Story Albums: Creating Fictional Stories from Personal Photograph Sets

Oz Radiano[1], Yonatan Graber[1], Moshe Mahler[2], Leonid Sigal[2], Ariel Shamir[1]

[1]**The Interdisciplinary Center, Herzliya,** [2]**Disney Research, Pittsburgh**
(ozradiano@gmail.com, yonatan.graber@gmail.com, moshe.mahler@disneyresearch.com, leonid.sigal@disneyresearch.com, arik@idc.ac.il)

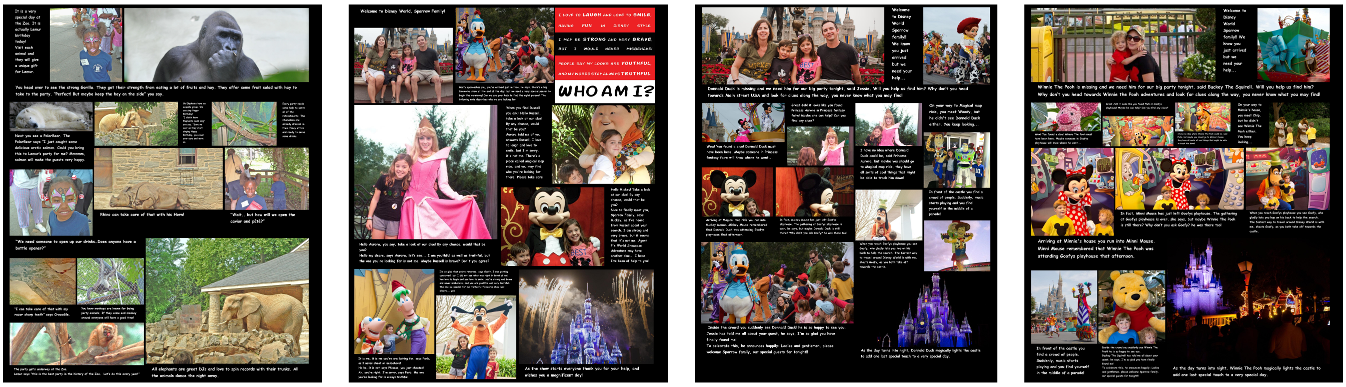

Figure 1: Four examples of fictional stories automatically created by our method. The same story can have a very different look when it is based on different personal photograph sets (the pair on the right). On the other hand, the same personal collection of photographs can be used to create different stories (middle pair).

Abstract

We present a method for the automatic creation of fictional story-books based on personal photographs. Unlike previous attempts that summarize such collections by picking salient or diverse photos, or creating personal literal narratives, we focus on the creation of fictional stories. This provides new value to users, as well as an engaging way for people (especially children) to experience their own photographs. We use a graph model to represent an artist-generated story, where each node is a "frame", akin to frames in comics or storyboards. A node is described by story elements, comprising actors, location, supporting objects and time. The edges in the graph encode connections between these elements and provide the discourse of the story. Based on this construction, we develop a constraint satisfaction algorithm for one-to-one assignment of nodes to photographs. Once each node is assigned to a photographs, a visual depiction of the story can be generated in different styles using various templates. We show results of several fictional visual stories created from different personal photo sets and in different styles.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—H.2.8 [Database Applications]: Image databases—

Keywords: Image Processing, Personal Visual Stories, Image Databases, Photo Summarization

1. Introduction

The number of photos people take is huge and constantly growing. Many works try to filter and summarize large photo collections, for example, by finding the best diverse subset of photos [LWSB08, SMJ11]. Some works go a step further and attempt to extract some "storyline" from the input photo sets [OdOO10] to create photo albums or digital presentations. This is a hard problem in general (see [HFM*16] for the first extensive data-set), and even if solved, it can only portray a selection of events that appeared in the set. What if a new fantasy or a fictional story could be created from your own photos? This could be appealing to children and can also

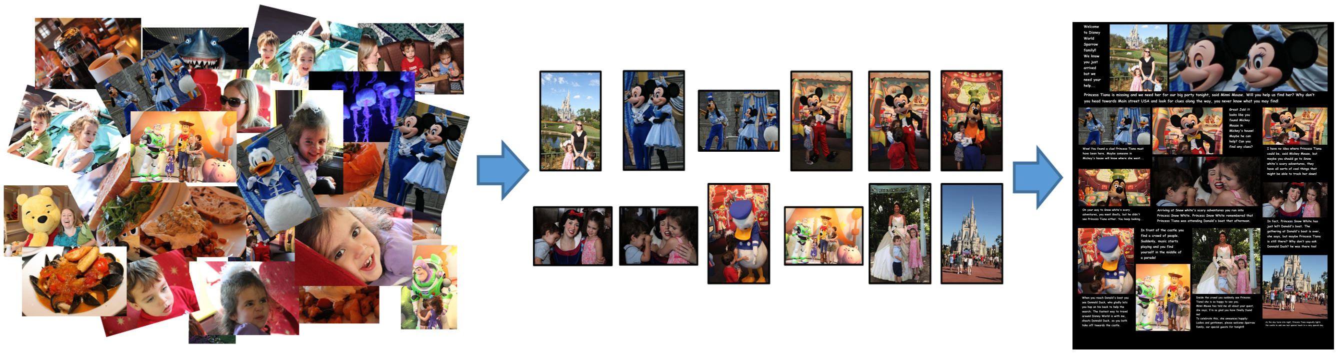

Figure 2: The challenge of creating a visual story from a personal photo collection (left) is twofold: first, finding a subset of photos that fit the events in the story well (middle), and second, creating a high quality visual depiction of the story (right).

stimulate looking at your own photos in a new and engaging way. In this work, instead of searching for a story in a set of photos we reverse the question and ask: can we ground a predefined fictional story in a given set of photos? We present a method that can create a visual depiction of a fictional story based on personal photos.

Stories and fairy-tales have always caught the imaginations of people in all ages and cultures. Applications and web-services allow the creation of customized visual stories, raising the level of engagement by personalizing parts of the story (this is linked to similarity attraction [Byr71] or egocentric bias [SCM03] in psychology). These services allow embedding some personal images and text (e.g. faces, names) into designated places in a printed book or a digital presentation. However, the majority of the visual content in these stories is not personal. In contrast, in our work almost all visual content is personal, resulting in a personalized visual depiction that is different for every photo set (see Figure 1).

Our work assumes a number of descriptive attributes are given with each photo (see Section 4). These include *where* and *when* the photo was taken, and *who* and *what* is in it. Some attributes are commonly found in the meta-data of photos (GPS, time-stamps), while others, we believe, could soon be automatically extracted with the support of state-of-the-art computer vision algorithms (e.g. face identification and object detection). As this is not the focus of this work, we use a combination of manual tagging and automatic methods in our examples. Hence, our assumption is that the user only needs to supply the photo set and the attributes will be extracted automatically.

Similarly, we assume that a set of pre-defined stories exist (created by artists) and do not have to be created by the user. To represent a story we use a graphical model, where each node is a "frame", akin to frames in comics or a storyboard. A node is described by some attributes regarding the event portrayed in the frame, for example, who participates or when it happened, and a textual description of the event. The edges in the graph encode dependencies between the nodes that pertain to the various story elements (see Section 3).

From the user perspective, the input to our method is a set of personal photos. The attributes could be extracted, and a definition of different possible story-graphs are given. The output is a visual depiction of a story (or several stories) based on the photos. It is obvious that not every photo set can match any story. We chose to focus on photos from vacation trips that tend to be more appealing and interesting. Accordingly, the stories we demonstrate relate to the place the photos are taken in. We demonstrate three scenarios: a day in an amusement park, a trip to the zoo, and a trip to a known landmark (Machu Picchu), and use simple fictional stories (adventure and birthday) that children can relate to. These examples demonstrate the diversity and applicability of our approach to other scenarios as well, once appropriate stories are defined.

