The OD Theory of TOD: The Use and Limits of Temporal Information for Object Discovery

Brandon C. S. Sanders and **Randal C. Nelson**
Department of Computer Science
University of Rochester
Rochester, NY 14627
[sanders,nelson]@cs.rochester.edu

Rahul Sukthankar
Compaq Research (CRL)
One Cambridge Center
Cambridge, MA 02142
rahul.sukthankar@compaq.com

Abstract

We present the theory behind TOD (the Temporal Object Discoverer), a novel unsupervised system that uses only temporal information to discover objects across image sequences acquired by any number of uncalibrated cameras. The process is divided into three phases: (1) Extraction of each pixel's *temporal signature*, a partition of the pixel's observations into sets that stem from different objects; (2) Construction of a global schedule that explains the signatures in terms of the lifetimes of a set of quasi-static objects; (3) Mapping of each pixel's observations to objects in the schedule according to the pixel's temporal signature. Our Global Scheduling (GSched) algorithm provably constructs a valid and complete global schedule when certain observability criteria are met. Our Quasi-Static Labeling (QSL) algorithm uses the schedule created by GSched to produce the maximally-informative mapping of each pixel's observations onto the objects they stem from. Using GSched and QSL, TOD ignores distracting motion, correctly deals with complicated occlusions, and naturally groups observations across cameras. The sets of 2D masks recovered are suitable for unsupervised training and initialization of object recognition and tracking systems.

Introduction

Computers capable of intelligent interaction with physical objects must first be able to discover and recognize them. "Object Discovery" (OD) is the problem of grouping all observations springing from a single object without including any observations generated by other objects (for an example see Figure 1). Because robust OD is a prerequisite for reasoning about physical objects, relationships, actions and activities, OD is of fundamental importance to AI systems seeking to interact with the physical world. A number of different approaches have been considered that make different assumptions about the world.

Static OD systems seek to discover objects in single images without using temporal information. Object recognizers may be used to discover known objects in static images (Papageorgiou & Poggio 2000; Schiele & Crowley 2000). The primary limitation of object recognizers is the often extensive training they require to discover objects. Static OD approaches that do not require an *a priori* model of each object of interest typically rely upon local homogeneity of color (Liu & Yang 1994), texture (Mao & Jain 1992), or a combination of these cues (Belongie *et al.* 1998). Because real objects are not visually homogeneous through space, traditional segmentation often returns pieces of objects or incorrectly groups parts stemming from multiple objects.

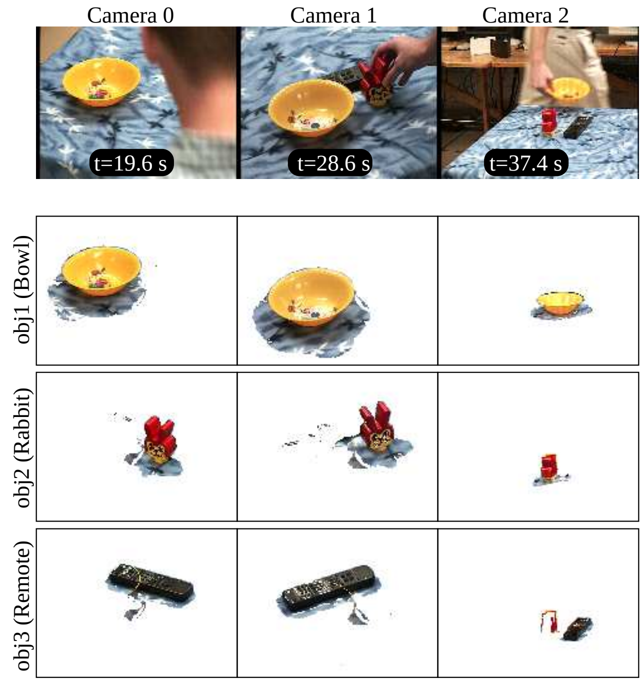

Figure 1: Sample OD results for TOD, a direct implementation of the theory presented in this paper. (*top*) Examples from sequences acquired by three different uncalibrated cameras. (*bottom*) The objects discovered. The complete remote is recovered even though it is partially occluded by either the bowl or the rabbit in every image in which cameras 0 and 1 observe it. Notably, temporal information alone is sufficient to group the pixels in and across the cameras.

Dynamic OD systems find objects that move independently in the world and so introduce temporal information into the mix. The discovery of moving objects typically depends upon spatial homogeneity of motion flow vectors

(Wang & Adelson 1994); sometimes this data is also combined with texture or color (Altunbasak, Eren, & Tekalp 1998). Dynamic OD systems often rely upon background subtraction (Toyama *et al.* 1999) to initially separate moving objects from a static background. Dynamic OD systems typically require high frame rates and cannot separate objects from the person manipulating them.

