Adaptive Background Modeling for Paused Object Regions

Atsushi Shimad, Satoshi Yoshinaga, and Rin-ichiro Taniguchi

Kyushu University, Fukuoka, Japan

Abstract. Background modeling has been widely researched to detect moving objects from image sequences. Most approaches have a false-negative problem caused by a stopped object. When a moving object stops in an observing scene, it will be gradually trained as background since the observed pixel value is directly used for updating the background model. In this paper, we propose 1) a method to inhibit background training, and 2) a method to update an original background region occluded by stopped object. We have used probabilistic approach and predictive approach of background model to solve these problems. The great contribution of this paper is that we can keep paused objects from being trained.

1 Introduction

A technique of background modeling has been widely applied to foreground object detection from video sequences. It is one of the most important issues to construct a background model which is robust for various illumination changes. Many approaches have been proposed to construct an effective background model; pixel-level approaches[1,2,3,4], region-level approaches[5,6], combinational approaches[7,8] or so on. Almost of these approaches have a common process of updating of background model. Actually, this process is very beneficial to adapt for various illumination changes. On the other hand, we can say that the traditional background model has an ability to detect "Moving Objects" only. In other words, it causes FN (false negative) problem when a foreground object stops in the scene. This is because the paused foreground object is gradually learned as background by blind updating process. Therefore, we have to handle following problems (also see Fig. 1) in order to keep detecting the paused object.

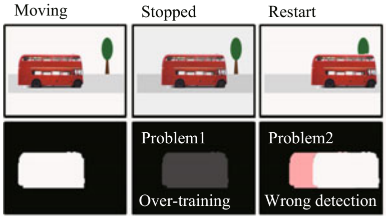

Fig. 1. Problem of blind updating of background model

1. Over-training of foreground objects
2. Wrong detection of original background regions

R. Koch et al. (Eds.): ACCV 2010 Workshops, Part I, LNCS 6468, pp. 12–22, 2011.

The first problem is caused by blind updating process of background model. Some researches tried to solve this problem by control learning rate of the background model. For example, decreasing the learning rate of some regions in which foreground objects probably stop[9] or utilizing two background model which have different learning rates[10] has been proposed. However, these approaches have not resolve the essential problem of over-training since they just extend the time for being learned as background.

The second problem is caused by a paused foreground object when it starts to move again. In such a case, an original background region hidden by the object might be detected wrongly since the paused foreground object has been included in the background model. Another possibility is that the FP problem will be caused when some illumination change occur while the foreground object stops. The hidden region will be detected wrongly since the background model does not know the illumination change occurred in the hidden region. A study which considers the illumination changes until a foreground object is regarded as paused object has been proposed[11], but it does not handle the illumination change (background change) in the region hidden by the paused foreground object.

In this paper, we propose a novel approach which use two different kinds of models; one is a pixel-level background model and the other is a predictive model. Two problems mentioned above can be resolved by utilizing these two models efficiently. The characteristics of our study are summarized as follows.

1. Our approach can control over-training of paused foreground objects without adjusting the learning rate.
2. Our approach can update the original background region hidden by paused objects.

In addition, our background model is robust against illumination changes by using two kinds of models in combination.

2 Framework

The processing flow of our proposed background model is shown in Fig. 2. At the first stage, background likelihoods of an observed image are calculated based on the probabilistic model(see section 3.1) and the predictive model(see 3.2). At the second stage, the foreground region is determined by integrating two background likelihoods evaluated by the pixel-level background model and the predictive model(see section 4). Finally, at the third stage, the parameters of both models are updated. Generally, the observed pixel

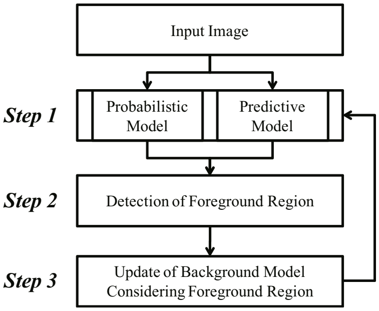
Input Image
Step 1
Probabilistic Model
Predictive Model
Step 2
Detection of Foreground Region
Step 3
Update of Background Model Considering Foreground Region

Fig. 2. Processing flow

value is directly used for updating the parameters. In our approach, meanwhile, when a pixel is judged as "foreground" at the second stage, we use alternative pixel value around the pixel which has similar background model. This process avoid the foreground object being trained as "background". We will give a detailed explanation in section 5.

