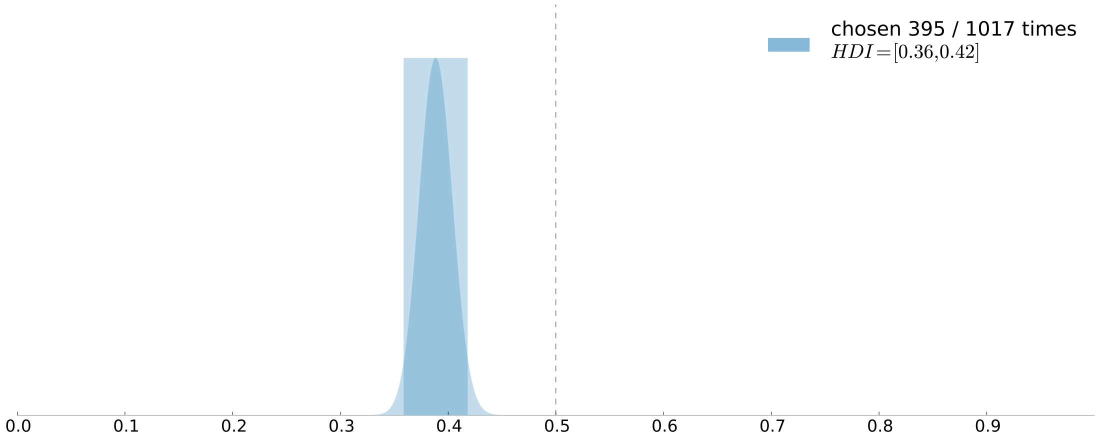


**Figure 5.2:** *Posterior probability of a synthetic image being chosen when shown next to a real one. The posterior is a Beta distribution with mean* $\mu = 0.39$ *and the shaded rectangular region signifies 95% HDI between* $[0.36, 0.42]$.

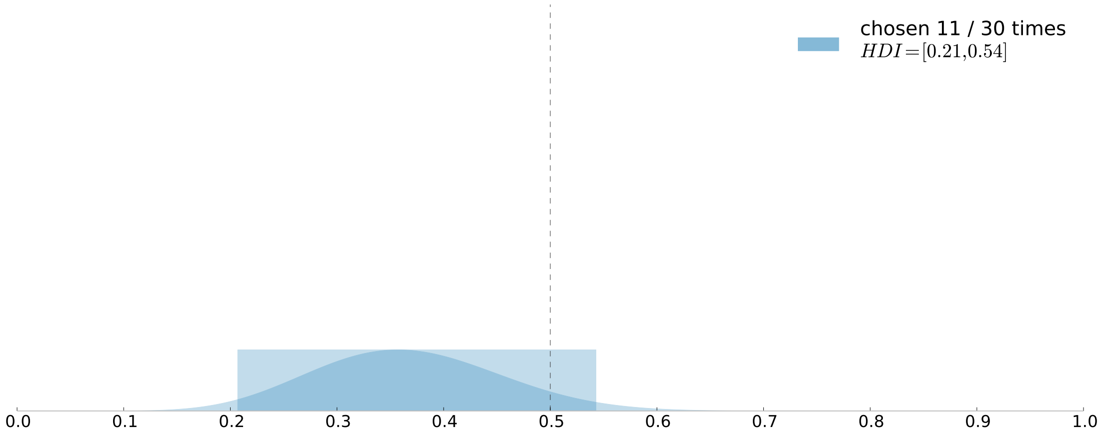


**Figure 5.3:** *Posterior probability of a synthetic image being chosen when shown next to a real one, when only 30 randomly-chosen data samples were included, as opposed to 1017 in Figure 5.2. This distribution has the mean of* $\mu = 0.37$ *and the HDI* $[0.21, 0.54]$*; its width should roughly illustrate the level of confidence in the original study.*

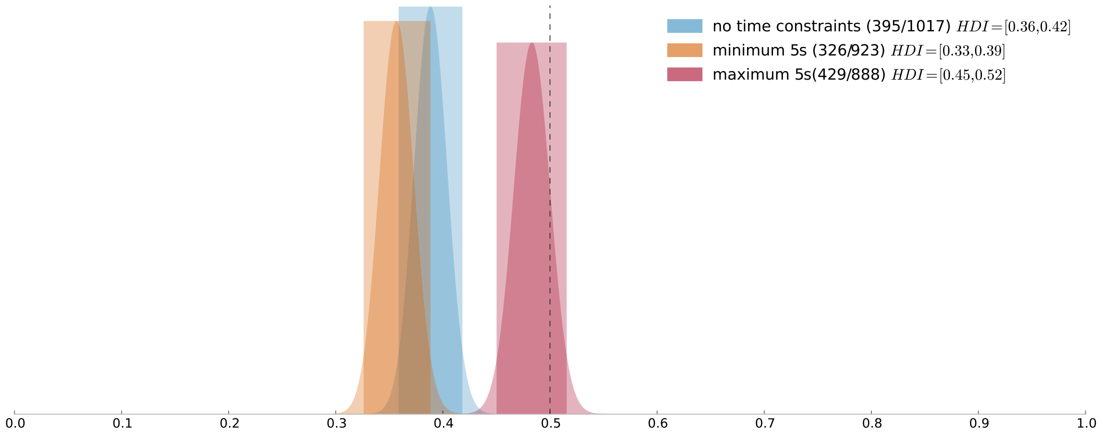


**Figure 5.4:** *Posterior probabilities of choosing a composite image under the "minimum 5 seconds" (yellow) and "maximum 5 seconds" (red) conditions, compared with the original, unconstrained results (blue).*

### 5.1.2 Additional Experiments

We have carried out additional experiments, in an attempt to better understand the merits and limitations of the proposed technique. In this section we describe how the results were affected after introducing time constraints, using a different image selection interface, and a different modality (classification task instead of ranking).

#### Time Constraints

We have re-run the same experiment described in the previous section twice, each time adding a time constraint on the participants. In the first run, we have added a minimum time required ("minimum 5 seconds" condition) to encourage the subjects to look at the images for at least 5 seconds before being able to indicate their choice. In the second run, we have instead enforced a *maximum* time ("maximum 5 seconds"; measured from the time that the images finished loading) that the participants could look at the images. After this time the images disappeared, but the buttons to indicate the decision were left on screen until clicked. In each case we have recruited 100 participants on Mechanical Turk. The studies were ran a few days apart. While we did not explicitly prevent the same participants to complete both runs, we filtered out the data from already-seen participants in subsequent studies, so the presented sample only contains people, who have not seen the images previously.

The results, shown in Figure 5.4, indicate that there is no difference between "minimum 5 seconds" and "unconstrained" conditions (while the "minimum 5 seconds" condition has lower MAP estimate to "unconstrained", the difference is not statistically significant given the number of samples). However, a clear impact of the "maximum 5 seconds" condition is demonstrated by the fact the HDI of the posterior distribution in this case does not overlap with the others and, notably, includes the 50% value meaning that the probability of choosing the composite image is close to chance. We interpret this result as an indication that, at first glance, the proposed algorithm produces very believable results and only closer inspection allows somewhat higher chances of making the correct choice.

Additionally, it is possible that the unconstrained results are skewed in favor of the new rendering

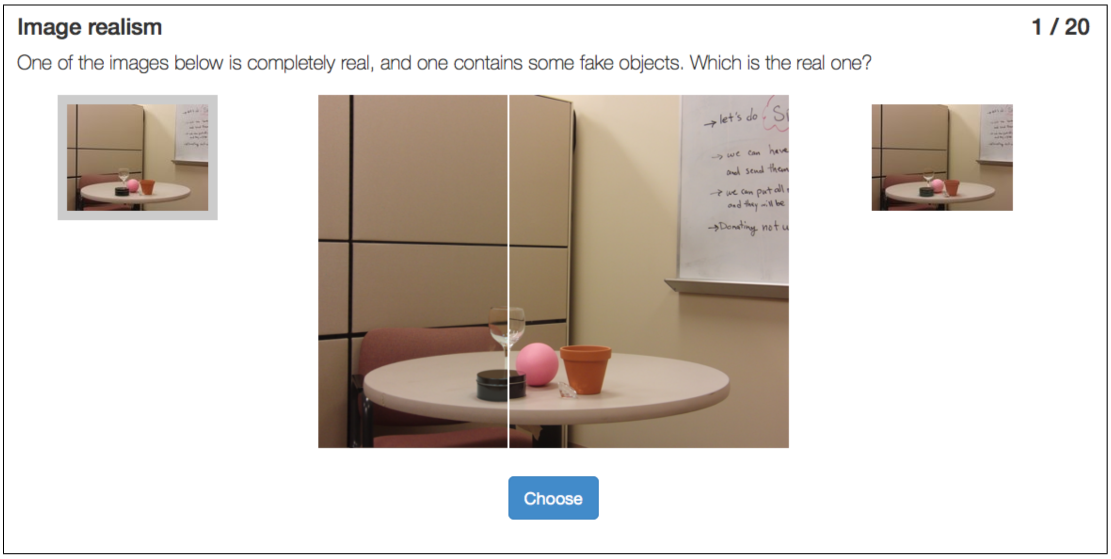


Image realism

1 / 20

One of the images below is completely real, and one contains some fake objects. Which is the real one?

**Figure 5.5:** *The "curtain" interface for the ranking task. Note that at this instant the mouse cursor is inside the image (indicated by the white vertical line) and the "curtain" is dragged half-way, revealing half of the left and half of the right image. Because the dividing line is closer to the left side, the left image is currently selected and highlighted using a thick, gray border.*

technique, because of the limitations we have mentioned in Section 4.7.3. Namely, if we expect some proportion of workers attempted to "game" the system by spending as little time as possible on the tasks, we can assume that they would select images at random, effectively pushing the posterior distribution towards 0.5. This can explain the slight difference in distributions after introducing "minimum 5 seconds" constraint designed to minimize cheating. However, since the difference is not large enough to be conclusive, and our results agree with the findings in the original paper, where in-person user study was performed, we believe that possible presence of some malicious subjects does not invalidate the conclusions of the experiment.

### Alternative Interface for the Ranking Task

In addition to the side-by-side interface shown in Figure 5.1 we have also implemented a different one, more suited to discriminating between very similar images. In this interface, which we call "curtain", images are overlaid on top of each other and the user can reveal one or the other by moving the mouse cursor from left to right. Users make their selection by revealing the image they want to choose. Additionally smaller representations of each image are placed on the left and right to the main canvas area to indicate which image is the currently selected (see Figure 5.5 for the screenshot of the interface).

Our hypothesis, before running this experiment, was that people will find it easier to correctly identify real photographs, because this interface allows more precise identification of subtle differences. As with the previous studies, we have recruited 100 Mechanical Turk workers. After collecting the feedback, we have found no statistically significant difference between the responses from alternative interfaces (see Figure 5.6).

