Flux: Enhancing photo organization through interaction and automation

Dominikus Baur, Otmar Hilliges, and Andreas Butz

{dominikus.baur|otmar.hilliges|andreas.butz}@ifi.lmu.de
University of Munich, LFE Media Informatics, Amalienstrasse 17, 80333 Munich, Germany

Abstract. In *Flux*, an application for digital photo collections, we provide methods for organizing and sharing photos in a convenient manner based on real-world physical interaction on a tabletop system. Additionally, the system supports four different orders based on quality, time, similarity and a combination of the latter two. The problem of scalability, which is especially relevant for a real-world sized photo collection is tackled with a combination of manual hierarchical clustering by the user and the automatic construction of "piles" of similar photos (depending on the currently active order) by the system. In this paper, we present the design goals, the visualization and interaction techniques deployed in *Flux*.

1 Introduction

With the advent of digital photography the number of photos in a collection radically increased: Former hindrances for taking photos like costs for development and material and the effort to go to a photo laboratory no longer existed which lead to an abundance of perceived photo opportunities. Still, although the new hardware provided the almost unlimited taking of photos, the corresponding software did not evolve equally fast: Even a hobbyist photographer still has to invest a lot of her time into organizing and pruning her collection in order to ever retrieve a photo again, a situation which in turn leads to the spread of more and more digital "shoeboxes" - akin to their real-world counterparts in being completely unsorted and never looked at again. Still, digital photos have one clear advantage compared to physical ones: They are much easier to endow with metadata that also enables machines to work with an otherwise inaccessible set of pixels.
In this paper we argue that the organization of digital photos can be enhanced by using automatic classification and analysis of images, but only if we also tightly integrate it with fitting interaction concepts. With Flux we created a system for photo collections that provides a natural, touch-based interaction on a tabletop display and relies on a physical metaphor to lower gateway hurdles. Automation and interaction are coupled to show as many photos as is conveniently possible and also let the user focus on certain aspects of the collection.

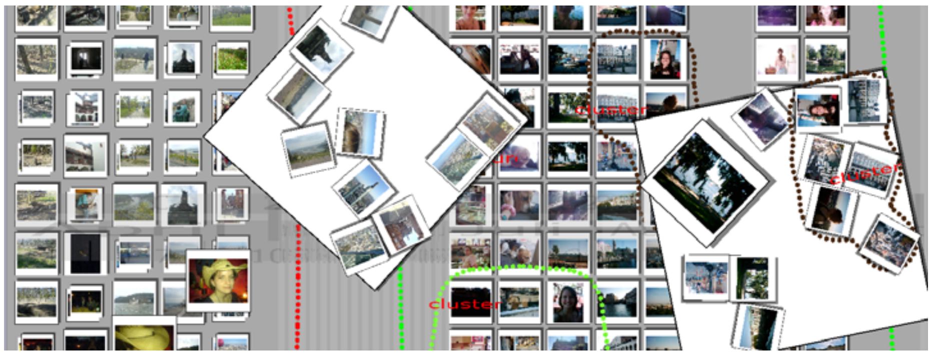

Fig. 1. A photo collection in Flux (time order, two additional workspaces)

2 Flux

2.1 Motivation

The organization of a digital photo collection is a task that grows in complexity and necessary effort with its size. Existing software solutions (see below) only marginally make use of the possibilities of automation and completely rely on the user's ability to categorize and organize the objects manually. This approach is reasonable in so far as the analysis of, for example, visual features still has its limitations: It only works on very low levels like, for example, color or edge distribution, while more sophisticated methods like face recognition are still too unstable under general conditions. But even on such a low level, a multitude of features can be extracted and have to be combined to arrive at one final similarity value which leads to the problem of choosing the right combination and weighting of different features. Additionally, the notion of similarity exists on multiple dimensions and levels of abstraction - while one person might, e.g., find two photos similar because of their bluish color schemes, another one might find them completely unrelated because of their showing different situations or persons. It is highly user dependent and thus should not be delegated to the machine, because especially in the context of a personal photo collection the user is the one to ask (as he also has to be able to retrieve a photo again later). Still, we think that the opportunities given by an automatic analysis can be gathered by coupling it closely with fluid interaction techniques to combine the knowledge of the system with that of the user. In this respect it would be best to let the machine do on its own what it will accomplish successfully or what would be too much bother for the user, then allow the user to adjust the result in a convenient way. To render this interaction more fluid and closer to its real-world counterpart we opted for a physical metaphor and direct manipulation without additional tools. Consequently, we chose a tabletop system as our platform which also corresponds to the typical setting where people work with real photos.

