Dynamic Image Stacks

David E. Jacobs
Stanford University
dejacobs@cs.stanford.edu

Orazio Gallo   Kari Pulli
NVIDIA
{ogallo,karip}@nvidia.com

Abstract

Traditionally, photography has been driven by a relatively fixed paradigm: capture, develop, and print. Even with the advent of digital photography, the photographic process still continues to focus on creating a single, final still image suitable for printing. This implicit association between a display pixel and a static RGB value can constrain a photographer's creative agency.

We present dynamic image stacks*, an interactive image viewer exploring what photography can become when this constraint is relaxed. Our system first captures a burst of images with varying capture parameters; then, in response to simple touch gestures on the image, our interactive viewer displays the best available image* at the user's focus of attention. *Exposure, focus, or white balance may be slightly compromised in the periphery, but the image parameters are optimal at the selected location.*

Dynamic image stacks turn photograph viewing into an interactive, exploratory experience that is engaging, evocative, and fun.

1. Introduction

Since its infancy a couple of hundred years ago, photography has followed a relatively fixed paradigm: capture, develop, and print. As analog films gave way to digital sensors, image enhancements once reserved to experts working in darkrooms became easy to perform with post-processing software, and the available media to display images have evolved dramatically. Yet, the workflow has mostly remained the same: after the light is captured, and some optional processing is performed, a single, final image is produced.

Today's tendency, however, is for pictures to be taken, processed, and rarely printed (see Section 4.1). Instead, they are viewed on digital screens: computer monitors, cameras, smart phones, tablets, or other potentially interactive display surfaces. Yet, despite the new degrees of freedom that digital displays offer, much of the time they still only display traditional, static images. The only real benefits we enjoy over old printed photographs are improved

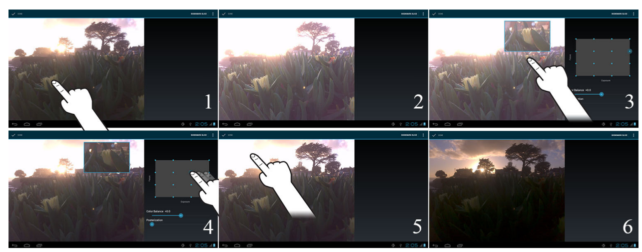

Figure 1. Our dynamic image stack interface. Users interactively explore a scene through a series of touch gestures in our tablet application. The top picture shows a typical interaction session as an **embedded animation**. (Animations can only be viewed in electronic form using a media-enabled PDF viewer such as Adobe Reader.) At the bottom, a series of static frames depicting portions of the same interaction. In the first view (1), the user taps on the foreground flower, and the display is updated to focus and expose the flower correctly (2). For areas where the user wants to specify the viewing parameters manually, a long-press exposes multiple sliders and a zoom loupe to allow for precise control (3); the user specifies that she wants to see a darker picture when she taps back on the same region (4). Finally the user taps the overexposed sky and the display is updated to reveal the sunset (6).

organization, easy distribution, and flexible digital zooming. We should push further and relax the implicit association between a display pixel and a static RGB value, necessary only for printed photographs. Once this constraint is weakened, more compelling and evocative viewing experiences can be explored and developed. Cinemagraphs, for instance, are simple, yet effective visualizations that utilize

Figure 2. Alternative ways to display an HDR scene. From left to right, a dynamic stack exploration shown as an **embedded animation** (a media-enabled PDF viewer such as Adobe Reader is required—two keyframes are shown for print and other electronic viewers), Photomatix HDR default tonemapping [11], HDR Efex Pro 2 default tonemapping [11], and exposure fusion [22]. Stack exploration preserves both global and local contrast, while static tonemapping techniques and exposure fusion must sacrifice one in order to show the other. As a result, the scene's irradiance range appears less dramatic for static approaches than it does for our method.

digital displays to combine static and dynamic imagery [4].

Most computational photography techniques are still designed to enhance the capture capabilities of modern cameras in the context of the static print paradigm, i.e., create *an* image that captures and visualizes a larger portion of the dynamic range, or create *an* image with a larger depth of field. Accomplishing these goals often involves hardware modifications in the form of exotic sensor architectures or unusual optics, as is the case for high-dynamic-range (HDR) imaging [28] or all-in-focus imaging [17]. Similar results can be achieved with standard camera hardware via image-stack-based techniques. However, most of these approaches do not capitalize on the fact that pictures are typically consumed on monitors and, therefore, the output does not need to be a single tonemapped or all-in-focus image.

