Multi-users Real-Time Interaction with Bacterial Biofilm Images Using Augmented Reality

Mohammadreza Hosseini[1], Tomasz Bednarz[2], and Arcot Sowmya[1]

[1] UNSW, Sydney NSW, Australia
{Mhosseini,Sowmya}@Cse.unsw.edu.au
2 CSIRO, Brisbane QLD, Australia
Tomasz.Bednarz@Csiro.au

Abstract. Augmented Reality (AR) applications may be used to enhance understanding of physical objects by addition of digital information to captured video streams. We propose new bio-secure system for interactions with bacterium biofilm images using the AR technology to improve safety in experimental lab. In proposed application we used state-of-the-art real-time features detection and matching methods. Also, various methods of feature detection and matching were compared with each other for real-time interaction and accuracy. The implementation of an app on a tablet device (Apple iPad) makes it useable by multi users in parallel.

Keywords: Multi-user, Real-time, biofilm, Augmented reality.

1 Introduction

Bacteria can reproduce simply and rapidly by doubling their contents and splitting in two. A colony of bacteria that sticks to a surface forms a biofilm. Furthermore, information such as the biofilm diffusion coefficient, bacterium dimension and trajectory are among quantities that scientists are interested in to understand and possibly explain the effect of new drugs on single species of bacteria. Computer vision and imaging techniques could be utilised to support better understanding of those mechanisms by helping to localize, track and measure bacteria features. Also, use of interactive visualisation techniques could enhance users' understanding; for instance, the user could explore naturally complex interior structures and morphology of bacteria during the course of biofilm formation. User interactions with visualization systems may be carried out using either a touch-based interface such as a keyboard and mouse, or a touchless interface such as gesture recognition cameras.

In the bio-imaging space, the user has the ability to pause a biofilm evolution movie and call up data annotations extracted from the database by selecting a bacterium. Based on an earlier study [1], users are more willing to use touch-based interfaces compared to a touchless ones. In most situations only one person interacts with the system. Additionally, users could use mobile handheld devices to capture biofilm fragments and call up augmented information on the top of it.

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part II, LNCS 8526, pp. 298–308, 2014.

In visualisation setup, any number of users could interact with the system at a time, without interfering with other users or even collaborating with them. For instance, tapping on a bacterium in the biofilm evolution movie watched through a tablet camera, could display related information on the tablet display held in the user's hand. The same augmented data displayed on a dynamic moving bacterium, must also be available to the user in following frames until the user selects another bacterium.

Bacterium morphological properties may vary from frame to frame. Therefore, determining a specific bacterium in the biofilm on tap commands, from an underlying moving image taken by a tablet camera in real time, is the major challenge of this research. The initial prototype will assume that tablet devices are aware of the frame number currently displayed on a large screen or hemispherical dome. Image cross-correlation techniques will allow detection of the biofilm sub image that will be further used to find a corresponding bacterium automatically.

Displaying a 3D object on the surface of a marker (a point of reference) and estimating camera position to stabilise the object is not a new concept [18]. Many algorithms have implemented and are available in various SDKs [16-18]. The API function allows displaying the virtual information over a predefined visual marker. Interaction using AR without a predefined marker is classified as marker-less AR, where any part of the real environment may be used as a target to be tracked in order to place a virtual object on. Marker-less augmented reality relies heavily on natural feature detection in images received through the camera. As soon as a known physical object is detected, the appropriate virtual object may be displayed over it. The detection of a known objects require that the features in an unknown image watched through a camera are matched with feature from a known object. Features are parts of an image that can be used for image matching or object detection and can usually be classified into three categories: regions, special segments and interest points [4]. Interests points such as corners are usually faster to detect in images and more suitable for real-time applications. Scale-invariant feature transform [5], Speeded Up Robust Features [6] and Harris [7] corner detector methods have been used widely in the literature to detect features but heavy mathematical computation involved in any of these methods may slow down an application significantly. SCARF [8] and ORB [9] are the recent attempts to improve the speed of feature detection.