2.2 Related Work

Many applications for organizing photos exist for the hobbyist photographer. They mostly provide an organization scheme borrowed from the operating system's file system and allow for the creation of virtual albums or other basic organization and browsing facilities and searching based on keywords or filenames (Apple iPhoto[1], Google Picasa[2]). A common obstacle in this regard is the size of a collection: Because of practically unlimited storage space the number of photos is easily in the thousands which makes an organization based on the file system hierarchy tediously to navigate and labor-intensive to maintain. Two solutions from the academic world in this regard are a navigation based on either panning and zooming ([1]) or a semantic zoom ([2]). Even more photos can be displayed if redundant information is removed by automatically grouping similar objects, either based on time or low-level features. Examples for the time-based approach are [3], Apple Aperture[3] and Adobe Photoshop Lightroom[4]. An evaluation of the second concept of analysis of low-level features (which uncovered inherent problems) was performed by Rodden et al[4].
Tabletop photo applications are mostly designed around a certain topic, for example, visualizing and working with a whole lifetime of annotated media ([5]) or providing tangible interaction with photos on a hybrid interface ([6]). Two examples for a physical metaphor in computer interfaces are [7] and [8], the former in easing the inhibitions of senior citizens in working with digital photos and the latter in enriching the classical desktop metaphor with physical objects. Flux takes ideas from these examples and tries to provide a convenient environment to work with a personal photo collection. Interaction techniques are based on a physical metaphor of photos and scalability is provided by combining automatic orders with quickly performable clustering.

2.3 Design

Our main goal for Flux was improving the organization of digital photo collections. This task is throughout the literature commonly referred to as *filing* (e.g., [9]) and relies on several other actions: The user has to be able to somehow sift through his collection to get an overview of the content and find certain objects. A purely search-driven approach is not feasible within the highly visual domain of photos, because finding a photo using text is only possible with meaningful keywords that have to be manually added by the user, a task that is only reluctantly performed, if at all ([10]). We therefore chose a browsing mechanism that lets the user quickly narrow in on a section of the collection and then linearly go through the contained photos. This ”narrowing-in” has, of course, to be backed by some kind of hierarchical organization which is exclusively created by the user via easy gestures. Automatic analysis can be tapped in this regard by telling the

[1] www.apple.com/iphoto
[2] picasa.google.com
[3] www.apple.com/aperture
[4] www.adobe.com/products/photoshoplightroom

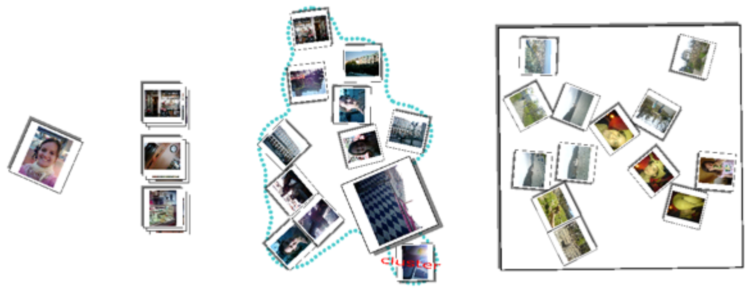

Fig. 2. Interface elements (from left to right): Photo, Piles, Cluster, Workspace

system to arrange the photos according to a certain aspect (time, similarity), but this only works as a support for the user: We did not use automation-driven organization because of the downsides mentioned above. In this regard, scaling becomes an important aspect: If a collection is largely unorganized, many photos appear on the same level and have to somehow be visualized on a limited space. To relieve the user of this task, the system relies on the current order and forms "piles" of similar (visual or temporal) photos, volatile groups that reduce the cognitive load of the user and allow showing more pictures on less space.
Closely related to organization is the deletion of failed or unwanted photos, which Flux supports again in a two-part fashion: The system is able to distinguish low-quality photos based on certain features and can arrange them accordingly, but the final decision whether to delete those is left with the user. Aside from the organization facilities, Flux was also built with sharing in mind, i.e., showing photos to another person, which is especially relevant in a tabletop context. Therefore, quick rotation and scaling are incorporated. Still, Flux is mainly aimed at a single user.

2.4 Overview of Flux

Flux (see figure 1) is a tabletop application for the visualization and organization of photos. Before launching the actual application the extraction of low-level features is performed to produce similarity and quality values. The initial screen shows the complete photo collection arranged by time (the default order). Photos can be combined into hierarchical clusters with a circular gesture. Overlaying workspaces can be created by an attached gesture and contain an arbitrary part of the whole collection. Objects on these workspaces can be more freely manipulated and act like physical objects but still provide the same options for interaction as the more restricted background. The system can sort all photos on a workspace or the background and afterwards automatically builds piles from related photos to save space and reduce clutter. Photos and workspaces are translated, rotated and scaled with two easy gestures.

