Int J Comput Vis (2010) 90: 283–303
DOI 10.1007/s11263-010-0341-y

Multi-view Occlusion Reasoning for Probabilistic Silhouette-Based Dynamic Scene Reconstruction

Li Guan · Jean-Sébastien Franco · Marc Pollefeys

Received: 22 July 2008 / Accepted: 8 April 2010 / Published online: 27 April 2010

Abstract In this paper, we present an algorithm to probabilistically estimate object shapes in a 3D dynamic scene using their silhouette information derived from multiple geometrically calibrated video camcorders. The scene is represented by a 3D volume. Every object in the scene is associated with a distinctive label to represent its existence at every voxel location. The label links together automatically-learned view-specific appearance models of the respective object, so as to avoid the photometric calibration of the cameras. Generative probabilistic sensor models can be derived by analyzing the dependencies between the sensor observations and object labels. Bayesian reasoning is then applied to achieve robust reconstruction against real-world environment challenges, such as lighting variations, changing background etc. Our main contribution is to explicitly model the visual occlusion process and show: (1) static objects (such as trees or lamp posts), as parts of the pre-learned background model, can be automatically recovered as a byproduct of the inference; (2) ambiguities due to inter-occlusion between multiple dynamic objects can be alleviated, and the final reconstruction quality is drastically improved. Several indoor and outdoor real-world datasets are evaluated to verify our framework.

Keywords Multi-view 3D reconstruction · Bayesian inference · Graphical model · Shape-from-silhouette · Occlusion

L. Guan (✉) · M. Pollefeys
UNC-Chapel Hill, Chapel Hill, USA
e-mail: lguan@cs.unc.edu

M. Pollefeys
e-mail: marc@cs.unc.edu

J.-S. Franco
LaBRI—INRIA Sud-Ouest, University of Bordeaux, Talence Cedex, France
e-mail: jean-sebastien.franco@labri.fr

M. Pollefeys
ETH-Zürich, Zürich, Switzerland
e-mail: marc.pollefeys@inf.ethz.ch

1 Introduction

3D shape reconstruction from real world imagery is an important research area in computer vision. In this paper, we focus on the problem of recovering a time-varying dynamic scene involving moving (and static) objects observed from multiple fix-positioned video streams with known geometric camera poses. This setup has been widely used in security surveillance, movies, medical surgery, sport broadcasting, digital 3D archiving, video games, etc.

There are mainly two categories of algorithms for such multi-view setups. The first is multi-view stereo/Shape from Photo-consistency (Kutulakos and Seitz 2000; Broadhurst et al. 2001; Scharstein and Szeliski 2002; Slabaugh et al. 2004; Seitz et al. 2006). They recover the surface of an object assuming its appearance is the same across views, so 3D surface points can be triangulated from multiple views. The output is usually a detailed surface model, because in theory object concavities can be recovered. However, in practice, many challenges exist. On the one hand, the "cross-view consistent appearance" assumption usually requires tedious radiometric calibration of the cameras. This is hard to realize in outdoor scenes without the constant illumination, which is required by most of the state-of-the-art radiometric calibration approaches (Ilie and Welsh 2005; Joshi et al. 2005; Takamatsu et al. 2008). In addition, limited camera field of view, motion blur, specular surfaces, object self-occlusion

and over-compression of the videos may all invalidate the consistency assumption. On the other hand, even the appearance is the same across views, the 3D point triangulation technique might as well fail in homogeneous regions (such as the shirts in Fig. 1(c), which are common in practical datasets), where no 2D feature point can be distinctively located.

The method we present in this paper fall in the second category—Shape from Silhouette methods (Matusik et al. 2000, 2001; Lazebnik et al. 2001; Franco and Boyer 2003), which usually depict the scene as *foreground* moving objects against known static *background*, and generally assume the silhouette of a foreground object in a camera view can be subtracted from the background, e.g. Fig. 1(d) and (e). Assume the camera network is geometrically calibrated beforehand, the back-projected silhouette cones intersect one another to form the *visual hull* (Baumgart 1974; Laurentini 1994), an approximate shape of the original object. Silhouette-based algorithms are relatively simple, fast, and output a global closed shape of the object. Therefore they are good choices for dynamic scene analysis. They also do not require object appearance to be similar across views, thus bypass the radiometric calibration of the camera network. And they are not affected by homogeneous regions of the objects either. For the above reasons, many state-of-the-art multi-view stereo approaches such as (Sinha and Pollefeys 2005; Furukawa and Ponce 2006) use a visual hull as initialization, or silhouette-based constraints.

However, Shape from Silhouette methods have their own caveats: most silhouette-based methods are highly dependent on appearance-based background modeling, which is usually sensitive to imaging sensor noise, shadows, illumination variations in the scene, etc. Also the background subtraction techniques are usually unstable when the modeled object has a similar appearance to the background. Therefore, silhouette-based 3D modeling techniques were usually used in a controlled, man-made environment, such as a turn-table setup or indoor laboratory. In order to extend these approaches in uncontrolled, natural environments, researchers have explored different possibilities to improve the robustness, such as adaptively updating the background models (Stauffer and Grimson 1999; Elgammal et al. 2002; Kim et al. 2005), using a discrete global optimization framework (Snow et al. 2000), proposing silhouette priors over multi-view sets (Grauman et al. 2003), and introducing a sensor fusion scheme to compute the probability of existence of the 3D shape (Franco and Boyer 2005).

There is one more challenge for silhouette-based methods to work in a general environment—occlusions, which can be categorized into three types: (1) *Self-occlusion*. It happens to every closed-surface object where a part of the object is blocking another part of itself. The lack of information in the occluded region introduces ambiguities, and is one of the main reasons why a visual hull is always larger than the real shape. Given a certain number of camera views, in the absence of further surface information, self-occlusions cannot always be handled because of silhouette ambiguities. In this paper, we mainly address the other two types of occlusions. (2) *Static occlusion*. It happens when a static object blocks a dynamic object with respect to a certain camera view, such as the sculpture blocking the person in Fig. 1(b). In this paper, we call the static object (the sculpture) a *static occluder* or simply an *occluder*, so as to differentiate from a dynamic subject, such as a person. As the sculpture in Fig. 1, occluders cannot always be removed from the scene in advance, causing their appearance to be included as part of the pre-learnt background model. When a dynamic object goes behind a static occluder, a static occlusion event happens. When subjects of interest move behind occluders with respect to a certain point of view, the colors perceived in that view correspond to the occluder and thus are identical to the colors learned in the background model. This results in apparently corrupted silhouettes, corresponding to the region of static occlusions, as in Fig. 1(d). Consequently, due to the intersection rule, such corrupted silhouettes result in an incomplete visual hull. This type of occlusion is specific to reconstruction approaches based on silhouettes extracted using static background modeling. (3) *Inter-occlusion*. Occlusions may also occur between two or more dynamic objects of interest, as shown in Fig. 1(c). With the increase of such occlusions, the discriminatory power of the silhouettes decreases, resulting in the reconstructed shapes much larger in volume than the real objects. In fact, when more dynamic objects cluster in the scene, the visibility ambiguity generally increases. This is discussed in detail in Sect. 2 (Fig. 3).

Both the static occlusion and inter-occlusion decrease the quality of the final reconstruction result, yet they are very common and almost unavoidable in everyday environments. If we plan to use silhouette-based methods in uncontrolled real-world scenes, we need to deal with both of them. The difference between static occlusion and inter-occlusion is that the static occluder's appearance is already part of the background model, but the dynamic objects' are not. This requires different consideration in the problem modeling.

In this paper, we explicitly model the static occlusion and inter-occlusion events in a volume representation of the reconstruction environment by analyzing the visibility relationships. We show that the shape of the static occluders can be recovered incrementally by accumulating occlusion cues from the motion of the dynamic objects. Also, by using a distinct per-view appearance model for each dynamic object, inter-occlusion and multi-object visibility ambiguities can be effectively solved, while avoiding the photometric calibration of the cameras. All the reasonings are performed in a probabilistic Bayesian sensor fusion framework, which builds on the probabilistic silhouette-based modeling

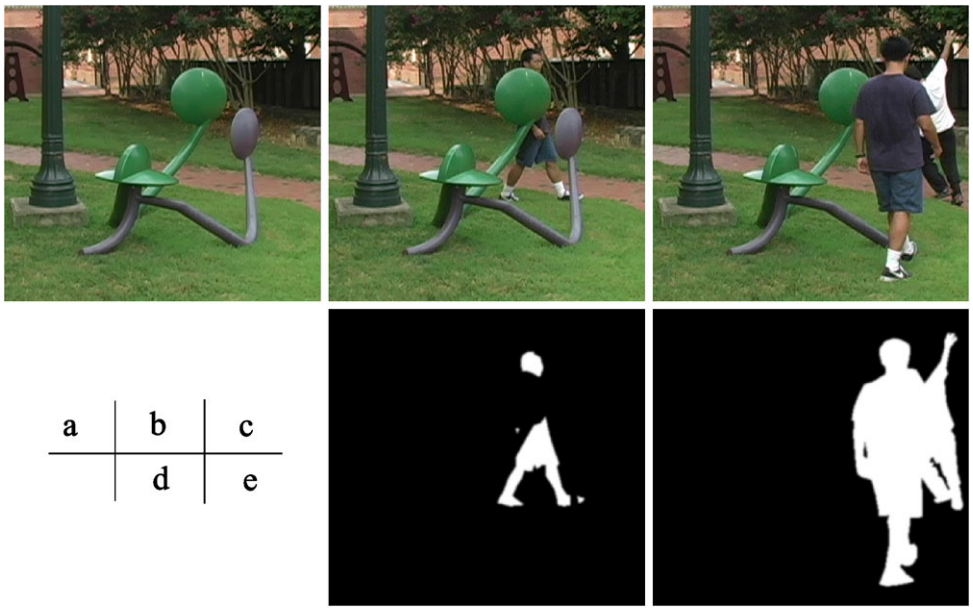

Fig. 1 The occlusion problem for a silhouette-based method. (**a**) A background view with a sculpture as an irremovable static visual occluder. (**b**) A person in the scene. (**c**) Two persons, one occluding the other. (**d**) Background subtraction silhouette for (**b**). (**e**) Background subtraction silhouette for (**c**)

(Franco and Boyer 2005) by introducing occlusion related terms. The major task is to compute the posterior probability for a given voxel to be part of a certain object shape, given multi-view observations. Our algorithm is verified against real datasets to be effective and robust in general indoor and outdoor environment of densely populated dynamic scenes with possible static occluders. We present the formulations of (Guan et al. 2007, 2008) in a more consistent way, analyze the theoretical properties of the recovered static occluder, discuss the drawbacks and propose some possible extensions of the framework.

2 Related Work and Overview

2.1 Static Occlusion

As shown in the last section, a static occlusion makes the silhouette incomplete, and thus has a negative impact over the silhouette-based modeling. Consequently, the inclusive property of the visual hull is no longer valid for models produced in this situation (Laurentini 1994): the real shape is no longer guaranteed to reside within the visual hull. Generally detecting and accounting for static occlusion has drawn much attention in areas such as depth layer extraction (Brostow and Essa 1999), occluding T-junction detection (Apostoloff and Fitzgibbon 2005), binary occluder mask extraction (Guan et al. 2006), and single image object boundary interpretation (Hoiem et al. 2007). All these works are limited to 2D image space.

Among papers regarding 3D occlusion, Favaro et al. (2003) uses sparse 3D occluding T-junctions as salient features to recover structure and motion. In De Bonet and Viola (1999), occlusions are implicitly modeled in the context of voxel coloring approaches, using an iterative scheme with semi-transparent voxels and multiple views of a scene from the same time instant. Our initial treatment of silhouette occlusions has lead to subsequent work designed to track objects from a small set of views (Keck and Davis 2008), with some differences in assumptions and modeling: they use iterative EM framework that at each frame first solving the voxel occupancy which is then fed back into the system to update the occlusion model. Also, for (Keck and Davis 2008), a hard threshold of silhouette information has to be provided during the initialization and the occluder information is maintained in a 4D (a 3D space volume per camera view) state space.

We represent the static occluder explicitly with a random variable at every location in the 3D scene. Theoretically occluder shapes can be accessed with careful reasoning about the visual hull of the incomplete silhouettes, as depicted in Fig. 2, which would lead to a deterministic algorithm to recover occluders. Let $\mathcal{S}^t$ be the set of incomplete silhouettes obtained at time t, and $\mathcal{VH}^t$ the incomplete visual hull obtained using these silhouettes. However $\mathcal{VH}^t$ is a region that is observed by all cameras as being both occupied by an object and unoccluded from any view. Thus we can deduce an entire region $\mathcal{R}^t$ of points in space that are free from any static occluder shape, as the shaded cones in Fig. 2(a). Formally, $\mathcal{R}^t$ is the set of points $X \in \mathbb{R}^3$ for which a view i exists, such that the viewing line of X from view i hits $\mathcal{VH}^t$ at a first visible point A_i, and $X \in O_i A_i$, with O_i the optical center of view i (Fig. 2(a)). This expresses the condition that X appears in front of the visual hull with respect to view i. The region $\mathcal{R}^t$ varies with t, thus assuming static occluders and broad coverage of the scene by dynamic ob-

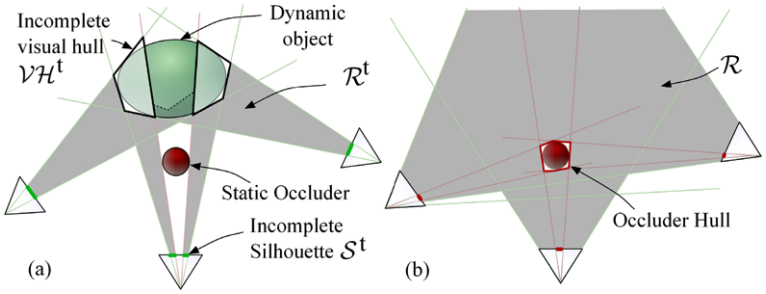

Fig. 2 Deterministic occlusion reasoning. (**a**) An occluder-free region $\mathcal{R}^t$ can be deduced from the incomplete visual hull at time t. (**b**) $\mathcal{R}$: occluder-free regions accumulated over time

ject motion, the free space in the scene can be deduced as the region $\mathcal{R} = \bigcup_{t=1}^{T} \mathcal{R}^t$. The shape of occluders, including concavities if they were covered by object motion, can be recovered as the complement of $\mathcal{R}$ in the common visibility region of all views (Fig. 2(b)).

