Abstract

This paper presents a novel real-time multi-feature multi-scale codebook-based background subtraction algorithm, targeted for challenging surveillance environments. Our contribution is three-fold. First, we present an extension of the Codebook background model that combines multiple features, such as intensity, colour and texture, in a principled way, simultaneously taking into account both the feature’s confidence and its similarity score. Second, a new local texture pattern descriptor is proposed, entitled Local Ratio Pattern, generalizing previously successful local pattern methods. Third, a generic multi-scale confidence fusion scheme is provided, in order to aggregate individual results at different scales. A thorough evaluation is performed on the challenging I2R dataset. In addition, a comparison is carried out with other competing methods, leading to state-of-the-art performance.

1. Introduction

A crucial first step in many vision systems is to distinguish the relevant foreground objects in a scene from the background. Accurate background subtraction, also referred to as foreground detection, allows higher-level tasks, such as object tracking, categorization or action recognition, to focus on the objects of interest. This is a challenging problem, since backgrounds can include large image variations due to lighting and repetitive motions. While the human visual system can filter out such complex backgrounds, this is not necessarily the case with computer vision systems. Accordingly, this problem has received a large amount of attention from the vision research community. For a recent review on background subtraction approaches, see [2].

As suggested in [20], the obstacles to robust background subtraction can be divided into several specific challenges. There is the problem of camouflage: objects can be very close in appearance to the background, especially if the system uses only a single feature, such as intensity or colour. Background regions may be occluded during training, requiring a bootstrapping approach to learning. Global and non-uniform illumination changes can occur over different time scales: everything from a light switch to a passing cloud to the transition between day and night. Background motion also occurs at different time scales, from trees waving in the wind to occasionally moved objects that must be reincorporated into the background. Of particular interest is the foreground aperture problem: identifying the right scale at which to interpret an object’s motion.

Section 2 briefly discusses how vision researchers have addressed these problems, with a focus on the spatial scale of analysis (i.e., pixel, region or frame). In Section 3, we describe our approach, which includes a background model formulation that can deal with multiple features, such as colour, intensity and texture, a novel local texture feature and a method for integrating information across spatial scales. Section 4 compares our results on a common suite of videos with a variety of other approaches. Conclusions are drawn in Section 5.

2. Related Work

Foreground detection methods can be categorized based on the spatial area of influence when modeling the background process. They can be classified as pixel based, region based and frame based approaches. Pixel-level methods only use information available at the same pixel location over time. Wren et al. modeled each background pixel with a single Gaussian [21]. Stauffer and Grimson proposed a mixture of Gaussians representation with an online update scheme, in order to improve the performance of a single Gaussian, where the background contains multiple modes, such as in the cases of waving trees or flickering lights. The mixture of Gaussians representation was later improved by Zivkovic [22] and Lee [7], by sup-
porting an adaptive number of modes for each pixel and by providing better convergence properties for the online update. Rather than estimating model parameters, other approaches use non-parametric density estimation, such as the Parzen window method: Elgammal et al. [1] used a Gaussian kernel, whereas Tanaka et al. [18] adopted a uniform kernel, for efficiency constraints. Kim et al. [4] proposed a compromise between parametric and non-parametric approaches, where each pixel’s background was represented by an adaptive set of codewords. A number of methods adopted prediction filters: a Kalman filter based method was proposed by Ridder et al. [16], whereas Toyama et al. [20] used a Wiener filter. Li et al. [8] maintained a histogram of colours and colour co-occurrence features.

Region-level methods integrate information available in the pixel neighbourhood, in space and/or time: Mittal and Paragios [12] integrated motion-based information obtained from optical-flow; Matsuyama et al. [11] maintained edge-based histograms; Ko et al. [5] preserved colour distributions; Mason and Duric [10] kept normalized vector distances; Reddy et al. [15] used the first 4 DCT coefficients of each colour channel to represent the low spatial frequencies of overlapping pixel blocks. A number of methods maintain texture pattern descriptors, such as Local Binary Pattern (LBP), proposed by Heikkilä et al. [3], the Shift Invariant Local Trinary Pattern (SILTP) extension recently proposed by Liao et al. [9] and the dynamic texture representation proposed by Monnet et al. [13].


Mixed methods combine information at the pixel, region and frame levels. Examples include the Wallflower algorithm proposed by Toyama et al. [20] and the system by Tanaka et al. [19].

