Real time Interactive Image Segmentation Using User Indicated Real-world Seeds

Rafael Beserra Gomes*,Rafael Vidal Aroca*,Bruno Motta de Carvalho†,Luiz Marcos Garcia Gonçalves*

* Department of Computing Engineering and Automation - UFRN - Natal, Brazil
† Department of Informatics and Applied Mathematics - UFRN - Natal, Brazil

Abstract—We propose a novel and fast interactive segmentation methodology for computer vision applications. Basically, the proposed system performs the tracking of seeds so that multiple seeds can be acquired over time, substantially improving the segmentation results. Moreover, instead of image coordinates, the user indicates points in the real-world that become seeds in the image. These seeds can be indicated, for example using a laser pointer or a smart-phone. The seeds can then be tracked and used by a segmentation algorithm. Experiments using the Lucas-Kanade Optical Flow and the Fast Multi-Object Fuzzy Segmentation (Fast-MOFS) algorithm demonstrate that the proposed technique successfully segments images in real-time and improves the user ability to directly segment an object in the real world. The proposed system has a high performance, allowing it to be used with high frame rates in devices with low processing capability and/or with restricted power requirements.

Keywords-Interactive Computer Vision, Real-time, Seed Selection, Fuzzy Image Segmentation

I. INTRODUCTION

Segmentation algorithms are important parts of many computer vision systems. Their main purpose is to separate objects from the background scene so that other techniques can be applied only to the objects of interest. Some segmentation algorithms use a seed (usually provided by the user) to start the segmentation process [1]. These algorithms have different nomenclatures such as semi-automatic segmentation, interactive segmentation, marker-based segmentation or seed-based segmentation.

Most interactive segmentation systems need user input in the form of mouse clicks. The input can be a single point or a set of points inside the region of interest. Some techniques also require one or more seeds outside the region of interest to identify the background.
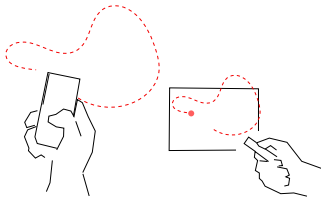

Fig. 1. Interactive segmentation using real-world seeds: using a cellphone and a laser.

Although commonly used in image editing software, another application for these algorithms are computer vision systems embedded in mobile devices, such as tablet computers or smart phones. Mobile devices are becoming pervasive in the current world [2] and computer vision systems have been increasingly used [3]. Although modern smart phones have high performance processors running faster than 1 GHz, efficient implementations of segmentation techniques are still needed to reduce processing time and battery consumption, as well as to improve the user experience.

Moreover, Zhong et al. argues that interactive segmentation is the most flexible way to get what the user wants from an image [4]. This is suitable for mobile devices because the user is usually holding the device and pointing it to an object of interest. Furthermore, instead of providing the seeds directly into image coordinates, we consider that the object can be indicated by the user in the real world using, for example, a laser pointing device. Figure 1 shows examples of an user pointing to the object of interest in the real world. In robotic vision this can be useful, as the user can indicate the boundaries of an object to a robot [5].

Another possible application of this segmentation system is for fruit grading. Fruit classification systems using computer vision typically use segmentation algorithms to extract and separate all fruits in a scene and classify them individually, which takes more time than classifying a single fruit. In that way, if a single fruit needs to be analysed, the user can point the camera with the attached laser to that fruit, instructing the system to segment only the pointed fruit. Moreover, with the laser point projected on the fruit, the distance from the camera to the fruit can be computed, which also allows computation of geometrical information of the fruit in the real world, such as area and perimeter.

This paper proposes a technique that uses points in the world, projected by a laser (see Figure 2), as seeds for a fast fuzzy segmentation algorithm. Moreover, the system tracks these seeds so that multiple seeds can be acquired over time, improving the segmentation results. The system can be used for videos or still images. We remark that the laser is not pointed to any display, but it is directly pointed to the object to be segmented in the real world.

In order to be adequately used in mobile devices and robotics vision, a list of requirements must be met by the algorithm as follows:

1) The seeds are specified by the user in the real world.
2) The seeds can be provided by the user only once for

Fig. 2. Example of laser projection tracking along 255 frames. Darker lines and points show points indicated by the user more recently.

each frame before segmentation of this frame;

3) The seeds can be provided at any time;
4) The system should be aware of spurious seeds and ignore them;
5) The image segmentation should be performed in real-time;

The proposal in this article is a methodology that fills all the requirements above. The article is organized as follows: first we describe related interactive segmentation systems (Section II), then our method is described in Section III. Next, Section IV describes results and a performance evaluation is presented in different computer architectures, and finally we discuss the results in the conclusion (Section V).