In fact, similar to most monitors and prints, our eyes can only handle an irradiance range of approximately two orders of magnitude at any given moment [21]; our ability to perceive an overall irradiance range spanning several orders of magnitude is a consequence of our visual system *adapting* to the area we are foveating. A similar observation holds for focus: unless all the objects in a scene are at the same effective distance from the eye, only some of them can be in sharp focus at any time. Inspired by this aspect of the human visual system, we present *dynamic image stacks*, an interactive alternative to static composite images. Dynamic image stacks empower users to interactively explore and perceive different aspects of a scene by visualizing different parts of the capture parameter space in response to user requests. Each image presented to the user is the best one from the available shots within the specified region of interest, maximizing the quality of the overall perception (see Figure 1).

Our method is a conceptual cousin to the Lytro camera's "living picture", which lets a user refocus the image after capture by clicking on areas of interest [18]. Unlike the Lytro interface, or the similar Fosera focal sweep camera's [29], our method does not require any special hardware and thus its impact on the photographers community can be larger.

As an alternative to hardware modifications, stack-based methods collapse a group of images into a single composite that is globally better than any of the input images by some measure. Typical measures include dynamic range [8, 19], depth of field [10], or subject pose and facial expression [2]. An important preprocessing step to these methods is the registration required to compensate for motion of the scene, camera, or both. Global rigid registration is robust, but assumes a planar, static scene. Local, non-rigid registration can handle parallax and scene motion, but can fail in the presence of large changes of the capture parameters. Recent work in HDR compositing can detect and deal with these failures to reduce artifacts, exchanging ghosting problems for a potentially lower local dynamic range [13]; these methods are designed specifically for exposure stacks and cannot be extended to focal stacks easily. Because we never attempt to merge the images, we only require a rough, "fat finger"-accurate registration to interpret user intentions (see Section 2.3.1); therefore, we are resilient to the scene or camera moving during the capture of the image stack.

Even when a pixel-accurate registration is possible, most compositing techniques aim at producing a single output image, and thus must compromise on one aspect of image quality in order to improve another. For example, when an HDR image is tonemapped to a low-dynamic-range (LDR) image suitable for print or conventional displays, either local or global contrast, often both, has to be lowered (as Figure 2 demonstrates). In fact, Čadík et al. found that viewers often prefer doing nothing over tonemapping operators that reduce contrast [6].

In the space of methods that assume perfect, subpixel registration, there also exist viewers offering a subset of the functionality of our viewer. The web viewer by Mantiuk and Heidrich allows the user to operate the equivalent of a slider to select what roughly corresponds to a global tonemapping [20]. More similar to our approach in terms of interaction is HDRView, which tone-maps HDR images to optimize the contrast at the location the user clicks on [26]; together with accurate registration, this viewer requires that the images be combined into a single irradiance map prior to the exploration [8]. Creating an irradiance map from a set of 8-bit images requires to undo the camera's non-linear processing (radiometric calibration); aside from the errors that might be introduced in this stage, we argue that the "black magic" that cameras put into compressing the RAW images to 8-bit is too valuable to be disregarded [9], and is the reason why exposure fusion approaches often produce more natural and compelling results than approaches that tonemap the actual irradiance map.

In this sense, the benefit of dynamic image stacks is two-fold: firstly, they guarantee a globally consistent, high-quality image for a given region of interest, and, secondly, they are more engaging than static images—they are just plain fun to play with. Dynamic image stacks represent a new location in the space of computational photography technologies, offering higher local image quality in exchange for some user interaction; however, contrary to traditional approaches, in our framework the interaction is not a burden, but rather a new, positive addition to the overall experience.

Other interactive techniques provide specialized interfaces for viewing images that are impractical to display using traditional methods. The gigapixel image viewer provides tools for interactively navigating images that are far too large to be appreciated statically [16]; it also adjusts the exposure based on what region of the panorama is shown on screen. The phototourism [25] and unstructured light fields [7] projects exploit the spatial relationships in image data to create engaging exploration experiences. Our system is similarly designed to explore a scene; however, the space we navigate is the space of camera parameters such as focus and exposure instead of the space of camera positions and orientations. All of these systems create a richer viewing experience as the user interactively explores these dimensions. We argue that the interaction itself adds to the viewing experience.

