A Phase Discrepancy Analysis of Object Motion

Bolei Zhou[1,2]*, Xiaodi Hou[3]*, Liqing Zhang[1]
[1]MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Department of Information Engineering, The Chinese University of Hong Kong
[3]Department of Computation and Neural Systems, California Institute of Technology
zhoubolei@gmail.com, xiaodi.hou@gmail.com, zhang-lq@cs.sjtu.edu.cn

Abstract. Detecting moving objects against dynamic backgrounds remains a challenge in computer vision and robotics. This paper presents a surprisingly simple algorithm to detect objects in such conditions. Based on theoretic analysis, we show that 1) the displacement of the foreground and the background can be represented by the phase change of Fourier spectra, and 2) the motion of background objects can be extracted by *Phase Discrepancy* in an efficient and robust way. The algorithm does not rely on prior training on particular features or categories of an image and can be implemented in 9 lines of MATLAB code.
In addition to the algorithm, we provide a new database for moving object detection with 20 video clips, 11 subjects and 4785 bounding boxes to be used as a public benchmark for algorithm evaluation.

1 Introduction

Detecting moving objects in a complex scene is one of the most challenging problems in computer vision. It is closely related to a variety of critical applications such as tracking, video analysis, content retrieval, and robotics. Generally speaking, motion detection methods can be categorized into three main approaches: background modeling, detection by recognition, and view geometry.

Many models try to attack the problem of detection under controlled situations. For instance, some algorithms assume a stationary camera. This assumption leads to a branch of techniques called background subtraction. The main idea is to learn the appearance model of the background [1] [2]. A moving object in the scene is then detected by subtracting the background image from the current image. However, scene appearance captured by a moving camera, with foreground and backgrounds in arbitrary depths and viewpoints, can be very complicated. Thus, most of the background models perform poorly on moving camera recordings [3].

Another branch of popular algorithms stems from object detection and recognition. Based on pre-trained detectors, an algorithm can detect objects from particular categories, such as faces [4] or pedestrians[5]. These algorithms usually require offline training and can only handle a very limited number of object categories. Moreover, finding an invariant object detector that overcomes

* These two authors contribute equally to this paper .

illumination/view-point changes and occlusion, is already a challenge in computer vision.

To circumvent these problems, some other algorithms detect motion via camera geometry [6] [7]. This approach estimates the camera parameters under certain geometric constraints, use these parameters to compensate for camera-induced motion, and separate the moving object from the residual motion in the scene [8].

In principle, a visual system needs *only* motion cues to detect an moving object – even if the scene is disturbed by camera's ego-motion. With full knowledge of the optical flow, the mission of object detection is to find the cluster of consistent motion that is induced by the foreground. Nevertheless, the computational burdens of an optical flow algorithm is usually very heavy.

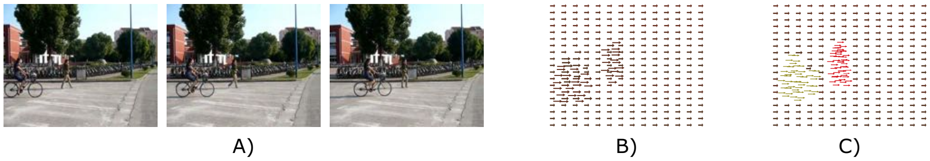

Fig. 1. An illustration of moving object detection from a perspective of optical flow analysis. **A**): A video sequence with both object motions and camera motion. **B**): The corresponding optical flow. **C**): The segmentation result that detects the moving objects.

1.1 Related work

In 2001, Vernon [9] proposed using a Fourier transform to untangle the complexity of object motions. In his theory, object segmentation and exact velocity recovery can be achieved by solving a linear system. Based on the translation property of Fourier transform, a moving object corresponds to a phase change in the Fourier spectrum. For a scene composed of m objects, exact recovery is achieved by solving a linear equation with $2m$ unknowns. The drawback of this approach is that the number m of objects must be specified beforehand. Moreover, the segmentation and velocity recovery requires observing $2m$ frames, which every object moving at a constant speed. These constraints preclude Vernon's approach from real-world applications.

1.2 An outline of our approach

We start from a similar perspective to that of Vernon: spatially distributed information can be efficiently accumulated in the Fourier spectrum. However, instead of finding the exact solution for a constrained problem, we find an approximate solution using a minimal number of assumptions.

To extract moving objects from dynamic backgrounds, our model follows the idea of predictive coding. First, we predict the next frame only considering background movements. Then by comparing our prediction against the actual observation, pixels representing the foreground emerge due to the large reconstruction error. With rigorous analysis, we show that a 9-line MATLAB approximation recovers the camera motion with bounded error.