The solution presented here integrates and generalizes a number of successful concepts: it makes use of information at different scales by integrating pixel-level, region-level and frame-level information in a multi-scale processing approach. We propose a background subtraction model based on the codebook framework [4], extended to represent each codeword with multiple features (i.e. intensity, colour and texture), taking into account both the feature similarity and its confidence. Intensity and colour features capture pixel-level information, whereas the newly proposed Local Ratio Pattern texture descriptor, a generalization of previous approaches [3, 9], describes region-level information. At frame-level, global illumination changes are detected, triggering model re-learning. In addition, we propose a novel coarse-to-fine multi-scale processing technique that fuses together information from all scales, taking confidence into account.

3. Method

The core contribution of our background model is the integration of intensity, colour and texture information across multiple scales. We propose a multi-feature Codebook model, as described in Section 3.1, to represent the multi-modal background at each scale. Section 3.2 describes our intensity feature. Section 3.3 introduces the color feature. Section 3.4 talks about the Local Ratio Pattern (LRP) feature that we propose to encode local edge information. We use a rapid relearning approach to adapt to global illumination changes, described in Section 3.5. Finally, Section 3.6 explains our process for integrating information across scales.

3.1. Multi-Feature Codebook Background Model

The Codebook background model, introduced by Kim et al. [4], represents the possible appearances of each pixel as a set of codewords. Each codeword is considered a match for a range of possible observed features. The model keeps track of the number of times a codeword has been observed, its first and last appearance and the length of the longest gap between appearances. Like Mixture of Gaussians, a Codebook background can represent multi-modal feature distributions, but it can be quicker to evaluate, since the observed features often need only be compared with the assigned codeword from the previous frame. At a given scale, our system is similar in spirit to that presented in [4], but extended to integrate multiple appearance features using a confidence-based approach. In addition, we use a unified, fixed-size bank of codewords per pixel.

We represent each pixel as a set of up to $n_c$ codewords, with foreground and background codewords represented in the same set. In normal operation, if foreground or background codewords have not been observed for $t_f$ or $t_b$ seconds respectively, they are deleted. This grace period grows with the number of observations of the codeword, up to a doubling. If foreground codewords have been observed for more than $t_{upgrade}$ seconds, they are upgraded to background. Because of the limited number of codewords per pixel, there might not be a codeword slot available to represent a new set of features that does not fit any of the existing codewords. In this case, the codeword that is closest to being deleted is removed early (usually this is a foreground codeword with a large observation gap).
Prior to normal operation, there is a learning stage of $t_{learning}$ seconds. During this period codewords are neither upgraded to background nor removed, unless the limit $n_c$ is reached. At the end of training, for each pixel, the system finds the codeword with the smallest observation gap (including time before the codeword was first observed) $t_{min}$. Codewords with gaps less than $t_{min} + 0.2t_{learning}$ are considered background elements, while the others are deleted.

Each codeword maintains a set of central features $c = \{c_1, c_{\text{RGB}}, c_{\text{LRP}}\}$ (representing intensity, colour and texture features, respectively) that can be compared with the pixel’s observed features $f = \{f_I, f_{\text{RGB}}, f_{\text{LRP}}\}$. Each component feature $k \in K$, where $K = \{I, \text{RGB}, \text{LRP}\}$, has a similarity function $\text{sim}_k(c, f) \in [0, 1]$ that measures how well the features match, a confidence function $\text{conf}_k(c, f) \in [0, 1]$ that indicates the expected accuracy of the similarity and a sensitivity parameter, $\lambda_k$. The overall similarity $\text{sim}(c, f)$ between a codeword and the observed feature set is the harmonic mean of the individual similarities, weighted by the associated confidence and sensitivity:

$$\text{sim}(c, f) = \frac{\sum_k^K \lambda_k \text{conf}_k(c, f) + 0.05}{\sum_k^K \lambda_k \text{conf}_k(c, f) / \text{sim}_k(c, f) + 0.1}. \quad (1)$$

The harmonic mean allows an extreme mismatch (low similarity) in one of the feature components to dominate the overall similarity. The small constant factors drive the similarity towards 0.5 if none of the feature components are confident. Each frame’s codewords are compared with current features in last-observed order to find the codeword with the highest similarity. Comparison finishes early if a background codeword has a similarity above 0.95. If no codeword has a similarity above 0.5, a new codeword is introduced. If a matching codeword exists, it undergoes a weighted update toward the observed features: $c = \alpha f + (1 - \alpha)c$. The overall foreground score $\gamma$ for a pixel is one minus the maximum background similarity.

