A string matching method to guide novice surgeons during training

C. Monserrat[1], A. Lucas[1], J. Hernández-Orallo[2], Mª José Rupérez[1], M. Alcañiz[1]

[1]LabHuman – Ciudad Politécnica de la Innovarión – Universitat Politècnica de València – Valencia – Spain.
[2]Departamento de Sistemas Informáticos y Computación – Universitat Politècnica de València – Valencia – Spain.

CORRESPONDING AUTHOR:
Carlos Monserrat
LabHuman
Universitat Politècnica de València
Ciudad Politécnica de la Innovación - Cubo Azul - Edif. 8B - Acceso N
Camino de Vera s/n, 46022 - Valencia (España)
Tel: +34 96 387 75 18 (Ext. 67017)
Fax: +34 96 387 95 10

e-mail: cmonserrat@labhuman.i3bh.es

A string matching method to guide novice surgeons during training

Abstract.-

Background: Virtual surgery simulators enable surgeons to learn by themselves shortening their learning curves. Virtual Simulators offer an objective evaluation of the surgeon's skills at the end of each training session. The considered evaluation parameters are based on the analysis of the surgeon's movements performed throughout the training session. Currently, this information is usually known by surgeons at the end of the training. However, this information is very limited during the training performance. In this paper, we present a novel method for automatic and interactive evaluation of surgeon's skills that is able to guide inexperienced surgeons during their training performance with surgical simulators.

Methods: The method is based on the assumption that the sequence of gestures carried out by an expert surgeon in the simulator can be translated into a sequence (a character string) which should be reproduced by a novice surgeon during a training session. In this work, a string-matching algorithm has been modified in order to calculate the alignment and distance between the sequences of both expert and novice during the training performance.

Results: The results have shown that it is possible to distinguish between different skill levels at all times during the surgical training.

Conclusions: The main contribution of this paper is a method where the difference between an expert's string and a novice's ongoing string is used to guide inexperienced surgeons. This is possible by indicating novices the gesture corrections to be applied during a surgical training as a continuous expert supervision would do.

1 Introduction

The development and evaluation of medical expertise is a challenging problem in educational and technical terms. One of the key issues for an effective solution is the possibility of defining well-defined tasks that can be repeated and that can produce informative feedback [1][2]. One useful way to understand and improve the acquisition of expertise performance is the notion of *deliberate practice* (DP), "where training (often designed and arranged by their teachers and coaches) is focused on improving particular tasks. DP also involves the provision of immediate feedback, time for problem-solving and evaluation, and opportunities for repeated performance to refine behavior" [3]. According to the "best evidence medical education" (BEME) systematic literature review [4] covering 670 journal articles, the "two most prominent features (for effective learning), ordered by cited frequency, are 1) provision of feedback during learning experiences and 2) repetitive practice."

In the area of surgery, and most especially in endoscopic surgery, the repetition of tasks with immediate feedback had traditionally suffered from two important limitations: the accessibility of realistic test beds and the availability of experts' time to give immediate feedback and guidance during each exercise. The first limitation has been significantly overcome by the introduction of virtual surgery simulators. However, the second limitation has been sometimes neglected or just slightly ameliorated by the development of surgery simulators that are able to evaluate each exercise in an automated way. In fact, one of the main goals in research and development of virtual surgery simulators is to enable inexperienced surgeons to learn *without* a continuous expert supervision. Nonetheless, from an educational point of view, continuous expert supervision (either manual or automated) is crucial for effective learning [2] [4].

Minimally Invasive Surgery is one of the main targets for the (virtual) training of new surgeons. In this kind of surgery, factors such as the difficulty of handling the surgical tools, the limited sense of touch and the need to work in a reduced space make the learning curves of novice surgeons very long. In the early days, the research was focused on creating realistic environments and realistic simulations of the soft tissue behavior. However, nowadays, the research is also focused on developing methods that reduce the learning curves by adding objective measures of surgery skills. One major problem of many surgical simulators that work with these objective measures is that they usually offer the information about the outcome of the surgery after finishing the training session. The potential of these measures to improve the surgeons' skills is limited if the tools become mere surgeon classifiers. Other systems do provide some information, but it is usually goal oriented and restricted to the coordinates, analysis of movements or overall score so far (see related work), but not a detailed and proximate feedback of what went wrong and the sequence of actions that should be done to correct the mistakes. As a result, without a human expert, the exercises are still mostly unsupervised and unassisted.