The challenge we face is twofold. First, we need to choose a subset of the photos that best fits a pre-defined story, and score this matching. Second, we need to convert the mapped subset of photos to a visual layout depicting the story (Figure 2). Not all photos in the set must be used in the story, some may not be used because they do not fit the storyline and others because of their low quality. Our algorithm chooses the best subset of photos automatically for a given story with no need to pre-filter the photo set. It can also be used to map the same set of photos to several stories and choosing the one with the best score, or allowing the user to choose.

The first challenge amounts to finding a one-to-one assignment of nodes to photos from the given input set that best satisfies the attributes and constraints in the graph. We solve this using a constraint satisfaction algorithm combined with a random search. The second challenge amounts to creating the visual layout of the story from the selected photos. We define a set of one-page layout templates for a different number and orientation of photos. We segment the selected photo sequence to consecutive sub-sequences and fit them to the page-templates, while preserving their order. We find the best segmentation using iterative local search based on the quality of assigning the sub-sequences of photos to the given templates. The quality measure is based on cropping each photo to its frame, and fitting the text to its designated position. We can support different styles of depictions for the story by choosing different sets of templates for each style (see Figure 6).

Our contributions are as follows: a method for the creation of visual fictional stories that are personalized by a set of input images, the definition of a graph model to represent a story and its constraints, an algorithm for assigning nodes to photos, and an algorithm for the creation of a visual depiction of the story based on the definition of layout templates. We show several examples of our method on four different stories, in different styles and for numerous photo sets. We also evaluate our mapping algorithm and compare our output to manual selection performed by humans.

2. Previous Work

Personal photo collections: Several works study how people manage their personal photos [RW03] and how to create digital archives of personal memories [SATV03, Sel11]. However, these works try to manage a whole collection while our work tries to choose specific images that fit a given story. Similar to our motivation, PoseShop [CTM*13] presents a system to create personalized comic-strip, however their work uses general internet photos and personalizes them by head replacement, and the stories are sketched interactively by the user.

Photo collection summarization: Summarization of a large photo collection is addressed in [YSPF13, YSF11, PCF03], but these works usually use clustering or filtering of images and do not try to create a storyline, let alone a fictional story. Recently "photo storytelling" has been used in the context of (large) photo collection summarization using both aesthetic measurements and high level features [OdOO10, Obr11, WLO12] as well as utilizing social networks [KX14], where multiple large collections of photos are used to reconstruct possible story lines of the events. However, these works mainly try to discover events that happened in the photos while we try to fit them to a pre-defined fictional narrative.

Choosing summary photos from a collection often relies on filtering based on quality with some measure of diversity. The study of photo quality uses measures that include technical quality, colorfulness, composition, face positioning and more [KDP09, DJLW06, LLC10]. Since our photo sets are mostly taken by amateurs, we use a simple model for measuring the technical quality of the photo that includes sharpness and lighting, and use face detection and saliency to assist image resizing, if needed. Several works automatically create visually appealing collages from an input photo sets by analyzing the photos and solving an optimization problem for placement [RKKB05, RBHB06, LWS*09]. However, these works only take appearance attributes into account and do not follow a storyline.

Fitting images: Several works try to add visual elements and photos to stories. For example, [DMC10b, DMC10a] use general web search to automate the illustration of news articles, but these do not contain a storyline and are not visual in nature. VizStory [HLS13] investigates the narrative structure of a story to segment it and select representative keywords for each segment. These are used in a web image search to find suitable pictures. In our case, the story graph along with its constraints is already pre-defined. In the future, one can think of ways to automatically create new story-graphs from examples such as in [LTWR14].

Constraint-satisfaction: Constraint-satisfaction (CS) deals with techniques for solving combinatorial optimization problems. A constraint-satisfaction problem consists of a set of variables, on their respective domains, and a set of constraints that limit the scope of subsets of variables. CS is a well studied problem in AI and a number of techniques have been proposed for solving CS problems [BSR04]. Among them, generate-and-test is the simplest, but requires generation of a combinatorial number of hypothesis, which tends to be impractical for most problems. Backtracking-type search algorithms explore depth-first search strategies to look for solutions [BR75]. Variable ordering optimization methods, of which our proposed approach is an instance, order variables in ways that optimally reduce the set of possible values in a domain [BR75]. Our optimization procedure is a form of variable ordering with an additional stochastic search component.

An important class of solutions to CS problems utilize distributed inference architectures [YH00]. Massage passing algorithms are among the most popular, due to their efficiency and provable optimality in tree-structured graphs. Belief propagation (BP) [YFW05] is an instance of massage passing algorithms that in practice has good performance and can be applied to both CS [MRTS07] and probabilistic inference formulations. The difficulty with BP is that complexity is exponential in the number of variables involved in the constraints/potentials, typically requiring constraints among variables to be local and/or pair-wise. Recently there has been a surgance of BP algorithms that allow use of specific forms of higher-order terms [TGZ10]; however, for tractable inference only certain forms of non-local potentials are allowed, e.g., cardinality-based potentials [VCR08]. We need to model exclusion among the photos (no photo can be placed in more than one frame), which requires a potential over all variables, disallowing effective use of traditional distributed CS solutions.

Layout: The problem of deciding which elements will appear in a page and how to arrange them visually is often called *layout*. Chao et al. [CTZA10] fit an input set of images to a canvas (with some exclusion zones), while preserving their aspect ratio (no cropping), and trying to preserve suggested relative sizes. However, they do not have a notion of sequence and do not preserve the order of images. Sandhaus et al. [SRB11] formulate aesthetic principals into a fitness function to place images and text on a page and use a genetic algorithm to find a good layout.

Our problem is more akin to comics since the main components are photos and the text that accompanies them. A model for optimizing the place of images and word balloons to follow a specific eye movement pattern in comics is presented in [CLC14]. There is a line of works that address converting videos or interactive games to comics. These works mostly follow the same pipeline of choosing key-frames from the video, arranging them on the page, positioning word balloons (if applicable), and sometimes using non-photorealistic stylization [WHY*12, CGC07, SRL06, UFGB99].

To preserve the original photos we do not apply any stylization (although this could be added in the future), and we do not use word balloons. Instead, our main challenge is to position the images on the page and place the text alongside. To allow various styles of layouts, we chose to use a pre-defined set of templates, similar to previous works such as [WHY*12, CCK*06, JLS*03]. The closest work to our layout algorithm is [CYW15], that maps a given sequence of images to comic pages. The sequence is partitioned to subsets using a genetic algorithm based on the color and motion coherency of the images. Then, the best page layout template is

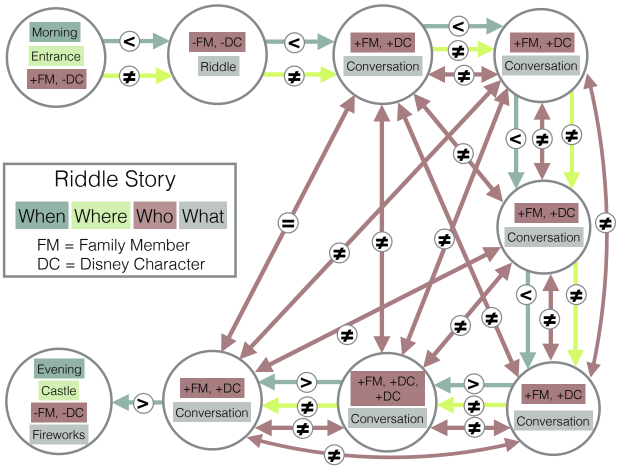

Figure 3: An example of a story graph (not all edges are shown): the attributes inside nodes are displayed as '+' for *contains*, '-' for *not-contain*, and the edge constraint include $=, \neq$ (for places, objects or participants) or $<, >$ for emporal attributes.

chosen for each subset by finding the best match between the sorted importance of the images and the sorted importance of frames similar to [CCK*06]. In contrast, our work solves the two problems simultaneously: partitioning the sequence of photos to subsets and selecting the best templates for the subset.