3 Probabilistic Model and Predictive Model

3.1 Probabilistic Model Base on GMM

We have modified the GMM-based background model[2]. The modified background model consists of 2 steps; evaluation of background likelihood and update of model parameters .

Evaluation of Background Likelihood. Let x_i^t be a pixel value on a pixel i at frame t. For simple expression, we omit the notation i when we explain each pixel process. The background likelihood is represented as

$$P(x^t) = \sum_{k=1}^{K} \frac{w_k^t}{(2\pi)^{\frac{n}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^t - \mu^t)^T \mathbf{\Sigma}^{-1}(x^t - \mu^t)\right) \tag{1}$$

The original approach[2] judges whether or not an observed pixel value belongs to "background". Our approach does not output such a judgment result explicitly. Instead, we calculate the background likelihood at this processing stage.

Update of Model Parameters. The model parameters are updated in the same way as the original method[2].

The weights of the K distributions at frame t, w_k^t, are adjusted as follows

$$w_k^t = (1-\alpha)w_k^{t-1} + \alpha M_k^t \tag{2}$$

where α is the learning rate and M_k^t is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized.

Every new pixel value x^t is examined against the existing K Gaussian distributions, until a match is found. A match is defined as a pixel value within 2.5 standard deviations of distribution. The parameters of unmatched distributions remain the same. When a match is found for the new pixel value, the parameters of the distribution are updated as follows.

$$\mu^t = (1-\rho)\mu^{t-1} + \rho y^t, \qquad \sigma^t = (1-\rho)\sigma^{t-1} + \rho(y^t - \mu^t)^T(y^t - \mu^t) \tag{3}$$

where the ρ is the second learning rate, y^t is a pixel value which is used for update of model parameters. We purposely distinguish the notation y^t from x^t since the pixel value y^t depends on the judgment result explained in following section 5.

If none of the K distribution matches the current pixel value, a new Gaussian distribution is made as follows.

$$w_{k+1}^t = W, \;\; \mu_{k+1}^t = y^t, \;\; \sigma_{k+1}^t = \sigma_k^t \tag{4}$$

where W is the initial weight value for the new Gaussian. If W is higher, the distribution is chosen as the background model for a long time. After this process, the weights are renormalized. Finally, when the weight of the least probable distribution is smaller than a threshold, the distribution is deleted, and the remaining weights are renormalized.

3.2 Predictive Model Based on Exponential Smoothing

Exponential Smoothing. We use an exponential smoothing method[12] to acquire a predictive pixel value z^t. Exponential smoothing is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make forecasts. The simplest form of exponential smoothing is given by the following formula.

$$m^t = \beta x^t + (1-\beta)m^{t-1} \tag{5}$$

where m^t is the estimate of the value, x^t is the observed value at frame t. β is the smoothing constant in the range $\beta(0 \leq \beta \leq 1)$. The forecast function, which gives an estimate of the series can be written as follows:

$$z^t = m^t + \frac{1-\beta}{\beta} r^{t-1}, \qquad r^t = \beta(z^t - z^{t-1}) + (1-\beta)r^{t-1} \tag{6}$$

where r^t is the current slope and z^t is the estimate of the value with a trend.

Evaluation of Background Likelihood. The predictive model mentioned above is used for two purposes. One is for searching a pixel which has a similar tendency with the pixel hidden by a foreground object, which will be explained in section 5. The other is for region-level background model explained in this section. Some literatures have reported that spatial locality information is effective for illumination changes[6,13]. This idea derives from a hypothesis that similar changes will be observed around the pixels when illumination change occurs. In the proposed method, we use not only the predictive value of target pixel but also the values of neighbor pixels simultaneously in order to evaluate background likelihood.

