IDENTIFYING OPTICAL TRAP

Yulwon Cho, Yuxin Zheng

12/16/2011

1. BACKGROUND AND MOTIVATION

Optical trapping (also called optical tweezer) is widely used in studying a variety of biological systems in recent years. In optical trapping a focused laser beam provides an attractive or repulsive force (typically on the order of pico-newtons), depending on the refractive index mismatch to physically hold and move microscopic dielectric objects. A free microscopic particle in fluid is doing Brownian motion all the time while the trapped particle has significantly lower Brownian motion and stay confined in the trap. When the trap region is big and the trapping strength is not strong multiple particles will aggregate around the trap.

The goal of this project is to identify and locate the optical trap based on the video clips of the fluorescent particle doing Brownian motion in various fluidic environments with the possible presence of other aggregation mechanisms such as convection and thermophoresis. Besides, the particles are of different sizes, resulting in fast or slow Brownian motions. Moreover, the particle may stop moving by simply adsorbing to the substrate. All of the above effects results in patterns similar to optical trapping in some aspects but not exactly the same and therefore can be distinguished using appropriate machine learning algorithms.

The data set collection is done by recording the optical trapping experiments performed in our lab. Two distinct models are attempted in terms of data processing and feature set design. The first one is based on the tracking of individual particle motion. The second approach is based on the principle component analysis of the temporal correlation map of each video. The algorithms we use are Bayesian logistic regression and SVM. Then we compare the two in terms of accuracy and computational efficiency. Finally we combine some of the features from the two models and achieve a test accuracy of 92.5%.

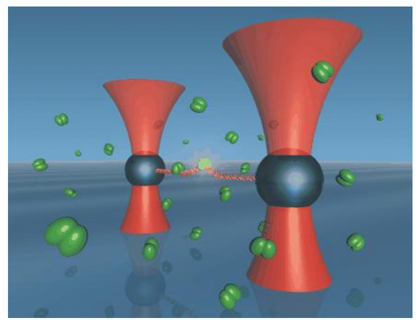

Fig 1. Illustration of optical trapping. Red cones are focused laser light and blue spheres are trapped beads.

2. DATA SET GATHERING

We have made 80 video clips of from the experiments in which half of them have optical trap. The polystyrene particles are suspended in pure water in various concentrations and their diameters range from 200nm to 1um. Each video is recorded in NTSC format (~30 frames / sec) for 3 seconds and the frame size is 320x240. Here are some snapshots of the videos.

As can be seen from Figure 2 in the next page, it is not possible to identify optical trap based on a single frame of the video. It is necessary to know the motion of the particle. Besides, the other non-trapping aggregation effects and various fluid conditions make identification much more challenging.

Trapping cases:

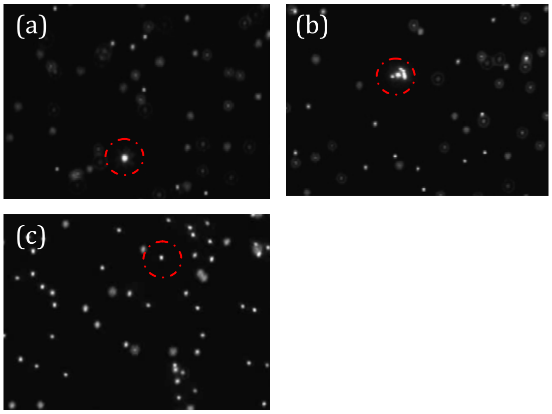

Fig 2. Trapping cases: (a) trapping of multiple particles, (b) weak trapping, (c) trapping of single particle

No trapping cases:

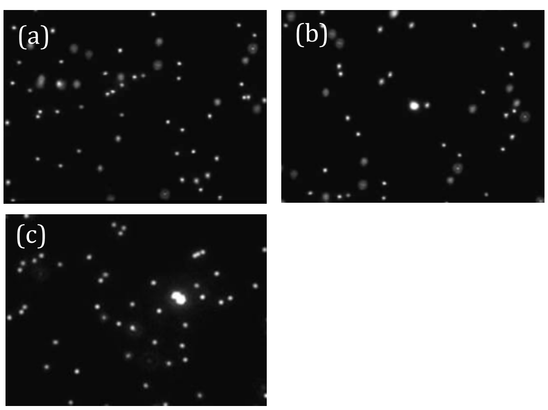

Fig 3. Non-trapping cases: (a) pure Brownian motion, (b) a big particle at the center, (c) heat-induced particle aggregation

3. ALGORITHM DESIGN AND TEST RESULTS

3.1. Approach #1 (Particle Motion Tracking)

In the first approach, we extract features based on the particle tracking results. Tracking is done via a free Matlab particle tracking program written by Prof. E.W. Hansen from Dartmouth University, since our focus is on the machine learning algorithm design. This is an example plot of the trajectory of a trapped particle.

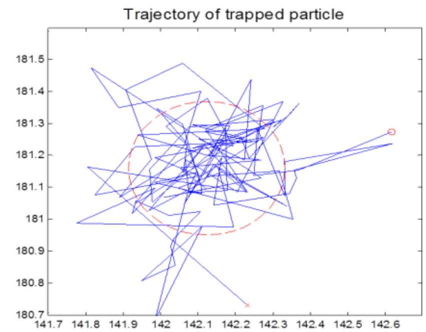

Fig 4. Example trajectory of a trapped particle.