II. RELATED WORKS

Several applications need accurate semantic object segmentation. In such cases, automatic segmentation algorithms hardly obtain the desired segments of an image. In these cases, usually, high-level information provided by the user is used to aid the segmentation system [6], [1].

According to Zhong et al [4], many interactive segmentation systems have been proposed recently. This section presents a brief overview of some interactive segmentation algorithms. More details and a comparative evaluation of interactive segmentation algorithms can be found in the work of MacGuinness and O'Connor [6].

It is common for segmentation applications to use seeds indicated by the user using a display interface and an input device (e.g. mouse). In other applications there is another interaction mechanism to indicate these seeds. Zwinderman [5] uses a laser pointer in order to indicate real world 3D points that are associated to SURF features. These features compose clusters and the largest one correspond to the 3D object.

One of the simplest segmentation systems available is the seeded region growing algorithm, originally proposed by Adams and Bischof [7]. This algorithm uses one set of points as seeds inside a region of interest and then grows from these seeds using connected pixels with similar colors. Some variations also use seeds in the background to help differ between regions.

A more recent technique, called Simple Interactive Object Extraction in Still Images (SIOX) [8] uses color signatures to segment objects using one or more seeds in the foreground object. According to the authors, SIOX performs well even in scenes where it is difficult to construct a ground truth. This system can be used for both still images and video. One of its drawbacks is that pictures with different illumination conditions are not well segmented.

Interactive graph cuts were first proposed by Boykov and Jolly [1]. The system builds a graph where each node represents a pixel and the edges connect adjacent pixels. The algorithm uses the seeds as constraints to minimize a cost function and cut the graph in the best place to separate the background from the foreground. Graph cuts are one of the most popular interactive segmentation systems and several variations of this technique have been proposed. One popular variation is the GrabCut created by Rother et al. [9]. The improvement over the original graph cut is that the user can interactively brush areas to improve the segmentation results. Furthermore, in many situations the user does not need to provide seeds specifically in the foreground object, but only a bounding rectangle around the object of interest.

Herman and Carvalho proposed an efficient fuzzy based segmentation system [10], [11] that uses a discrete affinity table (with values in the range [0,1]) between spels (an abstraction for pixels). The algorithm verifies the affinity of the seed pixels to its neighbors and adds each neighbor pixel to the affinity table at the index related to the affinity of this spel. The system starts expanding high affinities until all spels have their maximal label or a threshold is reached. The method proposed in this article uses the algorithm Fast-MOFS [11].

Another approach, advocated by Kang and Wang, mixes seeded region growing with fuzzy similarity between pixels. The similarity is used to automatically select the seeds [12].

Watershed is a classical segmentation algorithm originally introduced by Beucher and Lantuejoul [13]. It is based on an analogy with real world topology where a constrained area of land is flooded by water until a barrier is reached. In images, the same concept applies. A seed is put in a local minimum where the flooding starts until a contour is encountered. Although popular, some authors argue that Watershed is too slow for real time applications [14].

As we show in Section IV, our method, detailed in Section III, produces segmentation results that are qualitatively similar to some of the mentioned algorithms, but that is much faster, being able to execute in real time in desktop computers or mobile devices. In opposite to [5], the points detected in this work are used as seeds to an region growing based image segmentation instead of using them to segment 3D clustered points.

III. PROPOSED TECHNIQUE

The proposed system consists of two main modules. One is responsible for the segmentation of part of an image using the Fast-MOFS algorithm [11] based on a given seed set. The second part consists on a novel technique to track the seeds over a sequence of frames to improve the robustness of the segmentation in frame sequences or videos. Note that since
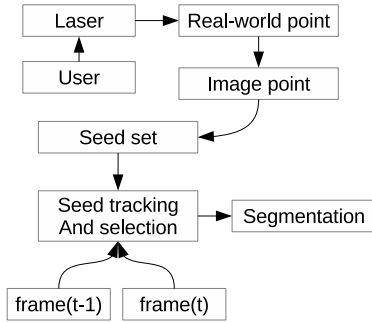
Fig. 3. Flowchart of the system.

this seeds tracking system is independent of the segmentation algorithm it can be applied to other segmentation techniques, such as GrabCut, Watershed, or any other seeded segmentation system.

As mentioned before, our technique segments objects using a set of seeds indicated by the user in the real world, but if the camera or the object moves, then the seeds' coordinates in image space, in general, do not correspond to the same point anymore. In this case, the seed tracking system is applied to keep the seeds approximately fixed relative to the object being segmented. Our implementation is based in the Optical Flow algorithm of Lucas-Kanade [15]. However, while pointing the object, the user can occasionally inadvertently indicate parts of the image that do not belong to the object (see Figure 1). To solve this problem, we propose a method called *selection by distance* that discards outlier seeds in such cases.

