Incorporating Object Tracking Feedback into Background Maintenance Framework

Leonid Taycher, John W. Fisher III and Trevor Darrell
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, 02139

Abstract

Adaptive background modeling/subtraction techniques are popular, in particular, because they are able to cope with background variations that are due to lighting variations. Unfortunately these models also tend to adapt to foreground objects that become stationary for a period of time; as a result such objects are no longer considered for further processing. In this paper, we propose the first (to our knowledge) statistically consistent method for incorporating feedback from high-level motion model to modify adaptation behavior. Our approach is based on formulating the background maintenance problem as inference in a continuous state Hidden Markov Model, and combining it with a similarly formulated object tracker in a multichain graphical model framework. We demonstrate that the approximate filtering algorithm in such a framework outperforms the common feed-forward system while not imposing a significant extra computational burden.

1. Introduction

Background subtraction is a first step in many object tracking applications. It is used to determine likely locations of objects of interest (foreground objects) by comparing a newly acquired frame with an internally maintained model of the scene without objects of interest (background). In this work, we consider one of the most popular classes of background maintenance systems, so called adaptive models [8, 4, 10, 2]. Such models are able to adjust to scene changes due to causes other than objects of interest (e.g., lighting variations).

Background models are usually designed to be task independent, and this often means that they can use very little high-level information. While region-based reasoning may be utilized at every individual frame [8, 10], temporal consistency is usually exploited only on a per-pixel basis. This limitation can cause the scene model to adapt to foreground objects that remain stationary for extended periods of time. After these objects "fade" into the background, their locations are no longer considered as regions of interest.

Several approaches to incorporating information about foreground objects into background maintenance have been proposed. They may be broadly separated into two categories: probabilistic frameworks that *jointly* model scene and foreground object evolution [5, 11], and systems consisting of separate modules for scene modeling and high-level inference (e.g., object tracking) [8, 4, 10]. Adjustments to the background model in modular systems depend on heuristic-based feedback from the higher-level modules.

In this paper, we propose the first, to our knowledge, approach that incorporates background modeling and object tracking in a unified statistical framework, while still enabling efficient modular implementation. Our approach is based on the observation that both background maintenance and object tracking may be formulated as state estimation in dynamic Bayesian networks representing generative models (see section 3). Each generative model is approximate. The background model models the underlying scene but is agnostic about pixels generated by the moving objects. On the other hand, the object tracker models foreground pixels but not the rest of the image.

These models have different failure modes: the background model fails when foreground pixel values are close enough to the expected background, and the tracker fails when the background contains patterns similar to ones expected for the objects being tracked. We would like to pool knowledge from both models to improve overall performance.

Combining information from these models is not straightforward, since each model uses a different state representation. We address this issue by introducing a latent appearance representation (pixel value and the associated foreground-background label), shared between both models, and combining models at this representation level. The interconnection allows information from one chain to influence the other one during the inference process, effectively serving as a data association filter. The resulting graphical model is loopy, which makes inference complicated in general, but we propose an approximate filtering algorithm for our framework based on sequentially applying Belief Prop-

agation to acyclic subgraphs of the loopy model. This approach is based on our prior work on multichain models in the articulated-body tracking domain [9], reviewed in section 3 as applicable to the background maintenance.

We demonstrate that the foreground labels and background model estimated by the proposed multichain model compare favorably with output of stand-alone background subtraction system while not incurring significant computational cost.

2. Related Work

While many approaches to adaptive background modeling have been proposed, it remains an active research area. Several methods have been proposed that incorporate background estimation and object tracking in a single monolithic system [5, 11], but most systems take a modular approach that allows using a single background subtraction subsystem in different applications.

Stand-alone background subtraction algorithms assign background/foreground labels based on the history of the local measurements in a particular location. Popular modeling techniques may be separated into two broad classes, parametric and nonparametric [1]. Nonparametric models [10, 12] use previously observed frames directly, and consider a pixel to belong to the foreground if its value is different from a sufficient number of stored values. Parametric models maintain a representation of pixel value probability distribution (such as a mixture of Gaussians in [8]) that is recursively updated at every frame.