Let R be a set of neighbor pixels around pixel i, the background likelihood $Q(x^t)$ is calculated by following formula.

$$Q(x_p^t) = \frac{\sum_{i \in R} \phi(x_i^t, z_i^t)}{|R|}, \qquad \phi(x^t, z^t) = \begin{cases} 1 & \text{if } |x^t - z^t| < th \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The $\phi(x^t, z^t)$ is a range which allows predictive error.

Update of Model Parameters. The parameters of predictive model are updated by an observed pixel value. In the same way with the probabilistic background model, we decide whether or not to use the observed value directly. The detailed explanation will be given in section 5.

4 Foreground Detection Based on MRF

The background model and foreground model output the evaluation result of background and foreground likelihood. The final decision whether or not each pixel is foreground is determined by integrating each evaluation result. We define an energy function based on Markov Random Field (MRF) and give each pixel proper label (foreground or background) by minimizing the energy function. Our energy function is defined as

$$E(L) = \lambda \sum_{i \in \mathcal{V}} G(l_i) + \sum_{(i,j) \in \mathcal{E}} H(l_i, l_j) \tag{8}$$

where $L = (l_1, \ldots, l_N)$ is the array of labels, and N is the number of pixels. The $\mathcal{V}$ and $\mathcal{E}$ represent a set of all pixels and a set of all nearest neighboring pixel pairs respectively. The $G(l_i)$ and $H(l_i, l_j)$ represent the penalty term and smoothing term respectively and they are calculated as follows.

$$G(l_i) = \frac{P(x_i) + Q(x_i)}{2}, \qquad H(l_i, l_j) = \frac{1}{\ln(\|x_i - x_j\| + 1 + \epsilon)} \tag{9}$$

We assign proper labels to pixels which minimize the total energy $E(L)$, and it is solved by a graph cut algorithm[14]. We make a graph which has two terminal nodes (Source (s) and Sink (t)) and some nodes corresponding to pixels. Edges are made between nodes. We give each edge a cost $u(i, j)$ defined as follows.

$$u(i,j) = H(l_i, l_j), \qquad u(s,i) = \lambda(1 - G(l_i)), \qquad u(i,t) = \lambda G(l_i) \tag{10}$$

5 Update of Model Parameters

If we directly use observed pixel values for model update process, not only background regions but also foreground regions are gradually trained by the model. It will cause FN (false negative) problem when an moving object stops in the scene (e.g. bus stop, intersection and so on). One of the solutions is to exclude foreground pixels from update process. However, such ad-hoc process will generate another problem that background model on the foreground pixel cannot adapt itself for illumination changes while the foreground object stops. As the result, when the paused object starts to move again, the occluded region will be detected wrongly (FP (false positive) problem). To solve this problem, our approach updates model parameters on the foreground pixels with the help of neighbor background pixels.

The specific update process of our proposed approach is as follows. Let F and B be a set of foreground pixels and background pixels judged in section 4 respectively, the pixel value y_i^t for model update is calculated as

$$y_i^t = \begin{cases} x_i^t & \text{if } i \in B \\ x_c^t & \text{if } i \in F \end{cases}, \qquad c = \underset{j \in B}{\operatorname{argmin}} f(\Theta_i, \Theta_j). \tag{11}$$

The Θ is a set of parameters of probabilistic model and predictive model on each pixel's. In our experiments, we set the Θ to be $\Theta^t = \{\mu_1^t, m^t, r^t\}$, which denotes the average background pixel value of the distribution which has the largest weight μ_1^t, exponential smoothing m^t and the slope of the observed value r^t. The most important contribution in this paper is to use x_c^t for model update. When a pixel is judged as foreground, our approach searches the model which has the most similar model parameters with the pixel. The similarity between model parameters is evaluated by the distance function $f(\Theta_i, \Theta_j)$, where we use the L1 norm in our experiments.

In this way, our approach does not use foreground pixel values to update model parameters. Alternatively, we use the pixel value on the background pixel whose model parameters are the most similar with the one on the foreground pixels. This procedure avoid the foreground object from being trained as background. Therefore, even if a foreground object stops in the scene, our approach keeps detecting the foreground object. In addition, the implicit update process of the background models hidden by the foreground object reduces FP problem when the paused object start to move again.