In this paper we present the theory behind *TOD*, the Temporal Object Discover. TOD is a system that uses temporal rather than spatial information to discover objects across multiple uncalibrated cameras. We decompose the problem of object discovery into three phases: (1) Generation of a temporal signature for each pixel; (2) Construction of a global schedule that satisfies the constraints encoded in the temporal signatures; (3) Explanation of each individual pixel's temporal signature in terms of a mapping from its observations to objects in the global schedule.

Because we do not use spatial information, our approach complements the existing body of segmentation work, most of which relies upon local spatial homogeneity of color, texture or optical flow. Despite ignoring spatial information, TOD achieves good results even on sequences having complex object occlusion relationships (see Figure 1). The advantages of our method include: (1) Human supervision is not required; (2) Low frame rates (*i.e.,* 1-5Hz) suffice; (3) Entire objects are discovered even in some cases where they are always partially occluded; (4) The approach scales naturally to and benefits from multiple uncalibrated cameras. The recovered multi-view 2D masks are suitable for unsupervised training and initialization of object recognition and tracking systems.

The remainder of the paper is structured as follows. First we introduce the quasi-static world assumed by TOD. Then we describe how each pixel's *temporal signature* is constructed. Following the section on signature construction, we introduce GSched, an algorithm that creates a valid and complete schedule of object lifetimes when certain observability criteria are met. We then present the QSL algorithm that solves the labeling problem using each pixel's temporal signature and the global schedule created by GSched. We conclude with a discussion of some limitations of temporal information and a short look at promising directions for future work.

TOD and the Quasi-static Model

In this section we define the quasi-static world model used throughout the remainder of the paper. This model is attractive because it imposes enough restrictions on the world to be theoretically treatable while maintaining practical application to real systems. The quasi-static model assumes that the only objects of interest are those that undergo motion on some time interval and are stationary on some other time interval (*i.e.,* objects that stay still for a while). Thus the quasi-static world model targets objects that are picked up and set down while ignoring the person manipulating them.[1]

[1]Of course, according to the quasi-static world model when a person is completely stationary he/she becomes an object of interest.

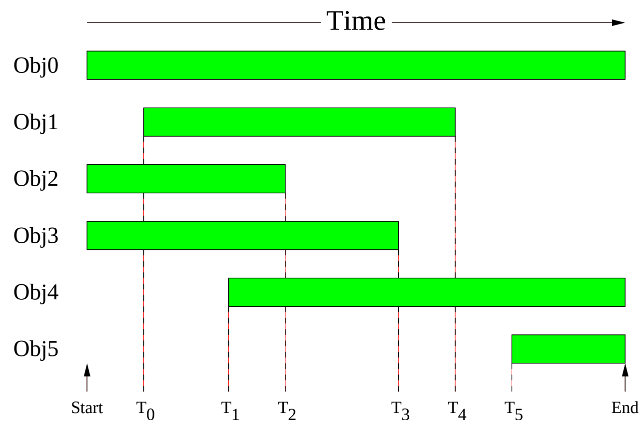

Figure 2: A global schedule consists of the lifetimes of a set of quasi-static objects and a special static background object (Obj0). The ordering of the object lifetimes in this figure is arbitrary and should not be interpreted as a layered representation containing occlusion information.

The following definitions will be used throughout the paper in connection with the quasi-static model:

Physical object: A chunk of matter that leads to consistent observations through space and time. Physical objects are objects in the intuitive sense. We define physical objects in order to contrast them with *quasi-static objects*. In the quasi-static world, a single physical object may be interpreted as any number of quasi-static objects. A physical object is *mobile* if it is observed to move in the scene.

Quasi-static object: The quasi-static world interpretation of a mobile physical object that is stationary over a particular time interval. For every interval on which a mobile physical object is observed to be stationary, the quasi-static world model interprets the physical object as a unique quasi-static object that arrives at the beginning of the stationary interval and departs at the end of the stationary interval. A single quasi-static object can only arrive once and depart once. We use the term *object* variously throughout the paper to refer to a physical object, a quasi-static object, and to a quasi-static object's entry in the global schedule. Where the usage of the word object is not clear from the context, we use a fully descriptive phrase instead.

Quasi-static object lifetime: The time interval over which a mobile physical object is stationary at a single location. When a mobile physical object m moves around the scene and is stationary at multiple physical locations, each stationary location i is interpreted as a separate quasi-static object o_i.

Global schedule: A set of quasi-static objects and their lifetimes (Figure 2).

Pixel visage: A set of observations at a given pixel that are interpreted as stemming from a particular quasi-static object. Each of a pixel's visages is disjoint with its other visages and forms a history of a particular quasi-static object's visual appearance through time according to the pixel. The pixel visage v is said to be *observed by* pixel p at time t if the observation made by p at t is in v. A pixel visage is *valid* if each of its observations stems from a single quasi-static object.