Each frame is processed as depicted in the flow chart of Figure 3. First, the user indicates a point in the real-world using a laser. Next, the system finds the projection of this point in the image using a specific computer vision algorithm for the type of user interaction (see Section III-A). This point is added to the seed set if it meets the criteria defined by the seed selection module. After that, all seeds from the seed set are tracked by the tracking algorithm. If a seed cannot be successfully tracked, then this seed is discarded. Finally, the remaining seeds are used by the segmentation algorithm. Before the segmentation step, a Gaussian smooth filter was used. This filter reduces high frequencies from the image and, therefore, leads to a smoother segmentation.

A. Seed detection

The first step of the system is the seed detection. Most of our experiments were based on a low cost laser pointer fixed below the camera and parallel to its optical axis, so that the user can point the object of interest with the red dot projected by the laser. However, since the laser is too bright, it hides local information where it is projected, usually by overlapping a red circle with high saturated pixels at the center. Adding a seed outside this circle is a straightforward solution. In practice, there are no major problems if this seed is still inside the object. Another solution would be using a digital output line to switch the laser on and off between frames so that the laser would be used to detect the seed but not in the image to be segmented.

To detect the laser pointer we make the assumption that the projected point is in the region with the brightest pixels of the image in the red channel. The algorithm looks for a region with at least 5 red pixels in a cross pattern with highest values. If the red laser point cannot be used, an infrared laser could be used so that the laser point will appear for the camera but not for the users in the real world. Additional advantages of using the laser is that it provides enough information to reliably measure the distance from the camera to the object being segmented [16] and consequently allowing the computation of its real area and perimeter.

A simpler interface for the user, specially in mobile devices, consists in a user interface with a transparent shooting target symbol in the middle of an image preview so the user can point the device adjusting the object of interest to be in the shooting target center. Then the seeds would be selected in the image center.

B. Seed selection

Coordinates pointed by the laser are candidate seeds. Each candidate seed has an associated counter (s) that is increased at each frame if the current position of the laser is inside a circle of radius r centered at that seed. When a seed is added, its *time to birth* (t_b) period is started, but it is not yet included in the seed set. When this period finishes, its counter is compared to the time of birth. If it is less than a percentage (p_b) of the number of frames contained in the time of birth period, then this seed is discarded. Otherwise, it becomes part of the seed set (see Figure 4). The time to birth is necessary to prevent a point outside the object to be consolidated as seed.

Figure 5 shows points indicated by the user ("a" and "b"). If the point is inside a circle of radius r, the associated counter with that seed is incremented. Otherwise, a new candidate seed is created.

If the radius r is too small, then the seeds' counters are updated less often. On the other hand, if the radius r is too big, the spurious' seeds counter may be increased, even thought they are far from the current position of the laser.

After this stage, the seed enters in the so called *time to live* period. In this stage, the seed is checked every frame. If the counter is less than a percentage (p_l) of the time since the seed was created, then this seed is discarded.

To select and accept the seeds we suppose that all the pixels belonging to the object have the same probability of being used. This restriction is needed in order to classify a seed as an outlier and discard it. The idea is that an outlier tends to be less counted than the seeds in the object, since these inner seeds are closer to the current laser point. The lower the counter of one seed is in relation to its life time, the higher is the probability that this seed can be discarded. As shown in Figure 6, each seed has a lifetime.

C. Fuzzy Segmentation

The basic concept that we are generalizing here is that of fuzzy connectedness [17], by making use of the Fast-MOFS
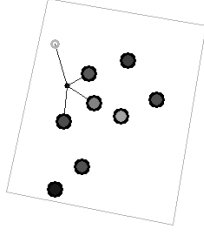
Fig. 4. Each circle represents a seed and its gray level represents the division of its counter by the number of frames that this seed exists (a lighter gray level means a high seed count and a darker one means a low seed count). The small black dot represents the current laser position and the lines indicate which seeds have its counter increased. The rectangle is the boundary of the ground truth. The circle with light gray boundary represents a seed in the *time to birth* period.
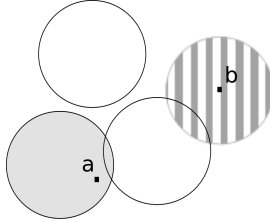

Fig. 5. The point "a" updates the counter of the seed centered at the darker circle. The point "b" is not inside any circle, then a new seed is placed and a new circle (hatched circle) is centered at it.

algorithm proposed by Carvalho [11]. The goal is to partition a set V (in our case, an image) into a specified number of objects, but in a fuzzy way; i.e., in addition to assigning a spel (short for spatial element) to a particular object, we also assign a grade of membership of that spel to the object. This grade of membership is a number between 0 and 1, where 0 indicates that the spel definitely does not belong to the object, and 1 indicates that it definitely does. In order to do that, we assign, to every ordered pair (c, d) of spels, a real number not less than 0 and not greater than 1, which is referred to as the *fuzzy connectedness* of c to d.