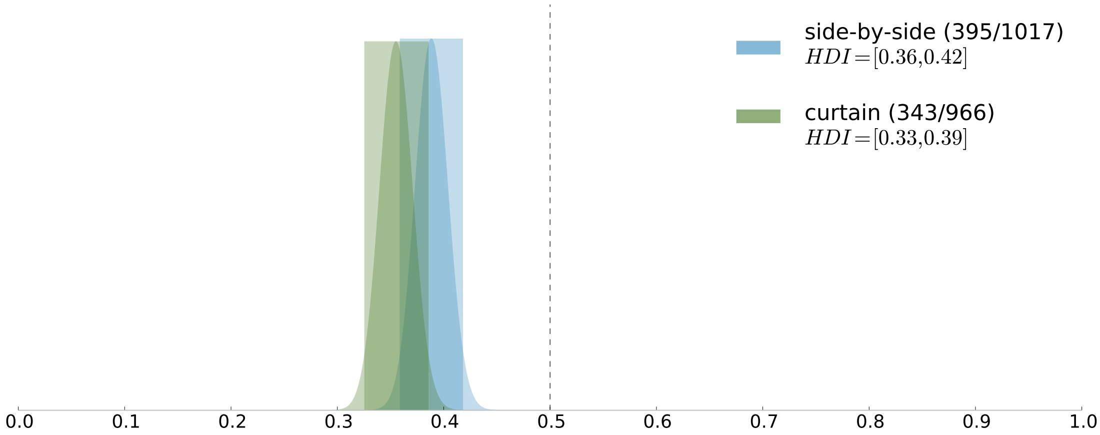


**Figure 5.6:** *The results of two experiments ran using different interfaces. While the posteriors are not identical, the obtained results do not indicate a detectable difference in participants' responses overall.*

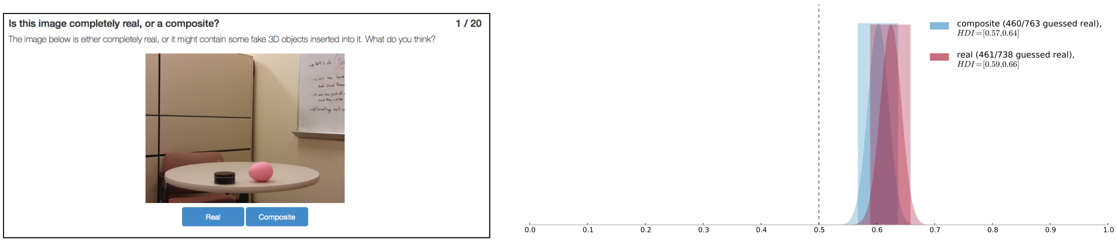
## Alternative Task Type

Our authoring system makes it easy to create user studies to find out peoples' opinions about visual stimuli. It is also important, however, how each experiment is structured and the tasks presented. This is why we have made it very easy to run small pilot studies, look at how participants behave and iteratively improve the experiments. An important part of this process is gathering peoples' opinions not only about the stimulus, but also about the study itself. To allow this kind of meta feedback, we have included a non-mandatory "comments" field after at the completion screen of every study.

While not every subject submits additional notes, and from those who do, not every comment is an actionable piece of advice, we still found it very useful to look at general impressions after completing the study. For instance, after running the experiments described in previous sections, we have noticed that many people mentioned that they were very uncertain about how well they did and that discriminating real from composite images seemed more difficult than they anticipated (a finding also reported in the original paper). Since there was a statistically significant difference between real and synthetic images in the previous run, we have decided to find out under which conditions the difference disappears.

Based on this feedback, we have designed an additional study, to measure whether people would be able to correctly identify composites when only one image was shown at a time. Therefore, we have created a classification task where, given an image, people were asked to classify it as either a real photograph or a composite. The data for this experiment was exactly the same as the one for previous studies and consisted of 10 real images and 10 corresponding composites. In the study introduction we have let the participants know that it is possible that either all or none of the images will be real.

Similarly to previous experiments we have decided to recruit 100 participants to answer the survey. After gathering their feedback we found that there was no detectable difference in the frequency with which people guessed that images are real, whether they were presented with a real or a composite image by Karsch *et al*. We conclude that the method is successful in creating photorealistic composites when presented separately.


Is this image completely real, or a composite?

1 / 20

The image below is either completely real, or it might contain some fake 3D objects inserted into it. What do you think?

Real
Composite
composite (460/763 guessed real), $HDI=[0.57,0.64]$
real (461/738 guessed real), $HDI=[0.59,0.66]$
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

(a) (b)

**Figure 5.7:** *The classification task interface (a) and results (b). After gathering opinions of 100 participants, we have detected no statistically significant difference between the frequency of guessing that the image is real, when presented with a real image (red curve) or with a composite (blue) from Karsch* et al.

## 5.2 Camouflage Images

Chu *et al*. [CHM+10] created an algorithm to synthesize "camouflage images": pictures with seamlessly integrated, hidden content that is only visible on closer inspection. Similar images have been previously produced by artists, but always required significant, manual effort.

Since the defining feature of any camouflage image hinges on specific features of the human visual system (*i.e*. the ability to detect camouflaged content, but only after some time), this research had to be evaluated based peoples' subjective opinions. Consequently the authors decided to run a user study to prove that their results were of comparable quality to artists' works.

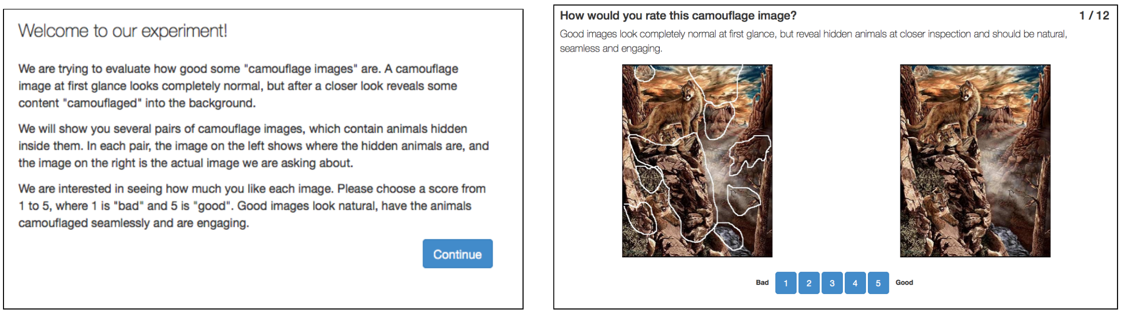
### 5.2.1 Our Implementation

While the original study contained two stages, our system is, at the moment, only capable of recreating the second one. Implementing the first stage exactly would require adding customizable logic to the study flow based on participants' actions; while this would be a potentially valuable addition, we have not encountered enough such examples in the literature to make it a priority.

The second stage, however, uses similar concepts to ones we have identified in other perceptual studies: the subjects are shown an image and asked to assign a score on a previously-defined spectrum. In the case of camouflage images, the participants were asked to assign a quality score 1–5, where high quality was defined as containing animals camouflaged in a natural, seamless and engaging way. The experiment was preceded by a short, one-page introduction explaining what camouflage images are, what will be presented and how to evaluate them.

Since in some of the images (both automatically and manually generated) the animals were easily overlooked, we have also included a reference image every time we asked the participant for a score. The exact experiment setup that we used, together with the data and results, can be found in the supplementary material, while Figure 5.8 shows how exactly the task was presented to the subjects. The reference images were also included in the original dataset we have obtained from the paper authors and, to our understanding, used in a similar manner in the reported study.

We have recruited 100 participants from Amazon Mechanical Turk to complete the study. The results we have obtained confirmed the findings from the original paper, even though our analysis procedure was slightly different. While the absolute values of the mean scores were different (3.58 and 3.65 in



Welcome to our experiment!

We are trying to evaluate how good some "camouflage images" are. A camouflage image at first glance looks completely normal, but after a closer look reveals some content "camouflaged" into the background.

We will show you several pairs of camouflage images, which contain animals hidden inside them. In each pair, the image on the left shows where the hidden animals are, and the image on the right is the actual image we are asking about.

We are interested in seeing how much you like each image. Please choose a score from 1 to 5, where 1 is "bad" and 5 is "good". Good images look natural, have the animals camouflaged seamlessly and are engaging.

(a)

**How would you rate this camouflage image?** 1 / 12

Good images look completely normal at first glance, but reveal hidden animals at closer inspection and should be natural, seamless and engaging.

(b)

**Figure 5.8:** *The introduction screen (a) and the task view (b) for the "Camouflage images" study re-run.*

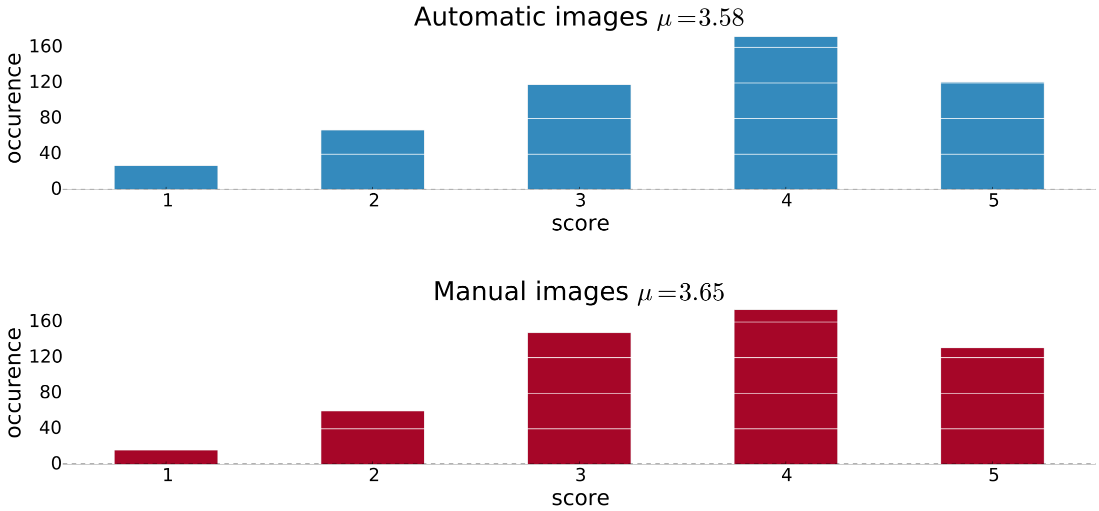


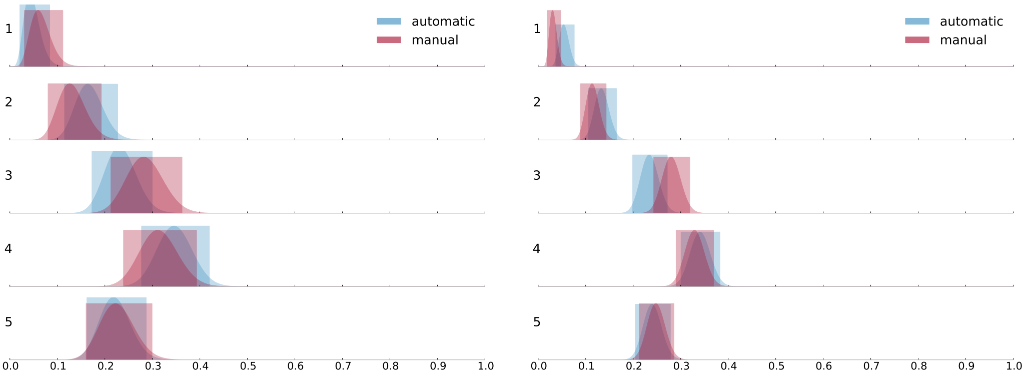
**Figure 5.9:** *Distribution of quality scores for the images created automatically by the algorithm of Chu* et al. *(top) and manually by artists (bottom).*

our results, versus 4.23 and 4.21 in Chu *et al.*'s), the relationship of the two methods' mean scores being virtually the same was replicated. While Chu *et al.* reported only mean values, we have also looked at the histograms of how the scores were distributed (Figure 5.9). While in this particular case the histograms do not provide additional insights, generally they are a useful tool *e.g.* to detect multimodal distributions.