Feature descriptors are used to describe image structure in the neighbourhood of a feature point. Using the descriptors, a feature points in an image can be matched with features in other images. SIFT, SURF and ORB are among feature descriptor methods that are rotation and scale invariant. Other feature descriptors such as BRIEF [10] and Daisy [11] are designed to be fast by sacrificing the rotation and scale-invariant properties. Similar feature points (i.e. points with similar feature descriptors) in source and destination images may represent the same point on single object in separate views. Matching features using brute force search (search among all features in the destination image)[12] is very time consuming and has little use in real-time applications. FLANN [15] is a library for performing fast approximate nearest neighbour searches

in high dimensional spaces. The library includes a collection of algorithms for performing the nearest neighbour hood search and a system for automatically picking the best algorithm based on the data characteristic. Based on a survey [13] the Vantage-Point tree is a good method for estimating the nearest neighbour for matching feature descriptors. The vantage-point tree has the best overall construction and search performance and is used in this work.

2 System Implementation

The system for interacting with biofilm images through an AR application is configured as shown in Figure 1.

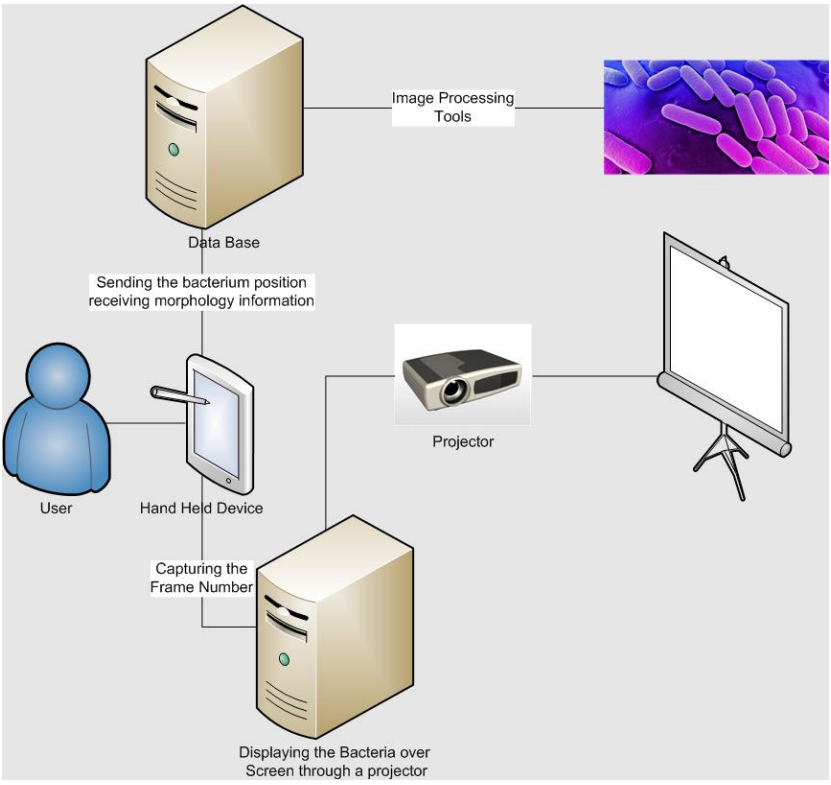


Fig. 1. Overall System Configuration

A bacterium tracking method [2] is used to extract morphological properties of each bacterium in every frame. The information is stored in a database, which can be accessed individually based on the bacterium position in the biofilm.

The biofilm evolution movie, which is displayed over a wall surface by a projector, is viewed through the camera of a handheld device (tablet). Interaction with images displayed on the wall is by an augmented reality application.