6 Experimental Results

We have used several public datasets to investigate the effectiveness of our proposed method. The computational speed of the proposed method was $7fps$ for QVGA image size by using a PC with a Core i7 3.07GHz CPU.

According to our preliminary experimental results, we have decided some parameters as follows; $\alpha = 0.5$, $\beta = 0.5$, $th = 15$. These parameters were common to following experiments.

6.1 Evaluation of Implicit Model Update

The dataset used in this section is released at PETS2001[1] including illumination changes in the outdoor scene. We have clipped two subscenes from the original image sequence; one is a scene in which illumination condition changes from dark to bright, and the other is a scene from bright to dark. The both scenes consist of about 600 frame images. Moreover, we have selected two 10×30 pixel areas; an area with simple background and an area with complex background. We have conducted a simulation experiment under the condition that the foreground object stopped on the 10×30 pixel region and evaluated how effective the proposed implicit update process mentioned in section 5 was.

Table 1 shows the error value and the number of FP pixels around illumination changes. The error value means the difference value between the estimate value of background model and the observed pixel value. Meanwhile, we counted up the number of pixels whose error value exceeded a threshold as FP pixels. This situation was under the assumption that paused object started to move again.

[1] Benchmark data of International Workshop on Performance Evaluation of Tracking and Surveillance. ftp://pets.rdg.ac.uk/PETS2001/

Table 1. Comparison of Model Update Methods: “B to D” denotes Bright to Dark, “D to B” denotes Dark to Bright

		B to D	B to D	D to B	D to B
		Simple BG	Complex BG	Simple BG	Complex BG
Without Update	Error	102.8	60.9	105.5	63.2
	FP	250	99	250	106
Traditional Update	Error	9.2	8.4	12.0	10.0
	FP	0	0	0	0
Proposed Method	Error	14.0	23.8	14.8	29.6
	FP	7	6	0	13

We have compared out proposed method with two methods; without model update (Table 1:without update) and with model update by traditional method (Table 1:traditional update). Note that traditional method used the observed pixel value directly for model update.

When we didn’t update model parameters, the error value was large and a lot of FP pixels were detected wrongly. The traditional update method could adapt for the illumination changes. As the result, the error value and the number of FP pixels were very small. On the other hand, our proposed method could also adapt the illumination changes even though the investigated area was occluded by the pseudo foreground object. The error value in the complex background became larger than those in the simple background. However, this didn’t lead to a sensible increase of the number of FP pixels. These discussions applied to both scenes; scene from dark to bright and scene from bright to dark. Therefore, we could conclude that the implicit update process of the background model was effective to update the region occluded by paused foreground object.

6.2 Accuracy of Paused Object Detection

We have user three outdoor scenes[2] to investigate the detection accuracy of paused foreground object regions. The Scene 1, Scene 2 and Scene 3 in Fig. 3 shows the snapshot of about 100^{th} frame, 60^{th} frame and 150^{th} frame after the moving object stopped. The illumination condition in Scene 1 is relatively stable compared with the other scenes. We have compared our proposed method with two representative methods; GMM based method[2] and fusion model of spatial-temporal features[7]. The parameters in these competitive methods were set to be the same as original papers. We have evaluated the accuracy by the precision ratio, recall ratio and F-measure given by following formulas.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \text{F} = 2/\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right) \quad (12)$$

The F-measure indicates the balance precision and recall. The larger value means better result. The TP, FP and FN denote the number of pixels detected correctly, detected wrongly, undetected wrongly respectively.

