3D Wikipedia: Using online text to automatically label and navigate reconstructed geometry

Bryan C. Russell[1] Ricardo Martin-Brualla[2] Daniel J. Butler[2] Steven M. Seitz[2] Luke Zettlemoyer[2]
[1]Intel Labs [2]University of Washington

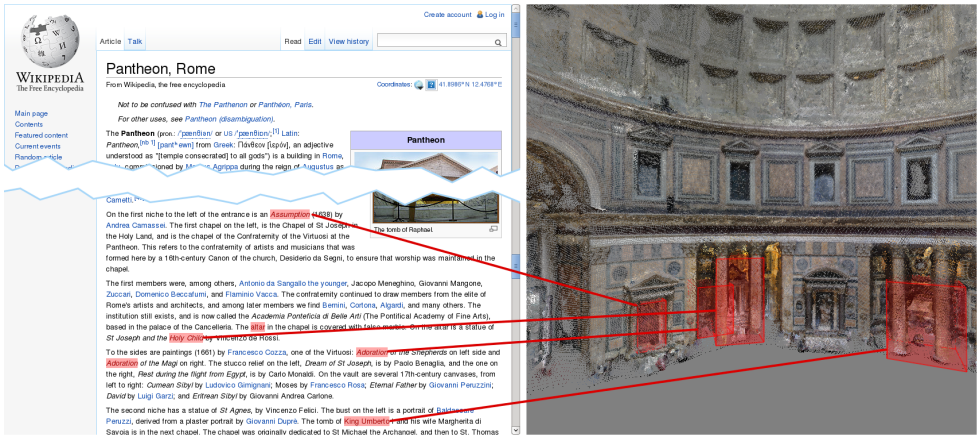


Figure 1: *Given a reference text describing a specific site, for example the Wikipedia article above for the Pantheon, we* ***automatically*** *create a labeled 3D reconstruction, with objects in the model linked to where they are mentioned in the text. The user interface enables coordinated browsing of the text with the visualization (see video).*

Abstract

We introduce an approach for analyzing Wikipedia and other text, together with online photos, to produce *annotated* 3D models of famous tourist sites. The approach is completely automated, and leverages online text and photo co-occurrences via Google Image Search. It enables a number of new interactions, which we demonstrate in a new 3D visualization tool. Text can be selected to move the camera to the corresponding objects, 3D bounding boxes provide anchors back to the text describing them, and the overall narrative of the text provides a temporal guide for automatically flying through the scene to visualize the world as you read about it. We show compelling results on several major tourist sites.

CR Categories: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities I.2.7 [Artificial Intelligence]: Natural Language Processing—Text analysis I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—Modeling and recovery of physical attributes

Keywords: image-based modeling and rendering, Wikipedia, natural language processing, 3D visualization

Links: DL PDF

ACM Reference Format
Russell, B., Martin-Brualla, R., Butler, D., Seitz, S., Zettlemoyer, L. 2013. 3D Wikipedia: Using online text to automatically label and navigate reconstructed geometry. ACM Trans. Graph. 32, 6, Article 193 (November 2013), 10 pages. DOI = 10.1145/2508363.2508425 http://doi.acm.org/10.1145/2508363.2508425.

0730-0301/13/11-ART193 $15.00.
DOI: http://dx.doi.org/10.1145/2508363.2508425

1 Introduction

Tourists have long relied on guidebooks and other reference texts to learn about and navigate sites of interest. While guidebooks are packed with interesting historical facts and descriptions of site-specific objects and spaces, it can be difficult to fully *visualize* the scenes they present. The primary cues come from images provided with the text, but coverage is sparse and it can be difficult to understand the spatial relationships between each image viewpoint. For example, the Berlitz and Lonely Planet guides [Berlitz International 2003; Garwood and Hole 2012] for Rome each contain just a single photo of the Pantheon, and have a similar lack of photographic coverage of other sites. Even online sites such as Wikipedia, which do not have space restrictions, have similarly sparse and disconnected visual coverage.

Instead of relying exclusively on static images embedded in text, suppose you could create an interactive, photorealistic visualization, where, for example, a Wikipedia page is shown next to a detailed 3D model of the described site. When you select an object (e.g., “Raphaels tomb”) in the text, it flies you to the corresponding location in the scene via a smooth, photorealistic transition. Similarly, when you click on an object in the visualization, it highlights the corresponding descriptive text on the Wikipedia page. Our goal is to create such a visualization **completely automatically** by analyzing the Wikipedia page itself, together with many photos of the site available online (Figure 1).

Automatically creating such a visualization presents a formidable challenge. The text and photos, in isolation, provide only very indirect cues about the structure of the scene. Although we can easily gather text describing the world, automatically extracting the names of objects (e.g.,“Raphaels tomb” or “Coronation of the Virgin”) is not trivial. For example, we know a noun phrase often describes an entity, which could be an object in the scene. However, it could also name the artist that created the object, or some other unrelated concept. Given the correct names, even more challenging is deter-

Figure 2: *The top Google image search results for two objects inside the Pantheon and one distractor string. The reliability of the search results varies. Top row: all returned search results depict the entire or part of The Annunciation. Middle row: Only the second returned search result is correct. Bottom row: An incorrect object description with several images that do depict the Pantheon.*

mining the precise *3D location* of each described object, since most textual descriptions within any given reference text are not accompanied by pictures or other explicit visual cues.

The key to our approach is to mine text and photo co-occurences across all of the Internet. For example, a photo anywhere on the Internet with the caption "Annunciation, Pantheon" signals that it may depict the named fresco. Indeed, a Google image search for "Annunciation, Pantheon" yields perfectly cropped images of the desired object (Figure 2, top). Given a Pantheon reconstruction, these images can be matched directly to the model to label the corresponding regions in 3D. Although this approach allows us to find 3D object locations, our challenge of finding object names in text remains. Our solution is to do a brute-force extraction of *every* noun phrase in the text, execute a Google search for that phrase (with ", Pantheon" added at the end), and select only the phrases with images that align with the model. Of course, this simple strategy does not completely solve the problem; image captions and web page co-occurences are notoriously noisy. Searching for correctly named objects can produce multiple matching images (Figure 2, middle) and phrases that do not describe actual objects can produce spurious matches (Figure 2, bottom). Hence, we treat the image results as a noisy signal to be integrated with other constraints in a joint, learned model for filtering out spurious phrase, image pairs. This approach can be considered as a form of *query expansion* [Chum et al. 2007; Buckley 1995; Salton and Buckley 1999] where we issue several queries on pieces of the text and then verify the results.