However this deterministic approach would yield an impractical and non-robust solution, due to inherent silhouette extraction sensitivities to noise and corruption that contribute irreversibly to the result. It also suffers from the limitation that only portions of objects that are seen by all views can contribute to occlusion reasoning. Moreover, the scheme only accumulates *negative* information, where occluders are certain not to be. However *positive* information is also underlying to the problem: the discrepancies between the object's projection and the actual recorded silhouette would tell us where an occlusion event is positively happening, as long as we know where the object shape is, which the current silhouette-based method is able to provide (Franco and Boyer 2005). To lift these limitations and provide a robust solution, we propose a probabilistic approach to the static occlusion reasoning, in which both the negative and positive cues are fused and compete in a complementary way towards the static occluder shape estimation.

2.2 Multiple Dynamic Objects Inter-occlusion

Most of the existing silhouette-based reconstruction methods focus on mono-object situations, and fail to address the more general multi-object cases. When multiple dynamic objects are present in the scene, besides the inter-occlusion problem in Fig. 1(c) and (e), binary silhouettes and the resulting visual hull are not able to disambiguate regions actually occupied by dynamic objects from silhouette-consistent "ghost regions"—the empty regions that project inside all dynamic objects' silhouettes, which is depicted by the polygonal gray region indicated by arrows in Fig. 3(a). The ghost regions are increasingly likely as the number of observed objects rises, because it then becomes more difficult to find views that visually separate any two objects in the scene and carve out unoccupied regions of space.

The ghost regions have been analyzed in the context of people counting/tracking to avoid producing ghost tracks (Yang et al. 2003; Otsuka and Mukawa 2004). The method

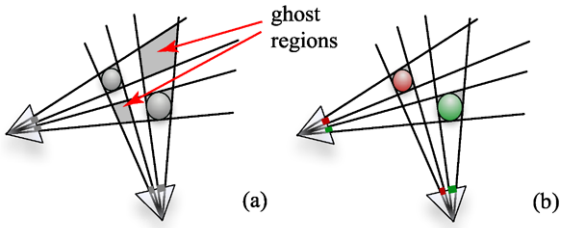

Fig. 3 The principle of multi-object silhouette reasoning for shape modeling disambiguation. (**a**) Ambiguous "ghost" regions in gray polygons, due to the binary silhouette back-projection does not have enough discriminability. (**b**) The ghost region ambiguities are reduced or eliminated by distinguishing between multiple objects' appearances

we propose casts the problem of silhouette modeling at the multi-object level, where ghosts can naturally be eliminated based on per object silhouette consistency. Multi-object silhouette reasoning has been applied in the context of multi-object tracking (Mittal and Davis 2003; Fleuret et al. 2007). The inter-occlusion problem has also been studied for the specific case of transparent objects (De Bonet and Viola 1999). Recent tracking efforts also use 2D probabilistic inter-occlusion reasoning to improve object localization (Gupta et al. 2007). But none of these methods are able to provide multiple probabilistic 3D shapes from silhouette cues as proposed here.

To address this problem, in addition to the background model learning, we also initialize a set of appearance models associated to every object in the scene. Given such extra information, the probability of the ghost regions can be reduced, because the set of silhouettes from different views that result into a ghost region are not drawn from consistent appearance models of any single object, as depicted in Fig. 3(b). Multiple silhouette labels have been introduced in a deterministic, purely geometric method (Ziegler et al. 2003), but this requires an arbitrary hard threshold for the number of views that define consistency. Moreover, silhouettes are assumed to be noiseless, which is violated or requires manual intervention for practical datasets. On the contrary, we propose a fully automatic framework. Similar to static occlusion formulation, using a volumetric representation of the 3D scene, we process multi-object sequences by examining the noisy causal relationship between every

voxel and the corresponding pixels in all camera views using a Bayesian formulation.

In particular, every voxel is modeled as a random variable. It can take any one of the m states representing the m possible objects in the scene. Given the knowledge that a voxel is occupied by a certain object, the camera sensor model explains what appearance distributions are supposed to be observed. This framework is able to explicitly model inter-occlusion with other objects, and estimate a window of object locations. The voxel sensor model semantics and simplifications are borrowed from the occupancy grid framework in robotics (Elfes 1989; Margaritis and Thrun 1998). The proposed method is naturally combined with the static occluder incremental recovery, because as mentioned before, the occluder is nothing but another state of a voxel. This scheme enables us to perform silhouette inference (Sect. 3.2) in a way that reinforces regions of space which are drawn from the same conjunction of color distributions, corresponding to one object, and penalizes appearance inconsistent regions, while accounting for object visibility.

In the rest of this paper, we first introduce the fundamental probabilistic sensor fusion framework and the detailed formulations in Sect. 3. We then describe problems related to building an automatic dynamic scene analysis system in Sect. 4. Specifically, we discuss how to initialize the appearance models and keep track the motion and status of each dynamic object. Section 5 shows the results of the proposed system and algorithm on real-world datasets. Despite the challenges in the datasets, such as lighting variation, shadows, background motion, reflection, dense population, drastic color inconsistency between views, etc, our system produces high quality reconstructions. Section 6 analyzes the advantages and limitations of this framework and compares the two types of occlusions in more depth, and draws the future picture.

3 Probabilistic Framework

Given the complete set of symbols listed in Table 1, we can define our problem formally: at a specific time instant, given a set of geometrically calibrated and temporally synchronized video frames $\mathcal{I}$ from n cameras, we infer for every discretized location X in a 3D occupancy volume grid its probability of being $\mathcal{L} \in \{\emptyset, \mathcal{O}, 1, \ldots, m, \mathcal{U}\}$. This means a voxel could be empty (denoted by $\emptyset$), occupied by a static occluder (denoted by $\mathcal{O}$), or by one of the m objects currently in the scene, which are of known appearance models. Last but not the least, one more label $\mathcal{U}$ could be assigned, for unidentified objects. It acts as a default label to capture all objects that are detected as different than background but not explicitly modeled by other labels. It is useful and effective for automatic detection of new objects coming into the scene, whose appearance has not yet been learned (Sect. 4.3).

Table 1 Notations of the multi-view system

n	number of cameras
m	number of dynamic objects
i	camera index
X	3D location, in the occupancy grid
x_i	pixel at camera view i corresponding to the voxel X
$\bar{l}_i$	viewing line of X to view i
$\hat{X}_i$	3D location, on the viewing ray of X, and in front of X with respect to view i
$\check{X}_i$	3D location, on the viewing ray of X, and behind X with respect to view i
$\mathcal{L}$	voxel labels
$\emptyset$	empty space label
$\mathcal{G}$	dynamic object label
$\mathcal{U}$	label for a newcoming dynamic object, whose appearance has not been learnt
$\mathcal{O}$	static occluder label
$\mathcal{I}_i^t$	image from camera i at time t
$\mathcal{B}_i$	camera i's background model
$\mathcal{C}_i^m$	dynamic object m's appearance model in view i
$\mathcal{S}$	silhouette formation hidden variable

Theoretically, the problem is to compute the posterior probability from the camera observations; but it is not easy in practice. Because the estimation of a voxel's state involves modeling dependencies with all other voxels on its viewing lines with respect to all cameras. Given the huge state space, i.e. the solid 3D volume, and multiple state labels of different objects, it is impossible to enumerate all state configurations to find the one with the highest probability. People have encountered similar problems and proposed solutions that suppress the 3D volume state space into 2D ground plane (Mittal and Davis 2003) and then solve the global solution as an offline process (Fleuret et al. 2007). However, since we want to recover the full 3D information for dynamic scenes, and to keep the potential of real-time processing, the previous proposals are not satisfactory. Instead, we borrow the iterative idea of an EM framework (Keck and Davis 2008) and break the estimation into two steps: for every time instant, we first estimate the occupancy probabilities for each of the individual dynamic object from silhouette information using a Bayesian formulation; then we estimate the inter-occlusion as well as static occlusion in a second pass. Although refinements can be achieved by doing iterations over this two-step solution, we demonstrate with real datasets that the shape estimation is already good for a single iteration of the two-step process.

In our formulation we address the inference of both static occluders and multiple dynamic objects in one scene.

Our treatment of static occluder is indifferent to the number of dynamic objects in the scene. To keep notations uncluttered we will present the occluder inference framework in the context of only one undiscriminated dynamic object label $\mathcal{G}$ with only two states $\mathcal{G} \in \{0, 1\}$ (Sect. 3.1). We will then specifically explain how to model multi-object inter-occlusions in Sect. 3.2. In the latter we leave out occluder inference for clarity, although we later show how to perform both tasks simultaneously.

3.1 Static Occluder

In this section, to introduce the static occluder formulation, for simplicity, we assume only one dynamic object is in the scene. In the result section Sect. 5, we show that our occlusion modeling technique also applies to multiple dynamic objects. Let a binary variable $\mathcal{G}$ denote the single dynamic object in the scene at voxel X, namely $\mathcal{G} = 1$ means the voxel is occupied by the object, and $\mathcal{G} = 0$ denotes it is not. The occluder occupancy state can also be expressed using the binary label $\mathcal{O}$. $\mathcal{O} = 1$ occupied, and $\mathcal{O} = 0$ not. Notice that the static occluder state $\mathcal{O}$ is assumed to be fixed, while the dynamic object state $\mathcal{G}$ varies over time $t \in \{1, \ldots, T\}$, where T denotes the time instant of the last available frame so far. The dynamic object occupancy of voxel X at time t is expressed by a $\mathcal{G}^t$. As shown in Fig. 4(a), the regions of interest to infer the probabilities of both $\mathcal{G}$ and $\mathcal{O}$ are on the viewing lines $\bar{l}_i$, $i \in \{1, \ldots, n\}$ from the camera centers through X. The voxel X projects to n image pixels $x_i, i \in \{1, \ldots, n\}$, whose color observed at time t in view i is expressed by the variable $\mathcal{I}_i^t$. We assume that background images, which are generally static, were pre-recorded free of dynamic objects, and that the appearance and variability of background colors for pixels x_i has been modeled using a set of parameters $\mathcal{B}_i$. Such observations can be used to infer the probability of dynamic object occupancy in the absence of static occluders. The problem of recovering occluder occupancy is more complex because it requires modeling interactions between voxels on the same viewing lines. Relevant statistical variables are shown in Fig. 4(b).

3.1.1 Viewing Line Modeling

Because of potential mutual occlusions, to infer $\mathcal{O}$, one must account for other occupancies along the viewing lines of X. Static occluder or dynamic shapes can be present along the same viewing line, leading to different image formations at the camera view i. Accounting for all the combinatorial number of possibilities for voxel states along X's viewing line is neither necessary nor meaningful: first because occupancies of neighboring voxels are fundamentally correlated to the presence or the absence of a single common object, second because the main useful information one needs to make occlusion decisions about X is to know whether something is in front of it or behind it, regardless of the exact locations along the viewing line.

With this in mind, the pixel observation at x_i with respect to a certain X's viewing line $\bar{l}_i$ can be described with three components: the state of X itself, the state of occlusion of X by anything in front, and the state of what is at the back of X. And the front and back components are modeled by extracting the two most influential modes in front of and behind X. Specifically, the locations of the two modes are given by two voxels $\hat{X}_i^t$ and $\check{X}_i^t$. We select $\hat{X}_i^t$ as the voxel at time t that most contributes to the belief that X is obstructed by a dynamic object along $\bar{l}_i$, and $\check{X}_i^t$ as the voxel most likely to be occupied by a dynamic object behind X on $\bar{l}_i$ at time t. With this three-component modeling, comes a number of related statistical variables illustrated in Fig. 4(b). The occupancy of voxels $\hat{X}_i^t$ and $\check{X}_i^t$ by the visual hull of a dynamic object at time t on $\bar{l}_i$ is expressed by two binary state variables, respectively $\hat{\mathcal{G}}_i^t$ and $\check{\mathcal{G}}_i^t$. Two binary state variables $\hat{\mathcal{O}}_i^t$ and $\check{\mathcal{O}}_i^t$ express the presence or absence of an occluder at voxels $\hat{X}_i^t$ and $\check{X}_i^t$ respectively.

Note the difference in semantics between the two variable groups $\hat{\mathcal{G}}_i^t$, $\check{\mathcal{G}}_i^t$ and $\hat{\mathcal{O}}_i^t$, $\check{\mathcal{O}}_i^t$. The former designates dynamic shape visual hull occupancies of different time instants and chosen positions, while the latter expresses the static occluder occupancies. The locations of $\hat{\mathcal{G}}_i^t$ and $\check{\mathcal{G}}_i^t$ at different times are different, because the dynamic shape may have moved over the time; while $\hat{\mathcal{O}}_i^t$ and $\check{\mathcal{O}}_i^t$ are always associated with the same locations as $\hat{\mathcal{G}}_i^t$ and $\check{\mathcal{G}}_i^t$, for the purpose of our simplified viewing line state enumeration scheme. All the aforementioned states need to be considered because they dependently influence the occupancy inference at X. From the image formation perspective, by varying the states of $\hat{\mathcal{G}}_i^t$, $\check{\mathcal{G}}_i^t$, $\hat{\mathcal{O}}_i^t$, $\check{\mathcal{O}}_i^t$, $\mathcal{G}$ and $\mathcal{O}$ (the latter are the dynamic object and occluder states at the voxel location X), it would form different image pixel values at x_i. For legibility, we occasionally refer to the conjunction of a group of variables by dropping the indices and exponents, e.g. $\mathcal{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$.