### Additional Analysis

While looking at means and histograms is informative, without probabilistic treatment it is difficult to establish the degree of confidence in the shown estimates. Further, without using Bayesian methods, it is not always possible to formally incorporate prior beliefs into the analysis (even though in this example the prior is a simple uniform distribution). This is why we have investigated the results more by inferring categorical distributions from the user-provided data. As explained in Section 4.5.1 we have used a uniform Dirichlet prior and a categorical distribution with 5 dimensions, one for each possible score.

From this view, shown in Figure 5.10, two important pieces of information can be inferred: our confidence in the estimates and the degree of overlap between the two distributions (blue and red). The confidence is shown as the width of the individual marginals: the wider the distribution the less confident the estimates. Similarly, using the convention of treating 95% highest-density interval of the probability



(a) Posteriors after 300 answers.

(b) Posteriors after 1034 answers.

**Figure 5.10:** *Marginal posteriors over the parameters of categorical distributions of scores from automatic (blue) and manual (red) methods after a) 300 and b) 1034 data points. The means of the distributions in b) correspond to the histogram values in Figure 5.9, however, this plot also shows confidence (the width of the marginal distributions) and the overlap between most likely values (the overlap of the shaded rectangular regions). Note that the additional data visibly increased the confidence in the resulting parameters.*

mass as the best guess for each parameter, we can easily estimate how similar different distributions are by summing their overlaps.

The left side of Figure 5.10 shows our estimates after 300 answers were provided *i.e*. using almost the same setup as Chu *et al*. (assuming that participants' answers are independent and identically distributed; we have randomly sampled 300 data points from our set to obtain this plot). The widths of the confidence intervals is clearly visible and the study author can decide whether they are comfortable with them or whether they need more precision. In this case, we have decided to recruit additional 70 participants to obtain more confident estimates.

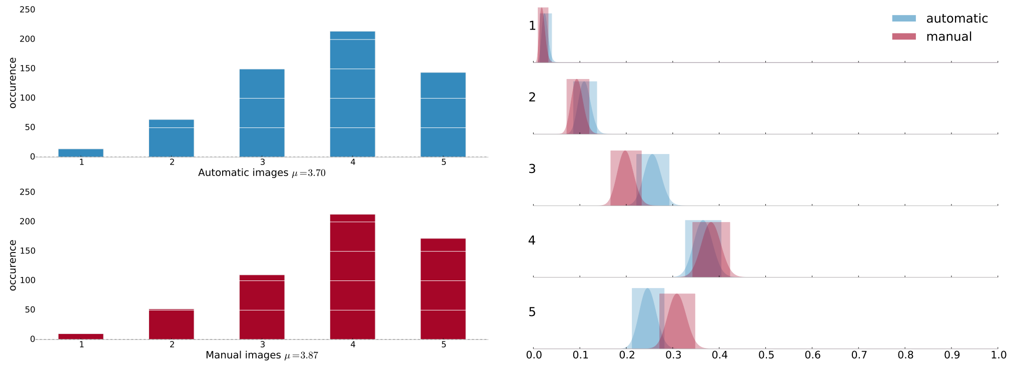
Using this data, we conclude that the distributions of quality scores for the manually- and automatically-created images are virtually identical. While there is some indication that the automatic method is less likely to get lowest scores (1 and 2), these trends are within error margins and could be also attributed to noise.

### 5.2.2 Additional Experiments

To investigate the results more thoroughly, we ran two additional experiments with the same data, but different study setups. The first experiment explored enforcing different constrains on the subjects, while still using the same "classification" task type, while the second experiment aimed to answer similar questions using a different modality: a "ranking" task.

#### Minimum time constraint

In the first additional experiment, we have enforced a minimum time, 3 seconds, that the participants had to spend looking at the images before being able to make their choice. The reason for this change was two-fold: firstly changing the constrains on the participants, might affect how they perceive presented stimulus and thus provide more information about the methods. The second reason is the minimization of malicious or negligent input from microtask workers. As shown in [KCS08], there are indications



(a) (b)

**Figure 5.11:** *Score histograms (a) and marginal posteriors (b) for the likelihood of each score 1–5 being assigned to an image produced by the proposed automatic (blue) and manual (red) methods with the "minumum 3 seconds" constraint enforced.*

that one of effective ways to minimize adverse behavior is to make unwanted actions just as expensive as desirable ones. In the context of this study, we have decided to enforce minimum time per decision, meaning that malicious participants could not simply click through the tasks immediately and had to spend at least 3 seconds looking at the images in question.

Again, we have recruited 100 Mechanical Turk workers to complete this study and the resulting marginal posteriors are shown in Figure 5.11. The data indicates that the users indeed behave differently when faced with this constraint: the likelihoods of obtaining scores 3 and 4 are visibly different for the two methods (however, since their HDIs still overlap, the difference is not conclusive). Notably, the difference between the mean scores only changed slightly from the previous experiment (0.07 previously versus 0.1 now) and having that information alone would not facilitate drawing new conclusions.

With the newly obtained data, we also have to conclude that the automatic and manual methods are equivalent, within a margin of error, since the HDIs of probabilities of obtaining each score overlap. However, it is likely that the manual method is slightly superior under these conditions, since according to MAP estimates, it is more likely to obtain the highest score and the HDI overlap in this case is small.

Note that enforcing such a constraint is not necessarily equivalent to filtering data from unconstrained study to only include answers, where the participants took at least 3 seconds to answer (the analysis of such a filtered dataset is shown in Figure 5.12). Firstly, filtering data out means that potentially fewer data points are included in the analysis and therefore the confidences are correspondingly lower. Secondly, the participants might behave differently knowing the constraints thus affecting the trends in data.

## Ranking task

In the second additional experiment, we have created a ranking task and asked subjects to choose one of the two images that they thought was better (where "better" was defined in the same way as in the previous task and the definition was always visible to the users). This time we have recruited 50 participants on Mechanical Turk and have used the intro screen and interface shown in Figure 5.13.

We have analyzed the data using the procedure described in Section 4.4.2: modeling the probability

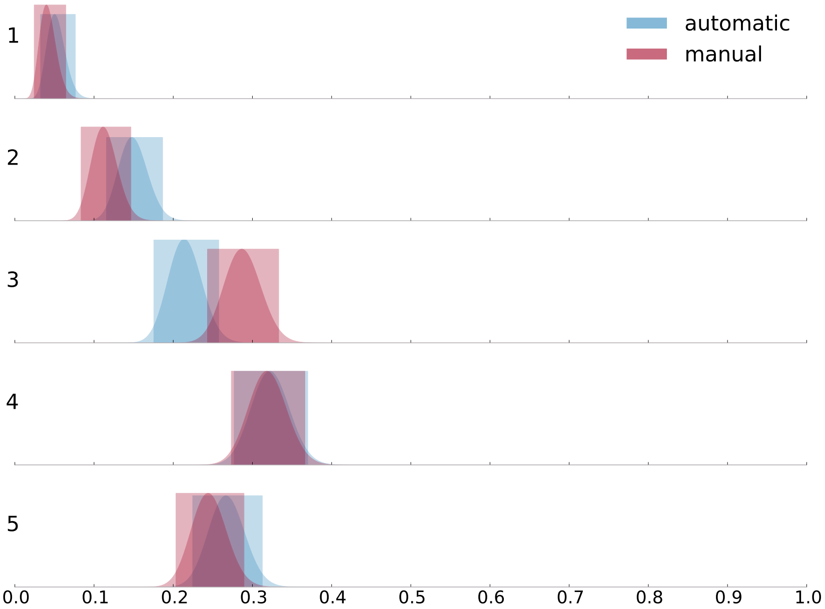
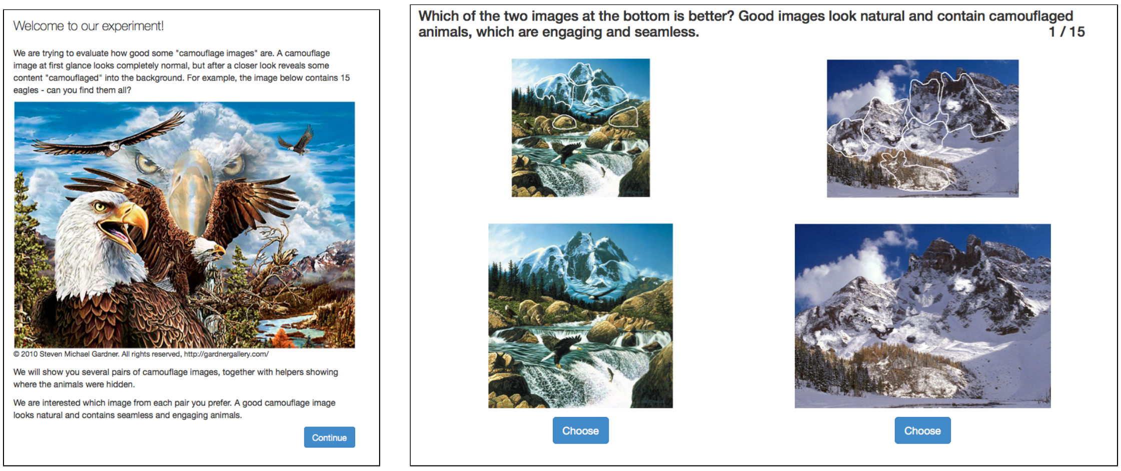


**Figure 5.12:** *Posteriors of score probabilities after filtering unconstrained-task dataset to only include answers, where the participants took at least 3 seconds to answer.*



Welcome to our experiment!

We are trying to evaluate how good some "camouflage images" are. A camouflage image at first glance looks completely normal, but after a closer look reveals some content "camouflaged" into the background. For example, the image below contains 15 eagles - can you find them all?

© 2010 Steven Michael Gardner. All rights reserved, http://gardnergallery.com/

We will show you several pairs of camouflage images, together with helpers showing where the animals were hidden.

We are interested which image from each pair you prefer. A good camouflage image looks natural and contains seamless and engaging animals.

