Image-Driven View Management for Augmented Reality Browsers

Raphaël Grasset* Tobias Langlotz† Denis Kalkofen‡ Markus Tatzgern § Dieter Schmalstieg¶

Graz University of Technology

Figure 1: Common labeling as used in many AR browsers (Left) compared to our image-based approach (Right). Position of the labels can be automatically optimized, but also their appearance, including depth cue for the labels' anchor or their leader lines.

Abstract

In this paper, we introduce a novel view management technique for placing labels in Augmented Reality systems. A common issue in many Augmented Reality applications is the absence of knowledge of the real environment, limiting the efficient representation and optimal layout of the digital information augmented onto the real world. To overcome this problem, we introduce an image-based approach, which combines a visual saliency algorithm with edge analysis to identify potentially important image regions and geometric constraints for placing labels. Our proposed solution also includes adaptive rendering techniques that allow a designer to control the appearance of depth cues. We describe the results obtained from a user study considering different scenarios, which we performed for validating our approach. Our technique will provide special benefits to Augmented Reality browsers that usually lack scene knowledge, but also to many other applications in the domain of Augmented Reality such as cultural heritage and maintenance applications.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented and virtual realities; H.5.2 [Information Interfaces and Presentation]: User Interface—User interface management system (UIMS)

*e-mail:raphael.grasset@icg.tugraz.at
†e-mail:langlotz@icg.tugraz.at
‡e-mail:kalkofen@icg.tugraz.at
§e-mail:tatzgern@icg.tugraz.at
¶e-mail:schmalstieg@icg.tugraz.at

1 Introduction

Augmented Reality (AR) presents digital information registered to real world objects and places. This allows to annotate real world buildings and places with textual or pictorial information. An *Augmented Reality Browser* (ARB) is a new type of commercial outdoor AR application, which makes use of labels to allow end-users to visualize, browse and search digital data in the context of their real world environment. Such systems provide digital information – e.g., on entertainment places or cultural heritage monuments – overlaid on top of a smartphone's video feed.

In an ARB, the digital information is usually registered based on pure geographical location, usually given as a *point of interest* (POI) with corresponding GPS position. Consequently, no further scene knowledge, such as a 3D model of the environment, is available to the system. But even if a 3D model was provided, the error-prone registration of sensor-based tracking would not permit an efficient use of the additional scene knowledge. This makes it difficult to apply state of the art AR view management techniques, which rely on the availability of a detailed three-dimensional representation of the environment that is precisely registered. For the same reason, techniques from label placement in Virtual Reality cannot directly resolve the problem, since no virtual representation of the real environment is available.

ARBs use iconic or textual labels to annotate POIs. Since no other information is available, the placement of a label is a projection of the POI to the screen, which is only determined by the POI's GPS position and the current tracking information. This will often result in cluttered scenes and labels occluding important real-world information, as the content of the video background underneath the augmentation is not considered. In consequence, the visual quality in ARBs often suffers from a poor placement of labels and violates common design rules (see Figure 1, left).

In this work, we present an image-driven view management ap-

proach improving the visual quality of annotated AR environments (Figure 1, right). Our approach targets especially AR systems that lack scene knowledge - such as the current generation of ARBs. Rather than relying on explicit scene knowledge, our approach analyzes the video image to determine the placement of the augmented labels. The video information is further used to adjust the appearance of the augmented labels. Our labels will be (1) placed in such a way that interference with important real world information will be reduced and (2) rendered so that each of the labels is easily related to its corresponding POI as well as readable over the background. To account for the interactive nature of AR, we moreover present (3) an approach to maintain frame coherent labels. We derive our implementation from a requirement analysis of AR view management systems, and therefore also describe general design guidelines for view management in AR.

2 RELATED WORK

Label placement is a well studied problem. A large number of techniques have been proposed for 2D or 3D computer graphics applications. Label representation techniques has also some relevance in AR and are covered in the following literature review, which is organized as follows: Firstly, we discuss view management techniques that rely on geometry-based layout techniques. Secondly, we present view management techniques targeting image-based layout creation. Finally, we discuss adaptive representation techniques of labels.

2.1 Geometric-based layout

Various techniques have been proposed for the layout of point features in geographic information systems and cartography. Christensen et al. [4] demonstrated that point feature label placement is an NP-hard problem and proposed simulated annealing and gradient descent as solutions. Mote et al. [26] introduced a real-time method based on geometric considerations. Wu et al. [40] proposed a genetic technique combined with image analysis of a vector representation of a map, whilst Ebner et al. [7] used a force-based approach. In the area of pictorial annotations, Luan et al. [20] presented a method for annotating gigapixel images.

Similar approaches have been proposed for 3D scenes. For instance, Hartmann et al. [13] showed a force-based label approach for annotating 3D models. Vollick et al. [37] proposed a learning based method, based on a minimization approach combined with simulated annealing. Stein and Decoret [31] improved a standard greedy algorithm technique by decomposing the 2D scenes with Voronoi diagrams and provided an implementation on the GPU. Molik et al. [5] also used a greedy algorithm, combined with fuzzy logic.

Maass and Döllner have proposed different solutions for annotations of virtual landscapes and geospatial 3D scenes [22]. They described a method considering the depth of the labels and the available screen space. In another work [23], they considered the geometric shape of the annotated objects and automatic adaptation of position and orientation based on the current camera viewpoint. In a following work [21], they discussed the difference between 2D labeling and 3D labeling and the importance of developing a dedicated method for the latter.

Labeling and label placement have also been investigated in AR, as they are highly relevant in a large number of AR applications [39]. Bell et al. [3] presented one of the earliest works in AR and introduced a geometry-based technique considering projection of static or dynamic objects. They were also the first to emphasize the importance of view management in AR, which is also the motivation of this paper. Azuma et al. [2] evaluated different algorithms for label placement in AR (greedy, gradient descent, simulated annealing, their new cluster method), discussing the technical performance and readability of the different methods. They point out the large number of factors influencing the choice of an algorithm, such as application or task specific conditions.