2 The Theory

We denote $f(\mathbf{x}, t)$ as our observation at time t[1], where $\mathbf{x} = [x_1, x_2]^\top$ is the 2-dimensional vector of a spatial location. Let $\mathcal{I}$ be the ensemble of pixels. For any image, we have the partition $\mathcal{I} = \{\mathcal{F}_t, \mathcal{B}_t\}$. Every pixel belongs to the foreground $\mathcal{F}_t$ or the background $\mathcal{B}_t$.

For typical sampling rates, the ego-motion of the camera is well approximated by a uniform translation of the background. If we know this displacement $\mathbf{v} = [v_1, v_2]^\top$, we can predict the appearance of the background in the next frame based on the *intensity constancy* assumption [10] that the spatial translation does not change pixel values:

$$f(\mathbf{x}, t) = f(\mathbf{x} + \mathbf{v}, t + 1), \quad \text{where } \mathbf{x} \in \mathcal{B}_t \bigcap \mathcal{B}_{t+1} \tag{1}$$

This assumption requires that pixels $\mathbf{x}$ at t and $\mathbf{x} + \mathbf{v}$ at $t + 1$ belong to the background. We further denote $\check{\mathcal{B}}_t = \hat{\mathcal{B}}_{t+1} = \mathcal{B}_t \bigcap \mathcal{B}_{t+1}$.

Once we have the ground-truth of the ego-motion, we can reconstruct the next frame by shifting every pixel from $\mathbf{x}$ to $\mathbf{x}+\mathbf{v}$. This reconstruction is expected to perform poorly for pixels in $\mathcal{I} - \check{\mathcal{B}}_t$, the foreground. Thus, we can take the error as a likelihood function of the appearance of moving objects at certain locations. In other words, the reconstruction error map $s(\mathbf{x}, t)$ can be considered as a *saliency map* [11] for moving objects:

$$s(\mathbf{x}, t) = \Big[f(\mathbf{x} + \mathbf{v}, t + 1) - f(\mathbf{x}, t)\Big]^2. \tag{2}$$

2.1 Phase discrepancy and ego-motion

In order to generate the saliency map, we need to know the displacement vector $\mathbf{v}$. In the Fourier domain, the spatial displacement in Eq.1 can be efficiently represented by the phase of the Fourier spectrum.

Let $F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) = \mathscr{F}[f(\mathbf{x}, t) \cdot \delta_{\mathbf{x}_i}(\mathbf{x})]$ denote the 2-D Discrete Fourier transform of a single pixel, where $\boldsymbol{\omega} = [\omega_1, \omega_2]^\top$, and the indicator function $\delta_{\mathbf{x}_i}(\mathbf{x})$ is defined as:

$$\delta_{\mathbf{x}_i}(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x} \in \mathbf{x}_i, \\ 0 \text{ otherwise.} \end{cases}$$

[1] For simplicity, we only consider gray-scale images in this section. A simple extension to color images is provided in Section.3

The Fourier spectrum of the entire image $F_t(\boldsymbol{\omega})$ can be obtained by:

$$F_t(\boldsymbol{\omega}) = \sum_{\mathbf{x}_i \in \mathcal{I}} F_{\mathbf{x}_i,t}(\boldsymbol{\omega})$$

Known as the translation property [12], a spatial displacement entails a phase change, yet leaves the Fourier amplitudes intact:

$$F_{\mathbf{x}+\mathbf{v},t+1}(\boldsymbol{\omega}) = F_{\mathbf{x},t}(\boldsymbol{\omega}) e^{-i \cdot \varPhi(\mathbf{v})}, \tag{3}$$

where $\varPhi(\mathbf{v}) = \boldsymbol{\omega}^\top \mathbf{v} = \omega_1 v_1 + \omega_2 v_2$, which we call the *phase discrepancy* in the following discussions.

Because the entire background has approximately the same displacement $\mathbf{v}$, Eq.3 has a compact form for $\check{\mathcal{B}}_t$:

$$\sum_{\mathbf{x}_i \in \hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}) = \sum_{\mathbf{x}_i \in \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) e^{-i \cdot \varPhi(\mathbf{v})}. \tag{4}$$

We have the following decomposition:

$$\begin{aligned} F_{t+1}(\boldsymbol{\omega}) = \sum_{\mathbf{x}_i \in \mathcal{I}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}) &= \sum_{\mathbf{x}_i \in \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) e^{-i \cdot \varPhi(\mathbf{v})} + \sum_{\mathbf{x}_i \in \mathcal{I} - \hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}) \\ &= F_t(\boldsymbol{\omega}) e^{-i \cdot \varPhi(\mathbf{v})} - \sum_{\mathbf{x}_i \in \mathcal{I} - \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega}) e^{-i \cdot \varPhi(\mathbf{v})} + \sum_{\mathbf{x}_i \in \mathcal{I} - \hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}). \end{aligned}$$

Although it seems impossible to calculate $\varPhi(\mathbf{v})$ without the foreground/background partition, in the next section we show that good approximations of phase discrepancy is achievable in some cases.