The main contributions we offer in this work are the following: (1) We demonstrate that interactive images can be created from sequential image stacks despite motion from the scene or camera, alleviating the need for specialized hardware. (2) We show that the interactive image paradigm can and should be extended to dimensions beyond focus, of which two examples are exposure and white balance.

Several figures in the paper contain **embedded animations**, which require a media-enabled PDF viewer such as Adobe Reader to play.

2. Design and implementation

The design of the dynamic stack system can be segmented into four constituent stages: capture, interaction, update, and display. These stages are illustrated in Figure 3 and are responsible for generating image stacks, sensing the focus of user attention, providing the image to display, and replaying user-defined "paths" through the viewing parameter space, respectively. We developed a prototype Android application to demonstrate the viability of a full system in practice.

2.1. Capture

The capture stage is similar to that of other stack-based techniques. We capture an $N \times M$ block of images (in our experiments we have taken $N = 4$ different exposures at $M = 4$ focus distances) using the FCam API [1] implemented for an NVIDIA Tegra 3 tablet. Our capture-time metering brackets in single-stop increments about the standard camera's auto-exposure result. The focus distances are evenly distributed in diopter space between a user-specified near-focus and infinity. More efficient strategies for capturing dynamic range [9] and focus [27] are known to the research community, but our simple approach is sufficient for our proof-of-concept implementation.

2.2. Interaction

The interaction stage is tightly coupled with the update stage in a producer-consumer relationship on user exploration requests. Our interface is shown annotated in Figure 3 and in action in Figure 1. From the user's point of view, the interaction is a series of indications of the locus of user attention and reactions by the display to show appropriate image content. There are two obvious ways of determining the locus of attention: gaze tracking and touch-based interaction. Although we have implemented a prototype of both, in this paper we focus on a touch interface. This design is guided by the direct manipulation principle, which suggests that a simple touch-to-explore paradigm should be easy to learn and to use [24]. Moreover, the hardware and computational power requirements for gaze tracking would limit the applicability of our technique to a very small set of modern devices. Finally, touch events are necessary in both cases, for instance to operate the sliders in Figure 3. We discuss gaze tracking further in Section 5.

The system then chooses the best image to show among the captured image stack (see the Update stage below). Typically that would be an image that is best exposed, focused, and white-balanced with respect to the image content at and around the location of attention. As a part of the creative process, the photographer can override the heuristics when they fail producing the desired outcome. A long-press touch exposes sliders the photographer can control to directly express her preference; after a manipulation of the sliders, the system "learns" her intention for the specific location (see Section 2.3.3).

2.3. Update

The update stage consumes user exploration requests produced by the interaction stage, selecting the best possi-

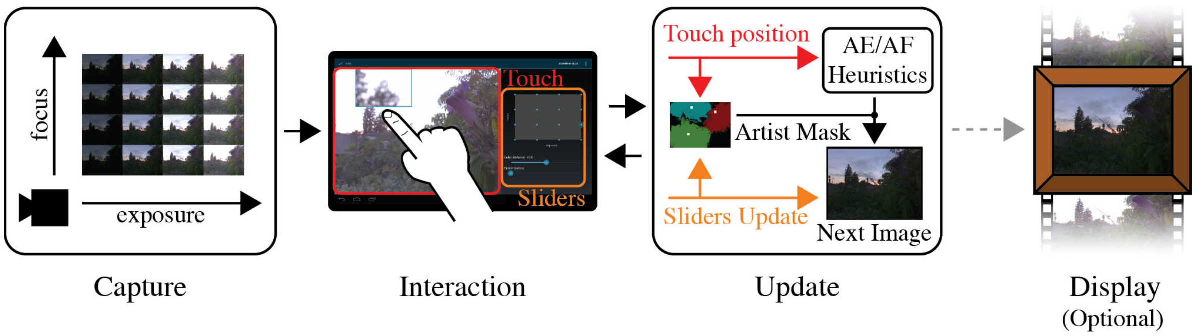

Figure 3. An overview of the dynamic stack system. During the capture stage, the camera records a 2D stack of images by varying focus and exposure; additional versions of the images may be created by applying different white-balancing color transformations. During the interaction stage, the user shows her exploration intent either through direct touch, or by manipulating a 2D grid or 1D sliders corresponding to the viewing parameter space. During the update stage, we interpret the user intent according to auto-focus and auto-exposure heuristics or an artist-specified viewing parameter mask, which is constructed by manual slider corrections. The resulting image can either be presented to the user for further exploration or, optionally, bookmarked for future non-interactive display. During the optional offline display stage, viewers are presented with an animated tour through the viewing parameter space using the bookmarks recorded in previous stages.