The following tasks are to be performed in order to add virtual information archived for every bacterium in the database through an AR application:

1. A server displays the biofilm movie on the large screen and continuously updates a variable used to keep track of the frame number.
2. Users watch the movie through the camera of handheld device (tablet). The video filmed by the camera is displayed on the handheld device.
3. Tapping on a single bacterium in the live video watched on a handheld device triggers the information retrieval process.
4. The frame number is fetched from the server.
5. The bacterium position in the biofilm image that is matched with frame number is calculated.
6. Bacterium position is used to extract the required information from database.
7. Bacterium is highlighted and information is displayed back (augmented) on the handheld device.
8. Until the next tapping, the previously detected bacteria position will be used to update the location of the bacterium virtual information in subsequent image. This is more explained in section 2.5.

The major concern as discussed before is the ability to locate the position of a tapped bacterium in the biofilm sub image on the handheld device. The methods used and the reasons for selecting them will be described in the following sections.

2.1 Feature Detector

Detection of the features and matching descriptors around a tapped bacterium is used to retrieve bacterium position in the original biofilm image. FAST is one of the fastest corner detection methods suitable for real-time applications [3]. It is based on the pixel intensity comparison around a circular neighbourhood of a query point. Each pixel in the circle is labeled from 1 to 16 clockwise. If *a set of N contiguous pixels in the circle is all brighter than the intensity of candidate pixel p plus a threshold value t or all darker than the intensity of candidate pixel p minus threshold value t, then p is classified as a corner.*

2.2 Feature Descriptor

We used BRIEF as the feature descriptor. The formulation of the BRIEF descriptor as follows:

1. Select a series of N, (X, Y) where $X=(x_1,y_1)$, $Y=(x_2,y_2)$ are location pairs around the feature point.

2. $$T(X,Y,P)=\begin{cases}1, & p(X)<p(Y)\\ 0, & p(X)\geq p(Y)\end{cases} \quad \text{for every selected } (X,Y) \qquad (1)$$

Here *P represents a patch around a feature point, T represents a test on patch P and p(X) is pixel intensity at pixel X*. Performing the above operation on *N* pairs creates a binary string as a feature descriptor. Selecting the size of the patch around the feature point and selection of *N* pairs can affect the accuracy of the method. A study [10] shows that selection of the pairs from a double Gaussian distribution or selecting randomly around the feature points can produce better results. The advantage of the binary string as a descriptor is the ability to calculate the distance between every pair (e.g. Hamming distance) very quickly on many processors [10]. Our experiments show that the major bottleneck of marker-less AR is the feature matching part (section 2.3). So selecting a not very sophisticated descriptor that is fast to calculate is the only way to achieve near real-time AR. This is discussed more in section 3.

2.3 Feature Matching

We use Vantage-Point as the feature matching method. The matching method itself and a technique to improve the search speed are discussed in the following section.

Vantage-Point (VP) Search Tree Construction. The idea of constructing a binary search tree is to divide the search space recursively based on a similarity measurement in order to increase search speed, which is possible by pruning nodes that cannot be better than the best answer already found. Rather than partitioning points on the basis of relative distance from multiple centres (as is the case with k-means), VP-tree splits points using the absolute distance from a single centre. Tree construction begins by assigning all points to the root node, and then recursively partitioning the points into one of several children of the node. This process continues until some termination criteria are met. Two common criteria are the maximum leaf size (the leaf contains fewer than a given number of points) and the maximum leaf boundary [12].

The algorithm for constructing the Vantage-point tree with hamming distance as a measure of similarity between two bit strings is summarized in Algorithm VantagePointTree.

```
VantagePointTree (lower, upper)

If Termination condition is met, return
Create a node
Select a random bit string in the search space as the
vantage point and place it in the node
Sort the other bit string in ascending order based on
their distance to the Vantage-Point
Select the median bit strings
Keep the distance between the vantage-Point and the me-
dian as the vantage-point boundary in the tree node
Node leftchild =VanatagePointTree(lower+1,median)
Node right child=VanatagePointTree(median,upper)
```