2.2 Approximating the phase discrepancy

Since it is impossible to quantify the appearance and location of the pixels in $\mathcal{I} - \check{\mathcal{B}}_t$, we assume $F_{\mathbf{x}_i,t}(\boldsymbol{\omega})$ follows an independent normal distribution in the complex domain, that is:

$$\text{Real}\{F_{\mathbf{x}_i,t}(\boldsymbol{\omega})\} \sim N(0,1); \qquad \text{Imag}\{F_{\mathbf{x}_i,t}(\boldsymbol{\omega})\} \sim N(0,1). \tag{5}$$

For a simpler notation, we define a complex variable $z_i = F_{\mathbf{x}_i,t}(\boldsymbol{\omega})$. Let $Z_n = \sum_{i=1}^n z_i$ be the sum of this sequence. We have the following:

$$\begin{gathered} \text{Real}\{Z_n\} \sim N(0,n) \\ \text{Imag}\{Z_n\} \sim N(0,n) \end{gathered}$$

Because $|Z_n| = \sqrt{\text{Real}\{z_i\}^2 + \text{Imag}\{z_i\}^2}$, it follows a χ distribution with 2 degrees of freedom:

$$p(|Z_n| = x) = \sqrt{n}\sigma x e^{-x^2/2}. \tag{6}$$

Thus, the expectation of the spectral amplitude is determined by the number of pixels in the summation. More specifically:

$$\frac{E(|F_t(\boldsymbol{\omega})|)}{E(|\sum_{\mathbf{x}_i \in \check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega})|)} = \frac{\sqrt{\#(\mathcal{I})}}{\sqrt{\#(\check{\mathcal{B}}_t)}}. \tag{7}$$

The number of pixels in the foreground and background are estimated from our hand labeled database (see Section.3). On average, our bounding box of the foreground (an over-estimation of the actual foreground) occupies 5% pixels of the frame [2]. Thus we approximate the phase discrepancy in Eq.5 by:

$$\tilde{\Phi}(\mathbf{v}) = \angle F_{t+1}(\boldsymbol{\omega}) - \angle F_t(\boldsymbol{\omega}). \tag{8}$$

The estimation error comes from the pixels of the foreground and occluded parts of the background. The cumulative effect of these pixels at frequency $\boldsymbol{\omega}$ can be considered as added noise to variable η to the original variable $F_t(\boldsymbol{\omega})e^{-i\cdot\Phi(\mathbf{v})}$ in Eq.5, where:

$$\eta = -\sum_{\mathbf{x}_i \in \mathcal{I}-\check{\mathcal{B}}_t} F_{\mathbf{x}_i,t}(\boldsymbol{\omega})e^{-i\cdot\Phi(\mathbf{v})} + \sum_{\mathbf{x}_i \in \mathcal{I}-\hat{\mathcal{B}}_{t+1}} F_{\mathbf{x}_i,t+1}(\boldsymbol{\omega}).$$

From Eq.7, we set $F_t(\boldsymbol{\omega})e^{-i\cdot\Phi(\mathbf{v})}$ to 1 to determine the distribution of η:

$$E(|\eta|) = \frac{\sqrt{2\#(\mathcal{I}-\check{\mathcal{B}}_t)}}{\sqrt{\#(\check{\mathcal{B}}_t)}} \approx \sqrt{0.1}; \qquad \angle\eta \sim U(0, 2\pi). \tag{9}$$

The upper bound of error in $\tilde{\Phi}(\mathbf{v})$ is therefore:

$$\max\left[\Phi(\mathbf{v}) - \tilde{\Phi}(\mathbf{v})\right] = \max\left\{\tan^{-1}\left[E(|\eta|)\right]\right\} \approx 0.31. \tag{10}$$

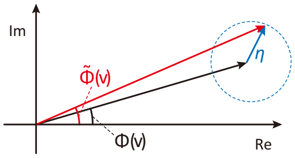

Fig. 2. A diagram of the angular error calculation. Given $E(|\eta|) = \sqrt{0.1}$, the upper bound of the angular error is 0.31 (17.6°), the mean angular error is 0.21 (12.3°)

As long as the approximation in Eq.8 holds, we can construct the estimated spectrum $\tilde{F}_{t+1}(\boldsymbol{\omega})$ from $F_t(\boldsymbol{\omega})$ and $\tilde{\Phi}$:

$$\begin{aligned}\tilde{F}_{t+1}(\boldsymbol{\omega}) = F_t(\boldsymbol{\omega})e^{-i\cdot\tilde{\Phi}(\mathbf{v})} &= |F_t(\boldsymbol{\omega})| \cdot e^{-i[\angle F_t(\boldsymbol{\omega})+\tilde{\Phi}(\mathbf{v})]} \\ &= |F_t(\boldsymbol{\omega})| \cdot e^{-i[\angle F_{t+1}(\boldsymbol{\omega})]}\end{aligned}$$