3.1.2 Joint Distribution

We now explain the dependencies between the problem variables to simplify the their joint probability distribution. An intuitive assumption is that different views can be independently predicted without the knowledge of other views, given the knowledge about the scene $\mathcal{G}$ and $\mathcal{O}$. The background model for one view can be independently trained. A second assumption is that space occupancy variables at X depend only on the information along optic rays that go through X, which may include not just the single pixel that the voxel is projected onto, but a 2D neighborhood of pixels around the voxel's projection. We assume that the viewing line variables are sufficient to model the dependencies

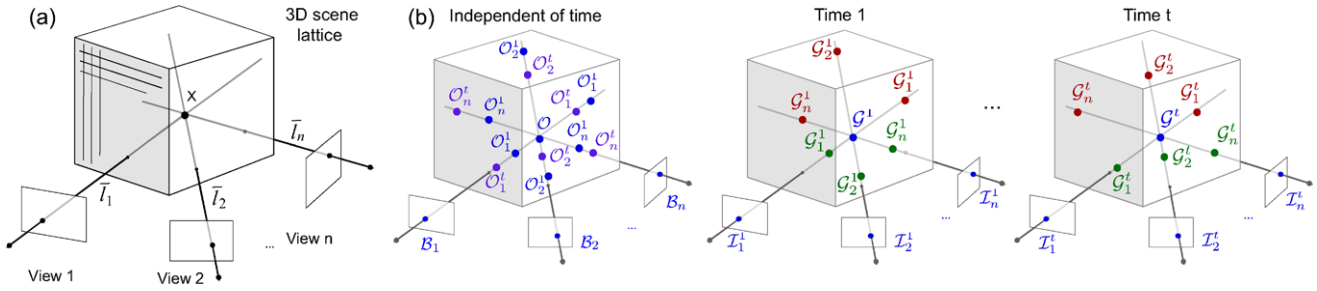

Fig. 4 Problem overview. (**a**) Geometric context of voxel X. (**b**) Main statistical variables used to infer the occluder occupancy probability of X. $\mathcal{G}^t$, $\hat{\mathcal{G}}_i^t$, $\check{\mathcal{G}}_i^t$: dynamic object occupancies at relevant voxels at, in front of, behind X respectively. $\mathcal{O}$, $\hat{\mathcal{O}}_i^t$, $\check{\mathcal{O}}_i^t$: static occluder occupancies at, in front of, behind X. $\mathcal{I}_i^t$, $\mathcal{B}_i$: colors and background color models observed where X projects in images

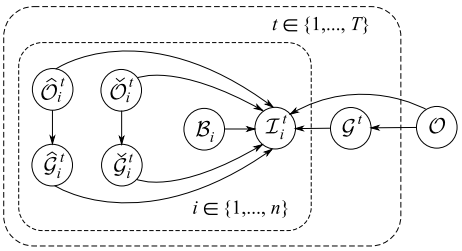

Fig. 5 The dependency graph for the static occluder inference at voxel X. $\mathcal{O}$ and $\mathcal{G}^t$ are the occluder occupancy and dynamic object occupancy state at time t at location X. Notice that the background $\mathcal{B}_i$ is assumed to be only dependent on the view but constant over time; while $\hat{\mathcal{O}}^t$ and $\check{\mathcal{O}}^t$ are at different locations at different times for X, though $\mathcal{O}$ itself is not a function of time

between a voxel X and other voxels on its viewing lines. This assumption allows us to use the common silhouette method simplification which consists in independently computing probabilities of each voxel X. This avoids the highly complex problem of updating the full grid state $\mathcal{O}$ while simultaneously accounting for viewing line dependencies. Besides, this assumption is reasonable because it is similar to deterministic volumetric visual hull algorithms, where every voxel's status is evaluated individually against its projections onto image pixels. Results show that independent estimation, while not as exhaustive as a global search over all voxel configurations, still provides very robust and usable information, at a much lower cost.

We now describe the noisy interactions between the variables considered, through the decomposition of their joint distribution $p(\mathcal{O}, \mathcal{G}, \hat{\mathcal{O}}, \hat{\mathcal{G}}, \check{\mathcal{O}}, \check{\mathcal{G}}, \mathcal{I}, \mathcal{B})$. Given the variable dependency graph shown in Fig. 5, we propose:

$$p(\mathcal{O}) \prod_{t=1}^{T} p(\mathcal{G}^t \mid \mathcal{O}) \prod_{i=1}^{n} p(\hat{\mathcal{O}}_i^t) p(\hat{\mathcal{G}}_i^t \mid \hat{\mathcal{O}}_i^t) p(\check{\mathcal{O}}_i^t) p(\check{\mathcal{G}}_i^t \mid \check{\mathcal{O}}_i^t) \times p(\mathcal{I}_i^t \mid \hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t, \mathcal{B}_i). \quad (1)$$

$p(\mathcal{O})$, $p(\hat{\mathcal{O}}_i^t)$ and $p(\check{\mathcal{O}}_i^t)$ are priors of occluder occupancy. We set them to a single constant distribution $\mathcal{P}_o$ which reflects the expected ratio between occluder voxels and non-occluder voxels in a scene. No particular region of space is to be favored *a priori*.

$p(\mathcal{G}^t \mid \mathcal{O})$, $p(\hat{\mathcal{G}}_i^t \mid \hat{\mathcal{O}}_i^t)$, $p(\check{\mathcal{G}}_i^t \mid \check{\mathcal{O}}_i^t)$ are priors of dynamic visual hull occupancy with identical semantics. This choice of terms reflects the following modeling decisions. First, the dynamic visual hull occupancies involved are considered independent of one another as they synthesize the information of three distinct regions for each viewing line. However they depend upon the knowledge of occluder occupancy at the corresponding voxel position, because occluder and dynamic object occupancies are mutually exclusive at a given scene location. Importantly however, because we only use silhouette cues, we do not have direct access to dynamic object occupancies but to the occupancies of its visual hull. Fortunately this ambiguity can be adequately modeled in a Bayesian framework, by introducing a local hidden variable $\mathcal{H}$ expressing the correlation between dynamic and occluder occupancy:

$$p(\mathcal{G}^t \mid \mathcal{O}) = \sum_{\mathcal{H}} p(\mathcal{H}) p(\mathcal{G}^t \mid \mathcal{H}, \mathcal{O}). \quad (2)$$

We set $p(\mathcal{H} = 1) = \mathcal{P}_c$ using a constant expressing our prior belief about the correlation between visual hull and occluder occupancy. The prior $p(\mathcal{G}^t \mid \mathcal{H}, \mathcal{O})$ explains what we expect to know about $\mathcal{G}^t$ given the state of $\mathcal{H}$ and $\mathcal{O}$:

$$p(\mathcal{G}^t = 1 \mid \mathcal{H} = 0, \mathcal{O} = \omega) = \mathcal{P}_{\mathcal{G}_t} \quad \forall \omega, \quad (3)$$

$$p(\mathcal{G}^t = 1 \mid \mathcal{H} = 1, \mathcal{O} = 0) = \mathcal{P}_{\mathcal{G}_t}, \quad (4)$$

$$p(\mathcal{G}^t = 1 \mid \mathcal{H} = 1, \mathcal{O} = 1) = \mathcal{P}_{go}, \quad (5)$$

with $\mathcal{P}_{\mathcal{G}_t}$ the prior dynamic object occupancy probability as computed independently of occlusions (Franco and Boyer 2005), and $\mathcal{P}_{go}$ set close to 0, expressing that it is unlikely that the voxel is occupied by dynamic object visual hulls when the voxel is known to be occupied by an occluder and both dynamic and occluder occupancy are known to be strongly correlated (5). The probability of visual hull occupancy is given by the previously computed occupancy prior,

in case of non-correlation (3), or when the states are correlated but occluder occupancy is known to be empty (4).

3.1.3 Image Sensor Model

We choose the sensor model $p(\mathcal{I}_i^t \mid \hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t, \mathcal{B}_i)$ in (1) to be governed by a hidden local per-pixel process $\mathcal{S}$. The binary variable $\mathcal{S}$ represents the hidden silhouette detection state (0 or 1) at this pixel. It is unobserved information and can be marginalized, given an adequate split into two subterms:

$$p(\mathcal{I}_i^t \mid \hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t, \mathcal{B}_i) = \sum_{\mathcal{S}} p(\mathcal{I}_i^t \mid \mathcal{S}, \mathcal{B}_i)\, p(\mathcal{S} \mid \hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t). \quad (6)$$

$p(\mathcal{I}_i^t \mid \mathcal{S}, \mathcal{B}_i)$ indicates what color distribution we expect to observe given the knowledge of silhouette detection and background color model at this pixel. When $\mathcal{S} = 0$, the silhouette is undetected and thus the color distribution is dictated by the pre-observed background model $\mathcal{B}_i$ (considered Gaussian in our experiments). When $\mathcal{S} = 1$, a dynamic object's silhouette is detected, in which case our knowledge of color is limited, thus we use a uniform distribution in this case, favoring no dynamic object color *a priori*.

$p(\mathcal{S} \mid \hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t)$ is the second part of our sensor model, which explicits what silhouette state is expected to be observed given the three dominant occupancy state variables of the corresponding viewing line. Since these are encountered in the order of visibility $\hat{X}_i^t, X, \check{X}_i^t$, the following relations hold:

$$\begin{aligned} &p(\mathcal{S} \mid \{\hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t\} \\ &\quad = \{o, g, \mathrm{k}, \mathrm{l}, \mathrm{m}, \mathrm{n}\}, \mathcal{B}_i) \\ &\quad = p(\mathcal{S} \mid \{\hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t\} = \{0, 0, o, g, \mathrm{p}, \mathrm{q}\}, \mathcal{B}_i) \\ &\quad = p(\mathcal{S} \mid \{\hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t\} = \{0, 0, 0, 0, o, g\}, \mathcal{B}_i) \\ &\quad = P_S(\mathcal{S} \mid o, g) \quad \forall (o, g) \neq (0, 0)\ \forall (\mathrm{k}, \mathrm{l}, \mathrm{m}, \mathrm{n}, \mathrm{p}, \mathrm{q}). \end{aligned} \quad (7)$$

These expressions convey two characteristics. First, that the form of this distribution is given by the first non-empty occupancy component in the order of visibility, regardless of what is behind this component on the viewing line. Second, that the form of the first non-empty component is given by an identical sensor prior $P_S(\mathcal{S} \mid o, g)$. We set the four parametric distributions of $P_S(\mathcal{S} \mid o, g)$ as following:

$$P_S(\mathcal{S} = 1 \mid 0, 0) = \mathcal{P}_{fa} \qquad P_S(\mathcal{S} = 1 \mid 1, 0) = \mathcal{P}_{fa}, \quad (8)$$

$$P_S(\mathcal{S} = 1 \mid 0, 1) = \mathcal{P}_d \qquad P_S(\mathcal{S} = 1 \mid 1, 1) = 0.5, \quad (9)$$

where $\mathcal{P}_{fa} \in [0, 1]$ and $\mathcal{P}_d \in [0, 1]$ are constants expressing the prior probability of *false alarm* and the probability of *detection*, respectively. They can be chosen once for all datasets as the method is not sensitive to the exact value of these priors. Meaningful values for $\mathcal{P}_{fa}$ are close to 0, while $\mathcal{P}_d$ is generally close to 1. Equation (8) expresses the cases where no silhouette is expected to be detected in images, i.e. either when there are no objects at all on the viewing line, or when the first encountered object is a static occluder, respectively. Equation (9) expresses two distinct cases. First, the case where a dynamic object's visual hull is encountered on the viewing line, in which case we expect to detect a silhouette at the matching pixel. Second, the case where both an occluder and dynamic visual hull are present at the first non-free voxel. This is perfectly possible, because the visual hull is an overestimate of the true dynamic object shape. While the true shape of objects and occluders are naturally mutually exclusive, the visual hull of dynamic objects can overlap with occluder voxels. In this case we set the distribution to uniform, because the silhouette detection state cannot be predicted: it can be caused by shadows casted by dynamic objects on occluders in the scene, and noise.

3.1.4 Static Occluder Occupancy Inference

Estimating the occluder occupancy at a voxel translates to estimating $p(\mathcal{O} \mid \mathcal{I}, \mathcal{B})$ in Bayesian terms. Applying Bayes rule to the modeled joint probability (1) leads to the following expression, once hidden variable sums are decomposed to factor out terms not required at each level of the sum:

$$p(\mathcal{O} \mid \mathcal{I}, \mathcal{B}) = \frac{1}{z} p(\mathcal{O}) \prod_{t=1}^{T} \left(\sum_{\mathcal{G}_t} p(\mathcal{G}^t \mid \mathcal{O}) \left(\prod_{i=1}^{n} \mathcal{P}_i^t \right) \right), \quad (10)$$

where

$$\mathcal{P}_i^t = \sum_{\check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t} p(\check{\mathcal{O}}_i^t)\, p(\check{\mathcal{G}}_i^t \mid \check{\mathcal{O}}_i^t) \sum_{\hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t} p(\hat{\mathcal{O}}_i^t)\, p(\hat{\mathcal{G}}_i^t \mid \hat{\mathcal{O}}_i^t) \times p(\mathcal{I}_i^t \mid \hat{\mathcal{O}}_i^t, \hat{\mathcal{G}}_i^t, \mathcal{O}, \mathcal{G}^t, \check{\mathcal{O}}_i^t, \check{\mathcal{G}}_i^t, \mathcal{B}_i). \quad (11)$$

$\mathcal{P}_i^t$ expresses the contribution of view i at a time t. The formulation therefore expresses Bayesian fusion over the various observed time instants and available views, with marginalization over unknown viewing line states (10). The normalization constant z is easily obtained by ensuring summation to 1 of the distribution.