3.1.1. FEATURE EXPLORATION

BASIC FEATURE SET

We focus on three special particles which are most likely to be optically trapped: (1) the one with smallest motion variance, (2) the one with biggest radius and (3) the brightest. For each of the three particles we extract the motion variance, the radius and the brightness to form 9 independent features $x_j\,(j=1\sim 9)$.

EXTENDED FEATURE SET

We added more features such as the square of each basic feature elements and their product terms. Our initial insight was that nonlinear decision boundary will be achieved by those extended terms. Our optical trapping examples have complicated characteristics. For example, the particles that have too low motion variance could be physically stationary particles, which is not an optical trap. At the same time, the particles with high variance could also be non-trap particles. Therefore, by adding square terms, we expected that when the value is too large, the small and negative coefficient of square terms will drive the final decision into non-trap. On the other hand, when the value is not large enough, the coefficient of the 1st order term is expected to be more influential. In addition, our thermodynamic analysis suggests that the product of variance, radius, and brightness has physical implication to identify optical traps.

DENSITY-BASED FEATURE SET

We also consider the fact that the number of particles in an optical trap will gradually become larger as time goes on. Actually, this feature turns out to be effective to differentiate the optical traps from stationary particles or heat induced aggregation of particles. Specifically, we divide the whole screen into N by N grid areas, and take the linear regression of density variance over time. As a result, we obtain two more features C1 and C2 as follows.

$$Density\,(x,y,t) = C_1(x,y)\cdot t + C_2(x,y)$$
$$(t = \textit{frame number},\ (x,y) = \textit{position})$$

Our close observation also found an interesting fact that the distribution of (C_1, C_2) for each example has different but consistent shape, depending on whether it is an optical trap or non-trap including stationary particles and heat-induced aggregation. Thus, we extracted one more feature which is a ***principal axis*** of the distribution.

3.1.2. FEATURE SELECTION

Totally, we have 93 feature candidates. Then, we perform logistic regression algorithm on those features. At the same time, we use cross validation and forward search technique to find the feature set that has the highest test accuracy.

When we first tried the basic feature set $x_i x_j (i, j = 1\sim 9)$, only three features: x_1, x_2, x_3 are enough to give the highest test accuracy of 87% where x_1 is the smallest mean square variance of motion and x_2, x_3 are the radius and brightness of the same particle, respectively, and training accuracy is 90%. After we added the extended feature set, one more feature which is the product of x_1 and x_2 is significant. As a result, the training accuracy becomes 93.75% and test accuracy is 88%. Lastly, after we added the density based feature, x_{92} and x_{93} are significant, so the training accuracy increased to 95% and testing accuracy of 91%.

	Basic Feature Set
$x_1\sim x_3$	variance, radius, brightness of the particle with small motion variance
$x_4\sim x_6$	variance, radius, brightness of the particle with biggest radius
$x_7\sim x_9$	variance, radius, brightness of the brightest particle
	Extended Feature Set
$x_{10}\sim x_{18}$	Square of basic features
$x_{19}\sim x_{90}$	product of different basic features
	Density based Feature Set
$x_{91}\sim x_{92}$	Linear regression coefficients (density = x_{91}·time+ x_{92})
x_{93}	The principal axis of the distribution (x_{91}, x_{92})

Figure 4 shows the decision boundary projected on (x1, x2) plane, and Figure 5 shows the average test error rate for each example when all the features are added.

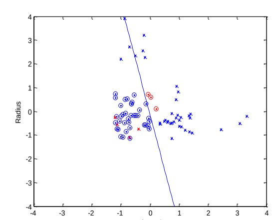

Fig 4. Data point and decision boundary projected on (x1, x2) feature subspace

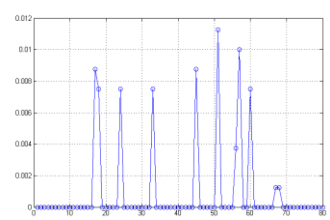

Fig 5. Test error plot generated by cross validation

3.1.3. ANALYSIS OF THIS APPROACH

Even though we achieved 95% training accuracy by adding all these features while excluding useless features, the test accuracy has been improved by only a little. Therefore, we analyze the original video data that have the highest error rate (wrong labeling in testing). We find that most of them are cases where the trapping is weak or particles cluster and adsorb to the substrate. In the first case the variance of particle motion is not small enough to be identified as optical trap. In the latter case, the individual particle behaves just like it is optically trapped. However, a close inspection reveals that those similar cases actually have different spatial pattern. For example, in weak trapping case, the particles tend to move along a ring-shape path around the center of the trap. In the adsorption case, particles adsorb all over the substrate. In short, optical trapping could be more identifiable if the spatial pattern is taken into account.