### 3.2. Intensity Features

Image intensity alone is a problematic feature for background modeling, since it is very sensitive to global and local illumination changes. However, in surveillance scenarios, many scenes are dominated by relatively flat and colourless regions where raw intensity is the only available local discriminator. We therefore include intensity as an important local feature and deal with global illumination changes separately. The similarity of the intensity scores is calculated as:

$$\text{sim}_{\text{RGB}}(c, f) = \frac{\delta_{\text{RGB}}^2}{\delta_{\text{RGB}}^2 + \lambda_{\text{RGB}} d_{\text{RGB}}(c, f)^2}, \quad (5)$$

where $\lambda_{\text{RGB}}$ is the sensitivity to colour change and $\delta_{\text{RGB}} = 15$ is a constant scaling factor.

Since colours with low saturation are very common in surveillance videos, colour match confidence is lower the closer the two colours are to gray. The grayness, $\phi \in [0, 1]$ of a colour decreases with distance from the black-white axis. Color match confidence is calculated as:

$$\text{conf}_I(c, f) = 1.0. \quad (3)$$

### 3.3. Colour Features

We adopt a colour similarity measure that is reasonably robust to additive image noise and to changes in intensity. As shown in Figure 1, similarity is calculated based on the distance between $f_{\text{RGB}}$ and the normalized vector of $c_{\text{RGB}}$, $\vec{\rho} = c_{\text{RGB}}/\|c_{\text{RGB}}\|_2$, as in:

$$d_{\text{RGB}}(c, f) = \|\vec{\rho} \cdot f_{\text{RGB}} - f_{\text{RGB}}\|_2. \quad (4)$$

This distance reflects the shift in colour space required to align $f_{\text{RGB}}$ with $c_{\text{RGB}}$, modulo intensity changes. It reports a small distance in dark regions where the angle between the vectors can be large. The similarity for colour features is calculated in a manner similar to intensity:

Figure 1. Measuring the perpendicular distance of $f_{\text{RGB}}$ from $c_{\text{RGB}}$ allows for variation in intensity.

$$\text{sim}_{\text{RGB}}(c, f) = \frac{\delta_{\text{RGB}}^2}{\delta_{\text{RGB}}^2 + \lambda_{\text{RGB}} d_{\text{RGB}}(c, f)^2}, \quad (5)$$

where $\lambda_{\text{RGB}}$ is the sensitivity to colour change and $\delta_{\text{RGB}} = 15$ is a constant scaling factor.
3.4. Local Ratio Pattern Features

In a surveillance application, the positions and the orientations of local intensity gradients are usually critical components of the scene. By extracting features based on the ratios of intensities of a central pixel versus nearby pixels, a system can capture much of this information in a largely lighting-invariant form (though nonlinear scaling, shadows and uneven lighting complicate the situation). Features that describe these ratios in very coarse forms, such as Local Binary Patterns (LBP, [3]) or the Local Trinary Pattern (LTP, [9]) have been employed successfully for background modeling. However, the very coarse binning of local intensity ratios can result in substantial feature noise when the true ratios fall close to a bin boundary.

We propose the Local Ratio Pattern (LRP), similar in spirit to the LBP and the LTP, but extended to 4 bits, in order to divide the range of possible intensity ratios into 16 bins, instead of 2 or 3. This finer-grained encoding allows the feature comparator to respond more proportionately to a change in the underlying image. For a central pixel at location \((x_0, y_0)\), \(LRP_{J,R}(x_0, y_0)\) concatenates 4-bit ratios with \(J\) neighboring pixels equally spaced on a circle of radius \(R\):

\[
LRP_{J,R}(x_0, y_0) = \bigoplus_{j=0}^{J-1} r(I_0, I_j),
\]

where \(I_0\) is the intensity of the central pixel, \(I_j\) the intensity of its neighbor and \(r(I_0, I_j)\) is the binned ratio value. We use \(J = 4\) and \(R = 1\) for computational simplicity. The ratio value is calculated as:

\[
r(I_0, I_j) = \left[ \frac{I_j - \tau_l}{\tau_s} \right],
\]

where \(\tau_l = 0.3\) is the upper bound of the first bin and \(\tau_s = 0.2\) is the bin spacing. \(r(I_0, I_j)\) is clipped to the range \([0, 15]\), meaning that the lower ratio bound of the upper bin is 3.1. Note that this spacing of the bins places 1.0 in the center of a bin, so as to minimize noise due to small variations around this common intensity ratio.

