Probabilistic Temporal Inference on Reconstructed 3D Scenes

Grant Schindler and Frank Dellaert
Georgia Institute of Technology
{schindler,dellaert}@cc.gatech.edu

Abstract

Modern structure from motion techniques are capable of building city-scale 3D reconstructions from large image collections, but have mostly ignored the problem of large-scale structural changes over time. We present a general framework for estimating temporal variables in structure from motion problems, including an unknown date for each camera and an unknown time interval for each structural element. Given a collection of images with mostly unknown or uncertain dates, we use this framework to automatically recover the dates of all images by reasoning probabilistically about the visibility and existence of objects in the scene. We present results on a collection of over 100 historical images of a city taken over decades of time.

1. Introduction

Recent progress in 3D reconstruction from images has enabled the automatic reconstruction of entire cities from large photo collections [1], and yet these techniques largely ignore the fact that scenes like cities can change drastically over time. In this paper, we introduce a language for representing time-varying structures, and a probabilistic framework for doing inference in these models. The goal of this framework is to enable the recovery of a date for each image and a time interval for each object in a reconstructed 3D scene.

As institutions digitize their archival photo collections, millions of photographs from the late 19th and 20th centuries are becoming available online, many of which have little or no precise date information. Recovering the date of an image is therefore an important task in the preservation of these historical images, and one currently performed by human experts. In addition, having a date on every image in a 3D reconstruction would allow for intuitive organization, navigation, and viewing of historical image collections registered to 3D city models. Discovering the time intervals of existence for every object in a scene is also an essential step toward automatically creating *time-varying* 3D models of cities directly from images. Toward this end, we introduce a probabilistic framework for performing temporal inference on reconstructed 3D scenes.

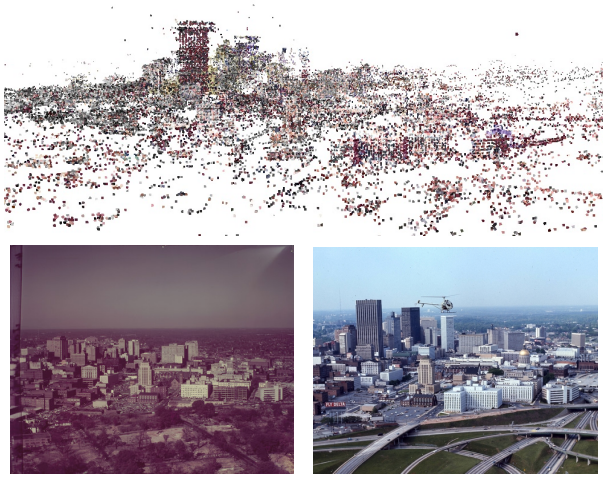


Figure 1: We build a 3D reconstruction automatically from images taken over multiple decades, and use this reconstruction to perform temporal inference on images and 3D objects. The left image was taken in 1956 while the right photo was captured in 1971 from nearly the same viewpoint.

1.1. Related Work

A number of recent approaches to large-scale urban modeling from images have produced impressive results [2, 1, 15], though none have yet dealt explicitly with time-varying structure. In [13], a historical Ansel Adams photograph is registered to a reconstructed model of Half Dome in Yosemite National Park, but there is no notion of time in this process – only the location of the image is recovered. Additionally, since we are dealing with historical photographs, approaches that rely on video [2], densely captured data [15], or additional sensors are not directly applicable to our problem.

Current *non-automated* techniques for dating historic photographs include identifying clothing, hairstyles, and cultural artifacts depicted in images [8, 9], and physical examination of photographs for specific paper fibers and

chemical agents [7]. Our approach deals with digitized photographs and contain few human subjects, so we instead opt to reason about the existence and visibility of semi-permanent objects in the scene.

Visibility and occlusion reasoning have a long history in computer vision with respect to the multi-view stereo problem [5, 6]. A space carving approach is used in [6] to recover the 3D shape of an object from multiple images with varying viewpoints. This involves reasoning about occlusions and visibility to evaluate the photo-consistency of scene points, and relies upon the assumption that the space between a camera center and a visible point is empty. More recently in [3], visibility is used to provide evidence for the emptiness of voxels in reconstructing building interiors. Our visibility reasoning approach differs from all of these in that both the potentially visible objects and potentially occluding objects vary with time, thus invalidating all the visibility assumptions that apply to static scenes. In our approach, we will be searching for a temporal story that explains why we do and don't see each object in each image.