Our reconstruction and visualization approach is inspired by Photo Tourism [Snavely et al. 2006], and we employ similar techniques to generate 3D models from Flickr photos and to render transitions to photos within those models [Wu et al. 2011; Wu a; Wu b]. Our innovation is not in the rendering per se, but in our ability to automatically transform descriptive texts such as Wikipedia pages into interactive 3D visual experiences, where the text links to corresponding points in a reconstructed 3D model. We show compelling results for several major tourist sites. While no automated method is perfect, we are able to reliably extract many of the objects in each scene, with relatively few errors (we provide a detailed analysis of precision and recall).

2 Related work

Our labeling problem lies at the interface between natural language processing and 3D computer vision; a very fertile area with little prior research. An exception is Simon et al.'s work [Simon and Seitz 2008] on segmenting and labeling 3D point clouds by analyzing SIFT feature co-occurence in tagged Flickr photos. Their approach works by associating commonly occurring image text tags with the model points contained in the associated images. However, Flickr tags are notoriously noisy and far less informative compared to Wikipedia and other authoritative guides. Their approach cannot be applied to Wikipedia, as it requires tagged photos as input.

In the 2D domain, there is a significant literature on correlating regions in images/video to captioned text or keywords, e.g. [Barnard et al. 2003; Laptev et al. 2008; Cour et al. 2011], and on generating sentences or captions for specific images [Farhadi et al. 2010; Berg et al. 2012; Mitchell et al. 2012]. These approaches reason about a relatively small set of object classes (e.g. *car*, *boat*) via trained object detectors, whereas we reason about object instances (e.g. the *Annunciation*). Furthermore, note that [Berg et al. 2012] require captioned photographs during the training of their model. Our use of 3D reconstructions allows us to avoid many of the object detection challenges these approaches face.

Our work builds on recent breakthroughs on reconstructing 3D models of tourist sites from Internet photo collections. These methods are based on structure-from-motion [Snavely et al. 2008; Agarwal et al. 2011; Raguram et al. 2011] and multi-view stereo [Furukawa and Ponce 2010; Furukawa et al. 2010; Goesele et al. 2007]. The advent of commodity depth sensors like Kinect has also inspired work in object category recognition in RGB-D and range-scan data [Ren et al. 2012; Silberman et al. 2012; Ladický et al. 2012]. This work is complementary to our effort; we focus on labeling instances.

There is a long history in computer vision on the problem of recognizing images of specific objects or places (instances). Especially relevant is recent work on large-scale image retrieval [Sivic and Zisserman 2003; Chum et al. 2007; Philbin et al. 2008] that operates by matching local features computed at interest points between an input image and a database of labeled images [Lowe 2004]. Also relevant is work that reasons about GPS-tagged images [Crandall et al. 2009; Hays and Efros 2008]. All of these techniques require a database of labeled objects as reference. In contrast, our focus is to *create* such a database from joint analysis of text and images.

3 System Overview

In this paper we present a fully automatic system that generates interactive visualizations that link authoritative text sources with photorealistic 3D models. The system requires two types of inputs: one or more reference text sources, such as Wikipedia, and a unique name for the site to reconstruct, such as the Pantheon in Rome.

Figure 3 presents an overview of the complete approach. There are two parallel streams of processing. The system downloads a set of images from Flickr by querying for the site name and then automatically reconstructs a 3D model using the freely available VisualSFM package [Wu b], followed by PMVS [Furukawa and Ponce 2010] to generate a dense 3D point cloud. It also does a query expansion analysis of the text, involving image search and registration for each possible noun phrase as described in Section 4, to find text that names objects and link it to the reconstructed 3D geometry. Using these object correspondences, our system creates interactive visualizations, as described in Section 5, that emphasize the discovered correspondences, providing innovative navigation experiences for the text and the 3D model.

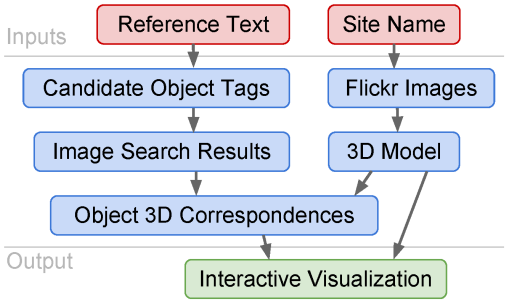


Figure 3: *System overview.*

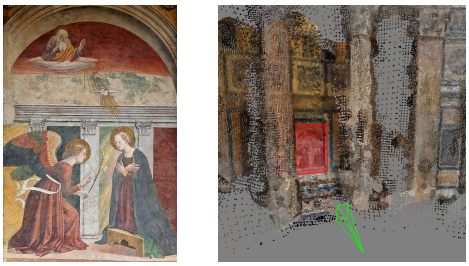


Figure 4: *Left: image returned from Google image search. Right: section of the 3D model, with a bounding box around the matched 3D and the camera frustrum.*

4 Automatic labeling of 3D models from text

In this section, we describe our algorithm for obtaining correspondences between regions on a 3D model to an object tag description in a reference text. Our algorithm consists of three steps: we generate an overcomplete list of candidate object hypothesis from the text; then we obtain their likely location on the 3D model via query expansion; finally we filter the large number of false positive detections by training a classifier over features gleaned from the text and the output of query expansion.

4.1 Obtaining object hypotheses from text

For each site, we seek to automatically obtain a list of candidate descriptive phrases. Our texts come from two sources that are freely available online: articles from Wikipedia, and text from other, site specific, third-party web pages. These text sources offer rich descriptions of the site's contents and their spatial layout, along with their history, architectural features, and cultural references.