In this algorithm, lower and upper are the indices of an array used to store the binary strings. The search algorithm is shown in Algorithm VantagePoinTree-Search.

```
VantagePoinTreeSearch (target, node, σ)

If node=Null return
dist= distance(node , target)
If dist < σ
  Keep tree root
  keep σ
if leftchild(node) =empty and rightChild(node)=empty re-
turn
if dist < node threshold
  if dist-σ <= node threshold
    VantagePoinTreeSearch(taget,node left child, σ)
  if dist+σ >= node threshold
    VantagePoinTreeSearch(taget,nde right child, σ)
else
  if dist+σ >= node threshold
    VantagePoinTreeSearch(taget,tree right child, σ)
  if dist-σ <= node threshold
    VantagePoinTreeSearch(taget,node left child, σ)
```

In this algorithm σ is the smallest distance that has been found so far. The target binary string is a descriptor of a feature in the source image.

The tree construction is performed for every frame in the original biofilm evolution movie beforehand, so that the tree construction phase does not have any effect on application processing speed.

Increasing Search Speed Using Triangle Inequality. The search algorithm may not need to process all the points in a leaf if the distance between every point in a leaf and the leaf node is calculated during tree construction. Let $\{b_1,b_2\ldots b_n\}$ be the points in the leaf, B the leaf node and $d(b_1,B) > d(b_2,B)>\ldots>d(b_n,B)$ where d is the Hamming distance. Based on triangle inequality we have

$$d(target,B)<d(target,b_i)+d(B,b_i) \text{ for } i=1,2,\ldots,n \quad (2)$$

$$d(target,B)-d(B,b_i)<d(target,b_i) \text{ for } i=1,2,\ldots,n \quad (3)$$

So $d(target,B)-d(B,b_i)$ is the lower bound for the $d(target,b_i)$. If at any stage of searching a leaf point we find $\mathbf{j}\in\{1\ldots n\}$ where $d(target, B)-d(B,b_j) > \boldsymbol{\sigma}$, the algorithm will stop searching the other points, as the distance between the target and

remaining leaf points will be higher that $\boldsymbol{\sigma}$ since $d(b_i,B)$ $i \in \{1,\ldots,n\}$ are sorted in descending order.

2.4 Matching and Outliers Removal

In Fig. 2, pairs of feature matching of the source image (images on left are viewed through handheld device camera) and destination image (biofilm images stored in a database) are displayed. As the images show although there are some correct matches there are also many mismatches that must be removed before further processing. Estimating homography using RANSAC [14] can be used to remove the outliers. The result after removing the outliers is shown in Fig. 3.