[2] We got ground truth dataset from http://limu.ait.kyushu-u.ac.jp/dataset/

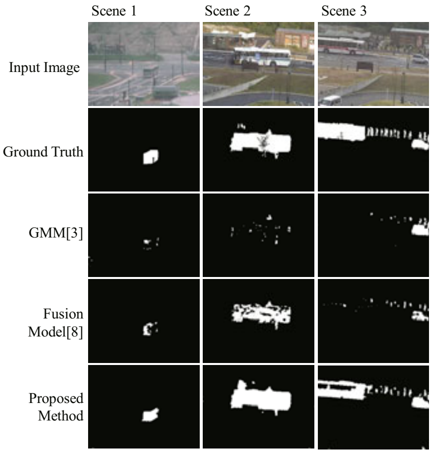

Fig. 3. Result of object detection after the moving object stopped. Scene 1: 100th frame after stopped, Scene 2: 60th frame after stopped, Scene 3: 150th frame after stopped

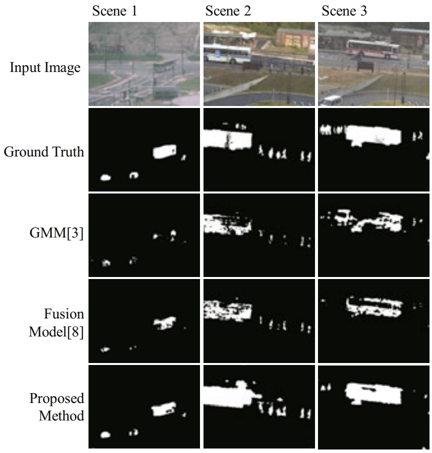

Fig. 4. Result of object detection after the object restarted to move

Fig. 3 shows the evaluated images, and Table 2 shows the evaluation results. The GMM based method[2] could detect just a few foreground pixels since it had learned the paused foreground object as "background". The fusion model[7] also gradually learned the foreground objects as "background". This is why the recall ratios of these methods were very low in all scenes. On the other hand, our proposed method gave much better recall ratio than competitive methods. The F-measure was also superior to the others.

Secondly, we have evaluated the precision ratio, recall ratio and F-measure with another scene in which the paused object had started to move again. The proposed method gave us better result than the other methods (See Table 3). The GMM based method[2] detected many FP pixels in the region where the foreground object had been paused(See Fig. 4). This is because illumination

Table 2. Accuracy evaluation of object detection after the moving object stopped

		Scene 1	Scene 2	Scene 3
GMM[2]	Precision	0.87	0.95	0.86
	Recall	0.13	0.05	0.16
	F-measure	0.23	0.10	0.27
Fusion Model[7]	Precision	0.98	0.95	0.94
	Recall	0.37	0.69	0.13
	F-measure	0.53	0.80	0.24
Proposed Method	Precision	0.90	0.85	0.87
	Recall	0.76	0.99	0.74
	F-measure	0.82	0.92	0.81

Table 3. Accuracy evaluation of object detection after the object restarted to move

		Scene 1	Scene 2	Scene 3
GMM[2]	Precision	0.95	0.93	0.80
	Recall	0.22	0.52	0.46
	F-measure	0.35	0.66	0.58
Fusion Model[7]	Precision	0.98	0.93	0.94
	Recall	0.48	0.61	0.46
	F-measure	0.65	0.73	0.61
Proposed Method	Precision	0.92	0.78	0.91
	Recall	0.78	0.98	0.82
	F-measure	0.85	0.87	0.86

change occurred during the period. Meanwhile, the fusion model[7] and the proposed method didn't detect the occluded region wrongly. However, the fusion model could not detect inside of the moving object because of over-training of foreground object. This is why the recall ratio of the fusion model was lower than the proposed method.

6.3 Evaluation of Robustness against Illumination Changes

We have used a outdoor image sequence in which illumination condition had sometimes changed rapidly, which was also used in the section 6.1. We have selected three images from 5,000 frames for evaluation. The parameters of background models including competitive methods were set to be the same as previous experiments.

