Multimedia Experience on Web-Connected CE Devices

Dan Tretter, Jerry Liu, Xuemei Zhang, Yuli Gao, Brian Atkins, Hui Chao, Jun Xiao, Peng Wu, Qian Lin

HP Laboratories
HPL-2010-48

Keyword(s):
multimedia, digital photography, image analysis, image composition, automatic layout, photobook, consumer electronics

Abstract:
Consumer electronics (CE) are changing from stand-alone single-function devices to products with increasing connectivity, convergence of functionality, and a focus on customer experience. We discuss the features that characterize the new generation of CE and illustrate this new paradigm through an examination of how web services can be integrated with CE products to deliver an improved user experience. In particular, we focus on one aspect of the CE segment, digital photography. We introduce AutoPhotobook, an automatic photobook creation service and provide a detailed look at how it addresses the complexity of photobook authoring through a portfolio of automatic photo analysis and composition technologies. We then show how this collection of technologies is integrated into a larger ecosystem with other web services and web-connected CE devices to deliver an enhanced user experience.

External Posting Date: April 6, 2010 [Fulltext] Approved for External Publication
Internal Posting Date: April 6, 2010 [Fulltext]

Accepted for Book Chapter by Springer.

Multimedia Experience on Web-Connected CE Devices

Dan Tretter, Jerry Liu, Xuemei Zhang, Yuli Gao, Brian Atkins, Hui Chao, Jun Xiao, Peng Wu, Qian Lin

Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304, USA
Dan.tretter@hp.com, jerry.liu@hp.com, xuemei.zhang@hp.com, yuli.gao@hp.com, brian.atkins2@hp.com, hui.chao@hp.com, jun.xiao@hp.com, peng.wu@hp.com, qian.lin@hp.com

Abstract

Consumer electronics (CE) are changing from stand-alone single-function devices to products with increasing connectivity, convergence of functionality, and a focus on customer experience. We discuss the features that characterize the new generation of CE and illustrate this new paradigm through an examination of how web services can be integrated with CE products to deliver an improved user experience. In particular, we focus on one aspect of the CE segment, digital photography. We introduce AutoPhotobook, an automatic photobook creation service and provide a detailed look at how it addresses the complexity of photobook authoring through a portfolio of automatic photo analysis and composition technologies. We then show how this collection of technologies is integrated into a larger ecosystem with other web services and web-connected CE devices to deliver an enhanced user experience.

Keywords: multimedia, digital photography, image analysis, image composition, automatic layout, photobook, consumer electronics.

1. Introduction

The consumer electronics industry is undergoing a transformation. Despite the word ‘electronics’ in its name, consumer electronics devices are increasingly differentiated not by the electronics within them, but by the breadth of functionality that these devices provide. These functionalities often consist, at least in part, of applications and services that rely on a connected infrastructure. Most major consumer electronics categories, such as smart phones, televisions, set-top boxes, game consoles, Blu-ray players, DVRs, digital photo frames, and printers, have associated services and content that can be accessed or streamed from the internet.

In fact, consumer electronics are no longer stand-alone devices that perform a single function but can often grow and change depending on the installed applications. Common applications and services include content services (videos, photos, music, podcast, news, weather, or sports), social networks (Twitter, Facebook, or MySpace), games, and productivity tools. The line between content services and applications is also becoming blurrier. Content owners now often provide client applications that adapt content, and interfaces to navigate within that content, to device-specific screen size and capabilities. These cloud-based applications and services are, in many cases, driving the growth of the consumer electronics industry, creating entirely new categories of devices while reinventing many existing categories.

A number of recent publications have introduced various aspects of the transformation in the consumer electronics industry. MIT Professor Henry Holtzman has labeled the next generation of consumer electronic as ‘Consumer Electronics 2.0’ (CE 2.0) and characterizes it as “internet of things” [1]. Ken Wirt of Cisco believes that connectivity will become the major impetus of consumer electronics [2]. In our view, there are a number of factors that characterize the evolution to CE 2.0 from CE 1.0.

One factor is the allocation of development resources. The development resources for traditional consumer electronic devices used to be very hardware centric, with the software or firmware portions playing a minor role of enabling the hardware. In contrast, in CE 2.0, the hardware tends to be based on off-the-shelf components, with software development occupying the dominant share of the development costs. Along the same vein, CE 1.0 products are differentiated by their product specifications. The feature list on the product box tends to be along the lines of clock speed of the processor or the size of the memory it contains. With CE 2.0 products, the differentiation comes from the look and feel of the product.

CE 2.0 products are also characterized by their increasing connectedness. Whereas CE 1.0 devices are stand-alone devices designed to operate by themselves, CE 2.0 products are designed to be placed on a network, whether through WiFi, cellular data, or cabled Ethernet, and are expected to be part of a larger ecosystem. Once on the network, CE 2.0 devices can connect to larger data stores, dynamic pools of applications, other devices, and, through these devices, other people. CE 2.0 devices are also much more context-aware. CE 1.0 products tend to be passive devices that operate based only on direct user input, with little knowledge of their environment. In contrast, CE 2.0 products, through embedded sensors and network connectivity, have the capability to find out about their environment and take action to improve their own functionality. A simple example is the Wii gaming console, which is capable of checking online for new versions of the system software and updating itself as appropriate.

The flexibility of these devices leads to a convergence of functionality, a move from special purpose devices to more general-purpose PC-like devices. For digital photography, this is captured well by looking at the evolution of cameras.

	Capture	View	Share	Create/Edit	Connectedness
Analog camera	X				None
1st Generation digital camera	X	X			None
Current digital cameras	X	X	x	x	Low
Smartphones	X	X	X	X	High

Table 1: Cameras exhibit increasing connectivity and convergence of functionality over time.

One of the boons of convergence is that it allows consumers to move seamlessly among the different experiences that are associated with an eco-system. In the case of digital photography, this means the user can capture a photo, view it instantly, edit it, and then share it via a variety of sharing mechanisms in a matter of minutes. Of course, this same benefit is true in other eco-systems as well. In the case of movies, for example, consumers can research movie reviews, select a film, watch it, and rate or recommend it without ever moving from their seat. Convergence and connectedness go hand in hand in CE 2.0 devices, giving access to ever-larger content libraries while simultaneously allowing the consumer to interact more richly with the content than ever before. One downside to increasing connectedness, however, is in the difficulty of navigating these large content libraries. As connectedness grows, the need for organization and visualization tools grows as well.

What all of these changes lead to, fundamentally, is a shift from a focus on consumer *electronics* to a focus on consumer *experience*. This shift is perhaps best illustrated by Apple's successful smartphone, the iPhone. Contrary to how most vendors promote their products, with specifications of processor clock speed or other numerical measures, Apple differentiates the iPhone from its competitors based on its software and look and feel. The device knows the user's environment (time, location, orientation) and usage habits, enables users to connect with friends and content anytime, anywhere, while the applications in the App Store keep up with the user's lifestyle changes.

As the consumer electronics industry shifts focus from devices to connected experiences, there are a number of implications for multimedia research. Innovation opportunities now span an ecosystem of home and mobile devices, and web-based services. Multimedia content transformation between devices and web services can provide unique differentiations for consumer electronics products. Research topics can be found in new architecture for distributed media processing, content and metadata organization, storage, and distribution, as well as user interaction models. In addition, web technologies can be used to create compelling research prototypes.

In this paper, we explore the implications for multimedia research in the context of Consumer Electronics 2.0. As a case study, we focus on one aspect of the consumer electronic ecosystem – digital photography. We set the context by discussing the current trends and research challenges in digital photography. We then introduce AutoPhotobook, a photobook creation service that addresses one of the common problems in digital photography and highlight its novel features. We show how the service can play a part within a larger ecosystem containing various consumer electronic devices to provide them with the breadth of functionality expected of CE 2.0 products. Within this narrative, we also describe a number of research areas in the intelligent creation of multimedia content, and the deployment of content creation web services.

2. Digital Photography Ecosystem

In this section, we discuss one example of consumer electronics eco-system, digital photography. The digital photography ecosystem supports five main consumer needs centered around photos: capture, storage, viewing, sharing and editing/product creation. The ecosystem started with cameras, computers, and home printers. Over the years, more and more devices and services have joined this ecosystem. Today, photos are captured by mobile phones and cameras, and stored in personal computers, phones, laptops, home media servers, online photo sharing services, and social networks. People further create photo products such as calendars, photobooks, greeting cards, etc, and print them at home, or order through websites and at retail. In addition, digital photo frames are becoming increasingly popular, and are used to display photos at home. Figure 1 shows a high level depiction of the digital photography ecosystem.

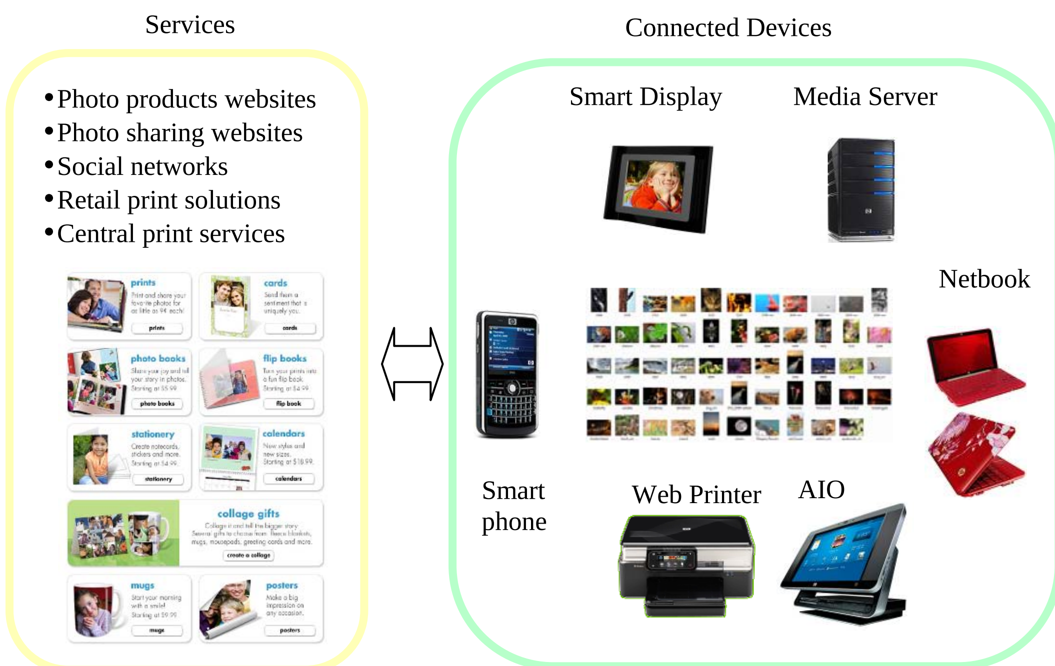


Figure 1: Example multimedia experience - digital photography ecosystem

Consumers face many challenges in dealing with the digital photography ecosystem. Of the five consumer needs mentioned above, capture, storage and simple viewing are all fairly well supported by the current mix of devices and services, but sharing and editing/product creation are more challenging, since they often involve selection of appropriate photos from a large archive. People snap large numbers of photos to capture events in their lives. According to Lyra Research, consumers will capture more than 500 billion photos in 2010. More than half of them will be captured by phone cameras. Of these photos, only a very small fraction is tagged. Most people don't have time to organize and tag their photos. Yet, they desire simple ways to browse and find their photos, and to interact with them to express their creativity.

