AUTOMATICALLY TUNING BACKGROUND SUBTRACTION PARAMETERS USING PARTICLE SWARM OPTIMIZATION

Brandyn White and Mubarak Shah

University of Central Florida
School of Electrical Engineering and Computer Science
{bwhite,shah}@cs.ucf.edu

ABSTRACT

A common trait of background subtraction algorithms is that they have learning rates, thresholds, and initial values that are hand-tuned for a scenario in order to produce the desired subtraction result; however, the need to tune these parameters makes it difficult to use state-of-the-art methods, fuse multiple methods, and choose an algorithm based on the current application as it requires the end-user to become proficient in tuning a new parameter set. The proposed solution is to automate this task by using a Particle Swarm Optimization (PSO) algorithm to maximize a fitness function compared to provided ground-truth images. The fitness function used is the F-measure, which is the harmonic mean of recall and precision. This method reduces the total pixel error of the Mixture of Gaussians background subtraction algorithm by more than 50% on the diverse Wallflower data-set.

1. INTRODUCTION

The increasing amount of surveillance cameras in our society places an increasing burden on the security professionals who have to monitor them. The field of automated surveillance alleviates this burden by using computers to detect objects in a scene, track their motion over time, identify their type, and recognize high level actions that they perform. Human activity recognition is the most sought-after of these tasks; however, it is generally based on the information gathered in the prior stages which places an upper bound on the reliability of its decisions. Of these stages, object detection is the starting point, with motion based background subtraction being the most popular detection method. In background subtraction, a statistical model of the scene's background is learned from an image sequence which is used to label pixels corresponding to foreground objects not present in the model. Diverse methods of background subtraction exist[1, 2, 3].

A common trait of background subtraction algorithms is that they have learning rates, thresholds, and initial values that must be tuned in order to produce the desired subtraction result. Despite the importance of parameter selection for each algorithm, end users often overlook or avoid this process. Certain scenarios are more difficult for the background subtraction algorithms and accordingly require more attention during parameter tuning: background occlusion (e.g., crowded scenes), stopped target objects, clutter motion (e.g., moving trees), and illumination changes. Due to this, tuning parameters for a scene can improve performance; however, the difficulty in tuning parameters and the expert knowledge that it requires makes this an expensive proposition. Since the learning parameters are often managed globally rather than at a pixel level, attention must be given to their selection to produce the lowest error overall as different regions in a scene may require different learning behaviors. The difficulty in parameter tuning is further compounded when papers dealing with algorithm descriptions omit initial working values that can be used to test an implementation. Moreover, the ideal values for an algorithm change between scenarios (e.g., indoor vs. outdoor, electro optical vs. infrared cameras, gray vs. color imagery). Different implementations of the same algorithm can have parameters on different scales making reuse of them more difficult (e.g., use of the variance as a threshold as opposed to the standard deviation in an effort to avoid the square root operation). All of these issues have real-world implications such as causing end users to avoid the state-of-the-art method as they must learn a new parameter set. Similarly, it makes using fusion between different methods and choosing algorithms based on the particular application difficult as well.

A proper solution to this problem should reduce the level of expertise required when tuning the parameters of an algorithm, require less human time and interaction, produce better parameters for a given scene, and be able to produce parameters that work well over a variety of scenes. A solution to this problem is to automatically determine the optimal parameters of algorithm given the pixel-level ground-truth images for the sequences to optimize and a fitness function to gauge the performance of the current parameters. The binary ground-truth images depict the objects of interest and may be created in any paint program using the convention of white representing foreground and black background. The fitness function measures the similarity of results obtained from the background subtraction algorithm using the candidate parameters on the training sequences to their provided ground-truth images. Now that a method of evaluation has been established, an algorithm for optimizing the fitness through modification of the parameters can be created. For such a system to be useful, it must use as few of the expensive fitness calls as possible, should reliably return similar quality results, and it should make as few assumptions about the search space as possible.