The quasi-static world model assumes that each pixel can reliably group observations that stem from a single quasi-static object. In other words, the observations belonging to one visage for a particular pixel are discriminable from the observations belonging to any other visage for that pixel. The next section presents the method we use to perform this grouping into visages. The following scheduling and labeling sections then describe how to determine the identity of the quasi-static object responsible for each visage.

Computing Temporal Signatures

In the first phase of object discovery, *signature extraction*, we start with a pixel's sequence of observations and partition them into pixel visages, sets of observations that all stem from a single object. A pixel's visages directly encode the temporal structure in the pixel's observation history in the form of a *temporal signature*. The set of temporal signatures gathered across all views will later be used in the *schedule generation* phase to hypothesize the existence of a small set of objects whose arrivals and departures explain the signatures. The hypothesized set of objects, or global schedule (Figure 2), is in turn used during the *labeling* phase to determine the mapping from observations to objects. Before describing our temporal signature generation algorithm, we first take a moment to define what we mean by *temporal signature* and several other related terms.

Stationary interval: A period of time during which every observation made by a given pixel stems from a single quasi-static object. A stationary interval is said to *belong to* the pixel visage that contains its observations. In Figure 3, the stationary intervals are labeled A through G.

Interruption: A non-stationary interval that comes between two stationary intervals belonging to the *same* pixel visage. In Figure 3, the gap between stationary intervals B and C and the gap between C and D and the gap between F and G are all interruptions.

Transition: An non-stationary interval that comes between two stationary intervals belonging to *different* pixel visages. Every transition contains either the arrival of an object or the departure of a different object. In Figure 3, the gaps between stationary intervals A and B, between D and E and between E and F are transitions.

Unambiguous transition: A transition in a signature that admits only one interpretation. If the object that supposedly arrived at a particular transition was previously observed at some time prior to the transition, then the object cannot have arrived during the transition. Similarly, if the object that supposedly departed at a particular transition is again observed at some time after the transition, then the object cannot have departed during the transition. In Figure 3 the first and third transitions are unambiguous because they cannot involve 0. The second transition is ambiguous because it could contain the arrival of 2 or the departure of 1.

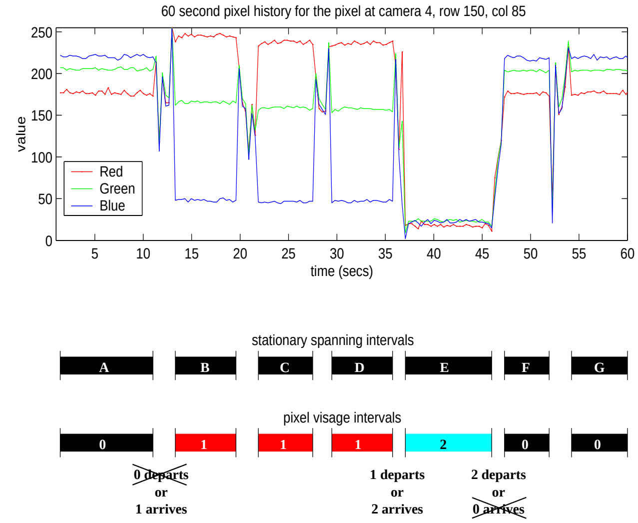

Figure 3: A pictorial walk-through of the constituents of a pixel's temporal signature starting from the pixel's sequence of observations and working down to through an unambiguous labeling of several transitions observed by the pixel.

Temporal signature: A pixel's temporal signature (Figure 3) encodes the temporal structure found in its observation history. This representation includes the transitions the pixel has witnessed as well as a pixel visage for each unique quasi-static object the pixel observes. In Figure 3, the stationary intervals for the three pixel visages are labeled 0 through 2 according to the visage observed on the interval.

Because we are interested in determining the utility of temporal information for OD, we ignore spatial information in every phase of the algorithm. This means that during the temporal signature generation phase, we assume that knowing the visual appearance of an object in one pixel provides zero information about the object's visual appearance in every other pixel. Our method for signature extraction depends upon the following definition of an atomic interval and involves several steps:

Atomic interval: An atomic interval A is any sequence of at least two observations $A = \langle x_i, x_{i+1}, \ldots, x_{i+n} \rangle$ such that the difference between the first observation's time-stamp and the last observation's time-stamp exceeds the minimum time for stability t_{min} and A does not contain a proper subsequence that also spans t_{min}.

Temporal signature construction

1. Check every *atomic interval* A for stationarity by verifying that every observation[2] $x_i \in A$ is close to every other observation $x_j \in A$ in visual appearance space (in

[2]The observations used to compute the temporal signatures are spatially averaged over a 3x3 region to remove high frequency spatial artifacts. This step is essential for real image sequences.

our current implementation we measure distance in RGB color space).