We use the syntactic structure of the language to define the set of possible descriptive phrases, primarily leveraging the fact that noun phrases can name physical objects in English. To extract noun phrases, we use the Stanford parser [Klein and Manning 2003], which achieves near state-of-the-art performance and is available as public-domain software. We ran the parser with the default parameter settings. To boost recall, we also extract prepositional phrases that are immediately followed by a noun phrase (e.g. *a fresco of the Annunciation*) and merge adjacent noun phrases (e.g. *a canvas by Clement Maioli of St. Lawrence and St. Agnes*). These additional phrases allow us to overcome parsing errors, e.g., when nouns are incorrectly labeled as prepositions. Extracting them boosts recall and provides a large set of candidates that we will later filter with a joint model that incorporates visual cues. Finally, to reduce false positives, we remove phrases containing only a single stop word, as defined by a commonly used stop word list [Sto], or only numerals.

4.2 From labels to regions via query expansion

Given the automatically obtained list of candidate named objects, we wish to generate proposal regions for their 3D location within the site. We leverage the fact that many objects are photographed in isolation, i.e. with the object shown in whole and filling nearly the full field of view. This *photographer's bias* has been previously used to discover and segment objects within 3D models [Simon and Seitz 2008].

For each candidate named object, we search for and download images using Google image search. We construct the query terms by concatenating the extracted noun phrase with the place name (e.g. *central figure Trevi Fountain*). To find candidate regions within the 3D model for the site, we build upon the success of feature matching and geometric verification used to construct 3D models from consumer photographs [Snavely et al. 2008]. We match SIFT key points [Lowe 2004] extracted from the downloaded images to the inlier SIFT key points corresponding to 3D points within the 3D model. Using the putative 2D-3D point correspondences, we recover the camera parameters for the image and inlier 3D points within the 3D model via camera resectioning [Hartley and Zisserman 2004] as shown in Figure 4. We find that matching to the 3D model is beneficial for three reasons: (i) our overall goal is to label the 3D model, (ii) we find that geometric verification and associating multiple SIFT features to each 3D point offers robustness in the matching step, and (iii) matching to local SIFT keypoints and reasoning about the 3D scene offers robustness to occlusion (c.f. the artwork behind the columns in the Pantheon, which are visible in the 3D model but not in the panorama of Figure 6).

As the object of interest is usually depicted within the top set of search results, we perform camera resectioning for the top 6 images returned for each search query. We keep the alignment if camera resectioning finds at least 9 inlier correspondences. We found that verification with at least 9 inlier features almost always yields a correct alignment to the 3D model; using fewer yields incorrect alignments. This requirement discards many images that do not depict the site at all and maintains a high recall for valid images that do depict the site.

4.3 Model for filtering hypotheses

The query expansion procedure returns for each candidate object tag a set of 3D points within the 3D model corresponding to candidate locations of the object. While Internet image search returns many valid images for the candidate object tags, there remains a high number of false positives. The false positives often result from generic images of the site that occur often across the different query terms and get successfully aligned to the 3D model. Moreover, we have the additional difficulty of an over-generated list of candidate objects resulting from the output of the natural language processing parser. In this section, we outline a procedure to separate the good object proposals from the bad ones.

Our goal is to extract good object detections from the hypothesis set of object-region pairs. We start by merging highly-overlapping camera frustra corresponding to the aligned images for a given object tag returned from Google image search during the query expansion step. To merge camera frustra, we first project each frustrum onto a reference image (i.e. panorama or perspective photograph) depicting the site that has been registered to the 3D model. We form a bounding box by taking the maximum x, y extent of the projected frustrum. We then merge two frustra if their relative overlap (i.e.

ratio of intersection area to their union) exceeds 0.5, with the mean of their bounding boxes returned. This results in a set of object tag and detection frustrum pairs for the site, dubbed the *candidate pool*.

Next, we extract features from the candidate pool and the original text. The visual features include: the number of merged frustra for the candidate; the rank number for the top-ranked image search result that aligned to the 3D model; and the total number of frustra across all object tags that highly overlap the candidate frustrum (a high number indicates a generic viewpoint of the site). The text features include: whether a non-spatial preposition (*ago, as, because of, before, despite, during, for, like, of, since, until*) resides in the same sentence as the extracted noun phrase, which often corresponds to historical descriptions; whether the tag corresponds to an author; and whether an author appears in the same sentence as the tag. We encode the presence of an author as a feature since the authorship of an object is often described together in the same sentence as the object. We detect the presence of an author in the text by analyzing prepositional *by* dependencies returned from the Stanford parser [Klein and Manning 2003] and return the second string argument in the dependency as the author.

We train a linear classifier $\mathbf{y} = \mathbf{w}^T\mathbf{x}$ over the features $\mathbf{x}$ and their labels $\mathbf{y} \in \{0, 1\}$ using logistic regression across a set of training sites and test on the remaining sites. We use Matlab's `glmfit` function with logit link function. To construct the training set, we project each frustra in the candidate pool for the site onto the reference image and intersect the projected frustra with objects that have been manually labeled via LabelMe [Russell et al. 2008]. For each labeled object, we keep the object tag/detection frustrum pair that has highest word F-score when comparing the object and labeled tags and having the center of their bounding boxes residing in the other's bouding box. We form the set of positive examples ($\mathbf{y} = 1$) from the tag/frustrum pairs that match to a ground truth label. We form the set of negative examples from tag/frustrum pairs that do not have tag or frustrum overlap with any of the positive training examples. During testing, we perform non-maximum suppression over the detections. We suppress detections if a higher confidence frustrum overlaps a lower confidence one (i.e. their relative overlap exceeds 0.3 and their centers reside in the other's bounding box) or if any of the tag words overlap in the same sentence.

5 Visualization tool for browsing objects in online text

We aim to create immersive visualizations that connect information from authoritative text sources to 3D models constructed from Internet photo collections. The extracted correspondences between object tags in the text and regions of the 3D model provide bidirectional links between the two types of media. In this work we present novel ways to explore and navigate these links, providing spatial and contextual information to the text and meaningful descriptions to the 3D model.