[2] In other databases such as [13] and [14], objects are in a similar size

Finally, the saliency map has the simple form:

$$\begin{aligned} s(\mathbf{x},t) &= \left\{ \mathscr{F}^{-1}\big[F_{t+1}(\boldsymbol{\omega})\big] - \mathscr{F}^{-1}\big[\tilde{F}_{t+1}(\boldsymbol{\omega})\big] \right\}^2 \\ &= \left\{ \mathscr{F}^{-1}\Big[\big(|F_{t+1}(\boldsymbol{\omega})| - |F_t(\boldsymbol{\omega})|\big) \cdot e^{-i\angle F_{t+1}(\boldsymbol{\omega})}\Big] \right\}^2 \end{aligned} \tag{11}$$

2.3 Eliminating boundary effects

The 2-D Discrete Fourier Transform implicitly implies periodicity of the signal. This property invalidates Eq.1 since pixels around the edge of the frame do not have their correspondences in the next frame. As a result, these frame-edges often have very large reconstruction errors and mislead the saliency maps (see Fig.3.C).

Assume we have two adjacent image frames. We use $\mathcal{C}_1$ and $\mathcal{C}_2$ to denote the pixels that lead to boundary effects. That is:

$$\mathcal{C}_1 = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{B}_1, \mathbf{x}_i + \mathbf{v} \notin \mathcal{I}\}; \qquad \mathcal{C}_2 = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{B}_2, \mathbf{x}_i - \mathbf{v} \notin \mathcal{I}\} \tag{12}$$

If we predict frame 2 based on frame 1 (as Eq.11 states), we will have a large error at $\mathcal{C}_1$. However, using Eq.11 we have no problem in recovering pixels in $\mathcal{C}_2$. Reciprocally, if we reverse the temporal order – reconstructing frame 1 from frame 2, only $\mathcal{C}_2$ has boundary effect.

In a more rigid format, we denote the temporally ordered saliency map that compares the predicted frame 2 with observed frame 2 as $\overrightarrow{s}(\mathbf{x},t)$, and the saliency map using reversed sequence (predicting frame 1 from frame 2) as $\overleftarrow{s}(\mathbf{x},t+1)$. We have:

$$\begin{array}{llll} \overrightarrow{s}(\mathbf{x}_i,t) > \varepsilon, & \text{where } \mathbf{x}_i \in \mathcal{C}_1; & \overleftarrow{s}(\mathbf{x}_i,t+1) \le \varepsilon, & \text{where } \mathbf{x}_i \in \mathcal{C}_1 \\ \overrightarrow{s}(\mathbf{x}_i,t) \le \varepsilon, & \text{where } \mathbf{x}_i \in \mathcal{C}_2; & \overleftarrow{s}(\mathbf{x}_i,t+1) > \varepsilon, & \text{where } \mathbf{x}_i \in \mathcal{C}_2, \end{array}$$

where ε is bounded by Eq.10.

In an elegant form, we finally eliminate the boundary effect by combining the two maps:

$$s(\mathbf{x},t) = \sqrt{\overrightarrow{s}(\mathbf{x},t) \cdot \overleftarrow{s}(\mathbf{x},t+1)} \tag{13}$$

For $\forall \mathbf{x}_i \in \mathcal{C}_1 \bigcup \mathcal{C}_2$, it is easy to see that $s(\mathbf{x}_i,t) \to 0$ as either $\overrightarrow{s}(\mathbf{x}_i,t) \to 0$, or $\overleftarrow{s}(\mathbf{x},t+1) \to 0$. The saliency map generated by Eq.13 is shown in Fig.3-D.

3 Experiments

3.1 Implementing the phase discrepancy algorithm

In MATLAB, the phase discrepancy algorithm is:

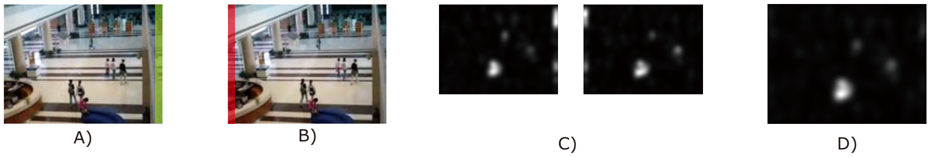

Fig. 3. An illustration of the boundary effect. **A**) & **B**): Two adjacent frames. Green and red shadows in each frame indicates $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. **C**): The saliency map based on single sided temporal order. Note that the border effect is as strong as the moving pedestrian in the center. **D**): The final saliency map.

```
FFT1=fft2(Frame1);
FFT2=fft2(Frame2);
Amp1=abs(FFT1);
Amp2=abs(FFT2);
Phase1=angle(FFT1);
Phase2=angle(FFT2);
mMap1=abs(ifft2((Amp2-Amp1).*exp(i*Phase1)));
mMap2=abs(ifft2((Amp2-Amp1).*exp(i*Phase2)));
mMap=mat2gray(mMap1.*mMap2);
```

`Frame1` and `Frame2` are consecutive frames. In our experiment, the size of image is gray-scaled and shrank to 120×160. On a 2.2GHz Core 2 Duo personal computer, this code performs at refresh rates as high as 75 frames per second.