The most similar work to ours is that of [11], which proposed a constraint-satisfaction method for determining temporal *ordering* of images based on manual point correspondences. This approach suffers from a number of weaknesses: only an image ordering is recovered, there is no way to incorporate known date information, the occlusion model is static, manual correspondences are required, and there is no concept of objects beyond individual points. In contrast, our approach offers a number of advantages:

Time-Dependent Occlusion Geometry. A major problem with the method of [11] is the assumption of a fixed set of occluding geometry. Here, we treat the *uncertain scene geometry itself* as the occlusion geometry, which complicates visibility reasoning but which is necessary for dealing with real-world scenes.

Continuous, Absolute Time. Our method recovers a specific continuous date and time for each image and is able to explicitly deal with missing and uncertain date information while incorporating known dates into the optimization problem. [11] only deals with orderings of images.

Automatic 3D Reconstruction. The manual correspondences in [11] act as perfect observations, which are not present in an automatic reconstruction. Automated feature matching cannot ensure that every feature is detected in every image, so we must deal with missing measurements.

Object-Based Reasoning. Rather than reasoning about the visibility of points as in [11], we reason about entire 3D objects which can be composed of numerous points, or any other geometric primitives. Crucially, each object explicitly has its own time interval of existence.

In addition, the method of [11] turns out to be a special case of our more general probabilistic framework. Through developing our probabilistic temporal inference framework, we simultaneously gain insight into the previous approach of [11] while creating a more powerful method for reasoning about temporal information in reconstructed 3D scenes.

2. Approach

The traditional Structure from Motion (SfM) problem is concerned with recovering the 3D geometry of a scene and of the cameras viewing that scene. In this work, in addition to this *spatial* information we are also interested in recovering *temporal* information about the scene structure and the cameras viewing the scene. This temporal information consists of a date for each camera and a time interval for each 3D point in the scene. Though we can theoretically solve for both the spatial and temporal SfM parameters simultaneously, we choose here to decompose the problem into two steps, first solving traditional SfM (Section 4.1) and then solving the temporal inference problem (Section 3).

2.1. Time-Varying Structure Representation

We first define the representation we will use to perform temporal inference on reconstructed 3D scenes. Given a set of n images $I_{1..n}$ registered to a set of m 3D objects $O_{1..m}$, we wish to estimate a time t associated with each image, and a time interval (a,b) associated with each 3D object. We represent the entirety of these temporal parameters with $T = (T^O, T^C)$ where

$$T^O = \{(a_i, b_i) : i = 1..m\}$$

is a set of time intervals, one for each object, and

$$T^C = \{t_j : j = 1..n\}$$

is a set of timestamps, one for each image.

We assume that we are given a set of geometric parameters $X = (X^O, X^C)$ for the scene, where $X^O = \{x_i : i = 1..m\}$ describes the geometry of each object and $X^C = \{c_j : j = 1..n\}$ describes the camera geometry for each image. The approach is general and these 3D objects can be, for example, points, planes, or polygonal buildings. The only requirement is that each 3D object must be detectable in images and must be capable of occluding other objects.

2.2. Sources of Temporal Information

In this work, we assume that for *some* images we have at least uncertain temporal information. Without any time information, the best we can do is determine an ordering as in [11]. In practice, we will usually have a mix of dated images, undated images, and images with uncertain date estimates.

Modern digital cameras nearly always embed the precise date and time of the photograph in the Exif tags of the resulting image file. This includes the year, month, day, hour,

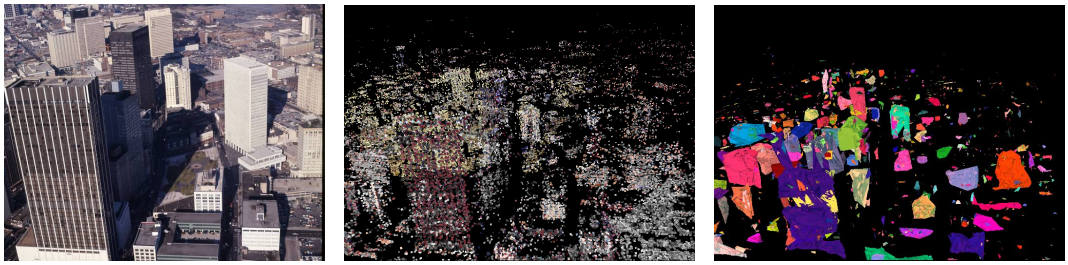


Figure 2: Point Groupings. The 3D points that result from Structure from Motion are unsuitable for use in visibility reasoning because (1) they are not reliably detected in every image, (2) they don't define solid occlusion geometry, and (3) there are too many of them. We solve all these problems by grouping 3D points into the objects about which we will reason. Points which are physically close and have been observed simultaneously in at least one image are grouped into these larger structures.