Let \(d^2_{LRP}(c, f)\) represent the sum-square-difference in bin values between LRP’s \(c_{LRP}\) and \(f_{LRP}\), belonging to the codewords \(c\) and \(f\). However, \(d^2_{LRP}\) does not always correspond well with perceptible differences in the scene. As in Figure 2, in dark regions imperceptible image noise may cause large swings in intensity ratios while relatively small changes in ratio in bright regions are easily observed. Given two identical image patches of intensities \(c_l\) and \(f_l\), subjected to additive Gaussian image noise with variance \(\sigma^2\), we estimate that the squared difference between the resulting codes will have an average value of:

\[
d^2_{LRP}(c, f) = J \cdot \left( \frac{\sigma}{c_l \tau_s} \right)^2 + \left( \frac{\sigma}{f_l \tau_s} \right)^2.
\]

Therefore, observing a given \(d^2_{LRP}\), we estimate the implied noise level as:

\[
y^2_{LRP}(c, f) = \left( \frac{\tau_s}{J} \right) \left( \frac{c_l^2 f_l^2}{c_l^2 + f_l^2} \right) d^2_{LRP}(c, f).
\]

Finally, the similarity of two LRPs is given by:

\[
sim_{LRP}(c, f) = \frac{\delta_{LRP}^2}{\delta_{LRP}^2 + \lambda_{LRP} y^2_{LRP}(c, f)},
\]

where \(\lambda_{LRP}\) is the sensitivity to intensity change and \(\delta_{LRP} = 20\) is a constant scaling factor.

Even though the LRP is an effective representation for textured areas, many images are dominated by flat or nearly-flat regions. A match between two flat regions is less reliable than a match between two edged regions, because flat areas are so much more common. The flatness, \(\psi\), of an LRP is its similarity score to a perfectly flat LRP. The confidence of a match between two features decreases with their flatness:

\[
\text{conf}_{LRP}(c, f) = 1 - \psi(c) \psi(f),
\]

where \(\psi(c)\) and \(\psi(f)\) are the flatness of the codeword and of observed feature, respectively.

3.5. Adapting to Global Illumination Changes

Large and fast changes in global illumination can generate significant areas of false foreground, even
when using only features that are designed to be robust to illumination change. The scene changes from a thrown light switch, for example, are dramatic, but typically not uniform. Our system tracks the mean global intensity and, if it changes too rapidly over a short time period, it triggers a brief period of background relearning. During this relearning stage, foreground output is suppressed, and learning of new background codewords is accelerated. Specifically, we reduce $t_{\text{upgrade}}$, the time required to learn a new background codeword and $t_{\text{fg}}$, the time required to forget a foreground codeword to be proportional to the minimum relearning period (usually a few seconds). Once global illumination has approximately stabilized, the relearning period ends and the algorithm resumes normal operation.

3.6. Multi-Scale Foreground Estimation

At the original image resolution, there may be many regions where the background model cannot confidently predict whether or not the pixels represent the foreground. This is especially true in the interior of large objects, where both the foreground and the background are relatively flat and colourless. However, this ambiguity may be resolved by considering coarser-scale representations of the scene – few scenes are flat at all scales. Therefore, when the model is uncertain whether an intensity change at a pixel represents the interior of a foreground object or a diffuse lighting change, the information is imported from coarser scales.

As shown in Figure 3, a set of background models at different scales generates a foreground score pyramid $\Gamma$. Let $\gamma_i$ represent the foreground score of a pixel at scale $i$, where $i \in \{0, ..., \omega - 1\}$, and $\omega$ represents the number of scales. Let $\theta_i = 2(\gamma_i - 0.5)$ represent the pixel’s classification ‘certainty’. Then, working down from the coarsest scale, where $i = \omega - 1$, we update the pixels foreground scores as:

$$\hat{\gamma}_i = \begin{cases} 
\gamma_i & \text{if } i = \omega - 1 \\
\theta_i \gamma_i + (1 - \theta_i) \hat{\gamma}_{i+1} & \text{if } i \neq \omega - 1 
\end{cases}, \quad (13)$$

where $\hat{\gamma}_{i+1}$ is the modified foreground score of the corresponding pixel in the next coarser scale. This merge operation propagates areas of high certainty at coarse scales into ambiguous areas at fine scales. Fine scale regions with already high certainty are not affected. Due to the nature of the LRP texture descriptor, this tends to have the effect of filling in the ambiguous interiors of objects that have high confidence bordering contours.