Other efforts have been made for improving label placement in AR. Shibata et al. [30] propose a framework for rule-based layout, whilst Tenmoku et al. [34] introduce two methods resolving occlusion (highlighting and de-emphasizing). Zhang et al. [41] investigated augmented videos, and used the projection of 3D buildings in the image to select the best placement area.

Peterson et al. [28, 27] studied an alternative approach based on the usage of stereoscopy to improve readability of labels and cluttering of standard techniques. In a comparative study, they show that their method proved to be as effective as planar (greedy and cluster) algorithms or height separation. Makita et al. [25] explored augmentation of dynamic artifacts (such as people) and described a greedy algorithm for identifying the best position of the labels.

A limitation of all these techniques is the lack of consideration of the view of the real world, which is assumed to have a uniform background. Most approaches rely on the availability of a geometric model (2D or 3D) of the content, while we concentrate on a technique that works without this information.

2.2 Image-based layout

There are few works that have explored image-based layout creation for AR. Leykin and Tuceryan [18] described a learning-based approach to estimate good regions to place labels in AR (in terms of readability), but do not describe an algorithm to chose the ideal position of the label.

Rosten et al. [29] implemented a real-time label placement for handheld devices using importance maps built from FAST features of the video image. Their technique can also add constraints to position labels in specific regions and addresses placement of labels under dynamic conditions. However, the technique was only demonstrated with few labels, only for indoor scenes, and does not consider the visual appearance of labels.

Tanaka et al. [33] consider occlusions of the video image when placing annotations and propose the average of RGB color, saturation and luminance of a region in a video image. The coefficients for parameterizing their solutions are determined through learning. In comparison, our technique is pixel-based rather than region-based and also considers temporal coherence for placing annotations (original positions and dynamic aspects).

2.3 Adaptive representation

Presentation of virtual content in AR is a complex problem, because the virtual information must be integrated into the video image. Julier et al. [16] proposed information filtering to reduce information overload and visual clutter. Livingston et al. [19] compare different drawing styles for resolving depth cues and occlusion issues between real and virtual content.

Uratani et al. [36] investigated the representation of labels in AR. They identified parameters that can be modulated to improve their readability, such as label size, color or transparency. However, their technique does not consider naturally salient backgrounds (the results were shown on a paper with printed markers), and they do not adapt the representation. More recently, Kim et al. [17] discussed modulations of label representations, and present some pilot studies in this context. Similar to Urani et al. [36], this information is not used in an automatic system.

Jankowski et al. [15] present a comparative study of annotations over synthetic images or video sequences (no tracking or dynamic labels). The results of their study show that the usage of billboarded labels with a rectangular background improves readability and is also preferred by the users.

Gabbard et al. [9, 8, 10] proposed the concept of a visually active AR interface, which inspired the approach proposed in this paper. They presented different studies on label representation for

AR. The authors solely focused on the representation of the labels (not the layout) and propose different user studies of the readability of labels, using a variety of scenarios as well as several drawing styles (no background, billboard, shadow or outlines). They also present two active techniques for representing the labels: one that uses the maximum HSV component and one that uses the maximum brightness contrast. We also looked at active techniques, considering background modulation, and propose alternative methods that we will describe in Section 5.2.

Unlike previous work, the approach presented in this paper addresses both the layout and the representation of the label in a single system. We also differ from previous approaches by considering a pixel-based method for label placement (compared to grid, object or region-based methods), and introduce a design framework that considers an extensive list of aspects for placing labels based on the video image content.

3 Terminology

We briefly describe here the terminology we are using in this paper. We identify a *POI* as a (geographical) point feature, which is associated with a pictorial or textual label, representing the information at a specific geo-location. A set of POIs can be aggregated into *channels*, similar to [24]. Contrary to some of the literature, we do not differentiate *annotation* from *label*, where a label is generally linked to data (domain content) and an annotation is any textual overlay on a picture. View management of POIs includes two main steps: the layout (point-based placement or labeling) and the representation (visual style) of the POIs. In this paper, we are not considering internal labeling, but only external labeling, following the argument that important objects should not be occluded by their own labels.

4 Design Considerations

In this section, we formulate the requirements for view management intended for AR in general and ARBs in particular. For this purpose, we combine some of the design rules introduced from previous work in AR with graphics design techniques for annotating printed and digital media, as well as general HCI literature.

We considered the general criteria used for dynamic computer generated label layout such as proposed by Hartmann et al. [14]: readability, unambiguity, aesthetics and frame-coherence. Here we concentrate on the applicability and the meaning of these rules for AR.

The emerging design rules can be classified into four main categories: *Standard graphics considerations* define the rules, which are generally used for traditional computer graphics labels. *Real world considerations* are responsible for the rules related to labeling over the view of the real world. *Context considerations* guide the rules related to the type of the task, physical and social context of the labels. *Authoring considerations* dictate the rules related to providing control to the designer of the view manager. In the remainder of this section, we present details on these considerations.

4.1 Considerations in VR

We condensed several basic rules and criteria previously established in the literature [5, 31, 37, 14]. According to these works, good properties of label layout with a synthetic scene are:

- Avoid labels overlapping the domain content.
- Avoid labels (and/or their background) overlapping each other.
- Avoid crossing leader lines.
- Provide enough space for labels, their background and their leader lines for readability.
- Minimize the leader line length. Labels stay close to their anchor.
- Maintain the direction of leader lines according to the selected layout style – circular or flushed.
- Consider the distance between the camera and labels (e.g., label size and leader line orientation) [22].
- Avoid jumping labels (temporal incoherence) during camera motion.
- Avoid jumping labels (temporal incoherence) during object movement.

Moreover, with regard to 3D external labeling, labels should provide depth cues to make it easy to associate them to near or far domain content (e.g., 3D objects).

4.2 Considerations in AR

Real world considerations address the design of label placement and representation based on information extracted from the current video image. We extend the work by Rosten et al. [29, 39] and formulate the following labeling rules (see also Figure 2):

Figure 2: Illustration of three of our real world considerations (from top to bottom), bad (Left) and good (Right) examples of: (a) avoiding overlap on salient areas, (b) avoiding overlap on edges, (c) improving contrast between video image and labels.