3.1.5 Online Incremental Computation

With the formulation of the previous sections, the static occluder probability is computed by considering all the occlusion events (between the dynamic shape and the static occluder) that have happened up to the current frame. It is an online process. However, it has a subtle problem: for a given voxel location which is supposed to be free of occupancy,

the occluder probability may be high just because an occlusion event has happened along the viewing line somewhere behind the voxel (a real occluder is behind the voxel) with respect to the camera, as shown in Fig. 9. And as the figure shows, it is only likely to happen at the beginning of the videos when too little information has been collected. The voxel's occluder probability would eventually drop to a reasonable near zero value when more evidences have shown over time that since this voxel is not blocking the dynamic shape behind it (maybe from all other views), it is more likely to be an empty voxel.

There is a way however to detect this bias in the voxel probability estimation. If we take a second look at the problem, this early decision is made from evidences that come from only a few camera views—either because those views's geometric calibration errors are larger than others, or because there happen to be some real static occluders in those views behind the misjudged voxel. Intuitively, a voxel X's probability estimation becomes more reliable as its occlusion information is confirmed from more views, i.e. when a dynamic object has passed behind X in a more views.

We thus introduce a measure of observability and trustworthiness of a voxel's estimation: the reliability R of a voxel at a certain time instant. Specifically, we model the intuition that voxels whose occlusion cues arise from an abnormally low number of views should not be trusted. Since this involves all cameras and their observations jointly, the inclusion of this constraint in our initial model would break the symmetry in the inference formulated in (10) and defeat the possibility for online updates. Instead, we opt to use a second criterion in the form of a reliability measure $R \in [0, 1]$. Small values indicate poor coverage of dynamic objects, while large values indicate sufficient cue accumulation. We define reliability using the following expression:

$$R = \frac{1}{n}\sum_{i=1}^{n} \max_{t} (1 - \mathcal{P}_{\hat{\mathcal{G}}_i^t})\mathcal{P}_{\check{\mathcal{G}}_i^t} \tag{12}$$

with $\mathcal{P}_{\hat{\mathcal{G}}_i^t}$ and $\mathcal{P}_{\check{\mathcal{G}}_i^t}$ the prior probabilities of dynamic visual hull occupancy. R examines, for each camera i, the maximum occurrence across the examined time sequence of X to be both unobstructed and in front of a dynamic object. This determines how well a given view i was able to contribute to the estimation across the sequence. R then averages these values across views, to measure the overall quality of observation, and underlying coverage of dynamic object motion for the purpose of occlusion inference.

The reliability R is not a probability, but an indicator. It can be used online in conjunction to the occlusion probability estimation to evaluate a conservative occluder shape at all times, by only considering voxels for which R exceeds a certain quality threshold. As shown in Sect. 5.1.1, it can be used to reduce the sensitivity to noise in regions of space that have only been observed marginally.

3.1.6 *Accounting the Recovered Occluder*

As more data becomes available and reliable, the results of occluder estimation can be accounted for when inferring the occupancies of dynamic objects. This translates to the evaluation of $p(\mathcal{G}^\tau \mid \mathcal{I}^\tau, \mathcal{B})$ for a given voxel X and time τ. The occlusion information obtained can be included as a prior in dynamic object inference, by adequately modifying the existing probabilistic framework (Franco and Boyer 2005), leading to the following simplified joint probability distribution:

$$p(\mathcal{O})p(\mathcal{G}^\tau \mid \mathcal{O}) \prod_{i=1}^{n} p(\hat{\mathcal{O}}_i^\tau)p(\hat{\mathcal{G}}_i^\tau \mid \hat{\mathcal{O}}_i^\tau) \times p(\mathcal{I}_i^\tau \mid \hat{\mathcal{O}}_i^\tau, \hat{\mathcal{G}}_i^\tau, \mathcal{O}, \mathcal{G}^\tau, \mathcal{B}_i),$$

where $\mathcal{G}^\tau$ and $\mathcal{O}$ are the dynamic and occluder occupancy at the inferred voxel, $\hat{\mathcal{O}}_i^\tau$, $\hat{\mathcal{G}}_i^\tau$ the variables matching the most influential static occluder component along $\bar{l}_i$ *in front of* X. This component is selected as the voxel whose prior of being occupied is maximal, as computed to date by occlusion inference. In this inference, there is no need to consider voxels behind X, because knowledge about their occlusion occupancy has no influence on the dynamic object occupancy state of X.

The parametric forms of this distribution have identical semantics as Sect. 3.1.2 but different assignments because of the nature of the inference. Naturally no prior information about dynamic occupancy is assumed here. $p(\mathcal{O})$ and $p(\hat{\mathcal{O}}_i^\tau)$ are set using the result to date of expression (10) at their respective voxels, as prior. $p(\mathcal{G}^\tau \mid \mathcal{O})$ and $p(\hat{\mathcal{G}}_i^\tau \mid \hat{\mathcal{O}}_i^\tau)$ are constant: $p(\mathcal{G}^\tau = 1 \mid \mathcal{O} = 0) = 0.5$ expresses a uniform prior for dynamic objects when the voxel is known to be occluder free. $p(\mathcal{G}^\tau = 1 \mid \mathcal{O} = 1) = \mathcal{P}_{go}$ expresses a low prior of dynamic visual hull occupancy given the knowledge of occluder occupancy, as in (5). The term $p(\mathcal{I}_i^\tau \mid \hat{\mathcal{O}}_i^\tau, \hat{\mathcal{G}}_i^\tau, \mathcal{O}, \mathcal{G}^\tau, \mathcal{B}_i)$ is set identical to (7), only stripped of the influence of $\check{\mathcal{O}}_i^\tau, \check{\mathcal{G}}_i^\tau$.

3.2 Multiple Dynamic Objects

In this section, we focus on the inference of multiple dynamic objects. Since a dynamic object changes shape and location constantly, our dynamic object reconstruction has to be computed for every frame in time, and there is no way to accumulate the information over time as we did for the static occluder. We assume static occlusion is computed in an independent thread and can be used as prior in this inference. We thus focus here on the multi-object problem occurring at one time instant t. We introduce new notations to account for up to m dynamic objects of interest, in a scene observed by n calibrated cameras. Occupancies $\mathcal{G}$ at a given voxel X need

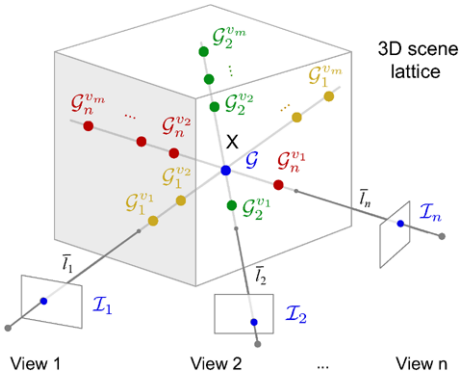

Fig. 6 Overview of main statistical variables and geometry of the problem. $\mathcal{G}$ is the occupancy at voxel X and lives in a state space $\mathcal{L}$ of object labels. $\{\mathcal{I}_i\}$ are the color states observed at the n pixels where X projects. $\{\mathcal{G}_i^{v_j}\}$ are the states in $\mathcal{L}$ of the most likely obstructing voxels on the viewing line, for each of the m objects, enumerated in their order of visibility $\{v_j\}_i$

now to be defined over an extended set of $m+2$ labels (described in the following section) rather than $\{0, 1\}$ to model occupancy distributions over several objects. We now also assume some prior knowledge about scene state is available for each voxel X in the lattice and can be used in the inference. Various uses of this assumption will be demonstrated in Sect. 4. Let us revisit the number of statistical variables used to model the scene state, the image generation process and to infer $\mathcal{G}$, as depicted in Fig. 6.

3.2.1 Statistical Variables

Scene Voxel State Space The occupancy state of X is represented by $\mathcal{G} \in \mathcal{L}$, where $\mathcal{L}$ is a set of labels $\{\emptyset, 1, \ldots, m, \mathcal{U}\}$. A voxel is either empty ($\emptyset$), one of m objects the model is keeping track of (numerical labels), or occupied by an unidentified object ($\mathcal{U}$). $\mathcal{U}$ is intended to act as a default label capturing all objects that are detected as different than background but not explicitly modeled by other labels, which proves useful for automatic detection of new objects (Sect. 4.3).

Observed Appearance The voxel X projects to a set of pixels, whose colors $\mathcal{I}_i$, $i \in 1, \ldots, n$ we observe in images. We assume these colors are drawn from a set of object and view specific color models whose parameters we note $\mathcal{C}_i^l$. More complex appearance models are possible using gradient or texture information, without loss of generality.

Latent Viewing Line Variables To account for inter-object occlusion, we need to model the contents of viewing lines and how it contributes to image formation. We assume some *a priori* knowledge about where objects lie in the scene. The presence of such objects can have an impact on the inference of $\mathcal{G}$ because of the visibility of objects and how they affect $\mathcal{G}$. Intuitively, conclusive information about $\mathcal{G}$ cannot be

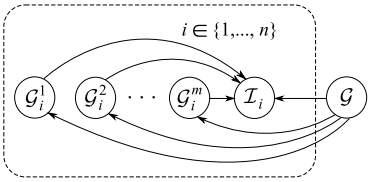

Fig. 7 The dependency graph for the dynamic object inference at voxel X, assuming m dynamic objects in the scene and the probability for X to be other labels are known. The background model for each view $\mathcal{B}$, the color model of each object for each view $\mathcal{C}$ and the static occluder $\mathcal{O}$ are not drawn for clarity

obtained from a view i if a voxel in front of $\mathcal{G}$ with respect to i is occupied by another object, for example. However, $\mathcal{G}$ directly influences the color observed if it is unoccluded and occupied by one of the objects. But if $\mathcal{G}$ is known to be empty, then the color observed at pixel $\mathcal{I}_i$ reflects the appearance of objects behind X in image i, if any. These visibility intuitions are modeled below (Sect. 3.2.2).

It is again not meaningful to account for the combinatorial number of occupancy possibilities along the viewing rays of X. This is because neighboring voxel occupancies on the viewing line usually reflect the presence of the same object and are therefore correlated. In fact, assuming we witness no more than one instance of every one of the m objects along the viewing line, the fundamental information that is required to reason about X is the knowledge of presence and ordering of the objects along this line. To represent this knowledge, as depicted in Fig. 6, assuming prior information about occupancies is already available at each voxel, we extract, for each label $l \in \mathcal{L}$ and each viewing line $i \in \{1, \ldots, n\}$, the voxel whose probability of occupancy is dominant for that label on the viewing line. This corresponds to electing the voxels which best represent the m objects and have the most influence on the inference of $\mathcal{G}$. We then account for this knowledge in the problem of inferring X, by introducing a set of statistical occupancy variables $\mathcal{G}_i^l \in \mathcal{L}$, corresponding to these extracted voxels. This is a generalization of the idea expressed in Sect. 3.1.1 for occluder voxels, to the general case of m object inter-occlusions.

3.2.2 Dependencies Consideration

Based on the dependency graph in Fig. 7, we propose a set of simplifications to the joint probability distribution of the set of variables, that reflect the prior knowledge we have about the problem. In order to simplify the writing we will often note the conjunction of a set of variables as follows: $\mathcal{G}_{1:n}^{1:m} = \{\mathcal{G}_i^l\}_{i\in\{1,\ldots,n\},l\in\{1,\ldots,m\}}$. We now decompose the joint probability $p(\mathcal{G}, \mathcal{G}_{1:n}^{1:m}, \mathcal{I}_{1:n}, \mathcal{C}_{1:n}^{1:m})$ as:

$$p(\mathcal{G}) \prod_{l\in\mathcal{L}} p(\mathcal{C}_{1:n}^l) \prod_{i,l\in\mathcal{L}} p(\mathcal{G}_i^l \mid \mathcal{G}) \prod_i p(\mathcal{I}_i \mid \mathcal{G}, \mathcal{G}_i^{1:m}, \mathcal{C}_i^{1:m}). \tag{13}$$

Prior Terms $p(\mathcal{G})$ carries prior information about the current voxel. This prior can reflect different types of knowledge and constraints already acquired about $\mathcal{G}$, e.g. localization information to guide the inference (Sect. 4). $p(\mathcal{C}^l_{1:n})$ is the prior over the view-specific appearance models of a given object l. The prior, as written over the conjunction of these parameters, could express expected relationships between the appearance models of different views, even if not color-calibrated. Since the focus in this paper is on the learning of voxel X, we do not use this capability here and assume $p(\mathcal{C}^l_{1:n})$ to be uniform.

Viewing Line Dependency Terms We have summarized the prior information along each viewing line using the m voxels most representative of the m objects, so as to model inter-object occlusion phenomena. However when examining a particular label $\mathcal{G} = l$, keeping the occupancy information about $\mathcal{G}^l_i$ would lead us to account for intra-object occlusion phenomena, which in effect would lead the inference to favor mostly voxels from the front visible surface of the object l. Because we wish to model the *volume* of object l, we discard the influence of $\mathcal{G}^l_i$ when $\mathcal{G} = l$:

$$p(\mathcal{G}^k_i \mid \{\mathcal{G} = l\}) = \mathcal{P}(\mathcal{G}^k_i) \quad \text{when } k \neq l, \tag{14}$$

$$p(\mathcal{G}^l_i \mid \{\mathcal{G} = l\}) = \delta_\emptyset(\mathcal{G}^l_i) \quad \forall l \in \mathcal{L}, \tag{15}$$

where $\mathcal{P}(\mathcal{G}^k_i)$ is a distribution reflecting the prior knowledge about $\mathcal{G}^k_i$, and $\delta_\emptyset(\mathcal{G}^k_i)$ is the distribution giving all the weight to label $\emptyset$. In (15) $p(\mathcal{G}^l_i \mid \{\mathcal{G} = l\})$ is thus enforced to be empty when $\mathcal{G}$ is known to be representing label l, which ensures that the same object is represented only once on the viewing line.