Alternatively, an interactive and continuous feedback could inform surgeons about their performance in real time, by spotting their mistakes and indicating how to correct them in real time. In this paper, we propose an evaluation method that includes these characteristics, in order to be more focused on surgeon self-learning than towards surgeon classification (as previous surgical simulators). Dubrowski et al. [5] experimentally showed that verbal feedback from an expert instructor during surgical training led to lasting improvements in technical skills performance. Besides, they demonstrated that this is more important in the first stages of the surgeon training. Therefore, the usefulness of an approach that is able to give continuous feedback is justified both form a general point of view of learning psychology and from the specific area of surgeon training.

To this purpose, each surgeon’s *gesture* (a relevant action in terms of position and use of the surgical instruments) has been considered as a *letter* (a character) in a special surgical alphabet. Importantly, the alphabet represents actions, not states. Based on this principle, expert sessions on the simulator are transcribed into a sequences of actions (actually character strings), which represent specific surgical procedures. The goal of the novice is to replicate the expert sequence (or pattern) as faithfully as possible. Any deviation of the novice's sequence from the expert's (reference) sequence is detected and notified to the novice during the training. These notifications can be accompanied by the corrections that the novice must perform in order to return to the reference sequence.

Expert and novice behaviors are then represented by character strings. We propose the use of string-matching algorithms [6] to detect the deviations between novice and skilled surgeons. The basic idea of this new method is that the simulator must be continuously looking for the best alignments between what novices *are* doing and what they *should* be doing. When a (significant) discrepancy is found, the algorithm must suggest the actions (in terms of character subsequences) that could lead to complete the sequence of actions of the whole session with the lowest distance to one of the reference *patterns*. This continuous guidance can provide immediate (real-time) feedback about what the trainee is doing, at the right time and about the right position, also suggesting new actions to get on track if the trainee makes a strong mistake that deviates from the reference patterns.

There are two main issues to solve in order to make this approach feasible. First, an appropriate coding of gestures was needed, by determining an expressive (but non-redundant) *alphabet* as well as a suitable frequency for discretizing the continuous movement of the surgeons according to gesture changes. Therefore, a coding of movements was devised, in terms of the degrees of freedom and the states of the clamp. Second, an adaptation of string-matching algorithms to our setting was necessary because, in our case, the comparison should be between a string (expert sequence) and an ongoing (partial) string (novice sequence). Therefore, some adaptations were introduced to approximate the string matching algorithms to this goal.

This incremental string-matching algorithm is the core of the developed method which allows the interactive evaluation of the novice surgeon during a training session with the surgical simulator. To the authors' knowledge, this is the first time that a method incorporates this kind of scoring feedback in a surgical simulator. This represents a great novelty in the field of surgical training.

2 Related work

Currently, the methods used to evaluate surgeon's skills can be classified into four categories: methods based on weights, methods based on Linear Discriminant Analyses (LDA), methods based on Markov models and methods based on fuzzy classifiers. The Clinical-Based Computer Enhanced Laparoscopic Training System (CELTS) [7] is a representative example of an evaluation method based on weights. It measures the accumulated depth carried out by the surgical instrument, the orientation of the tool, the length of the path followed by the instrument and the time required to finish the training. The system normalizes the obtained values before providing a final mark which represents the novice's skill level. WKS-2RII [8] uses methods that are halfway between the methods based on weights and the methods based on LDA. It measures a set of parameters related to the evaluation of sutures: time required to finish the task, pressure applied on the skin, distance between the suture point and the limit of the wound, distance between sutures and the final wound opening. Finally, it obtains a single value that is a linear combination of the values obtained for each parameter. Then, it applies LDA to classify the surgeons into *expert*, *intermediate* or *novice*.

The systems proposed by Lin et al. [9], Chmarra et al. [10] and Lin et al. [11] could be classified as systems whose methods use LDA as the main technique for obtaining a classification of surgeon's skills. All these systems have carried out a previous training process to obtain the characterization of the surgical skill levels into the three clusters above mentioned: expert, intermediate and novice. They take all the variables obtained in the training process and apply Principal Component Analysis in order to reduce the number of parameters that are considered in the classification. Next, they apply LDA to create the clusters. Lin et al.'s system [11] also applies a Bayes classifier to decompose the surgeon training session into different phases. This last step also takes into account whether the sequence of phases performed by a novel surgeon matches the sequence of phases performed by an expert surgeon.