Local measurements, such as depth [4] and spatial and temporal gradients [7] have been used in addition to raw intensity values to improve segmentation.

While methods have been proposed for using high-level information to handle global changes (e.g., lights being switched on and off) [3], we are not aware of statistically consistent approaches to incorporating temporal information from object tracking into background modeling.

3. Combining Background and Tracking Inference

Consider a background maintenance system, similar to that described in [8]. As has been stated, at every timestep it has two tasks: to assign each pixel in the image with a probability of belonging to the background (foreground) class, and to modify the internal representation of the scene based on the current input. Its operation may be described as inference (filtering) in the dynamic Bayesian network shown in Figure 1(a).

This network represents a *generative* model of image formation as follows: first the background model, M^t, is predicted based on the model at the previous timestep, M^{t-1}, and transition probability $p(M^t|M^{t-1})$. A binary background label, B_j^t, is generated according to the prior probability, $P(B_j^t)$, for every pixel j. The latent pixel value, L_j^t, is generated according to the predicted model, M^t, if the pixel belongs to background ($B_j^t = 1$) and by a uniform distribution otherwise. The value of L_j^t contaminated by observation noise is then observed as I_j^t. The posterior background label probability for every pixel and the updated model may be computed using standard inference techniques. It may be shown that the model update rules in such methods as [8] may be derived in this manner.

Note that while per-pixel models (M_j^t) are usually used, this is not assumed. In the following discussion, we use the notation $I^t = \cup_j I_j^t$ to represent the complete observed image, $B^t = \cup_j B_j^t$ for the background probability image, etc.

To simplify derivations, we use a slightly modified generative model shown in figure 1(b), where $F_j^t = (L_j^t, B_j^t)$ is the instantaneous pixel representation, $p(F_j^t|M^t) = p(L_j^t, B_j^t|M^t) = p(L_j^t|M^j, B_j^t)p(B_j^t)$, and $p(I_j^t|F_j^t) = p(I_j^t|L_j^t)$.

As mentioned, this generative model represents the evolution of the scene only approximately, since while it models background pixels, it makes an (incorrect) assumption that the foreground pixels are generated by a uniform distribution. Thus temporal dependency between foreground pixel locations and values is not modeled, and the independence assumptions made in DBN do not hold. The inference algorithm assumes that joint distribution of the state and the appearance conditioned on all previous observations is

$$p(F^t, M^t|I^{0..t-1}) = p(F^t|M^t) \int dM^{t-1} p(M^t|M^{t-1}) p(M^{t-1}|I^{0..t-1}), \quad (1)$$

which is not generally true, since

$$p(F^t, M^t|I^{0..t-1}) = q(F^t, M^t; I^{0..t-1}) \int dM^{t-1} p(M^t|M^{t-1}) p(M^{t-1}|I^{0..t-1}) \quad (2)$$

$$q(F^t, M^t; I^{0..t-1}) \neq p(F^t|M^t).$$

Modeling the dependency between F^t and prior observations, which that remains unrepresented by this generative model, would allows for better estimation of both the background model and the background/foreground labels. When a model of foreground object motion is available (e.g., when the output of the background subtraction system is used for object tracking), we would like to incorporate it into our inference algorithm.

Object tracking systems are often described as filtering in the generative model shown in figure 1(c). At time t, T^t describes the positions and appearances of moving objects. The state is evolved according to the motion model