In the approach used below, fuzzy connectedness is defined in the following way. We call a sequence of spels a *chain*, where its *links* are the ordered pairs of consecutive spels in the sequence. The strength of a link is also a fuzzy concept, since the ψ-*strength* of a link is the value of a *fuzzy spel affinity* function $\psi : V^2 \to [0, 1]$, i.e., a function that assigns a value between 0 and 1 to every pair of spels in V. A chain is formed by one or more links and the ψ-*strength of a chain*, as it happens in real life, is the ψ-strength of its weakest link. The ψ-strength of a chain with only one spel in it is 1 by definition.

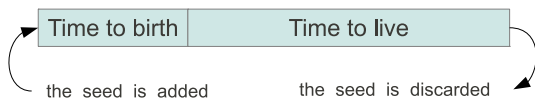The fuzzy connectedness from c to d is computed using a

Fig. 6. The lifetime of a seed.

fuzzy connectedness function $\mu_\psi : V^2 \to [0, 1]$ defined by the ψ-strength of the strongest chain from c to d. Then, we can define the ψ-*connectedness map f* of a set V for a seed spel o as the picture formed by the fuzzy connectedness values of o to c ($f(c) = \mu_\psi(o, c)$), for all $c \in V$. By using this map, one *hard* or *crisp object C* can then be defined on a ψ-connectedness map by selecting a threshold t and associating with C all spels c for which $f(c)$ is above this threshold, i.e., $C = \{c | c \in V, f(c) > t\}$. An *M-segmentation* is an unique *M-semisegmentation* in which all spels have their maximal connectedness.

For all definitions related to the fuzzy segmentation algorithm, as well as the proofs associated to the main theorem that states the existence and uniqueness of the segmentations, please refer to [11]. The Fast-MOFS algorithm [11] is a greedy algorithm that computes the grades of membership of all spels to the objects. The values of the fuzzy spel affinities are rounded to three decimal places, thus, allowing us to use an array as a priority queue, instead of a heap, and lowering the computational complexity of the algorithm.

Since we are segmenting only one object, and we do not have other objects competing for the spels and segmenting the background, we had to modify the original algorithm. We modified the described algorithm to use the distance from the seeds to the pixel being segmented in order to avoid the segmented area to easily "leak" from the region of interest. In this way, the segmented pixels are always nearby the seeds. We redefined the original ψ-*connectedness map f* as:

$$f(c) = \mu_\psi(o, c) * (1 - \frac{g(c)}{d_t}), \tag{1}$$

where $g(c)$ is the length of the strongest chain between a seed and the spel c and d_t is a positive distance factor. If d_t is low then the chains cannot be very long. Note that $f(c)$ can be less than 0. In this case, $f(c)$ is set to zero. Since we are interested in segmenting only one object, in order to avoid to compute affinities for all the pixels in the image, an affinity threshold (t_a) is specified. A pixel is not part of the segmented object if its affinity is below this threshold.

D. Seed tracking

Since the camera and/or the object can move, the seeds will also move as well. In such cases, the seeds should be tracked in order to stay fixed in relation to the moving object of interest. Any tracking algorithm can be used for these cases. In our experiments, we use the Multi-scale Optical Flow proposed by Lucas Kanade [15].

To obtain better performance, instead of tracking classical image features such as the ones proposed by Tomasi [18], our implementation tracks only the detected seeds. Therefore, each seed is moved by the optical flow vector from one frame to the next. If the difference between a patch around the original seed and the moved points is higher than a threshold, this seed is discarded.
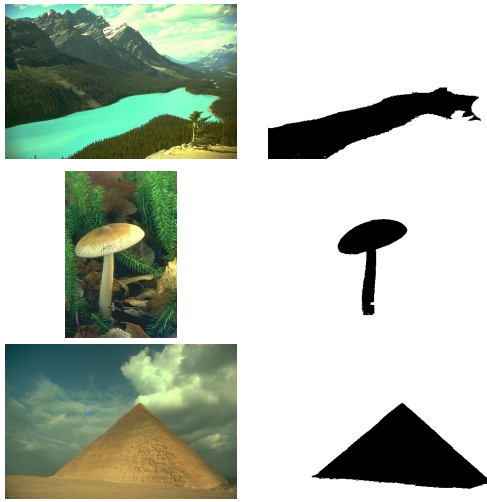
Fig. 7. Original images and ground truths used in the evaluation. Source: Berkeley Segmentation Database (BSD) [19].

Fig. 8. Original images with chosen seeds(red marks) and Fast-MOFS segmented images. Accuracy rates (compared to ground truth of Figure 7) for the 3 images are, respectively, 95.17%, 95.49%, 86.74%.