Our visualization has two panes: on the left it displays the website containing the reference text, such as Wikipedia, and on the right a 3D visualization of the landmark, that uses automatically generated 3D bounding boxes to highlight discovered objects. We augment the functionality of the website to enable text-to-3D navigation, 3D-to-text navigation, and automatic tours of the landmarks. Figure 5 shows screen captures of our visualizations, but we refer the reader to the accompanying video to best experience the system. In the following subsections, we describe the different navigation modes, as well as the implementation details of the visualization.

5.1 Text-to-3D navigation

In the web pane (left) of our visualization, we create special hyperlinks in the text at all occurrences of discovered object tags. When you mouse over one of the hyperlinks, it first highlights the object tag in the text pane and then the 3D visualization highlights a 3D bounding box around the corresponding object, showing you its location and relative size within the scene. Additionally, to emphasize the connection between the highlighted text and 3D bounding box, the visualization draws a line between them across the two panes.

When the named object is not visible in the current viewpoint, the visualization smoothly moves the camera until the object is displayed in the center of the image. To see an image of the highlighted object you can click on the object tag and the visualization first transitions to the viewpoint of a close-up image of the object and then fades in the image. For each object, the visualization chooses the image that maximizes the area of the object's projected 3D bounding box. Once you move the mouse out of the object tag, the line and the bounding box fade out.

The webpages often contain images that depict some of the objects being described. Our visualization further enhances these images by converting them to hyperlinks into the 3D model. When you click on the image, the camera transitions to the corresponding viewpoint in the 3D pane and fades in a high resolution version of it. This functionality is helpful when navigating webpages with many photos (e.g. U.S. Capitol Rotunda Wikipedia page), by providing the spatial context that relates them.

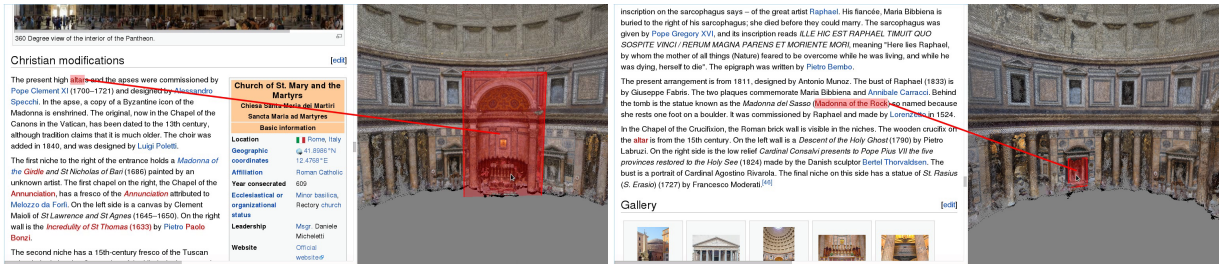
5.2 3D-to-text navigation

You can also navigate in the 3D visualization; dragging the mouse or using the mouse wheel changes the camera viewpoint. When the mouse pointer moves over the projection of the 3D bounding box of a discovered object, the visualization fades in the object's bounding box, hinting that you have found an object. After a short delay, the visualization automatically scrolls the website to show the corresponding object tag in the text pane, highlights it, and draws a connecting line between the tag and the object's bounding box across the two panes. In this way, you can learn about the objects in the scene by simply clicking on the areas of the 3D model and reading the text surrounding the highlighted object tag.

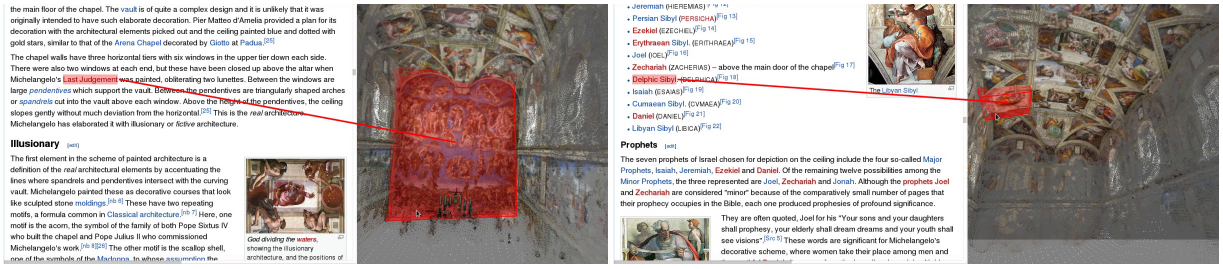
5.3 Tour navigation

In most authoritative text sources objects are not described in a random fashion, but follow a sensible pattern around the site. For example, the Wikipedia article of the Pergamon Altar describes the different sculptures in the Gygantomachy frieze from left to right. The Pantheon Wikipedia article first describes the apse, then the chapels and niches on the right side of the apse, followed by the ones on the left side. We can exploit this text structure to create more seamless navigation experiences, where an automated sequence of transitions between relevant images is played as the user reads through the text.

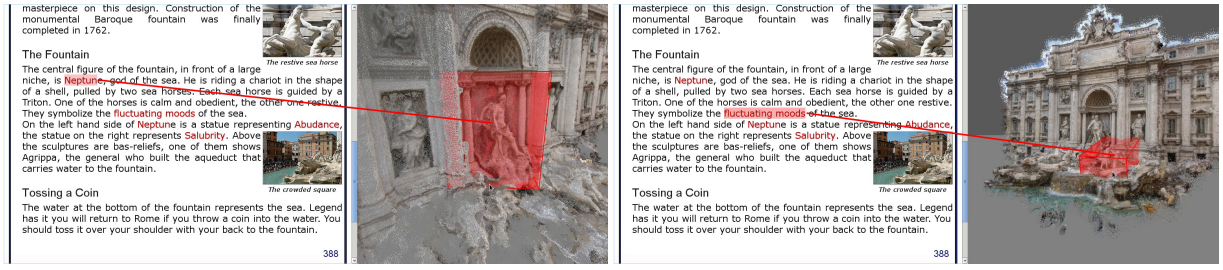
When the user activates the tour mode, a thin highlighting box appears over the text that covers the width of the pane. As the user scrolls through the text, object tags that enter the highlighting box cause the visualization to highlight the tags and automatically move the camera to show a close-up picture of the highlighted object. In this way, the camera automatically follows the exposition.