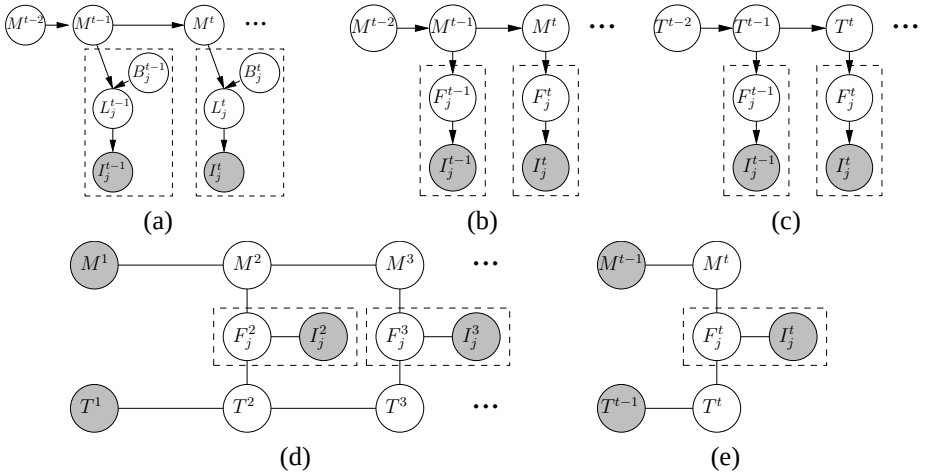

Figure 1: Combining background maintenance and object tracking models. (a) A generative model used for background maintenance. At time t, pixel j belongs to the background if $B_j^t = 1$. In this case its latent value, L_j^t, is generated according to $p(L_j^t|M^t)$, where M_j^t are the sufficient statistics of the scene background distribution. Otherwise the latent value is generated from a uniform distribution. L_j^t contaminated by noise is observed as I_j^t. Nodes enclosed in dashed rectangles are duplicated for every j. (b) The intermediate feature representation, $F_j^t = (L_j^t, B_j^t)$, $p(F_j^t|M^t) = p(L_j^t|M^j, B_j^t)p(B_j^t)$. (c) Generative model used for object tracking. The state T^t contains both spatial and appearance information about moving objects. If a pixel belongs to an object, then $B_j^t = 0$, and L_j^t is set depending on the object appearance. Otherwise $B_j^t = 1$ and L_j^t is generated by a uniform distribution. (d) Combined model with potentials corresponding to the conditional probabilities in the individual models (e.g., $\phi(M^t, M^{t-1}) = p(M^t|M^{t-1})$, etc.). (e) A tree-shaped subgraph on which a single step of approximate inference is performed. The marginal distributions, $p(M^{t-1}|I^{0..t-1})$ and $p(T^{t-1}|I^{0..t-1})$, have been computed at the previous iteration, and are not modified; I_j^t are observed.

$p(T^t|T^{t-1})$. If a pixel belongs to an object, then $B_j^t = 0$ and L_j^t is set depending on the object appearance. Otherwise $B_j^t = 1$ and L_j^t is generated by a uniform distribution.

We can now define an approximation to $q(F^t, S^t; I^{0..t-1})$ that combines information from both systems via a product

$$\hat{q}(F^t, M; I^{0..t-1}) \propto p(F^t|M^t) \int dT^t p(F^t|T^t) p(T^t|I^{0..t-1}). \quad (3)$$

This captures the desired property that an individual model should consider only those values of appearance that are assigned significant prior probability by all models. That is, the background model should not be adapted to pixels that the tracking system predicts to be generated by the foreground objects, and visa versa, pixels that are predicted to belong to background should not be considered by the tracker.

The resulting system may be represented by the model shown in Figure 1(d). The potentials in this undirected model are defined based on conditional distributions from constituent models (that is $\phi(M^t, M^{t-1}) = p(M^t|M^{t-1}), \phi(F^t, M^t) = p(F^t|M^t)$, etc.).

3.1. Inference in the Dual-Chain Model

Single-chain models are popular because there are efficient inference algorithms for them. While the proposed dual-chain model is loopy (Figure 1(d)) and exact inference is complicated in such models, we take advantage of the fact that we are interested only in marginal distributions for the state nodes to propose an efficient algorithm for *filtering* in such a model.

Consider the model in Figure 1(e). At time $t = 1$, we are concerned with nodes with times (superscripts) $t \leq 1$. If the initial states M^0 and T^0 are independent (as shown), then the resulting subgraph is a tree, and we can use standard Belief Propagation techniques [6] to compute exact marginal distributions at state nodes M^1 and T^1.

$$p(M^1|I^1) = \frac{1}{Z} \int dM^0 \phi(M^1, M^0) p(M^0) \int dF^1 \left[\phi(F^1) \phi(F^1, M^1) \int_{T^1} \left[\phi(F^1, T^1) \int T^0 \phi(T^1, T^0) p(T^0) \right] \right], \quad (4)$$

where $\phi(F^1) = \phi(I^1, F^1)$ (the equivalent expression of $p(T^1|O_1)$ is not shown).