One natural way to extend this algorithm to color images is to process each color channel separately, and combine saliency maps for each channel linearly. However, by tripling computational cost, the foreground pixels of color images does not seem to violate the intensity constancy assumption three times stronger than the gray-scale image. Indeed, our observation is corroborated by experiments. A comparison experiment of color image detection is in Section.3.3. Since our algorithm emphasizes processing speed, we use gray scale images in most of our experiments.

We also notice that in real world scenes, the intensity constancy assumption is subject to noises, such as background perturbation (moving leaves of a tree), sampling alias, or CCD noise. One way to reduce such noise is to combine the results from adjacent frames. However, we can only do so when the sampling rate is high enough such that the object motion in the saliency map is tolerable. In our experiments, we produce a reliable saliency map from 5 consecutive frames. At 20Hz, 5 frames takes about 0.25 second, this approach reduces the noise effectively without causing a drift in the salient region (see Fig.4).

3.2 A new database for moving object detection

There are several public databases for evaluating motion detectors and trackers, such as PETS [13] and CAVIAR [14]. However, very few of them considered camera motion. In this section we introduce a new database to evaluate the performance of an moving object detection algorithm.

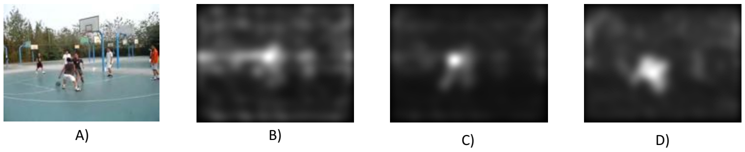

Fig. 4. A comparison of combining the saliency maps of different frames. **A**):One frame of one video clip. **B**): The saliency map computed by 2 frames. **C**): The saliency map by combining 5 frames (0.25 second). **D**): The saliency map by combining 20 frames (1 second).

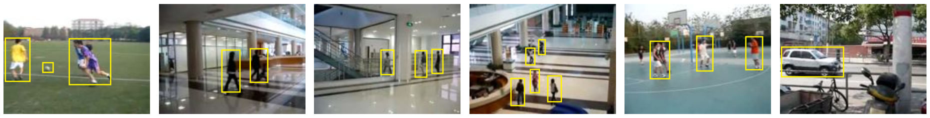

Fig. 5. Sample frames of clips in the database of object motion detection. Both scenes and moving objects vary from clips to clips.

Our database consists of indoor/outdoor scenes (see Fig.5). All clips were collected by a moving video camera under 20 FPS sampling rate. Different categories of objects are included in the video clip, such as walking pedestrians, cars and bicycles, and sports players. Given the high refresh rate, motion in adjacent frames are very similar. Therefore it is unnecessary to label every frame. The original 20 FPS videos are given to our subjects for motion detection. For labeling, we asked each subject to draw bounding boxes on a small number of key frames by sub-sampling the sequence on a 0.5-second interval. Eleven naïve subjects labeled all moving objects in the video. Some numbers from this database are in Table.1.

Items	Clips	Frames	Labelers	Key frames	Bounding boxes
Number	20	2557	11	297	4785

Table 1. A summary of our database.

The evaluation metric of the database is the same as PETS [15]. Although we have data from multiple subjects, the output of an algorithm is compared to one individual at a time. Let R_{GT} denotes the ground truth from the subject. The result generated by the algorithm is denoted as R_D. A detection is considered a true positive if:

$$\frac{Area(R_{GT} \cap R_D)}{Area(R_{GT} \cup R_D)} \geq Th, \tag{14}$$

The threshold Th defines the tolerance of a post-system that is connected to an object detector. If we use a loose criterion (Th is small) even a minimal overlap between the generated bounding box and ground truth is considered a success. However, for many applications, a much higher overlap, equivalent to a much tighter criterion and a larger value of Th, is needed. In our experiments, we use $Th = 0.5$.