2. Group temporally overlapping atomic stationary intervals into *spanning stationary intervals*. Because we consider stationarity on atomic intervals rather than on spanning intervals, the visual appearance of an object in a pixel is allowed to change as long as the change is gradual (*e.g.*, movement of the sun across an outdoor scene).
3. Construct a pixel visage by collecting the observations from spanning stationary intervals that share the same visual appearance. To evaluate whether two spanning stationary intervals share the same visual appearance, we evaluate whether the temporally nearest ends of the two spanning intervals are close in RGB color space (using the means of the nearest atomic intervals).

Establishing a Global Schedule

In the second phase of object discovery, *schedule construction*, we use the set of temporal signatures gathered across all pixels in all views to hypothesize the existence of a small set of objects whose arrivals and departures satisfy the constraints of the signatures and thus explain them. This hypothesized set of objects and object lifetimes constitutes a global schedule (Figure 2). For a global schedule to be *valid*, the lifetime of each quasi-static object it contains must exactly match an interval on which some mobile physical object was stationary in the scene. To be *complete*, a global schedule must explain every transition observed by some pixel. A valid and complete global schedule is a correct schedule in the intuitive sense.

Each pixel's temporal signature places constraints upon the global schedule (see Figure 4). In order to be valid and complete, a global schedule must all of these constraints. In general, the constraints from temporal information alone are not enough to completely determine the schedule. Specifically, temporal information cannot determine whether an object o has arrived or departed unless o has both arrived and departed during the period of observation. Even though temporal information cannot, in general, completely determine a global schedule, many cases exist where temporal information does suffice. In fact, if the following observability criteria are met, the Global Scheduling (GSched) algorithm we present later in this section is guaranteed to find a complete and valid global schedule using only temporal information.

GSched Observability Criteria: Temporal information alone is sufficient to construct a valid and complete global schedule if the following observability criteria are met:

1. **Valid visages criterion:** Every pixel visage is valid. In other words, within each pixel, observations of each object are correctly grouped. In our implementation this generally implies that all of a stationary pixel's observations of a stationary object lie in a small region of RGB space.
2. **Temporally discriminability criterion:** Across all pixels, every arrival and departure event is temporally discriminable from every other event. Essentially, when the transition intervals from all the signatures are considered together, each event must clearly stand out as separate from the others.
3. **Clean world criterion:** Every quasi-stationary object both arrives and departs.[3]
4. **Observability criterion:** For every object o, some pixel p observes both the arrival and departure of o (neither event is hidden by some other object) and furthermore p is able to unambiguously identify either o's arrival or o's departure (Figure 3). This criterion becomes more likely to be met as the number of different viewpoints of each object increases.

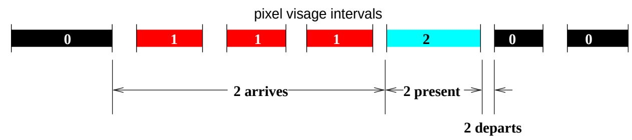

Figure 4: The arrival and departure times for a visage v are constrained by observation of another visage that bounds v. In this example, 0 bounds both 1 and 2. The constraints for 2 are shown. To explain 2, the global schedule must contain an object that arrives during 2's arrival interval and departs during 2's departure interval.

The criteria listed above lead directly to the straightforward *global scheduling algorithm* presented below.

Global Scheduling (GSched) algorithm Given that the GSched observability criteria listed above are met, the following algorithm establishes a valid and complete global schedule:

1. Build a global list E of unambiguous events by creating a set of events that explains each unambiguous transition. The unambiguous transitions are processed in order from shortest transition to longest. If no event in E explains an unambiguous transition when the transition is processed, a new event is added to E that does explain the transition. If the *observability criterion* is met, E will contain at least one unambiguous event (arrival or departure) for every quasi-static object in the scene.
2. For each event e in E:
   (a) Remove e from E.
   (b) If e is the *arrival* of an object o, find the corresponding *departure* of o by determining the latest time t at which o is observed by some pixel that observes e. If some event $e' \in E$ matches t, then e' must be the departure of o according to the *temporal discriminability criterion*. If e' exists, remove it from E so that it is not processed twice. Create a global object hypothesis g_i with lifetime spanning from the time of arrival (determined from e) to the time of departure t. Enter g_i into the the global schedule S.

[3]The background is treated specially and is the union of all objects that never arrive nor depart.

(c) If e is the *departure* of an object o, find the corresponding *arrival* of o by determining the earliest time t at which o is observed by some pixel that observes e. If some event $e' \in E$ matches t, then e' must be the arrival of o according to the *temporal discriminability criterion*. If e' exists, remove it from E so that it is not processed twice. Create a global object hypothesis g_i with lifetime spanning from the time of arrival t to the time of departure (determined from e). Enter g_i into the the global schedule S.

Steps *a* and *b* are valid because some pixel p' has observed both the arrival and departure of o (according to the *observability criterion*). Thus p' is guaranteed to have observed e regardless of whether e is an arrival or departure, and p' has observed o at least as early and at least as late as any other pixel.