IV. EXPERIMENTS AND RESULTS

The main objectives of our experiments are to show that the proposed system can be efficiently used on several types of devices, specially mobile devices, and to show that the resulting segmentation is comparable to resulting segmentations performed by classical algorithms. First we present results of the Fast-MOFS in a single frame with timings and segmented images. Then, we present the results for the integrated system using Fuzzy segmentation and seed tracking.

The system was executed and its performance measured on two different platforms: a PC (x86) laptop computer with a 2.2GHz processor (configured to run at a fixed frequency) and a Pandaboard development board with a Cortex-A9 ARM processor running with a fixed speed of 1GHz. This ARM processor is the same used in many cell phones, tablet computers and several other mobile devices. Both systems run the Linux operating system (kernel 2.6.34) and the algorithm was developed using the OpenCV-2.3.1 computer vision library.

A. Image Segmentation

The test images and their respective ground truth images were obtained from the Berkeley Segmentation Database (BSD) [19]. We modified the ground truths, that originally had several image segments, to have only one segment, as shown in Figure 7. These images and ground truths were then used to evaluate several segmentation algorithms. The resolution of the test images are 321x481, 481x321 and 321x481 pixels.

In order to compare our system performance, we executed several classical seeded segmentation algorithms. We used the WaterShed and GrabCut implementations from OpenCV and we implemented the Region Growing and Fuzzy segmentation in C++. Figures 8, 9, 10 and 11 show the original images with the given seeds marked in red and the resulting images after segmentation. The resulting images were compared to the ground truth (Figure 7) to produce the segmentation accuracy rates. The comparison consists on counting how
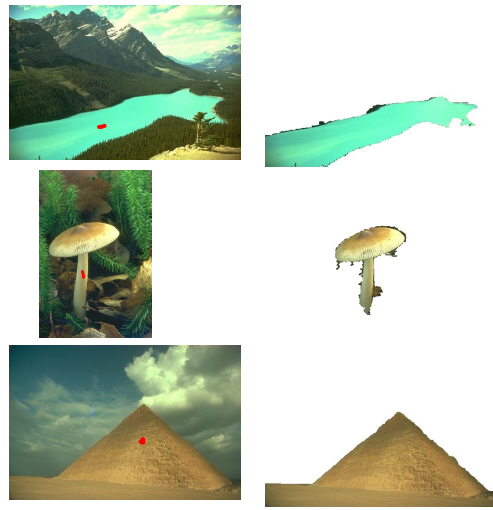

Fig. 9. Original images with chosen seeds (red marks: foreground seeds, white marks: background seeds) and GrabCut segmented images. Accuracy rates (compared to ground truth of Figure 7) for the 3 images are, respectively, 73.94%, 95.71%, 91.09%.

many pixels on the background of the ground truth are on the background of the resulting image and how many pixels of the ground truth's foreground (black) are on the segmented image foreground. Then we compute the percentage of these pixels related to the total amount of pixels of the image and call it accuracy.

Since the same seed set causes different results for each algorithm, we experimented with several seeds for each algorithm and tested them with the minimal seed set that provides similar segmentation results among the tested algorithms.

For these still images, no Gaussian filter was applied and $t_a = 5$ was used for the pyramid and the plant. For the lake, $t_a = 8$ was used. As can be seen in Figures 8, 9, 10, and 11 the Fast-MOFS segmentation obtained results that are very similar to classical algorithms such as Watershed and GrabCut,
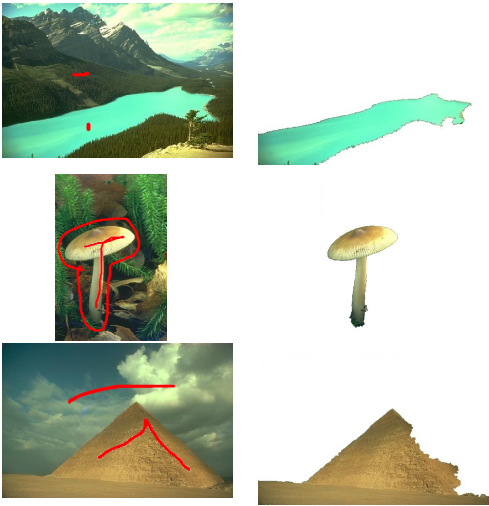
Fig. 10. Original images with chosen seeds (red marks) and Watershed segmented images. Accuracy rates (compared to ground truth of Figure 7) for the 3 images are, respectively, 89.37%, 89.65%, 80.88%.
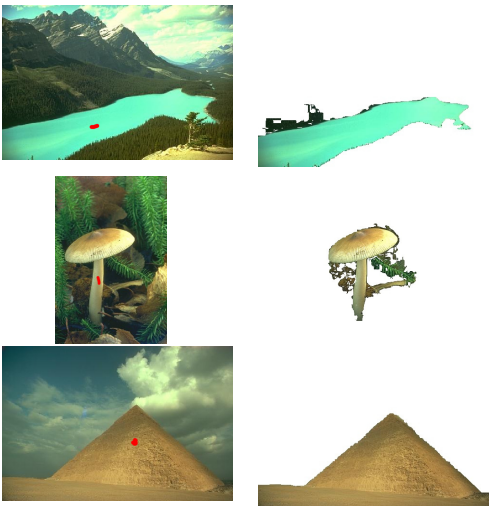