minute, and second at which the image was captured. Thus, we have nearly a decade of time-stamped digital photos compared to the previous 17 decades of photography which lacks this precise temporal information. Digitized historical photographs will have associated date information only when a human archivist manually enters such a date into a database. When available, precise dates can be found in the original photographer's notes, but the more common case is that a human exercises judgment to place a date label like "circa 1960" on the photograph.

We examined the date information on a set of 337 historical images from the Atlanta History Center and found that less than 11% of the images have a known year, month, and day. Of all images, 47% are "circa" some year, 29% have a known year, 6% have a known year and month, 3% are "before" or "after" some year, and 4% are completely undated. This lack of precise temporal information for a majority of historical photographs motivates our work.

Given a photograph I_j labeled with a year y_j, month m_j, and day d_j, the date of the photograph $t_j \in \mathbb{R}$ is represented as $t_j = y_j + f(m_j, d_j)/365$. This is the value of the year plus the fractional amount of a year accounted for by the day and month where $f()$ is a function from month and day to sequential day of the year. We make this explicit because historical photographs are often labeled with a year only, for example 1917, in which case we only know that the true date of the photograph lies within an interval $t_j \in [1917.0, 1918.0)$. In such a case, we take the midpoint of the interval as an initial estimate of t_j.

3. Probabilistic Temporal Inference Model

Our goal is to estimate the time parameters T of a set of images and objects given the geometric parameters X of a reconstructed 3D scene. In addition, we assume that we are given a set of observations $Z = \{z_{ij} : i = 1..m, j = 1..n\}$ where each z_{ij} is a binary variable indicating whether object i was observed in image j. In what follows, we will be searching for the set of temporal parameters T that best explain the observations Z, telling us why we see certain objects in some images but not in others. In Bayesian terms, we wish to perform inference on all temporal parameters T given observations Z and scene geometry X,

$$P(T|Z,X) \propto P(Z|T,X)P(T) \tag{1}$$

In the following two sections, we discuss the likelihood term $P(Z|T,X)$ first and then the prior term $P(T)$.

3.1. Observation Model

The key term which we need to evaluate is the likelihood $P(Z|T,X)$. Because the observations are conditionally independent given T, we can factor the likelihood as:

$$P(Z|T,X) = \prod_{z_{ij} \in Z} P(z_{ij}|T,X) \tag{2}$$

This is the product, over all objects in all images, of the probability of each individual observation z_{ij} given T and X. Evaluation of the terms $P(z_{ij}|T,X)$ relies on three factors:

Viewability: Is object i within the field of view of camera j? This only depends on the geometry X, more specifically for each measurement z_{ij} we can deterministically evaluate the function $InFOV_{ij}(X)$ that depends only on the object and camera geometry x_i and c_j.

Existence: Did object i exist at the time image j was captured? This only depends on the temporal information T, as given T we can deterministically evaluate the functions $Existence_{ij}(T) = a_i \leq t_j \leq b_i$.

Occlusion: Is object i occluded by some other object(s) in image j? This attribute, $Occluded_{ij}(T,X)$, depends on both temporal information T and geometry X. Specifically, $Occluded_{ij}(T,X)$ depends upon *all* time intervals T^O, *all* object geometry X^O, and camera parameters (t_j, c_j).

Below we discuss each of these factors in turn.

3.1.1 Viewability

Based on viewability alone, we can factor the likelihood (2) in two parts: one that depends on the temporal information

T and one that does not. Indeed, if we define the *viewable set* $Z_V = \{z_{ij} | InFOV_{ij}(X)\}$, we have

$$P(Z|T,X) = k \prod_{z_{ij} \in Z_V} P(z_{ij}|T,X) \quad (3)$$

where k is a constant that does not depend on T, and hence is irrelevant to our inference problem. In practice all the measurements z_{ij} not in the viewable set Z_V are 0, so the above simply states that we do not even need to consider them. However, the viewability calculation has to be done to be able to know *which* measurements z_{ij} to disregard.

