

PREFERENCE-BASED ADAPTATION OF MULTIMEDIA PRESENTATIONS FOR DIFFERENT DISPLAY SIZES

Yamini Nimmagadda, Karthik Kumar and Yung-Hsiang Lu

School of Electrical and Computer Engineering, Purdue University, Indiana, USA

ABSTRACT

The increased availability of electronic devices with multimedia capabilities has resulted in a dramatic growth in the usage of multimedia presentations. In this paper, we present a framework for automatic generation of layouts for multimedia presentations for different screen resolutions. We develop a heuristic to find the layout for a screen of given resolution in polynomial time. The heuristic achieves 92-98% performance compared with the exhaustive search for different screen sizes. The execution time of the heuristic is as low as 15 seconds for a presentation of 100 media files whereas the exhaustive search takes an average of 6 hours.

Index Terms—content adaptation, display sizes, multimedia presentation, layout generation

1. INTRODUCTION

Advances in mobile technology result in the development of mobile devices like cellular phones and PDAs with smaller display sizes. At the same time, bigger displays are preferred for stationary uses like informational kiosks in airports. Availability of the Internet and increase in the number of channels lead to a large increase in the display of multimedia presentations on these devices. Multimedia presentations are collections of different media files (called “media objects” in this paper) like text, images, videos, and animations with different resolutions, durations, and start-times. An example of a multimedia presentation is a news broadcast with an anchor, weather reports, sports, and commercials.

The layout of multimedia presentations is defined by the locations and the start times of the objects. The layout designed for one display cannot be easily used for other sizes. Rather than requiring authors to create layouts explicitly for each size, the layout should be generated automatically based on the preferences of authors. For example, an author prefers to display a commercial on sports shoes to rain coats when sports news are shown; rain coats is preferred when a weather report is shown.

In this paper, we present automatic preference-based adaptation for multimedia presentations characterized by temporal constraints that specify start times, deadlines and display times of objects. Existing studies on content adap-

tation [1–5] consider objects with the same start times and durations. In contrast, we present adaptation for objects with different start times and durations. The following is an example of a presentation on basketball games with preferences and temporal constraints. The superscripts + and – denote displaying and hiding an object. $A \triangleright B$ means A is more important than B. $A > B$ denotes A is preferred to B. $C : A > B$ means for given C, A is preferred to B. The following is a list of objects and preferences in the presentation. In this list, the ordered pairs correspond to the display times and the resolutions of the objects respectively.

- Video V (60s, 640×480) showing a practice for the first 25s and interviews with players for the next 35s. $V^+ > V^-$.
- Schedule S of games for the year (20s, 900×720). The schedule should be displayed when the practice is shown. $S^+ > S^-$ and $V \triangleright S$.
- Next game N (30s, 400×50). The author prefers displaying N if S cannot fit on the screen. $S^+ : N^- > N^+$ and $S^- : N^+ > N^-$.
- Player profiles P (30s, 600×120) during the interview with players. $V \triangleright P$.
- Commercials $C1$ (35s, 600×120) and $C2$ (35s, 1280×300) on snow boots and roller skates respectively. Preferences on $C1$ and $C2$ depend on the location L (Indiana IN or California CA) of the viewer. $P \triangleright C1$, $P \triangleright C2$, $IN : C1 > C2$ and $CA : C2 > C1$.

Figures 1(a) and (b) show two screenshots of the presentation at 0s and 30s on a bigger screen (resolution: 1600×1200) for a viewer in California. Hence, $C2$ is displayed. Figures 1 (c), (d) and (e), (f) show two pairs of screenshots of the same presentation adapted using two different layout strategies (described in Section 3.2) on a smaller screen (resolution: 800×600) for a viewer in Indiana. $C1$ is preferred to $C2$ in this case. On the smaller screen, N and $C1$ cannot be displayed at the same time. N is selected using one strategy and $C1$ is selected using the other. In all these figures, the black background represents unused regions of the screens.