ble image to satisfy each request. This requires two separate steps: we first need to register patches to track the object of the user's focus, and then evaluate heuristics to perform the actual selection.

2.3.1 Registration

Given a touch point, we want to display the best image at that point. For locations not affected by camera or scene motion, we know what to compare in the image stack; when the user touches an object that moves, however, the analysis is more challenging—we should track the object across the image stack so we can compare the exposure and focus at the object. Our registration finds correspondences to properly evaluate our heuristics. Figure 4 shows that unregistered patches (marked in red in the figure) may lead to the wrong comparisons; only with registration (patches marked in green) can we select the optimal image. A pixel-accurate registration, however, is not required: we found that the interaction is pleasing as long as the accuracy of the registration is within a "fat finger's" width, i.e., it is roughly as accurate as the user's touch.

The main challenge for registration is the dramatic changes in appearance of the scene accompanying capture parameter variations, which make a naïve nearest neighbor search relatively brittle. To compensate for changes in focus and exposure, we downsample the input images (to address changing focus) and normalize them using histogram equalization (to address changing exposure). We then search for nearest-neighbor patches (roughly fingertip size at 17×17 pixels) in a local window (55×55 pixels) surrounding the corresponding location in the other images of the stack. Our patch distance uses the sum-of-absolute-differences metric. Figure 4 illustrates the quality of our registration on a stack with varying focus and exposure.

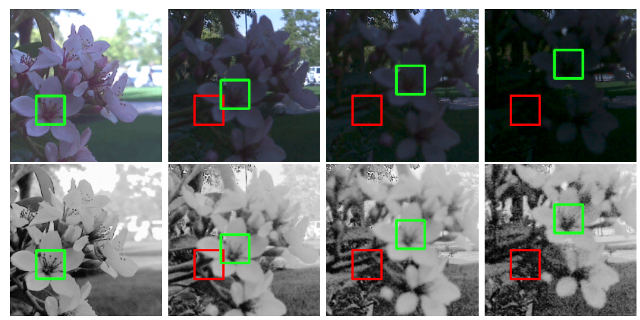

Figure 4. Rough local registration. The top and bottom rows show our registration results overlaid on corresponding windows from a dynamic image stack before and after histogram equalization, respectively. Comparing unregistered patches (shown in red) would lead to incorrect metering decisions, because the underlying flower moves. Registered patches (shown in green) track the flower well, despite changes in exposure and focus.

2.3.2 Exposure and focus heuristics

Our algorithm for choosing the best focus and exposure uses modified versions of traditional auto-exposure and auto-focus heuristics. First, we freeze focus and search for the picture with the best exposure at the touch location. A simple heuristic for exposure would be to choose the exposure level that yields an average gray value near 50%—bright enough to not be dominated by noise, but unlikely to saturate many pixels. In practice, however, this strategy can be too aggressive for objects with particularly high or low albedo (e.g., it could cause excessively darkening of the sky or brightening of asphalt). Accordingly, our heuristic compensates for an object's albedo, approximated as the object's gray value after histogram equalization, by changing the desired average gray value to more closely match the albedo estimate. We find that a 3 to 1 weighted average between a 50%-gray value and the estimated albedo works well for most scenes. Once the exposure has been chosen, we select focus among the properly-exposed patches by maximizing contrast (sum of absolute Laplacians) in the grayscale patch. The exposure-first approach is motivated by the fact that the optimal exposure is relatively independent of proper focus; on the contrary, severe under- or over-exposure may strongly affect the contrast of a patch.

We apply these same heuristics over the whole image to select which image to show when the stack is first loaded.