3.1.2 Existence

The viewable set Z_V can, given the temporal information T, be further sub-divided into two sets Z_N and Z_P, where $Z_P = \{z_{ij} | z_{ij} \in Z_V \wedge Existence_{ij}(T)\}$ corresponds to the set of image-object pairs (i, j) that co-exist given T, and its complement $Z_N = Z_V \setminus Z_P$ is the set of all measurements predicted to be negative because the object and image did *not* co-exist. *Crucially, note that this division depends on the temporal parameters T.* Hence, the likelihood (3) can be further factored as

$$P(Z|T,X) = k \prod_{z_{ij} \in Z_N} P_N(z_{ij}) \prod_{z_{ij} \in Z_P} P_P(z_{ij}|T,X)$$

The first product above dominates the likelihood, as it is very improbable that an object i will be reported as visible in camera j if in fact it did not exist at the time image j was taken. In other words, $P_N(z_{ij} = 1) = \rho$, with the *false positive probability* ρ a very small number. Hence the likelihood stemming from the observations in Z_N is simply

$$P(Z_N|T,X) = \prod_{z_{ij} \in Z_N} P_N(z_{ij}) = \rho^{FP}(1-\rho)^{CN} \quad (4)$$

where FP and CN are the number of *false positives* and *correct negatives* in the set Z_N, with $FP + CN = |Z_N|$. Note that in the case $\rho = 0$ the likelihood $P(Z_N|T,X)$ evaluates to zero for any assignment T violating an existence constraint.

3.1.3 Occlusion

Finally, if object i *does* exist when image j is taken, then the probability $P_P(z_{ij}|T,X)$ that it is observed depends upon whether it is occluded by other objects in the scene, i.e.,

$$P_P(z_{ij}|T,X) = \eta \times P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O) \quad (5)$$

with η the *detection probability* for unoccluded objects. Since we rely on SfM algorithms, even unoccluded objects might not be reconstructed properly: the reasons include failure during feature detection or matching, or occlusion by an un-modeled object such as a tree or car. Although we use a constant term η here, this probability could be evaluated on a per object/per image basis using the known scene and camera geometry. For example, we could capture the notion that a small object is unlikely to be observed from a great distance despite being in the field of view.

The occlusion factor $P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O)$ can in turn be written as the probability of object i not being occluded by any other object k,

$$P(\overline{Occluded_{ij}}|t_j, c_j, T^O, X^O) = \prod_{k \neq i} (1 - P(Occlusion_{ijk}|t_j, c_j, a_k, b_k, x_k, x_i))$$

where $Occlusion_{ijk}$ is a binary variable indicating whether or not object i is occluded by object k in image j. The probability $P(Occlusion_{ijk}|.)$ can vary from 0 to 1 to account for partial occlusions of objects. With this model, the overall probability $P(Occluded_{ij}|t_j, c_j, T^O, X^O)$ that object i has been occluded by *something* in image j *increases* as more individual objects k partially occlude object i. A specific occlusion model will be discussed further in Section 4.3.

3.2. Temporal Prior

The term $P(T)$ in Equation (1) is a prior term on temporal parameters. This can be further broken down into image date priors $P(T^C) = \prod_{j=1..n} P(t_j)$ and object time interval priors $P(T^O) = \prod_{i=1..m} P(a_i, b_i)$.

If we have any prior knowledge about when an image was taken, we account for it in the individual $P(t_j)$ prior terms. We may know an image's time down to the second, we may just know the year, or we may have a multi-year estimate like "circa 1960". In all such cases, we choose a normal distribution $P(t_j) = N(\mu, \sigma^2)$ with a σ appropriate to the level of uncertainty in the given date. When we have no date information at all for a given image, we use a uniform distribution appropriate to the data set – for example, a uniform distribution over the time between the invention of photography and the present. Though not used here, object interval priors $P(a_i, b_i)$ can also be chosen to impose an expected duration for each object.