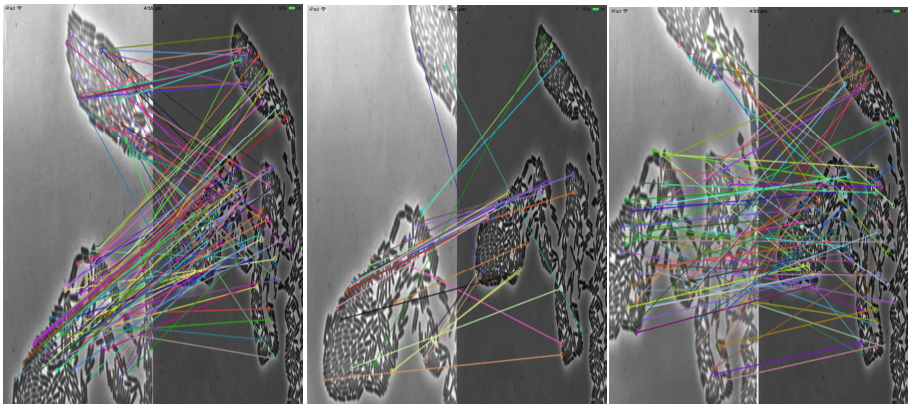


Fig. 2. Matching feature based on similarity of feature descriptor

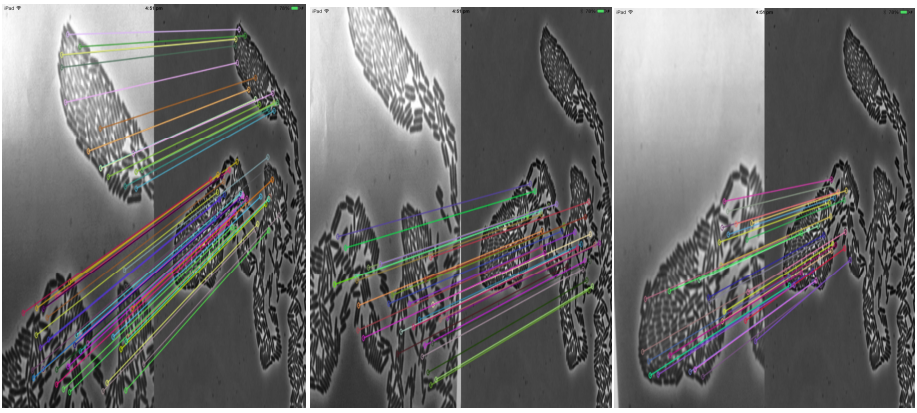


Fig. 3. Removing outliers

2.5 Bacteria Position Retrieval and Displaying Information

Homography matrix is used to translate the tapped position in held device coordinates to image coordinates. A search inside the database is carried out to find the closest bacterium. The information for this bacterium will then be displayed on handheld device. The inverse of the homography matrix and the bacterium position in image coordinates is also used to track the position of the last tapped bacterium in subsequent frames before any new tapping. This can be used to display the virtual information at the right position even if the handheld device moves in a different direction (Fig. 4).

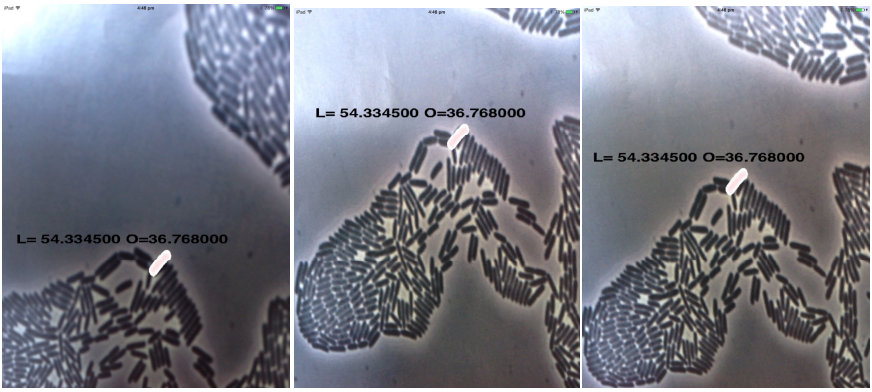


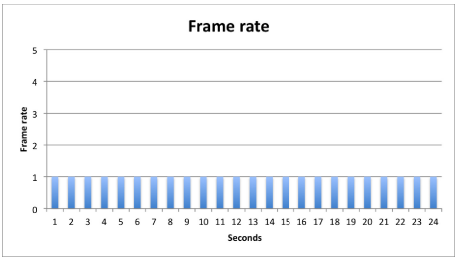
Fig. 4. Displaying the information in right position in different device orientation

3 Experimental Results

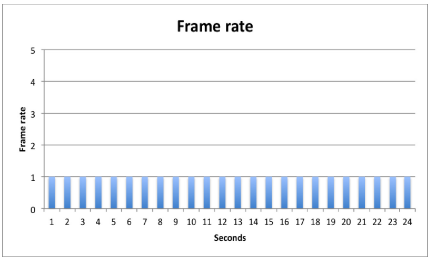
The application frame rates when implemented using different combination of feature detector and descriptor methods is calculated. The application runs for 30 seconds and frame rate was recorded prior to feature matching. As Fig. 5 shows, the combination of FAST feature detector and BRIEF feature descriptor method (Fig. 5 c) is the best choice for a real-time application. It is necessary to mention that this result is valid for high-density biofilm image sets and may not be valid for other image sets.