2.3.3 Capturing subjective preferences

Regardless of the quality of any selection heuristic, the photographer may have a personal preference for a specific region. In our implementation, a long-press event indicates to the system that the artist is requesting to manually select how the region should look; in such situations the system shows a zoom loupe at the location of the touch, and exposes manual controls (see Figure 1): a 2D grid, which serves as a 2D slider allowing simultaneous changes in exposure and focus, and additional sliders for other global operators. (In our proof-of-concept system we only implemented a white balance and posterization slider, but more can be added.) Under the hood, our system builds what we call an *artistic map*: whenever the user moves a slider after a long-press, we create a *control point* in this map at the touch location, indicating the preferred settings for that region of the image. After a control point is created, we would like the recorded, preferred settings to override our default auto-focus and auto-exposure heuristics whenever the same object or area is touched again by the user. Because a full segmentation or object recognition may be prohibitive in terms of computational time, we use instead an approximation of a joint-bilateral filter. Basically, a touch event on pixel (x, y) is converted to (x, y, R, G, B) space; if a control point in the artistic map is within a given distance d_{max} of the current location in this 5-dimensional space, the system loads what the user previously set for the control point, otherwise it uses the standard heuristics. An example of an artistic map can be seen in the Update stage in Figure 3, where the locations of 3 long-press events are shown (white dots) together with the associated regions (red, green, and blue splotches).

2.3.4 Performance

Including registration, the total touch-to-redisplay latency is usually under 250ms. Although code optimizations could reduce this time even further, we find the delay unobtrusive as is. Alternatively, we could precompute the correspondences, focus, and metering results for every patch and integrate them into the artist mask. Preprocessing in this manner would take several minutes to complete, so in order to minimize the delay between capture and interaction, we instead evaluate the update stage on an as-needed basis.

2.4. Display

The display stage consists of a set of optional, non-interactive alternatives to dynamic stack exploration. We provide this capability because interaction is not always possible or practical due to limitations imposed by the display technology or the viewing circumstances—or, because a photographer may simply desire to maintain full control over which aspects and versions of the scene are shown to the viewer. In such situations, the sequence of locations in the viewing parameter space that are selected by a user can be "bookmarked" and subsequently loaded to recreate the same experience for other users.

3. The white balance axis

While not inherently a capture-time parameter, white balance (WB) strongly influences the mood and aesthetic of a photograph. However, a single WB for the whole picture is rarely an optimal solution: in scenes with multiple types of illumination, no single illuminant can be compensated for without causing the other illuminants to create color casts. Recent work in white balance for mixed lighting conditions [12, 5] applies spatially varying white balance to eliminate color casts, but the photographs produced can appear sterile and may be unfaithful to the actual appearance of the scene.

In contrast, a touch-guided white balance interaction can remove color casts in a region of interest while still preserving perceptually accurate lighting conditions in the periphery. This problem is distinct from the auto-focus and auto-exposure problems because the correct white balance cannot be estimated from local information. Instead, we borrow the spatially varying white balance map from Boyadzhiev et al. [5] for use as an oracle. When a region of interest is selected, we apply the white balance appropriate for that region to the entire image. A demonstration of this interaction is shown in Figure 5. If a precomputed white balance map is not available, or if the user prefers using white balance as an artistic tool, the same interaction can be supported manually by adding white balance control points to the artist mask as described in Section 2.3.

4. Evaluation

To evaluate our work, we performed studies exploring three questions related to dynamic stacks: printing habits, interest in dynamic image stacks, and the awareness and impact of motion within the stacks.

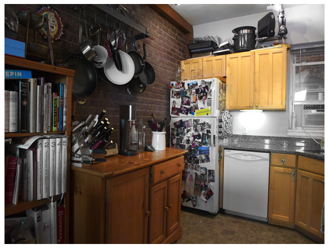

Figure 5. A comparison between dynamic and static white balance techniques. Left: an example interaction of our white balance exploration interface for a mixed-lighting scene, shown as an **embedded animation**. Upon user touch, we look up the best local white balance according to Boyadzhiev et al. [5] and apply it globally. Right: the "corrected" image after the application of spatially varying white balance as proposed by Boyadzhiev et al. Our approach does not completely fix regions where the color of the illuminant changes rapidly (e.g., on curved surfaces such as the vase in the center of the scene), however, for each touch-point, it does induce a percept of the scene that is more plausible than the otherwise remarkable result by Boyadzhiev et al. To best replicate the interactive experience, please view the animation full-screen in Adobe Reader and keep your eyes trained on the tip of the finger. Image courtesy Boyadzhiev et al.