3.3. Framework Extensions

An added benefit of this probabilistic temporal inference framework is that it becomes easy to extend the model to account for additional domain knowledge (though we do not use these extensions here). We can introduce a term $P(X^O|T^O)$ which encodes information about the expected heights of buildings given their construction dates, exploiting the fact that buildings have gotten progressively taller at a known rate over the last century, or a term $P(X^C|T^C)$ which incorporates prior information on the expected altitude of cameras given image dates, again exploiting the fact that we have records describing when airplanes, helicopters, and tall rooftops came into being and enabled

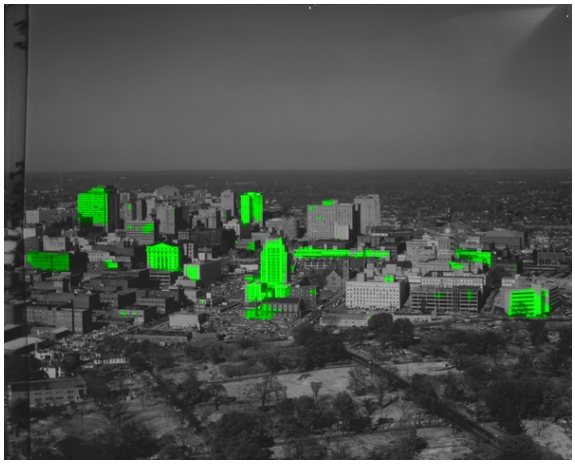


Figure 3: Object Observations. Our framework reasons about observations of 3D objects in images. We group the 3D points from SfM into larger structures and count the detection of at least one point in the group as an observation of the entire structure. Regions highlighted in green (above) represent observed objects in this image. False negative observations are undesirable but unavoidable, and we account for them in our probabilistic framework.

higher-altitude photographs to be captured. Both of these extensions would require the measurement of a known object to be specified in the scene in order to reason in non-arbitrary units.

Finally, we can introduce a term $P(I|T^C)$ specifying a distribution on image features for photos captured at a given time. Such features might include color or texture statistics, or even detections of cultural artifacts like cars or signs which are typical of specific historical eras, properties which already allow humans to roughly estimate the date of a photograph of an unfamiliar city scene. This would be especially significant in the case of historic cities which have not structurally changed much during the era of photography, where visibility reasoning alone may not be sufficient to pinpoint the date of an image.

3.4. Temporal Inference Algorithms

We are interested in finding the the optimal value T^* for the temporal parameters according to the maximum a posteriori (MAP) criterion:

$$T^* = \underset{T}{\operatorname{argmax}} \; P(T|Z,X)$$

Observe that, based on the above formulation, given a hypothesized set of temporal parameters T we can directly evaluate Equation (1) to get the probability of the hypothesized time parameters. Therefore, we perform temporal inference by *sampling* time parameters to find those that maximize the probability of the data.

3.4.1 Markov Chain Monte Carlo

We adopt a Markov Chain Monte Carlo (MCMC) approach to draw samples from the posterior distribution $P(T|Z,X)$ in order to find the optimal set of parameters T^*. Following the Metropolis-Hastings [4] algorithm, we start from an initial set of temporal parameters T and propose a move to T' in state space by changing one of the t_j, a_i, or b_i values according to a proposal density $Q(T';T)$ of moving from T to T'. We accept such a move according to the acceptance ratio:

$$\alpha = \min\left\{\frac{P(T'|Z,X)\,Q(T;T')}{P(T|Z,X)\,Q(T';T)}, 1\right\} \tag{6}$$

Our proposals involve randomly choosing a time parameter and adding Gaussian noise to its current value, such that our proposal distribution is symmetric, and the acceptance ratio is simply the ratio of the posterior probability $P(T|Z,X)$ of each set of temporal variables. Following this approach, we draw samples from the posterior probability $P(T|Z,X)$, keeping track of our best estimate for T^* as we do so.

We make this sampling approach more efficient by sampling only on image dates T^C, and analytically solving for the optimal object time intervals T^O for a given configuration of T^C. To do so, we note that the dominant likelihood part given by Equation (4) factors over objects i:

$$\prod_{z_{ij}\in Z_N} P_N(z_{ij}) = \prod_i \left\{ \prod_{j|z_{ij}\in Z_N} \rho^{FP_i}(1-\rho)^{CN_i} \right\}$$

Given the image dates T^C, we can eliminate false positives FP_i for each object i by setting

$$a_i \le \min\left\{t_j | z_{ij}=1\right\} \text{ and } b_i \ge \max\left\{t_j | z_{ij}=1\right\}$$

In other words, and obvious in hindsight, we make each object's interval such that it starts before its first "sighting" and ends after its last "sighting". In practice we found that extending the intervals beyond the minimum range indicated above has a negative effect on the solution: while extending an interval can help "explain away" negative observations of other objects, this also automatically incurs a $(1-\eta)$ likelihood penalty for every image in which the object is now not observed. This dominates the potentially beneficial effects.