- Sensors and tracking: Label layout and representation should adapt to sensor and tracking characteristics (e.g., artifacts, errors [6]).
- Video image/pictorial content: Provide visual coherence between the label representation and the visual characteristics of the video image (such as luminance and chrominance).
- Video image/pictorial content: Avoid overlap in salient areas of the video image and prioritize uniform areas (Figure 2a).
- Video image/geometrical content: Avoid overlap with geometrical structures such as edges (Figure 2b).
- Video image/pictorial content: Support good readability of the POIs by assuring contrast between the label's components and the video image [15] (Figure 2c).
- Video image/geometrical content: Align to geometrical structures either in 2D (e.g., wall, roof) or 3D [35].

- Video image/geometrical content: Consider the principal direction of the content of the image for placing the labels (e.g., horizontal for buildings, radial for monuments).
- Video image/dynamic content: Consider the dynamic content of the video image and avoid jumping labels. Also consider placing labels in regions with continuous movement (in terms of direction and magnitude, such as constant flow of passing cars).
- Video image/dynamic content: Consider proximity of the label from an annotated dynamic real world object (e.g., annotating a moving person or a passing car).
- Camera: Consider non-linear and noisy user movement (e.g., motion of the device, user moving) and avoid jumping labels.

4.3 Context considerations

The representation of labels is highly dependent on the type of task done by the user, ranging from searching and browsing content to pedestrian navigation or following step-by-step maintenance instructions. The view management must therefore consider context in terms of tasks, basic actions (selection, physical motion) and user preferences. An important design consideration is to make the placement and representation of POIs dependent on the user context.

4.4 Authoring considerations

In modern multimedia applications such as web pages, it is common practice to separate style from content. The same consideration should be applied to the layout and visual appearance of labels in an ARB. For example, label colors can be adapted automatically as suggested in [9]. In general, a designer should be able to specify high-level style decisions, and let the system attempt to conform to the desired styling at runtime as good as possible, according to a number of styling criteria. Our design rules are therefore:

- Assure configurability by the designer and control over automatic or manual selection of styling.
- Support high level definition of layout and representation based on visual style.
- Support a simple definition of this visual style using a markup language or authoring tool.

5 Adaptive Technique

In contrast to previous work, we propose a hybrid technique, which combines a layout algorithm for placing the labels (section 5.1) with an adaptive rendering technique for representing the labels (section 5.2). Temporal coherence is treated separately (section 5.3). We finally address the design issues with a structural description of visual styles for ARB view management (section 5.4).

5.1 Image-based layout

Since no scene knowledge is available, our layout technique only uses information from the video image to control the position of the labels. The goal is to avoid occluding important content of an image. In this paper, we especially looked at two of the main real world criteria: salient features and edges.

Image analysis. For the identification of important areas in the original image, we combine a visual saliency algorithm with a simple edge analysis (Canny edge detector) and apply it to the current video image. The saliency computation produces an intensity image (saliency map), where the grey level represents the importance of the information in the image. The edge map complements the saliency map, because for our purposes, edges do not show up prominently enough in the saliency map (they only contribute to high frequency saliency). Taken together, the salient information and the edges information encode the pixel positions where labels should not be placed.

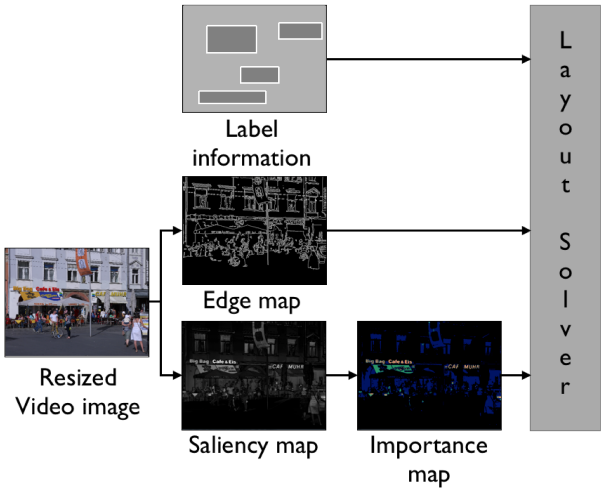

Figure 3: Image analysis for our layout algorithm: the video image is resized, saliency map and edges map are computed. A threshold is applied to the saliency image and used to classify 3 levels of importance.

We use the saliency algorithm proposed by Achanta et al. [1], which is offer the best compromise for our purpose: the algorithm is fast, the computed saliency map has the same size as the processed image, and it eliminates regular patterns in the image as being salient elements. For performance reasons, we resize the input video image, run the saliency algorithm and classify the saliency information in three different classes. These levels offering a simpler representation of the saliency information for further processing. The thresholded saliency map and the edge map are then used for the layout solver (Figure 3).

Objective function. To find the optimal layout of the labels, we describe the label placement as an optimization problem and define and minimize an objective function. The objective function O encodes some of the standard graphics and real world considerations as weighted penalty factors:

$$O(L,x) = \sum_{i=1..n} \alpha_i p_i(L,x) \quad (1)$$

where L defines the label, x its screen position, α the weight and p the penalty factor. The different penalties factors are defined below:

- Overlap on the importance map:

  $p_{IMap}(L,x) = \sum_{i=1..sx,j=1..sy} IM(i,j)$

  where sx and sy define the size of the label L, and $IM(i,y)$ is the value of the importance map at the pixel position (i,j).
- Overlap on the edge map:

  $p_{EMap}(L,x) = \sum_{i=1..sx,j=1..sy} EM(i,j)$

  where sx and sy define the size of the label L, and $EM(i,y)$ is the value of the edge map at the pixel position (i,j).
- Leader line length:

  $p_{LDist}(L,x,x_0) = \mid (x,x_0) \mid$

  where x_0 defines the original position of the label L, and (x,x_0) is the vector between x_0 and the label position.

- Leader line orientation:

  $p_{Ori}(L,x,x_0) = \mid \emptyset(x,x_0) - f(layout) \mid$

  where $\emptyset(x,x_0)$ defines the orientation of the leader line and $f(layout)$ the preferred value of the orientation (e.g., $\pi/2$ for vertical or 0 for horizontal alignment).
- Label overlap:

  $p_{Overl}(L,x,x_0) = \sum_{i=1..n} overlap(L,G_i)$

  where we compute the overlapping region between the current label L and the n labels $\{G\}$, which have been already placed. We use a similar parametrization as Vollick et al. [37], where the function $overlap(L,G_i)$ computes the Euclidian distance between the label L and the label G_i, detects overlap between the labels based on their respective sizes, and returns an overlap value.