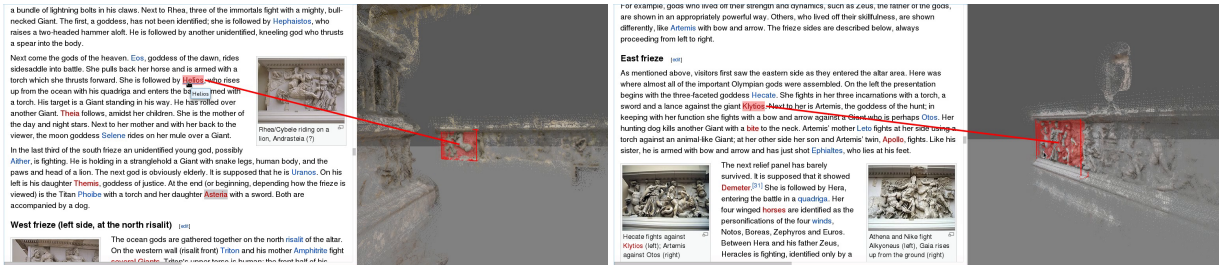
(a) Pantheon, Rome



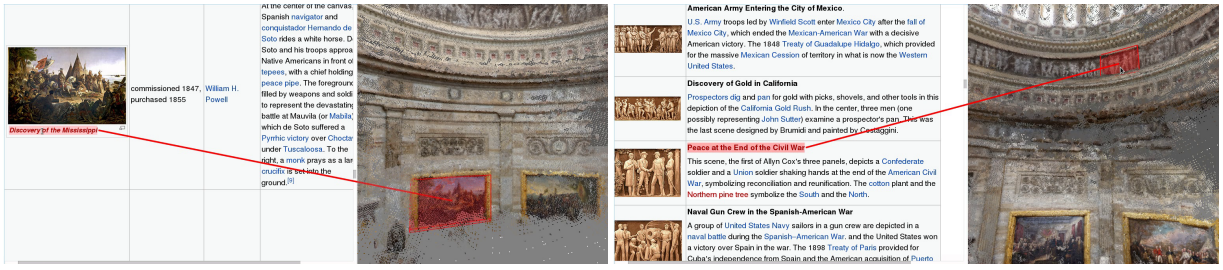
(b) Sistine Chapel, Vatican City



(c) Trevi Fountain, Rome



(d) Pergamon Altar, Berlin



(e) US Capitol Rotunda, Washington D.C.

Figure 5: *Screenshots of our visualizations for five different sites. Website and photographs in (c) courtesy of www.aviewoncities.com.*

Site	Pantheon, Rome	Trevi Fountain	Sistine Chapel	US Capitol Rotunda	Pergamon Altar
# 3D points	146K	208K	121K	84K	55K
# ground truth	31	16	31	38	49
# noun phrases	1796	821	3288	2179	2949
# image matches	510	348	2282	884	1600

Table 1: *Site statistics: # 3D points – number of points in 3D model, # ground truth – number of labeled ground truth objects, # noun phrases – number of automatically extracted noun phrases using the Stanford parser [Klein and Manning 2003], # image matches – number of noun phrases with an image returned from Google image search that aligned to the 3D model. When compared to the number of labeled ground truth objects, there are a large number of (spurious) automatically generated candidate detections (# image matches) that we must cope with.*

Site	Pantheon, Rome	Trevi Fountain	Sistine Chapel	US Capitol Rotunda	Pergamon Altar
Recall	0.39	0.31	0.71	0.21	0.18
Raw Precision	0.80	0.31	0.46	0.35	0.56
Full Precision	**0.87**	**0.78**	**0.79**	**0.65**	**0.94**

Table 2: *Detection accuracy. We measure the proportion of detected objects that are correctly localized in the scene (precision – 1.0 is optimal) and proportion of ground truth objects in the scene that are detected (recall – 1.0 is optimal). Chance is negligible, being proportional to the number of words or phrases on the input text. We report two precision numbers: the* raw precision*, which is the proportion of correctly localized objects using the manually labeled ground truth as a guide; and* full precision*, which uses manually verified detections corresponding to smaller unlabeled parts of the scene, such as the trumpets in Michelango's Last Judgement (see text).*

5.4 Implementation details

We used publicly available bundle adjustment and multi-view stereo software to automatically create the 3D models from Internet photo collections using VisualSFM [Wu et al. 2011; Wu a; Wu b] for generating a sparse point cloud, followed by PMVS [Furukawa and Ponce 2010] for generating a dense point cloud. As a post-processing step to filter noisy 3D points from the PMVS output, we apply Poisson Surface Reconstruction [Kazhdan et al. 2006] to generate a mesh. We then delete small connected components of the mesh and vertices that lie far away from the PMVS points. We then color the mesh vertices according to the closest PMVS points and keep the vertices of the mesh as our final point cloud. Although we generate colored meshes, we only use the vertices from the mesh for visualizations as we found the point cloud is visually more forgiving of artifacts; it avoids the uncanny valley effect and looks better than the mesh.

To highlight the objects in the 3D model, we generate 3D bounding boxes for each object that are rendered semi-transparently in our visualizations. First, we automatically estimate the principal axes of the 3D reconstructions using the recovered 3D scene information. We seek to estimate 3D bounding boxes that maximally align to the dominant orientation of the object, while remaining aligned to the principal axes. We first compute the mode $\mathbf{m}$ over the distribution of the normals of the points that lie in the frustra of the images. We then choose a coordinate unit-vector $\mathbf{x}$ in the world-coordinate frame of the reconstruction that is approximately orthogonal to the mode, preferring the z-axis over the x or y-axes. Finally, we calculate the other axis vector $\mathbf{y} = \frac{\bar{\mathbf{y}}}{\|\bar{\mathbf{y}}\|}$ with $\bar{\mathbf{y}} = \mathbf{m} - (\mathbf{m} \cdot \mathbf{x})\mathbf{x}$ and $\mathbf{z} = \mathbf{x} \times \mathbf{y}$. This approach produces compelling bounding boxes as seen in Figure 5.