The accuracy of application was also evaluated and compared with other implementation of the application using different feature detection and matching methods. The application accuracy is estimated by measuring the acceptable range of device rotation. The acceptable range is the maximum rotation in every direction before the application loses the bacterium position between two consecutive taps (refer to section 2.5). This is carried out by comparing the positions extracted from inverse homography of different matching methods with results from SURF matching inverse

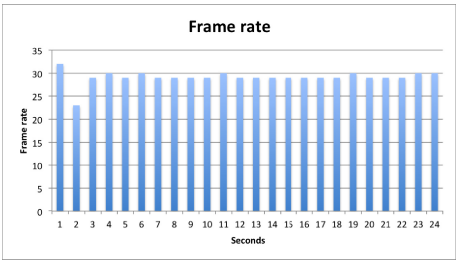
homography method in different device orientations. The reason for selecting SURF as the base model is because of its rotation and scale invariant properties. The results are shown in Fig. 6. These images are produced when the device rotated around the vertical axis. Fig. 6 shows that FAST/BRIEF feature matching acceptable device rotation range is limited to [-5.05, 25.80] (Fig. 6 b) which is shorter that other rotation and scale invariant feature detector and descriptor. This means that the user can only use the application in situation where there are no significant changes in handheld vertical device orientation.



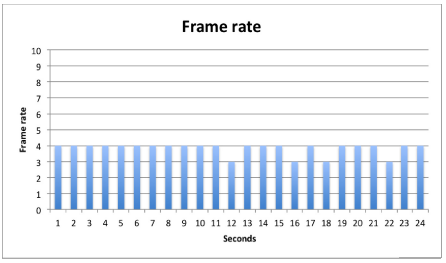
a) SIFT Detector, SIFT Feature Descriptor



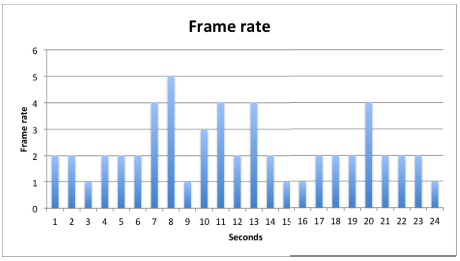
b) SURF Detector, SURF Feature Descriptor



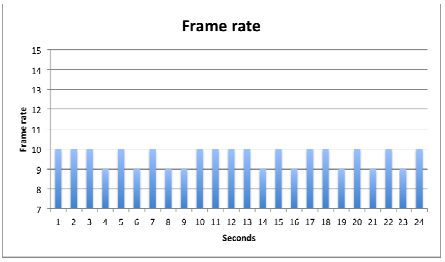
c) FAST Feature Detector, BRIEF Feature Descriptor