These trends pose a number of research challenges. Primarily, they are in the area of semantic understanding of photos, and image management and composition to generate creative output.

(1) Semantic understanding. A lot of research is being conducted to understand what a photo is about. The research falls into several categories.

- Object detection: this answers the question of what is in a photo. For example, are there people in the photo? How many people are in the photo? Are there pets, cars, houses?
- Event detection: there is also a lot of interest in finding out the type of events depicted by a photo collection. For example, are these photos taken during a birthday party, on a beach, during a ski trip?
- People recognition: this addresses the question of who is in a photo. Often using face recognition as the base algorithm, one can analyze an image collection and find out answers to questions such as "How many images have John?", "Who appears in the same images with John?" etc.
- Image quality assessment: this can include metrics such as image sharpness, brightness and contrast, eye open/close detection, etc.

This information is often augmented by timestamps and possibly location information, which are provided by the capture devices.

(2) Image management and composition for creative output. Research in this area explores the organization, browsing, search, enhancement, and composition aspects of digital photo collections.

- Organization, search, and browsing: as digital photo collections expand, more and more people are encountering the problem of how to find a certain photo or class of photos. Automatic image tagging using object detection and event detection often does not provide a complete solution. Visual browsing interfaces and multiple search mechanisms can be combined to provide a more powerful solution.
- Image enhancement: early image enhancement algorithms focused on brightness and contrast enhancement. As research in this area evolved, more sophisticated enhancement algorithms can correct redeyes automatically, and even beautify a face.
- Image composition: using a combination of image segmentation and layout algorithms, there is a huge opportunity in generating interesting image collages.

Much progress has been made to address these challenges in recent years. Given the importance of people as subjects in consumer photography, face analysis and recognition is used to automatically group photos according to people and tagged, so that it is easier to browse and navigate photo collections according to people [3]. Contextual information such as time and location are also used to annotate and classify photos [4][5]. While automatic image tagging with arbitrary objects remains an unsolved problem, progress has been made on a semi-automatic approach where a subset of photos are manually tagged and the tags are propagated to the rest of the album [6].

With new generative image composition technologies such as Blocked Recursive Image Composition [7], it is possible to create flexible image layouts with good aesthetics in real time. In comparison with more common layout techniques based on a library of fixed templates, algorithms like BRIC can accommodate a larger variety of photo shapes and sizes on a wider range of pages, since layouts are generated on the fly as needed and do not have to be created in advance. Also, generative approaches have the potential to support more seamless user editing actions, since compositions can be quickly altered or updated in response to user actions. Coupled with new user interaction models such as Mixed Initiative Collage Authoring [8], it becomes relatively easy for ordinary people without special graphical design skills to tell their stories from photo collections. In addition, web content can be added to make the story-telling more compelling [9]. The experience of preserving and sharing memories on paper can be enhanced by linking with digital content [10].

In the next section, we will describe the AutoPhotobook system, which uses advanced techniques in image processing to move digital story-telling to the next level.

3. AutoPhotobook System

One example of an application that could benefit greatly from advances in semantic understanding of photos and image composition is photobook creation. Studies have shown that while photobooks have the highest appeal among all photo merchandise categories, people are deterred by the time and effort involved in making such artifacts. A 2008 PMA survey showed that 47.7% of people who started a photobook and didn't finish said the reason they didn't finish is that the process took too long or was too difficult [11], as illustrated in Figure 2.

When it comes to computer-assisted photobook creation, users want simplicity, quality, customizability and speed. However, conventional solutions leave a lot to be desired, typically trading these four characteristics off against one another, sacrificing simplicity for customizability, or speed for quality. On one hand, the limited options of page layout templates and associated artwork hamper true customizability. All too often the difference between what the author would like to present, and what actually ends up being printed, is a disappointing force-fit. On the other hand, essential difficulties associated with storytelling through personalized photobooks remain; these include photo selection, photo grouping into pages, image cropping, page layout and background selection, all of which can be time-consuming and difficult to optimize. Effective solutions must utilize knowledge of good design practice to present proposed albums that both tell the underlying story and are aesthetically pleasing.

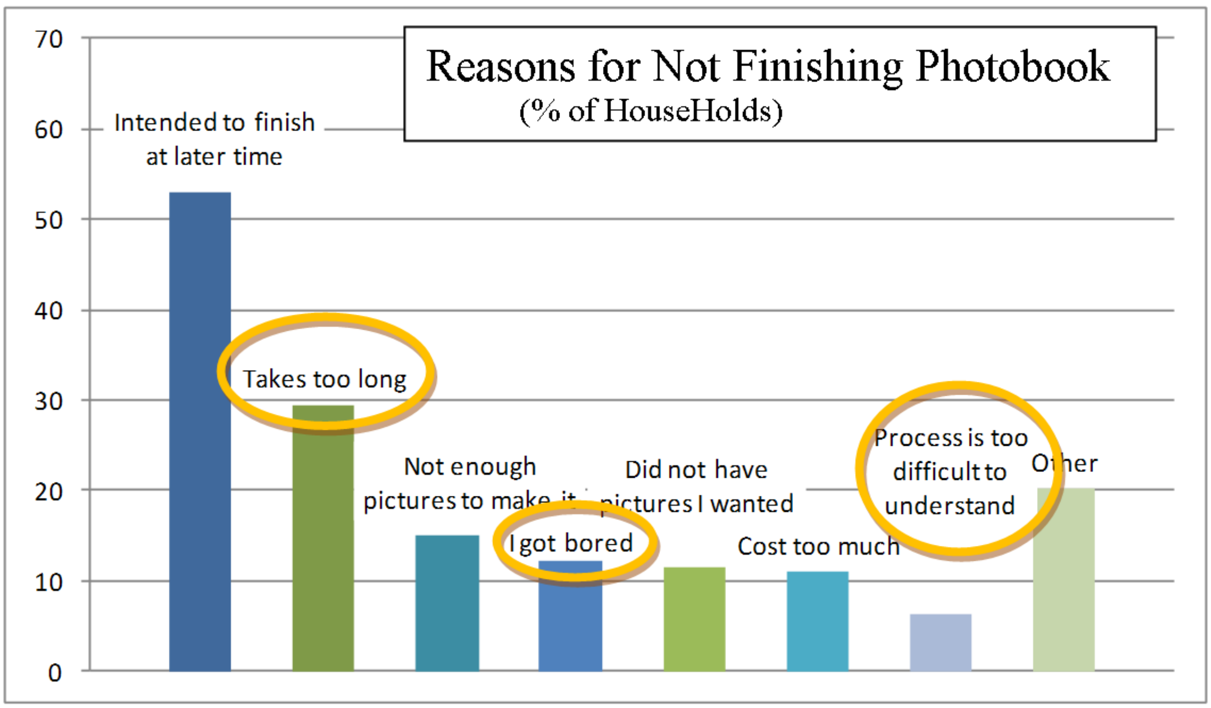


Figure 2: 47.7% of households who did not finish a photobook abandoned it because it was too hard, too boring, or simply took too long.

Today's auto build functionalities are a good start toward resolving some of these difficulties. Considerable recent work has addressed image selection. For example, an automatic photobook generation system was developed in [14] using content-based and context-based image analysis; a scalable image selection system was presented in [13]; and a personal photo management system with the capability to remove undesirable low-quality images was described in [16]. Pagination and layout is another major pain point in the conventional photobook creation experience, but most existing solutions are template based and few explore the possibility of dynamic page layout and background artwork adjustment. Finally, the issue of how to expose all these capabilities in a unified, intuitive user interface remains largely unsolved.

Thus, photobook creation is one of the most technically challenging workflows for solution and service providers, and the workflow from photo collection to final photobook tends to be fragmented and time-consuming. These problems are only magnified when the authoring platform is expanded beyond PCs to other consumer electronics platforms. The HP Labs AutoPhotobook system uses core imaging algorithms to streamline the workflow and automate many of the more cumbersome steps, which both simplifies the creation process on PCs and makes it more amenable to other authoring platforms. This system and its component technologies will serve as a good example of some of the research challenges and approaches to addressing the needs of CE 2.0 products and systems.

Figure 3 shows how the AutoPhotobook system fits into the CE 2.0 environment. Photos captured by digital cameras are typically stored in a computer. A user can access AutoPhotobook through a web browser. Alternatively, a user can also access the system through a thin client on a smart phone, or other mobile computing devices. The AutoPhotobook system performs automatic image selection and pagination, smart artistic background resizing and assignment, and automatic layout. We will discuss in detail how this workflow can enable automatic creation of photobooks, as well as supporting interactive editing, if the user chooses to make additional changes, in the rest of this section. We will discuss how AutoPhotobook can be linked to additional web services such as social networking sites in the next section.