Hence, for every proposed change to the image dates T^C, we adapt the object intervals (a_i, b_i) to minimize the existence constraints (4). This changes the set Z_P for which the occlusion/detection likelihood (5) needs to be evaluated. It is computationally efficient to propose to only change one image date t_j at a time, in which case only objects in view of camera j have their intervals adjusted, and calculating the acceptance ratio (6) is easier. However, occlusion effects will still have non-local consequences: in Section (4.3) we discuss how to deal with those efficiently as well.

(a) May 1971 (b) Jan 1969 (c) Dec 1969

Figure 4: Optimal Image Dates. These images were originally labeled as "circa 1965", 1868, and 1967 in a historical image database created by human experts. Our temporal inference method is able to improve upon these date labels as indicated below each image. Building construction records show these new dates estimates are more accurate than the human estimates.

4. Implementation

The above formulation is a general temporal inference framework applicable to a variety of situations. For the specific case of reasoning about cities over decades of time, we must specify how we recover geometry X using SfM and what kind of objects O we are dealing with, as well as how these objects are detected and how they occlude each other.

4.1. Structure from Motion

Before performing any temporal inference, we run traditional SfM to recover the camera geometry X^C and a set of 3D points which will form the basis for the geometry of our 3D objects X^O. For this purpose, we use the Bundler SfM software from Snavely [12] with SIFT implementation from VLFeat [14]. Depending on the connectivity of the match table, there may be multiple disconnected reconstructions that result from this SfM procedure. In our case, we are not interested in the reconstruction with the largest number of images, but rather the one containing images which *span the largest estimated time period*.

4.2. Object Model

We must define the set of 3D objects $O_{1..m}$ on which to perform temporal inference. The output of SfM is a large number of 3D points, but in a large-scale urban reconstruction, it makes more sense to reason directly about 3D buildings than 3D points. Segmenting point clouds into buildings is a difficult task, complicated here by the fact that *multiple buildings can exist in the same location* separated only by time. To solve this problem, we perform an oversegmentation of the points into point-groups, analogous to superpixels used in 2D segmentation [10]. Specifically, if two 3D points are closer than a threshold d_{group} and are also observed simultaneously in at least N_{group} images, we link them together and then find connected components among all linked points (see Figure 2).

Grouping points in this way leads to several benefits. First, we can count an observation of any one point in a group as an observation of the whole group (see Figure 3). This increases the chance of successfully detecting each object in as many images as possible, reducing false negatives. By reducing the number of 3D objects, we also vastly reduce the computational burden during occlusion testing. For the purposes of visibility reasoning, we triangulate each group of points (based on either a 3D convex hull or a union of view-point specific Delaunay triangulations) and use this triangulated geometry to determine which groups potentially occlude each other.

4.3. Occlusion Model

We must determine which objects in our scene potentially occlude which other objects, as this information plays a pivotal role in evaluating the probability of a given configuration of temporal parameters as described in Section 3.1.3. This involves the creation of an occlusion table, a three-dimensional table of size $m \times m \times n$ which specifies, for each image, the probability $P(Occlusion_{ijk}|X,T)$ that object k occludes object i in image j if *both objects exist at the same time*. The occlusion table is extremely sparse, but it is the most expensive computation in the entire algorithm due to the fact that m^2n geometric calculations must be made to compute it.

This expensive occlusion table computation is where we pay the price for not committing to a static set of occlusion geometry as in [11]. As our model's time parameters vary during optimization, the number of unique occlusion scenarios is 2^m where the number of objects m reaches into the thousands. We cannot precompute occlusion information for all these scenarios, nor do we want to compute occlusion events on the fly while evaluating the probability of a specific set of temporal parameters – this slows down evaluation by an order of magnitude.