Which of the two images at the bottom is better? Good images look natural and contain camouflaged animals, which are engaging and seamless. 1 / 15

(a) (b)

**Figure 5.13:** *a) Introduction screen to the ranking task and b) ranking interface.*

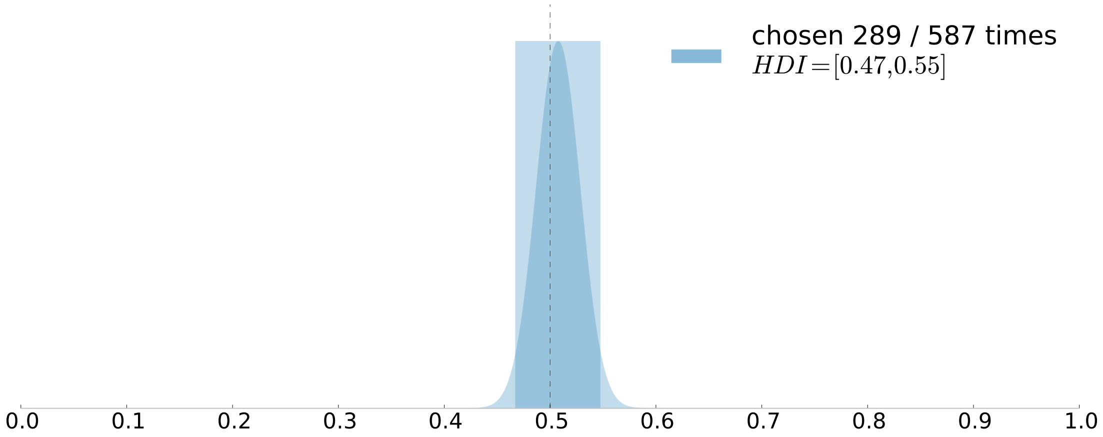


**Figure 5.14:** *Posterior distribution over the chance of automatically-created image being chosen as better than manually-created one. Since the chance value of 0.5 does not fall within the HDI, we can conclude that the methods are equivalent.*

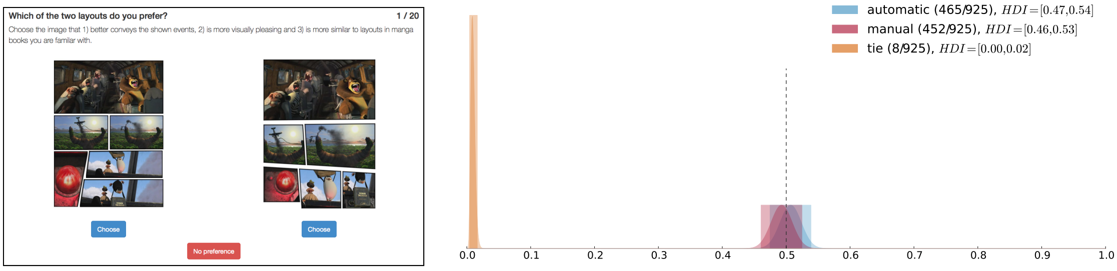
of choosing the automatically-created image as better as a Bernoulli random variable and using a uniform Beta distribution as a prior. After 587 answers the posterior estimate was almost exactly centered around the "chance" value of 0.5 confirming our earlier findings that the proposed algorithm produces images of equivalent quality to the manual ones. (see Figure 5.14 for the posterior).

## 5.3 Automatic Stylistic Manga Layout

Cao *et al*. [CCL12] presented another work which required a perceptual user study to demonstrate its merit. The aim of the method was automatic creation of Japanese-style comic (manga) strips given a sequence of frames form *e.g*. a movie trailer. While manga layouts are very characteristic, they are often created manually and require significant amount of work. Cao *et al*. set out to prove that their layouting algorithm compares favorably when presented against other, mostly manual methods.

Their user study had two phases: the first phase asked people to use a manual tool and the authors' new tool to create manga layouts. The second phase, which we have recreated using imCompadre, showed people corresponding layouts produced using different tools and asked them which image of the pair is better on each of the three evaluation criteria: functionality, visual appeal, and style.

In our reconstruction of the study, we have made two changes to the procedure that Cao *et al*. used. We have replaced their three binary choices per image pair to a single choice, instead asking people to asses overall quality of the image, while defining quality using a combination of the three terms they used separately. We believe that this makes both the participants' decision as well as analysis of the data simpler without giving up valuable insights. This belief is also confirmed by the fact that, in the results reported in the paper, all three evaluation criteria have very similar proportions of people preferring one method over another. The second aspect of the study we have carried out differently is participant selection and stimuli presentation order. Cao *et al*. recruited 10 people who occasionally read manga and distributed 100 pairs of images between them, so that no pair was seen more than once. In contrast, we have recruited 50 Mechanical Turk workers, and have shown them 20 randomly-chosen pairs out of 24 (the difference in data comes from the fact that we were only able to obtain the subset of the



Which of the two layouts do you prefer? 1 / 20

Choose the image that 1) better conveys the shown events, 2) is more visually pleasing and 3) is more similar to layouts in manga books you are familar with.

(a) (b)

**Figure 5.15:** *The selection interface (a) and the posterior probabilities of choosing each tool's output (b) for the recreation of [CCL12] user study.*

images from the original paper). While the results of the two runs might not be directly comparable, our aim is to illustrate that imCompadre can be used to create effective user studies. The ability to recruit many "unqualified" respondents is a benefit in many situations, but it does not stop researchers from carrying out their own recruiting if special needs arise. Note that, in contrast to earlier studies, this selection interface also contains the "No preference" option in case participants thought both image were of similar quality.

After gathering the responses using the interface shown in Figure 5.15 (a) we found no statistically significant difference between how often people select images from the manual tool versus the automatic solution of Cao *et al.*; the plot showing posterior probabilities of choosing either image or "no preference" option is shown in Figure 5.15 (b).

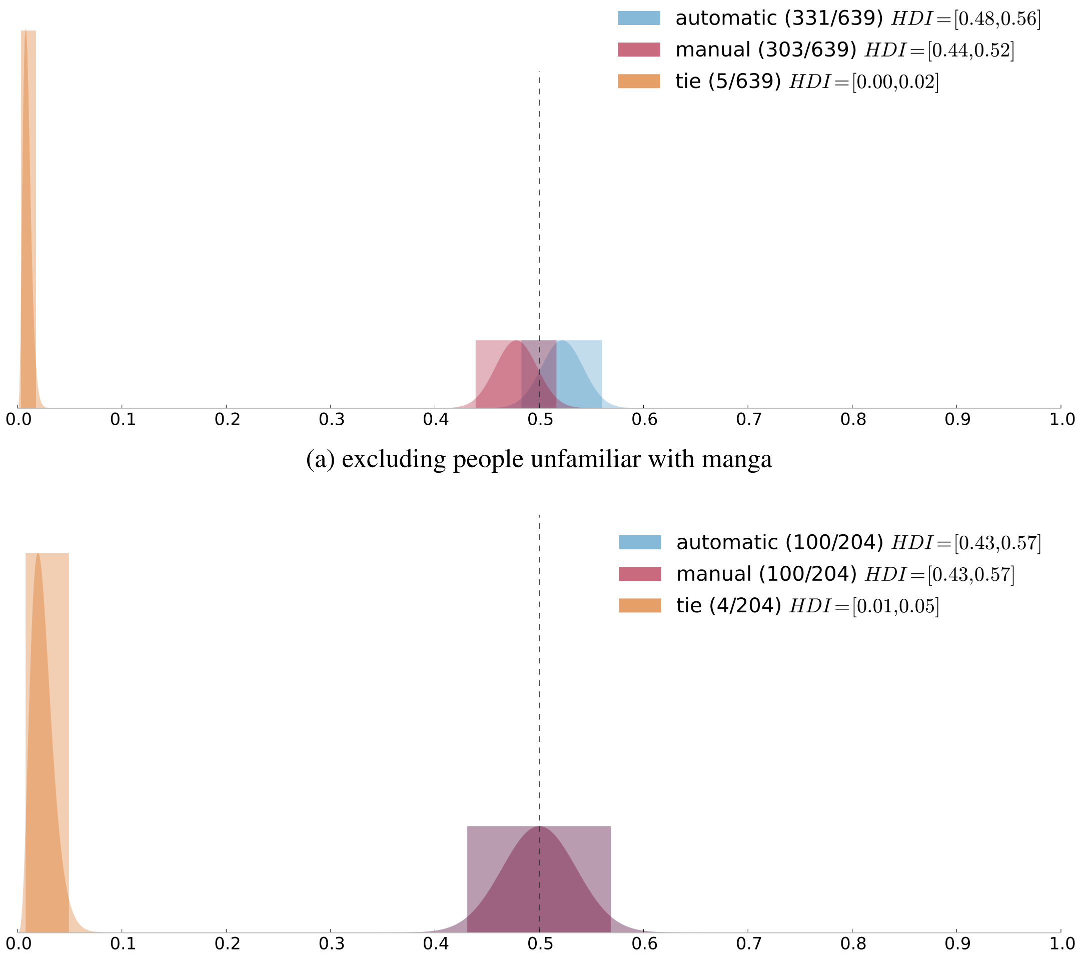
At the start of the experiment, we have asked each participant how often they read manga and given them three options to choose from: "I've never read it" (chosen by 16 people), "I've read it a few times" (26 people) and "I read it regularly" (12 people). We have performed additional analysis on filtered data to find whether people with different degrees of familiarity with manga express measurably different preferences. First, we have excluded answers provided by people, who said they have never read it (Figure 5.16 (a)), and finally we have taken answers only from people claiming to read it regularly (Figure 5.16 (b)). Based on these results, we concluded that regardless of familiarity level, the two tools produce results of indistinguishable quality.

## 5.4 Learning to Remove Soft Shadows

As the final case study, we have recreated the experiments we ran for [GTB15]. The original paper contained a study with two phases:

1. A ranking task (15 questions), where the participants were asked to order images based on how natural they appeared. For each round either 2 or 3 images were presented, depending on how many methods a given scene was processed with.

2. A scoring task (15 questions), where the participants were shown a single image and asked to assign a score 1–4 based on how successful the shadow removal was. Since we did not always want to remove all shadows from every image, we have included a small arrow into each image



(a) excluding people unfamiliar with manga

(b) regular readers only

**Figure 5.16:** *Preference probability densities for data filtered based on respondents' familiarity with manga. Note that in (b) both tools were preferred exactly the same number of times, resulting in exactly overlapping probabilities.*

pointing to the shadow that was supposed to be removed.

Our explicit aim in asking for naturalness of images first and only later mentioning that shadow removal techniques were evaluated, was to measure whether the methods we looked at produced natural-looking images, regardless of the participants' assumptions or knowledge.

Since the study was originally constructed manually, we have recreated it using imCompadre. While the experiments reported in [GTB15] took us 2 weeks to prepare (based on source code commits), we were able to create an equivalent study using imCompadre in under 5 minutes. Unfortunately the first timespan includes also conceptual work and preparation of the study design, while the second includes only building the actual experiment; the conceptual design work, preparation of data, *etc*. were not accounted for. Nevertheless, it seems safe to assume that building an online user study interface and analysis tools takes at least a week of uninterrupted work, even given all requirements.