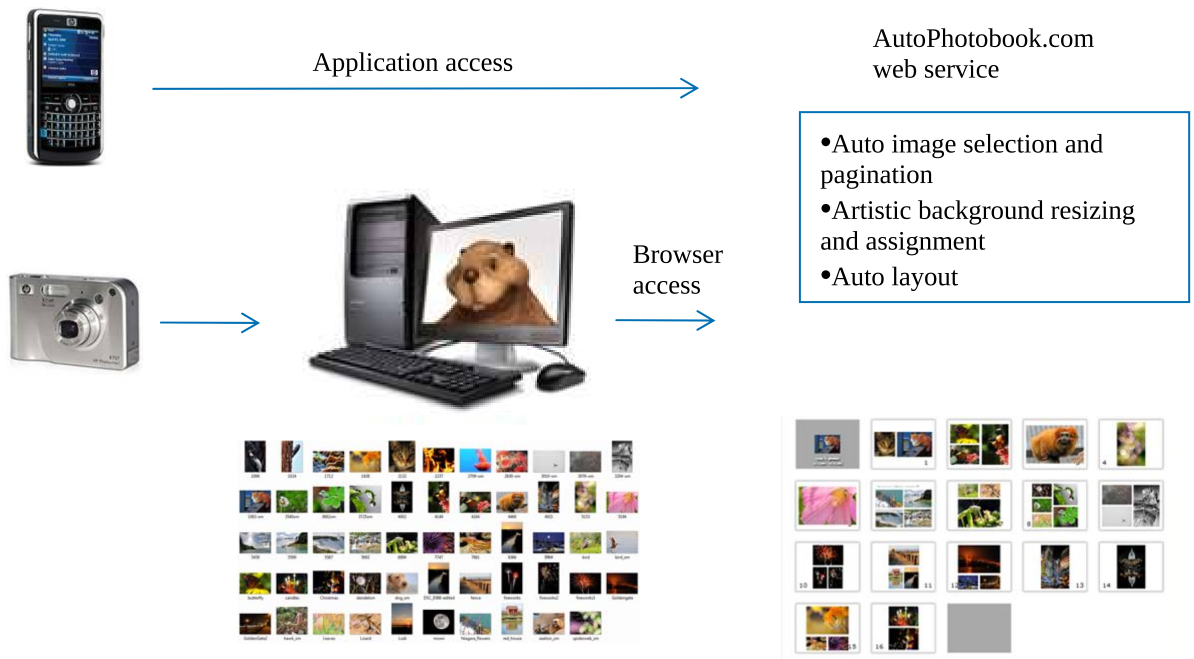


Figure 3: Automatic Photobook Creation

The AutoPhotobook system addresses the complexities of photobook authoring with advances over prior solutions in the following areas: automatic image selection and theme-based image grouping; dynamic page layout; automatic cropping; automatic background selection; design-preserving background artwork transformation; and a simple yet powerful user interface for personalization. Our overall approach is to create a high quality candidate photobook automatically and then allow the user to easily edit and customize the photobook to meet their preferences. We leverage both design knowledge and image understanding algorithms to automate time-consuming tasks like image selection, grouping, cropping and layout. This streamlines the initial creation phase, so the user is never stuck staring at a blank page wondering where to begin. Our composition engine then allows users to easily edit the book: adding, swapping or moving photos, exploring different page layouts and themes, and even dynamically adjusting the aspect ratio of the final book. All of these technologies are delivered through a rich internet application, so the compute-intensive photo analysis algorithms can be carried out in the cloud, and the interface and interaction mechanisms can be run locally on devices with modest computation capabilities.

The block diagram in Figure 4 shows the main components and workflow of AutoPhotobook. Image content analysis is done on the fly through parallel processing when users upload their photos to the system. After photo upload is complete, an automatically generated photobook is presented to the user, along with a simple yet powerful user interface for personalization.

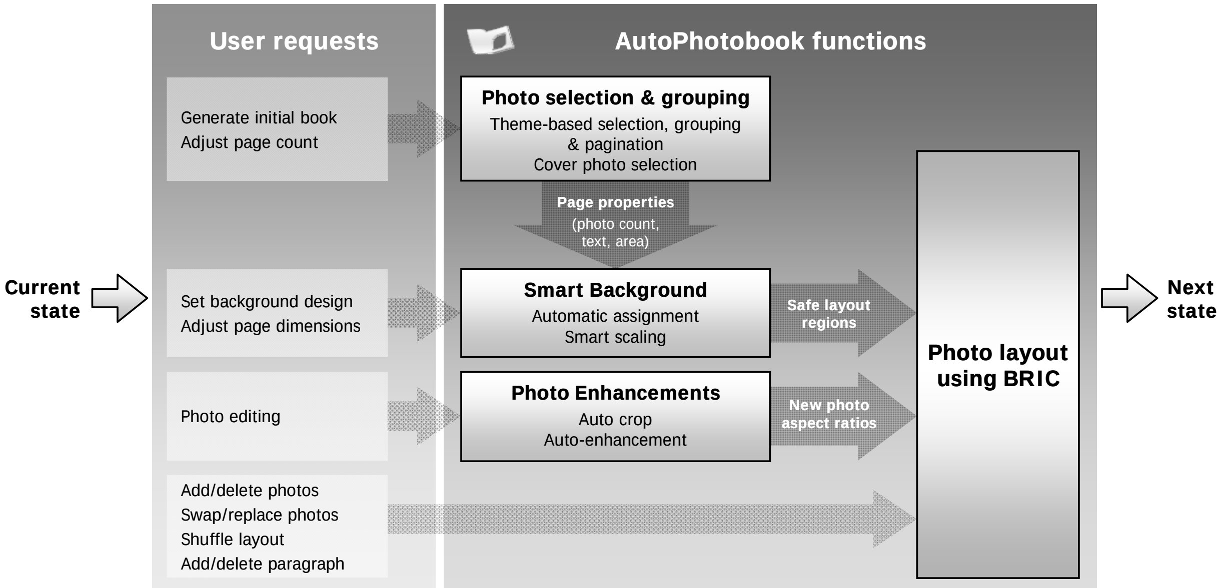


Figure 4. AutoPhotobook system. Note that any user request can be made at any point in the creation process.

Content analysis results are first used to select and group photos to produce a structured representation that helps to tell the story behind the photos. These algorithms are explained in more detail in section 3.1. We then use our Structured Art technology to select and adapt designed backgrounds to the pages of the book, adapting to the size and shape of the book while creating style-consistent page spreads. This technology is covered in section 3.2. Finally, our BRIC layout engine is used to dynamically create custom layout templates for each page that accommodate the book size and the number and shapes of photos assigned to that page. We discuss this technology in section 3.3.

The user is then presented with a finished book for editing and fine-tuning. Photo selection, page assignments, and layouts can all be adjusted with simple drag and drop functionality, which can be supported on a variety of devices. Users can edit individual photos with a single touch or click of the mouse, using toggle buttons to auto-crop and auto-enhance photos as desired. For the auto-crop functionality, we use the HP Labs auto-crop algorithm described in **[17]**, and for auto-enhance, we use HP's HIPIE algorithm **[16]**. The user interface and interaction mechanisms are designed to allow users to quickly explore photobook variants and converge to a desired customized version. We discuss the user interactions and flow in more detail in section 3.4 below. The final result is a photobook creation system that adapts automatically and intelligently to user photos and editing actions.

3.1 Design-driven photo selection and pagination

In order to auto-generate photobooks that serve as good starting points for users, we have conducted experiments with graphic designers to better understand photobook creation and design principles. According to insights from these experiments, we then developed an algorithm that proceeds as follows: First, discard any image that is too blurry or obviously "bad". Second, discard any image that is near duplicate [22] to, but of lower quality than, another image in the collection. Third, divide the remaining photos into pages using "themes", as inferred from time clusters, color, detected faces, and detected locations.

3.1.1 Blurry image removal

In consumer image collections, it's not uncommon to find blurry images. In practice, designers tend to remove these images from further consideration because they generally do not look good in a photobook. To achieve this goal, we have designed a sharpness metric to identify these blurry images.

Blur in images is often caused by camera motion or out of focus. In either case, blur weakens the major edges in images. For example, in Figure 5, the edge strength histograms are shown for two very similar images, one blurry and the other non-blurry. Observe that the edge strength histogram of the blurry image is flatter in shape, and smaller in range than the non-blurry one due to the smoothing effect from out-of-focus blur. This observation leads us to the formulation of a simple sharpness score as the following:

$$Q = \frac{strength\ (e)}{entropy\ (h)}$$

where strength(e) is the average edge strength of the top 10% strongest edges and entropy(h) is the entropy of the normalized edge strength histogram. Intuitively, non-blur images have stronger edges and more peaky edge strength distribution, which leads to large strength(e) and smaller entropy(h), resulting in a larger Q value. We simply threshold this Q value to remove blurry images.

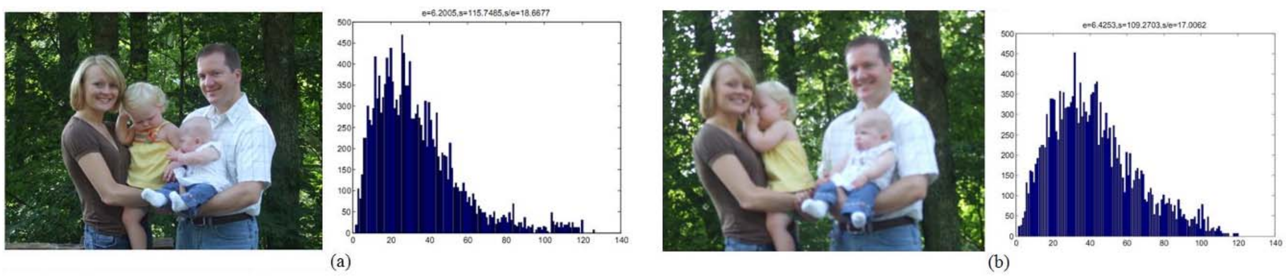


Figure 5. Edge strength histograms of non-blurry and blurry images. The non-blurry image shown in (a) has a 18.7 Q score vs. the 17.0 Q score of the blurry image shown in (b).