Additional constraint such as presented in [31] and [37] can be added to our algorithm. Figure 12 (bottom, right) presents an example of the layout technique for a basic outdoor scenario using the presented constraints.

Based on the evaluation of different labeling algorithms presented by Azuma and Furmanski [2], we considered two algorithms for implementing the optimization: a greedy algorithm and a force-based algorithm. The greedy algorithm sequentially optimizes each label and evaluates the objective function for each. The minimal value among the candidate positions is selected as the position of our label. In contrast, the force-based algorithm implements penalty factors as a set of forces, and labels are moved in parallel in this force field. Labels obtain their final position after a certain number of iterations or according to a termination criterion. Simulated annealing was ruled out, as it provides accurate results, but is too costly for the targeted mobile computer platform.

Initial tests showed limited results using the force-based algorithm. We dilated the importance map, and calculated a distance transform image. Then, we computed the gradient to create a repulsive force for our system (labels are pushed away from important regions). We proceeded similarly with the edge map. The other penalty criteria were implemented as procedural functions. Contrary to our initial expectations, we obtained a rather complex force field (dense and isotropic). Issues with weighting the different forces and finding an ideal number of iterations for our test image dataset made it unusable in practice.

We therefore adopted the greedy algorithm (Figure 4). In an initial step, we sort out the labels currently visible from the left to right and sort out the labels in depth, from the closest to the farthest. We iterate for each label, and for different positions in the search space, and minimize the objective function. A configuration of the search space offers flexibility for the layout orientation of the labels: top, bottom, left, right, radial, and combinations of them. The top configuration can be suitable for far POIs in outdoor scenes, whereas a radial configuration can be relevant for annotating close objects (Figure 5).

To handle image motion and dynamic content in the video image, the layout algorithm is executed at low frequency after initially placing all labels. To avoid jumping labels, we locally test for each label if there is any change of the saliency or edges information and avoid recomputation as needed. We also consider to move the label only if the best score provided by the objective function on the current frame is better than the former score. We finally add smooth animation of the labels to avoid abrupt movement. Our approach for handling camera motion (i.e., device rotation or translation) is discussed separately in Section 5.3.

5.2 Adaptive representation

Adaptive rendering has been poorly studied in AR, especially regarding label representation. Our approach differs from Gabbard et al. [9], as we consider labels with background and multiple labels. To approach this problem, we modulate the representation of each of the label's components.

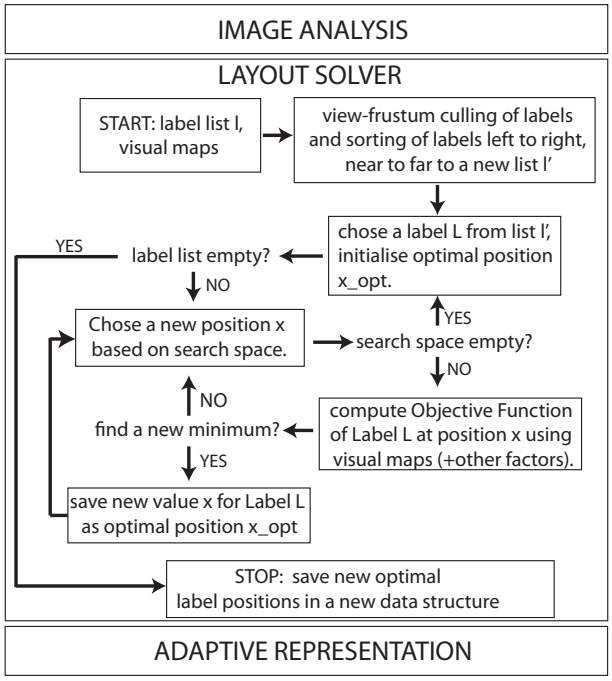

Figure 4: Flow chart of our image-based algorithm.

Leader lines. When moving labels further away from their anchor position, leader lines are required to link them. Users must be able to identify the leader line, which can be hard to discriminate from the video background when the contrast between the color of the line and the surrounding pixels is low.

To address this problem, we apply a similar method as Steinberger et al. [32], but here applied to AR. The main idea is to modulate the color of the edge to make it more salient compared to its vicinity. Increasing this contrast can be be done by modifying the intensity channel in a suitable color space.

Figure 6: Color of the leader lines are adapted to the background: darker in bright area (Left), brighter in dark area (Right).

In our case, we consider the lightness of the line in HLS space. We compute an average of the lightness of the pixels surrounding the leader line and modify the color of the leader line to yield a certain contrast. In our experiments, we have determined a contrast threshold of 20% to be suitable. The contrast modification can be positive (leader line getting brighter) or negative (leader line getting darker), in function of the lightness intensity of the leader line. An

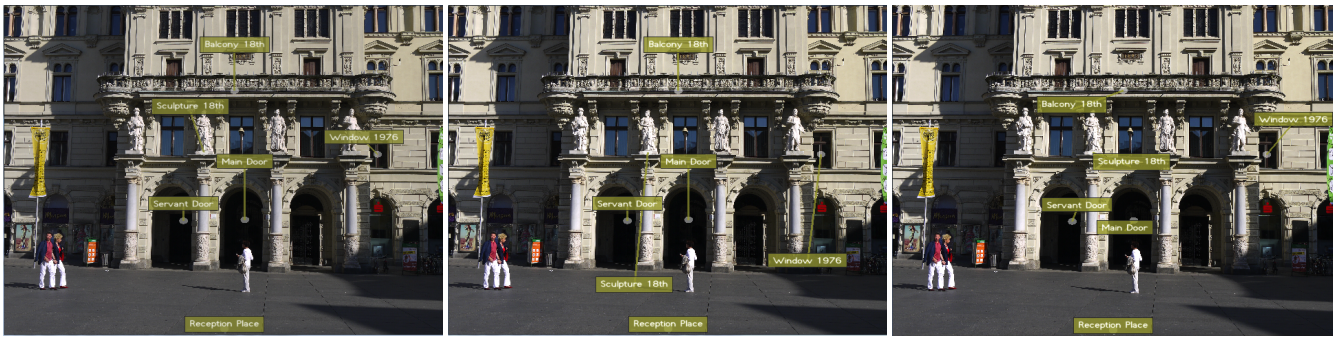

Figure 5: Different search spaces resulting to different type of layout (left to right): top, bottom-top or radial layout.

example is presented in Figure 6.