For the n^{th} clip, using the i^{th} subject as the ground truth, we use GT_n^i, TP_n^i, FP_n^i to denote the number of ground truth, true positive, and false positive bounding boxes, respectively. The Detection Rate(DR) and False Alarm Rate (FAR) is determined by:

$$DR_n = \frac{\sum_i TP_n^i}{\sum_i GT_n^i} \qquad FAR_n = \frac{\sum_i FP_n^i}{\sum_i TP_n^i + FP_n^i}. \tag{15}$$

In a frame where multiple bounding boxes are presented, finding the correct correspondence for Eq.14 can be very hard. Given a test bounding box, we simply compare it against every ground truth bounding box, and pick the best match. Although this scheme does not guarantee that one ground truth bounding box is used only once, in practice, confusions are rare.

3.3 Performance evaluation

To determine bounding boxes from the saliency map, an algorithm needs to know certain parameters such as spatial scale and sensitivity. To achieve a good performance without being trapped by parameter tuning, we use Non-Maximal Suppression [16] to localize the bounding boxes from the saliency map. This algorithm has three parameters $[\theta_1, \theta_2, \theta_3]$.

First, the algorithm finds all local maxima within a radius θ_1. Every local maximum greater than θ_2 is selected as the seed of a bounding box. Then, the saliency map is binarized by threshold θ_3. Finally, the rectangular contour that encompasses the white region surrounding every seed is considered as a bounding box.

It is straightforward to assume that the parametrization is consistent over different clips in our database, and the locations of objects are independent among different clips. Therefore, we use cross-validation to avoid over-fitting the model. In each iteration, we take 19 clips as the training set to find the parameters that maximizes:

$$\sum_{m \in \{training\}} DR_m(1 - FAR_m),$$

And use the remaining clip to test the performance. The final results of DR and FAR are the average among different clips. Samples of detected objects are shown in Fig.6. The quantitative result of our model is listed in Table 2.

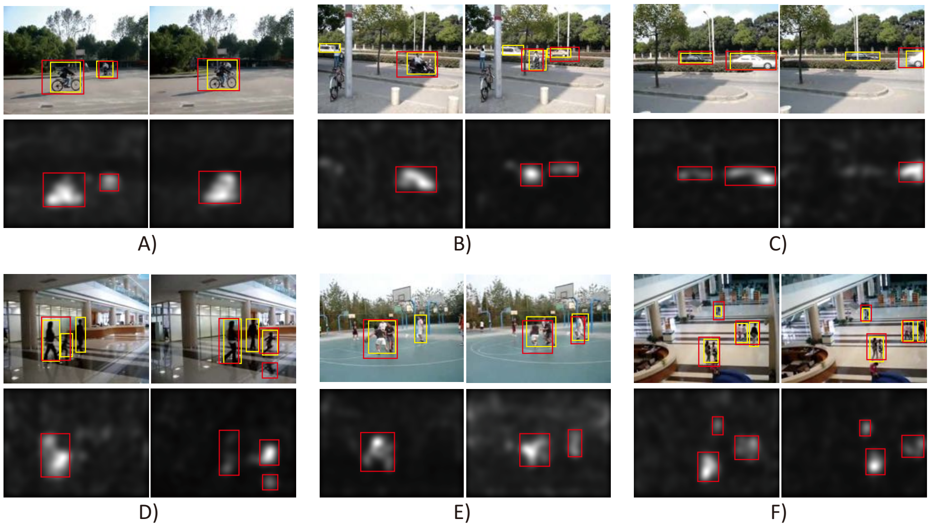

Fig. 6. Result saliency maps and the bounding boxes. In each image/saliency map pair, red bounding boxes are generated by our algorithm. Yellow bounding boxes are the ground truth drawn by a human subject.

3.4 Comparison to previous methods

To evaluate the performance of our algorithm, four representative algorithms are introduced to give comparative results on our database: the Mixture of Gaussian model [1], the Dynamic Visual Attention model [17], the Bayesian Surprise model [18], and the Saliency model [11]. MATLAB/C++ implementation of all these algorithms are available on authors' websites. Examples of the generated saliency maps are shown in Fig.7. As for the quantitative experimental part, the parameters of Non-Maximal Suppression is trained in the same way as we described in Section.3.3 to generate bounding boxes from the saliency maps. The quantitative results are shown in Table.2. Our phase discrepancy model is the best in detecting moving objects.

It is worth noting that not all of these algorithms are designed to detect moving objects in a dynamic scene. In fact, the performance of an algorithm is determined by how well its underlying hypothesis is consistent with the data. In our database, an "object" is defined by its motion in contrast to the background. There is no assumption such as objects possessing unique feature, or background being monotonous. Therefore, it is not surprising that some algorithms did not perform very well in this experiment.