Mapping Observations to Objects

During the labeling phase of object discovery, we use the schedule generated by the GSched algorithm and the temporal signature computed during the first phase to map the observations in each pixel visage to the objects in the schedule that those observations could have stemmed from. This labeling of observations is the ultimate goal of object discovery. Once each observation has been mapped to the objects that could have given rise to it, we can easily assemble multi-view 2D masks of each object from the observations attributed to the object. In this section we describe our Quasi-Static Labeling (QSL) algorithm for solving the mapping problem, and argue that (1) Given a valid and complete global event schedule such as that returned by GSched, each pixel's observation labeling problem is independent of every other pixel's observations; (2) The QSL algorithm produces the maximally-informative mapping of a pixel's observations onto the objects they stem from. We begin this section by defining the observation labeling problem.

Observation labeling problem Given a pixel p and a valid and complete global schedule, determine for each of p's visages v_i the smallest set of quasi-static objects in the schedule guaranteed to contain the actual quasi-static object that generated p's observations of v_i.

Because we do not assume an object is visually homogeneous through space, we cannot link observations across pixels based on similarity of color and/or texture. Rather, to conclude that two pixels have observed the same object o, both pixels must have made observations that are temporally consistent with the arrival and departure of o. The only information salient to this decision are the times at which o arrives and departs. For every object, these arrival and departure times are contained in the global schedule. Thus, given the complete global schedule, each pixel's labeling problem is independent of every other pixels' observations.

The independence of labeling problems has several important consequences. First, any labeling algorithm that only uses temporal information may be easily parallelized simply by running multiple copies of a single labeler on subsets of the pixels. Second, since each pixel's labeling problem is independent of the pixel's spatial location, we may treat every pixel identically regardless of its physical location. In other words, it doesn't matter what camera, row, and column a pixel comes from. Finally, this independence property allows us to show that QSL generates the globally maximally-informative mapping from observations to objects simply by showing that QSL correctly solves the pixel labeling problem for any given pixel taken in isolation.

The remainder of this section is written from the perspective of a single pixel's labeling problem and assumes the existence of a valid and complete global schedule containing all known objects. We begin by defining several terms used to describe the QSL algorithm. We then introduce the inference rules that drive QSL and show that each inference rule leads to a valid mapping according to the constraints of the quasi-static world model. Finally, we introduce the QSL algorithm and argue that it recovers the maximally informative mapping from observations to objects. To describe the inference rules and the labeling algorithm we need the following definitions:

Contemporaneous object set: A contemporaneous object set X is a set of objects such that each object $o \in X$ is present in the scene at some time t. A *maximal* contemporaneous object set X_t^* is the set of *all* objects present in the scene at time t.

$$X_t^* = \{o : o \text{ is present at time } t\}$$

Intersection set: For a pixel visage v, v's intersection set I_v is the set of all objects such that each object is present at *every* time at which v is observed.

$$I_v = \bigcap_t X_t^* \qquad t : v \text{ is observed at time } t$$

Union set: For a pixel visage v, v's union set U_v is the set of all objects such that each object is present at *some* time at which v is observed.

$$U_v = \bigcup_t X_t^* \qquad t : v \text{ is observed at time } t$$

Bounding visage: For pixel visages v and b, b is a bounding visage of v if the following hold:

1. b is observed sometime *prior to* every observation of v;
2. b is observed sometime *after* every observation of v.

Bounded set For a pixel visage v, v's bounded set B_v is the set of all objects such that each object is present at *every* time that v is observed, and no object is present at *any* time at which a bounding visage of v is observed. In other words, a visage's bounded set contains every object whose lifetime is consistent with the visage's constrained arrival and departure intervals (see Figure 4).

$$B_v = I_v - \bigcup_b U_b \qquad b : b \text{ is a bounding visage of } v$$

Given the quasi-static assumptions, the object bounded set B_v for a visage v contains the actual object observed by v.

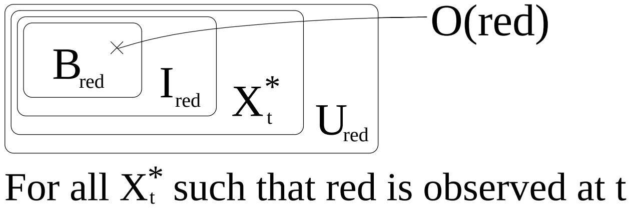

Figure 5: The relationship between the sets for a visage red. Because $O(red)$ is in B_{red}, $O(red)$ is also in each of the other sets. Similarly, since $O(red)$ is in front of every object in U_{red}, $O(red)$ is also in front of every object in each of the other sets.

Object function: The object function $O(v) = o$ maps a visage v onto an object o. This function represents abstractly the true state of the world. The goal of the labeling process is to find the smallest set of candidate objects C such that $O(v) \in C$ is true given that the model assumptions hold. In some cases it is not physically possible to narrow C down to a singleton.