The start times of some of the objects are also different for different screen resolutions and different layout strategies.

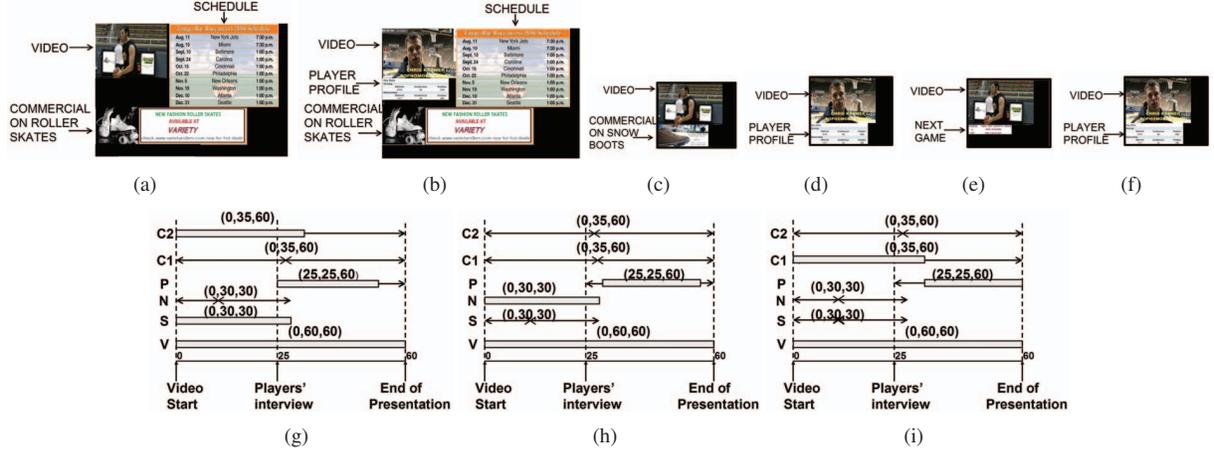


Fig. 1. (a) and (b) show screenshots of the presentation on a screen with resolution 1600×1200 . (c) and (d) show screenshots of the same presentation on a screen with resolution 800×600 . (e) and (f) show screenshots of the same presentation on the screen of size 800×600 again, using a different layout strategy. (g),(h) and (i) show time-line representations of the above layouts.

Figures 1 (g), (h), and (i) show the time line representations of the objects; the triplets represent (start time, duration, end time). A rectangular block represents the duration of an object, whereas the arrows indicate that the objects can start earlier (left arrow) or later (right arrow). The crosses indicate that the objects are not displayed. The start times of P are different in Figures 1 (g), (h), and (i). On the bigger screen, P starts at 25s. The durations of $C1$ and N are 35s and 30s respectively; hence P starts at 30s and 35s in Figures 1(h) and (i) because both N and P or $C1$ and P cannot fit on the screen.

2. RELATED WORK

Several adaptation methods have been developed for multimedia content based on display sizes. Transcoding renders file resolutions suitable for display sizes [1] [2]. Transmoding converts rich media like videos and images to simple media like text [3]. Extraction of hierarchy displays higher level objects and link them to objects in lower levels [4] [5]. Preferences specified by authors are considered in some studies [6] [7]. However, the placement of objects is entirely specified by the author and the temporal aspect is not considered in these studies.

We use Trade-off Conditional Preference nets (TCP-nets) [8] to represent the preferences on the media objects. The nodes in a TCP-net correspond to the objects; the edges represent the relationship between the objects. The TCP-net of the presentation in Section 1 is shown in Figure 2. The directed edge with a triangle drawn from V to S indicates that $V \triangleright S$. The directed edge without a triangle drawn from S to N shows that values of N depend on values of S ($S^+ : N^- > N^+, S^- : N^+ > N^-$). Conditional preferential de-

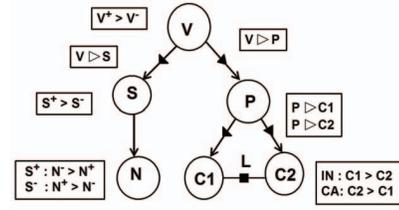


Fig. 2. TCP-net of the presentation in Section 1.

pendence is represented by an undirected edge with a square; for example the preferences on $C1$ and $C2$ depend on L .