3.5 Database consistency

The motivation behind the analysis of database consistency comes from the fact there is no objective "ground truth" for moving object detection. Although

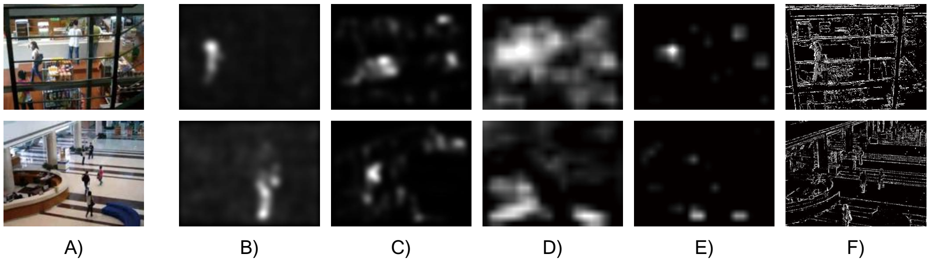

Fig. 7. Saliency maps generated by different algorithms. **A**): Original image. **B**): Our model. **C**): Dynamic Visual Attention [17]. **D**): Bayesian Surprise [18]. **E**): Saliency [11]. **F**): Mixture of Gaussian [1].

	Detection Rate	False Alarm Rate
Human average	0.84 ± 0.08	0.15 ± 0.08
Our model	0.46 ± 0.14	0.58 ± 0.24
Our model (color)	0.48 ± 0.18	0.57 ± 0.24
Dynamic Visual Attention [17]	0.32±0.22	0.86±0.10
Bayesian Surprise [18]	0.12±0.09	0.92±0.04
Saliency [11]	0.09±0.08	0.98±0.01
Mixture of Gaussian [1]	0.00±0.00	1.00±0.00

Table 2.

ground truth consistency issue is not widely concerned in the object detection and tracking databases, List *et.al.* [19] analyzed the statistical variation in the hand label data of CAVIAR [14], and showed that inter-subject variability can compromise benchmark results. In our database, we also observed that the same video clip can be interpreted in different ways. For instance, in Fig.8.A, some subjects label multiple players as one group, yet other subjects label every individual as one object.

A good benchmark should have consistent labels across subjects. To evaluate the consistency of our database, we assess the performance of the i^{th} subject based on the j^{th} subject's ground truth. Therefore, for each individual we have 10 points on the FAR-DR plot. As a comparison, the performance of our algorithm is also provided. Each data point is generated by selecting one individual as the ground truth and perform cross-validation over 20 trials. The result is shown in Fig.8.

From these results we see that even a human subject cannot achieve perfect detection. In other words, a computer algorithm is "good enough" if its performance has the same distribution as humans' on the FAR-DR plot.

Threshold and accuracy tolerance Note in Eq.14, the choice of $Th = 0.5$ is arbitrary. This parameter determines the detection tolerance. To evaluate Th's influence, FAR and DR are computed as functions of Th (see Table 3).

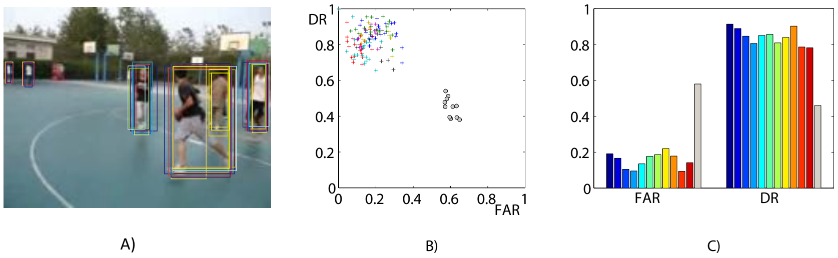

Fig. 8. A): Different interpretations of moving objects by different subjects. This image overlays the bounding boxes of 11 subjects. Boxes in the same color are drawn by the same person. We see that the incongruence among different subject is not negligible. **B**): The FAR-DR plot of all subjects and our algorithm. Each + in the same color represents the assessment of the same subject. Each ○ indicates the performance of our algorithm. Among different subjects the DR fluctuates from 0.65 to 1, whereas the FAR fluctuates from 0 to 0.4. The average human performance is $FAR = 0.15 \pm 0.08$, $DR = 0.84 \pm 0.08$.**C**): Color bars indicate the FAR and DR for the subjects. The gray bars is the performance of our algorithm.

Th	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Human Detection Rate	0.92	0.91	0.91	0.90	0.88	0.84	0.77	0.62	0.37	0.15	0.00
Human False Alarm Rate	0.07	0.07	0.07	0.08	0.11	0.15	0.22	0.37	0.62	0.85	1.00
Model Detection Rate	0.83	0.82	0.80	0.75	0.63	0.46	0.20	0.07	0.02	0.00	0.00
Model False Alarm Rate	0.18	0.20	0.24	0.31	0.43	0.58	0.80	0.93	0.98	1.00	1.00

Table 3. Human average (DR,FAR) and model average (DR,FAR) with respect to threshold.