	Sequence 1 (2478 frames)			Sequence 2 (2096 frames)		
Error Class	1	2	3	1	2	3
Stand-alone background subtraction	830	2861	93	155	877	50
Dual-chain model	688	0	0	175	0	0

Figure 2: Quantitative evaluation of background subtraction performance on PETS 2001 image sequences. Three error classes were differentiated. 1: no foreground components corresponding to a **pedestrian** have been detected. 2: no foreground components corresponding to a **vehicle** have been detected. 3: foreground component detected when no foreground object is present. Note that the number of class 2 errors for the first sequence includes 1760 errors due to a vehicle that remained stationary from frame 690 to the end of the sequence, and was incorporated into the background model at frame 719. The total number of errors is greater than the number of frames since more than one vehicle was undetected at some frames (cf. frame 2400 in Figure 3) See the text for more details.

Filtering at the next timestep ($t = 2$) is more complex since the model now contains loops and the exact inference would require representing the joint $p(M^1, T^1|I^1)$:

$$p(M^2|I^1, I^2) = \frac{1}{Z}\int dF^2\left[\phi(F^2)\phi(F^2, M^2)\int dT^2\left[\phi(F^2, T^2)\int dM^1 dT^1 \phi(M^2, M^1)\phi(T^2, T^1)p(M^1, T^1|I^1)\right]\right]. \quad (5)$$

In order to simplify computations, we approximate the joint distribution, $p(M^1, T^1|I^1)$ with a product, $q(M^1)q(T^1)$. It is easily shown that the best such approximation (in the KL-divergence sense) is the product of marginal distributions, $p(M^1|I^1)$ and $p(T^1|I^1)$. Substituting $p(M^1|I^1)p(T^1|I^1)$ for $p(M^1, T^1|I^1)$ in Equation 5, we obtain an approximate inference equation:

$$p(M^2|I^1, I^2) \approx \frac{1}{Z}\int dM^1\phi(M^2, M^1)p(M^1|I^1)\int dF^2\left[\phi(F^2)\phi(F^2, M^2)\int dT^2\left[\phi(F^2, T^2)\int dT^1\phi(T^2, T^1)p(T^1|I^1)\right]\right]. \quad (6)$$

The similarity between Equations (4) and (6) suggests an approximate filtering algorithm that estimates marginal distributions of the state variables by recursively applying Belief Propagation to acyclic subgraphs of the form shown in Figure 1(e), using the marginal state distribution obtained at time $t-1$ as priors at time t.

At each timestep, an approximate inference may be performed using message passing in 3 steps.

1. Predict the current background model and positions of the foreground objects based on the previous estimates by computing messages $\mu_{M^{t-1}\to M^t} = \int dM^{t-1}\phi(M^t, M^{t-1})p(M^{t-1}|I^{0..t-1})$ and $\mu_{T^{t-1}\to T^t} = \int dT^{t-1}\phi(T^t, T^{t-1})p(T^{t-1}|I^{0..t-1})$.
2. Estimate the foreground/background labels based on the background model or the foreground object estimates and image information: compute messages $\mu_{M^t\to F^t} = \int dM^t\phi(F^t, M^t)\mu_{M^{t-1}\to M^t}$, $\mu_{T^t\to F^t} = \int dT^t\phi(F^t, T^t)\mu_{T^{t-1}\to T^t}$, $\mu_{F^t\to M^t} = \int dF^t\mu_{S^t_{j\neq i}\to F^t}\phi(I^t, F^t)$, and $\mu_{F^t\to T^t} = \int dF^t\mu_{S^t_{j\neq i}\to F^t}\phi(I^t, F^t)$.
3. Update the background model using foreground labels predicted by the object tracker and foreground object state using labels produced by the background model ($p(M^t|I^{0..t}) \propto \mu_{M^{t-1}\to M^t}\mu_{F^t\to M^t}$ and $p(T^t|I^{0..t}) \propto \mu_{T^{t-1}\to T^t}\mu_{F^t\to T^t}$).