3. PREFERENCE-BASED ADAPTATION OF MULTIMEDIA PRESENTATIONS

Adaptation of multimedia presentations for different display sizes poses two problems. The first problem is selecting objects for different display sizes. The objects with higher importance should be displayed while those with lower importance can be hidden. The importance of objects is computed from TCP-nets as shown in Section 3.1. The second problem is inefficient usage of screen areas called “fragmentation”. Fragmentation occurs due to different start times and durations of objects. Figure 3 illustrates fragmentation when three objects A , B , and C start at times $t = t_0, t_1, t_2$ respectively. The first row shows that C cannot fit on the screen because it overlaps with B . The second row shows an arrangement of A , B , and C without fragmentation. Hence, our method has two steps: (1) computing importances of objects (2) finding the spatio-temporal arrangement of objects.

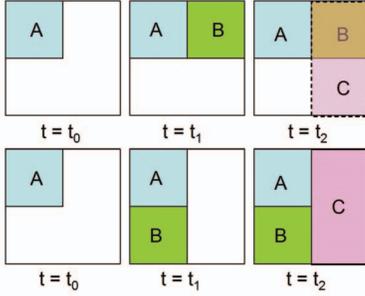


Fig. 3. The top row shows fragmentation. The bottom row shows an arrangement of objects without fragmentation.

3.1. Importance of Media Objects

During the adaptation of multimedia presentations, some objects cannot be displayed as preferred. For example, in Section 1, either V or S can be displayed on a screen of resolution 800×600 . However, the author prefers displaying both objects. In this case, V should be selected over S because $V \triangleright S$. The objects that are more important than the others should be selected. Importance in TCP-nets is a partial order. We propose a method to generate a total order of importance of the objects.

In a TCP-net, object A is called a parent of object B , if A is more important than B or if the preferences of B depend on A . Parent nodes have higher importance and importance satisfies transitivity. Importance of an object depends on the number of ancestors and the area occupied by the object. Objects with higher numbers of ancestors are less important because their preferences depend on many objects. The importance of an object i is higher than the importance of any object k if $|A(k)| > |A(i)|$, where $|A(i)|$ and $|A(k)|$ are the number of ancestors of i and k . Among objects with the same number of ancestors, importance is determined based on the area occupied by the objects on the screen. The total screen area available for the presentation is the product $A_S D_S$ of the screen area A_S and the duration of the presentation D_S . If object i has a duration D_i and an area A_i , then the portion of the area occupied by the object i is $f_i = \frac{A_i D_i}{A_S D_S}$. Among the objects with the same number of ancestors, the importance is higher for objects with larger areas and longer durations. Hence, the importance is given by $\rho_i = \max(\rho_k) + f_i$, for all k such that $|A(k)| > |A(i)|$.

Each object in the presentation can either be displayed or hidden. For the presentation described in Section 1, $V^+ > V^-$, $S^+ > S^-$ and $S^+ : N^- > N^+$. Hence, the most preferred selection is $\{V^+, S^+, N^-\}$, and the least preferred selection is $\{V^-, S^-, N^-\}$. We define a metric called “preference value” for each object to quantify the desirability of a selection. For each object, we assign weights in the ascending order starting from the least preferred selection to the most preferred selection. In our example, we can either display the object or hide it; hence we use 0 and 1 for the least

and the most preferred selections. The preference value of an object is dependent on the parent nodes. If the selection is $\{V^+, S^+, N^-\}$, the preference value of N is 1. If the selection is $\{V^+, S^+, N^+\}$, the preference value of N is 0 because N^- is preferred to N^+ when V^+ and S^+ are chosen.