Among the systems that use Markov Models [8-11]. Rosen et al. [12] establish a similarity between the sequence of gestures (here represented by *states*) carried out during the surgery training and a sequence of words in spoken language. With this consideration, they construct two Hidden Markov Models (HMM), one to represent the state transitions of the expert surgeons during laparoscopic surgery and the other one to represent the state transitions of the novice surgeon, both based on a previous intensive training process of surgeons classified into these groups. Once the two HMMs are constructed, a new surgeon is classified by calculating the distance between the sequence of states performed by the surgeon and the sequence of states from both HMMs. The new surgeon is classified into the nearest group (minimum distance between the new surgeon and the considered group). The systems proposed by Lahanas et al. [13] and Leong et al. [15] are similar to the one proposed by Rosen et al. [12]. The main differences are the states considered for the construction of the HMM. On the other hand, Megali et al. [14] do not consider any previous set of states but the set of surgical states are obtained in a previous step which extracts the main characteristics of the gestures carried out by the surgeons. Then, the system operates in a similar way to the other HMM-based systems.

Finally, the systems proposed by Huang et al. [16] and Hajshirmogammadi and Payandeh [17] use fuzzy classifiers to obtain three clusters of surgeons with the known skill levels: novice, intermediate and expert. These three clusters are built from previous training with surgeons of known skills and using a classifier based on Fuzzy C-Means. Once these clusters are built, the fuzzy classifier returns the skill level of a new user.

Our method is based on a particular coding of actions. This is different to the representation of states used in the approaches based on Markov Processes. Our method relies on the application of string matching algorithms [6] which use metrics known as edit distances [14, 15]. These algorithms obtain a value (metric) that measures the distance between two strings: S_1 and S_2. These metrics measure the number of basic operations (insertion, deletion and substitution) that must be carried out to transform one string S_1 into a second string S_2. One of the most famous metrics is the Levenshtein Distance [19] which gives the same cost value (usually 1) to all basic operations. Another frequently used metric is the Damerau-Levenshtein Distance [20] which considers the transposition operation between adjacent characters in addition to the basic operations. For instance, the Longest Common Subsequence (LCS) algorithm [21] finds the longest sequence of two strings which is common to both by only considering two basic operations: insertion and deletion. The corresponding distance is given by adding the length of the two strings and subtracting the length of the LCS. The LCS algorithm can be considered as a particular case of the Levenshtein Distance, where the operation of substitution has twice the value of the other two. Edit distances have been used in many applications: language processing, planning, robotics, etc. However, to our knowledge, this is the first time that string matching algorithms are used for surgical training and evaluation.

3 Materials and methods

In order to analyze the efficiency and effectiveness of our interactive evaluation method, this has been implemented into a minimally invasive surgical simulator for the training of basic laparoscopic skills. The basic structure of this system is shown in Figure 1. As Figure 1 shows, the surgical trainer developed for analyzing the applicability of string distances in interactive surgical training has these four main components:

- The tracker, which follows the movements of the surgical tools and its actions.
- The virtual scene, where surgeons train the basic surgical skills.
- The string generator, which translates any surgeon's gesture into its corresponding character.
- The distance calculator, which compares the ongoing novice surgeon string with an expert string using a modified string matching algorithm (see section 3.3). This is the core of our interactive evaluation method.