Occlusion Computation As described above, we have

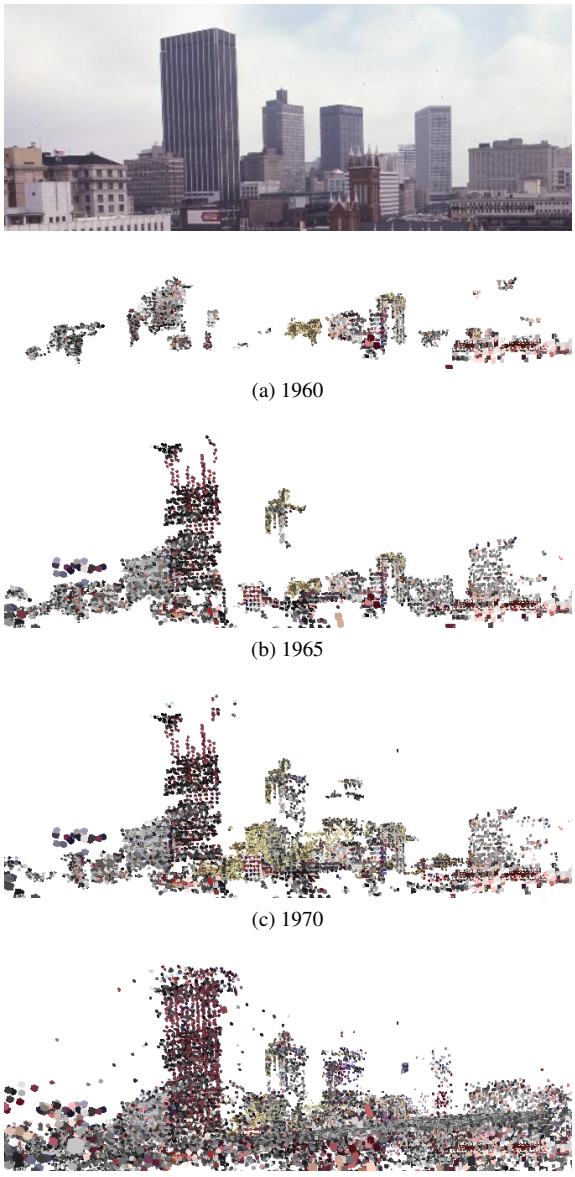
a list of 3D triangles associated with each object for occlusion purposes. Rather than explicitly computing ray-triangle intersections between each camera center and each structure point for every triangle in the occlusion geometry as in [11], we use an image-space approach. We first render a binary image for each object in each camera – despite the large number of rendered images ($m \times n$) this is a very fast operation either on the GPU or in software. Each image is white where the potentially occluding object's triangles project into the image and black everywhere else. By projecting each 3D structure point into each image, we can quickly detect potential occlusion events by examining the pixel color at the projected location of each point. If a point projects onto a white pixel, further depth tests are performed to determine occlusion, but in our experiments greater than 99.9% of points project onto black background pixels, which means no further tests are necessary, saving enormous computation. Note that we have computed point-object occlusion events. To compute object-object occlusion probabilities, we do the following: when an object k occludes any points belonging to another object i, the probability of occlusion $P(Occlusion_{ijk}|X,T)$ is equal to the fraction of object i points which were occluded by object k.

Having pre-computed all *potential* occlusion events in this way, at run time we use the current time parameter estimate T to determine which of these occlusions actually occur at the time of each image in the model. Importantly, using this *time-dependent occlusion* approach, we can not only explain away missing observations as in [11] but if an object is observed when the model indicates that it should be occluded, this provides strong evidence that the occluder itself should not exist at the present time.

5. Results

We perform temporal inference experiments on both synthetic and real data. For temporal priors, we use a normal distribution with $\sigma = 10.0$ if an image is "circa" some time, $\sigma = 1.0$ if given a year, $\sigma = 0.1$ if given a year and month, and $\sigma = 0.001$ if a full date is specified. The proposal density for MCMC is a normal distribution with $\sigma = 50$. In all experiments, we use point-grouping parameters $N_{group} = 1$ with threshold d_{group} depending upon each scene's arbitrarily scaled geometry.

For the synthetic scene, we have 100 images, taken over an 80 year period, observing 2112 3D points lying on the surface of 30 synthetic buildings. Of these 100 images, 33% have known date, 33% are "circa" some year, and 34% have completely unknown dates. The initial date for each image is, respectively, set to its known value, rounded to the nearest decade, or uniformly sampled between 1930 and 2010. We draw 20,000 samples of temporal parameters using MCMC, keeping the most probable sample, which reduces the root mean square error (over all images with respect to ground truth dates) from 19.31 years for the initial configuration to just 2.87 years for our solution.