3. Story Graph

We represent a story as a graph G = {V,E}, where each node represents an event frame (Figure 3). We support only linear discourses (the presentation sequence of events) in one path, and therefore the nodes in the story graph have an ordering from the first to the last. We use an index i to denote a node $i \in \mathsf{V}$. An event in a node is defined using the following possible attributes:

1. **who**: describes who participates in the event.
2. **where**: describes where the event has taken place.
3. **when**: describes when the event happened.
4. **what**: describes any special object that is part of the event.

Each attribute type can take a value from a discrete set. For example, the **who** set contains all family members and Disney characters in a park or all animals in a zoo. The **where** set contain geographical region such as attractions in the park or areas in a city. The **when** set includes {morning, noon, evening, night}. The **what** can contain elements that are included in a specific story such as a balloon or water or hay. Various values can also be grouped together to form subsets. For example, several animals can be grouped together into *horned-animals*, several characters can be grouped into *Disney-princesses*, and all family members can be grouped into a *family*.

Each node includes a set of boolean expressions on these groups of attribute values that are used to determine the matching of photos to the node. We use three levels of simple inclusion expressions on groups, where an individual element can be seen as a group containing a single element: *contains* (e.g. the event must include Mickey Mouse, or must contain one of the family members), *not-contain* (e.g. the event cannot happen in the castle), and *possibly-contain* (e.g. the event can happen in the morning, but not necessarily). The first two define hard constraints and the last one a soft constraint for fitting photos to the node. In addition, each node includes a text description of the event that will appear next to the frame in the visual depiction of the story.

Every edge $e \in \mathsf{E}$ in the graph defines a pairwise dependency between two nodes. Because there could be more than one edge between two nodes, our graph is formally a *multi-graph*. We use *equals* and *not-equal* relations on discrete elements (e.g. the character in node a must not equal to a character in node b). We also use *greater-than*, *less-than* on the temporal attributes (e.g. the time of event in node a is later than the time in node b, where morning $<$ noon $<$ evening $<$ night).

Automatic conversion of a textual story to a graph involves semantic understanding and is still an extremely challenging problem. Further, since our stories are template-based, it requires reasoning over story elements and attributes to figure out which elements to include and how they can be generalized (e.g. is it important to have Rapunzel as an actor, or would any Disney princes provide a suitable story discourse?). The four story graphs used in this paper were created manually in about two hours each, using a simple graphical interface that allows to add/remove nodes and constraints. Once a story graph is created, it is added to a library and can be used to attempt to match any new set of photos. The library of story graphs could be extended easily to contain different types of stories to match various photo scenario. Figure 3 shows an example of one such story graph for the Riddle Story.

4. Photographs Attributes and Quality

To map photos to nodes in the story graph, we extract the same four types of attributes from the photos. We use GPS coordinates to find the position where a photo was taken, and assign the name of the closest attraction or landmark in the park as the **where** attribute. We convert the time-stamp of the photo to the respective part of the day (e.g. morning, noon, evening or night) for the **when** attribute. For the **who** attribute we trained a classifier for Disney characters and animals based on [RHGS15]. We used manual corrections of the automatic results as well as to tag family members and find objects in images needed for **what** attribute. Note that as vision algorithm progress, face detectors could possibly be used to detect faces, estimate gender and age [KRE09], and label family members [DCSH15]. Similarly, objects could be detected with recent computer vision techniques for object detection [GDDM14, FGMR10] or image categorization [KSH12]. Obviously, as more information is extracted from the photo, better support for more complex stories would be possible.

To support better photo selection for the stories, we define several measures for the quality of an photo I. First, we count the number of characters in the photo (i.e. Disney character, family member or animal) and use it to measure the "crowdedness" of the photo. Crowdedness is defined as the ratio of the number of characters relevant to the story, to the total number of characters in the photo. The relevant characters are the ones that participate in a given event portrayed in a node. Second, for each relevant character, we mark whether it is front-facing, back-facing or in profile, and measure the relative size of each character in the photo as the ratio between its bounding box and the whole photo. We use highest score (1) for front facing character, lower for profile (0.25) and lowest for back-facing (0.05). We define the character score as the multiplication of the relative size and facing direction. Third, since we assume non-professionals captured the photos we extract two global image characteristics. We use measures for blurriness of the photo [CDLN07] and for poor-lighting conditions of the photo – both for over-exposure and under-exposure [YS12].

A weighted average of the above factors (normalized between 0 and 1) provides **quality**(I), the quality measure of the photo I, that lies between 0 (poor quality) and 1 (high quality). The weights we use in all our examples are 0.2 for crowdedness, blurriness and lighting, and 0.4 for the character score. In addition, the saliency of every pixel in the photo is calculated based on the approach of [GZMT12], for later use in the layout algorithm to measure the cropping quality.

5. Photographs Assignments

Given a set of N photos $\mathsf{I} = \{I_1,\dots,I_N\}$, and a story graph representing an ordered sequence of K frames $\mathsf{B} = \langle F_1,\dots,F_K\rangle$ ($K << N$), we are looking for a one-to-one assignment of frames (nodes) to photos $\mathsf{B} \rightarrow \mathsf{I}$. We represent an assignment as a vector $\boldsymbol{x} = (x_1,\dots,x_K)$ of length K where each entry $x_i \in 1\dots N$ is an integer number representing the index of the photo assigned to node i (frame F_i).

We define the node-to-photo assignment problem as a form of constraint satisfaction problem (CSP) using the story graph. As noted in Section 3, inclusion (*contain*) and exclusion (*not contain*) are hard constraints that must be satisfied. All other constraints, including all edge constraints are considered as soft-constraints. However, large sets of input photos can create many possible solutions that all satisfy the hard constraints in the nodes. To distinguish between these solutions we define a real-valued "fitness" function of a photo set that satisfies the hard-constraints. We rely on soft constraint satisfaction and this fitness measure to find the best solution out of all solutions that satisfy the hard constraints.

Converting Story-Graph Attributes to Constraints: We separate the attributes of the nodes (defining who, what, where, and when) into two types: those that induce hard, and those that induce soft constraints. Some attributes that are essential to the story (e.g., appearance of a family member in the event described by a node), will be defined as *hard* constraints in our CSP problem. Other attributes that are optional (e.g., appearance of more than one family member in an event), will be defined as *soft* constraints.

Denote the set of all soft constraints for node i as **soft**(i) and the set of all hard constraints as **hard**(i); we separate the set of edge constraints (that are all soft) between node i and node j into subsets corresponding to specific attributes **who**(i,j), **what**(i,j), **where**(i,j), **when**(i,j). Let $E(I_x) \in \{\mathbf{hard}(i),\mathbf{soft}(i)\}$ be a binary evaluation function for a constraint generated by a node and $G(I_x,I_y) \in \{\mathbf{who}(i,j),\mathbf{what}(i,j),\mathbf{where}(i,j),\mathbf{when}(i,j)\}$ be a binary constraint function for an edge, where I_x and I_y are photos. Unlike $E(\cdot)$ that takes the values 0 or 1, $G(\cdot,\cdot)$ takes the values of 1 if the constraint is satisfied and -1 if not. Negative values are used to more strongly enforce soft edge constraints, ensuring that solutions that do not satisfy edge relationships are more heavily penalized.