d) FAST Feature Detector, SURF Feature Descriptor

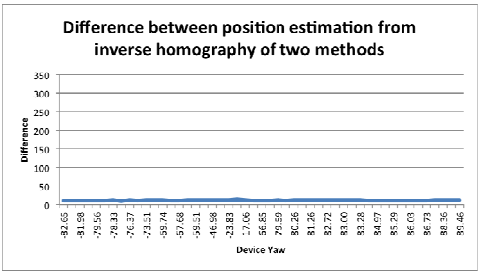


e) FAST Feature Detector, SIFT Feature Descriptor

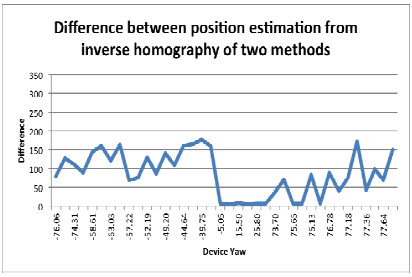


f) ORB Feature Detector, ORB Feature Descriptor

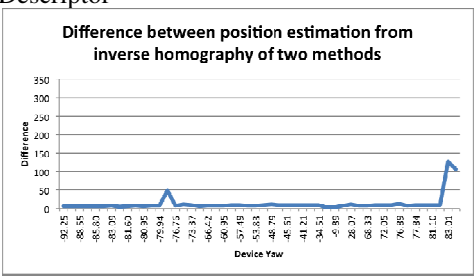
Fig. 5. Frame rate achieve during 30 seconds experiments using different method



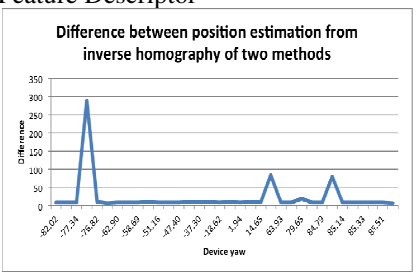
a) FAST Feature Detector, SURF Feature Descriptor



b) FAST Feature Detector, BRIEF Feature Descriptor



c) ORB Feature Detector, ORB Feature Descriptor



d) FAST Feature Detector, SIFT Feature Descriptor

Fig. 6. Difference between estimated positions using inverse homography of various methods and SURF feature and descriptor matching

4 Conclusions

Lower processing power of handheld devices in comparison with desktop computers raise the necessity of developing a low-computational approach for real-time application. Employing a feature descriptor method, which is not scaled and rotation invariant was an approach used in this paper. The application lets the user experience a real-time AR but limited device acceptable rotation, drop usability of the application. The whole experiments reveal that a real-time and a rotation and scale invariant feature detector and descriptor in high-dense environment are still an ongoing research.

References

1. Hosseini, M., Vallotton, P., Bednarz, T., Sowmya, A.: A Study of Touchless Versus Touch-based Interactions with Bacterial Biofilm Images. In: 12th ACM International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI 2013). The Chinese University of Hong Kong, Hong Kong (2013)

2. Vallotton, P., Sun, C., Wang, D., Ranganathan, P., Turnbull, L., Whitchurch, C.: Segmentation and tracking of individual Pseudomonas aeruginosa bacteria in dense populations of motile cells. In: Image and Vision Computing New Zealand (IVCNZ), Wellington, New Zealand (2009)
3. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
4. Gundogdu, E., Alatan, A.A.: Feature detection and matching towards augmented reality applications on mobile devices. In: 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON) 2012, October 15-17, pp. 1,4 (2012)
5. Lowe, D.G.: Distinctive image features from scale-invariant key points. IJCV 60(2), 91–110 (2004)
6. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
7. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference (1988)
8. Thomas, S.J., MacDonald, B.A., Stol, K.A.: Real-time robust image feature description and matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part II. LNCS, vol. 6493, pp. 334–345. Springer, Heidelberg (2011)
9. Rublee, R., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), p. 13 (2011)
10. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
11. Tola, E., Lepetit, V., Fua, P.: A Fast Local Descriptor for Dense Matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
12. Nielsen, F., Piro, P., Barlaud, M.: Bregman Vantage Point Trees for Efficient Nearest Neighbor Queries. In: Proceedings of IEEE International Conference on Multimedia and Expo, pp. 878–881 (2009)
13. Kumar, N., Zhang, L., Nayar, S.: What Is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images? In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 364–378. Springer, Heidelberg (2008)
14. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Commun. Assoc. Comp. Mach. 24, 381–395 (1981)
15. Muja, M., Lowe, D.G.: FLANN – Fast Library for Approximate Nearest Neighbors, `http://people.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN`
16. Metaio Augmented Solutions, `http://www.metaio.com`
17. Qualcomm Vuforia, `https://www.vuforia.com`
18. ARLab, `http://www.arlab.com`