3.2. Spatio-Temporal Arrangement of Objects

The objective function that needs to be maximized for preference based content adaptation is $F = \sum_{i=1}^N v_i \rho_i$. Here N is the number of objects, v_i is the preference value and ρ_i is the importance of the object i . The function F considers the preference values and the importance of the objects. Maximizing F increases the number of selected important objects. Hence, F is used as a measure of performance. The following constraints should be met: (1) The location of objects should be fixed for their entire durations. (2) Each object should either be displayed for its minimum duration or should not be displayed at all. (3) Two objects cannot overlap spatially and temporally.

We develop an algorithm called “Greedy Maximum Usable Empty Rectangles” (GMUER) to find the layout of a presentation. This algorithm is similar to KAMER proposed in [9], but uses a different fitting rule and considers the temporal information of the objects. We partition the empty area on the screen into maximum empty rectangles. The term maximum is used because these rectangles cannot be contained within any other empty rectangle. Each media object is placed at the top left corner in one of the empty rectangles. We select this empty rectangle such that the new maximum empty rectangles formed after placing the object can accommodate as many unplaced objects as possible.

Fragmentation is reduced by considering temporal overlap between the objects. If the temporal overlap between objects A and C is greater than the temporal overlap between A and B , placement of C is considered before B , even if C starts later than B . We allocate space for C on the screen before placing B . However, due to the allocation of space for C , the objects that are more important than C and that start before C may not fit on the screen. To solve this problem, the space allocated for C is released and the placement starts to accommodate the objects with higher importance. We use the term “back-tracking” to describe the process of releasing space allocated for less important objects by tracking back few steps in order to accommodate more important objects. In the worst case, back-tracking results in tracking all the way to the placement of the first object, increasing the execution time tremendously. Hence, we control the number of steps the algorithm back-tracks.

4. IMPLEMENTATION AND RESULTS

We use C# to implement and evaluate the performance of our algorithm (GMUER) on a Intel Core 2 Duo processor with

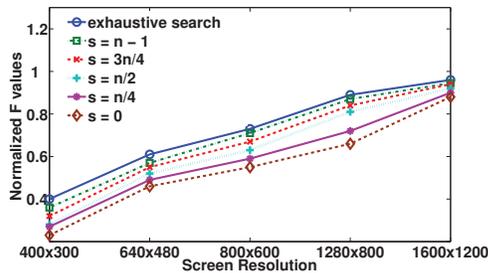


Fig. 4. Normalized F values vs. different screen resolutions. The number of objects in the presentation is n and the number of back-track steps is s . Exhaustive search achieves the highest performance. The performance increases with the screen resolutions because more objects can be displayed on screen.

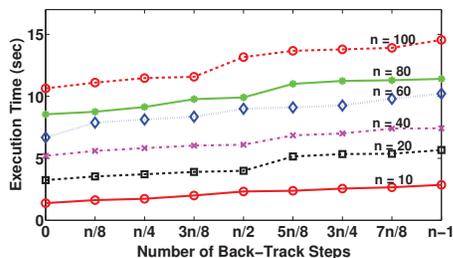


Fig. 5. Execution time vs. number of back-track steps for different numbers of objects n in presentations. The number of back-track steps correspond to the nearest integers less than or equal to the values on the x-axis.

2GHz speed and 2GB RAM. Synchronous Multimedia Integration Language (SMIL) is used to specify the locations, start-times and durations of objects based on GMUER. Figures 1 (c), (d), (e), (f) show the presentation in Section 1 on a screen of resolution 800×600 for a viewer in Indiana. Figures 1 (c) and (d) show the screenshots without back-tracking. The temporal overlap between V and $C1$ is higher than the overlap between V and N . Hence, $C1$ is displayed and N is not displayed, though N is more important than $C1$. Figures 1 (e) and (f) show the screen-shots with a back-tracking of 2 steps; here N is displayed instead of $C1$. Figure 1 (e) is better than Figure 1 (c) because back-tracking selects N , which is more important than $C1$.