Anchor. When displacing labels from the POI, anchor points become more prominent in the view of the user. This supports the user in identifying the real position of the POI. However, because we do not have any scene knowledge, we do not know if the POI is in front or behind an object present in the view. This poses a potential depth cue conflict.

To address this issue, we use an ring shaped anchor and use its inner radius to encode the distance. If a POI is close to the user, the ring will be a full disk, if the POI is far from the user, the ring will converge to a circle. We also modulate the opacity by encoding its value in function of the distance to the viewer. Hence, close POIs are fully opaque, while distant POI are mostly transparent. To determine the radius, we rescale the distance of the POIs from the user's viewpoint to a normalized range.

As with the leader lines, the color of the anchor can be modulated using the technique presented above.

Figure 7: Anchor Ring concept and example with POIs at different distance.

Background and text. Current standard representations of information channels in ARBs use a static rendering style and generally emphasize contrast by using negative or positive color schemes for the background color/text color (similar to the findings of Jankowski et al. [15]). However, when the label overlays a dark or bright area of a video image, the readability is impaired. Following the work by Gabbard et al. [10], we investigated active rendering styles for the labels. Our main focus was finding an active style that can support representation modulation of multiple POIs or multiple visible channels at the same time.

Literature on representation of text over regions provides mixed guidance, as it considers unified modulation of both luminance and chroma of a label [11]. We therefore propose a separated technique, which works in HLS color space and allows to adapt lightness or saturation of a label background or of its content.

For the lightness and saturation, we investigated three different styles: global, local or salient-relative. For the global approach, we compute the average lightness over the full image and modulate the lightness of the label background to have a contrast difference above a certain threshold (Figure 8). The local approach considers only the computation of the average lightness in the neighborhood of each label's background, and contrast adjustment is applied separately for each label. The salient-relative technique considers the average lightness of the salient regions, so the labels can be more prominent with respect to the saliency information on the image.

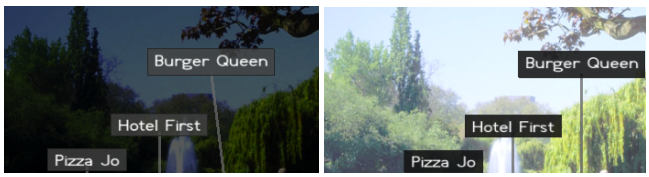

Figure 8: Results of our adaptive representation using global lightness modulation: the grey background of the labels is modulated for a dark or a bright image.

5.3 Context and temporal coherence

To achieve temporal coherence, we minimize label movement caused by jitter introduced by unsteadily holding the device. We also avoid moving labels if there are only small dynamic changes in the scene.

We considered three common motions in ARB: camera motion (large change of rotation/position), hand shaking/jitter motion (small change of rotation/position) and object motion (dynamic content in the video image). We treated camera motion as the primary factor and developed our approach based on the results of a survey we conducted recently on user behavior and adoption of ARBs by the general public [12]. Contrary to popular opinion, end-users do not interact with their ARB during walking: our survey shows that movement patterns are mainly standing+rotation (90%) where multiple large movements (>5m) combined with rotation being largely unused (42%). An ARB is mainly used while intermittently stopping between locations, and consequently physical interaction is constrained to primarily rotational movement.

We build our approach on this finding and use an inertial sensor to determine the yaw magnitude of the current rotational camera motion. We use a state hysteresis technique as presented Figure 9. When users rotate around their axis, a large movement is detected and the system will use the default representation. We only trigger our adaptive layout and rendering if there is no large movement for a certain number n of frames.

If the user holds the device steady to observe the scene, we

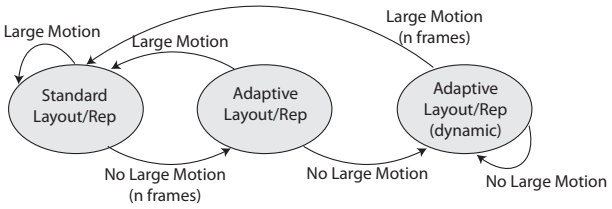

Figure 9: State hysteresis under camera motion, which is estimated by an inertial sensor.

switch to the dynamic state, where we execute our algorithm at a low frequency. In this condition, a label is only moved if the computed best position is relatively distant from the current position. This filtering behavior suppresses small dynamic changes, such people or bikes passing by. Additionally, we use animation to interpolate between the consecutive label positions to avoid abrupt label movement.

5.4 Designer control through style sheets

A designer can control the properties of the different functions of our pipeline by defining and applying visual styles. Inspired by the reasoning of MacIntyre et al. [24] for creating high level description of an AR scene, we allow the representation of the content to be described using a simple markup language. Therefore, we define visual styles that can be applied to separate channels (retrieved with the POI information) or for a combination of channels (defined for a specific configuration of an ARB such as shopping places, cultural heritage, etc).

Similarly to KML, we define an `arlabelstyle` element, which has an `id` identifier. The AR label style can be applied to one channel (by referencing the style, `styleurl`) or to the current view (`generalArLabelstyle`) when multiple channels are activated. An example of label style is presented in Figure 10.

```
<arlabelstyle id="34" name="moutainstyle">
    <context activated="off" />
    <layout activated="on" optimizationType="saliency"
    alignmentType="top" optimizationAlgo="greedy" />
    <representation activated="on"
    backgroundOptimizationType="luminance"
    backgroundOptimizationStyle="global" />
</arlabelstyle>
```

Figure 10: Access for a designer to our view management technique: she can control the type of layout of POIs or the adaptive representation of a channel.