In this paper, we propose to use a Particle Swarm Optimization (PSO) algorithm for automatically determining the parameters of the Mixture of Gaussians (MoG) background subtraction algorithm[1]. We present experiments on the publicly available Wallflower dataset [3] consisting of seven diverse video sequences and demonstrate how the proposed method navigates through the parameter space to maximize the fitness function. We use the F-measure[4] as our fitness function, which is the harmonic mean of recall (the percentage of the true positives detected) and precision (the percentage of the detections that are correct), which allows both to be expressed in one scalar value with equal weight. We show that by tuning the parameters of MoG algorithm using PSO; we more than halve the total MoG pixel error compared to reference results. While the proposed

method's performance in automatically tuning background subtraction algorithm's parameters is being shown on the MoG algorithm, it should be noted that our approach is not limited whatsoever to this particular background subtraction algorithm and that its selection was motivated by its ubiquity in the automated surveillance field.

2. MOG PARAMETERS

Mixture of Gaussians is one of the most prominent forms of background subtraction due to its fast performance, ability to handle multi-modal background distributions, and high accuracy[2]. The parameters of interest are α (learning constant), K (number of Gaussians), T (prior background probability), ω_0 (initial Gaussian weight), σ_0 (initial Gaussian standard deviation) and λ (standard deviation threshold).

The learning constant, α, is arguably the most important parameter over all scene types; it governs how quickly the algorithm adapts to changes in the scene. In a simple setting, the learning rate used is fairly small as it allows adaption to illumination changes and prevents foreground objects from being learned into the background; however, in realistic outdoor scenes where illumination changes happen quickly, objects in the scene are displaced (e.g.,a table chair is moved, door is opened), or trees that are irregularly moving due to the wind require a higher learning rate, which may cause foreground objects to be incorporated into the background. Balancing between learning fast enough to model the complex background, but slow enough to not model foreground objects is a difficult task for the end user and requires careful tuning as this parameter is dependent on the scene and the frames per second of the camera used.

The number of Gaussians used to model each pixel, K, is dependent on the maximum number of modes in a pixel's background distribution. This places a lower limit on what this value should be, as any value for K less than this will cause errors in the background model, an integer value from 3 to 7 is generally used. The prior background probability T specifies the probability of a pixel value belonging to the background. If this value is set too low, scenes with multi-modal background distributions may have only some of their modes considered background; however, if set too high, it can force foreground distributions to represent the background.

The initial weight, ω_0, and variance, σ_0, parameters are often overlooked in discussion and when tuning the algorithm's parameters, but when the algorithm is used in an environment where it is important to quickly produce meaningful results, these are as important as the others. The standard deviation threshold λ is commonly fixed at 2.5; however, our experiments suggest that using this as a variable can be beneficial as-well. Our implementation of this algorithm follows directly from the reference paper [1].

3. METHOD TO QUANTIFY FITNESS

To optimize background subtraction algorithm's parameters on an image sequence, a measure of quality must be established that can quantify how similar a resulting subtraction image is to the ground-truth for a frame in one scalar value. The reason for requiring the fitness representation to return a scalar value is that it is necessary to unambiguously and automatically decide if one set of parameters is better than another. As shown in Fig. 1, maximization of precision, the percentage of the detections that are correct, would allow for reduction of false positives, but wouldn't take into account the false negatives which will likely result in loss of foreground object area, see Fig. 1(d). Similarly, maximization of recall, the percentage of the

(a) Original (b) Ideal (c) F-measure (d) Precision (e) Recall

Fig. 1. Results of maximizing the F-measure (c), precision (d), and recall (e) on the B sequence in the Wallflower data-set. This sequence has foreground objects present during the background subtraction initialization, and as such it tests how quickly a background subtraction algorithm can learn the true background and let the false background become foreground.

true positives detected, reduces the number of false negatives; however, it does not reflect how precise the resulting subtraction image is about what is designated positive, which results in excessive noise, see Fig. 1(e). A method that could maximize both of these would produce superior results than maximization of either by itself. This is done by taking the F-measure, F, of the background subtraction results and the ground-truth:

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (1)$$

The F-measure is the harmonic mean of recall and precision, it allows both to be expressed in one scalar value with equal weight. Maximization of this provides better subtraction results as the resulting subtraction images are necessarily close to the ground-truth image, see Fig. 1(c). Note that the recall relies on the existence of true positives in the ground-truth, and as such it is necessary for all ground-truth frames to have foreground objects in them.