Image Formation Terms $p(\mathcal{I}_i \mid \mathcal{G}, \mathcal{G}^{1:m}_i, \mathcal{C}^{1:m}_i)$ is the image formation term. It explains what color we expect to observe given the knowledge of viewing line states and per-object color models. We decompose each such term in two subterms, by introducing a local latent variable $\mathcal{S} \in \mathcal{L}$ representing the hidden silhouette state:

$$p(\mathcal{I}_i \mid \mathcal{G}, \mathcal{G}^{1:m}_i, \mathcal{C}^{1:m}_i) = \sum_{\mathcal{S}} p(\mathcal{I}_i \mid \mathcal{S}, \mathcal{C}^{1:m}_i) p(\mathcal{S} \mid \mathcal{G}, \mathcal{G}^{1:m}_i). \tag{16}$$

The term $p(\mathcal{I}_i \mid \mathcal{S}, \mathcal{C}^{1:m}_i)$ simply describes what color is likely to be observed in the image given the knowledge of the silhouette state and the appearance models corresponding to each object. $\mathcal{S}$ acts as a mixture label: if $\{\mathcal{S} = l\}$ then $\mathcal{I}_i$ is drawn from the color model $\mathcal{C}^l_i$. For objects ($l \in \{1, \ldots, m\}$) we typically use Gaussian Mixture Models (GMM) (Stauffer and Grimson 1999) to efficiently describe the appearance information of dynamic object silhouettes. For background ($l = \emptyset$) we use per-pixel Gaussian as learned from pre-observed sequences, although other models are possible. When $l = \mathcal{U}$ the color is drawn from the uniform distribution, as we make no assumption about the color of previously unobserved objects.

The silhouette formation term $p(\mathcal{S} \mid \mathcal{G}, \mathcal{G}^{1:m}_i)$ requires that the variables be considered in their visibility order to model the occlusion possibilities. Note that this order can be different from $1, \ldots, m$. We note $\{\mathcal{G}^{v_j}_i\}_{j \in \{1,\ldots,m\}}$ the variables $\mathcal{G}^{1:m}_i$ as enumerated in the permutated order $\{v_j\}_i$ reflecting their visibility ordering on viewing line $\bar{l}_i$. If $\{g\}_i$ denotes the particular index after which the voxel X itself appears on viewing line $\bar{l}_i$, then we can re-write the silhouette formation term as $p(\mathcal{S} \mid \mathcal{G}^{v_1}_i \cdots \mathcal{G}^{v_g}_i, \mathcal{G}, \mathcal{G}^{v_{g+1}}_i \cdots \mathcal{G}^{v_m}_i)$. A distribution of the following form can then be assigned to this term:

$$p(\mathcal{S} \mid \emptyset \cdots \emptyset, l, * \cdots *) = d_l(\mathcal{S}) \quad \text{with } l \neq \emptyset \tag{17}$$

$$p(\mathcal{S} \mid \emptyset, \ldots\ldots\ldots, \emptyset) = d_\emptyset(\mathcal{S}), \tag{18}$$

where $d_k(\mathcal{S}), k \in \mathcal{L}$ is a family of distributions giving strong weight to label k and lower equal weight to others, determined by a constant probability of detection $P_d \in [0, 1]$: $d_k(\mathcal{S} = k) = P_d$ and $d_k(\mathcal{S} \neq k) = \frac{1-P_d}{|\mathcal{L}|-1}$ to ensure summation to 1. Equation (17) thus expresses that the silhouette pixel state reflects the state of the first visible non-empty voxel on the viewing line, regardless of the state of voxels behind it ("$*$"). Equation (18) expresses the particular case where no occupied voxel lies on the viewing line, the only case where the state of $\mathcal{S}$ should be background: $d_\emptyset(\mathcal{S})$ ensures that $\mathcal{I}_i$ is mostly drawn from the background appearance model.

3.2.3 Dynamic Object Inference

Estimating the occupancy at voxel X translates to estimating $p(\mathcal{G} \mid \mathcal{I}_{1:n}, \mathcal{C}^{1:m}_{1:n})$ in Bayesian terms. We apply Bayes' rule using the joint probability distribution, marginalizing out the unobserved variables $\mathcal{G}^{1:m}_{1:n}$:

$$p(\mathcal{G} \mid \mathcal{I}_{1:n}, \mathcal{C}^{1:m}_{1:n}) = \frac{1}{z} \sum_{\mathcal{G}^{1:m}_{1:n}} p(\mathcal{G}, \mathcal{G}^{1:m}_{1:n}, \mathcal{I}_{1:n}, \mathcal{C}^{1:m}_{1:n}) \tag{19}$$

$$= \frac{1}{z} p(\mathcal{G}) \prod_{i=1}^{n} f^1_i \tag{20}$$

where

$$f^k_i = \sum_{\mathcal{G}^{v_k}_i} p(\mathcal{G}^{v_k}_i \mid \mathcal{G}) f^{k+1}_i \quad \text{for } k < m, \tag{21}$$

and

$$f^m_i = \sum_{\mathcal{G}^{v_m}_i} p(\mathcal{G}^{v_m}_i \mid \mathcal{G}) p(\mathcal{I}_i \mid \mathcal{G}, \mathcal{G}^{1:m}_i, \mathcal{C}^{1:m}_i). \tag{22}$$

Similar to (10), the normalization constant z is obtained by ensuring that the distribution of (19) sum up to 1: $z = \sum_{\mathcal{G},\mathcal{G}_{1:n}^{1:m}} p(\mathcal{G}, \mathcal{G}_{1:n}^{1:m}, \mathcal{I}_{1:n}, \mathcal{C}_{1:n}^{1:m})$. The sum in this form is intractable, thus we factorize the sum in (20). The sequence of m functions f_i^k specify how to recursively compute the marginalization with the sums of individual $\mathcal{G}_i^k$ variables appropriately subsumed, so as to factor out terms not required at each level of the sum. Because of the particular form of silhouette terms in (17), this sum can be efficiently computed, given that all terms after a first occupied voxel of the same visibility rank k share a term of identical value in $p(\mathcal{I}_i \mid \emptyset \cdots \emptyset, \{\mathcal{G}_i^{v_k} = l\}, * \cdots *) = \mathcal{P}_l(\mathcal{I}_i)$. They can be factored out of the remaining sum, which sums to 1 being a sum of terms of a probability distribution, leading to the following simplification of (21), $\forall k \in \{1, \ldots, m-1\}$:

$$f_i^k = p(\mathcal{G}_i^{v_k} = \emptyset \mid \mathcal{G})\, f_i^{k+1} + \sum_{l \neq \emptyset} p(\mathcal{G}_i^{v_k} = l \mid \mathcal{G}) \mathcal{P}_l(\mathcal{I}_i). \quad (23)$$

4 Automatic Learning and Tracking

We have presented in Sect. 3 a generic framework to infer the occupancy probability of a voxel X and thus deduce how likely it is for X to belong to one of m objects. Some additional work is required to use it to model objects in practice. The formulation explains how to compute the occupancy of X if some occupancy information about the viewing lines is already known. Thus the algorithm needs to be initialized with a coarse shape estimate, whose computation is discussed in Sect. 4.1. Intuitively, object shape estimation and tracking are complementary and mutually helpful tasks. We explain in Sect. 4.2 how object localization information is computed and used in the modeling. To be fully automatic, our method uses the inference label $\mathcal{U}$ to detect objects not yet assigned to a given label and learn their appearance models (Sect. 4.3). Finally, static occluder computation can easily be integrated in the system and help the inference be robust to static occluders (Sect. 4.4). The algorithm at every time instance is summarized in Algorithm 1.

4.1 Shape Initialization and Refinement

The proposed formulation relies on some prior knowledge about the scene occupancies and dynamic object ordering. Thus part of the occupancy problem must be solved to bootstrap the algorithm. Fortunately, using multi-label silhouette inference with no prior knowledge about occupancies or consideration for inter-object occlusions provides a decent initial m-occupancy estimate. This simpler inference case can easily be formulated by simplifying occlusion related variables from (20):

$$p(\mathcal{G} \mid \mathcal{I}_{1:n}, \mathcal{C}_{1:n}^{1:m}) = \frac{1}{z} p(\mathcal{G}) \prod_{i=1}^{n} p(\mathcal{I}_i \mid \mathcal{G}, \mathcal{C}_i^{1:m}). \quad (24)$$

Algorithm 1 Dynamic Scene Reconstruction

Input: Frames at a new time instant for all views
Output: 3D object shapes in the scene
Coarse Inference;
if a new object enters the scene **then**
  add a label for the new object;
  initialize foreground appearance model;
  go back to **Coarse Inference**;
end if
Refined Inference;
static occluder inference;
update object location and prior;
return

This initial *coarse inference* can then be used to infer a second, *refined inference*, this time accounting for viewing line obstructions, given the voxel priors $p(\mathcal{G})$ and $\mathcal{P}(\mathcal{G}_i^j)$ of (14) computed from the coarse inference. The prior over $p(\mathcal{G})$ is then used to introduce soft constraints to the inference. This is possible by using the coarse inference result as the input of a simple localization scheme, and using the localization information in $p(\mathcal{G})$ to enforce a compactness prior over the m objects, as discussed in Sect. 4.2.

4.2 Object Localization

We use a localization prior to enforce the compactness of objects in the inference steps. For the particular case where walking people represent the dynamic objects, we take advantage of the underlying structure of the dataset, by projecting the maximum probability over a vertical voxel column on the horizontal reference plane. We then localize the most likely position of objects by sliding a fixed-size window over the resulting 2D probability map for each object. The resulting center is subsequently used to initialize $p(\mathcal{G})$, using a cylindrical spatial prior. This favors objects localized in one and only one portion of the scene and is intended as a soft guide to the inference. Although simple, this tracking scheme is shown to outperform state of the art methods (Sect. 5.2.2), thanks to the rich shape and occlusion information modeled.

4.3 Automatic Detection of New Objects

The main information about objects used by the proposed method is their set of appearances in the different views.

These sets can be learned offline by segmenting each observed object alone in a clear, uncluttered scene before processing multi-objects scenes. More generally, we can initialize object color models in the scene automatically. To detect new objects we compute $\mathcal{U}$'s object location and volume size during the coarse inference, and track the unknown volume just like other objects as described in Sect. 4.2. A new dynamic object inference label is created (and m incremented), if all of the following criteria are satisfied:

- The entrance is only at the scene boundaries
- $\mathcal{U}$'s volume size is larger than a threshold
- $\mathcal{U}$ is not too close to the scene boundary
- Subsequent updates of $\mathcal{U}$'s track are bounded

To build the color model of the new object, we project the maximum voxel probability along the viewing ray to the camera view, threshold the image to form a "silhouette mask", and choose pixels within the mask as training samples for a GMM appearance model. Samples are only collected from unoccluded silhouette portions of the object, which can be verified from the inference. Because the cameras may be badly color-calibrated, we propose to train an appearance model for each camera view separately. This approach is fully evaluated in Sect. 5.2.1.

4.4 Occluder computation

The static occluder computation can easily be integrated with the multiple dynamic object reconstruction described in Sect. 3.1. At every time instant the dominant occupancy probabilities of m objects are already extracted; the two dominant occupancies in front and behind the current voxel X can be used in the occupancy inference formulation of Sect. 3.1. It could be thought that the multi-label dynamic object inference discussed in this section is an extension to the single dynamic object cases assumed in Sect. 3.1. In fact, the occlusion occupancy inference does benefit from the disambiguation inherent to multi-silhouette reasoning, as the real-world experiment shows, in Fig. 16, in Sect. 5.

5 Result and Evaluation

5.1 Occlusion Inference Results

To demonstrate the validity of the static occluder shape recovery, we mainly use a single person as the dynamic object in the scene. In the next section, we also show that it can be recovered in the presence of multiple dynamic objects. We show three sequences: the PILLARS and SCULPTURE sequences, acquired outdoors, and the CHAIR sequence, acquired indoors, with combined artificial and natural light from large bay windows. In all sequences nine DV cameras surround the scene of interest, background models are learned in the absence of moving objects. A single person as our dynamic object walks around and through the occluder in each scene. The shape of the person is estimated at each considered time step and used as prior to occlusion inference. The data is used to compute an estimate of the occluder shape using (10). Results are presented in Fig. 8.

Nine geometrically calibrated 720×480 resolution cameras all record at 30 fps. Color calibration is unnecessary because the model uses silhouette information only. The background model is learned per-view using a single RGB Gaussian color model per pixel, and training images. Although simple, the model is proved sufficient, even in outdoor sequences subject to background motion, foreground object shadows, and substantial illumination changes, illustrating the strong robustness of the method to difficult real conditions. The method copes well with background misclassifications that do not lead to large coherent false positive dynamic object estimations: pedestrians are routinely seen in the background for the SCULPTURE and PILLARS sequences (e.g. Fig. 8(a1)), without any significant corruption of the inference.

Adjacent frames in the input videos contain largely redundant information for occluder modeling, thus videos can safely be subsampled. PILLARS was processed using 50% of the frames (1053 frames processed), SCULPTURE and CHAIR with 10% (160 and 168 processed frames respectively).