3.2. Approach #2 (Temporal and Spatial Correlation)

In our second approach, we try to design features that convey both temporal information and spatial information. At the beginning, we simply make the whole video a long vector which is a super high dimension feature and feed it to SVM. But we always get 100% training accuracy and less than 50% test accuracy, suggesting high variance in learning. So we redesign the feature. We calculate for each pixel in the frame the autocorrelation function.

$$g(i,j,m)=\sum_k f(i,j,k)\,f(i,j,k-m)$$

where f is pixel value (brightness), (i, j) is pixel position in the frame and m is frame number. Then we obtain the effective width of the autocorrelation function of time to roughly represent the duration for which the pixel remains bright.

$$\tau(i,j)=\frac{\sum_m g(i,j,m)}{\max_m g(i,j,m)}$$

Thus $\tau(i,j)$ is what we call temporal correlation map. We have verified that it effectively captures Brownian motion suppression in optical trapping and other aggregation phenomena. Large magnitude in the map indicates that there are particles hovering around. Some examples are shown below.

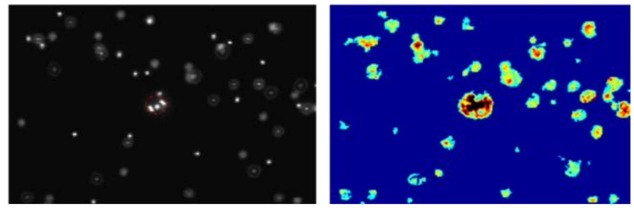

Fig 6. Optical trapping: (a) snapshot of the video and (b) its temporal correlation map

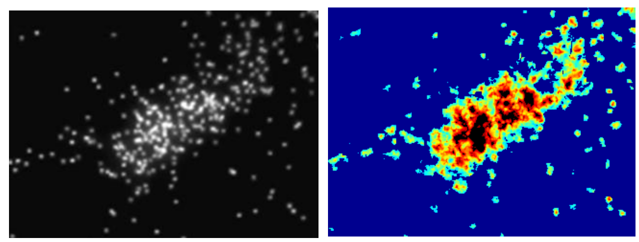

Fig 7. Thermal aggregation (no trap): (a) snapshot of the video and (b) its temporal correlation map

ALGORITHM

Firstly, we compute $\tau(i,j)$ for all pixels, and find the maximum τ as a potential trapping location. Secondly, we take a smaller window (30x30) around the selected location, and perform PCA (Principal Component Analysis) to reduce the dimension of feature. Lastly, we feed this feature to SVM.

RESULT

Comparing with our initial SVM scheme, the dimension of the feature is reduced from 320*240 (frame size) to 35 (number of influential principal components).

Training accuracy of both schemes are 100%, but the test accuracy of the new algorithm using the 35 ***eigen-pattern*** is 90% whereas the initial SVM schemes gives less than 50%.

If we use the 30x30 correlation map as a training vector without performing PCA, the test accuracy is 85%. However, when we choose 35 principal components, the high variance problem is partly fixed, so the final test accuracy becomes 90%.

A few eigenpatterns are shown below.

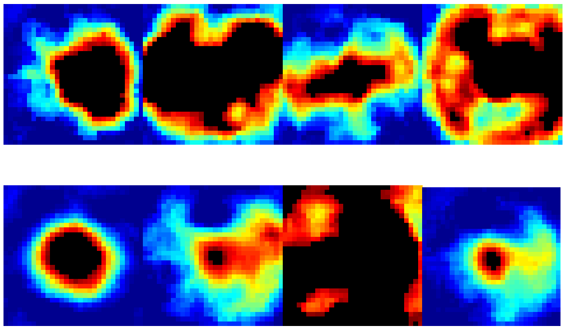

Fig 8. Examples of eigenpatterns generated by PCA

DISCUSSION

SVM algorithm using these features gives a test accuracy of 90%. Again cross validation technique is used to obtain reliable test accuracy. While this method has similar accuracy to the previous one, it is much faster than the other as it does not involve tracking all the particles. When tested in our computer (Intel i5 core 2.67GHz, 4GB DRAM), this method is roughly 48 times faster than the previous method.

3.3. Combining of two approaches

Finally, we have combined the two approaches by adding a few more features to the first approaches. In the second approach, we used 35 dimension of vector as a feature for SVM. However, instead of using the full size of vector, we extracted the following three distinguishing factors.

- Radius of potential trap regions
- Maximum τ
- The number of potential trap regions

After adding these three features, the training and testing accuracy increased as follows.

Feature set	Training accuracy	Test accuracy
Basic feature set	90%	87%
+ extended feature set	93.75%	88%
+ density based feature set	95%	91%
+ pseudo eigenpattern feature set	98%	92.5%

This is expected as the two methods capture different aspects and are therefore complimentary.

4. CONCLUSION

We have applied different machine learning algorithms to identify optical trapping effect in complex environment where other similar aggregation mechanisms can occur. One is based on particle tracking and the other is based on temporal autocorrelation of each pixel of the video. The two modeling methods give comparable test accuracy but the second one is more computationally efficient. By combining the features of the two methods we achieve a high accuracy of 92.5%. Last, this project may be generalized and extended to identify various physical, chemical or biological processes in more complicated systems, which may be of more interest in terms of applications.