3.1.2 Duplicate photo detection

Removing near duplicate photos is generally desirable as users often take multiple shots of the same scene/people only to keep the best one. However, the meaning of "near duplicates" varies depending on the problem domain. In this work, we use the following definition:

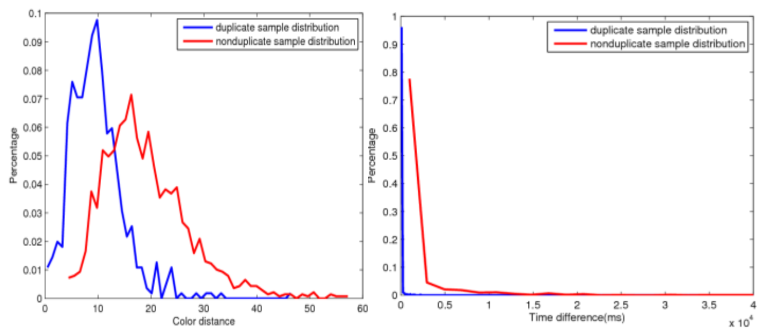
Two images are a "near duplicate pair" (NDP) when they are two snapshots of the same scene, i.e., they are 2D projective transforms of the same 3D scene under the same camera internal parameters but different external parameters. We allow subjects in the 3D scene to have slight non-rigid motions between two shots. Examples of NDPs are shown in Figure 6, where camera rotation, zoom, perspective change and subject motions are observed between the NDP pairs. If an image pair is not a NDP, it is labeled as a "distinct pair" (DP).

Figure 6. Examples of image pairs labeled as near duplicates

One popular solution to this detection problem is to use local feature based image matching such as SIFT [18], because it has been proven to be much more accurate compared to global features such as color histogram.

However, local features are computationally expensive to detect and match, and this approach is too slow for a lot of practical multimedia applications where real-time processing of large image collections is required. We want to address this performance issue without sacrificing matching accuracy. In this work, we develop a novel computation-sensitive cascade framework to tackle this problem.

Near duplicates are “rare events”, and most of the image pairs can be easily be classified as “distinct pairs” (DP) using simple features like color histogram. Although this is a good idea in general, it is hard to determine the optimal number of bins as an effective color representation; therefore we use adaptive color histogram instead as our global image representation, where the number of bins and their quantization are determined by adaptively clustering image pixels in LAB color space. To measure the dissimilarity between two variable-length color histograms, we use the well-studied Earth Mover Distance [19]. Color distance distributions between NDPs and DPs are plotted in Figure 7(a). It can be observed that a large number of DPs can be correctly classified without too many false alarms with a large threshold in color distance.



(3) (b)

Figure 7. Distributions of picture-taking time difference and color distance for NDPs vs. DPs

Modern digital cameras record, in each photo’s EXIF header, a rich set of metadata such as camera model, shot parameters and image properties. Intuitively, if two photos are near duplicates, their EXIF metadata should be fairly similar to each other. In Figure 7(b), we plot out the distribution of the difference in picture-taking time for both NDPs and DPs, and it can be seen that, similar to color distance, this feature clearly has discriminative power to identify true DPs with a large time difference threshold without many false alarms.

Obviously, these three types of image features (local feature, color histogram and EXIF) are very different in their discriminative power and the cost of extraction. Generally speaking, features with higher discriminative power require higher computational cost. Most prior work assumes features are pre-extracted, stored and indexed in a database. In many online applications, however, extracting all the features is simply too slow. Therefore we adopt a minimalist principle, where we only extract features that are absolutely necessary for accurate classification. Since NDPs are rare events compared to DPs, this “on-demand” feature extraction scheme should result in significant saving in computational cost.

In their seminal paper [20], Viola and Jones proposed a cascade classifier learning framework to quickly reject image patches that are non-face like. They use Haar-like features along with integral images, making the cost of extracting different Haar-like features essentially constant. In our case, however, we have features that are widely different in computational cost. The basic idea is to use cheap features as much as possible to classify image pairs, and to only extract the more expensive features when those cheap features cannot determine if they are duplicates. Building on Viola and Jones’ original work, we extend their cascade training algorithm to be “computation-sensitive” as follows:

Formally, given a set of training samples $X = \{X+, X-\}$, where X+ are positive samples and X^- are negative samples, represented in a feature space $F = \{f_1, f_2, \ldots, f_n\}$,

1. Cluster features based on their computational cost into m categories, i.e., $F = \{f^{(1)}, f^{(2)}, \ldots, f^{(m)}\}$, where $f^{(i)} = \{f_{i1}, f_{i2}, \ldots, f_{ij}\}$ and $\forall f_{ij} \in f^{(i)}$ has similar computational cost. Note that feature clusters are ranked, so that the cost of computing $f^{(u)}$ is cheaper than $f^{(v)}$, if $u<v$;

2. For i = 1:k

a. Bootstrap X to $\{X_t^+, X_t^-\} \cup \{X_v^+, X_v^-\}$ and train a stage boosting classifier C_i using feature set $f^{(1)} \cup ... \cup f^{(i)}$ on training set $X_t^+ \cup X_t^-$.

b. Set threshold T_i for C_i such that the recall rate of $C_i(T_i)$ on the validation set $X_v^+ \cup X_v^-$ is over a preset level R close to 1 (this is to enforce the final classifier has a high recall).

c. Remove from X the samples that are classified by $C_i(T_i)$ as negative.

3. The final classifier C is the cascade of all stage classifiers $C_i(T_i)$, i = 1,…,k.

Note that stage classifiers are trained on progressively more expensive, yet more powerful feature spaces. At test time, if a test sample is rejected by cheap stage classifier $C_i(T_i)$, none of the more complex stage classifiers $C_j(T_j)$, j>i, will be triggered, therefore avoiding the extraction of more expensive features.

In order to evaluate the performance of the cascade learning, we randomly downloaded 975 image pairs of personal photo collections from Picasaweb and manually labeled them as Near Duplicate Pairs (NDP) or Distinct Pairs (DP). We split the dataset into the training set with 475 image pairs and testing set with 500 image pairs. The features are ranked according to their computational cost as: EXIF feature, global color histogram and local feature matching score. They correspond to the three stage classifier in our detector.

As a test of the efficiency of our approach, we ran the 500 test samples over our trained computation-sensitive cascade classifier, and observed that only 419 samples extracted color histogram feature and 297 samples extracted the most expensive local features. On average, computing EXIF features takes about 0.0001s, the global histogram feature takes about 0.49s, and the structured local feature matching takes about 1.16s, thus the total time for classifying all 500 samples using our cascade classifier is about 500*0.0001+0.49*415+1.16*297= 455.145s, compared to 500*(0.0001+0.49+1.16)= 825.05s if all features are extracted for all images. This is almost a 2X speed up. Notice that the speed improvement is affected by several factors, for example the number of true duplicates in the dataset, the similarity between image content, etc. The more NDPs in the dataset, the more computation is needed, because duplicates need to go through all the cascade stages before a correct prediction can be made. Considering that in real applications, the ratio between NDPs and DPs is far lower than our experimental setup (<30% of random image pairs are NDPs), the speed up therefore should be much larger in practice.

The precision/recall metrics of our approach are also measured and shown in Figure 8. We compare the computation-sensitive cascade's performance against (a) non-cascade boosting classifier with all features and (b) with each individual feature alone. As can be observed in the figure, our computation-sensitive cascade gives much better performance than using only each single feature. Compared to regular boosting with all features, it gives a similar classification accuracy, while at the same being much more computationally efficient.

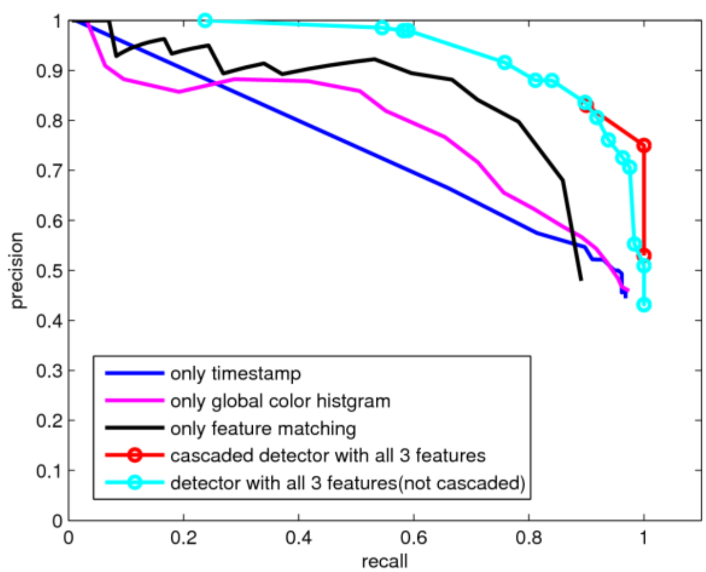


Figure 8. Comparison of the classification precision and recall between computation-sensitive cascade (red); each individual feature (blue, magenta and black) and regular boosting but without cascade (cyan).

3.1.3 Theme-based pagination and layout

Once the low-quality images and duplicates are removed, we then try to cluster the remaining images by "themes". Time based photo clustering algorithms have been extensively studied in literature [21]. Our algorithm goes beyond that and takes advantage of additional high-level semantic features for better image grouping. The themes concept is used frequently by graphics designers when they create artifacts from image collections, which generally means certain similarities in dimensions such as time, color, people and places. We capture these dimensions using the following functions:

(1) Time distance function Dt(x,y), defined as the absolute difference between the photo-taking time of image x and y.

(2) Color distance function Dc(x,y), defined as the Earth Mover Distance [19] between the color clusters extracted from image x and y.

(3) Face distance function F(x,y), defined as the average distance between faces detected in image x and y.

Due to the variety of similarities one can measure between images, simply computing all metrics on all possible pairs of images in the sequence could lead to a very high computational cost. To reduce the cost, we take advantage of the following locality assumption: images that were taken closer in time are more likely to be grouped into a page than images that were taken further apart in time. This locality property motivates us to restrict the expensive theme-based clustering process within a certain time window, therefore reducing the computational cost significantly. We describe our theme-based clustering algorithm as the following:

(1) Partition the image sequence into non-overlapping image sub-sequences using a predefined time gap T.

(2) Within each image sub-sequence, a theme graph is constructed by treating all images as nodes, and edges between nodes represent their thematic distance, measured by a linear combination of Dt(x,y), Dc(x,y) and F(x, y).

(3) The graph is then pruned by removing edges whose distances are over a tunable threshold, which is set by the end user to control the number of output pages.

(4) Finding theme clusters is then cast as finding non-trivial cliques in these theme graphs.