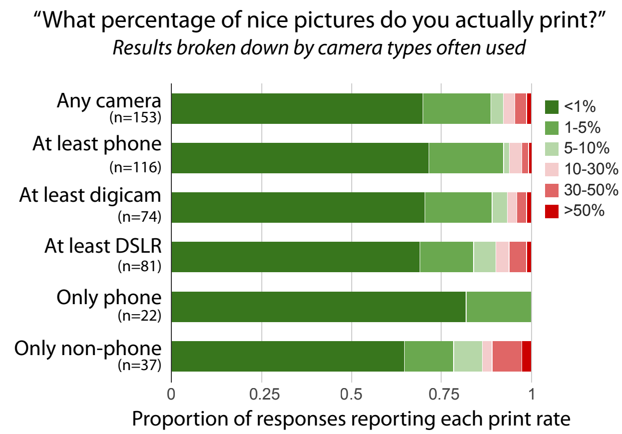

Figure 6. Summary of printing survey responses. Results are compiled for subpopulations defined by their frequently-used camera types. From top to bottom: all participants, participants using non-dedicated cameras (e.g., smart phones and tablets), participants using digital still cameras without interchangeable lenses, participants using cameras with interchangeable lenses, participants using *only* phone or tablet cameras, and participants who *never* use phone cameras. The size of each subpopulation is listed underneath its label. The data suggests that only a small fraction of digital pictures are actually printed.

4.1. Printing habits

In order to verify our impression that prints made from digital snapshots are becoming rare, we performed a small-scale web survey focused on printing habits. Our pool of participants ranged from smart-phone photographers to people who mainly shoot with DSLRs. In total, we received 224 responses. We limit our analysis to the 153 respondents who reported having taken more than fifty photographs in the last six months. Among these, 89% reported printing less than 5% of the pictures they considered worth keeping (i.e., poor photographs are discounted from this total). The reported printing frequency is lower for smart phone photographers and higher for dedicated camera photographers, but overall, printing is rare regardless of photographic skill. Our survey results are summarized in Figure 6.

4.2. Interest in dynamic image stacks

We evaluated the response to dynamic image stacks by performing an informal user study with a diverse group of 14 subjects. The subjects varied in photographic experience from novices who rarely take pictures—but often view them—to (former) professional photographers. The study consisted of a brief demonstration of the Lytro interface, to introduce the idea of interactive images, followed by a hands-on session with our tablet application. We then asked the subjects for their thoughts on dynamic image stacks and the potential role they could play in their photography.

We found that the reaction to dynamic image stacks varies roughly as a function of photographic experience. Although subjects across the experience spectrum said that the interaction was "cool," "fun," and "game-like," they had differing opinions about how dynamic stacks would affect their photographic habits. Veteran photographers, for example, felt that providing viewers with the capability to dynamically explore the scene weakened their creative control over the viewing experience, and thus did not see dynamic stack exploration as a replacement for their normal workflow. They were, however, quite receptive to the idea of presenting dynamic stack tours, in conjunction with regular photographs, to their viewers and customers. One expert subject said she liked being able to save the sequence of locations of interest because it satisfied her "instincts to create a final product." Amateur photographers, on the other hand, were much more enthusiastic about embracing the interactive viewing experience as an end unto itself rather than a means. Nearly all amateur photographers reported they would use a dynamic stack feature if it were available on their camera and would explore dynamic stacks shared with them by friends and family.

4.3. Impact of inter-frame motion

One important question we had for the subjects concerned the presence of inter-frame motion in the image stacks. There may be some motion between the images either because the camera or some scene objects moved, but if the stack is captured quickly, the motions should be relatively small. In Section 2.3.1, we discussed the necessity of registration for correctly updating the display, but we have not yet discussed how seeing inter-frame motion when transitioning between images affects the user's experience. Nearly all subjects (10 of 14) agreed that the inter-frame motion present in the datasets did not detract from the dynamic stack exploring experience. Only a handful (5) of the subjects mentioned the motion without being prompted. One subject even said that the small inter-frame motions "gave the interface more life." The subjects that objected to the motion were specific to point out that the motions only bothered them when using the manual, slider-based interface (the same used to override heuristics, which we provided as an alternative interface for the study participants), or viewing dynamic stack tours. We suspect that the reason motion sensitivity is increased for these situations is because the subject's attention is spread across the entire scene instead of a smaller region of interest, whereas the primary intended interaction mode is for the viewer to study details of the scene in optimal conditions.