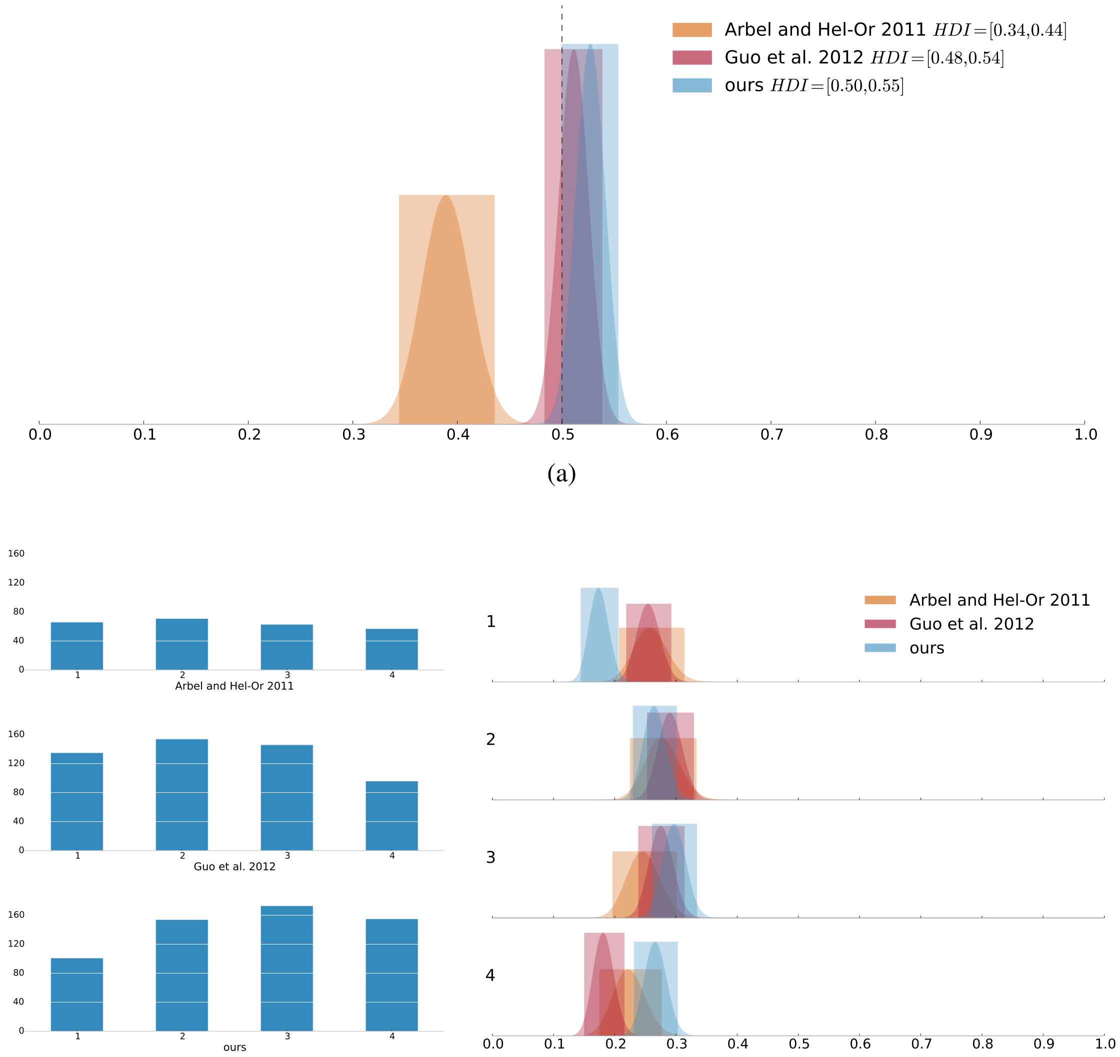
There were two differences between the experiment runs. Firstly, instead of showing people either 2 or 3 images at a time depending on how many were available, we have decided to only show pairs. This decision was motived by both making the task easier for the participants and by making the data analysis more obvious. In the previous case, while for some images we had signals about being "best", "worst" and "half-way", we had to quantize these into "winning" and "not winning" for analysis. Therefore, by being explicit in asking subjects for binary decisions, we were now able to simplify their decision without giving up information. The second difference was that we have omitted asking participants for their confidence in the answers given (during the first analysis we did not find any evidence that confidence scores correlated with any other aspect of the data that we measured and thus were not informative).

While previously we have recruited study participants on volunteering basis through email lists, word of mouth and posts on social media, we have performed new experiments by recruiting 100 paid microtask workers from Mechanical Turk. After gathering their feedback, we have analyzed the results in the same manner as before (Figure 5.17 (a) and (b)), with the addition of estimation of categorical distributions of scores assigned to each method in the second phase (Figure 5.17 (c)).

The results of the first phase did not agree with the findings reported in Chapter 3, where our method clearly outperformed both alternatives. To understand why we have looked at a subset of images which won the ranking round and found that in many cases, where the method of Guo *et al.* was judged more natural, the shadow was not removed at all. This resulted in a natural looking image, which was nevertheless a failure; Figure 5.18 shows a few examples of images, where the method Guo *et al.* failed to remove the shadows, but which won a ranking round. While we were aware of this problem in our study setup previously, and explored the relationship between winning ranking and losing scoring phases in Figure 3.20, we have not seen it impact the results as much.

### 5.4.1 Additional Experiments

To test our understanding of how the naturalness criterion was impacting the study outcome, we have created an additional study of exactly the same format, but this time asking subjects directly about the success of shadow removal; we have also included the guiding arrows in the images, like in the second phase. This time, our findings agreed with the original conclusions and the distributions from this run



(a)

(b)

(c)

**Figure 5.17:** *User study results for the re-run of the soft shadow removal experiments described in Chapter 3. The posterior probabilities of winning in a comparison in the ranking phase are shown in (a). While the method of Arbel and Hel-Or (yellow) performed significantly worse, both Guo* et al.*'s (red) and our (blue) methods are comparable. Plots (b) and (c) both show the data from the second phase, however, (c) also illustrates the uncertainties our our estimates. In the second phase our method has significantly lower probability of obtaining low scores than the other two and significantly higher chance of getting the highest scores than Guo* et al.

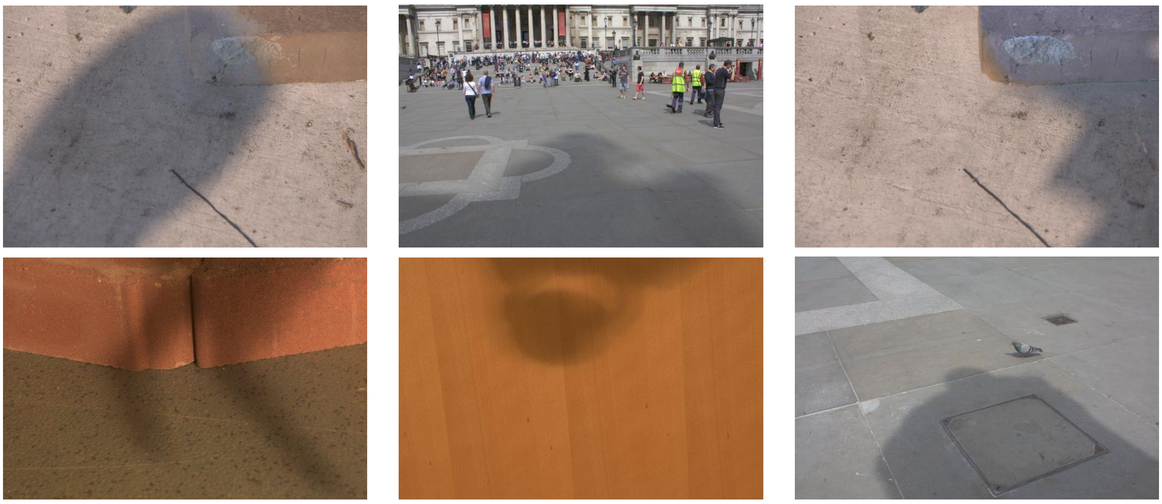


**Figure 5.18:** *Some examples of images, where the method of Guo* et al. *failed to remove the shadows completely, but which won at least one ranking round.*

can be seen in Figure 5.19.

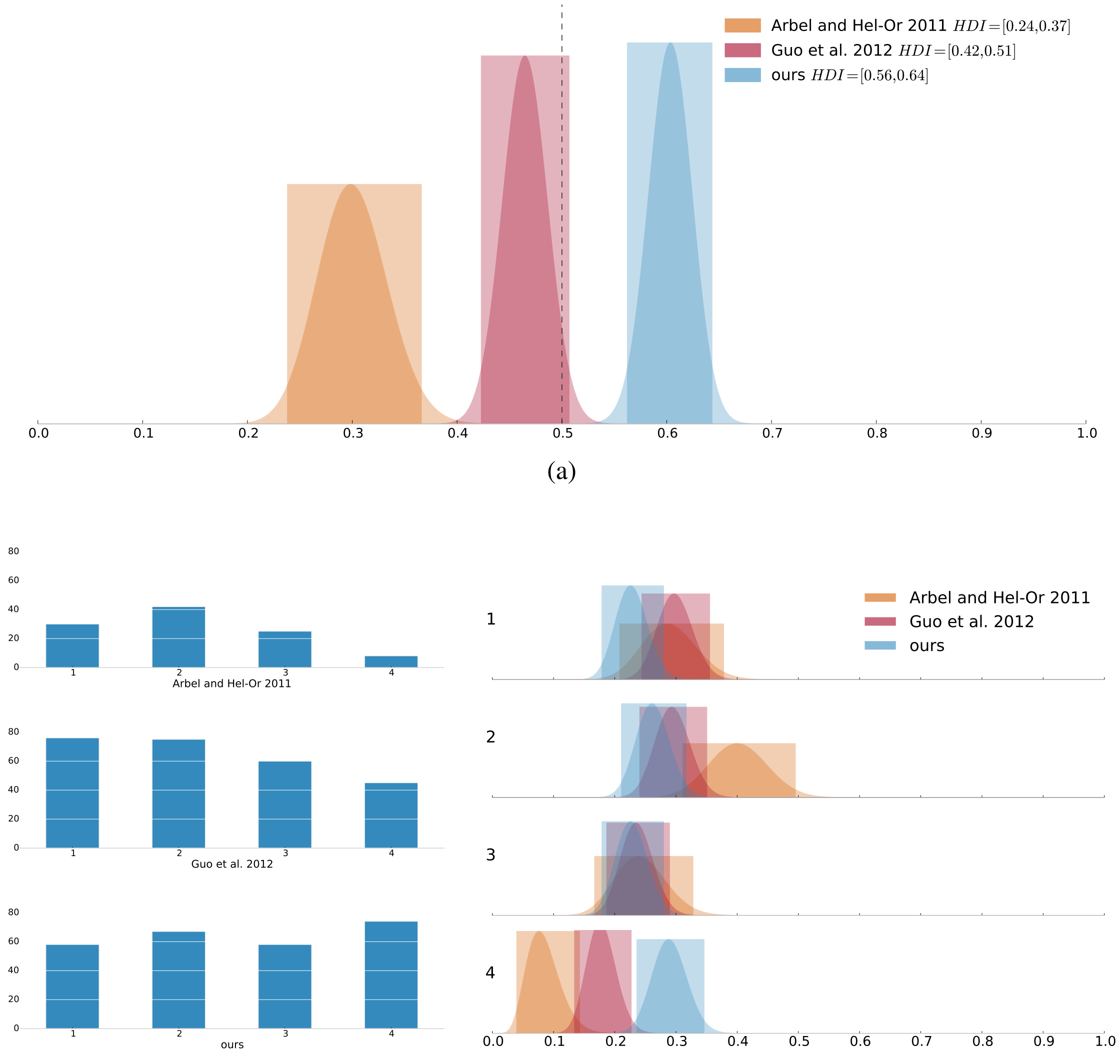
We hypothesize, that the difference in results between previous and new runs was caused by the bias in the audience. Since we were personally involved in recruiting participants for our previous study, and some of them were aware that we were creating an algorithm to remove shadows (but not which images were produced by our method), it is possible that they disregarded the naturalness criteria and made their decisions instead based on shadow removal success. This issue highlights the importance of both careful design of a user study and thoughtful audience recruitment. While enlisting the help of microtask workers raises a number of important issues that do not exist when running user studies in person, it does have the benefit of bringing an audience that is less likely to have a systematic bias such as knowing a common set of people, belonging to a specific group *etc*.

It is also worth noting, that above experiments implicitly trusted the participants' feedback and no filtering of input was performed. As already mentioned, this might result in conservative estimates of differences between the conditions and it is possible that the three distributions would diverge even more under ideal testing conditions.