5.1.1 Online Computation Results

All experiments can be computed using incremental inference updates. Figure 9 depicts the inference's progression, using the sensor fusion formulation alone or in combination with the reliability criterion. For the purpose of this experiment, we used the PILLARS sequence and manually segmented the occluder in each view for a ground truth comparison, and focused on a subregion of the scene in which the expected behaviors are well isolated. Figure 9 shows that both schemes converge reasonably close to the visual hull of the considered pillar. In scenes with concave parts accessible to dynamic objects, the estimation would carve into concavities and reach a better estimate than the occluder's visual hull. A somewhat larger volume is reached with both schemes in this example. This is attributable to calibration errors which over-tightens the visual hull with respect to the true silhouettes, and accumulation of errors in both schemes toward the end of the sequence. We trace those to the redundant, periodical poses contained in the video, that sustain consistent noise. This suggests the existence of an optimal finite number of frames to be used for processing. Jolts can be observed in both volumes corresponding to instants where the person walks behind the pillar, thereby adding positive contributions to the inference. The use of the reliability criterion defined in Sect. 3.1.5 contributes to lower sensitivity

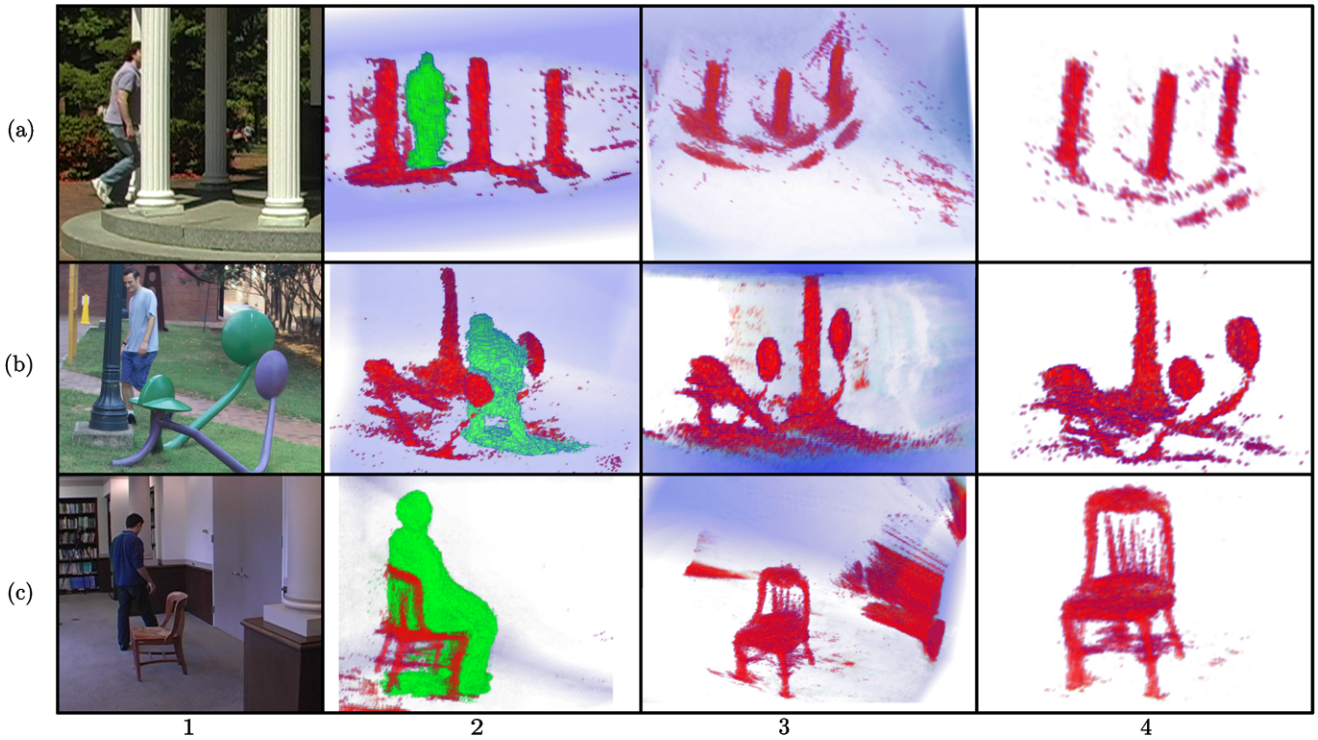

Fig. 8 Occluder shape retrieval results. Sequences: (**a**) PILLARS, (**b**) SCULPTURE, (**c**) CHAIR. (1) Scene overview. Note the harsh light, difficult backgrounds for (**a**) and (**b**), and specularity of the sculpture, causing no significant modeling failure. (2–3) Occluder inference according to (10). *Blue*: neutral regions (prior $\mathcal{P}_o$), *red*: high probability regions. Brighter/clear regions indicate the inferred absence of occluders. Fine levels of detail are modeled, sometimes lost—mostly to calibration. In (a) the structure's steps are also detected. (4) Same inference with additional exclusion of zones with reliability under 0.8. Peripheral noise and marginally observed regions are eliminated. The background protruding shape in (c3) is accounting for an actual occlusion from a single view, the pillar visible in (c3)

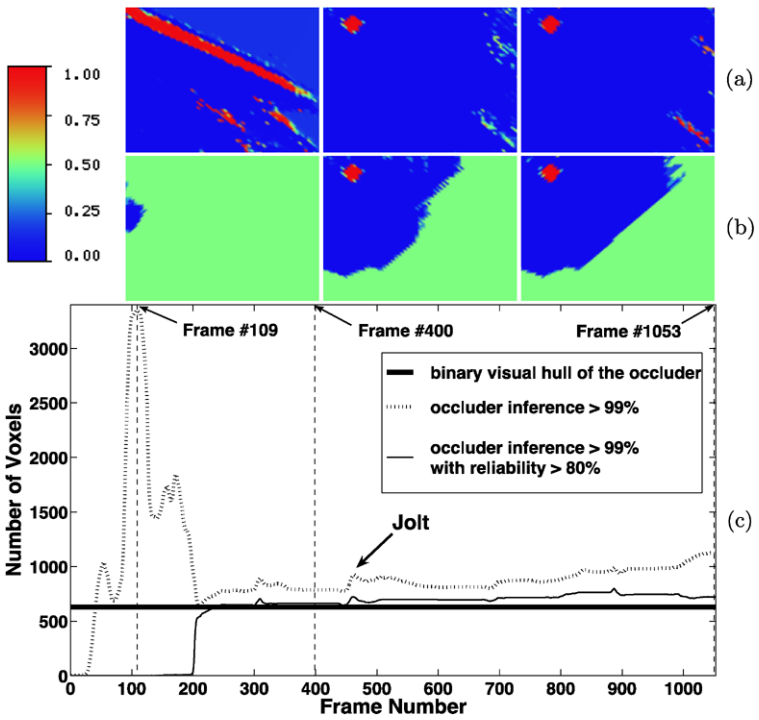

Fig. 9 Online inference analysis and ground truth visual hull comparison, using PILLARS dataset, focusing on a slice including the middle pillar. (**a**) Frames 109, 400 and 1053, inferred using (10). (**b**) Same frames, this time excluding zones with reliability under 0.8 (reverted here to 0.5). (**c**) Number of voxels compared to ground truth visual hull across time

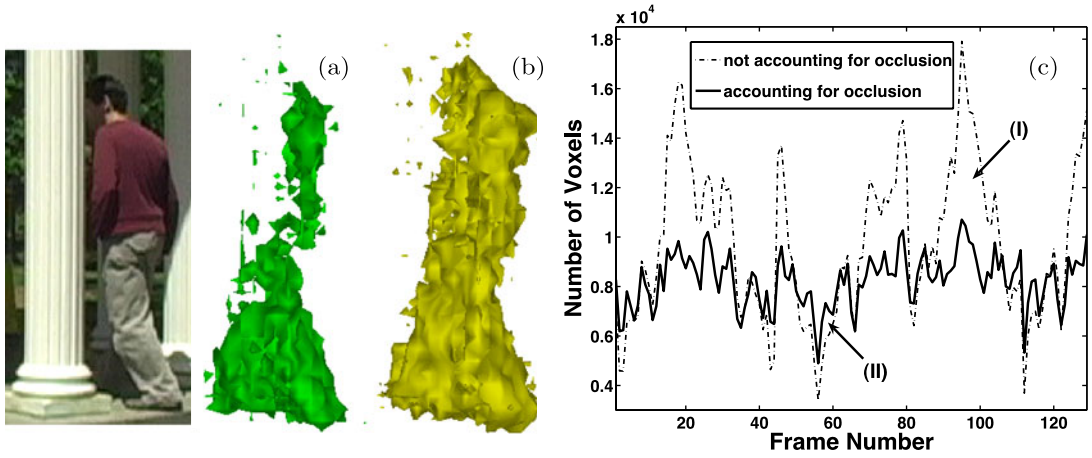

Fig. 10 (**a**) Person shape estimate from PILLARS sequence, as occluded by the rightmost pillar and computed without accounting for occlusion. (**b**) Same situation accounting for occlusion, showing better completeness of the estimate. (**c**) Volume plot in both cases. Accounting for occlusion leads to more stable estimates across time, decreases false positives and overestimates due to shadows cast on occluders (*I*), increases estimation probabilities in case of occlusion (*II*)

to noise, as well as a permanently conservative estimate of the occluder volume as the curves show in frames 100–200. Raw inference (10) momentarily yields large hypothetical occluder volumes when data is biased toward contributions of an abnormally low subset of views (frame 109).

5.1.2 Accounting for Occlusion in SfS

Our formulation (Sect. 3.1.6) can be used to account for the accumulated occluder information in dynamic shape inference. We only use occlusion cues from reliable voxels ($R > 0.8$) to minimize false positive occluder estimates, whose excessive presence would lead to sustained errors. While in many cases the original dynamic object formulation (Franco and Boyer 2005) performs robustly, a number of situations benefit from the additional occlusion knowledge (Fig. 10). Person volume estimates can be obtained when accounting for occluders. These estimates appear on average to be a stable multiple of the real volume of the person, which depends mainly on camera configuration. This suggests a possible biometrics application of the method, for disambiguation of person recognition based on computed volumes.

5.2 Multi-Object Shape Inference Results

We have used four multi-view sequences to validate multi-object shape inference. Eight 30 fps 720 × 480 DV cameras surrounding the scene in a semi-circle were used for the CLUSTER and BENCH sequences. The LAB sequence is provided by (Gupta et al. 2007) and SCULPTURE was used to reconstruct the static sculpture (Fig. 8(b)) in the previous section. Here, we show the result of multiple persons walking in the scene together with the reconstructed sculpture.

Table 2

	No. of Cam.	No. of Dynamic Obj.	Occluder
CLUSTER (outdoor)	8	5	no
BENCH (outdoor)	8	0–3	yes
LAB (indoor)	15	4	no
SCULPTURE (outdoor)	9	2	yes

Cameras in each data sequence are geometrically calibrated but not color calibrated. The background model is learned per-view using a single Gaussian color model at every pixel, with training images. Although simple, the model is proved to be sufficient, even for outdoor sequences subject to background motion, foreground object shadows, window reflections and substantial illumination changes, showing the robustness of the method to difficult real conditions. For dynamic object appearance models of the CLUSTER, LAB and SCULPTURE data sets, we off-line train a per-view RGB GMM model for each person with manually segmented foreground images. For the BENCH sequence however, appearance models are initialized online automatically, using the method described in Sect. 4.3.

5.2.1 Appearance Modeling Validation

It is extremely hard to color-calibrate a large number of cameras, not to mention under varying lighting conditions, as in a natural outdoor environment. To show this, we compare different appearance modeling schemes in Fig. 11, for a frame of the outdoor BENCH dataset. Without loss of generality, we use GMMs. The first two rows compare silhouette extraction probabilities using the color models of spatially

neighboring views. These indicate that stereo approaches which heavily depend on color correspondence across views are very likely to fail in the natural scenarios, especially when the cameras have dramatic color variations, such as in view 4 and 5. The global appearance model on row 3 performs better than row 1 and 2, but this is mainly due to its compensation between large color variations across camera views, which at the same time, decreases the model's discriminability. The last row obviously is the winner where a color appearance model is independently maintained for every camera view. We hereby use the last scheme in our system. Once the model is trained, we do not update it as time goes by. But this online updating of the appearance models could be an easy extension for robustness.

One more thing to note, is that in our approach, even though an object's appearances are learnt for each view separately, they are still linked together in 3D by the same object label. In this sense, our per-view based appearances can be taken as an intermediate model between the global model as used in Shape from Photo-consistency and multi-view stereo, and the pure 2D image models used by video surveillance and tracking literatures.

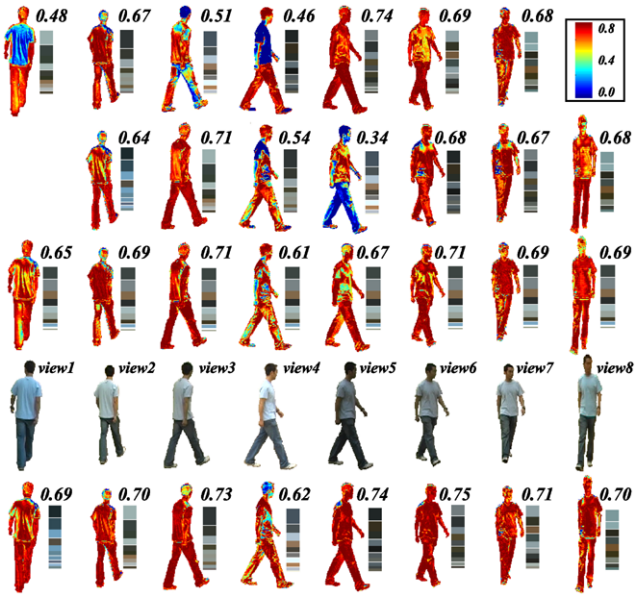

Fig. 11 Appearance model analysis. A person in eight views is displayed in row 4. A GMM model $\mathcal{C}_i$ is trained for view $i \in [1, 8]$. A global GMM model $\mathcal{C}_0$ over all views is also trained. Row 1, 2, 3 and 5 compute $\mathcal{P}(\mathcal{S} \mid \mathcal{I}, \mathcal{B}, \mathcal{C}_{i+1})$, $\mathcal{P}(\mathcal{S} \mid \mathcal{I}, \mathcal{B}, \mathcal{C}_{i-1})$, $\mathcal{P}(\mathcal{S} \mid \mathcal{I}, \mathcal{B}, \mathcal{C}_0)$ and $\mathcal{P}(\mathcal{S} \mid \mathcal{I}, \mathcal{B}, \mathcal{C}_i)$ for view i respectively, with $\mathcal{S}$ the foreground label, $\mathcal{I}$ the pixel color, $\mathcal{B}$ the uniform background model. The probability is displayed according to the color scheme at the top right corner. The average probability over all pixels in the silhouette region and the mean color modes of the applied GMM model are shown for each figure

5.2.2 *Densely Populated Scene*

The CLUSTER sequence is a particularly challenging configuration: five people are on a circle of less than 3 m. in diameter, yielding an extremely ambiguous and occluded situation at the circle center. Despite the fact that none of them are being observed in all views, we are still able to recover the people's label and shape. Images and results are shown in Fig. 12. The naive 2-label reconstruction (probabilistic visual hull) yields large volumes with little separation between objects, because the entire scene configuration is too ambiguous. Adding tracking prior information estimates the most probable compact regions and eliminates large errors, at the expense of dilation and lower precision. Accounting for viewing line occlusions enables the model to recover more detailed information, such as the limbs.