Fig. 11. Original images with chosen seeds (red marks) and Region Growing segmented images. Accuracy rates (compared to ground truth of Figure 7) for the 3 images are, respectively, 93.25%, 86.76%, 86.88%.

		Lake	Plant	Pyramid
FM	ARM	12.86 ± 2.57	8.38 ± 0.72	21.46 ± 2.81
	PC	4.01 ± 0.75	2.70 ± 0.20	7.85 ± 3.12
GC	ARM	7709 ± 1569	7047 ± 924	9004 ± 1201
	PC	2561 ± 1253	2267 ± 464	3105 ± 1253
WS	ARM	47.46 ± 4.99	56.87 ± 10.92	47.89 ± 4.67
	PC	18.74 ± 0.55	15.21 ± 0.75	13.99 ± 0.37
RG	ARM	23.09 ± 3.55	17.15 ± 3.33	15.15 ± 3.05
	PC	4.47 ± 0.22	5.44 ± 1.68	7.10 ± 0.22

TABLE I
PERFORMANCE MEASUREMENT IN BOTH PLATFORMS. ALL VALUES ARE IN MILLISECONDS AND SHOW THE AVERAGE VALUE AND STANDARD DEVIATION FOR 10 RUNS. THE TESTED ALGORITHMS ARE FAST-MOFS (FM), GRABCUT (GC), WATERSHED (WS) AND REGION GROWING (RG).

Speedup				
		Lake	Plant	Pyramid
GC	ARM	599	841	419
	PC	639	840	395
WS	ARM	3.69	6.78	2.23
	PC	4.67	5.63	1.78
RG	ARM	1.79	2.04	0.70
	PC	1.44	2.30	1.44

TABLE II
PROCESSING TIME IMPROVEMENT (SPEEDUP). ALL VALUES ARE RELATIVE TO THE FAST-MOFS AVERAGE EXECUTION TIMES, WHERE A NUMBER GREATER THAN 1 MEANS THAT THE FAST-MOFS IS FASTER THAN THE OTHER ALGORITHM AND A NUMBER LESS THAN 1 MEANS THAT THE FAST-MOFS IS SLOWER THAN THE OTHER ALGORITHM.

but used many less seeds.

Table I shows the performance evaluation executed in the devices. All presented values are averages of 10 measurement samples executed in the target machines. Table II summarizes the execution time improvement factor of the Fast-MOFS segmentation with seeds when compared with other techniques.

From Tables I and II we note that the fuzzy algorithm had a better performance in almost all test cases. In the specific case of the ARM processor, the processing time was several hundreds of times faster than GrabCut, making its use suitable for real time segmentation in mobile devices. Theoretically, the Fast-MOFS algorithm has linear complexity O(n) [11] on the number of spels (pixels, in this case), if the number of edges per spel is a constant.

Note that one of the advantages of the Fast-MOFS approach is its robustness to noise. Figure 12, for example, shows a noisy input image and the segmentation results when using the Fast-MOFS and standard Region Growing approaches. The segmentation results are visually better when the Fast-MOFS is used.

B. Seed tracking

The same x86 and ARM devices already mentioned were used for the seed tracking experiments. The experiments performed in this work were done using the ground truth (MOUSE PAD) proposed by Zimmermann [20] and publicly available, which consists of several videos and their associated ground truths. Four of the frames of the mouse pad ground truth can be seen in Figure 13. This ground truth sequence was originally designed for object tracking experiments, but we use it here to test if our real-time segmentation approach with seeds pointed by the user is feasible and reliable.

Fig. 12. Noisy input image (left) and resulting segmentation with Fast-MOFS (middle) and RG (right).