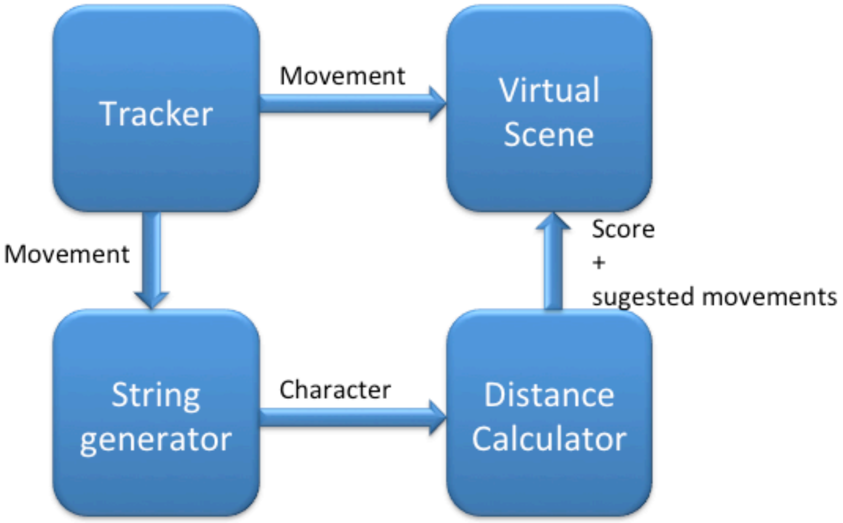

Fig. 1 Global view of the main components of the developed method.

As Figure 1 shows, the Tracker detects the movement and the state of the surgical tool in each time step (in our case, at a frequency of 45 Hz). The tracker sends the movement to the Virtual Scene and to the String Generator. The Virtual Scene translates this movement into a new virtual tool state whilst the String Generator translates the movement into a character (see Table 1) and sends it to the Distance Calculator. Each time that the Distance Calculator receives a character, it is appended to the ongoing novice string and compared with the stored expert string (see section 3.3). The result of the comparison is a Score and a sequence of Suggested Movements. The Score is an indicator of how well the novice surgeon is performing throughout the training and the Suggested Movements represent the movements that the novice surgeon must do to follow the expert indications in order to improve the Score. These are shown to the surgeon on the Virtual Scene. Both Score and Suggested Movements appear in the upper right corner of the Virtual Scene. The processes of String Generator and Distance Calculator are synchronized with the movement translation in the Virtual Scene. This means that all processes are carried out at the frequency of 45 Hz. The following subsections present a detailed description of each component.

3.1 The tracker

The tracker is the component that captures the movements carried out by the novel surgeons during the training. To capture these movements, a video-based optical tracking system was implemented. This tracking system follows the movement of the surgical tools during laparoscopic surgery training. The software library ARToolKit was used to follow the movement of a mark set on the tip of the surgical tool. With this configuration, the tracker is capable of capturing the surgical tool movements with five degrees of freedom: the position of the surgical tool tip in the 3D space (X,Y,Z), the surgical tool orientation (rotation with respect to the tool main axis) and state of the surgical clamp (open or closed).

To get an accurate tracking of the movement, the next tasks were performed:

1. The camera used in the optical tracking was a web-cam (Logitec QuickCam Pro 5000), which was calibrated using the Zhang method [22]. This calibration method uses a set of 2D images of a chessboard of known size to detect and correct the camera aberrations.

2. The mark size used for the optical tracking was fitted to the minimum precision required. The bigger the mark the more accurate the tracking. However, marks that are too big hamper the movement of the surgical tools. Using the Retro-Projection Error algorithm with the calibrated camera, a square mark of 77 mm side was obtained as the optimum mark for obtaining a tracking error below 1 mm for the used web-cam.

3. A threshold for every movement (translation and rotation) was added in order to prevent oscillations as a consequence of inaccuracies of the optical capture system and instabilities due to the mark tracking software. Considering the analysis carried out in steps 1 and 2, a threshold movement of 2 mm for translations and 1 grade for rotations were fixed. Each time one movement exceeded these thresholds, this movement was sent to both the Virtual Scene and the String Generator (see Figure 1).

3.2 The virtual scene

For the evaluation of surgeon's skills, two basic surgical scenarios were implemented (Figure 2). These scenarios allow surgeons to practice the basic skills of laparoscopic surgery in a similar way to other commercial surgical simulators [23]. The goal of the first scenario is that the inexperienced surgeons acquire skills in the use of the camera tool in laparoscopic environments (Figure 2(a)). This scenario requires the use of the 5 degrees of freedom for the surgical tool because surgeons must place the surgical tool at the proper position (center of the sphere) and orientation (indicated by an arrow) and, then, they must clamp. This last action makes the sphere disappear while appearing at another place of the surgical scenario with a different orientation. The goal of the second scenario is that surgeons acquire skills in manipulating objects and eye-hand coordination in a laparoscopic environment (Figure 2(b)). Students must clamp a sphere that appears in the surgical scenario and they must drop the sphere into the specified container.