The LAB sequence (Gupta et al. 2007) with poor image contrast is also processed. The reconstruction result from all 15 cameras is shown in Fig. 13. Moreover, in order to evaluate our localization prior estimation, we compare our tracking method (Sect. 4.2) with the ground truth data, the result of (Gupta et al. 2007) and (Mittal and Davis 2003). We use the exactly same eight cameras as in (Mittal and Davis 2003)

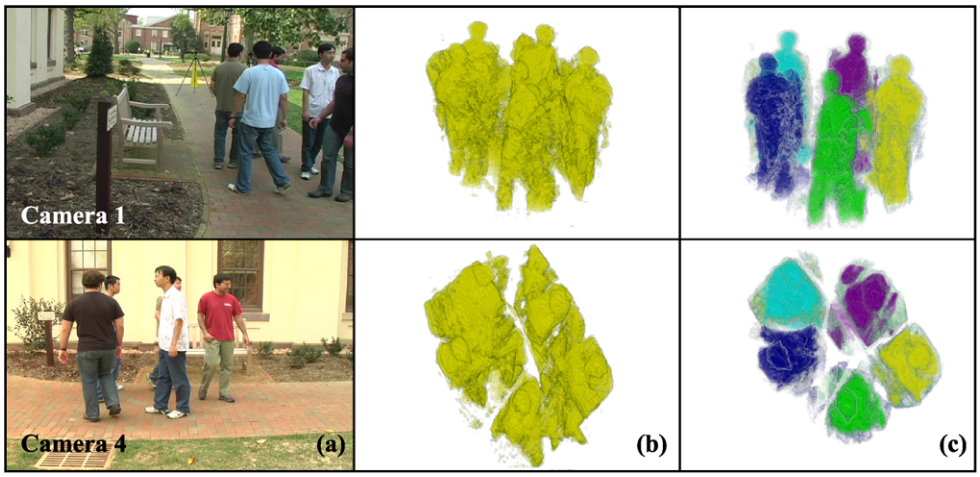

Fig. 12 Result from 8-view CLUSTER dataset. (**a**) Two views at frame 0. (**b**) Respective 2-labeled reconstruction. (**c**) More accurate shape estimation using our algorithm

for the comparison, shown in Fig. 13(b). Our method is generally more robust in tracking, and also builds 3D shape information. Most existing tracking methods only focus on a tracking envelope and do not compute precise 3D shapes. It is this shape information that enables our method to achieve comparable or better precision.

5.2.3 Automatic Appearance Model Initialization

The automatic dynamic object appearance model initialization has been tested using the BENCH sequence. Three people are walking into the empty scene one after another. By examining the unidentified label $\mathcal{U}$, object appearance models are initialized and used for shape estimation in subsequent frames. Volume size evolution of all labels are shown in Fig. 14 and the reconstructions at two time instants are shown in Fig. 15.

During the sequence, $\mathcal{U}$ has three major volume peaks due to three new persons entering the scene. Some smaller perturbations are due to shadows on the bench or the ground. Besides automatic object appearance model initialization, the system robustly re-detects and tracks the person who leaves and re-enters the scene. This is because once the label is initialized, it is evaluated for every time instant, even if the person is out of the scene. The algorithm can easily be improved to handle leaving/reentering labels transparently.

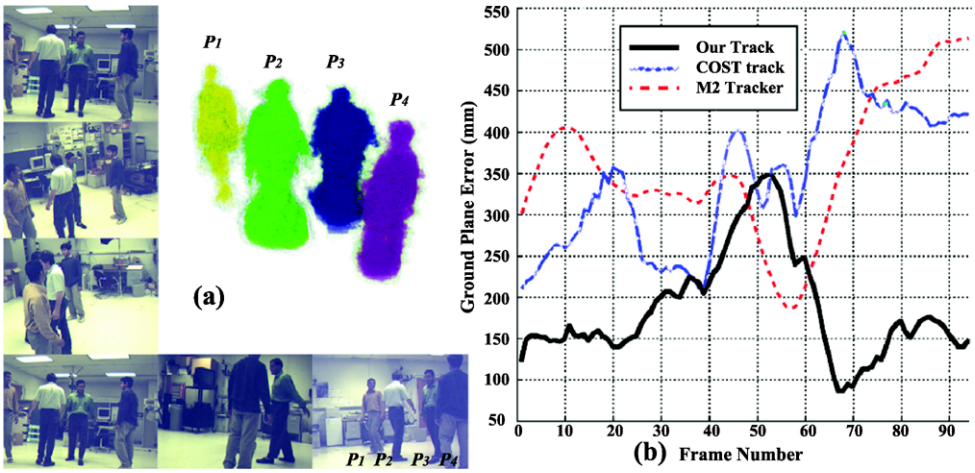

Fig. 13 LAB dataset result from (Gupta et al. 2007). (**a**) 3D reconstruction with 15 views at frame 199. (**b**) 8-view tracking result comparison with methods in (Gupta et al. 2007; Mittal and Davis 2003) and the ground truth data. Mean error in ground plane estimate in *mm* is plotted

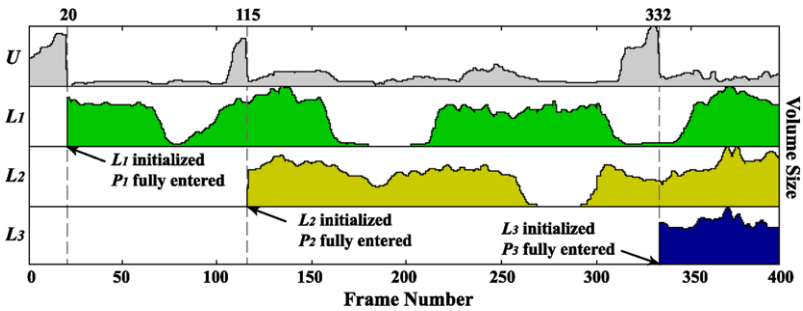

Fig. 14 Appearance model automatic initialization with the BENCH sequence. The volume of $\mathcal{U}$ increases if a new person enters the scene. When an appearance model is learned, a new label is initialized. During the sequence, L_1 and L_2 volumes drop to near zero value because they walk out of the scene on those occasions

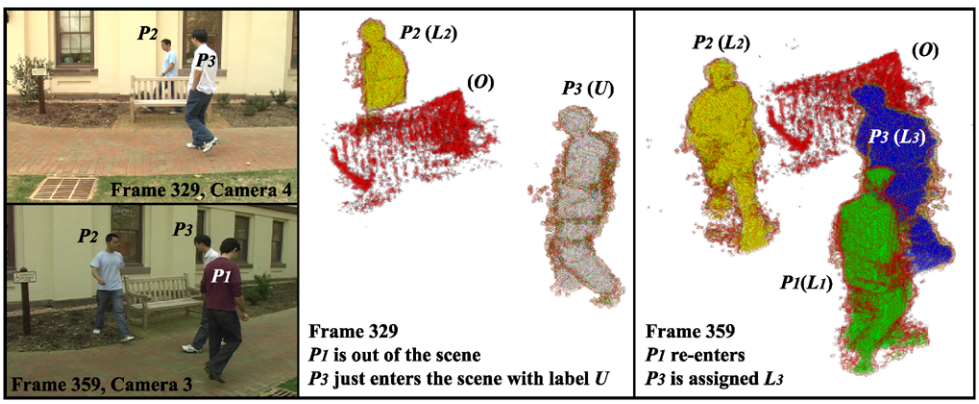

Fig. 15 BENCH result. Person numbers are assigned according to the order their appearance models are initialized. At frame 329, P_3 is entering the scene. Since it's P_3's first time into the scene, he is captured by label $\mathcal{U}$ (*gray color*). P_1 is out of the scene at the moment. At frame 359, P_1 has re-entered the scene. P_3 has its GMM model already trained and label L_3 assigned. The bench as a static occluder is being recovered

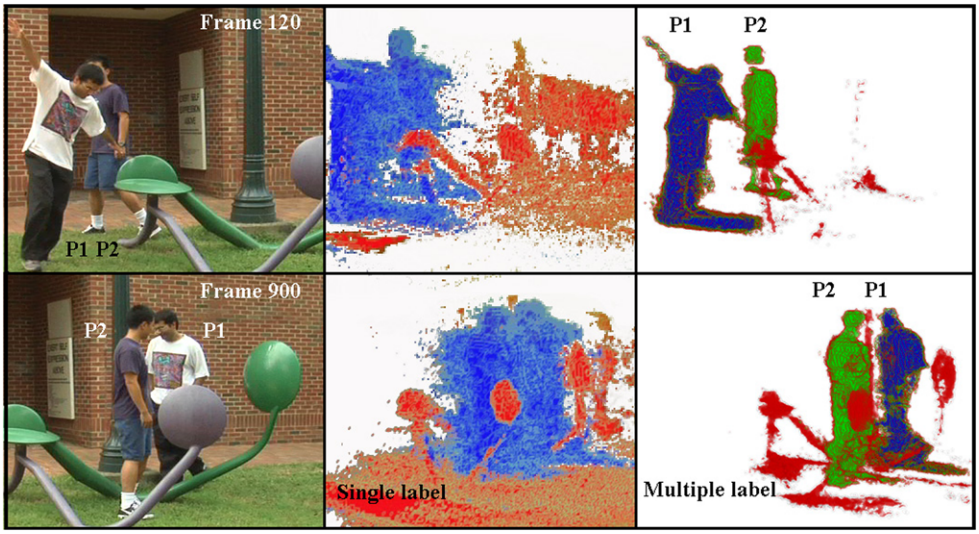

Fig. 16 SCULPTURE data set comparison. The *middle column* shows the reconstruction with a single foreground label. The *right column* shows the reconstruction with a label for each person. This figure shows that, by resolving inter-occlusion ambiguities, both the static occluder and dynamic objects achieve better quality

5.2.4 Dynamic Object and Occluder Inference

The BENCH sequence demonstrates the power of our automatic appearance model initialization as well as the integrated occluder inference of the *"bench"* as shown in Fig. 15 between frame 329 and 359. Figure 14 illustrates the status of our scene tracking and modeling across time.

We also compute result for SCULPTURE sequence with two persons walking in the scene, as shown in Fig. 16. For the dynamic objects, we manage to get much cleaner shapes when the two persons are close to each other, and more detailed shapes such as extended arms. For the occluder, thanks to the multiple foreground modes and the consideration of inter-occlusion between the dynamic objects in the scene, we are able to recover the fine shape as well. The occluder inference would otherwise be perturbed by dynamic shape overestimations.

6 Discussion

6.1 Dynamic Object and Static Occluder Comparison

We have shown the probabilistic models and real datasets for static and dynamic shapes inference. Although both types of entities are computed only from silhouette cues from camera views and both require the consideration of occlusions, they actually have fundamentally different characteristics.

First of all, there is no way to learn an appearance model for a static occluder, because its appearance is initially embedded in the background model of a certain view. Only when an occlusion event happens between the dynamic object and the occluder, can we detect that certain appearance should belong to the occluder but not the background, and the occluder probability should increase along that viewing direction. Whereas for dynamic objects, we have mentioned and will show in more detail in the next section, that their appearance models for all camera views could be manually or automatically learnt before reconstruction.

Secondly, because occluders are static, regions in the 3D scene that have been recovered as highly probable to be occluder will always maintain the high probabilities, not considering noise. This enables the accumulation of the static occluder in our algorithm. But for the inter-occlusion between dynamic objects, it is just a one time instant event. This effect is actually reflected in the inference formulae of the static occluder and the dynamic objects.

Thirdly, a recovered dynamic object can be thought of as a probabilistic visual hull representation, because it is after all a fusion of silhouette information, based on (Franco and Boyer 2005). However, the static occluder that we recover is actually not a visual hull representation. In fact, it is an entity that is carved out using moving visual hulls (of the dynamic objects), as shown in Fig. 2. Therefore, our estimated occluder shape can maintain some concavities, as long as a dynamic object can move into the concave regions and be witnessed by camera views.