6 Evaluation

As the different navigation modes for the visualization tool depend on the quality of the automatically generated text-object correspondences, we manually evaluate the accuracy of the correspondences. To measure performance, we collected reference texts and computed 3D models for 5 sites, which are listed in Table 1. For Trevi Fountain, we used three webpages. [1] For the remaining sites, we extracted text from their corresponding Wikipedia pages.[2] We evaluate performance relative to a set of *canonical views*, which are a set of registered panoramas or perspective photographs depicting most or all of a site. To define the gold standard, we manually labeled the name and a bounding box for all notable and well-described objects in the reference text using LabelMe [Russell et al. 2008].

We use a modified Pascal VOC criteria [Everingham et al. 2010] to score detections. First, we relax the relative overlap score to a lower value and require that the center of the detection window lies inside the ground truth. This is to account for the bias in how photographers frame the object in an image (typically not exactly cropped to the object). Moreover, we find that in our visualization task the relaxed bounding boxes still provide useful information about the objects in the scene. Second, we require that at least one of the returned words match the ground truth object tag after stop word removal. This is to handle the noisy nature of object tags where there can be many equally good descriptions, such as "a statue of the Madonna of the Rock by Lorenzetto", "Madonna of the Rock", "Modonna of the Rock by Lorenzetto".

We report site cross validation numbers, where the linear classifier is tested on a single site after being trained on all of the others. We return the top 37% scoring detections after non-maximum suppression. We found that this threshold provides a good balance between recall (many good detections) and precision (good accuracy).

We show example system outputs for the top most confident detections on the canonical views in Figures 6-8. Correct detections are shown in green and false positives in red. We also display the returned object tags near the corresponding detection window.

In scoring the detections, we found that synonyms in the object tags posed a problem. For example, the central statue in Trevi Fountain can be referred to as "Neptune" or "Ocean". We included such detected synonyms in the ground truth object tags. For scoring detections, we ignore any additional detections due to a synonym, after the most confident detection. In this way, a synonym detection does not help or hurt detection performance.

[1] http://www.aviewoncities.com/rome/trevi.htm, http://www.trevifountain.net, http://www.rome.info/sights/trevi-fountain

[2] We used the following Wikipedia articles: Pantheon, Rome; United States Capitol rotunda; Pergamon Altar; Sistine Chapel ceiling.

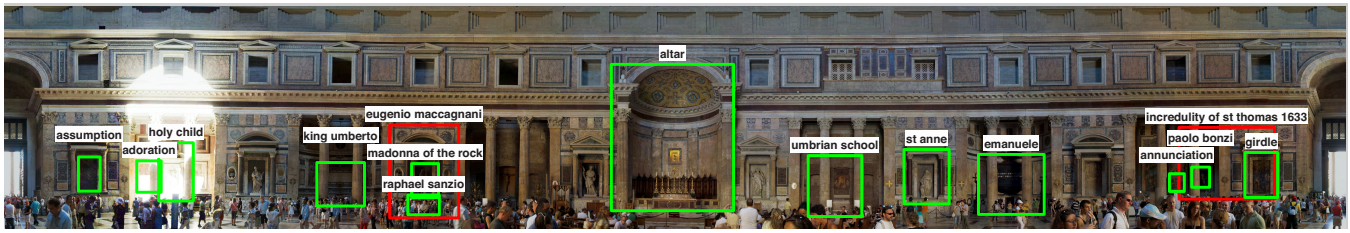


Figure 6: *Output detections for Pantheon, Rome using named objects automatically extracted from the reference text. Green – correct detection, with the returned object label overlaid. Red – false positives. Photograph by Patrick Landy.*

Figure 7: *Output detections for Sistine Chapel. Photograph by Roy Zipstein, 360Cities.net.*

Table 2 reports detection precision and recall over the returned detections for all sites. Most recall numbers range from 0.18 for Pergamon Altar to 0.39 for the Pantheon. The Sistine Chapel is a notable exception, which has 0.71 recall. This is mostly due to the fact that the site is well catalogued and features many close-up images available on the Internet. While these recall numbers may seem low, they are quite reasonable in the context of our visualization application. Note, for example, the current Pantheon Wikipedia page includes captioned close-up photos of only **two** out of 31 objects–a recall of only 6%. Our automatic procedure boosts this to 39% and provides a better integrated 3D experience, i.e., in the context of our application, it is not critical to detect *every* object in the scene to provide a compelling and useful visualization experience. More important is that we capture the most important objects in the scene, and important objects tend to be well cataloged with labeled photos on the Internet (which lead to better recall in our system).

For detection precision, we report two numbers: the *raw precision*, which is the proportion of correctly localized objects using the manually labeled ground truth as a guide; and the *full precision*, which uses manually verified detections. The latter is motivated by the fact that often the returned detections correspond to smaller parts of the scene, such as the *trumpets* in *Michelangelo's Last Judgement*, link to relevant descriptive text for other objects, such as *Garden of Eden* for *The Temptation and Expulsion*, or refer to a generic object category, such as *wall frescoes* in the Sistine Chapel. Including these additional detections, we achieve a full precision ranging from 0.65 for US Capitol Rotunda to 0.94 for Pergamon Altar. We believe that this accuracy is reasonable for our 3D visualization, as it allows the user to browse the text and scene with a minimum of incorrect correspondences.

To put these numbers in context, as a baseline, we also computed object recall using the tags associated with the Flickr images that were used to build the 3D models. This Flickr baseline is an upper bound on prior work that segments and labels 3D point clouds by analyzing SIFT feature co-occurence in tagged Flickr photos [Simon and Seitz 2008]. We computed the proportion of ground truth objects that find a Flickr image whose tag overlaps at least partially with the ground truth tag and depict the ground truth object. We report the object recall for the sites in which we retained the Flickr tags: Pantheon – 0.06; Trevi Fountain – 0; US Capitol Rotunda – 0.21. Notice that the Flickr baseline performs significantly worse for the Pantheon and Trevi Fountain. On the other hand, the US Capitol Rotunda appears to be well-documented on Flickr and achieves the same recall as our full system, with many of these images appearing in the query expansion step. However, it is not straightforward to filter the many false positive tags that appear in the Flickr image set.