4. Evaluation

We have performed a thorough evaluation of the comprehensive and challenging I2R dataset, first introduced in [8].

**Dataset.** The dataset consists of 9 videos, entitled: Bootstrap, Campus, Curtain, Escalator, Fountain, Hall, Lobby, ShoppingMall, and WaterSurface, covering a variety of challenging background subtraction scenarios. Sample frames, together with the obtained results, can be observed in Figure 4.

The Lobby sequence deals with the global illumination change scenario: the lights are turned on / off. The Bootstrap scenario covers the case where, during training, parts of the background are mostly obstructed by moving objects in the scene. The challenge in ShoppingMall is to ignore the moving shadows cast by the passing people in a mall. The Hall sequence is set in an airport and covers the scenario where people can be loitering / dwelling for extended periods of time (i.e. 2 minutes).

The Campus, Curtain, Escalator, Fountain and WaterSurface sequences deal with the challenging issue of repetitive background motion. The Campus sequence is set outdoors and shows passing cars on a road, while trees are scintillating in the background. The Curtain sequence is set indoors and features a person moving in front of a fluttering curtain. The Fountain sequence observes people crossing in front of an operating water fountain, requiring the algorithm to learn a semi-transparent repetitive pattern. The WaterSurface se-

![Figure 3. Coarse-to-fine multi-scale processing example with texture as a the only feature.](http://perception.i2r.a-star.edu.sg)
sequence overlooks a body of moving water, while a person is approaching. On top of having to deal with the irregular water motion patterns, there is a relatively large change in intensity, since the appearance of the person triggers the camera’s auto-white balance mechanism. The *Escalator* sequence is set in a public indoor space, observing people in the scene while the escalators are in continuous motion.

**Evaluation Method.** For each dataset, there are 20 frames of manually marked groundtruth. A quantitative evaluation was performed using the F-measure, defined as the harmonic mean of precision and recall:

\[
F = \frac{2 \cdot RC \cdot PR}{RC + PR} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN},
\]

where RC is the recall, PR is the precision, TP are the true positives, FP are the false positives and FN are the false negatives. Higher values of the F-measure correspond to better segmentation results.

**Parameters.** A set of consistent parameters was chosen for all experiments. The codebook model parameters were set to \( n_c = 10 \), \( t_{fg} = 5 \), \( t_{bg} = 200 \) and \( t_{upgrade} = 10 \). The mixing weights for the different features were set to \( \lambda_{LRP} = 1 \), \( \lambda_I = 2 \) and \( \lambda_{RGB} = 3 \). The model update rate was set to \( \alpha = 0.01 \). There were \( \omega = 6 \) pyramid levels used for multi-scale processing.

**Implementation.** The method has been implemented in C++, making use of the OpenCV 2.1 open-source vision library. All tests were run on a standard PC with a 2.4GHz CPU and 4GB memory. In order to make the framerate performance numbers comparable with [9], the running framerate was accumulated over the 1286 frames of the *ShoppingMall* dataset, with the frame size of \( 320 \times 256 \). At maximum load, the process only required around 25MB of memory. No multi-threading was used.

**Results.** The first set of results, presented in Table 1, shows the performance of the proposed method when different combinations of features are being used: texture, intensity and colour (denoted by \( t \), \( i \) and \( c \), respectively), as well as when making use of multi-scale processing (denoted by \( H_S \), versus \( H \) for the single scale case).

The second set of results is presented in Table 2, comparing the currently proposed method with recent state-of-the-art methods [15, 9]. In addition to also benchmarking their performance on the I2R dataset, both [15, 9] include additional comparisons with other previous approaches. All competing methods are included. More specifically, [15] proposes a method that incorporates Discrete Cosine Transform coefficients (DCT), comparing with a Gaussian mixture model (GMM) method [17], a histogram of fea-

---

[http://opencv.willowgarage.com](http://opencv.willowgarage.com)
tures method (HST) [8] and a normalized vector distances method (NVD) [11]. All the results were converted to the corresponding F-measure, based on the raw TP/FP/FN values provided by the authors. [9] proposes a scale invariant local trinary pattern (SILT) method as an extension of a local binary pattern (LBP) [3], used for comparison. In addition, Table 2 contains results for a simple Gaussian method [21], as well as for the currently proposed method, entitled Hybrid (HYBD), corresponding to $H_S(c, i, t)$ from Table 1.