Finally, the computed static occluder shape is in a probabilistic form. Its counterpart in the deterministic representation is given in Fig. 2. Since it is formed by carving away dynamic shapes, it has some unique properties that are different from the traditional visual hull. Consider a dynamic shape D with infinitesimal volume. We define an *occluder hull* as an approximation volume to a static occluder recovered with a infinitesimal dynamic shape D moving randomly in all accessible parts of the scene. It can be shown that the occluder hull is the region of space visible to at most one camera, including the inside of the actual occluder shape. In Fig. 17(a) and (b), the thick black lines delineate the occluder hull. In comparison, in Fig. 17(c) and (d), the

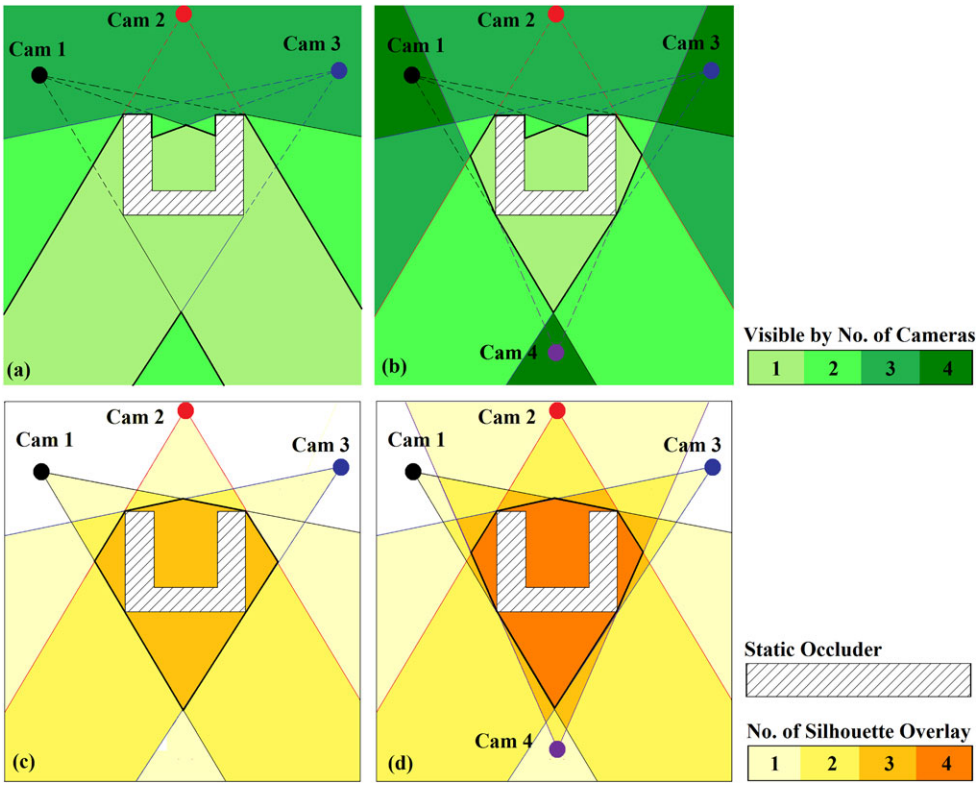

Fig. 17 2D theoretical occluder hull and visual hull. (**a**) 3 camera occluder hull; (**b**) 4 camera occluder hull; (**c**) 3 camera visual hull; (**d**) 4 camera visual hull. Concavities can be recovered by occluder hull

thick black lines delineate the visual hull of the occluder, assuming the silhouettes of the objects are known. The visual hull is the intersection of all the silhouettes' visual cones.

Figure 17 shows that contrary to the visual hull, the occluder hull can recover concavities. In fact, when cameras are distributed all over space, the actual shape of an arbitrary static occluder can be recovered. A finite number of cameras may be sufficient to recover arbitrary occluder shapes in certain cases. However, the occluder hull shape is highly dependent on the camera placement. As (a) shows, the occluder hull may even not be closed. For occluder hull, there is no lower bound number to guarantee a closed shape. Although the occluder hull in (b) is closed, if the fourth camera changes its orientation or position, the occluder hull may be open again. On the contrary, only two silhouettes from different views can guarantee a closed visual hull, which is the minimum number of cameras required for a visual hull. Given the above analysis, some empirical requirements for good quality occluder estimation are summarized as follows:

– There is no guarantee that how many cameras would produce a closed occluder shape. But when the size of the occluder is small relative to the camera focal length, or the occluder position is so far from the cameras that the region where only one camera can see the dynamic shape is limited, a closed occluder shape can usually be recovered with the proposed the algorithm.
– For a region behind the occluder, where no camera view has sampled, the algorithm cannot infer any information. For example, the algorithm does not recover the wall, if a person is hiding completely behind it. In this case, the person's occupancy is not recovered in the first place. One solution may be to add more camera views behind the wall.
– Since the closed dynamic shape (needs at least two camera views) is required by the algorithm, plus an occluded view for the occluding incidence, in theory, the minimum requirement for the occluder inference is three cameras.

6.2 Computation Complexity and Acceleration

The occluder occupancy computation was tested on a 2.8 GHz PC at approximately 1 minute per frame. The strong locality inherent to the algorithm and preliminary benchmarks suggest that real-time performance could be achieved using a GPU implementation. We choose nVIDIA CUDA pipeline, yielding a 15× speedup for the complete algorithm. The dynamic shape computation alone reaches a speed-up of more than 80 times and a speed of 0.2 second per frame on the test machine, which is already satisfactory for real-time applications.

Although it has gained reasonable speedup, the static occluder computation could not yet achieve interactive frame rate (0.9 to 3.15 seconds per time instant), due to the high computational cost for finding the dynamic components in

front of and behind every voxel location. However, since dynamic objects of adjacent frames often yield redundant information, an interactive system can be obtained by updating the occluder at a lower frequency.

The time complexity of our multiple dynamic shape plus static occluder system is bounded by the dynamic object inference, where viewing ray maximum probabilities for each label need and each view need to be know. This means a computation of $\mathcal{O}(nmV)$, with n the number of cameras, m the number of objects in the scene, and V the scene volume resolution. We process the multiple dynamic object sequences in Sect. 5.2 on a 2.4 GHz Core Quad PC with computation times varying of 1 to 4 minutes per frame. The very strong locality inherent to the algorithm and preliminary benchmarks suggest faster performance could be accomplished by a GPU acceleration.

6.3 Limitations and Future Works

There are a few limitations to our approach. First of all, although the static occluder estimation is robust in a general outdoor environment, it is not generally an alternative for static object reconstruction purpose (although it works in some cases, like the CHAIR sequence). This is because our occluder inference is only based on occlusion cues, meaning if there is no occlusion between a dynamic object with the static occluder in a view, we cannot discover the occluder shape. This is why we cannot recover the top of the pillar and lamp post in Fig. 8. However, for dynamic scene analysis, our main focus is on the dynamic objects, in this case, our recovered knowledge about where a dynamic object may possibly be occluded by a static occluder is very important.

Secondly, the dynamic shape GMM appearance models can be improved. If two persons with similar color appearances are in the scene, this is a problem to our current scheme—it always introduce ambiguities to our dynamic object inference scheme. In this case, the proposed tracking scheme and object location prior will be the main information source to disambiguate individuals. But the tracking scheme can also be improved. The cylindrical object location prior can be extended to more sophisticated structure/shape models that further enforce temporal consistency.

Finally, the optimal camera count and placement for acquisitions in a given scenario could be the subject of further studies.

7 Summary

In this paper, we have presented a complete approach to reconstruct 3D shapes in a dynamic event from silhouettes extracted from multiple videos recorded using a geometrically calibrated camera network. The key elements of our approach is a probabilistic volumetric framework for automatic 3D dynamic scene reconstruction, which is robust to many perturbations, including occlusion, lighting variation, shadows etc. It does not require photometric calibration among the cameras in the system. It automatically learns the appearance of the dynamic objects, tracks the motions and detects surveillance events such as entering/leaving the scene. It also automatically discovers the static occluder, whose appearance is initially hidden in the background and recovers its shape by observing the dynamic objects' movement in the scene over a given time period. Combining all the algorithms described in this paper, it is possible to develop a fully automatic and robust system for dynamic scene analysis in general uncontrolled indoor/outdoor environment.

Acknowledgements We would like to thank A. Gupta et al. (2007), Mittal and Davis (2003) for providing us the 16-camera dataset. We also gratefully acknowledge the support of David and Lucille Packard Foundation Fellowship and NSF Career award IIS-0237533.

References

Apostoloff, N., & Fitzgibbon, A. (2005). Learning spatiotemporal T-junctions for occlusion detection. In *CVPR* (Vol. 2, pp. 553–559).

Baumgart, B. G. (1974). *Geometric modeling for computer vision*. PhD thesis, 1974.

De Bonet, J. S., & Viola, P. (1999). Roxels: responsibility weighted 3d volume reconstruction. In *ICCV* (Vol. 1, pp. 418–425).

Broadhurst, A., Drummond, T., & Cipolla, R. (2001). A probabilistic framework for the space carving algorithm. In *ICCV* (Vol. 1, pp. 388–393).

Brostow, G., & Essa, I. (1999). Motion based decompositing of video. In *ICCV* (Vol. 1, pp. 8–13).

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, *22*(6), 46–57. Special issue on autonomous intelligent machines.

Elgammal, A., Duraiswami, R., Harwood, D., & Davis, L. (2002). Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, *90*(7), 1151–1163.

Favaro, P., Duci, A., Ma, Y., & Soatto, S. (2003). On exploiting occlusions in multiple-view geometry. In *ICCV* (Vol. 1, pp. 479–486).

Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2007). Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Patern Analysis and Machine Intelligence*, *30*(2), 267–282.

Franco, J.-S., & Boyer, E. (2003). Exact polyhedral visual hulls. In *BMVC* (Vol. 1, pp. 329–338).

Franco, J.-S., & Boyer, E. (2005). Fusion of multi-view silhouette cues using a space occupancy grid. In *ICCV* (Vol. 2, pp. 1747–1753).

Furukawa, Y., & Ponce, J. (2006). Carved visual hulls for image-based modeling. In *ECCV* (Vol. 1, pp. 564–577).

Grauman, K., Shakhnarovich, G., & Darrell, T. (2003). A Bayesian approach to image-based visual hull reconstruction. In *CVPR* (Vol. 1, pp. 187–194).

Guan, L., Sinha, S., Franco, J.-S., & Pollefeys, M. (2006). Visual hull construction in the presence of partial occlusion. In *3DPVT* (Vol. 1, pp. 413–420).

Guan, L., Franco, J.-S., & Pollefeys, M. (2007). 3D occlusion inference from silhouette cues. In *CVPR* (pp. 1–8).

Guan, L., Franco, J.-S., & Pollefeys, M. (2008). Multi-object shape estimation and tracking from silhouette cues. In *CVPR* (pp. 1–8).

Gupta, A., Mittal, A., & Davis, L. S. (2007). Cost: an approach for camera selection and multi-object inference ordering in dynamic scenes. In *ICCV* (pp. 1–8).

Hoiem, D., Stein, A., Efros, A., & Hebert, M. (2007). Recovering occlusion boundaries from a single image. In *ICCV* (pp. 1–8).

Ilie, A., & Welsh, G. (2005). Ensuring color consistency across multiple cameras. In *ICCV* (Vol. 2, pp. 1268–1275).

Joshi, N., Wilburn, B., Vaish, V., Levoy, M., & Horowitz, M. (2005). *Automatic color calibration for large camera arrays* (UCSD CSE Tech Report CS2005-0821).

Keck, M., & Davis, J. (2008). 3D occlusion recovery using few cameras. In *CVPR* (pp. 1–8).

Kim, K., Harwood, D., & Davis, L. (2005). Background updating for visual surveillance. In *ISVC* (Vol. 1, pp. 337–346).

Kutulakos, K., & Seitz, S. (2000). A theory of shape by space carving. *International Journal of Computer Vision*, *38*(3), 199–218.

Laurentini, A. (1994). The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Patern Analysis and Machine Intelligence*, *16*(2), 150–162.

Lazebnik, S., Boyer, E., & Ponce, J. (2001). On computing exact visual hulls of solids bounded by smooth surfaces. In *CVPR* (Vol. 1, pp. 156–161).

Margaritis, D., & Thrun, S. (1998). Learning to locate an object in 3d space from a sequence of camera images. In *ICML* (Vol. 1, pp. 332–340).

Matusik, W., Buehler, C., Raskar, R., Gortler, S., & McMillan, L. (2000). Image-based visual hulls. In *Siggraph* (Vol. 1, pp. 369–374).

Matusik, W., Buehler, C., & Mcmillan, L. (2001). Polyhedral visual hulls for real-time rendering. In *Proceedings of eurographics workshop on rendering* (Vol. 1, pp. 115–126).

Mittal, A., & Davis, L. S. (2003). M2tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, *51*(3), 189–203.

Otsuka, K., & Mukawa, N. (2004). Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles. In *CVPR* (Vol. 1, pp. 90–97).

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*, 7–42.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR* (Vol. 1, pp. 519–528).

Sinha, S., & Pollefeys, M. (2005). Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation. In *ICCV* (Vol. 1, pp. 349–356).

Slabaugh, G., Culbertson, B. W., Malzbender, T., Stevens, M. R., & Schafer, R. (2004). Methods for volumetric reconstruction of visual scenes. *International Journal of Computer Vision*, *57*, 179–199.

Snow, D., Viola, P., & Zabih, R. (2000). Exact voxel occupancy with graph cuts. In *CVPR* (Vol. 1, pp. 345–353).

Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *CVPR* (Vol. 2, pp. 246–252).

Takamatsu, J., Matsushita, Y., & Ikeuchi, K. (2008). Estimating camera response functions using probabilistic intensity similarity. In *CVPR* (pp. 1–8).

Yang, D., Gonzalez-Banos, H., & Guibas, L. (2003). Counting people in crowds with a real-time network of simple image sensors. In *ICCV* (Vol. 1, pp. 122–129).

Ziegler, R., Matusik, W., Pfister, H., & McMillan, L. (2003). 3D reconstruction using labeled image regions. In *EG symposium on geometry processing* (Vol. 1, pp. 248–259).