6 Implementation

We implemented our view management on both a desktop and a handheld platform. The desktop platform is a MacBook Pro, 2.2 Ghz Intel Core i7 2.2Ghz, 8GB RAM. We used a Logitech webcam C905 for the video and an Intersense InertiaCube 3 for inertial input. The handheld test platform was a Motion J3500 Tablet PC, Intel Core i7 1.46Ghz, 2GB RAM using the same sensors configuration.

For the software, our view manager has been developed with OSGART [1] for the AR rendering and OpenCV[2] for image analysis and computer vision. The visual saliency algorithm was kindly made available online[3] by its authors [1]. It was integrated in our framework and optimized for our needs.

[1] http://www.osgart.org
[2] http://opencv.willowgarage.com
[3] http://ivrg.epfl.ch/page-74626-en.html

The software architecture of our view manager was separated in different modules, including a tracker manager (inertial/GPS or other tracking technology), a label manager (retrieving POIs, standard display of the labels), a context manager (detection of user behavior and user preferences) and the core view manager (layout and optimized representation computation).

7 Results

7.1 Visual quality

We evaluated our technique using static images, recorded videos and live video (see the accompanying video). As illustrated in Figure 11(Left), our algorithm reliably avoids placing labels on detailed features or visually prominent elements (people, cars, visual signs, complex facades) and places labels in uniform areas (sky, ground area, grass). For urban facades, labels are generally moved between windows and reliably avoid overlaying shop signs.

The saliency map computation is also efficient for filtering highly repetitive textures. However, our approach is not robust enough for extremely salient scenes, where no optimal layout can be found (Figure 11(Right)).

Figure 11: Extreme scenarios: average saliency (Left) and extremely high saliency (Right). The worst condition has no optimal solution and our technique should not be used in this case.

The edge map computation generates a large number of lines for complex scenes with trees or complex building structures. We therefore use a small weight factor for the edges map in our objective function.

We qualitatively compared several layout techniques: "naïve" layout, height separation, planar separation and our technique for a variety of environments (an example is presented in Figure 12). The "naive" layouting presents a large number of POIs, which are just projected into the current view.

This is the most common method used by currently available ARBs, resulting in overlaps between labels and with the content of the real scene. The only exception is the Junaio ARB, which is based on a height separation method. In their implementation, labels are grouped in function of the distance of the POIs, aligned bottom to up between the different groups and front to back inside a group. They do not consider the border of the screen as the constraint for the placement, forcing the user to rotate the device for exploring the labels and thereby loose focus on the current real scene.

For our comparison, we therefore selected the height and planar separation techniques as implemented and presented by Peterson et al. [27]. Compared to height separation or planar separation, our method is more successful in avoiding visually important areas, but at the cost of redistributing the labels over the image. For the planar separation, no image analysis is performed. Labels are tested in a search space of 36 different angles with 5 different positions going outwards from the initial label position, spaced at 5 pixels each (clamped at the borders of the screen). For the height separation, we do a search from bottom to up, starting from the initial position of the label, positions separated by 5 pixels, until we reach the top of the screen.

Figure 12: Comparison of layout techniques (left to right): naive, height separation, planar separation, our approach with top placement.

We evaluated the effect of different search spaces for the optimization function using different pictures and varying the type of layout. We found that bottom-up layout tends to significantly disperse the labels in the image, while top or bottom layouts cluster the labels more.

We examined our method using differently colored background, varying hue, lightness or saturation for synthetic or real scenes. Figure 8 shows an example of how the background lightness is adapted for an image which has been manually modified (decreasing and increasing lightness). Using a global threshold adjustment results in a small improvement when lightness is in the mid-range. Our approach is different from Gabbard et al. [10] by not focusing on hue modification and considering heterogeneous background with multiple labels.

We tested the dynamic coherence with synthetic scenes (for object motion and shaking motion) and with hand-held recorded video of street roads with vehicles and pedestrian passing by. The technique succeeds in avoiding the movement of labels for fast object motion and small hand shaking. Our approach thus improves over Rosten et al. [29], which moves labels even on small dynamic changes such as people passing by.

7.2 Performance

We tested the performance of our technique on the desktop and the handheld platform mentioned in section 6. We captured images at 640x480 pixels and then resized them to 160x120 pixels for processing. This size turned out to be a suitable trade-off between real-time performance and visual quality. We tested the technique by varying the number of labels on both platforms, and we computed the average time for 10 different images (Table 1). The saliency computation is the most costly step, and even for 30 labels (which is an extremely cluttered scenario), the greedy algorithm has still a negligible cost.

Operations	PC, 10L	PC, 30L	H, 10L	H, 30L
Resize	3.04	2.91	9.96	9.97
Saliency	12.59	12.6	22.64	21.46
Edges	0.65	0.66	0.93	0.91
Thresholding	0.02	0.03	0.04	0.07
Layout	0.54	3.01	0.77	3.42
Representation	1.27	1.53	1.68	2.08
Total (ms)	18.11	20.73	35.86	37.91

Table 1: Average time performance (ms) for two platforms (PC, handheld H) and two visual clutter configurations (10 and 30 Labels).

7.3 User Feedback

We conducted a preliminary user study for gathering first user feedback of our image-based layout and adaptive representation techniques. We focused on comparing different layouts (including our technique) and assessing the benefit of the adaptive rendering.

Study design. Our main interest was on initial validity of our technique for the following criteria: scene understanding (including real/virtual), readability and aesthetics (subjective satisfaction of end-users). For repeatability of the comparison of these aspects, we did not use a live AR setup with changing scenes, but we chose static scenarios and made a qualitative evaluation.

The evaluation included two separated factors: the layout and the adaptive representation. We had 4 conditions for the layout: naive technique (LN), height separation (LH), planar separation (LP) and our technique using saliency (LS). For the adaptive representation, our layout technique was enabled and we had 5 conditions: no adaptation (RN), global background adaptation (RGB), local background adaptation (RLB), global background adaptation + leader lines adaptation (RE), global background adaptation + leader lines + anchor (RA). We had 6 repetitions for both parts, using 6 different representative scenarios: urban/outdoor scene, visible/occluded objects, low saliency/high saliency.