Visualization: All objects in Flux (see figure 2) belong to a hierarchy (from bottom to top): **Photo objects** have a uniform appearance to make them readily recognizable. **Pile objects** are system-generated groups of similar photos. **Clusters** are user-created groups of photos and/or other clusters, have a color and a name and are presented as colored strings of circles. **Workspaces** act as views on the underlying cluster/photo-model of the collection. One special workspace that cannot be manipulated by the user is the background, which is always visible, non-transformable and always shows the whole collection. The user is able to change the order (see below) of the background, but cannot translate photos or piles on it, so the order is always preserved (to prevent confusion as to whether an object lies at a certain position because of the order or because it was moved there).

Three techniques are used to maximize the users' overview: First, the background works as a fall-back point, where all photos are visible at any time in one given order and copies of them can be created and manipulated on additional workspaces. Second, to increase the number of photos visible, piles are automatically built thus reducing redundancy and visual clutter. Third, photos (on a workspace) and workspaces themselves behave like physical objects in the sense that they collide with one another and thus ensure that no object overlaps another and occludes information.

Four orders are available which can be changed independently for every workspace: Time, Similarity, Quality and Stream. **Time** shows all photos arranged along a calendar that is divided into a fixed number of intervals. The user can scale the time slots and make them either larger or smaller with a one- (moving one vertical border of the time slot) or two-finger gesture (moving both borders). The other time slots shrink or grow respectively. This approach allows a focusing on certain parts of the collection while preserving the complete global context (compared to, for example, a pure zoom-and-pan approach). Piles are built based on chronological adjacency and cluster membership. **Similarity** arranges all main-clusters vertically (and their subclusters horizontally) and builds piles out of visually similar photos. Within **Quality**, all photos are arranged based on their quality on a scale ranging from high quality on one side of the table to low quality on the other. The quality is symbolized by the color of the photos' borders that changes from green to red depending on the value. This quality value is automatically calculated based on the previously extracted low-level features and works with aspects such as blurriness, over-/underexposure, etc. For the sake of simplicity, the algorithm returns discrete values only, so a photo is either treated as "good" or as "bad". The last order, **Stream**, combines Time and Similarity: All clusters are again arranged vertically (subclusters lie close to their parents) with the photos shown along a horizontal axis based on the date they were taken (in this order there is no underlying (global) time line, so all photos are uniformly placed next to each other). The photos are scaled so that the contents of one whole cluster fit into the horizontal space. To optimize the number of photos displayed piles are built, but not purely based on time or low-level similarity but on the combination of the two: Every photo's similarity value is compared to its chronological neighbour's only, so that photos that were taken of the same motif or as snapshots in a row are automatically grouped and the chance that photos that are somehow related are combined into a pile is increased.

Interaction: Interaction in Flux happens on three different levels separated by dwell time (using additional graphical elements like buttons would have increased the on-screen clutter and forced the user to touch a certain section of the object). The user can choose the action's scope by touching the object and determine the type of action itself by waiting. The waiting time for a task depends on how frequently it is used: Transforming an object (translating, rotating, scaling) is readily available. Less common options like forming clusters become active after waiting for 500 miliseconds. Finally, rarely used options like changing the order of photos can be reached by waiting one second. A timeout is visualized by a change of color of the circle surrounding the user's finger or the appearance of a marking menu (depending on the type of action). Three geometric transformations are merged into two gestures: "Rotate'n'Translate" ([11]) rotates an object with a pseudo-physical friction while the user drags it with one finger. "Rotate'n'Scale" ([7]) lets the user set an object's scale with the distance of two fingers and its rotation with their movement. Except for changing appearances, these transformations can be used to delete photos and workspaces by scaling them below a certain threshold and unfolding piles by moving their topmost photo.
After waiting one timeout new clusters can be created by simply drawing a circle around the objects that should be contained and lifting the finger. When drawing a complete circle the circle's color switches and a lasso gesture becomes active: All objects that were selected (i.e., lay within the original circle) are copied to a new workspace, whose center lies at the position where the finger is finally lifted (both gestures are similar to the ones used in [8]).
Workspaces and clusters have marking menus that are shown after the second timeout: The workspace marking menu changes the current order. Cluster marking menus let the user adjust the color (directly with a color wheel) and the name (on an overlay writing field) or delete it. Choices are executed as soon as the user lifts the finger from the table.