6.1 Error Analysis

We have observed different sources of errors of our system, which result in inaccurate labels returned by our system and missed objects. We describe these errors and illustrate them in Figure 9.

One common case is when text spans are paired with bounding boxes that contain the named object, but are too large. This happens

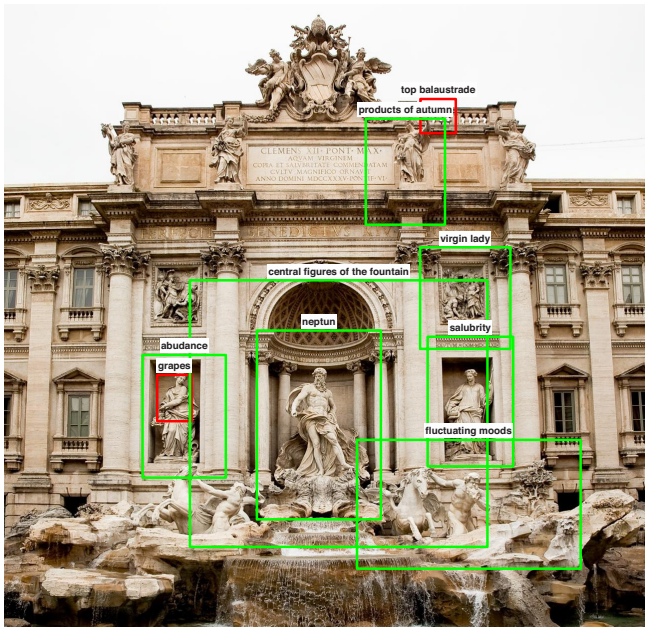
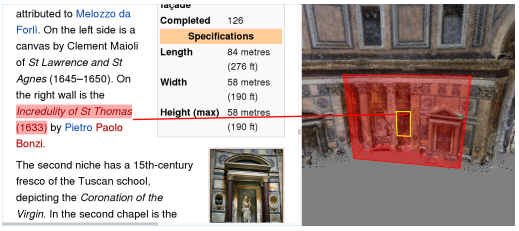
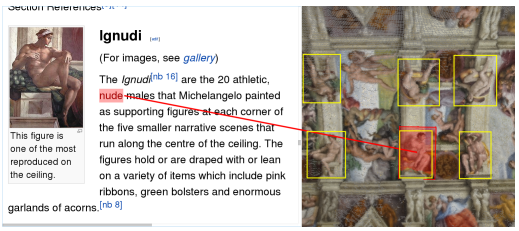


Figure 8: *Output detections for Trevi Fountain. Photograph by garygraphy.*



(a) Bounding box is too large

(b) Incorrect object tag



(c) Multiple object class instances

Figure 9: *Failure cases: obtained bounding box is shown in red and correct objects are shown in yellow.*

when Google image search returns images that correctly depict the object of interest, but are not tightly cropped to the object. For example, in Figure 9(a) the bounding box for the painting *The Incredulity of St. Thomas* is large and encloses the object itself, along with the first niche and the first chapel.

We have also observed incorrect object correspondences, such as the one shown in Figure 9(b). The recovered bounding box for the object *Eugenio Maccagnani* encloses the niche around the *tomb of Raphael*, which is described in the following paragraph in the text. These errors typically come from noisy co-occurrences between images and text in the online sources.

A challenging case is when an object is not a specific instance, but belongs to a set, as shown in Figure 9(c). Here, the *ignudi* describes the set of depicted nudes in the Sistine chapel. Our current system cannot identify all of them since the system assumes a one-to-one correpondence between a named object and its depiction in the 3D scene. While we could relax this constraint, it would result in lower overall precision due to the noisy results of the Google image search.

In addition, we have observed failures that result in object misses. These are primarily due to: (1) incorrect images that are returned from Google image search for a candidate object, and (2) when the object is poorly reconstructed during the structure-from-motion step, causing the Google images not to match. This can be partially remedied by better online documentation of the objects and improved 3D models.

We find that there is evidence in the text to cope with some of these errors. For example, the *Incredulity of St. Thomas* is described to be "on the right wall" of the *Chapel of the Annunciation*; there is a clear description of the *ignudi* being multiple figures: "the Ignudi are the 20 athletic, nude males." Also, there is often information in the text about the class of the objects, e.g. a named object can be described as being a *painting* or *statue*. The category of the object could be extracted from the text and used with object detectors trained for the category. Moreover, bottom-up segmentation could be used to improve object localization. Developing a model that could incorporate such cues is an important area for future work.

7 Conclusion

This paper introduced the first system capable of using online text and photo collections to automatically build immersive 3D visualizations of popular sites. These included a number of new interactions: text can be selected to move the camera to the corresponding objects, 3D bounding boxes provide anchors back to the text describing them, and the overall narrative of the text provides a temporal guide for automatically flying through the scene to visualize the world as you read about it.

While our system is built using off-the-shelf ingredients, we argue that the ideas and the system are new. In particular, we introduce (1) the concept for a 3D Wikipedia based on crowd-sourced data on the Internet, (2) the insight of using text parsing + Google image search to connect web text to 3D shape data, and (3) a viable approach that accomplishes this goal, incorporating a series of sophisticated steps (3D reconstruction, text parsing, and a classifier to improve precision). Experiments on multiple sites demonstrate that this approach has consistently high precision, which is crucial for enabling high quality interactions, even if all objects are not yet recognized. Our current system works on the most popular sites, as it requires lots of images and good text. Going forward, with growth in photo and text corpi, the system will work "as is" for more scenes as the underlying data improves. Improvements to 3D reconstruction algorithms and text parsers will also further improve applicability.

While the results are encouraging, there is room for improvement.