Once the theme clusters are found, we simply map each cluster into a photo page. The resulted pages are then fed into the BRIC layout engine with relative weights to reflect their "importance" in the group, and the BRIC engine will determine the size of the photos according to their importance scores.

3.2 Artistic background resizing and assignment

Graphical artwork is increasingly used to enhance the photo layout and provide a theme to the event or story told in a photobook. It is an important part of high quality photobook solutions. Such artworks are usually prepared only for a fixed page dimension by graphic artists. Even though multiple versions of the same design may be prepared manually for several "standard" page sizes, it is a time consuming task, and difficulty arises when the user's desired page size is not one of the standards. This limits the use of the artwork. In addition, for each theme, there is usually a set of related artwork, and when applying them to a photobook, challenges arise in managing the consistent, harmonic and interesting appearance of designs across facing pages and the whole book. To this end, we propose a method that addresses some of the challenges: it automatically resizes the artwork to different paper sizes; it orchestrates relative positioning of design elements with the photo placement by computing a new allowable photo layout region for each page; at book level, a theme grammar specifies usage constraints of each background artwork.

3.2.1 STArt Design for automatic resizable artwork

There have been some prior works on automatic image retargeting for different display or page size. The seam-carving technique described in [23][24][25] was able to automatically resize images to different aspect ratios by removing low energy paths. This approach works well for resizing natural scene photographic images. However, because background art often contains patterned graphic elements with strong regularity and symmetry, directly applying this technique to the whole image could introduce severe and obvious distortion and artifacts. Other proposed resizing methods [26] also don't address some of the more complex graphic objects in background art images, where page resizing sometimes not only requires the scaling of the graphic objects but could also require addition, abstraction or synthesis of new graphic objects.

Motivated by various theme art samples created by graphic artists, Birkhoff's original aesthetic order [27] and general aesthetic measure [28] in document layout, we have developed a Structured Art (STArt) design method. It allows automatic transformation of the background art to new aspect ratio while preserving the original aesthetics by preserving the semantics, symmetry, alignment, continuity, connectivity, uniformity, regularity and the relative positions and size of the elements on the page and to other objects in the original design. This method is facilitated through a design language and an associated transformation algorithm.

The design language is an XML description of the artwork in the form of a list of design elements as its semantic structure. For each element, there is a content object and an array of attributes, such as type, geometric layout position, style and alignment. The content of the element is usually an illustration in the format of an image or another STArt design. For each element, its type can be "stretchable", "non-stretchable" or "repeating". The position attribute can be one of the ten locations as "top", "bottom", "left", "right", "middle", "area", "topLeft", "topRight", "bottomLeft" and "bottomRight". The "style" and "alignment" attributes describe how the "repeating" patterns are placed on the page. For example, a repeating element in "perfectFit" style draws an integer number of patterns within the described region and along indicated alignment direction, and an element in "texture" or "looseFit" style, draws patterns repetitively until it runs out of space, in this case a pattern can be partially drawn. A detailed description of the language and structure can be found in the reference [29][30].

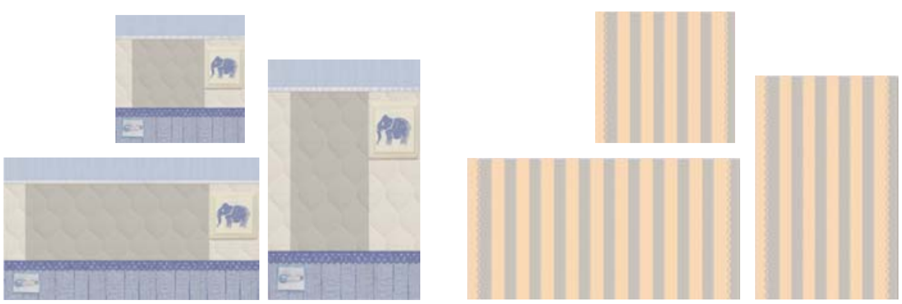
Transformation algorithm:

The transformation algorithm takes elements' attributes into consideration and dictates how each element should be scaled and translated on a page during resizing. Symmetry and alignment are observed by keeping the relative positions of design elements to the edges of references indicated by their location attributes. For example, a "topLeft" positioned element keeps its relative distance to both top and left edges of the page; a "left" positioned element keeps its relative distance to the left edge in horizontal direction while preserving the distances (for stretchable) or ratio of the distances (for non-stretchable element) to the top and bottom edge in vertical direction; and a middle-positioned element preserves the distances (for stretchable element) or ratio of distances (for non stretchable) in both horizontal and vertical directions. Therefore a corner element always stays in the corner; an element that was symmetrically placed in the middle of the page will stay in the middle. Elements aligned in the same direction as their position attributes stay aligned. Continuity and connectivity are also preserved with similar mechanism and with different types of elements. For a stretchable element, it is the distance to the referenced edge in one direction and distances to the other two edges in the other direction are preserved with the same scaling factor as the non-stretchable elements on the page. Therefore, for example, for a stretchable "top" positioned element, when page resizing stretches more in horizontal direction, its width scales up. Its distances to the left and right edge and its height are scaled proportionally as the page scaling in vertical direction (the smaller page scaling factor). Therefore, if this element is originally connected with a "topRight" or "topLeft" non-stretchable corner element, which, during resizing, also scales proportionally as the page in the vertical direction with a locked aspect ratio, after page resizing, it remains connected with the corner element. Uniformity and regularity are achieved through repeating pattern. As a page is resized, the number of patterns appearing on the page can be recalculated to preserve the style.