If inference on constituent Markov chains were performed individually, as is the case with common feed-forward systems where background subtraction results are used as input to an object tracking system, it would still involve steps analogous to 1 and 3 (and partially 2); consequently, combining models introduces very little additional complexity to the inference process.

4. Implementation and Results

Since the objective of this paper is to compare the performance of the background maintenance system with and without tracking feedback, we chose to implement a very simple adaptive module, although a more advanced system can certainly be used. The background distribution was modeled with a single (per-pixel) Gaussian with fixed variance and variable mean. Model dynamics and observation noise were also represented with Gaussian distributions with fixed variances. We used an object (blob) tracker with first-order linear dynamics similar to one described in [8].

Based on these modules, we implemented and compared the outputs and runtimes of the dual-chain algorithm described in the previous section and of the feed-forward system where a stand-alone background subtraction module provided input to an object tracker.

Both systems were evaluated on two datasets provided for the PETS 2001 workshop

(`ftp://pets.rdg.ac.uk/PETS2001/`). These datasets contain observations of an outdoor traffic scene, where foreground objects include pedestrians and vehicles. Algorithms were evaluated as follows: at every frame we have computed a raw foreground map by thresholding the background probability value at every pixel. After applying a set of morphological dilation operations connected components of a size greater than a certain threshold were extracted.

Three classes of errors were detected: (1) less than 50% of a pedestrian covered by extracted components; (2) less than 50% of a vehicle covered by extracted components; and (3) a foreground component was detected in a location where no moving objects were present. The quantitative comparison results are summarized in Figure 2. The high number of class 1 errors is due to a relatively high threshold on a connected component size. Sample frames from the first sequence with corresponding estimated background images and foreground components are shown in Figure 3.

In our experiments, the difference between running times of dual-chain algorithm and feed-forward system was less than 4%. Unoptimized code on a 2.8GHz workstation was able to achieve 9.6fps for sequential processing and 9.3fps for dual-chain processing on 768×576 images (this time includes reading images from the hard drive).

5. Conclusions

We have proposed a method for combining probabilistic background maintenance and object tracking systems that significantly improves segmentation accuracy without sacrificing performance. Our approach was motivated by the simple observation that both of these models marginalize over an intermediate feature representation between state and observation. By making the feature representation explicit in our approach, we obtained a straightforward means of mediating between the constituent models.

Exact inference on the resulting structure is complicated due to the introduction of loops in the graphical structure representing the combined models. However, as a consequence of the fact that we are interested in filtering (tracking), rather than smoothing, an approximate inference method based on sequential inference on acyclic subgraphs provides a suitable alternative to exact inference. As opposed to monolithic systems that jointly model background and foreground, our approach allows background maintenance and tracking systems to be designed independently.

Our method has been demonstrated to compare favorably to the pure feed-forward approach when applied to outdoor traffic scenes, especially when faced with foreground objects that remained stationary for extended periods of time.

References

[1] S.-C. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *Video Communications and Image Processing, SPIE Electronic Imaging*, San Jose, Jan 2004.

[2] Bohyung Han, Dorin Comaniciu, and Larry Davis. Sequential kernel density approximation through mode propagation: Applications to background modeling. In *ACCV*, 2004.

[3] M. Harville. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In *Proc ECCV*, 2002.

[4] Michael Harville, Gaile G. Gordon, and John Woodfill. Foreground segmentation using adaptive mixture models in color and depth. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 3–11, 2001.

[5] M. Isard and J.P. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *ICCV01*, pages II: 34–41, 2001.

[6] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1997.

[7] R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of dynamic background. In *Proc CVPR*, 2003.

[8] C. Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of CVPR'99*, 1999.

[9] Leonid Taycher, John W. Fisher III, and Trevor Darrell. Combining simple models to approximate complex dynamics. In *Proc. Workshop on Statistical Methods in Video Processing*, May 2004.

[10] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and practice of background maintenance. In *ICCV (1)*, pages 255–261, 1999.

[11] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[12] Q. Zhou and J. Aggarwal. Tracking and classifying moving objects from videos. In *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.