Front function: The front function $F(C) = o$ for a pixel p maps a set of candidate objects C onto the object $o \in C$ that is in front of the other objects. The front object o is said to *occlude* the other objects in C. $F(C) = o$ is unique for all sets C such that for every other $o' \in C$, at some time t, both o and o' are simultaneously present and p observes o at time t. Any subset of the union set for a visage v that contains $O(v)$ meets this condition. Like the object function $O()$, $F()$ represents abstractly the true state of the world, not what we know about it.

The following lemmas and theorems provide the foundation for the labeling algorithm. To make the discussion easier to follow, we use color names to refer to particular pixel visages.

Lemma 1 *For any pixel visage red, and any candidate object set C such that every object in C is present at some time when red is observed and $O(red) \in C$, $O(red) = F(C)$ (i.e., $O(red)$ is in front of every other object in C).*

This follows directly from the physics of the quasi-static world.

Lemma 2 *According to lemma 1, the following four statements are all true:*

1. $O(red) = F(B_{red})$;
2. $O(red) = F(I_{red})$;
3. $O(red) = F(X_t^*)$, $\quad \forall t:$ *red is observed at t;*
4. $O(red) = F(U_{red})$.

These four statements follow directly from lemma 1 and the relations: $O(red) \in B_{red} \subseteq I_{red} \subseteq X_t^* \subseteq U_{red}$, for all t such that red is observed at t. The relationships between the sets and the object and front functions is illustrated in figure 5. These relationships follow directly from the definitions of the sets and the object and front functions.

The following QSL Theorem is central to the QSL algorithm. In essence, the QSL Theorem provides a general rule that allows us to use one pixel visage's union set (*e.g.*, U_{blue}) to rule out candidates for $O(red)$ for some other visage red. Iterative invocation of this theorem forms the heart of QSL and allows us to find the most informative mapping from visages to objects.

Theorem 3 ***QSL Theorem*** *Given distinct visages $red, blue$ and contemporaneous subsets X_{red}, X_{blue} such that $O(red) = F(X_{red})$ and $O(blue) = F(X_{blue})$:*

$$\begin{aligned} X_{blue} \subset U_{red} \quad &\Rightarrow \quad O(red) \textit{ occludes } O(blue) \\ &\Rightarrow \quad O(red) = F(X_{red} - U_{blue}) \end{aligned}$$

Proof The gist of the proof hangs upon determining when the front object of one set of objects occludes the front object of another set of objects.

1. $\mathbf{X_{blue} \subset U_{red} \Rightarrow O(red) \notin X_{blue}}$. The front object in U_{red} is $O(red)$. Whenever a subset of U_{red} contains $O(red)$, the front object of the subset is $O(red)$. Since X_{blue} is a subset of U_{red} and the front object of X_{blue} is $O(blue)$ not $O(red)$, X_{blue} cannot contain $O(red)$.
2. $\mathbf{O(red) \notin X_{blue} \Rightarrow O(red)}$ **occludes** $\mathbf{o}$ **for every object** $\mathbf{o \in X_{blue}}$. According to the definition of the union set, every object in U_{red} is present at some time when red is observed. Thus $X_{blue} \subset U_{red}$ guarantees that red is observed at some time t when o is present. Since $O(red)$ is the object visible whenever red is observed, $O(red)$ is in front of o at time t.
3. $\mathbf{O(red)}$ **occludes** $\mathbf{o}$ **for every object** $\mathbf{o \in X_{blue} \Rightarrow O(red)}$ **occludes** $\mathbf{O(blue)}$. $O(blue) \in X_{blue}$ satisfies the previous step.
4. $\mathbf{O(red)}$ **occludes** $\mathbf{O(blue) \Rightarrow O(red) \notin U_{blue}}$. Since $O(red)$ is in front of $O(blue)$, $O(red)$ cannot be any object that is ever present when $blue$ is observed. Since every object in U_{blue} is present at some time when $blue$ is observed, no object in U_{blue} can be $O(red)$.
5. $\mathbf{O(red) \notin U_{blue} \wedge O(red) = F(X_{red}) \Rightarrow O(red) = F(X_{red} - U_{blue})}$. ∎

The definitions and results presented above allow us to state the Quasi Static Labeling (QSL) algorithm succinctly. The algorithm maintains a collection **R** of statements of the form $O(v_i) = F(C_i)$, one for each visage v_i, where the elements of a set C_i essentially encode a set of candidates for the object that maps to visage v_i, as determined by QSL at some point in the algorithm. We start with an initial set of true statements and attempt to produce new, smaller true statements by applying the QSL Theorem. The ultimate goal is to obtain for each visage the true statement with the smallest possible front function subset argument.