While improvements in search technology will reduce false negatives (missed objects), we have barely tapped into the structure and constraints expressed in the text, which have significant potential to reduce false positives (mislabeled objects). For example, one especially promising topic for future work will be to leverage *spatial* terms (e.g., "in the first niche to the left of the door...", "the painting above the altar...") to constrain the placement of objects in the scene. Developing semi-automated methods that leverage people to assist in the labeling task is another interesting topic of future work.

Acknowledgements

The research was supported in part by the National Science Foundation (IIS-1250793), the Intel Science and Technology Centers for Visual Computing (ISTC-VC) and Pervasive Computing (ISTC-PC), and Google. We thank Kristiaan Van Ermengem for allowing us to include the screenshots of the website www.aviewoncities.com in Figure 5(c) and Roy Zipstein, 360Cities.net for allowing us to reproduce their photograph in Figure 7. We acknowledge the following people whose photographs we reproduced under Creative Commons licenses: Patrick Landy[3], garygraphy[3], Jean-Pol Grandmont[3], Ricardo André Frantz[4], Bengt Nyman[3], Claus Ableiter[4].

References

AGARWAL, S., FURUKAWA, Y., SNAVELY, N., SIMON, I., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2011. Building rome in a day. *Communications of the ACM 54*, 10 (Oct.), 105–112.

BARNARD, K., DUYGULU, P., DE FREITAS, N., FORSYTH, D., BLEI, D., AND JORDAN, M. I. 2003. Matching words and pictures. *Journal of Machine Learning Research 3*, 1107–1135.

BERG, A. C., BERG, T. L., III, H. D., DODGE, J., GOYAL, A., HAN, X., MENSCH, A., MITCHELL, M., SOOD, A., STRATOS, K., AND YAMAGUCHI, K. 2012. Understanding and predicting importance in images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3562–3569.

BERLITZ INTERNATIONAL, I. 2003. *Berlitz Rome Pocket Guide*. Berlitz Pocket Guides Series. Berlitz International, Incorporated.

BUCKLEY, C. 1995. Automatic query expansion using SMART : TREC 3. In *Proceedings of the third Text REtrieval Conference (TREC-3)*, 69–80.

CHUM, O., PHILBIN, J., SIVIC, J., ISARD, M., AND ZISSERMAN, A. 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE 11th International Conference on Computer Vision (ICCV)*, 1–8.

COUR, T., SAPP, B., AND TASKAR, B. 2011. Learning from partial labels. *Journal of Machine Learning Research 12* (May), 1501–1536.

CRANDALL, D., BACKSTROM, L., HUTTENLOCHER, D., AND KLEINBERG, J. 2009. Mapping the world's photos. In *Proceedings of the 18th International Conference on World Wide Web (WWW)*, 761–770.

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., WINN, J., AND ZISSERMAN, A. 2010. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision 88*, 2, 303–338.

[3]http://creativecommons.org/licenses/by/2.0

[4]http://creativecommons.org/licenses/by-sa/3.0/deed.en

FARHADI, A., HEJRATI, M., SADEGHI, M. A., YOUNG, P., RASHTCHIAN, C., HOCKENMAIER, J., AND FORSYTH, D. 2010. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision (ECCV)*, 15–29.

FURUKAWA, Y., AND PONCE, J. 2010. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 8, 1362–1376.

FURUKAWA, Y., CURLESS, B., SEITZ, S. M., AND SZELISKI, R. 2010. Towards internet-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1434–1441.

GARWOOD, D., AND HOLE, A. 2012. *Lonely Planet Rome*. Travel Guide. Lonely Planet Publications.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *IEEE 11th International Conference on Computer Vision (ICCV)*, 1–8.

HARTLEY, R. I., AND ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN: 0521540518.

HAYS, J., AND EFROS, A. A. 2008. IM2GPS: estimating geographic information from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing (SGP)*, 61–70.

KLEIN, D., AND MANNING, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 423–430.

LADICKÝ, L., STURGESS, P., RUSSELL, C., SENGUPTA, S., BASTANLAR, Y., CLOCKSIN, W., AND TORR, P. H. S. 2012. Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision 100*, 2, 122–133.

LAPTEV, I., MARSZALEK, M., SCHMID, C., AND ROZENFELD, B. 2008. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2, 91–110.

MITCHELL, M., DODGE, J., GOYAL, A., YAMAGUCHI, K., STRATOS, K., HAN, X., MENSCH, A., BERG, A. C., BERG, T. L., AND DAUMÉ III, H. 2012. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 747–756.

PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

RAGURAM, R., WU, C., FRAHM, J.-M., AND LAZEBNIK, S. 2011. Modeling and recognition of landmark image collections using iconic scene graphs. *International Journal of Computer Vision 95*, 3, 213–239.

REN, X., BO, L., AND FOX, D. 2012. RGB-(D) Scene labeling: Features and algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2759–2766.

RUSSELL, B. C., TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2008. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision 77*, 1-3, 157–173.

SALTON, G., AND BUCKLEY, C. 1999. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 41*, 4, 288–297.

SILBERMAN, N., HOIEM, D., KOHLI, P., AND FERGUS, R. 2012. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision (ECCV)*, 746–760.

SIMON, I., AND SEITZ, S. M. 2008. Scene segmentation using the wisdom of crowds. In *European Conference on Computer Vision (ECCV)*, 541–553.

SIVIC, J., AND ZISSERMAN, A. 2003. Video Google: A text retrieval approach to object matching in videos. In *IEEE 9th International Conference on Computer Vision (ICCV)*, 1470–1477.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (SIGGRAPH) 25*, 3, 835–846.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2008. Modeling the world from Internet photo collections. *International Journal of Computer Vision 80*, 2, 189–210.

Stop words list. `http://norm.al/2009/04/14/list-of-english-stop-words/`.

Wikipedia. `http://www.wikipedia.org`.

WU, C. SiftGPU: A GPU implementation of scale invaraint feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`.

WU, C. VisualSFM: A visual structure from motion system. `http://homes.cs.washington.edu/~ccwu/vsfm/`.

WU, C., AGARWAL, S., CURLESS, B., AND SEITZ, S. M. 2011. Multicore bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3057–3064.