The influence of object sizes As we have shown in Eq.10, the upper bound of error is a function of object size. To provide a empirical validation of our algorithm performance on large objects, we selected 2 clips in our database that contains the biggest objects, and tested our algorithm. The average area of the foreground objects is 10% of the image size (comparing to 5% of the original experiment). The new performance is shown in Table 4.

	Original experiment	Clips with large objects
Detection Rate	0.46 ± 0.14	0.41 ± 0.14
False Alarm Rate	0.58 ± 0.24	0.65 ± 0.08

Table 4. The algorithm performance over large object database. The performance drop is small.

4 Discussion and Future Work

4.1 Sources of errors

One of the challenges is to estimate the bounding boxes for adjacent, sometimes occluded objects that move in the same direction (such as in Fig.6F). To unravel the complexity of multiple moving objects, either long term tracking, or a more powerful segmentation from saliency map to bounding boxes is required.

In some cases, we also need to incorporate top-down modulations from a level of object recognition. Since the saliency map is a pixel based representation, it favors moving parts of an object (such as a waving hand) over the entire object. A canonical interesting example is in Fig.6D: our algorithm identifies the reflection on the floor as an object. Yet none of our subjects labeled the reflection as an object.

4.2 Connections to Spectral Residual

In 2007, Hou *et.al.* proposed an interesting theory called the Spectral Residual [20]. This algorithm uses the Fourier transform of a single image to generate the saliency map of the static scene. As a follow-up paper suggests [21], the actual formulation of the Spectral Residual algorithm is to take the phase part of the spectrum of an image, and do the inverse transform. In other words, the saliency map generated by the Spectral Residual is the asymptotic limit of Phase Discrepancy when the second frame has $\mathbf{v} \to 0^+$. However, $\mathbf{v} \to 0^+$ is ill-defined in our problem, as the displacement approaches infinitesimal, no motion information will be available. To fully unveil the connections between these two algorithms, further research on the statistical properties of natural images is necessary.

4.3 Concluding remarks

In this paper, we propose a new algorithm for motion detection with a moving camera in the Fourier domain. We define a new concept named Phase Discrepancy to explore camera motions. The spectrum energy of an image is generally dominated by its background. Using this, we derive an approximation to the phase discrepancy. A simple motion saliency map generation algorithm is introduced to detect moving foreground regions. The saliency map is constructed by the Inverse Fourier Transform of the difference of two successive frames spectrum energies, keeping the phase of two images invariant. The proposed algorithm does not rely on prior training on a particular feature or categories of an image. A large number of computer simulations are performed to show the strong performance of the proposed method for motion detection.

5 Acknowledgement

The work was supported by the National Natural Science Foundation of China (Grant No. 60775007, 90920014).

References

1. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Volume 2. (1999) 246–252
2. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Volume 2. (2004)
3. Cheung, S., Kamath, C.: Robust techniques for background subtraction in urban traffic video. Video Communications and Image Processing, SPIE Electronic Imaging **5308** (2004) 881–892
4. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision **57** (2004) 137–154
5. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2009) 304–311
6. Tian, T., Tomasi, C., Heeger, D.: Comparison of approaches to egomotion computation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (1996) 315–320
7. Han, M., Kanade, T.: Reconstruction of a scene with multiple linearly moving objects. International Journal of Computer Vision **59** (2004) 285–300
8. Irani, M., Anandan, P.: A unified approach to moving object detection in 2D and 3D scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 577–589
9. Vernon, D.: Fourier vision: segmentation and velocity measurement using the Fourier transform. Kluwer Academic Publishers (2001)
10. Black, M., Anandan, P.: A framework for the robust estimation of optical flow. In: Proc. IEEE Conf. on International Conference of Computer Vision. (1993) 231–236
11. Itti, L., Koch, C., Niebur, E., et al.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence **20** (1998) 1254–1259
12. Mallat, S.: A wavelet tour of signal processing. Academic Press (1999)
13. : (http://ftp.pets.rdg.ac.uk.)
14. : (http://homepages.inf.ed.ac.uk/rbf/caviar/)
15. Bashir, F., Porikli, F.: Performance evaluation of object detection and tracking systems. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2006). (2006)
16. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Cengage-Engineering (2007)
17. Hou, X., Zhang, L.: Dynamic Visual Attention: Searching for coding length increments. Advances in Neural Information Processing Systems **21** (2008) 681–688
18. Itti, L., Baldi, P.: Bayesian surprise attracts human attention. (2006) 547–554
19. List, T., Bins, J., Vazquez, J., Fisher, R.: Performance evaluating the evaluator. In: Proc. IEEE Joint Workshop on Visual Surveillance and Performance Analysis of Video Surveillance and Tracking. (2005)
20. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR07). IEEE Computer Society, Citeseer (2007) 1–8
21. Guo, C., Ma, Q., Zhang, L.: Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2008)