These three studies support our belief that dynamic image stacks have a role in future photographic paradigms. Although for many photographers they will not wholly replace the capture-develop-print workflow, it seems clear that they fill an important niche and offer something new for creative exploration.

5. Discussion and future work

Our dynamic image stacks are not meant to replace traditional photography, rather they may coexist with it. In this paper we explored the possibilities offered by the relaxation of the requirement that the product of the photographic process is a static picture. This concept is not entirely new even for fixed focus and exposure. Dynamic images are short animations designed to give life to an otherwise static image. Examples of dynamic elements include simple pan and zoom animations (also known as the Ken Burns effect [15]) and more complicated juxtapositions of static and dynamic content like cinemagraphs, video textures [23], and other similar projects [14, 3].

Although our technique is designed around explicit image stacks, there is no reason why it should not be applied to single images. RAW format images, or films digitized at high bit depth, are particularly well suited to this application due to the many artistic decisions that govern the conversion from *higher*-dynamic-range images to low-dynamic-range images suitable for display. Often a single RAW photograph can yield many equally interesting interpretations. Under the current print paradigm, a photographer must commit to just one at a time. The same rationale motivating our dynamic image stacks can be explored to sidestep this constraint, enabling photographers to tell a story with a single source image. RAW image files can be "developed" into different sequences of pictures that can then be animated, see Figure 7: we can refer to these as *dynamic prints*.

Figure 7. Dynamic prints created from two artists' interpretations of the same RAW image, exploring the possibilities offered by the fact that the image is displayed on a screen rather than printed. The frames of these examples were created with desktop photo editing software as a demonstration, but such expanded editing controls could easily be integrated into our application. To start the **embedded animation**, click on one of the pictures above in Adobe Reader.

Even the electronic version of this very paper is *per se* an exploration of the possibilities offered by dynamic displays. Our animated figures exploit the communicative power of digital displays to better convey our message, just as our proposed dynamic stacks do. Many computational photography research papers are already formatted in such a way that readers have to view a paper electronically to fully appreciate their contributions—due to limited space for high resolution figures or poor image contrast in print, etc.—so we can get this extra expressive power for "free," again not entirely unlike dynamic stacks.

Although well received in our user study, there are fundamental challenges that all embodiments of dynamic image stacks must face. For example, dynamic image stacks will not perform well in situations where the raw stack slices are unsuitable for viewing as is (e.g., dark environments where multiframe denoising is required) or when the photographer wants to capture a particular instant in time, such as the case of sporting events: it is possible that the object is in the desired pose only in frames where it is not correctly exposed or focused. These issues are not unique to dynamic image stacks, however, and present a challenge for all styles of photography.

We also implemented our dynamic stack demo on a desktop computer replacing the touch interface with gaze tracking. For this purpose, we used a Tobii Rex Developer Edition gaze tracker; this gaze tracker reports gaze positions at 30Hz with an accuracy of about 1°. A gaze fixation on a particular region of the image, triggers our heuristics to select the image to show. (In the gaze-based implementation of our dynamic stacks the user cannot override the built-in heuristics.) In future work, we plan to fully investigate how the potential of a gaze-based strategy compares to that of a touch-based approach in terms of usability and user experience.

Another area that we plan to improve in future work is our capture-time interface, so that creating dynamic stacks would be easier for unskilled photographers. Smarter metering algorithms could reduce the number of pictures required to capture a scene. For example, while the dynamic range of a scene can be very large, the dynamic range of a particular depth layer within that scene is often small [9]. Our metering should take advantage of this to speed up capture.

6. Conclusions

In this paper, we have introduced dynamic image stacks, a paradigm for digital photography that can complement the traditional approach. Based on the observation that images are mostly consumed on digital displays, we attempted to break the requirement that a pixel should be associated to a single RGB value.

We have shown that dynamic image stacks offer a fun, engaging alternative to the static photographs of the traditional, print-centric camera ecosystem. Dynamic stack exploration increases the perceived image quality of a scene in exchange for user interaction—allowing the viewer to better understand the scene illuminance and spatial relations between the scene objects.