## 5.5 Discussion

In this section we have presented our recreations of perceptual user studies from 4 SIGGRAPH and ToG papers and how we used imCompadre to verify the results they reported. Re-running the studies required minimal effort, since the authoring interface is tailored to creation of the kinds of studies that are commonly done in the graphics community.

In each instance, in addition to reproducing the experiments, we have found further insights by sampling larger population, performing additional analysis or both. In one case where we were not able to verify the results obtained by the original authors, we believe that the difference can be explained by the differences in the study audience. We believe that imCompadre is a valuable addition into the set of tools researchers use to inspect their results and understand perceptual differences between outputs of



(a)

(b)

(c)

**Figure 5.19:** *User study results for the re-run of the soft shadow removal experiments, after replacing the "naturalness" evaluation criterion in the first phase with "shadow removal success".*

investigated algorithms. Another extremely valuable aspect of imCompadre is the ability to both recreate the past studies and precisely define new ones enhancing the reproducibility of findings in our field.

# Chapter 6

# Conclusions

We have shown a novel method for removing soft shadows from 2D RGB images with minimal user effort, that outperforms previous attempts. While previous methods had to assume a specific model of shadow appearance, we have successfully created a customized machine learning algorithm that effectively builds such a model from training data. We have evaluated our results using a comprehensive user study and observed that such an evaluation is a critical part of image manipulation design. Our evaluation was, so far, the largest published effort to compare soft shadow removal methods and was accompanied by a release of the largest dataset of soft shadow images.

It is our hope that our work will further stimulate development of semantically meaningful methods trained on synthetic data. We have demonstrated that the current state-of-the-art graphics tools allow creation of synthetic, but real-looking data, that can fuel supervised learning-based image processing methods.

While the results we have obtained were proven more perceptually successful than previous attempts, some challenges still remain. Since our method was specifically designed to work on soft shadows, it lacks the ability to deal with narrow penumbrae. An extensions of this method, or perhaps a unified approach with a hard-shadow-specific model, could enable a more general solution. Further, even in the presence of soft shadows, there were cases where our results were not convincing enough to score highly in perceptual experiments. The main reason for these failures we have identified, was the reliance on off-the-shelf inpainting and discarding of valuable information during the initialization step. It is possible that a better, guided initialization method would provide superior performance.

While successfully removing soft shadows was a significant, outstanding problem, increasing the ability to create high quality perceptual user studies might have a larger impact on the community. In this work, we have provided a set of guidelines about how to perform user studies when evaluating novel approaches to image manipulation. By creating a detailed protocol for the most widely used tasks, together with recipes for running the studies and analysis of results, we hope to have established a new standard for how researchers in our field characterize their methods. Finally, we have created a system that allows easy creation of user studies that adhere to the above guidelines automatically. To prove its usefulness, we have recreated perceptual experiments from four graphics papers, in each case succeeding in not only verifying their results, but also obtaining new insights not reported by the original authors.

We hope that our work will drive the adoption of improved practices both in the academic research as well as enabling more rapid feedback in the industrial setting.

Given the ability to create high-quality perceptual studies, the biggest difficulty researchers will face is perhaps creating a correct experimental design and using the right constraints. While the work we have presented and the system itself give several recommendations and encourage good practices, the burden is still on the study author to establish which phenomena to measure and not to create biased experimental setups. Additionally, obtaining and preparing the data necessary to conduct any comparison is a non-trivial task and, consequently, might limit the appeal of the proposed solution.

## 6.1 Future Work

The shadow removal method we have presented relies on effectively learning the relationships between corresponding image patches under different physical conditions. While we have only applied it to shadow removal, it might also be possible to follow the same procedure and, given specialized data, create *e.g.* highlight removal or dehazing tools.

Having proven that large amounts of relevant training data can help in non-parametric unshadowing, revisiting the problem of finding a general, parametric shadow model could bear fruit. We have discovered that previously-assumed one-dimensional sigmoid functions are too simplistic, but it is possible that more complex models could be built to reflect the entire shadow. One of the challenges in building such a model, would be aligning two conflicting requirements: modeling 2D shadow surfaces (as opposed to 1D slices) without losing generality in the face for different occluder shapes.

There are several areas of research that are made significantly more accessible by using imCompadre. The most obvious, and also one that could improve the system itself, would be the measurement of answer reliability from untrusted users and development of prevention and filtering tools backed by data collected from many experiments. Such research could also have broader applications in attention modeling, psychology and psychophysics.

Easy creation of perceptual experiments could also facilitate new discoveries on problems such as evaluation of massive datasets of images. While asking many people about every possible image pair becomes intractable as the number of images becomes large, random selection is often not suitable either. Perhaps borrowing from areas such as active learning, "active ranking" algorithms could be successfully used to guide the system to dedicate most resources to answer difficult questions while not wasting effort on the easily-discriminated cases. Such an application would be another example where Bayesian methods would provide tangible benefits, since both the "under-evaluated" and "over-evaluated" conditions would have automatically assigned and visible confidences.

Further, automating study creation and evaluation would open more possibilities for tools similar to [LCGM09] and [GSCO12], which could allow dynamic, feedback-based, and possibly near-real-time applications to intelligent image manipulation. Finally generalizing the system to work with other data formats, *e.g.* video and 3D geometry would broaden the scope of possible experiments.

# Chapter 7

# Industrial Applications

## 7.1 Introduction

In this chapter, we examine the potential industrial applications of above research and a rough execution plan to bring it to market. We focus on the second part of the efforts, where user study creation system was developed, since it is more widely applicable and seems to have clearer path to sustainability. Nevertheless, it is worth noting that shadows are some of the most important image elements when it comes to the perception and realism of a scene [RLCW01]. Their removal is therefore one of the most fundamental operations necessary for image modification and compositing. Removing shadows, however, is difficult as it requires an accurate model of how they behave, especially when the boundaries are not well defined. This results in a large demand for shadow removal techniques and a significant numbers of web-based tutorials showing how to achieve this manually, but limited supply of reliable, automatic methods. This demand is also confirmed by the fact that Anthropics, a company producing intelligent photo manipulation software and co-sponsors of this research, expressed an interest in automatic shadow removal techniques based on the knowledge of their customers' needs.

## 7.2 Exploring Opportunities for a User Study Authoring System

At first, focusing solely on researchers as the audience for the study creation system did not seem attractive, considering the market size and monetization potential. To widen the appeal we decided to find other people, for whom such a solution might be useful. Out of photographers, fashion retailers, product and user interface designers, professional photographers were the most accessible.

We have spoken to 5 photographers who specialize in portraits, weddings and other family occasions, and commercial product sessions. To each of them we have presented an idea of a system that would allow them to quickly choose the most attractive photographs out of the hundreds or, in some cases, even thousands shot during a session. This would be achieved by utilizing crowdsourcing to iteratively rank a collection of images and select the "best" ones quickly and with minimal resources. Additional benefit of using crowdsourcing over any automatic method, is that the definition of "best" is left up to the person commissioning the study to define, providing great flexibility. The premise was that the tool would dramatically reduce the time the photographer needs to invest in evaluating the shots before even starting any post-processing.

Each of our interviewees confirmed that photograph selection consumes significant amount of their time: depending on the person between 2 and 5 hours per session. Consequently, the perspective of reducing this time seemed attractive, however, our solution did not seem to be as appealing as we had hoped.

The major problems could all be attributed fundamentally to the lack of trust. Firstly, all the photographers expressed concern in the ability of “random strangers on the Internet” to provide valuable feedback for something as objective as photograph attractiveness. Secondly, even assuming that the raters were well-meaning and able, choosing the right shots is a very important ingredient of a photographers’ style and they would be reluctant to hand over the control of it. Further, showing raw, unedited images, even under the conditions of anonymity, is not something that many photographers would be willing to do. Finally, the photo sessions are often confidential and the perspective of a crowd of anonymous workers evaluating photographs from private family events or pre-product launch sessions was a serious problem.

While the problems that prevented us from providing an attractive solution to photographers were fundamental to the approach we took, we discovered that other opportunities in this market might still exist: namely the difficulty of choosing which (finished) photographs to showcase in a portfolio or use on a website seems like something that crowdsourcing solutions might be able to address and something that some of the people we have spoken to tackled with classical A/B testing. The kind of workflow required to achieve this, though, is very generalizable and we have decided to pursue opportunities elsewhere and possibly revisit photographer-specific customizations later.

Another area we have briefly investigated was fashion retail. The value proposition we were bringing forward was rapid and cost-efficient estimates of which items or items’ photographs seem more attractive and appealing to large groups of people. We have presented our approach to the CTO of Secretsales (a marketplace for luxury fashion and homeware items).

The feedback we got, however, was highlighting similar problems that we encountered when trying to appeal to professional photographers. The biggest reservation was that fashion items, especially on the high end of the market, are heavily curated by opinionated individuals and sentiment analysis of the general population is not necessarily a valuable input into their decision making process. Further, many luxury items are not supposed to appeal to too large an audience and are targeted at very narrow groups, whose opinions are difficult to capture, especially using microtask markets.

## 7.3 Focusing on a Known Market

After a brief exploration of other fields, we did not find a niche that seemed to be in a dire need of the solutions we were proposing. It is possible that such a niche exists and we might revisit the exploration in a more exhaustive manner in the future. For now, however, we feel it is important to focus on building a product that is immensely useful, even to just a small group of people. This decision is further reinforced by the general advice given by startup founders and prominent venture capitalists: “solve a problem you have” and relatedly, “build something people want”. Consequently, we have decided to keep building the system from the perspective of a graphics researchers trying to maximize their chances of producing research published in high-profile venues.

Three important, annual events stand out in the calendars of computer graphics scientists and engineers: the SIGGAPH, SIGGRAPH Asia and Eurographics conferences. There are, of course more high-profile events, but these three stand out as the most prestigious. Overall there is around 200-300 computer graphics papers published every year. Based on our analysis of historical submission data, around 10% of papers published, either include a perceptual user study of the type imCompadre currently supports, or should include one. Further, a smaller proportions of research in the computer vision community might find the system useful. This translates directly to a very small initial market of 20-30 customers per year, which does not seem attractive from the perspective of building a sustainable product. We believe, however, that there are significantly larger opportunities that will open, once this segment is catered to.

### 7.3.1 Value Proposition

The overarching goal of our product is to help scientists produce high-quality research. We aim to achieve this by ensuring that creation and reporting of perceptual user studies is as easy as possible. We know from first hand experience as well as from feedback gathered from senior reviewers (see Section 4.1) that good user studies are hard to conduct and consequently hard to find in literature.

Our web-based system allows for very quick creation of commonly-seen user studies for evaluation of images. It makes it possible for graphics researchers to focus on algorithmic challenges, instead of learning about psychophysical aspects of how to conduct perceptual experiments and report their results in a statistically sound manner. Besides being easier, it is also faster, since majority of the setup work is not necessary any more.

Another advantage is standardization and popularization of high-quality practices and reporting methods. Several voices in different disciplines of science recently started criticizing current trends in reporting the results of large-scale experiments: see for instance [SNS11], [MPH12], [MBL+13] and [dWD14]. A large portion of these criticism can be applied to user studies in any discipline: so called "p-hacking", poor reproducibility *etc.* imCompadre makes these mistakes more difficult to repeat and thus enhances the chances of a paper being accepted into a high-profile venue.