We define the CSP objective as:

$$\xi(\boldsymbol{x}) = \prod_{i=1}^{K} \xi_i(\boldsymbol{x}) = \prod_{i=1}^{K} \prod_{E\in\mathbf{hard}(i)} E(I_{x_i}). \tag{1}$$

For each node i, $\xi_i(x_i)$ is either 0, if one or more hard constraints are not satisfied, or 1, when all hard constraints are satisfied. All $\boldsymbol{x}$ for which $\xi(\boldsymbol{x}) = 1$ are solutions to the CSP. To rank these solutions we define a measure of quality of photo assignment to the node, which is based on satisfaction of soft constraints and photo quality. In other words, we want photo assigned to each frame to satisfy as many soft constraints as possible and also to be of high quality. This is formalized by the following equation,

$$\phi_i(x_i) = w_0 \cdot \mathbf{quality}(I_{x_i}) + \frac{(1-w_0)}{\max(|\mathbf{soft}(i)|,1)} \sum_{E\in\mathbf{soft}(i)} E(I_{x_i}) \tag{2}$$

where **quality**(I_{x_i}) is a normalized measure of the photo quality (discussed in Section 4), and w_0 can be used as a weighting factor between the soft constraints and the quality of the photo (simply setting $w_0 = 0.5$ worked well in all our examples).

For the set of soft-constraints of each attribute type $\mathbf{S}$ on the edge between nodes i and j, we define the ratio:

$$R(\mathbf{S},i,j) = \frac{\sum_{G\in\mathbf{S}(i,j)} G(I_{x_i},I_{x_j})}{\max(|\mathbf{S}(i,j)|,1)}, \tag{3}$$

where $\mathbf{S} \in \{\mathbf{who}(i,j),\ \mathbf{what}(i,j),\ \mathbf{where}(i,j),\ \mathbf{when}(i,j)\}$. For all edge constraints together, we define the quality measure:

$$\begin{aligned} \psi_{i,j}(x_i,x_j) &= \\ w_1R(\mathbf{who},i,j) &+ w_2R(\mathbf{what},i,j) + \\ w_3R(\mathbf{where},i,j) &+ w_4R(\mathbf{when},i,j) \end{aligned} \tag{4}$$

We use $w_1 = w_2 = w_3 = w_4 = 0.25$ as weighting factor between types. Note that because $G(\cdot,\cdot)$ can have a negative value as penalty when the constraint is not satisfied, $\psi_{i,j}(x_i,x_j)$ can also have a negative value.

Now, we are ready to define the overall fitness function for the assignment $\boldsymbol{x}$ of nodes to photos of the whole story:

$$\mathbf{fitness}(\boldsymbol{x}) = \frac{1}{|\mathsf{V}|}\sum_{i\in\mathsf{V}} \phi_i(x_i) + \frac{1}{|\mathsf{E}|}\sum_{i,j\in\mathsf{E}} \psi_{i,j}(x_i,x_j) \tag{5}$$

Finding an Assignment In theory, the number of possible assignments of nodes to photos is N^K, which is exponential. It is impractical to check all possibilities and, due to the nature of our formalization, a heuristic search algorithm is needed to find a solution that satisfies the constraints and has the highest "fitness". Using known heuristic searches such as A* or beam-search demand some estimation of the fitness of the final solution based on partial solutions, as well as some memory cost to store the partial solutions. Instead, we suggest an algorithm that combines classic CSP techniques with a stochastic search using a two stage approach tailored to our problem.

```
// -- Initialization --
$x \leftarrow \emptyset$;
for every node $i$ in story graph $G$ do
    $C_i \leftarrow \emptyset$;
    for every photo $I_k \in S$ do
        if $\xi_i(I_k) = 1$ then $C_i \leftarrow C_i \cup k$ ;
    end
end
// -- Step1: Topological Sorting --
for every node $i$ in story graph $G$ do
    insert $i$ into priority queue P based on increasing $|C_i|$
end
while P not empty do
    node $i \leftarrow$ top P;
    if $|C_i| = 0$ then break and exit (no solution) ;
    if $|C_i| > 1$ then break and continue to step 2 ;
    get the photo $I_k$ from $C_i$;
    $x_i = k$ (assign $I_k$ to node $i$);
    for all nodes $j \in P$ do
        remove $k$ from $C_j$;
    end
    dequeue node $i$;
end
if $x$ is a full assignment then
    exit ($x$ is the only solution);
else
    $x_{\text{partial}} = x$;
end
// -- Step2: CSP search --
bestFit$\leftarrow \infty$, $x_{\text{best}} \leftarrow \emptyset$ ;
for $t$ = 1 to $M$ do
    // -- initialize a new solution --
    $x = x_{\text{partial}}$, set $\eta(t)$;
    for every node $i$ remaining in P do
        insert $i$ into priority queue Q based on $f(i,t)$ (eq. 6)
    end
    // -- find a full solution --
    while Q not empty do
        node $i \leftarrow$ top Q;
        if $|C_i| = 0$ then break and continue to next solution ;
        get highest quality photo $I_k \in C_i$;
        $x_i = k$ (assign $I_k$ to node $i$);
        for all nodes $j \in P$ do
            remove $k$ from $C_j$;
        end
        dequeue node $i$;
    end
    // -- compare to best so far --
    if full assignment $x$ found then
        fit = evaluate-fitness($x$) (eq. 5) ;
        if fit < bestfit then
            bestfit $\leftarrow$ fit;
            $x_{\text{best}} \leftarrow x$;
        end
    end
end
return $x_{best}$
```

Figure 4: The algorithm for assignment of photos from input set S to the nodes of the story graph G.

In many cases, the hard constraints in the graph combined with a given set of photos, limit the possible assignment of specific nodes to a single photo. We call these nodes *strongly constrained*. In the first stage, we use topological sorting to find the strongly-constrained nodes and remove them, and their assigned photos, from the search in the next stage. In the second stage, we use a stochastic search where each internal search step uses a CSP algorithm that combines a random factor to escape local minima of Eq. 5.

Initially, we create a set C_i of candidate photos that can be assigned to each node i based only on the hard constraints. We loop over all nodes, and for each node i we loop over all photos and evaluate $\xi_i(j)$ for each photo j. If $\xi_i(j) = 1$, i.e. photo j matches the hard constraints of node i, we loop over all possible assignments of characters in photo j to relevant characters in node i and add a copy of photo j to C_i with this assignment. To find the set of nodes that are strongly constrained we sort the nodes in a priority queue based on the number of photos in the set of candidates, $|C_i|$. The lower this number is, the higher the priority. We loop and extract the top queue node, i.e. the one with the current lowest $|C_i|$. If $|C_i| = 0$, this means that we found a node that cannot be assigned to any photo and in this case there is no solution using this photos set. If $|C_i| = 1$, i.e. this node's candidates set contains a single photo, we assign this photo to the node, and remove it from the candidates sets of all other nodes. We continue this until we find a node where $|C_i| > 1$, or the queue is empty. If the queue is empty – only one solution exists, and we are done. If there are still nodes in the queue, then each of them has more than one candidate photo from the input set, and we continue to the next search stage.

In the second stage, we have several possible assignments for every node i (i.e. $|C_i| > 1$), and we want to find the best assignment accounting for both edge and node constraints in the graph, by measuring "fitness". Since all candidate photos in C_i are compatible with the hard constraints on node i, we sort them according to their fitness measure $\phi_i(x_i)$. Similar to the first stage, we build a priority queue for nodes, but this time the priority of node i is defined as follows:

$$f(i,t) = \alpha(1 - \frac{|C_i|}{\max_i |C_i|}) + (1-\alpha)\frac{\text{inDegree}(i)}{\max_i \text{inDegree}(i)} + \eta(t) \quad (6)$$

This priority favors nodes that have fewer candidates in C_i and less in-coming constraints from other nodes, α is a weighting factor (we use $\alpha = 0.5$), and $\eta(t)$ is a random factor that depends on the number of iterations t (see below, we use a random value between 0 to $1-(t/\text{maximum loops})^2$). Next, we loop on the queue extracting the top node i. If $C_i = \emptyset$, this means again, that we found a node that cannot be assigned to any photo, and there is no solution using this specific partial-assignment. Otherwise, we pick the top fitting photo from C_i, assign it to node i, and remove it from the sets of all other nodes. Once the queue is empty, we found an assignment that satisfies all hard-constraints, and we evaluate its fitness based on Eq. 5.