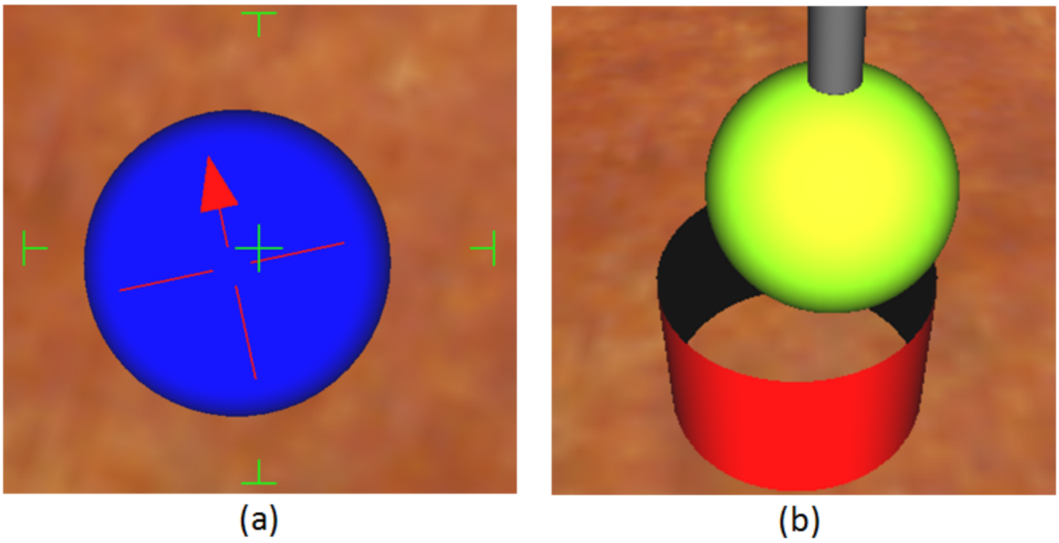

Fig. 2 Screen capture of scenario 1 (a) and screen capture of scenario 2 (b). The image of the scenario 1 shows the localized sphere: the goal is to match the center of the sphere with the center of the image

camera and the sphere arrow pointing up. The image of the scenario 2 shows the container, the laparoscopic tool and the sphere to be dropped into the container.

3.3 String generator

This component is responsible for periodically checking the position, the orientation and the state of the clamp. If the string generator detects a change, it translates it into a character and adds this character at the end of the string which is under construction.

The tracked tool has 5 degrees of freedom: 4 for position and 1 for the clamp. The position changes are represented by 8 possible movements: left, right, up, down, in, out, clockwise rotation, counterclockwise rotation. The 2 possible clamp states are: open clamp and close clamp. Furthermore, the Movement Generator needs to consider that many movements are only correct if they are performed with a given state of the clamp. This leads to 8×2 = 16 symbols. On the other hand, the Movement Generator needs to add two extra symbols for the clamp, Q and P, in order to detect dubious and erratic movements with the clamp, such as QPQPQ. With these two new symbols, there is a total of 16 + 2 = 18 possible characters. The association between movement and its corresponding character is shown in Table 1.

	No clamp		**Something clamped**	
	-	**+**	**-**	**+**
X	L	R	K	S
Y	D	U	E	T
Z	C	F	J	I
Orientation	H	A	G	B
Clamp state	Q (open clamp)		P (closed clamp)	

Table 1.- Relationship between movement (row), direction of the movement and clamp state (columns); and the generated character (inner cells).

Another aspect to be considered is the temporal cost of the surgical training. The evaluation method should distinguish between surgeons that perform five movements in one second and surgeons that perform the same five movements in five seconds. On the other hand, the evaluation method should also distinguish between surgeons that perform fluid movements during the training session and surgeons that perform frequent stops to think about the next gesture. In the former case, the evaluation method should analyze the speed of the novice training and, in the latter case, the evaluation method should analyze the fluidness of the surgeon's movements. For all these reasons, the movement generator also issues two characters directly related to the speed and fluidness of the surgical training:

1. The ***W*** character: ***W*** is generated every second and allows the method to know the movement speed, i.e., the method can distinguish. For instance, both WDDDW and WDWDWDW represent the same movements (negative translation in the Y-axis direction) but the former has been performed faster than the latter.