Fig. 13. Frames from the mouse pad video (first row) and their respective ground truth (second row). Adapted from Zimmermann [20].
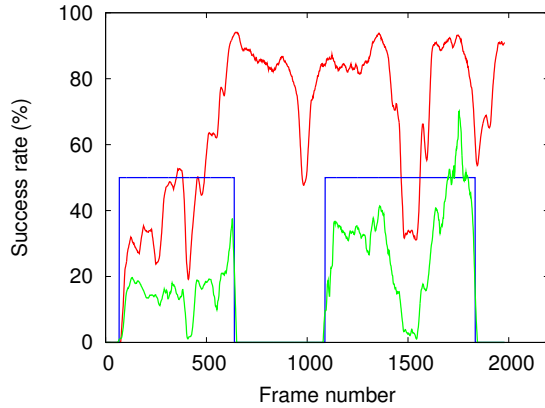

Fig. 14. Foreground segmentation success rate with and without seeds tracking for the first 2000 frames. (Red line: with seeds tracking; green line: without seeds tracking; blue line: indicates if the user is pointing the seeds or not.)

The video was played and users used a mouse to point seeds in the object of interest during the playback. The pointed seeds positions were recorded in a text file to simulate user inputs during the algorithm tests. This file contains a sequence of mouse positions pointed by an user that simulates the projection of the laser in the object. In some frames, the user turns off the seed indication, while in other frames, the user occasionally indicates seeds outside the object of interest randomly. The parameters used were: $t_a = 70$, a Gaussian window of size 13×13, $t_b = 15$, $p_b = 20\%$, $p_l = 8\%$ and $r = 40$.

Figure 14 shows the segmentation accuracy rate of the algorithm execution for 2000 frames of the MOUSE PAD ground truth for the object of interest. The blue line in the plot indicates if the user is pointing to the object or not (laser ON or OFF, for example) while the green line shows the segmentation success rate compared to the ground truth data.

In the segmentations associated with the green line of Figure 14, the only seed is the current point indicated by the user. As expected, the system segments only a small part of the object with a single seed, due to the fact that the image of the target object contains significant gray level variation that cannot be captured by the statistics collected around a single seed. The segmentation is not done when there is no seed.
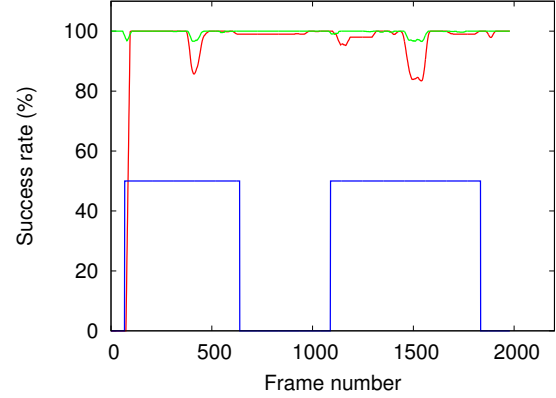

Fig. 15. Background segmentation success rate with and without seeds tracking for the first 2000 frames. (Red line: with seeds tracking; green line: without seeds tracking; blue line: indicates if the user is pointing the seeds or not.)

Note that when the pointing is off (blue line), the segmentation success rate is zero (green line). On the other hand, if the seed tracking is performed (shown in the red line), then the success rate of the foreground is higher because several seeds cover different parts of the object. Moreover, when the user turns off the indication, the segmentation keeps an acceptable accuracy rate even without the user pointing seeds because of our seed tracking system.

Figure 15 shows the segmentation accuracy rate for the image background. The point of this graph is to show that when several seeds are used, some spurious seeds can be indicated in the background by mistake, causing undesired background segmentations. Ideally, the background should be always 100%. Note, however that although there is a decrease of background success rate, it is not that significant, and more importantly, it does not last very long, as the spurious seed is removed from the seed set.

The seed tracking system execution times on the mentioned video with resolution 320x240 were measured for each frame in the 2000 frames sequence. Each measurement consisted of the sum of the seed tracking time plus the segmentation algorithm time. The average measured value for all 2000 frames is 17.40ms for the 1GHz ARM processor (with 6.36ms standard deviation and a worst value of 41.71ms). For the 2.2GHz x86, the average measured value is of 5.27ms with 3.31ms standard deviation and a 34.98ms worst value. In both cases, the system can perform the tracking and segmentation of 30 frames per second.

The next sequence of figures show the stages of the segmentation approach advocated here. Figure 16 shows a seed set from an image acquired with a stone placed on the top of an office table and its respective segmentation.

V. CONCLUSION

We proposed in this paper a novel method for real-time image segmentation using seeds indicated by an user. By using this technique, it is possible to segment objects, and therefore indicate objects in the real world without any computer
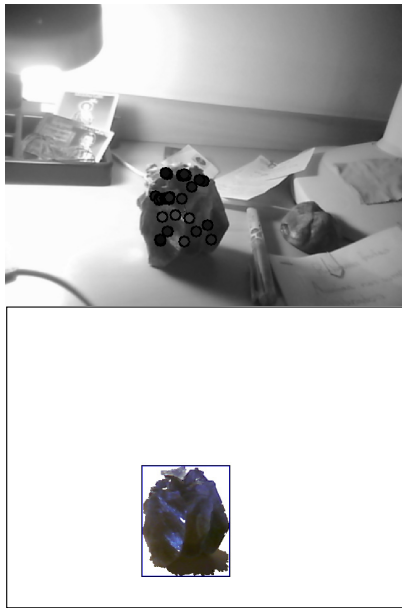
Fig. 16. Optical flow with seeds in different lifetime periods and respective segmentation.

interface. Such approach can be used by several applications in computer vision when an intuitive type of interaction between the user and the computer is needed in real time.