While we offer clear benefits to the authors, a wide-spread use of imCompadre will also make it easier for reviewers to determine the merits of proposed work, since similar protocols and standards will be applied widely.

### 7.3.2 Further Expansion

Given a successful adoption by the graphics community, we will attempt to expand the study authoring system to also appeal to the psychology and psychophysics fields. Several factors make it a slightly bigger challenge, but with correspondingly larger potential rewards, since these fields are both large and a "home territory" with a significant numbers of people conducting perceptual user studies.

The challenges will be mainly posed by not being intimately familiar with the field and by existing solutions. Currently there exist offline software packages allowing display of stimulus under exactly controlled conditions with calibrated screens, highly precise timing *etc.* Replacing all this functionality in an online system is likely not possible using current technology. However, reaching a reasonably competitive feature set, while still offering all the benefits of a online system, seems feasible. Further, continuously improving web technologies, notably mainstream adoption of WebGL, will enable further benefits in the near future.

Finally, we hope to eventually act as a public study register, where all experiments, including pilot runs, are by default publicly visible and archived indefinitely. This would be a major step towards eliminating the problem of positive selection bias in publications. Secondly, it would clearly show experiment evolution, making "p-hacking" and similar practices more difficult. Introducing this feature, however, will require a delicate balance between the good of the community in general and convenience for individual researchers. We feel, therefore, that introducing it too early might impede adoption.

#### Initial Marketing

For the evolution stages outlined here, we do not plan any scalable marketing efforts and will instead rely on personal connections and word of mouth. Since the initial market is a small, tightly connected community this should not pose a significant obstacle for adoption.

Additionally, we have at least one scientific publication planned to introduce the concepts in this thesis into the community. Should it be accepted, publication in conference proceedings and/or a presentation will provide ample opportunity to make scientists aware of the benefits of our solution.

### 7.3.3 Financial Feasibility

As outlined above, we do not expect any significant income from the initial stages of running imCompadre. Given the optimistic estimate of 20 users with assumed fee of £10 per month we might expect revenue of around £2000 per year. This is, however, counterbalanced by negligible running costs (web hosting). Development time is not currently factored in, since it is likely to be ran as a side project in the coming year. This means that imCompadre might be ran initially without sustaining losses.

After some popularity is gained, however, increase in prices (for new customers only) will be considered to reflect increased benefits users will presumably get from a more evolved product. Experiments with pricing strategies (*e.g.* tiered pricing for different needs) and more serious marketing efforts might provide enough fuel to invest more effort into building the product. Further, large-scale deals with entire departments or universities might become possible after the platform gains enough trust and mindshare in the community. Such deals open entirely new possibilities for creating a feasible business, but are out of scope for this report.

## 7.4 Risks and Mitigations

We have conducted a brief risk analysis to establish which factors are the most likely to negatively impact development and growth of imCompadre and how to mitigate them.

By far the greatest risk seems to be building a product that is not useful to other people. This would be a critical failure and, short of pivoting radically, impossible to recover from. To mitigate this risk we are actively soliciting feedback from researchers in computer vision and graphics. Several of them have provided valuable input after completing the meta study described in Section 4.8. Further, we are preparing a publication to showcase our work and to prove its value for the community.

The second factor, that will need to be mitigated are existing competing solutions. While there is no direct competition in terms of an online system targeted to the same audience, open source tools like Psychopy and Psychtoolbox as well as the proprietary system surrounding Amazon Mechanical Turk fill similar goals. To succeed we will need to show that for specific cases, our solution is superior and, once that is clear, expand by building new features and revising the product. This risk can only be mitigated by closely following the developments of competing platforms and ensuring that imCompadre never falls behind when it comes to satisfying our very narrow target audience.

Finally, there might be some difficulties convincing researchers that putting trust in a proprietary system in the name of open science is the right thing to do. This is a sentiment that we can relate to and will need to strike the right balance between transparency, open sourcing parts of our code and staying competitive. There is no clear mitigation, but being aware of these issues is important.

## 7.5 Conclusions and Immediate Next Steps

We currently have a product that was used in the creation of two novel publications and is able to reliably reproduce (with significantly less effort) experiments from three already published articles (Chapter 5).

Our most immediate next step is the publication of a paper describing the system itself and proving that it is useful to the community. A dedicated publication in a high-profile venue will be a great tool to drive customer acquisition and a confirmation of the existing need for such a tool.

# Appendices

# Appendix A

# Publications

## A.1 Learning to Remove Soft Shadows

## Appendix B

# Meta User Study

# Meta Study

Thanks for agreeing to take part in our meta user study! There are two exercises here: one that I will walk you through with video, and one that you will have to complete on your own.

The general goal is to establish how easy or difficult it is to create visual user studies using our system and how long it takes. **Please measure the time it takes you to complete Exercise 2.**

In case you have any questions, feel free to write me at maciej@imcompadre.com.

1. Create an imCompadre account here (but keep this window open to continue afterwards): http://www.imcompadre.com/accounts/register/ You will need to confirm your email address by clicking on the link that's sent there.
2. Go through *Exercise 1*, which has a video walkthrough included.
3. Complete *Exercise 2* on your own.

Once you're done please let me know how long it took you to complete Exercise 2 as well as any impressions or comments you have. Particularly, do you feel you could now create a user study using your own data and requirements?

Thanks again!

# Meta Study, Exercise 1

The goal of this exercise is to create a user study comparing different image retargeting (resizing) methods. Several images were processed with different techniques and aggregated into a dataset that you will need to download.

We will now create a user study to discover which method produces best results. To do that we will display the original unmodified image and two modified images and ask our participants which modified image is the best representation of the original.

Now, watch the video below and follow the walkthrough:

1. Download the data.
2. Open your imCompadre study list.
3. Create a new user study with the title: “Evaluating image retargeting methods”.
4. Add a starting survey to estimate participants’ familiarity with image editing. The intro text should read:

> “Thanks for taking part in our experiment! For each question, you will see three images: one original and two modified images. We will ask you which one of the modified images represents the reference image better.
>
> Before we start, just a quick question:”

5. There should be one question: “Have you ever done some image editing before?” with possible answers

   1. Never
   2. A few times
   3. I occasionally edit images
   4. I edit images regularly

6. The study should contain a task that shows a reference image and two modified images and asks the participant to choose the modified image that is the best representation of the reference. You can use the following text for the title:

   > “Which modified image best represents the reference image?”,
   >
   > and description:
   >
   > “An image is represented well, when it contains all the important content and no visible deformations.”

7. For the test data, use the set of images provided above.

8. Ensure that the reference image has a descriptive caption.

9. Ensure that each user answers 10 questions.

10. Go through the “pilot” study you have created and see if there are any things that you would like to change. If so, clone the study, do your modifications and go through it again.

That’s it. Thanks!

# Meta Study, Exercise 2

ⓘ Note

Please start a timer now, so you can measure how long it takes you to complete this exercise.

The goal of the second exercise is also to create a user study, this time to establish whether people can distinguish real images from rendered ones.

Similarly to before, we will create a study asking people to choose one of two images, however, in this case there will be no reference image and instead, we will have an additional constraint. Follow these steps:

1. Download the dataset
2. Open your imCompadre study list.
3. Create a new user study with the title: “Image realism”.
4. Add a starting survey to estimate participants’ confidence in the task before they see the images.

   > “Thanks for taking part in our experiment! We will ask you to answer 10 questions, each time choosing the one of the two images that appears more realistic. Every time one of the images will be a completely real photograph and the other, a similar photograph with some computer-generated objects added afterwards.
   >
   > Before we start, just a quick question:”

5. There should be one question: “How confident are you that you will be able to tell the difference?” with possible answers
   1. Not confident at all
   2. Somewhat confident
   3. I’m sure I will get it right
6. The study should contain one task that shows two images and asks the participant to choose the one that looks more realistic. You can use the following text for the title:

“Which of the two images is real?”,

and leave the description blank.

7. For the test data, use the set of images provided above.
8. Ensure that users are forced to make a choice, i.e. the “No preference” button is not available (use a “forced choice” setting in the task view).
9. Ensure that each user answers 10 questions.
10. Go through the “pilot” study you have created and see if there are any things that you would like to change. If so, clone the study from the study list, do your modifications, activate and go through it again.

Note

Take note of the time it took you to complete the exercise and send it to me

That’s it. Thanks!

# Appendix C

# Program Committee Survey Responses

Raw data (scanned responses of the committee members) can be downloaded from http://www0.cs.ucl.ac.uk/staff/M.Gryka/download/SIGGRAPH-PC-UserStudy-Questionnaire.pdf. Below we present a short summary of what we discovered.

## Quantitative Analysis

**Number of years reviewing graphics papers.**

| | |
|---|---|
| mean | 11 |
| standard deviation | 4.66 |

**Mean number of times on the committee.**

| | |
|---|---|
| mean | 4.96 |
| standard deviation | 3.57 |

**Q2a. Can a user study make or break a paper?**

| | |
|---|---|
| yes | 23 |
| no | 3 |

**Q3. User studies in TOG are:**

| | |
|---|---|
| well carried out | 0 |
| sufficient | 8 |
| not thorough enough | 13 |
| can't remember | 1 |

**Q4. Which analysis method do you prefer: Bayesian, classical or don't mind?**

| | |
|---|---|
| Bayesian | 4 |
| classical | 4 |
| don't mind | 10 |

## Qualitative Analysis

According to the respondents:

| | |
|---|---|
| User studies are important when... | the paper presents visual, perceptual results<br>there are no obvious objective measurements<br>there is an aesthetic aspect to the task<br>users are in the loop |
| I prefer NHST because... | it is more familiar, I understand it better<br>it is more common<br>it is sufficient |
| I prefer Bayesian methods because... | they seem more thorough<br>they seem more honest |
| In addition to (or instead of) perceptual user studies, how else do you validate the results of a novel technique? | my own visual inspection, subjective evaluation<br>lots of results<br>objective metrics (e.g. precision/recall)<br>comparison to previous work<br>comparison to groun truth<br>recording of user interaction<br>opinion of an expert |

# References

[AHO11] E. Arbel and H. Hel-Or. Shadow removal using intensity surfaces and texture anchor points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6), 2011.

[BFSO84] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[BLDA11] F. Berthouzoz, W. Li, M. Dontcheva, and M. Agrawala. A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations. *ACM Transactions on Graphics*, 30(5), 2011.

[BM12] J.T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. In *ECCV*, 2012.

[BPD09] A. Bousseau, S. Paris, and F. Durand. User assisted intrinsic images. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 28(5), 2009.

[BPK13] I. Boyadzhiev, S. Paris, and B. Kavita. User-assisted image compositing for photographic lighting. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2013.

[Bra97] David H Brainard. The psychophysics toolbox. *Spatial vision*, 10:433–436, 1997.

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[BSFG09] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), 2009.

[BT78] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. *Computer Vision Systems*, 1978.

[BVSO03] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, August 2003.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 2001.

[CCL12] Y. Cao, A.B. Chan, and R.W.H. Lau. Automatic stylistic manga layout. *ACM Transactions on Graphics (TOG)*, 31(6):141, 2012.

[CD03] E. Chan and F. Durand. Rendering fake soft shadows with smoothies. In *Proceedings of the Eurographics Symposium on Rendering*, pages 208–218. Eurographics Association, 2003.

[CGC$^+$03] Y. Y. Chuang, D. B. Goldman, B. Curless, D. H. Salesin, and R. Szeliski. Shadow matting and compositing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3), 2003.