Examples are shown in Figure 9. A sample design file is shown in Figure 9(a). It includes various stretchable, non-stretchable and repeating elements. The artwork described by the design file is shown in Figure 9(b). Shown in Figure 9(c) is a more complex repeating design, where integer numbers of stripes are placed in the horizontal direction and stripes can continuously grown or shrink in the vertical direction.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<BACKGROUND  x="0.0" y="0.0" width="523.0" height="380" leftMargin="54.605" rightMargin="128.98798"
            topMargin="70" bottomMargin="105"  >
  <Element type="stretchable" position="top" x="0" y="0" width="380" height="65">
    <image source="bluefabric.jpg" height="50.0" width="100.0"/>
  </Element>
  <Element  type="repeating" position="middle" alignment="both" style="texture" x="0" y="65" width="379" height="221">
    <image source="quilted.png" height="112.0" width="72.0"/>
  </Element>
  <Element  type="repeating" position="bottom" alignment="H" style="looseFit" x="0" y="287" width="379" height="91">
    <image source="bluestitch.jpg" height="204.0" width="130.0"/>
  </Element>
  <Element type="repeating" position="top" alignment="H" style="looseFit" x="0" y="58.948" width="380" height="13.824" >
    <image source="sm_dots.png" width="30" height="13" />
  </Element>
  <Element type="repeating" position="bottom" style="looseFit" alignment="H" x="0" y="270" width="379" height="32" >
    <image source="crochet.png" width="187" height="32.399" />
  </Element>
  <Element type="non-stretchable" position="topRight" x="243" y="85" height="136" width="136">
    <image source="stamppad.png" height="136" width="136"/>
  </Element>
  <Element type="non-stretchable" position="topRight"  x="266" y="106" width="93" height="93" >
    <image source="elephant.png" width="93" height="93" />
  </Element>
  <Element type="non-stretchable" position="bottomLeft" x="10" y="285" width="94" height="94">
    <image source="safetypin.png" width="94" height="94" />
  </Element>
</BACKGROUND>
```


(a)



(b) (c)

Figure 9. Example of the STArt designs. A sample XML design file is shown in (a). The artwork described by (a) is shown with different aspect ratios in (b). The shaded areas shown in (b) are the allowable photo layout regions. An artwork in (c) shows a repeating pattern.

A user study was performed using 18 designs which have been laid out for two different aspect ratios by graphic artists, a square layout and a landscape layout of 11.25 x 8.5. Based on the square layouts, we created XML descriptions of the designs and automatically resized them to the landscape layouts based on our method. Ten users rate the auto-resized layouts and the artist-resized layouts based on visual appearance and usability. The "no-difference" trials were evenly split between the "artist-better" and the "auto-better" groups. The proportion of "artist-better" trials were 0.48, slightly lower than the "auto-better" group, but the difference is not significant (two-sided binomial test, 180 trials from 10 subjects and 18 designs, $p>0.05$). Therefore, the auto-resized theme art images give comparable results to manually resized ones.

3.2.2 Dynamic photo layout region on the page

As the artwork resizes, STArt design computes a new allowable photo layout region according to the new page dimensions. This area is computed in two ways. It is first computed by scaling the margin proportionally to the smaller one of the horizontal and vertical scaling factors. It is then further modified based on the bounding boxes of the elements labeled as margin pushers. The content of the margin pusher attribute of the element indicates which margin it influences. In 4(b), the shaded region in the middle shows the allowable photo region for difference page sizes.

3.2.3 Theme Grammar for photobook

The design language is also extended to describe design themes containing multiple related backgrounds. The theme grammar specifies usage constraints of each background, such as suitability for text display or multiple-photo layout, appropriate text color, left or right side, and which backgrounds are compatible with each other on facing pages. These constraints ensure the consistent and harmonic appearance of the photobook. For each background, there are usually a few candidates for its facing page. The final selection of the facing page is a random selection among the choices. This random process makes the facing page more interesting. Backgrounds can be automatically assigned according to these constraints with a single click. Figure 10 shows an example of a small photobook with automatic background artwork assignment with a user-selected theme. Facing pages (pairs in black rectangular frames) have been assigned matching background designs. Users can override each selection easily, but the automatic assignment allows them to quickly change the design theme of a book with minimal effort.

Figure 10. Example of automatic background assignment according to theme grammar.

3.3 Automatic layout

In most photobook authoring solutions, the user is provided with templates, each having fixed regions into which photos and text may be inserted. However, in AutoPhotobook there is no template library. Instead, a module called a "layout engine" creates and edits page layouts in response to commands from the user. The layout engine is based on a photo layout method called Blocked Recursive Image Composition, or BRIC, which is introduced in [7]. The use of BRIC as a collage layout engine is documented in [8]. AutoPhotobook borrows most of its creation and editing functionality from [8] directly, but one difference is that AutoPhotobook supports placement of a text block on the page. Aspects of text including typeface, point size and line spacing are all regarded as fixed, although as discussed below alternative presentations of a text block usually differ in how the text is broken into lines.

3.3.1 Prior related work

As mentioned above, most photobook authoring solutions rely on templates. Templates are usually generated by graphic artists, and as such may offer some guarantee of aesthetic quality. However, template libraries can be costly to generate and manage. If a template library is too big, then it may be burdensome for the user to navigate. If it is too small there may be instances where the user wishes to present a certain set of content, but the available templates are inappropriate with respect to one or more of the following: number of photos; photo aspect ratios; photo sizes and positions; whether text is allowed; and the maximum length of inserted text. This suggests there is a need for more automated methods that support the creation and editing of such composites.

Some prior work in automated *photo* layout is reviewed in [7] and [8]. Other recent work includes the automatic method of [31], which uses a Bayesian formulation to optimize layout such that a visually important subset of each photo is visible; and the photo layout optimization techniques of [32]. There is also significant prior work in automated creation and editing of *mixed-content* layouts, which we define as including both text and images. We are not aware of any other work specifically geared toward composites with one text block and any number of photos. However, a generally related area is that of automatic document layout; a recent survey is given in [33]. Much of this work focuses on pages that are primarily textual. In AutoPhotobook pages that have text *at all* tend to have a much greater proportion of image content. Moreover, many document layout and adjustment techniques rely on the premise that the content will respect some prescribed structure, such as a column of a certain width, or a given tabular arrangement. In AutoPhotobook, there is no such specifically prescribed structure. Within these restrictions, there are a few potentially relevant approaches including some based on genetic algorithms [34][35][36], and others introduced by de Oliveira [37]. However, from published accounts it is not clear whether any of them would be suitable for an interactive experience as described here.

3.3.2 The AutoPhotobook layout engine

As mentioned above, the AutoPhotobook layout engine is based on BRIC [7] as used in [8], with the novel development that the user is free to add a text block of virtually any size. In this subsection, we provide a brief overview of BRIC, and we illustrate how the AutoPhotobook layout engine supports inclusion of a text block.

BRIC is an algorithm for arranging virtually any number of photos on a rectangular canvas having any dimensions[7]. In contrast to other photo layout methods, BRIC satisfies the following two criteria: First, photo aspect ratios are respected, so that all of each photo remains visible in the composite; and second, the space between adjacent photos is precisely controlled. Formally, the layout is characterized as a binary tree that corresponds to a recursive partition of the page. Each of the terminal nodes in the tree is associated with a photo, and each interior node corresponds to a rectangular bounding box. In any actual layout, the area of each photo is determined by its position relative to the other photos (*i.e.* the tree structure), taken together with the canvas dimensions and with the constraints on space between adjacent photos.

Many of the editing mechanisms introduced in [8] and used in AutoPhotobook follow the procedure of first modifying the tree (in a manner prescribed by the editing command), then “reflowing the page” or computing an updated layout based on the modified tree. For example, to swap two photos, we swap the respective terminal nodes in the tree and then reflow the page. AutoPhotobook uses this procedure to support the operations of swapping objects; replacing an object (with a photo or text block as allowed subject to the maximum of one text block); cropping a photo; and editing a text block. (To add an object to the page, or to delete an object, we simply generate a new layout.)

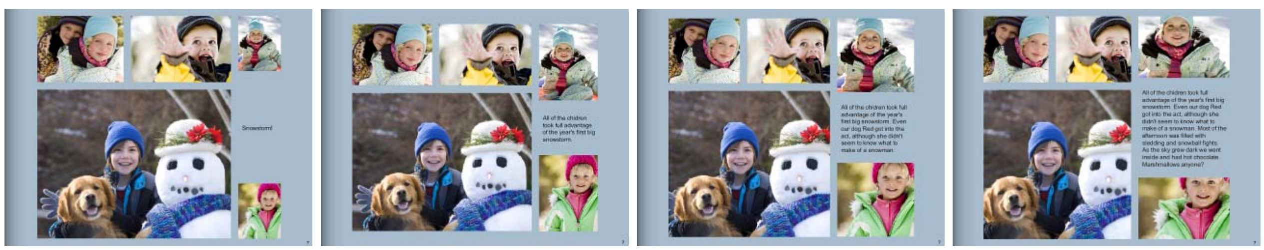
To implement text support in AutoPhotobook, we characterize a text block as having multiple presentations, where each presentation is fixed in both aspect ratio *and* area. Each presentation is defined by a specific set of dimensions, (*i.e.* a (height, width) pair); and all presentations are regarded as equally acceptable. During layout creation, or when reflowing the page, we generate a different set of candidate layouts for each presentation; and the candidate having the highest score is used.

3.3.3 Results illustrating text support