We believe that, when designed well, user interaction provides another way in which we can improve and extend the relationship between people and their pictures.

References

[1] A. Adams, E.-V. Talvala, S. H. Park, D. E. Jacobs, B. Ajdin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek, M. Tico, H. P. A. Lensch, W. Matusik, K. Pulli, M. Horowitz, and M. Levoy. The frankencamera: an experimental platform for computational photography. *ACM Trans. Graph.*, 29(4), 2010. 3

[2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3), 2004. 2

[3] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi. Selectively de-animating video. *ACM Trans. Graph.*, 31(4), 2012. 7

[4] J. Beck and K. Burg. `http://cinemagraphs.com`, Accessed on March 25, 2014. 2

[5] I. Boyadzhiev, K. Bala, S. Paris, and F. Durand. User-guided white balance for mixed lighting conditions. *ACM Trans. Graph.*, 31(6), 2012. 5, 6

[6] M. Čadík, M. Wimmer, L. Neumann, and A. Artusi. Evaluation of HDR tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32(3), 2008. 2

[7] A. Davis, M. Levoy, and F. Durand. Unstructured light fields. *Comp. Graph. Forum*, 31(2), 2012. 3

[8] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. ACM SIGGRAPH*, 1997. 2, 3

[9] O. Gallo, M. Tico, R. Manduchi, N. Gelfand, and K. Pulli. Metering for exposure stacks. *Comp. Graph. Forum*, 31(2), 2012. 3, 8

[10] S. Hasinoff and K. Kutulakos. A layer-based restoration framework for variable-aperture photography. In *Proc. IEEE ICCV*, 2007. 2

[11] HDRSoft. Photomatix Pro 4.2. `http://www.hdrsoft.com`, Accessed on November 27, 2012. 2

[12] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. *ACM Trans. Graph.*, 27(3), 2008. 5

[13] J. Hu, O. Gallo, K. Pulli, and X. Sun. Hdr deghosting: How to deal with saturation? In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1163–1170. IEEE Computer Society, 2013. 2

[14] N. Joshi, S. Mehta, S. Drucker, E. Stollnitz, H. Hoppe, M. Uyttendaele, and M. Cohen. Cliplets: Juxtaposing still and dynamic imagery. In *Proc. ACM UIST*, 2012. 7

[15] R. Kennedy. The still-life mentor to a filmmaking generation. *The New York Times*, October 19, 2006. 7

[16] J. Kopf, M. Uyttendaele, O. Deussen, and M. F. Cohen. Capturing and viewing gigapixel images. *ACM Trans. Graph.*, 26(3), 2007. 3

[17] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3), 2007. 2

[18] Lytro, Inc. `http://www.lytro.com`, Accessed on November 27, 2012. 2

[19] S. Mann and R. Picard. Being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures. In *Proc. of IS&T*, 1995. 2

[20] R. Mantiuk and W. Heidrich. Visualizing high dynamic range images in a web browser. *Journal of Graphics, GPU, & Game Tools*, 2009. 2

[21] J. McCann and A. Rizzi. *The Art and Science of HDR Imaging*. Wiley, 2011. 2

[22] T. Mertens, J. Kautz, and F. Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Comp. Graph. Forum*, 28(1), 2009. 2

[23] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proc. ACM SIGGRAPH*, 2000. 7

[24] B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8), 1983. 3

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3), 2006. 3

[26] C. Tchou, T. Hawkins, and P. Debevec. Hdrview 1.2. `http://www.pauldebevec.com/FiatLux/hdrview/`, Accessed on March 5, 2013. 2

[27] D. Vaquero, N. Gelfand, M. Tico, K. Pulli, and M. Turk. Generalized autofocus. In *Proc. IEEE Workshop on Applications of Computer Vision*, 2011. 3

[28] G. Wan, X. Li, G. Agranov, M. Levoy, and M. Horowitz. CMOS image sensors with multi-bucket pixels for computational photography. *IEEE Journal of Solid-State Circuits*, 47(4), 2012. 2

[29] C. Zhou, D. Miau, and S. K. Nayar. Focal Sweep Camera for Space-Time Refocusing. Technical report, Department of Computer Science, Columbia University, 2012. 2