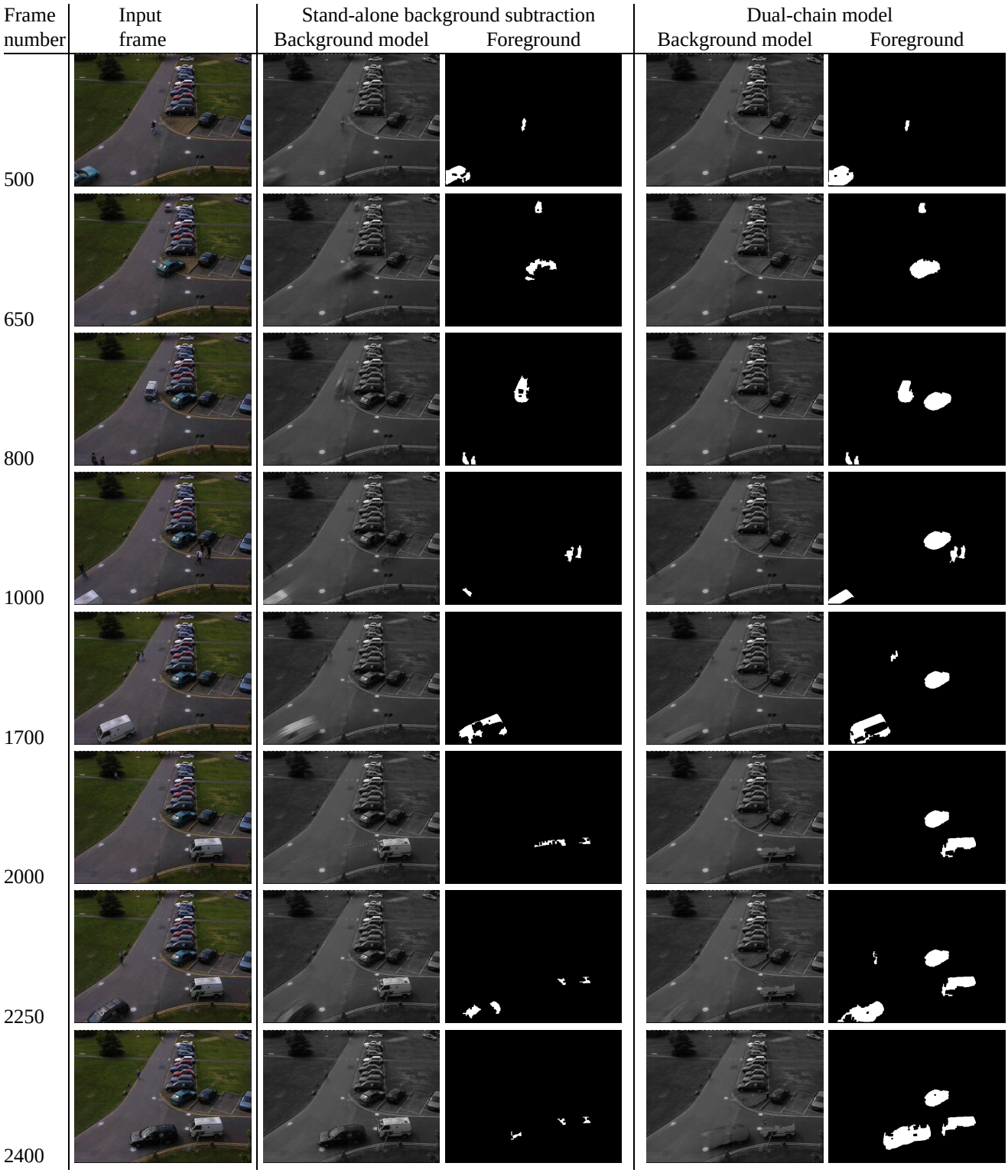

Figure 3: Qualitative comparison of background subtraction performance on one of PETS2001 image sequence. Second column holds input frames. Estimated background model and the computed foreground components are presented in the third and fourth columns for stand-alone background subtraction and in fifth and sixth columns for dual-chain model. Note that while input images are in color, all computations were performed in grayscale. See text for more details.