Table 4. Accuracy evaluation with PETS2001 dataset

		# 831	# 1461	# 4251
GMM[2]	FN	0	211	234
	FP	1111	133	665
	F-measure	–	0.76	0.22
Fusion Model[7]	FN	0	450	311
	FP	0	41	1
	F-measure	–	0.57	0.24
Proposed Method	FN	0	82	120
	FP	0	478	422
	F-measure	–	0.71	0.47

The recall ratio, precision ratio and F-measure are shown in Table 4. In the case of FP or FN to be zero, we showed the F-measure "–" in Table 4 since it cannot be calculated. The illumination condition of scene # 831 was changed around the time. The GMM based method[2] detected many FP pixels since it was hard for GMM to adapt for rapid illumination changes. Meanwhile, our proposed method didn't detect any FP pixels as good as the fusion model[7], which was reported that it is very robust against various illumination changes. The scene # 1461 included foreground objects under the stable illumination condition. The fusion model[7] detected the foreground object in the smaller size than the ground truth. This is because the fusion process was achieved by calculating logical AND operation between two kinds of background models. Therefore, the FN became large and the FP became smaller compared with the GMM based method. On the other hand, the proposed method detected the foreground objects including their shadow region. Note that shadow regions were not target to detect in the ground truth. This is why the FP became larger in the proposed method. To solve this problem, we are going to introduce a shadow detection method such as [15] in the future work. Finally, the scene # 4251 included foreground objects with illumination changes. This scene is one of the most difficult scenes for object detection. The proposed method gave better result than other two competitive methods. Note that the illumination change was not a factor of FP pixels. It was caused by shadow regions. Through above discussion, we are sure that our proposed method is very robust for illumination changes.

7 Conclusion

We have proposed a novel background modeling method. The proposed method could update a background region even when the region was occluded by a foreground object. This process was very effective for not only implicit background update but also keeping foreground object to being detected when the foreground object stopped in the scene. Through several experiments, we have confirmed the effectiveness of our approach from the viewpoints of robustness against illumination changes, handling of foreground objects and update of background model parameters. In our future works, we will study about efficiency strategy of initializing background model, complement of undetected pixels such as inside of the objects.

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. Computer Vision and Pattern Recognition 2, 246–252 (1999)
2. Shimada, A., Arita, D., Taniguchi, R.: Dynamic Control of Adaptive Mixture-of-Gaussians Background Model. In: CD-ROM Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance 2006 (2006)
3. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and Foreground Modeling Using Non-parametric Kernel Density Estimation for Visual Surveillance. Proceedings of the IEEE 90, 1151–1163 (2002)
4. Tanaka, T., Shimada, A., Arita, D., Taniguchi, R.: A Fast Algorithm for Adaptive Background Model Construction Using Parzen Density Estimation. In: CD-ROM Proc. of IEEE International Conference on Advanced Video and Signal based Surveillance (2007)
5. Shimada, A., Taniguchi, R.: Hybrid Background Model using Spatial-Temporal LBP. In: IEEE International Conference on Advanced Video and Signal based Surveillance 2009 (2009)
6. Satoh, Y., Shun'ichi Kaneko, N.Y., Yamamoto, K.: Robust object detection using a Radial Reach Filter(RRF). Systems and Computers in Japan 35, 63–73 (2004)
7. Tanaka, T., Shimada, A., Taniguchi, R., Yamashita, T., Arita, D.: Towards robust object detection: integrated background modeling based on spatio-temporal features. In: Asian Conference on Computer Vision 2009 (2009)
8. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principle and Practice of Background Maintenance. In: International Conference on Computer Vision, pp. 255–261 (1999)
9. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. Computer Vision and Pattern Recognition, 1–8 (2008)
10. Porikli, F., Ivanov, Y., Haga., T.: Robust abandoned object detection using dual foreground. EURASIP Journal on Advances in Signal Processing (2008)
11. li Tian, Y., Feris, R., Hampapur, A.: Real-time detection of abandoned and removed objects in complex environments. In: International Workshop on Visual Surveillance - VS 2008 (2008)
12. Holt Charles, C.: Forecasting seasonals and trends by exponentially weighted moving averages. International Journal of Forecasting 20, 5–10 (2004)

13. Heikkilä, M., Pietikäinen, M., Heikkilä, J.: A texture based method for detecting moving objects. In: British Machine Vision Conf., vol. 1, pp. 187–196 (2004)
14. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1124–1137 (2004)
15. Martel-Brisson, N., Zaccarin, A.: Moving cast shadow detection from a gaussian mixture shadow model. Computer Vision and Pattern Recognition, 643–648 (2005)