Figure 11 shows alternate layouts of four photos and a text block. As described in [8], to generate alternate layouts we run separate instances of the layout engine, with each instance based on a different set of suggested relative areas for photos. In a majority of instances, as illustrated in Figure 11, each of the alternate layouts uses a different presentation of the text block. This is not deliberate and we attribute it to use of different suggested relative area assignments.

Figure 11. Alternate layouts of a content set. Note the use of different text block presentations.

Although the AutoPhotobook layout engine strives to respect photo aspect ratios, text presentations are regarded as fixed, and in some cases photos are cropped as a last resort. To illustrate, Figure 12 shows four layouts that differ only in text block length. Layout (b) was generated first. In this result photos are either not cropped, or cropped minimally. To create Layouts (c) and (d), we only added text to their respective precursors; and to create Layout (a) we deleted text from Layout (b). Note that in Layout (d) photos have been cropped considerably to make room for text.



Layout (a): Result of deleting text from the text block in Layout (b).	Layout (b): Initial arrangement generated by layout engine.	Layout (c): result of adding text to the text block in Layout (b).	Layout (d): result of adding text to the text block in Layout (c).

Figure 12. Layouts resulting from editing the text block.

3.4 User interface design

In developing the user interface (UI), we strove for a minimalist design that still retains full advantage of the features themselves as our previous work [15]. Our goal is to avoid restraining the user to the prescribed steps, but to seek synergies between focusing solely on user control or computer automation.

AutoPhotobook anticipates users' needs with contextual UI controls appearing in two views: index view (overview of all pages of a photobook) and book view (close-up view of a single two-page spread). At any of the creation, editing, reviewing stages, the user can switch between the two views that offer both context and detail on demand. All user interactions are in-place, without opening up new editing or preview windows, so that users only need to get familiar with two views.

When a user first opens an album to create a book, the automatically generated book is presented in index view (Figure 13), which offers an overview of all pages in the book with their layout. In this view, users can easily swap or move photos between pages and add or delete pages, as well as perform editing actions on individual pages to shuffle, add/delete photos, swap photo locations, and replace a photo on the page.

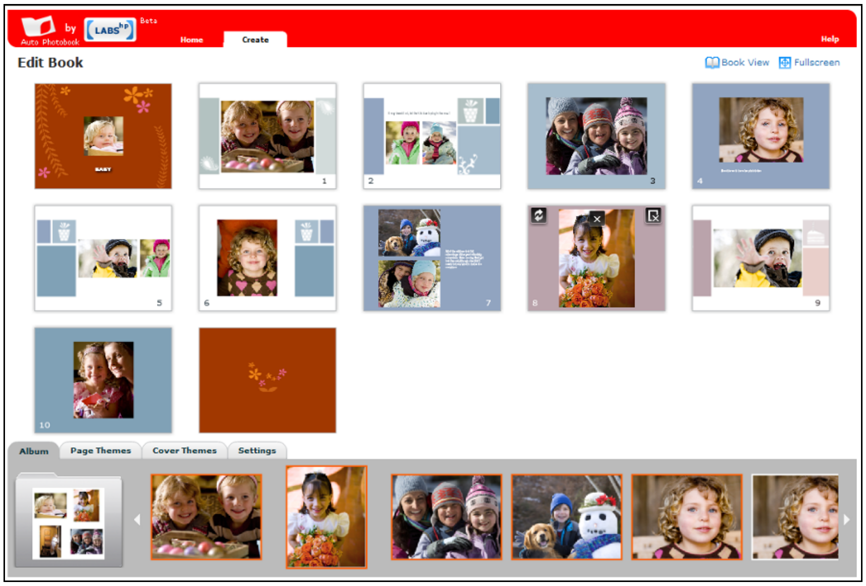


Figure 13. Index view in AutoPhotobook. Between-page editing, as well as most page-level editing actions, can be done in this view.

For finer-scale editing on each page, users can switch to the book view (and back to index view) via a single button click. This brings up a single two-page spread with flip-enabled pages, so users can easily flip to other pages in the book for editing or preview (Figure 14). In this view, users can add a text block to the page, edit the text content or perform auto-crop and auto-enhance on photos. Shuffle, add/delete, swap, and replace photos are also enabled in this view.

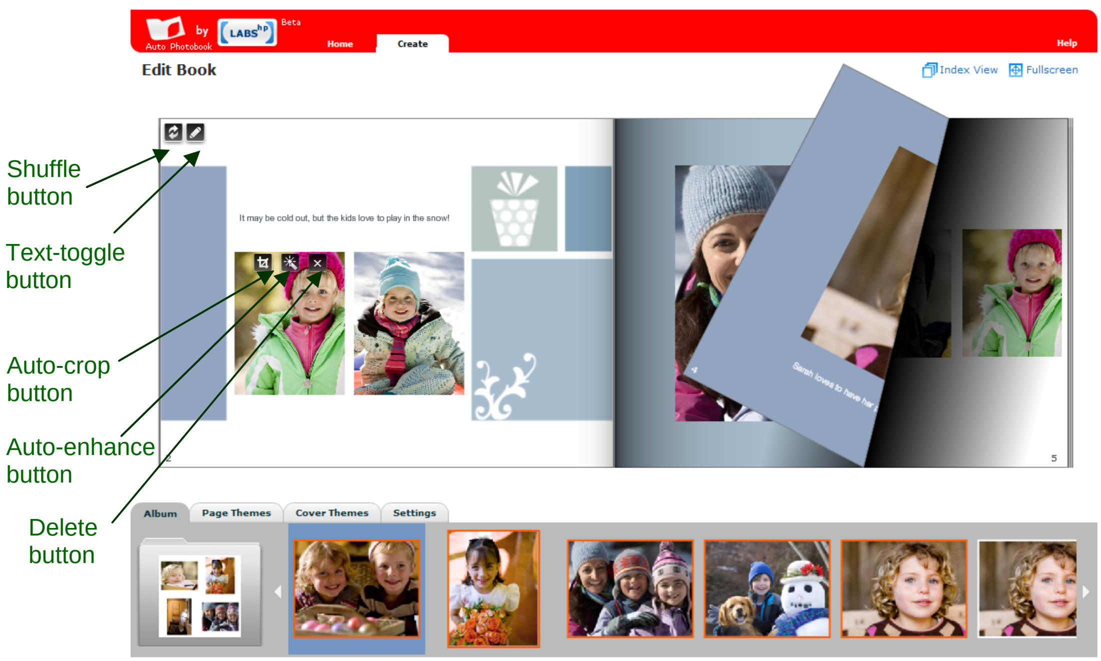


Figure 14. Book view in AutoPhotobook. Photo-level editing and text editing can be done in this view, in addition to other page-level editing actions.

Each page has at most two button controls that become visible only when the user moves the mouse onto that page. Clicking the shuffle button scrambles the layout on the page so users can easily preview different layout choices. On the cover page, the shuffle button serves a slightly different role. It toggles through all candidate cover photos selected by the automatic algorithm if there is only one photo on that page. There is also a text button that toggles the text block on a page. Other editing actions such as swapping, replacing, and adding photos are done by drag-and-drop operations.

There are up to 3 button controls on each photo on a page. One toggles auto-crop on and off, another toggles auto-enhance on and off, and the 3rd one is the delete button. The auto-crop and auto-enhance functions give users some flexibility in photo-level editing while keeping the editing UI minimal and easy to learn. The one-click crop and enhance toggle buttons make it easy for the users to preview and decide whether they want to keep the auto-crop and auto-enhance results.

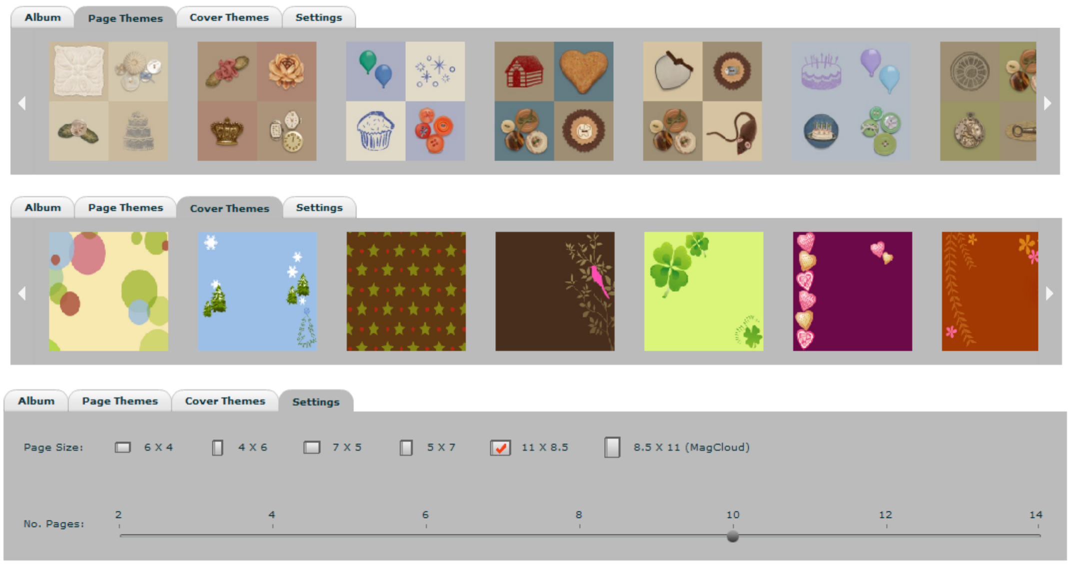


Figure 15. Page Theme, Cover Theme, and Settings view of AutoPhotobook control tabs.

Resources and controls used at book-level are organized into 4 tabs on the bottom of the screen (Figure 15). The "Album" tab lists all available photo collections uploaded by the user. The user clicks on an album icon to open it and then scroll through the photo strip to browse and select photos if they wish to modify the selection done by the automatic pagination algorithm (see Figure 14). The "Page Themes" tab lists all page designs available to choose. Clicking on any of the design will update the entire photobook to that design, by assigning specific backgrounds to each page automatically, and adjusting all page layouts accordingly. The "Cover Themes" tab lists all photobook cover designs. Our pilot test indicated that users prefer to have cover design selection independent from page design selection, hence the decision to separate these two design lists. Finally, the "Settings" tab is used to adjust number of pages in the photobook and page dimensions. When the users mark the selection or drag the slider to make changes, the selection of the photos, pagination, page layout and the artwork sizing are all adjusted automatically in real time to accommodate the changes. Again, all changes appear in-place with context, without opening up new windows.

Much of the editing flexibility we provide through this UI was enabled by the powerful automatic processing algorithms such as scalable pagination algorithm, structured art framework, and automatic layout generation algorithm. The ability to make significant changes to the book pagination and layout with a few simple clicks, without leaving the book editing environment, gives a natural and effortless feel to the photobook tool. The user interface described herein is well suited for devices that have reasonably large displays that support drag and drop as well as clickable actions either through touch screens or computer mouse actions. Some creative product authoring might also be done using devices with smaller displays or more limited interactions. For these, the approach we take of automatically creating an initial photobook and then allowing the user to edit as desired can easily be adapted to a target device. The automatic creation step will still work for any networked device, since all of the computation is done in the cloud. Interactive editing options can then be customized to fit the capabilities of the device. Some devices might not support drag and drop functionality and instead only allow the user to shuffle among the automatically created page alternatives. Image enhancement and auto-crop might be automatically used for all images on some devices, and page designs might be more basic if the device does not support high quality color or high resolution. Overall, we believe that our approach can be readily adapted across a wide range of networked devices, making it ideal for a CE 2.0 world.

4. Powering CE 2.0 with AutoPhotobook

AutoPhotobook is a full-function web destination that is accessed using a standard web browser. We can increase its effectiveness and reach by incorporating AutoPhotobook within a larger ecosystem that makes use of functionalities from other sources, as well providing photobook creation services to other clients. This integration allows users to build solutions where the AutoPhotobook capability is one component among many others within a software stack. For instance, we may create photobooks using photos stored on Flickr, a popular photo storage web site. We may also provide users not using a standard web browser the capability to create photobooks using AutoPhotobook. This latter use case is particularly pertinent to consumer electronics as few CE devices incorporate standard web browsers. In this section, we will describe how this integration is done and show how the AutoPhotobook features are delivered to devices as part of a broader customer solution.

AutoPhotobook can consume the services and import content from other web sites. It also can allow clients to control various steps in the AutoPhotobook creation process and export final photobooks to other destinations programmatically, without the use of a human driven interactive interface. The most popular style of web services communication is REST (Representational State Transfer) [38], a simple protocol commonly implemented on the ubiquitous HTTP protocol. Many Web 2.0 sites, such as Flickr or Facebook, provide a REST API to allow clients access to the stored content. AutoPhotobook is equipped with a number of REST client modules to access the services from these sites.

There are two primary categories of services that AutoPhotobook integrates with: Web 2.0 social community services with large amounts of media content and fulfillment services. Social community sites, including sites such as Flickr, Blogger, or Facebook, are popular content sharing sites capable of archiving large amounts of data. As AutoPhotobook is primarily a photobook creation site, rather than replicating functionality found in existing services, we simply reuse their functionalities. Thus, it is possible for an AutoPhotobook user to import her photo collections from one of these popular photo sharing sites into AutoPhotobook and create a photobook from these pictures.

Another class of web services used by AutoPhotobook is related to print fulfillment. After creating a photobook online, users often want a physical copy of the book as a keepsake. Although AutoPhotobook provides a PDF for users to download and print at home, print fulfillment providers can produce photobooks on higher quality material and binding which may be more suitable for a keepsake book. Currently, AutoPhotobook can submit photobooks to print fulfillment sites such as HP’s MagCloud print-on-demand service, although this capability can be extended easily to any site that provides a REST interface for clients.

By using web services provided by the media sharing and print fulfillment sites, the AutoPhotobook system expands from a site specializing in photobook creation to a customer solution that truly spans the workflow from photo collections to finished books. Depending on the end user application, it is possible to integrate the current solution with additional services to address niche offerings. For instance, by integrating AutoPhotobook with a travel review site and importing location descriptions and photos, it would be possible to publish travel photobooks as a travel guide book.