We presented the different pictures (scenarios) to the participants for both the layout and the adaptive rendering (tested on an Apple iPad device). Participants were ask to explore the different pictures and reply to a list of questions. We asked the user to chose the best pictures for overall satisfaction, optimal scene understanding and best readability. A semi-structured interview was conducted at the end. We counterbalanced the order of the presentation of the layout and the adaptive representation conditions to the participants, and we randomized the scenarios.

Results. We had 7 participants for the user study (4 female, 3 male, age range 21-29, students), for a total of 42 trials. Five had no knowledge of AR, one had used an ARB before and one was knowledgeable in AR.

Regarding the layout, the participants answered in 45,2% of the cases that they favored our image-driven layout technique as their overall preference (height separation 30,1%, planar separation 14,2%, naive technique 9,5%). Most of the cases, where our technique was not their preferred choice, corresponded to scenarios in which no (or hardly any) labels overlap in any layout. The height separation technique was favored strongly in one scenario where the layout of the labels was identical to the geometry within the picture. They also stated that other conditions can provide a more balanced and symmetrical layout in some situations. This was especially the case in one scenario (an indoor scenario showing a dashboard) that was highly symmetrical.

The participants mentioned the additional clutter introduced by the anchor lines as a main reason for not selecting our technique. Penalizing anchor line length stronger in the optimization may be a suitable countermeasure. Additionally, when asked about a better solution for the layout (not proposed in any of our conditions), the users struggled to propose a solution, understanding the difficult trade-offs involved.

We also asked which technique preserves most image information, and the participants favored our approach. In 76.2% of the cases, they preferred our image-driven layout technique, while LH (16.6%), LN (4.8%) and LP (2.4%) were rarely selected. In cases where our technique was not selected, the anchor lines sometimes hid image information, while the competing layout techniques had no or only short anchor lines.

Regarding the adaptive representation, reaching a decision was more demanding for the participants. We especially noted that some participants were still undecided after few minutes. In two cases (4.8%), the users could not make a decision. They mostly favored RE (44.4%), followed by RN (25%) and RGB (22.2%). The other techniques were only rarely selected (2.7% for RA and 0% for RLB). When asked about the best readability for the representation, the users mostly chose RE (38.9%) followed by RGB (27.8%). All other techniques were again only rarely selected.

Overall the users commented that they did not like the adaptation of the anchors. Highlighting the anchors and encoding information on it was perceived as occluding the POI. Local background adaptation was usually not perceived as increasing scene understanding and coherence between the labels. The users preferred background adaptation decreasing lightness over adaptation increasing lightness.

8 Discussion and Future Work

We obtained overall positive feedback from the users, especially regarding our layout technique. Our approach seems to offer better result than standard techniques. As our initial user assessment was ecological, our technique provide on average good results, but can fail in certain scenarios due to the structure and semantics in the image. As expected, local background adaptation was not preferred, as end-users value coherence of the presented information.

The cost of moving labels, and thus introducing more visible leader lines or anchors should be balanced carefully concerning the length of the leader lines, the size of anchors and the amount of saliency in the image. Similar to maps, leader lines' length should be minimized. Unlike maps, the background video of an ARB contains areas of low significance, which can be exploited for adding labels and leader lines. We hope to further explore how we can define penalty criteria to judge how much clutter we introduce with our technique. As our work was not focused on safety critical applications, where readability is a major factor (such as in the work of Gabbard et al. [10]), finding a balance between readability and aesthetics remains challenging.

We also wish to investigate further how to combine the layout and the adaptive representation through a global optimization process. As our technique is based on image processing methods, the camera's photometric response and auto-adjustment (such as auto-contrast) is relatively important for the robustness of our approach. We also did not focus on how our approach can benefit from using more advanced tracking technology [38]. We will explore that in future work.

Our view manager has been designed for handling a limited amount of channels and POIs simultaneously visible on the screen. For a large number of labels (above 30), the objective function generally fails to deliver an optimal configuration and the screen gets highly cluttered. We will look how we can combine our current approach with clustering techniques that can dramatically reduce the number of visible objects on the screen.

Finally, as the current evaluation provides only initial assessment of the view manager, further user evaluations should be considered, especially regarding the validation of the dynamics of an ARB.

9 Conclusion

We presented new view management techniques for ARBs. We introduced a new design framework for the design of augmented reality labelings that can be used for future development of ARBs or similar applications. We presented a first prototype of a novel view manager for POIs using both an image-based layout and an adaptive representation. Our approach integrates several new techniques that can be easily deployed in future generations of ARBs.

As future work, we want to explore further real-time implementation of adaptive rendering and especially techniques responsive to the user context and user behavior. Another interest lies in the robustness of this approach for other type of augmented content (such as flyers or magazines).

Acknowledgements

The authors wish to thank Stefanie Zollmann for her initial participation on the project and technical help with some computer vision techniques. This work was supported by the Christian Doppler Laboratory for Handheld Augmented Reality.

References

[1] R. Achanta and S. Susstrunk. Saliency detection using maximum symmetric surround. In *International Conference on Image Processing (ICIP), Hong Kong, September 2010.*, 2010.

[2] R. Azuma and C. Furmanski. Evaluating label placement for augmented reality view management. In *Proceedings of the 2nd*

IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '03, pages 66–, 2003.

[3] B. Bell, S. Feiner, and T. Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 101–110, 2001.

[4] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Trans. Graph.*, 14(3):203–232, July 1995.

[5] L. Cmolík and J. Bittner. Layout-aware optimization for interactive labeling of 3d models. *Comput. Graph.*, 34(4):378–387, Aug. 2010.

[6] E. M. Coelho, B. MacIntyre, and S. J. Julier. Osgar: A scene graph with uncertain transformations. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, 0:6–15, 2004.

[7] D. Ebner, G. W. Klau, and R. Weiskircher. Force-based label number maximation. Technical report, 2003.

[8] J. L. Gabbard, J. E. S. II, and D. Hix. The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor augmented reality. *Presence: Teleoper. Virtual Environ.*, 15(1):16–32, Feb. 2006.

[9] J. L. Gabbard, J. E. S. II, D. Hix, J. Lucas, and D. Gupta. An empirical user-based study of text drawing styles and outdoor background textures for augmented reality. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 11–18, 317, 2005.