The system works by first detecting seeds pointed by the user using a laser pointer or another pointing system, and tracking them using an optical flow algorithm, so that these seeds can be used even in the segmentation of frames in which they were not selected. This is done in order to provide more information to the segmentation function about the object to be segmented, thus producing more accurate segmentations.

Experiments demonstrated that our system can perform image segmentation in real-time using a conventional PC computer or on an embedded system based on an ARM processor, which is typically used in tablets and smartphones. Moreover, the system also demonstrated its ability to discard outlier seeds that can be mistakenly pointed by the user.

One disadvantage of using the laser to indicate real-world seeds is that the laser can only be detected in a limited range. Furthermore, the color of the laser may be similar to its surrounding region. In such case, a composition of different lasers frequencies can be used in order to distinguish between them, although this was not done in the experiments.

One possible improvement is the usage of feature detectors such as the Speeded Up Robust Features (SURF) [21] to make the tracking system more robust and capable of dealing with occlusions. Such improvement would also work on cases where the segmented object disappears from the camera view and then returns, because the features would be matched and the seeds recovered.

REFERENCES

[1] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images," in *Proc. Eighth IEEE Int'l Conf. on Computer Vision*, vol. 1, 2001, pp. 105 –112 vol.1.

[2] J. Guo and M. Ma, "A low resource consumption image region extraction algorithm for mobile devices," in *2007 IEEE Int'l Conf. on Multimedia and Expo*, july 2007, pp. 336 –339.

[3] J. Kim and H. Jun, "Implementation of image processing and augmented reality programs for smart mobile device," in *6th Int'l Forum on Strategic Technology, 2011*, vol. 2, aug. 2011, pp. 1070 –1073.

[4] F. Zhong, X. Qin, and Q. Peng, "Robust image segmentation against complex color distribution," *The Visual Computer*, vol. 27, pp. 707–716, 2011.

[5] M. Zwinderman, P. Rybski, and G. Kootstra, "A human-assisted approach for a mobile robot to learn 3d object models using active vision," in *RO-MAN, 2010 IEEE*, sept. 2010, pp. 397 –403.

[6] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, February 2010.

[7] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. on Pattern Analysis and Machine Intelligence,*, vol. 16, no. 6, pp. 641 – 647, jun 1994.

[8] G. Friedland, K. Jantz, and R. Rojas, "Siox: simple interactive object extraction in still images," in *Seventh IEEE Int'l Symposium on Multimedia*, dec. 2005, p. 7 pp.

[9] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, Aug. 2004.

[10] G. Herman and B. Carvalho, "Multiseeded segmentation using fuzzy connectedness," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 460 –474, may 2001.

[11] B. M. Carvalho, G. T. Herman, and T. Y. Kong, "Simultaneous fuzzy segmentation of multiple objects," *Disc. Appl. Math.*, vol. 151, no. 1-3, pp. 55–77, 2005.

[12] C.-C. Kang and W.-J. Wang, "Fuzzy based seeded region growing for image segmentation," in *Fuzzy Information Processing Society. NAFIPS 2009. Annual Meeting of the North American*, june 2009, pp. 1 –5.

[13] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in *Int'l Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation*, Sep. 1979.

[14] S.-Y. Chien, Y.-W. Huang, and L.-G. Chen, "Predictive watershed: a fast watershed algorithm for video segmentation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 5, pp. 453 – 461, may 2003.

[15] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the 7th Int'l joint conf. on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[16] C. E. Portugal-Zambrano and J. P. Mena-Chalco, "Robust range finder through a laser pointer and a webcam," in *Electronic Notes in Theoretical Computer Science*, vol. 281, 2011, pp. 143–157, proc. of the 2011 Latin American Conf. in Informatics (CLEI).

[17] A. Rosenfeld, "Fuzzy digital topology," *Inform. and Control*, vol. 40, no. 1, pp. 76–87, 1979.

[18] J. Shi and C. Tomasi, "Good features to track," in *Proc. CVPR '94*, jun 1994, pp. 593 –600.

[19] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[20] K. Zimmermann, J. Matas, and T. Svoboda, "Tracking by an optimal sequence of linear predictors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 677 –692, april 2009.

[21] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in *European Conf. on Computer Vision*. Springer, 2006, pp. 404–417.