We apply this CSP procedure M times (we use $M = 300,000$, see section 8 for a discussion on M), lowering the randomness factor in each iteration, and comparing each solution to the best one so far. In the end, we return the solution with best fitness found, or an empty solution if none is found. Figure 4 summarizes the steps of our method in pseudo-code. Since the fitness value of Eq. 5 is normalized by the number of nodes and edges, it also provides a way to compare assignment of the same photo collection to different stories. This allows picking the best story that fits a given photo collection.

We denote K as the number of nodes in the graph, N as the number of photos in the photo set I, and D as the maximum number of characters in a photo. In the worst case (i.e., when every photo matches every node, and all photos contain D characters) the complexity of building the candidate set for all nodes is $O(NKD)$. In practice, however, the number of possible assignments of relevant characters in a photo is very small, and the average number of photos that match a node is a constant (see Table 1), so we are left with $O(N)$. The CSP search runs for M loops and in each loop empties a priority queue of size $O(K)$. However, in the worst case in each loop it also touches or removes all photos from all the nodes in the graph (until a solution is reached or it gets stuck). The total number of photos is again $O(NKD)$ in the worst case, but in practice can be seen as $O(N)$. Therefore, the complexity of the CSP stage is $O(M(N+K\log(K)))$, which is also the complexity of the whole algorithm (Table 1 reports actual runtime on all our examples).

6. Visual Layout

Once we have the assignment of nodes in the story graph to photos, as well as the mapping of all relevant characters to story entities, we create the final visual depiction of the story. The assignment vector $\boldsymbol{x} = (x_1, \ldots, x_K)$ provides an ordered sequence of photos $I_{x_1}, \ldots, I_{x_K}$ that should be displayed in order along with the text stored at their respective nodes.

We use a collection of pre-defined page layout templates that can convey different styles such as a Comics style or a children-book style. Different page templates in a given style differ in either the number, the dimension or the positioning of the photos and text on the page. We first choose a given style and then assign subsets of photos to template pages. After assignment of subsets, each photo will be mapped to one frame by cropping it to fit the frame's dimensions, and fitting the text belonging to the photo's node to its text box. We search for the best one-dimensional cropping window of each photo by measuring the cost using two factors: the saliency map and the faces detected in the photo (see Figure 5). The text of each story node describes the event happening in that node. This text can contain replaceable elements (names of characters, places, items etc.) that are determined by the attributes of the photo assigned to the node. The final text is displayed inside text boxes designated in the layout.

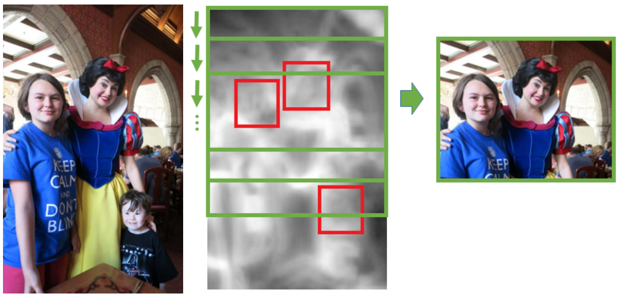

Figure 5: Fitting a photo to a target frame: depending on the frame's aspect ratio (green bounding box in middle), we preserve either the width or height of the original photo (width in this example) and use a 1D search for the best cropping window. The value of the crop is measured using saliency (gray value in middle image) and face detection (red bounding boxes).

Choosing Best Template Set We use a quality measure to evaluate the mapping of a subset of photos to page templates. This quality is defined by the average value of fitting each photo and its text to their frames. More frames in a template allows larger possibilities in arranging the photos (we can always simply fit one or two photos per page but that would make the results seem dull). Therefore, to create larger diversity in the visual results we give preference to templates with a large number of frames. The quality of depicting the full story is defined as the average of the quality of mapping each sub-sequence of photo to their respective page layout template. This allows comparing the layout quality of different sets of page-templates to choose the best one.

To find a best mapping of photo to page templates, the sequence of photos must be separated to sub-sequence according to the number of photos per page. For example, a story with 12 photos (nodes) could be separated to sub-sequences $(4,4,2,2)$, or $(1,2,3,3,2,1)$. For n photos there are 2^{n-1} possible divisions to subsequences with no constraints. Assuming the maximum number of photos per element in the subsequence is limited by the maximum number of frames in a template, the number of possible subsequences is smaller but still exponential. To find a good solution we use iterative local search algorithm (ILS), which is a stochastic optimization framework (see [HS04]).

7. Results

We downloaded several Creative Commons datasets of personal photos from flickr by searching for "Disneyland", "Zoo" or "Machu Picchu" and related tags. Each set separately defines one personal photo collection. In an initial experiment we showed artists some of these collections (different than the ones we used for experiments) and asked them to come up with fictional stories that could be told with such collections. Based on the artists stories we created four story graphs: a search story, where a character is lost and the family is searching for him/her, and a riddle story, where a riddle needs to be solved by the family, a birthday-story of an animal in the zoo and a travel-story to ruins of an ancient civilization. Note that in these stories the whole family is the protagonist, not individual members. Figure 3 shows an illustration of the riddle story

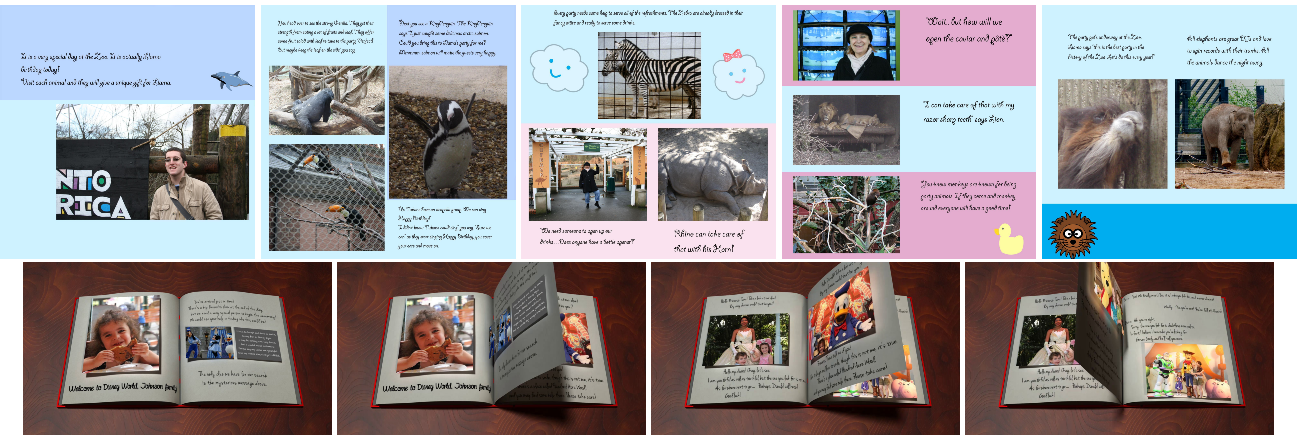
It is a very special day at the Zoo. It is actually Llama birthday today!
Visit each animal and they will give a unique gift for Llama.
NTO
RICA
You head over to see the strong Gorilla. They get their strength from eating a lot of fruits and leaf. They offer some fruit salad with leaf to take to the party. "Perfect! But maybe keep the leaf on the side" you say.
Next you see a KingPenguin. The KingPenguin says "I just caught some delicious arctic salmon. Could you bring this to Llama's party for me? Mmmmm, salmon will make the guests very happy.
Us Tukans have an acapella group. We can sing Happy Birthday!
"I didn't know Tukans could sing" you say. "Sure we can" as they start singing Happy Birthday, you cover your ears and move on.
Every party needs some help to serve all of the refreshments. The Zebra are already dressed in their fancy attire and ready to serve some drinks.
"We need someone to open up our drinks… Does anyone have a bottle opener?"
Rhino can take care of that with his Horn!
"Wait.. but how will we open the caviar and pâtè?"
"I can take care of that with my razor sharp teeth" says Lion.
You know monkeys are known for being party animals. If they come and monkey around everyone will have a good time!
The party get's underway at the Zoo. Llama says "this is the best party in the history of the Zoo. Let's do this every year!"
All elephants are great DJs and love to spin records with their trunks. All the animals dance the night away.
You've arrived just in time!
There's a big fireworks show at the end of the day, but we need a very special person to begin the ceremony! We could use your help in finding who this could be!
Welcome to Disney World, Johnson family
Hello Princess Tiana! Take a look at our clue!
By any chance, would that be you?
Hello my dears! Okay, let's see.
Hello Donald! Take a look at our clue!
By any chance, would that be you?
Woody: No you're not! You're full of deceit!
Ah, you're right.
Go see Goofy and he'll tell you more.