[10] J. L. Gabbard, J. E. Swan, D. Hix, S.-J. Kim, and G. Fitch. Active text drawing styles for outdoor augmented reality: A user-based study and design implications. *Virtual Reality Conference, IEEE*, 0:35–42, 2007.

[11] S. Garg, K. Padalkar, and K. Mueller. Magic marker: a color analytics interface for image annotation. In *Proceedings of the 7th international conference on Advances in visual computing - Volume Part I*, ISVC'11, pages 629–640, 2011.

[12] J. Grubert, T. Langlotz, and R. Grasset. Augmented reality browser survey. Technical report, Graz University of Technology, 2011.

[13] K. Hartmann, K. Ali, and T. Strothotte. Floating labels: Applying dynamic potential fields for label layout. In *In 4th International Symposium on Smart Graphics*, SG'04, pages 101–113, 2004.

[14] K. Hartmann, T. Götzelmann, K. Ali, and T. Strothotte. Metrics for functional and aesthetic label layouts. In *Proceedings of the 5th international conference on Smart Graphics*, SG'05, pages 115–126, 2005.

[15] J. Jankowski, K. Samp, I. Irzynska, M. Jozwowicz, and S. Decker. Integrating text with video and 3d graphics: The effects of text drawing styles on text readability. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1321–1330, 2010.

[16] S. J. Julier, Y. Baillot, D. Brown, and M. Lanzagorta. Information filtering for mobile augmented reality. *IEEE Comput. Graph. Appl.*, 22(5):12–15, Sept. 2002.

[17] S. K. Kim, S. H. Moon, J. Park, and S. Y. Han. Efficient annotation visualization using distinctive features. In *Proceedings of the Symposium on Human Interface 2009 on Human Interface and the Management of Information. Information and Interaction. Part II*, pages 295–303, 2009.

[18] A. Leykin and M. Tuceryan. Automatic determination of text readability over textured backgrounds for augmented reality systems. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, pages 224–230, 2004.

[19] M. A. Livingston, J. E. S. II, J. L. Gabbard, T. Höllerer, D. Hix, S. J. Julier, Y. Baillot, and D. Brown. Resolving multiple occluded layers in augmented reality. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '03, pages 56–, 2003.

[20] Q. Luan, S. M. Drucker, J. Kopf, Y.-Q. Xu, and M. F. Cohen. Annotating gigapixel images. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 33–36, 2008.

[21] S. Maas, M. Jobst, and J. Düllner. Use of depth cues for the annotation of 3d geo-virtual environments. In *23rd International Cartographic Conference*, 2007.

[22] S. Maass and J. Düllner. Efficient view management for dynamic annotation placement in virtual landscapes. In *International Symposium on Smart Graphics*, SG'06, pages 1–12.

[23] S. Maass and J. Düllner. Dynamic annotation of interactive environments using object-integrated billboards. In *14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, WSCG'06, pages 327–334, 2006.

[24] B. MacIntyre, A. Hill, H. Rouzati, M. Gandy, and B. Davidson. The argon ar web browser and standards-based ar application environment. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 65–74, 2011.

[25] K. Makita, M. Kanbara, and N. Yokoya. View management of annotations for wearable augmented reality. In *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, ICME'09, pages 982–985, 2009.

[26] K. Mote. Fast point-feature label placement for dynamic visualizations. *Information Visualization*, 6(4):249–260, Dec. 2007.

[27] S. D. Peterson, M. Axholt, M. Cooper, and S. R. Ellis. Evaluation of alternative label placement techniques in dynamic virtual environments. In *Proceedings of the 10th International Symposium on Smart Graphics*, SG '09, pages 43–55, 2009.

[28] S. D. Peterson, M. Axholt, and S. R. Ellis. Comparing disparity based label segregation in augmented and virtual reality. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, VRST '08, pages 285–286, 2008.

[29] E. Rosten, G. Reitmayr, and T. Drummond. Real-time video annotations for augmented reality. In *Proceedings of the First international conference on Advances in Visual Computing*, ISVC'05, pages 294–302, 2005.

[30] F. Shibata, H. Nakamoto, R. Sasaki, A. Kimura, and H. Tamura. A view management method for mobile mixed reality systems. In *IPT/EGVE*, pages 17–24, 2008.

[31] T. Stein and X. Décoret. Dynamic label placement for improved interactive exploration. In *Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, NPAR '08, pages 15–21, 2008.

[32] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-preserving visual links. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2249–2258, October 2011.

[33] K. Tanaka, Y. Kishino, M. Miyamae, T. Terada, and S. Nishio. An information layout method for an optical see-through head mounted display focusing on the viewability. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 139–142, 2008.

[34] R. Tenmoku, M. Kanbara, and N. Yokoya. Annotating user-viewed objects for wearable ar systems. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, pages 192–193, 2005.

[35] V. Thanedar and T. Höllerer. Semi-automated placement of annotations on videos. Technical Report 2004-11, UC, Santa Barbara, 2004.

[36] K. Uratani and H. Takemura. A study of depth visualization techniques for virtual annotations in augmented reality. In *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, VR '05, pages 295–296, 2005.

[37] I. Vollick, D. Vogel, M. Agrawala, and A. Hertzmann. Specifying label layout style by example. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 221–230, 2007.

[38] D. Wagner, A. Mulloni, T. Langlotz, and D. Schmalstieg. Real-time panoramic mapping and tracking on mobile phones. In *IEEE Virtual Reality Conference*, VR'2010, pages 211–218. IEEE, Mar. 2010.

[39] J. Wither, S. DiVerdi, and T. Höllerer. Annotation in outdoor augmented reality. *Comput. Graph.*, 33(6):679–689, Dec. 2009.

[40] H.-Y. Wu, S. Takahashi, C.-C. Lin, and H.-C. Yen. A zone-based approach for placing annotation labels on metro maps. In *Proceedings of the 11th international conference on Smart graphics*, SG'11, pages 91–102, 2011.

[41] B. Zhang, Q. Li, H. Chao, B. Chen, E. Ofek, and Y.-Q. Xu. Annotating and navigating tourist videos. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 260–269, 2010.