Deployment: Flux was written in Microsoft C# within the .NET framework 2.0 and Direct3D 9. We relied on a framework by the University of Munich for low-level feature extraction and analysis, the PhysX™[5] physics engine by Ageia Technologies Inc. for the physical behavior of the objects and the Microsoft TabletPC SDK[6] for handwriting recognition in the naming of clusters.
The system was deployed on a horizontally mounted 42" LCD-screen with a touch-sensitive DViT-overlay[7] by Smart Technologies, which unfortunately limits the number of concurrent input points to two.

[5] www.ageia.com/physx
[6] msdn2.microsoft.com/en-us/windowsvista/aa905027.aspx
[7] smarttech.com/DViT/

2.5 Discussion

Flux uses a combination of interaction and automation to ease the organization of a photo collection: The user gives a general structure by defining clusters and the enlarging and shrinking of parts of the collection in the Time-order, but the myriad of decisions of how to visualize the result (what objects to put where, what photos to use to build a pile) is left with the system. After presenting a visualization the user is still able to completely change it or force the system to repeat the process. Using this close coupling thus works better than an approach based purely on automation (that ignores the user's knowledge of the collection) or interaction (the leaves tedious tasks, e.g., checking the quality of a photo, with the user). Important in this regard is that all automatically performed actions are reversible (e.g., unfolding a pile to access photos) and that the underlying visualization mechanisms are graspable for the user by using animations. Still, limitations to this approach exist, especially when first starting Flux without a high-level organization by the user: The system is then forced to build piles of distinct photos that have to be tediously sorted into clusters. Here, the combination of the two backfires and even produces worse results than a standard scrollable or zoomable approach.
So, one point of improvement would be providing support for the initial organisation in a two-step fashion, where a pre-clustering can be quickly performed by the users which then leaves them with top-level clusters that provide an easier entry into the collection. Another option would be to automatically create initial clusters based on time ([3]). Additionally, large collections often become relatively cluttered especially on the background. A solution to this problem might be dropping the idea of complete overview at all times and allow the user to set the focus of his attention independently of the current order, for example, by allowing him to scale clusters on the background, thus reducing their recognisability but increasing the overall clarity.

3 Conclusion

In this paper we presented Flux, a photo organization tool with a focus on interaction and automation. By the close combination of those two a convenient workflow. For our current system, a formal evaluation is needed to gather if our design succeeded and to find overlooked weaknesses.
In a successor to Flux we might change the available orders - two of the four did not yield the expected results: Neither (binary) quality values nor one-dimensional similarity values should be visualized on a two-dimensional plane. The Similarity-order might be improved by changing the placement of clusters based on the average similarity of their members, so a visual Query-By-Example becomes possible. Still, it is doubtful whether a pure similarity view is useful at all ([4]) - a future version might merge the four orders into one with an emphasis on the chronological order ([10]) - possibly combining the Stream-order with the adaptability of time slots from Time and marking low-quality photos (e.g., displaying a small symbol on their border). Together with unlimited scaling of

clusters on the background such a system would in all likelihood be more powerful and flexible than the current version and further elaborate on our concept of the coupling of interaction and automation.

References

1. Bederson, B.B.: Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In: UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology, New York, NY, USA, ACM (2001) 71–80
2. Huynh, D.F., Drucker, S.M., Baudisch, P., Wong, C.: Time quilt: scaling up zoomable photo browsers for large, unstructured photo collections. In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM (2005) 1937–1940
3. Cooper, M., Foote, J., Girgensohn, A., Wilcox, L.: Temporal event clustering for digital photo collections. ACM Trans. Multimedia Comput. Commun. Appl. **1**(3) (2005) 269–288
4. Rodden, K., Basalaj, W., Sinclair, D., Wood, K.: Does organisation by similarity assist image browsing? In: CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2001) 190–197
5. Shen, C., Lesh, N., Vernier, F.: Personal digital historian: story sharing around the table. interactions **10**(2) (2003) 15–22
6. Hilliges, O., Baur, D., Butz, A.: Photohelix: Browsing, sorting and sharing digital photo collections. In: Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems(TABLETOP'07), Los Alamitos, CA, USA, IEEE Computer Society (2007) 87–94
7. Apted, T., Kay, J., Quigley, A.: Tabletop sharing of digital photographs for the elderly. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 781–790
8. Agarawala, A., Balakrishnan, R.: Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In: Proceedings of CHI '06. (2006) 1283–1292
9. Kirk, D., Sellen, A., Rother, C., Wood, K.: Understanding photowork. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 761–770
10. Rodden, K., Wood, K.R.: How do people manage their digital photographs? In: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2003) 409–416
11. Kruger, R., Carpendale, S., Scott, S.D., Tang, A.: Fluid integration of rotation and translation. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2005) 601–610