In general, the ground-truth frames should be distributed uniformly through the sequence and contain foreground objects. The number of ground-truth frames to be used for a given sequence is proportional to the difficulty of the sequence for the background subtraction algorithm and to a lesser extent, the number of frames in it. The number of frames in the sequence determines how well the parameters produced perform over longer periods of time and as such it is recommended to have at least a few minutes of data to allow for the parameters to be optimized for more than just the initialization phase. In our experiments on challenging data-sets, one ground-truth frame for every 200 frames in a sequence proved to be sufficient. Multiple sequences can be used to reduce the effect of over-tuning to a particular video. Depending on the goal of the parameter tuning, one particular camera's video can be used to find optimal parameters for that scene, or several diverse videos can be used to create more resilient, but less specialized parameters. Results will be gathered from both parameter tuning techniques to allow for comparison between them.

There are instances where a pixel can rightly be either foreground or background; this makes the task of ground-truthing the optimal subtraction images more difficult. An example of this is a reflection under a target foreground object, in an idealized case, it should be absent from the foreground; however, many algorithms are not meant to handle this problem directly, and as such shouldn't be penalized for any decision they make. To solve this problem, a pixel may be ignored by the fitness, thus if a region can not be clearly judged as foreground or background it will be marked with a gray value between pure white and pure black in the, now grayscale, ground-truth image. Pixels that are ignored will not have any effect on the F-measure, which allows it to be more meaningful and representative of the true quality of the subtraction. When computing the 2-class confusion matrix between the subtraction image and its corresponding ground-truth, if a gray value is encountered, it is simply

disregarded. This has been used to ignore shadows for algorithms that don't handle shadow removal on a background subtraction level, imprinted surveillance information, and active television screens.

4. FITNESS SPACE NAVIGATION

It is now possible, given a fitness function, to automatically explore the space in an attempt to locate the background subtraction algorithm's parameters that maximize the fitness. Randomly searching the fitness space is a way of coarsely finding areas of high fitness by randomly selecting a parameter vector; however, this requires a large number of samples to produce useful results when many parameters are being tuned. Uniform sampling can be used to fully explore the space provided that the granularity for each parameter is known ahead of time, yet it is very time-consuming. Moreover, when the granularity of the parameters is small, the bounds of search are large, or number of dimensions is large, this method requires a prohibitively large number of samples; however, uniform sampling is useful for understanding the structure of the fitness space of an algorithm, which can provide insight that leads to more intelligent optimization. The fitness plots in Fig. 2 were created by uniform sampling between the parameters α and T of the MoG algorithm.

Given the background subtraction parameters, a closed form expression for the fitness doesn't exist; therefore, it is exceedingly difficult to calculate the gradient and step-size required by gradient descent. A reasonable compromise is to use a low number of random samples and attempt to improve their fitness by updating them over a series of time-steps directed towards areas where higher fitness has been found in an attempt to both explore the space and optimize the fitness of local areas. Therefore, in this paper we propose to use an evolutionary method, similar to the previously described process, called Particle Swarm Optimization (PSO), which is discussed next.