One aspect of integrating into a network of services and devices is to create more complex solutions; the other aspect of this integration is to make this solution available to more clients. Although the most visible part of AutoPhotobook is the standard web page interface for web browsers, the services and content from AutoPhotobook are also available via web services using a REST programming interface. A number of the algorithms that operate on a single photo, such as auto-crop or photo enhancements, are structured as web services. Many of the operations which map to user actions, such as uploading photos, shuffling photos within a page, or moving a photo from one page to another, are also exposed as web services. In fact, the standard user interface in AutoPhotobook, which is implemented with Adobe Flex, is simply a client that knows how to invoke the AutoPhotobook REST interfaces. The server itself does not know, nor care, the origin of the client requests. Thus, it is possible to develop multiple client front-ends for AutoPhotobook. Figure 16 shows how the components come together to form a number of different solution.

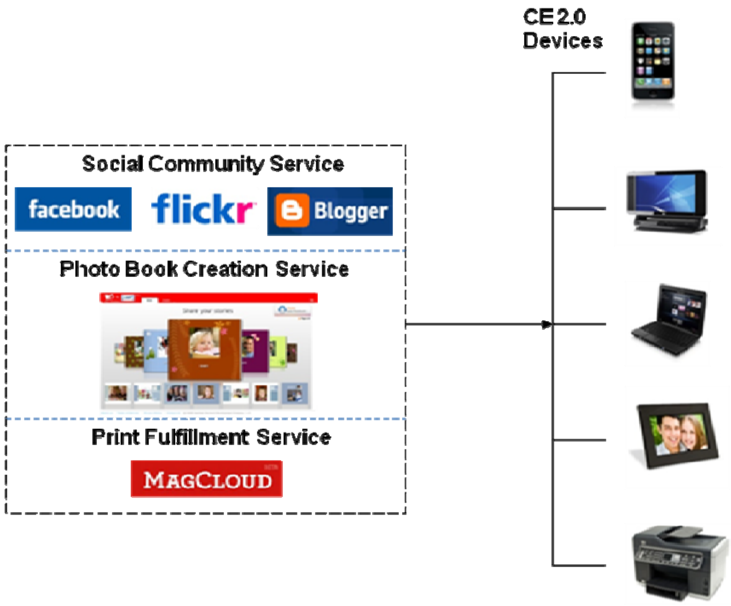


Figure 16. The extended AutoPhotobook ecosystem.

It is the exposure of the AutoPhotobook functionalities as programmatically invokable web services which makes it possible to use these same services to augment the functionalities found in consumer electronic devices. The consumer electronic device is a client no different than the standard Flash client on the AutoPhotobook web site. One obvious consumer electronic platform for integration is the camera-equipped smartphone. A service such as AutoPhotobook, when integrated into the smartphone software environment via an application, can transform the camera phone from a picture taking device into a photobook creation device, with the end product being not just a digital photograph, but potentially a physical photobook as well.

Figure 17 shows an example of a photobook application running on an iPhone. Note that in this case, the user interface on the phone looks nothing like the actual web site but has been designed and customized for the consumer electronics device itself. When a consumer electronics device is enhanced with web services, the additional service functionality is integrated into the device, but user interfaces are typically custom created to best leverage the specific capabilities and characteristics of the device.

Figure 17. iPhone client connected to the AutoPhotobook creation service.

This pattern of augmenting devices with web services can be extended to other consumer electonics devices besides the smart phone. Another example is the digital photo frame. In addition to displaying the photo collections in different slideshow format, by connecting to AutoPhotobook service, the digital frame can be enhanced to display photobooks as well, with no additional hardware. As many digital photo frames now download the source photos from photo sharing sites, AutoPhotobook can be used to import the photos from those sites and export finished photobooks to the digital frame as well. Another viable candidate for enhancement is the home printer. When a user inserts the memory card into the printer for photo printing, the printer can use the AutoPhotobook service to create photobooks to provide additional product alternatives for the user. In terms of CE 2.0 products' ability to grow and adapt to the users' needs, enhancing their capabilities through connections with web services is an effective method that is growing in popularity.

5. Conclusion

Consumer electronics products are increasing differentiated by software and services rather than by hardware specifications. Many new research challenges come up that require consideration of not only single products but a combination of products and services in an eco-system. In the AutoPhotobook sytem as an example, intelligent image analysis and composition are core technologies that are used in multiple implementation paradigms targeted at creating a seamless multimedia experience.

Acknowledgments

The authors would like to thank Phil Cheatle, David Slatter, Chris Willis, Andrew Carter and Roland Penny for technology contributions and many helpful discussions on the Autophotobook application. We would also like to thank HP's Global Delivery China Center team members, lead by Hua Zhang, for their dedicated efforts in building the Autophotobook.com site described in this paper: Tonghai Duan, Fei Ma, Zhidong Mao, Zeshuo Yin, and Yue Yuan.

References

[1] Richard MacManus, “Consumer Electronics 2.0: MIT's Henry Holtzman on The Internet of Things”, ReadWriteWeb, 2009.

[2] Junko Yoshida, “What Cisco's 'CE 2.0' challenge means to consumer electronics market”, EE Times, 2009.

[3] Tong Zhang, et al, “Face Based Image Navigation and Search”, Proceedings of the 17th ACM International Conference on Multimedia, 2009.

[4] Andrew C. Gallagher, et al, “Image Annotation Using Personal Calendars as Context”, Proceedings of the 16th ACM International Conference on Multimedia, 2008.

[5] Junsong Yuan, et al, “Mining GPS Traces and Visual Words for Event Classification”, Proceedings of the 16th ACM International Conference on Multimedia, 2008.

[6] Dong,Liu, et al, “Smart Batch Tagging of Photo Albums”, Proceedings of the 17th ACM International Conference on Multimedia, 2009.

[7] C. B. Atkins, “Blocked Recursive Image Composition”, Proceedings of the 16th ACM International Conference on Multimedia, 2008.

[8] Xiao, J., Zhang, X., Cheatle, P., Gao, Y., and Atkins, C. B., “Mixed-initiative photo collage authoring,” Proceeding of the 16th ACM International Conference on Multimedia, ACM, 509-518 (2008).

[9] Susanne Boll, et al, “Semantics, Content, and Structure of Many for the Creation of Personal Photo Albums”, Proceedings of the 16th ACM International Conference on Multimedia, 2008.

[10] Niels Henze, et al, “Snap and Share Your Photobooks”, Proceedings of the 16th ACM International Conference on Multimedia, 2008.

[11] PMA Marketing Research, “2008 PMA Camera/Camcorder and Digital Imaging Survey,” 2008.

[12] Chao, H., Gabbur, P., and Wiley, A., “Preserving the aesthetics during non-fixed aspect ratio scaling of the digital border,” Proceedings of the 2007 ACM Symposium on Document Engineering, ACM, 144-146 (2007).

[13] Obrador, P., Moroney, N., MacDowell, I., and O'Brien-Strain, E., “Image collection taxonomies for photo-book auto-population with intuitive interaction,” Proceeding of the eighth ACM symposium on Document engineering, ACM, 102-103 (2008).

[14] Sandhaus, P., Thieme, S., and Boll, S., “Processes of photo book production,” Multimedia Systems 14, 6, 351-357 (2008).

[15] Yu, M., Liang, Y., Lee, K., Chen, B., and Ouhyoung, M., “Smart album: photo filtering by effect detections,” ACM SIGGRAPH 2008 posters, ACM (2008).

[16] Renato Keshet, Carl Staelin, Boris Oicherman, Mani Fischer, Pavel Kisilev, Sagi Schein, Doron Shaked, Marie Vans, Hila Nachlieli, Ruth Bergman, Ron Maurer, Michal Aharon, Alexander Berkovich, Ran Waidman, Oren Bengigi, Gal Amit, Shlomo Harush, Alon Asher, Hui Chao, Darryl Greig, Matthew Gaubatz and Steven J. Simske, "Automatic Photo Enhancement Server (HIPIE 2)," International Symposium on Technologies for Digital Fulfillment, Las Vegas, Nevada, (2009).

[17] Phil Cheatle, “Automatic image cropping for republishing,” Proceedings of SPIE Volume 7540A, Imaging and Printing in a Web 2.0 World, San Jose, CA, (2010).

[18] D. Lowe. “Distinctive image features from scale-invariant keypoints,” IJCV, 60 (2): 91–110.

[19] Y. Rubner, C. Tomasi and L. Guibas. “The earth mover distance as a metric for image retrieval,” IJCV, 40(2), (2000).

[20] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features,” CVPR (2001).

[21] Graham, A., Garcia-Molina, H., Paepcke, A., and Winograd, T., “Time as essence for photo browsing through personal digital libraries,” *JCDL*, (2002).

[22] Tang, F. and Gao, Y., “Fast Near Duplicate Detection for Personal Image Collections,” Proceedings of the 17th ACM International Conference on Multimedia, (2009).

[23] S Avidan, A Shamir, “Seam carving for content-aware image resizing”, ACM Transactions on Graphics, 2007.

[24] M Rubinstein, A Shamir, S Avidan, “Improved seam carving for video retargeting”, ACM Transactions on Graphics-TOG, (2008).

[25] Y Wang, C Tai, O Sorkine, T Lee “Optimized Scale-and-Stretch for Image Resizing”, ACM Transactions on Graphics (TOG), (2008)

[26] P Dragicevic, S Chatty, D Thevenin, JL Vinot, “Artistic resizing: a technique for rich scale-sensitive vector graphics”, Proceedings of the 18th annual ACM symposium on User Interface Software and Technology, (2005).

[27] Birkhoff, G.D.,“Aesthetic measure” Harvard University Press, (1933).

[28] Harrington, S.J., Naveda, J.F. et. al.“Aesthetic Measures for Automated Document layout”, Proceedings of the ACM Symposium on Document Engineering, (2000)

[29] Chao, H., Gabbur, P., and Wiley, A., “Preserving the aesthetics during non-fixed aspect ratio scaling of the digital border,” Proceedings of the 2007 ACM symposium on Document engineering, ACM, 144-146 (2007).

[30] Chao, H., Zhang, X., and Tretter, D., “Structured Layout for Resizable Background Art”, Proceedings of the ACMMM associated workshop on Interactive Multimedia on Consumer Electronics (ICME), (2009).

[31] Jingdong Wang, Jian Sun, Long Quan, Xiaoou Tang and Heung-Yeung Shum. “Picture Collage,” In Proc. of Computer Vision and Pattern Recognition (CVPR), (2006).

[32] Yichen Wei, Yasuyuki Matsushita, and Yingzhen Yang, “Efficient Optimization of Photo Collage,” Technical Report, Microsoft Research, MSR-TR-2009-59, May, (2009).

[33] Hurst, N., Li, W., and Marriott, K. 2009. “Review of automatic document formatting,” In Proceedings of the 9th ACM Symposium on Document Engineering (DocEng ’09). 99-108. (2009).

[34] E. Goldenberg, “Automatic Layout of Variable-Content Print Data,” M.S. thesis, University of Sussex, Brighton, UK, (2002).

[35] L. Purvis. “A Genetic Algorithm Approach to Automated Custom Document Assembly,” Proceedings of the 2nd International Conference on Intelligent Systems Design and Applications (ISDA’02) Atlanta, GA, August (2002).

[36] L. Purvis, S. Harrington, B. O’Sullivan and E. Freuder. “Creating Personalized Documents: An Optimization Approach”, Proceedings of the ACM Symposium on Document Engineering, Grenoble France, (2003).

[37] de Oliveira, J. B. 2008. “Two algorithms for automatic document page layout,” In Proceedings of the Eighth ACM Symposium on Document Engineering (DocEng ’08). Sao Paulo, Brazil, September 16-19, pp. 141-149. (2008).

[38] R, Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. thesis, University of California, Irvine, (2000).