Figure 6: Examples of two more possible presentation types: a children story-books layout style and an animated storybook movie.

graph (the full graphs definition for our stories can be found in the supplemental materials).

For our experiments we used four personal photo collections of people in the park, three for zoo visits, and three for Machu Picchu travels. We chose collections of different sizes ranging from a small number of photos (49) to a large number (629). We wanted them to include portraits of family members (or Disney characters) as well as non-portrait photos such as animal or food photos and scenery. We found that three of the park sets could satisfy both the Riddle and Search stories, while one set could not satisfy the Riddle story as it did not include a photo of two Disney characters together, which is a hard constraint in the story. Obviously, the park sets could not satisfy the zoo or travel stories and vice versa. Table 1 presents some statistics on the running time of the algorithms for the various sets and examples.

We demonstrate three styles of depiction using three different template sets. Figures 1, and 2, show various examples for a comics poster created for the different stories and for different layouts. These results were obtained by selecting the best subset of photos that fit the stories using Eq. 5 and selecting the best layout out of four different layouts based on their quality measure. More examples can be found in the supplemental materials. Two different presentation styles akin to a photo-album are demonstrated in Figure 6. First, a children book style is used where a combination of 6 one-frame page layouts, 7 two-frame page layouts, and 5 three-frame page layouts were used. Then, the results are used to render an animation where a narrator can read the story while the pages are flipped (see accompanying video). This illustrates that different types of media can also be supported by our technique for creating both static and dynamic depictions.

8. Evaluation

Choosing M. The value of maximum-loops M, is a key parameter that governs both the number of solutions checked (and time complexity of the search), and the amount of randomness used (see Eq. 6). We produce a set of results by varying the value of M. Using $M = 10$ is equivalent to a simple *Greedy* approach that chooses

Photo set	**No. of photos**	**Avg. photos per node**	**Mapping time**	**Layout time**
Riddle Story (9 nodes)				
setD1	73	NA	2	NA
setD2	133	14.1	43	14.68
setD3	192	18.7	67	13.34
setD4	629	29	110	12.91
Search Story (12 nodes)				
setD1	73	19.4	90	11.83
setD2	133	20.3	112	11.39
setD3	192	27	143	14.46
setD4	629	46.8	200	12.02
Zoo Story (9 nodes)				
SetZ1	49	10.3	42	12.67
SetZ2	123	15.9	64	14.77
SetZ3	196	18.5	76	13.71
Machu Picchu Story (12 nodes)				
SetM1	50	3.1	13	13.82
SetM2	150	4.4	13	16.73
SetM3	176	12.4	36	12.02

Table 1: Details on the data sets used and running times (in seconds). The third column represents the average number of photos that satisfy the hard-constraints of nodes. setD1 could not satisfy the constraints of the Riddle story already in the topological sorting stage. Mapping time refers to the running time of the mapping algorithm (Figure 4) using $M = 300,000$ iterations. The layout time depends on the number of templates in a given style, the time displayed is for children book style, but for all styles and examples we show it was below 30 seconds.

Which story did you like more?		Which story is more coherent?	
M=10	M=10K	M=10	M=10K
12.5%	**87.5%**	25%	**75%**
M=10K	M=100K	M=10K	M=100K
22%	**78%**	33.3%	**66.7%**

Table 2: Percent of people choosing one story over the other in the comparison tests for varying values of M settings. “No preference” answers were counted as choosing both. Larger M values create better stories.

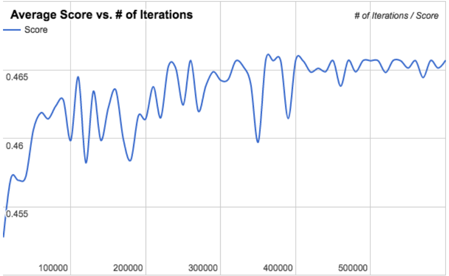

Figure 7: Example of the plot of the average quality of the assignment of nodes to images vs. M, the number of iterations in the algorithm. For each value of M we ran the algorithm three times and averaged the final assignment value. As can be seen quality converges around $M = 200,000$.

nodes based on their fitness directly, while increasing its value increases the randomness in the fitness and also checks a larger number of solutions. We conducted a preliminary study to compare the results produced by the different settings of M. We used the Riddle Story with two different photo sets (SetD3 and SetD4). To remove the effect of layout, all results were created with a story-album template with one photo per page and no cropping. We first showed participants a poster version of the story with SetD1 (Example1 in supplemental material) to familiarize themselves with the story. Then, we used a paired comparison test to compare between results created with $M = 10,000$, and the greedy results with $M = 10$, and between results created with $M = 10,000$ and results created with $M = 100,000$. We asked two questions: “which story seem more coherent?” and “which story did you like more?”. We had 23 participants (11 males and 12 females) whose ages range from 21 to 38, each participant answered one (random) paired test for each of the two sets, where the order of the first story shown was randomized as well. The results of this study are shown in Table 2 and indicate that using larger values of M creates better results. To determine the effective value for M, we plot the averaged quality measure of many results (Eq. 5) for a given M. As can be seen (Figure 7), the average quality converges at around $M = 200,000$ in this experiment and more iterations usually do not help. Since we checked the effect of M only on the Riddle Story we chose to use $M = 300,000$ to create our final results.

Evaluating Results. To the best of our knowledge, there is no alternative method that can produce similar results to ours. Comparing to simple photo sets summarization will not be meaningful as our results provide an additional value – a fictional story – out of the input photo set. In addition, our final results depend on many factors: the story itself, the quality of the input photos, the mapping algorithm, the layout chosen, and the layout algorithm. We support several kinds of layout and can also easily add others, hence, we chose to concentrate our evaluation on the photo mapping and assignment algorithm. This stage is also the most time-consuming and complex (see Table 1).

Comparing to Simpler Alternative. A straw-man alternative algorithm for photo mapping would simply pick random photos from the set and assign them to nodes. Such mapping would very likely create non-coherent results that are not interesting. Instead, we chose to use as a *baseline*, a simple version of our algorithm where only node constraints are imposed. This means that soft constraints and pairwise constraints are ignored. We conducted a paired-comparison test similar to the above between the *Baseline* and our algorithm results. Our results were preferred at 66% and were marked as more coherent at 83.3%. This indicate that using the full set of constraints creates better stories.

Comparing to Manual. We also compare our results to manual selection of photos by humans. To create manual results we introduce the story to the user as a presentation where each page includes an empty frame and the text that goes along with the intended photo (with empty boxes to fill in the words e.g. of story characters). We provide the set of photos in folders where all images of a given character or a given animal have their own subfolder to assist fast lookup for these characters. We ask the user to pick appropriate photos and place them in each page while filling in the corresponding missing words. We tested the Riddle story and the Zoo story, using two sets of photos each: SetD3 containing 192 photos, SetD4 containing 629 photos, SetZ1 containing 49 images, and SetZ3 containing 196 images. We had 5 participants (graduate CS and Art students), that produced two results each, arriving at 5 results for each story, and either 2 or 3 for each set.