4.1. Particle Swarm Optimization

Particle Swarm Optimization is an algorithm that can be used to find extrema in a higher dimensional search space by utilizing swarm intelligence [5]. The swarm consists of many particles, each of which has a position in the parameter space, x_t, and a velocity, v_t, that allows them to move through the space over a series of time-steps, which are referred to as generations. In our case, the particle's position is a set of parameters for the background subtraction algorithm and its velocity represents the change in the parameters between generations. Each particle has a history of its highest fitness value and the parameter vector, p_i, that produced it over all of its previous generations, and similarly the swarm as a whole is aware of its highest fitness value that any of the particles have seen and the parameter vector, p_g, that produced it. The velocity of each particle is directed towards its best parameters by a factor referred to as its cognitive component, c_1, and the swarm's best parameters which is its social component, c_2[6]. In an effort to reduce the effect of moving in constant velocity steps and to better explore the search space, both velocity contributions are multiplied by a random factor, r. To prevent premature convergence and to maintain a reasonable velocity, a constant, K, referred to as a constriction factor[7], is used as an overall weight governing the change in velocity between generations: $K = \frac{2}{\left|2-\psi-\sqrt{\psi^2-4\psi}\right|}$, where $\psi = c_1 + c_2, \psi > 4$. Finally, the velocity and parameter vector for each particle at each generation, t, after the first is computed as:

$$v_t = K(v_{t-1} + c_1 r_1(p_i - x_{t-1}) + c_2 r_2(p_g - x_{t-1})), \quad (2)$$

$$x_t = x_{t-1} + v_t. \quad (3)$$

The cognitive and social components of the velocity can be modified to achieve a variety of behaviors that better suit the given search space; however, work has been done to come up with a suitable set of parameters and implementation decisions that are effective across many difficult optimization landscapes that can be used as a starting point for a PSO algorithm implementation [8, 9]. Unless otherwise specified, we use the canonical PSO algorithm which implies $c_1 = 2.8$ and $c_2 = 1.3$, but with 10 generations and a population size of 20.

The background subtraction algorithm imposes limits on what values its parameters may be; consequently, each dimension of the particle's parameter vector must reflect this by not allowing its value to exceed the prescribed limits. When updating the particle's parameter vector, if a dimension's corresponding velocity would cause it to exceed these limits, the velocity in that dimension would be set to zero to allow the particle to be pulled back at a natural rate. During initialization, each particle's parameter vector is randomly distributed in the specified bounds and its velocity vector is zeroed. Every generation, the background subtraction algorithm is run on the parameter vector that the particle's position represents. For each frame run that has a corresponding ground-truth frame, the F-measure is computed. To produce the particle's fitness on a sequence, the mean F-measure for all ground-truth frames in the sequence is used.

It is generally recommended that more than one sequence be used to avoid 'overfitting' the parameters to a particular sequence. The total fitness for a particle when using multiple sequences is the sum of each sequence's fitness; this gives equal weight to all sequences and ground-truth images. The result of this process over several generations is that the fitness of each particle will rise as the selected parameters continue to get better, and then gradually level off when the swarm can no longer find a better combination of parameters (Fig. 2 shows this process in action). For final parameter selection, the best performing global parameters are used; however, a separate validation step may be used for final selection on each particle's best parameters. The described process has been used to successfully tune both real and integer valued parameters; however, as the parameter vectors are real valued (due to how the velocity is computed in Eqn. 2), care must be taken when using integer valued parameters to avoid loss of possible values due to conflicts between the rounding method used and the location of the parameter's bounds.

5. EXPERIMENTAL RESULTS

The Mixture of Gaussians algorithm is used for all experiments, and Fig. 2 shows graphically how PSO navigates the fitness landscape of its two most significant parameters, α and T, with the others fixed at values previously optimized by this method (for display purposes). The Wallflower data-set is used for all experiments[3]. This data-set includes 7 canonical scenes which cover many of the difficult scenarios for background subtraction algorithms: displaced background object (MO) (omitted as no algorithm had an error on it), gradual global illumination change (TOD), instant global illumination change (LS), clutter motion (WT), low foreground to background contrast (C), foreground objects present in the sequence during initialization (B), and slow moving foreground object present in scene during initialization (FA). For comparison, the evaluation method for PSO1 is the same as in [3]: only one set of parameters for each