The performance of our algorithm is measured by the function F as described in Section 3.2. Figure 4 shows the averages of F values normalized to the maximum value of F at different resolutions. The maximum value of F is computed by assuming that all the objects can be displayed as per the preferences on a screen of infinite resolution. Figure 4 shows the performance obtained by averaging normalized F values for multimedia presentations with different number of objects (1 to 100). Higher values on y-axis indicate that the F values are closer to the maximum F value; hence the performance is higher. Smaller screens limit the number of objects that can be displayed; hence the performance is lower.

We use exhaustive search as the baseline for comparison because it finds the optimal solution. Our algorithm achieves 92%-98% performance of the exhaustive search for different screen sizes.

Figure 5 shows the execution time versus the maximum number of steps back-tracked for presentations with different numbers of objects. The execution time increases with the number of back-tracking steps and the number of media objects. The worst case execution time is intractable because the execution time also depends on the screen size. The average execution time for exhaustive search is 63 min for 10 objects and 6 hours for 100 objects whereas GMUER takes an average of 2s for 10 objects and 15s for 100 objects.

5. CONCLUSION

We present automatic generation of spatio-temporal arrangement of media objects in a multimedia presentation. We propose a heuristic to find the placement of objects on a given screen. The heuristic achieves 92%-98% performance for different screen resolutions compared with the exhaustive search. The average execution time of the heuristic is 15 seconds for 100 objects.

Acknowledgments: This project is supported in part by NSF CNS-0347466 and CCF-0541267. Any opinions, findings, and conclusions or recommendations in the projects are those of the investigators and do not necessarily reflect the views of the sponsors.

6. REFERENCES

- [1] Stephen J. H. Yang, Irene Y. L. Chen, and Rick Chen, "Applying Content Adaptation to Mobile Learning," in *ICICIC*, 2007.
- [2] Javier Soriano and Genoveva Lopez and Miguel Jimenez and Rafael Fernandez and Juan J. Hierro, "Semantic Web Content Adaptation and Services Delivery on Morfeo's Semantic Mobility Channel," in *MDM*, 2006.
- [3] Anthony J. Solon, Kevin Curran, and Paul Mc Kevitt, "Bandwidth Determined Transmoding Through Fuzzy Logic in Mobile Intelligent Multimedia Presentation Systems," *Journal of Artificial Intelligence Review*, vol. 26, no. 4, 2006.
- [4] Stephen J. H. Yang and Irene Y. L. Chen, "Universal Access and Content Adaptation in Mobile Learning," in *ICALT*, 2006.
- [5] Wai Yip Lum and Francis C. M. Lau, "Relationship-aware Content Adaptation of Structured Web Documents for Mobile Computing," in *ICPADS*, 2005.
- [6] Ronen I. Brafman and Doron Friedman, "Adaptive Rich Media Presentations via Preference-Based Constrained Optimization," in *IJCAI Workshop on Advances in Preference Handling*, 2005.
- [7] Iqbal Mohamed, Alvin Chin, Jim Chengming Cai, and Eyal De Lara, "Community-driven Adaptation: Automatic Content Adaptation in Pervasive Environments," in *Workshop on Mobile Computing Systems and Applications*, 2004.
- [8] Ronen I. Brafman and Carmel Domshlak, "Introducing Variable Importance Tradeoffs into CP-nets," in *UAI*, 2002, pp. 69-76.
- [9] Kiayash Bazargan, Ryan Kaslner, and Majid Sarrafzadeh, "Fast Template Placement for Reconfigurable Computing Systems," in *IEEE Design and Test of Computers*, 2000, pp. 68-83.