In addition, some qualitative results for the proposed method are presented in Figure 4.

Discussion. As it can be seen in Table 1, the results for the single scale texture descriptor alone are rather poor, with an average F-measure of $H(t) = 27.45\%$. This can be explained by the fact that many foreground areas are flat, thus leading to a small texture confidence. Because the current algorithm was designed to integrate multiple features, it can safely discard low confidence measurements. With texture alone, using multi-scale processing almost doubles the resulting F-measure: $H_S(t) = 52.14\%$. Multi-scale processing also generates significant, if less dramatic, improvements when using other feature combinations: $H(c, i) = 78.54\%$ vs $H_S(c, i) = 81.54\%$ and $H(c, i, t) = 79.97\%$ vs $H_S(c, i, t) = 82.19\%$. As expected, the best overall results are obtained when combining all features with multi-scale processing. The reason for which the computational time for $H_S(t)$ is higher than the one for $H_S(c, i, t)$ is due to the fact that more codewords and comparisons are need to explain the scene.

Table 2 shows that the proposed method performs extremely well, in comparison with other approaches, obtaining top F-measure scores on 6 out of the 9 sequences. Moreover, it obtains the best average score of 82.19% among the other 7 comparing methods. The method that comes second, with an average score of 80.33%, is DCT [15]. In addition to the better performance, one other advantage of the proposed method with respect to the DCT method is the running time. In [15], a spectral decomposition using the Direct Cosine Transform is performed using $8 \times 8$ support windows spaced 2 pixels apart, which amounts to performing 16 full image size DCTs. While exact running times for the DCT method were not published, we believe that there would be a considerable computational overhead, if we were to compare it with the current method.

5. Conclusions

We have presented a new background subtraction technique based on an extension of the traditional codebook model to multiple features (i.e., intensity, colour and texture), integrating both the feature similarity score and its confidence. In addition, we have proposed a new texture descriptor and a novel technique to aggregate foreground detection information at multiple scales. We have tested the robustness of the approach on the challenging I2R dataset. The proposed method has compared favourably with other methods, obtaining state-of-the-art performance.

References

<table>
<thead>
<tr>
<th>Sequence / Model</th>
<th>$H(t)$</th>
<th>$H_S(t)$</th>
<th>$H(c,i)$</th>
<th>$H_S(c,i)$</th>
<th>$H(c,i,t)$</th>
<th>$H_S(c,i,t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intensity (I)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colour (RGB)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Texture (LRP)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-scale</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>32.63</th>
<th>49.95</th>
<th>75.56</th>
<th><strong>77.15</strong></th>
<th>75.28</th>
<th><strong>76.80</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootcamp</td>
<td>16.34</td>
<td>41.35</td>
<td>72.66</td>
<td><strong>82.44</strong></td>
<td>71.12</td>
<td>81.79</td>
</tr>
<tr>
<td>Curtain</td>
<td>20.76</td>
<td>53.60</td>
<td>89.05</td>
<td>92.14</td>
<td>89.18</td>
<td><strong>93.11</strong></td>
</tr>
<tr>
<td>Escalator</td>
<td>35.10</td>
<td>50.14</td>
<td>60.30</td>
<td>62.30</td>
<td>63.40</td>
<td><strong>66.41</strong></td>
</tr>
<tr>
<td>Fountain</td>
<td>26.36</td>
<td>61.76</td>
<td>80.03</td>
<td>85.15</td>
<td>80.73</td>
<td><strong>85.31</strong></td>
</tr>
<tr>
<td>Hall</td>
<td>33.32</td>
<td>52.87</td>
<td>75.03</td>
<td>76.89</td>
<td>76.18</td>
<td><strong>78.49</strong></td>
</tr>
<tr>
<td>Lobby</td>
<td>38.13</td>
<td>48.06</td>
<td>83.86</td>
<td>84.18</td>
<td>84.70</td>
<td><strong>85.60</strong></td>
</tr>
<tr>
<td>Shopping Mall</td>
<td>19.03</td>
<td>55.33</td>
<td>78.11</td>
<td><strong>79.33</strong></td>
<td>77.77</td>
<td>79.07</td>
</tr>
<tr>
<td>Water Surface</td>
<td>25.39</td>
<td>56.23</td>
<td>92.30</td>
<td><strong>92.40</strong></td>
<td>92.39</td>
<td>93.11</td>
</tr>
<tr>
<td>Average</td>
<td>27.45</td>
<td>52.14</td>
<td>78.54</td>
<td>81.54</td>
<td>78.97</td>
<td><strong>82.19</strong></td>
</tr>
</tbody>
</table>