Quasi-Static Labeling (QSL) algorithm Given a complete global schedule, for each pixel p and its set of pixel visages V_p:

1. For each pixel visage $v \in V_p$ find v's union set U_v.
2. For each pixel visage $v \in V_p$ find v's bounded set B_v, and add the statement $O(v) = F(B_v)$ to the collection **R**. By Lemma 2 these are all true statements.

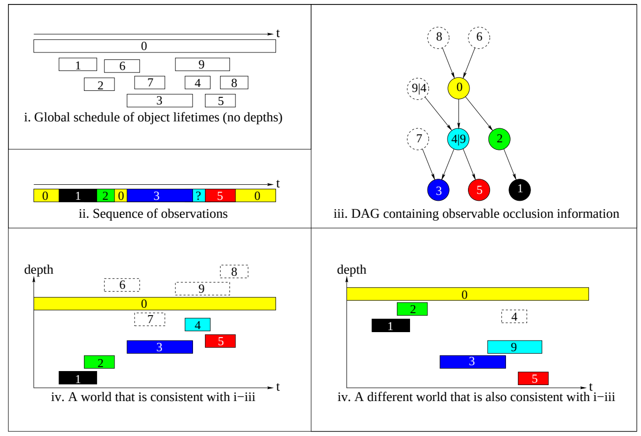

Figure 6: Given a global schedule (*i*), observation of a pixel's visages (*ii*) determines a partial depth ordering of the objects that map to the visages. This partial ordering can be represented as a directed acyclical graph (*iii*). The objects with the dashed boundaries are not directly observed by the pixel and so may be occluded or may simply not be present in the part of the scene observed this pixel. When a directed path between two objects exists, the object at the end of the path is said to be *in front* of the object at the start of the path while the object at the start of the path is said to be *behind* the object at the end of the path. If two such objects are present in the scene simultaneously, the in-front object is said to *occlude* the behind object. Often (as in this case) it is possible to determine the occlusion order of two pixel visages without necessarily being able to uniquely determine the global object that one or the other visage maps to.

3. Repeatedly apply the QSL Theorem to appropriate pairs of statements in **R** to shrink the subset argument of one of the statements. Repeat until no further applications are possible.

If the QSL Theorem applies to a pair of statements, it equally applies to the pair of statements if either statement's subset argument shrinks. Thus the result is independent of the order in which the transformations are applied. Because the size of an argument subset is always smaller than one of the parent statements and the size of these subsets must remain positive, the algorithm is guaranteed to terminate. The final candidate label set C_v^* for each pixel visage v can be read from the subset argument for v's statement $O(v) = F(C_v)$ in **R**.

If there are m visages in the temporal signature and n objects in the global schedule ($m \leq n$), the number of times the QSL Theorem can be invoked to remove objects from subset arguments is bounded by mn, the maximum number of objects in all subset arguments in **R**. The number of comparisons between subset arguments and union sets required to find a match for the QSL Theorem is m^2 in the worst case. Each set comparison involves at most n element comparisons. This gives QSL a worst-case runtime complexity of m^3n^2.

Given a complete and valid global schedule S and a pixel p having only valid visages, the labeling the QSL algorithm returns is maximally-informative in the sense that it fully preserves and utilizes the following observable properties of the quasi-static world. Out of space considerations, we refer you to (Sanders, Nelson, & Sukthankar 2002) for the proofs of these theorems.

Theorem 4 *For any object $o \in S$ and any two distinct visages v, v' observed by p, the QSL algorithm never assigns $O(v) = O(v') = o$.*

Theorem 5 *For any visage v observed by p and any two distinct objects $o, o' \in S$, the QSL algorithm never assigns $O(v) = o = o'$.*

Theorem 6 *For any object $o \in S$, if p observes o's arrival, the QSL algorithm correctly assigns $O(v) = o$ to the visage v observed by p immediately after the arrival. Likewise if p observes o's departure, the QSL algorithm correctly assigns $O(v) = o$ to the visage v observed by p immediately before the departure.*

Theorem 7 *For any visage v observed by p and object $o \in S$, the QSL algorithm determines $O(v) \neq o$ if there exists outside of o's lifetime a time t at which p observes v.*

Theorem 8 *For each of p's visages v_i, the QSL algorithm determines $O(v_i) \neq o$ for every object $o \in S$ that is* in front *of $O(v_i)$ in every world configuration that is consistent with S and each of p's visages.*

Conjecture 9 *For each of p's visages v_i, the QSL algorithm determines $O(v_i) \neq o$ for every object $o \in S$ that is* behind *$O(v_i)$ in every world configuration that is consistent with S and each of p's visages.*

The QSL algorithm uses the QSL theorem to implicitly generate a directed acyclical graph encoding all occlusion relationships between schedule objects that are observable given the schedule and a particular pixel's signature (Figure 6). Once the observations in the sequence have been mapped to the objects they stem from, it is trivial to assemble multi-view 2D masks of the objects from the observations attributed to them. Since the observations used to construct the masks may come from any time during the sequence, a complete object mask of an object that is never entirely visible at at any one time can be created from observations made at various times when different parts of the object were visible.