The average times for creating the stories were as follows: 6:20 min. for the zoo with the small set, 11:25 min. for the zoo with the large set, 8:40 for the Riddle with the small set and 15 min. for the Riddle with the large set. These results indicate that more photos in the set and more nodes in the story produce a harder task, but other factors also come into play. These factors include the number of constraints, the quality of photos, the number of potential photos that fit a frame and more.

We used a paired comparison tests on Amazon Mechanical Turk by providing turkers with two versions of the same story in the form of on-line presentation: one manual (created as described above) and one created by our algorithm. The order of stories was randomized and counterbalanced and the compared albums were made from same set of photos to ensure fair comparison. For each pair, we asked turkers two questions - which album they thought was more “consistent” and which one they “preferred”. They had three choices for an answer: first album is more consistent/preferred, second album is more consistent/preferred, or two albums are the same in terms of consistency/preference. We solicited 30 responses for each question and pairing arriving at 300 responses all together (2 stories × 5 manually created albums × 30 responses).

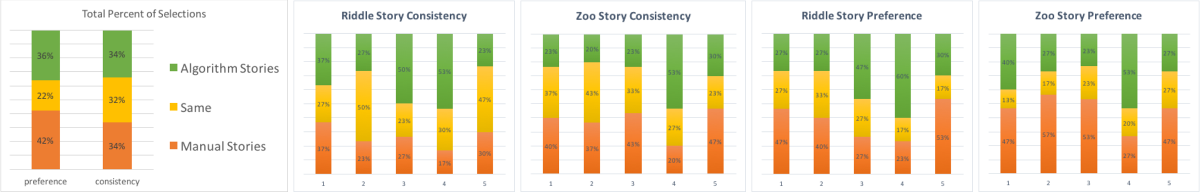

Figure 8: Results of MTurk survey comparing preference and consistency of albums created manually and albums created using our algorithm. Preferences show no statistical difference, meaning one can use our algorithm instead of a manual artist. See details in text.

Figure 8 shows a plot of the aggregated results. These results suggest that there is no statistical difference between the manually created stories and the automatic ones. In terms of consistency the automatic algorithm was actually found to be marginally better than the manually created counterparts. In terms of preference, the manual albums were preferred slightly. This was explained by subjects having more affection/liking towards certain characters over others and the quality of photos in certain cases (e.g. more front-facing characters or more smiling faces). We consider these results promising as they suggest one can use our algorithm instead of having an artist created them. Looking at the breakdown of the results by story type and the subject creating the manual album, we see that not all subject are equally good at creating manual albums. For example, the automatically created albums are consistently preferred in all cases for subject 4, while the opposite is true for subject 2. We also see that the automatic algorithm appears to perform better in the case of the Riddle story compared to the Zoo story, meaning the choice of story also matters.

Evaluating Motivation. Lastly, we showed study subjects several examples of our story albums and asked two questions: “if your family was the one shown, would you have printed this story?”, and “if you had a choice, would you prefer getting such a story album or a simple photo-album of photos?”. Out of 23 participants, 11 answered they would print the story, 7 answered they would print it if they were (or had) kids, 2 answered ‘maybe’, and only 3 answered ‘no’. For the second question, 12 replied they prefer the story-album, 9 preferred a simple photo-album and 2 wanted both. These answers suggest that there is interest in creating such story-albums even if they are fictional.

9. Discussion

We have presented a method for the creation of personal fictional stories in various styles from a photo collection. By defining a story as a graph of constraints we can examine whether a certain set can be used to depict a given story, and we can also measure how well the photo set fits this story. After choosing a sequence of photo to represent the story, visual depictions in different styles are created using predefined templates. To conform to different templates, each photo is cropped to fit its frame in the template layout, and the text is laid out inside its text box. The division of photos to pages is found using iterative local search optimization.

There are several limitations to our work, and many possible future extensions. First, as indicated by our evaluation, extracting more information from photos can create better stories. Currently, only simple identity, location and time information are used, and even these are not fully solved using automatic computer vision algorithms. Using high-level information in photos such as gaze direction and facial expression can assist in determining the photo quality. Causality relations between photos, if they appear, could also assist in fitting to a story but are still very difficult to extract automatically. Similarly, our cropping algorithm relies on face detection and saliency. However, there are many cases where the target aspect ratio cannot contain some of the faces, and they are cropped out (see e.g. Figure 5).

Second, in terms of optimization, our method works in two stages - first finding the set of photos that best fits the given story, and then creating the best layout. A more global approach would combine the two stages to optimize the best set of photos that fits a given story and provides the best depiction as well. For example, by choosing images in the same orientation as their frame in the layout would reduce the need for cropping, and could possibly create better final results.

Interesting stories are always the key to engagement. We presented several example for stories in an amusement park, in a zoo and in a travel setting, but obviously not every story can match any personal photo collection. It would be interesting to examine other scenarios such as a day at the beach or a visit to a city to support other types of personal photo sets. Furthermore, tools for creating story graphs manually or semi-automatically are a fertile direction for future research, and it would also be interesting to try and convert predefined stories to story-graphs automatically. The story graphs themselves could be extended to cover non-linear and multiple discourse stories. This could possibly assist in supporting more photo sets and prevent reaching unsatisfiable conditions.

Acknowledgements

This work was partially supported by the Israel Science Foundation grant no. 324/11. Flickr photographs are from @joefoodie, AngieSix, jdg32373 and danuv

References

[BR75] Bitner J. R., Reingold E. M.: Backtracking programming techniques. In *Communications of ACM* (1975), vol. 18, pp. 651–655. 3

[BSR04] Bartak R., Salido M. A., Rossi F.: New trends in constraint satisfaction, planning, and scheduling: A survey. In *The Knowledge Engineering Review* (2004). 3

[Byr71] Byrne D.: *The Attraction Paradigm*. Academic Press, New York, 1971. 2

[CCK*06] Chen J.-C., Chu W.-T., Kuo J.-H., Weng C.-Y., Wu J.-L.: Tiling slideshow. In *Proceedings of the 14th Annual ACM International Conference on Multimedia* (2006), pp. 25–34. 3, 4

[CDLN07] Crete F., Dolmiere T., Ladret P., Nicolas M.: The blur effect: perception and estimation with a new no-reference perceptual blur metric, 2007. 5

[CGC07] Calic J., Gibson D., Campbell N.: Efficient layout of comic-like video summaries. *IEEE Transactions on Circuits and Systems for Video Technology 17*, 7 (July 2007), 931–936. 3

[CLC14] Cao Y., Lau R. W. H., Chan A. B.: Look over here: Attention-directing composition of manga elements. *ACM Transactions on Graphics 33*, 4 (2014), 94:1–94:11. 3

[CTM*13] Chen T., Tan P., Ma L.-Q., Cheng M.-M., Shamir A., Hu S.-M.: Poseshop: Human image database construction and personalized content synthesis. *IEEE Transactions on Visualization and Computer Graphics 19*, 5 (2013), 824–837. 3

[CTZA10] Chao H., Tretter D. R., Zhang X., Atkins C. B.: Blocked recursive image composition with exclusion zones. In *Proceedings of the 10th ACM Symposium on Document Engineering* (2010), DocEng ’10, pp. 111–114. 3

[CYW15] Chu W.-T., Yu C.-H., Wang H.-H.: Optimized comics-based storytelling for temporal image sequences. *IEEE Transactions on Multimedia 17*, 2 (2015), 201–215. 3

[DCSH15] Dai Q., Carr P., Sigal L., Hoiem D.: Family member identification from photo collections. In *IEEE Winter Conference on Applications of Computer Vision (WACV)* (2015), pp. 982–989. 4

[DJLW06] Datta R., Joshi D., Li J., Wang J. Z.: Studying aesthetics in photographic images using a computational approach. In *Proceedings of the 9th European Conference on Computer Vision* (2006), ECCV’06, pp. 288–301. 3