Table 1. Performance of the F-measure (%) on the I2R dataset: results of the currently proposed Hybrid method using different features and multi-scale processing. The features are t - texture, i - intensity and c - colour, whereas $H_S$ indicates that multi-scale processing is being used.

<table>
<thead>
<tr>
<th>Sequence / Model</th>
<th>G</th>
<th>GMM</th>
<th>HST</th>
<th>NVD</th>
<th>LBP</th>
<th>SILTP</th>
<th>DCT</th>
<th>HYBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootcamp</td>
<td>40.38</td>
<td>56.51</td>
<td>54.73</td>
<td>60.24</td>
<td>52.00</td>
<td>72.90</td>
<td>68.64</td>
<td><strong>76.80</strong></td>
</tr>
<tr>
<td>Campus</td>
<td>36.81</td>
<td>69.06</td>
<td>67.60</td>
<td>12.98</td>
<td>60.57</td>
<td>67.83</td>
<td>80.87</td>
<td><strong>81.79</strong></td>
</tr>
<tr>
<td>Curtain</td>
<td>65.91</td>
<td>67.34</td>
<td>52.79</td>
<td>70.22</td>
<td>71.42</td>
<td>92.40</td>
<td>91.40</td>
<td><strong>93.11</strong></td>
</tr>
<tr>
<td>Escalator</td>
<td>37.18</td>
<td>61.72</td>
<td>31.89</td>
<td>29.97</td>
<td>53.93</td>
<td>68.66</td>
<td><strong>71.83</strong></td>
<td>66.41</td>
</tr>
<tr>
<td>Fountain</td>
<td>49.78</td>
<td>75.95</td>
<td>53.74</td>
<td>49.85</td>
<td>75.33</td>
<td>85.04</td>
<td>83.64</td>
<td><strong>85.31</strong></td>
</tr>
<tr>
<td>Hall</td>
<td>46.01</td>
<td>56.46</td>
<td>38.70</td>
<td>52.63</td>
<td>50.29</td>
<td>68.02</td>
<td>76.58</td>
<td><strong>78.49</strong></td>
</tr>
<tr>
<td>Lobby</td>
<td>18.15</td>
<td>61.72</td>
<td>29.94</td>
<td>54.62</td>
<td>52.34</td>
<td>79.21</td>
<td>73.88</td>
<td><strong>85.60</strong></td>
</tr>
<tr>
<td>Shopping Mall</td>
<td>44.10</td>
<td>64.10</td>
<td>67.39</td>
<td>67.95</td>
<td>62.92</td>
<td>97.16</td>
<td>79.07</td>
<td><strong>81.74</strong></td>
</tr>
<tr>
<td>Water Surface</td>
<td>77.73</td>
<td>45.80</td>
<td>62.79</td>
<td>68.37</td>
<td>82.21</td>
<td>83.15</td>
<td>94.41</td>
<td>93.11</td>
</tr>
<tr>
<td>Average</td>
<td>46.23</td>
<td>62.07</td>
<td>51.06</td>
<td>51.87</td>
<td>62.33</td>
<td>77.43</td>
<td>80.33</td>
<td><strong>82.19</strong></td>
</tr>
</tbody>
</table>

Table 2. Performance of the F-measure (%) on the I2R dataset: comparison with other proposed methods: G - Gaussian [21]; GMM - Gaussian Mixture Model [17]; HST - Histogram of features [8]; NVD - Normalized Vector Distances [11]; LBP - Local Binary Pattern [3]; SILTP - Shift Invariant Local Trinary Pattern [9]; DCT - Discrete Cosine Transform [15]; HYBD - Hybrid, the currently proposed method.