Limitations of Temporal Information

The quasi-static world model assumes that a given stationary physical object looks the same through time from each vantage point that observes it. In practice, when other objects arrive or depart from the scene, the lighting conditions for a stationary object (*e.g.*, a toy rabbit) may be affected. Even if the objects arriving and leaving do not occlude the part of the rabbit a pixel observes, shadows and reflections from the other objects can significantly alter the rabbit's visual appearance in the pixel. However, not all representations of visual appearance are equally susceptible to shadows and reflections. For example, an HSI color space may allow easier rejection of appearance changes due to shadows than the corresponding RGB color space.

While temporal information alone is often enough to both construct a global schedule and map observations to objects using that schedule, there are situations for each of these tasks in which the problem cannot be completely solved using temporal information alone. Consider the scheduling problem. In many real world situations, the *clean world* criterion is not satisfied by some objects that either only arrive or only depart. As was mentioned earlier, temporal information by itself cannot determine whether an event is the arrival or departure of an object o, unless o has also respectively departed or arrived. While temporal information cannot resolve these tricky events, several straightforward and robust spatial methods based upon edge features may be used in concert with the temporal information to finish the job.

As with construction of a global schedule, mapping observations to objects cannot always be completely solved using temporal information alone. Certain world configurations are inherently ambiguous with respect to temporal information alone. Instead of a single candidate quasi-static object per visage, some visages have a set of possible objects they could map to. Even in these difficult situations, QSL determines the smallest set of candidates that fully covers all possible world configurations. These sets, as provided by QSL, could be combined with spatial techniques, such as connected components, to finish constraining the assignment of observations to objects.

Conclusion

TOD, as described by the theory in this paper, ignores distracting motion, correctly deals with complex occlusions, and recovers entire objects even in cases where the objects are partially occluded in every frame (see Figure 1 for example results). Because we do not use spatial information to perform our clustering, our approach is significantly different from and complements traditional spatially based segmentation algorithms. The advantages of our method include: (1) Human supervision is not required; (2) Low frame rates (*i.e.*, 1-5Hz) suffice; (3) Entire objects are discovered even in some cases where they are always partially occluded; (4) The approach scales naturally to and benefits from multiple uncalibrated cameras. Since our approach is well suited to train and initialize object recognition and tracking systems without requiring human supervision, our method represents significant progress toward solving the object discovery problem.

A few promising directions for future work include: (1) Using the 2D masks generated by TOD to train an object recognizer automatically; (2) Integrating TOD with a spoken language system where TOD is used to perceptually ground the nouns; (3) Evaluating a variety of techniques for generating temporal signatures that allow the *distinct visages* and *temporal discriminability* observability criteria to be weakened; (4) Combining temporal and spatial information in a unified framework that removes the requirement of *temporal discriminability* altogether; (5) Extending TOD to run online by causing GSched and QSL to periodically commit to their interpretations of the sequences; (6) Converting the deterministic scheduling and labeling phases into probabilistic versions; (7) Integrating the currently separate scheduling and labeling phases into a single phase. More details are in (Sanders, Nelson, & Sukthankar 2002) available from `http://www.cs.rochester.edu/~sanders`.

Acknowledgments

This work funded in part by NSF Grant EIA-0080124, NSF Grant IIS-9977206, Department of Education GAANN Grant P200A000306 and a Compaq Research Internship.

References

Altunbasak, Y.; Eren, P. E.; and Tekalp, A. M. 1998. Region-based parametric motion segmentation using color information. *GMIP* 60(1).

Belongie, S.; Carson, C.; Greenspan, H.; and Malik, J. 1998. Color- and texture-based image seg. using EM and its app. to content-based image retrieval. In *Proc. ICCV*.

Liu, J., and Yang, Y. 1994. Multiresolution color image segmentation. *IEEE PAMI* 16(7).

Mao, J., and Jain, A. K. 1992. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *PR* 25(2).

Papageorgiou, C., and Poggio, T. 2000. A trainable system for object detection. *IJCV* 38(1).

Sanders, B. C. S.; Nelson, R. C.; and Sukthankar, R. 2002. Discovering objects using temporal information. Technical Report 772, U. of Rochester CS Dept, Rochester, NY 14627.

Schiele, B., and Crowley, J. L. 2000. Recognition without correspondence using multidimensional receptive field histograms. *IJCV* 36(1).

Toyama, K.; Krumm, J.; Brumitt, B.; and Meyers, B. 1999. Wallflower: Principles and practice of background maintenance. In *Proc. ICCV*.

Wang, J. Y. A., and Adelson, E. H. 1994. Representing moving images with layers. *IEEE Trans. on Image Proc. Special Issue: Image Sequence Compression* 3(5).