[DMC10a] Delgado D., Magalhaes J., Correia N.: Assisted news reading with automated illustration. In *Proceedings of the International Conference on Multimedia* (2010), pp. 1647–1650. 3

[DMC10b] Delgado D., Magalhaes J., Correia N.: Automated illustration of news stories. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on* (Sept 2010), pp. 73–78. 3

[FGMR10] Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D.: Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010). 4

[GDDM14] Girshick R., Donahue J., Darrell T., Malik J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Computer Vision and Pattern Recognition* (2014), pp. 580–587. 4

[GZMT12] Goferman S., Zelnik-Manor L., Tal A.: Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 10 (2012), 1915–1926. 5

[HFM*16] Huang T.-H. K., Ferraro F., Mostafazadeh N., Misra I., Agrawal A., Devlin J., Girshick R., He X., Kohli P., Batra D., Zitnick C. L., Parikh D., Vanderwende L., Galley M., Mitchell M.: Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (April 2016). 1

[HLS13] Huang C.-J., Li C.-T., Shan M.-K.: Vizstory: Visualization of digital narrative for fairy tales. In *Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on* (2013), IEEE, pp. 67–72. 3

[HS04] Hoos H. H., Stützle T.: *Stochastic Local Search—Foundations and Applications*. Morgan Kaufmann Publishers, 2004. 7

[JLS*03] Jacobs C., Li W., Schrier E., Bargeron D., Salesin D.: Adaptive grid-based document layout. *ACM Transactions on Graphics 22*, 3 (2003), 838–847. 3

[KDP09] Kormann D., Dunker P., Paduschek R.: Automatic rating and selection of digital photographs. In *Proceedings of the 4th International Conference on Semantic and Digital Media Technologies: Semantic Multimedia* (2009), pp. 192–195. 3

[KRE09] Kublbeck C., Ruf T., Ernst. A.: A modular framework to detect and analyze faces for audience measurement systems. *GI Jahrestagung* (2009). 4

[KSH12] Krizhevsky A., Sutskever I., Hinton G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105. 4

[KX14] Kim G., Xing E. P.: Reconstructing storyline graphs for image recommendation from web community photos. In *27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)* (2014). 3

[LLC10] Li C., Loui A. C., Chen T.: Towards aesthetics: A photo quality assessment and photo selection system. In *Proceedings of the International Conference on Multimedia* (2010), pp. 827–830. 3

[LTWR14] Li B., Thakkar M., Wang Y., Riedl M. O.: Storytelling with adjustable narrator styles and sentiments. In *Interactive Storytelling*, Mitchell A., Fernández-Vara C., Thue D., (Eds.), vol. 8832 of *Lecture Notes in Computer Science*. 2014, pp. 1–12. 3

[LWS*09] Liu T., Wang J., Sun J., Zheng N., Tang X., Shum H.-Y.: Picture collage. *Transactions on Multimedia 11*, 7 (Nov. 2009), 1225–1239. 3

[LWSB08] Loui A. C., Wood M. D., Scalise A., Birkelund J.: Multidimensional image value assessment and rating for automated albuming and retrieval. In *IEEE International Conference on Image Processing* (2008), pp. 97–100. 1

[MRTS07] Montanari A., Ricci-Tersenghi F., Semerjian G.: Solving constraint satisfaction problems through belief propagation-guided decimation. *45 Anual Allerton Conference* (2007). 3

[Obr11] Obrador P.: *Media aesthetics based multimedia storytelling.* Phd thesis, Universitat Politecnica de Catalunya, 2011. 3

[OdOO10] Obrador P., de Oliveira R., Oliver N.: Supporting personal photo storytelling for social albums. In *Proceedings of the International Conference on Multimedia* (2010), pp. 561–570. 1, 3

[PCF03] Platt J. C., Czerwinski M., Field B.: Phototoc: automatic clustering for browsing personal photographs. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on* (Dec 2003), vol. 1, pp. 6–10 Vol.1. 3

[RBHB06] Rother C., Bordeaux L., Hamadi Y., Blake A.: Autocollage. *ACM Transactions on Graphics 25*, 3 (2006), 847–852. 3

[RHGS15] Ren S., He K., Girshick R., Sun J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)* (2015). 4

[RKKB05] Rother C., Kumar S., Kolmogorov V., Blake A.: Digital tapestry. In *Proceedings of IEEE Computer Vision and Pattern Recognition* (2005), CVPR ’05, pp. 589–596. 3

[RW03] Rodden K., Wood K. R.: How do people manage their digital photographs? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2003), CHI ’03, pp. 409–416. 3

[SATV03] Stevens M. M., Abowd G. D., Truong K. N., Vollmer F.: Getting into the living memory box: Family archives & holistic design. *Personal Ubiquitous Comput. 7*, 3-4 (July 2003), 210–216. 3

[SCM03] Schacter D. L., Chiao J. Y., Mitchell J. P.: The seven sins of memory. *Annals of the New York Academy of Sciences 1001*, 1 (2003), 226–239. 2

[Sel11] Sellen A.: Family archiving in the digital age. In *The Connected Home: The Future of Domestic Life*, Harper R., (Ed.). 2011, pp. 203–236. 3

[SMJ11] Sinha P., Mehrotra S., Jain R.: Summarization of personal photologs using multidimensional content and context. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval* (2011), pp. 4:1–4:8. 1

[SRB11] Sandhaus P., Rabbath M., Boll S.: Employing aesthetic principles for automatic photo book layout. In *Proceedings of the International Conference on Advances in Multimedia Modeling* (2011), pp. 84–95. 3

[SRL06] Shamir A., Rubinstein M., Levinboim T.: Generating comics from 3d interactive computer graphics. *IEEE Computer Graphics & Applications 26*, 3 (2006), 30–38. 3

[TGZ10] Tarlow D., Givoni I. E., Zemel R. S.: Hop-map: Effcient message passing with high order potentials. *International Conference on Artificial Intelligence and Statistics* (2010). 3

[UFGB99] Uchihashi S., Foote J., Girgensohn A., Boreczky J.: Video manga: Generating semantically meaningful video summaries. In *Proceedings of the Seventh ACM International Conference on Multimedia* (1999), pp. 383–392. 3

[VCR08] Vicente S., Comogorov V., Rother C.: Graph cut based image segmentation with connectivity priors. In *IEEE Conference on Computer Vision and Pattern Recognition.* (2008), pp. 1–8. 3

[WHY*12] Wang M., Hong R., Yuan X.-T., Yan S., Chua T.-S.: Movie2comics: Towards a lively video content presentation. *IEEE Transactions on Multimedia 14*, 3 (June 2012), 858–870. 3

[WLO12] Wang D., Li T., Ogihara M.: Generating pictorial storylines via minimum-weight connected dominating set approximation in multi-view graphs. In *Proceedings of the conference of Association for the Advancement of Artificial Intelligence (AAAI)* (2012). 3

[YFW05] Yedidia J. S., Freeman W. T., Weiss Y. Y.: Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory 51*, 7 (2005), 2282—2312. 3

[YH00] Yokoo M., Hirayama K.: Algorithms for distributed constraint satisfaction a review. *Autonomous Agents and Multi-Agent Systems 3*, 2 (2000), 198–212. 3

[YS12] Yuan L., Sun J.: Automatic exposure correction of consumer photographs. In *Proceedings of the European Conference on Computer Vision* (2012), ECCV'12, pp. 771–785. 5

[YSF11] Yang C., Shen J., Fan J.: Effective summarization of large-scale web images. In *Proceedings of the 19th ACM International Conference on Multimedia* (2011), pp. 1145–1148. 3

[YSPF13] Yang C., Shen J., Peng J., Fan J.: Image collection summarization via dictionary learning for sparse representation. *Pattern Recognition 46*, 3 (Mar. 2013), 948–961. 3