[CHM$^+$10] H.K. Chu, W.H. Hsu, N.J. Mitra, D. Cohen-Or, T.T. Wong, and T.Y. Lee. Camouflage images. *ACM Transactions on Graphics*, 29(4):51, 2010.

[CHMA10] L. B. Chilton, J. J. Horton, R. C. Miller, and S. Azenkot. Task search in a human computation market. In *ACM SIGKDD Workshop on Human Computation*. ACM, 2010.

[CKK11] J.C. Caicedo, A. Kapoor, and S.B. Kang. Collaborative personalization of image enhancement. In *CVPR*, 2011.

[CMM$^+$00] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1), 2000.

[CPT03] A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar-based inpainting. In *CVPR*, 2003.

[CRK$^+$13] A. Criminisi, D. Robertson, E. Konukoglu, J. Shotton, S. Pathak, S. White, and K. Siddiqui. Regression forests for efficient anatomy detection and localization in computed tomography scans. *Medical Image Analysis*, 2013.

[Cro] CrowdFlower.

[CSD$^+$09] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh. How well do line drawings depict shape? *ACM Transactions on Graphics*, 28(3), 2009.

[ČWNA08] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi. Evaluation of hdr tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32(3), 2008.

[DCOY03] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. *ACM Trans. Graph.*, 22(3):303–312, 2003.

[dWD14] J.C.F de Winter and D. Dodou. A surge of p-values between 0.040 and 0.049 in recent decades (but negative results are increasing rapidly too). Technical report, PeerJ PrePrints, 2014.

[EE99] W.H. Ehrenstein and A. Ehrenstein. Psychophysical methods. In *Modern techniques in neuroscience research*, pages 1211–1241. Springer, 1999.

[EL99] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1033, Washington, DC, USA, 1999. IEEE Computer Society.

[FDL09] G.D. Finlayson, M.S. Drew, and C. Lu. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 85(1), 2009.

[Fer08] J.A. Ferwerda. Psychophysics 101: how to run perception experiments in computer graphics. In *SIGGRAPH 2008 workshop*, 2008.

[Fis66] Sir R. A. Fisher. *The design of experiments*. Oliver and Boyd, Edinburgh, London, 1966.

[GBI09] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009.

[GDH12] R. Guo, Q. Dai, and D. Hoiem. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[GHP07] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[GJAF09] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*, 2009.

[GSCO12] Y. Gingold, A. Shamir, and D. Cohen-Or. Micro perceptual human computation for visual tasks. *ACM Transactions on Graphics*, 31(5), 2012.

[GTB15] M. Gryka, M. Terry, and G.J. Brostow. Learning to remove soft shadows. *ACM Transactions on Graphics (TOG)*, to appear, 2015.

[Hai01] E. Haines. Soft planar shadows using plateaus. *Journal of Graphics Tools*, 6:2001, 2001.

[HE07] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[HKK12] S.J. Hwang, A. Kapoor, and S.B. Kang. Context-based automatic local image enhancement. In *ECCV*, 2012.

[HSGL11] Y. HaCohen, E. Shechtman, D.B. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011)*, 30(4):70:1–70:9, 2011.

[Hun07] J.D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):0090–95, 2007.

[IPW10] P.G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *ACM SIGKDD Workshop on Human Computation*, 2010.

[JOP$^+$] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python. [Online; accessed 2014-09-06].

[KC10a] A. Kapelner and D. Chandler. Preventing satisficing in online surveys. In *CrowdConf*, 2010.

[KC10b] A. Kapelner and D. Chandler. turksurveyor, 2010.

[KCS08] A. Kittur, E.H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–456, 2008.

[Ken74] J. M. Kennedy. *A Psychology of Picture Perception*. Jossey-Bass Publ., 1974.

[KHA10] N. Kong, J. Heer, and M. Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 2010.

[KHFH11] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):157, 2011.

[KKDK12] J. Kopf, W. Kienzle, S. Drucker, and S.B. Kang. Quality prediction for image completion. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 31(6), 2012.

[KKL10] S.B. Kang, A. Kapoor, and D. Lischinski. Personalization of image enhancement. In *CVPR*, 2010.

[KLW12] L. Kaufman, D. Lischinski, and M. Werman. Content-aware automatic photo enhancement. In *Computer Graphics Forum*, volume 31, 2012.

[KOF13] E. Kee, J. F O'Brien, and H. Farid. Exposing photo manipulation with inconsistent shadows. *ACM Transactions on Graphics*, 32(4), 2013. Presented at SIGGRAPH 2013.

[Kol06] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10), 2006.

[Kru11] J. K. Kruschke. *Doing Bayesian Data Analysis*. Academic Press, 2011.

[Kru13] J.K. Kruschke. Bayesian estimation supersedes the t test. *Journal of Experimental Psychology*, 142(2):573, 2013.

[KT06] N. Komodakis and G. Tziritas. Image completion using global optimization. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 442–452, 2006.

[LBD13] P. Laffont, A. Bousseau, and G. Drettakis. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Transactions on Visualization and Computer Graphics*, 19(2), 2013.

[LCGM09] G. Little, L.B. Chilton, M. Goldman, and R.C. Miller. Turkit: tools for iterative tasks on mechanical turk. In *ACM SIGKDD Workshop on Human Computation*, 2009.

[LG08] F. Liu and M. Gleicher. Texture-consistent shadow removal. In *ECCV*, 2008.

[LLW08] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 2008.

[LM71] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1), 1971.

[LXDR13] Y. D. Lockerman, S. Xue, J. Dorsey, and H. Rushmeier. Creating texture exemplars from unconstrained images. 2013.

[LZW03] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. *Proceedings Ninth IEEE International Conference on Computer Vision*, 1:305–312, 2003.

[MABP10] O. Mac Aodha, G.J. Brostow, and M. Pollefeys. Segmenting video into classes of algorithm-suitability. In *CVPR*, 2010.

[MBL$^+$13] T. Munder, O. Brütsch, R. Leonhart, H. Gerger, and J. Barth. Researcher allegiance in psychotherapy outcome research: an overview of reviews. *Clinical psychology review*, 33(4):501–511, 2013.

[MJCB14] T. Matera, J. Jakes, M. Cheng, and S. Belongie. A user friendly crowdsourcing task manager. In *Workshop on Computer Vision and Human Computation*, Columbus, OH, June 2014.

[MM98] S. Masnou and J.-M. Morel. Level lines based disocclusion. *1998 International Conference on Image Processing*, 3:259–263, 1998.

[MPH12] M.C. Makel, J.A. Plucker, and B. Hegarty. Replications in psychology research how often do they really occur? *Perspectives on Psychological Science*, 7(6):537–542, 2012.

[MTC07] A. Mohan, J. Tumblin, and P. Choudhury. Editing soft shadows in a digital photograph. *IEEE Computer Graphics and Applications*, 27(2), 2007.

[NG06] D. Neumann and K.R. Gegenfurtner. Image retrieval and perceptual similarity. *ACM Transactions on Applied Perception*, 3(1), 2006.

[NMS93] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, segmentation, and depth*. Springer, 1993.

[OAH11] P. O'Donovan, A. Agarwala, and A. Hertzmann. Color compatibility from large datasets. *ACM Transactions on Graphics*, 30(4), 2011.

[OMD09] D.M. Oppenheimer, T. Meyvis, and N. Davidenko. Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology*, 45(4):867–872, 2009.

[OSL$^+$11] D. Oleson, A. Sorokin, G.P. Laughlin, V. Hester, J. Le, and L. Biewald. Programmatic Gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 11:11, 2011.

[OT06] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research: Visual perception*, 155:23–36, 2006.

[PCI10] G. Paolacci, J. Chandler, and P.G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010.

[PCY$^+$01] T.V. Papathomas, I.J. Cox, P.N. Yianilos, M.L. Miller, T.P. Minka, T.E. Conway, and J. Ghosn. Psychophysical experiments on the pichunter image retrieval system. *Journal of Electronic Imaging*, 10(1), 2001.

[Pei08] Jonathan W Peirce. Generating stimuli for neuroscience using psychopy. *Frontiers in neuroinformatics*, 2, 2008.

[PKVP09] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *ICCV*, Kyoto, Sept 2009.

[PSS98] S. Parker, P. Shirley, and B. Smits. Single sample soft shadows. Technical Report UUCS-98-019, University of Utah, 1998.

[RAGS01] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, pages 34–41, 2001.

[RDP$^+$11] M. Reynolds, J. Doboš, L. Peel, T. Weyrich, and G.J. Brostow. Capturing time-of-flight data with confidence. In *CVPR*, 2011.

[RFS$^+$98] B.E. Rogowitz, T. Frese, J.R Smith, C.A. Bouman, and E. Kalin. Perceptual image similarity experiments. *Human Vision and Electronic Imaging III*, 3299(1), 1998.

[RGSS10] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. In *ACM Transactions on Graphics*, volume 29, page 160. ACM, 2010.

[RK11] J.M. Rzeszotarski and A. Kittur. Instrumenting the crowd: Using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User Interface Software and Technology*, pages 13–22, 2011.

[RLCW01] P. Rademacher, J. Lengyel, E. Cutrell, and T. Whitted. Measuring the perception of visual realism in images. EGWR. Eurographics Association, 2001.

[SA93] P. Sinha and E. Adelson. Recovering reflectance and illumination in a world of painted polyhedra. In *ICCV*, 1993.

[SGF$^+$12] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[SL08] Y. Shor and D. Lischinski. The shadow meets the mask: Pyramid-based shadow removal. *Computer Graphics Forum*, 27(2), 2008.

[SNS11] J.P. Simmons, L.D. Nelson, and U. Simonsohn. False-positive psychology undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological science*, 22(11):1359–1366, 2011.

[SPDF13] Y. Shih, S. Paris, F. Durand, and W.T. Freeman. Data-driven hallucination for different times of day from a single outdoor photo. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 2013.

[SYJS05] J. Sun, L. Yuan, J. Jia, and H.Y. Shum. Image completion with structure propagation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 24(3), 2005.

[TFA05] M. F. Tappen, W. T. Freeman, and E. H. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9), 2005.

[VDWCV11] S. Van Der Walt, S.C. Colbert, and G. Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[VO03] L. A. Vese and S. J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19:553–572, 2003.

[WAC07] J. Wang, M. Agrawala, and M.F. Cohen. Soft scissors: an interactive tool for realtime high quality matting. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[Wag07] E.-J. Wagenmakers. A practical solution to the pervasive problems of p values. *Psychonomic bulletin & review*, 14(5):779–804, 2007.

[Wat93] A. Watt. *3d Computer Graphics*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1993.

[WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.

[WBTC05] M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla. Hole filling through photomontage. In *16th British Machine Vision Conference 2005 - BMVC'2005, Oxford, United Kingdom*, pages 492–501, July 2005.

[Wei01] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001.

[WFGS07] H. Winnemöller, D. Feng, B. Gooch, and S. Suzuki. Using NPR to evaluate perceptual shape cues in dynamic environments. In *International Symposium on Non-photorealistic Animation and Rendering*, 2007.

[WTBS07] T. Wu, C. Tang, M. S Brown, and H. Shum. Natural shadow matting. *ACM Transactions on Graphics*, 26(8), 2007.

[YS12] L. Yuan and J. Sun. Automatic exposure correction of consumer photographs. In *ECCV*, pages 771–785. 2012.