(a) 1960

(b) 1965

(c) 1970

(d) All Points

Figure 5: Object Time Intervals. By performing temporal inference, we recover a time interval for every object in the scene. Here, we use these recovered time intervals to visualize the scene at different points in time (a)(b)(c) from the viewpoint of a given photograph. In contrast, the raw point cloud (d) resulting from SfM has no temporal information.

For the real scene, starting from a collection of 490 images of Atlanta dating from the 1930s to the 2000s, the result of SfM is a set of 102 images registered to 89,619 3D points and spanning the 1950s, 1960s, and 1970s (see Figure 1). We use the above point-grouping procedure to cre-

ate 3,749 objects from the original 89,619 points. We note that the largest reconstructed set of images was actually a set of 127 images all taken in the 2000s. Our images were not uniformly distributed across time, with a notable lack of images from the 1980s and 1990s which are not yet well-represented in either historical databases or online photo-sharing collections. We hypothesize that a denser sampling of images in both time and space would be required to link these reconstructions together.

For each image in our reconstruction, we initialized temporal parameters according to the historical date information accompanying the photographs and used the MCMC sampling procedure described above to arrive at the most probable temporal solution for the entire set of 102 images in the reconstruction. On a 2.33 GHz Intel Core 2 Duo, evaluating one sample takes 0.06 seconds, so we can evaluate 1000 samples per minute. The occlusion table itself takes on average 5.5 seconds per image, and is a one-time operation totaling less than 10 minutes for this dataset. Note that actual ground truth is difficult to achieve for this historical data – most images with missing dates have already been labeled by human experts to the best of their ability, and it is these very labels which are uncertain. Instead, we highlight a few illustrative examples (Figure 4) to demonstrate our method's effectiveness on real-world data:

- An image labeled "circa 1965" was moved to May 1971 in the most probable time configuration. Upon further inspection of the photograph's dozens of buildings, the image depicts a building completed in 1971, as well as buildings from 1968 and 1966.
- For an image originally dated 1868 (apparently a data entry error in the historical database with the intended date of 1968) the resulting date using our method was January 1969, much nearer to the truth.
- An image labeled 1967 was moved up to December of 1969. Upon examination, this image primarily depicts a building which began construction in 1969 and another building which was demolished in 1970. While we can confirm this using building construction records, our method is able to perform this reasoning from images alone.

After performing temporal inference on all image dates and object time intervals, we visualize the results (Figure 5) by choosing a point in time and rendering only those objects which exist at this time according to the recovered time intervals. When we view the 3D reconstruction from the same viewpoint but at different points in time, the successfully recovered time-varying structure becomes clear.

6. Conclusion

We have presented a general probabilistic temporal inference framework and applied it to a city-scale 3D reconstruction spanning multiple decades. In addition, we have demonstrated the first completely automatic method for image dating and recovery of time-varying structure from images. In future work, we hope to reconstruct vastly larger image sets spanning larger time periods, to employ more building-like object models, and to develop SfM techniques that explicitly deal with the unique problems of large changes in structure and appearance over time.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *Intl. Conf. on Computer Vision (ICCV)*, 2009. 1

[2] M. Pollefeys et al. Detailed real-time urban 3d reconstruction from video. *Int. J. Comput. Vision*, 78(2-3):143–167, 2008. 1

[3] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Intl. Conf. on Computer Vision (ICCV)*, 2009. 2

[4] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970. 5

[5] S.B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001. 2

[6] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *Intl. J. of Computer Vision*, 38(3):199–218, 2000. 2

[7] P. Messier. Notes on dating photographic paper. *Topics in Photograph Preservation*, 11, 2005. 2

[8] Halvor Moorshead. *Dating Old Photographs 1840-1929*. Moorshead Magazines Ltd, 2000. 1

[9] Robert Pols. *Family Photographs, 1860-1945: A Guide to Researching, Dating and Contextuallising Family Photographs*. Public Record Office Publications, 2002. 1

[10] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV*, 2003. 6

[11] G. Schindler, F. Dellaert, and S.B. Kang. Inferring temporal order of images from 3D structure. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2, 6, 7

[12] N. Snavely. Bundler: Structure from motion for unordered image collections. http://phototour.cs.washington.edu/bundler/. 6

[13] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006. 1

[14] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 6

[15] L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Eur. Conf. on Computer Vision (ECCV)*, 2008. 1