2. The ***N*** character: ***N*** is generated when a period of time has elapsed (for example, half a second) without the method detecting any movement. It allows the method to know how fluidly the surgeon's movements have been performed, i.e., the method can distinguish between RNRNR and RRR. Both strings represent the same movements but the former has been carried out with several stops and the latter has been carried out fluently. By varying the rate in which N is generated we can control the importance of movement fluency.

All this makes an overall number of 20 different characters depending on the kind of movement (or time events in case of W and N characters) of the laparoscopic tool.

3.4 Distance calculator

The Distance Calculator is based on the LCS algorithm [17]. The reasons of this choice are: the simplicity of the algorithm, the minimum number of basic operations that it considers (necessary to our skill evaluation method) and the facility to adapt this algorithm to a progressive calculation of distances between strings. Furthermore, considering the reduced length of the strings to be compared and the temporal cost associated to the character generation, the time needed for the calculation of string distances and alignments is not a critical parameter for our method. Consequently, the most basic algorithm proposed for LCS by Wagner-Fischer [24] together with a spatial optimization proposed by Hirschberg [25] was implemented.

Table 2 shows an example of the ***R*** matrix construction by the LCS algorithm. At the end of the process, the ***x*** vector contains the string of size *m* generated by the expert (first row of Table 2) and the ***y*** vector contains the string of size *n* generated by the novice surgeon (first column of Table 2). Consequently, $R_{m,n}$ contains the similarity (the LCS length) between strings ***x*** and ***y***. This cell is situated on the bottom-right corner of the matrix represented in this table (the shaded cell in Table 2).

		R	**U**	**C**	**U**	**R**	**R**
	0	0	0	0	0	0	0
R	0	1	1	1	1	1	1
U	0	1	2	2	2	2	2
U	0	1	2	2	3	3	3
R	0	1	2	2	3	4	4
U	0	1	2	2	3	4	4
R	0	1	2	2	3	4	*5*

Table 2.- LCS length (absolute similarity) matrix between strings ***x***= RUCURR (expert) and ***y***= RUURUR (novice).

The Score computation

The LCS algorithm was modified to compare the expert's string (***x***) with the novice's ongoing string (***y***). This new version of LCS algorithm was called on-line LCS or oLCS. The oLCS algorithm applies LCS only to the same number of characters of the two strings. Therefore, the ***R*** matrix of the oLCS is always square. Tables 3 and 4 show an example of the ***R*** matrix construction by the oLCS algorithm. The success level (or relative similarity) of the novice is measured as the ratio between the number of hits, i.e., the length of the oLCS (bottom-right of shaded cells in Table 3 and 4) and the total number of characters introduced by the novice. With this, the ongoing success level is measured as Equation (2) shows:

$$Ongoing\ Success\ Level = \frac{|oLCS|}{\mathrm{Length(y)}} \cdot 100 \qquad (2)$$

		R	U	C	U	R	R
	0	0	0	0	0	0	0
R	0	*1*	1	1	1	1	1
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

Table 3.- oLCS length (absolute similarity) matrix when the novice has only introduced the first character of the string **y**=RUURUR (the **–** symbol represents an empty cell). oLCS only considers the shaded cells for the string comparison. The $|oLCS|$ is the value in the bottom-right corner of the shaded cells, (1 in this case) and the Length(**y**) is also 1. The ongoing success level (relative similarity) of the novice is 100% (1/1) at this moment.

		R	U	C	U	R	R
	0	0	0	0	0	0	0
R	0	1	1	1	1	1	1
U	0	1	2	2	2	2	2
U	0	1	2	2	3	3	3
R	0	1	2	2	*3*	4	4
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-

Table 4.- oLCS length (absolute similarity) matrix when the novice has introduced four characters of the string **y**=RUURUR (the **–** symbol represents an empty cell). oLCS only considers the shaded cells for the string comparison. The $|oLCS|$ is the value at the bottom-right corner of the shaded cells (3 in this case) and the Length(**y**) is 4. The ongoing success level (relative similarity) of the novice is 75% (3/4) at this moment.

The ***R*** matrix in oLCS is square during the evaluation. However, there are two cases for which this could not happen: (1) when the novice finishes too early (***R*** would have fewer rows than columns); and (2) when the novice performs more gestures than the expert (***R*** would have more rows than columns). Both cases are considered as mistakes. In these particular cases, the matrix is kept square using the minimum between the length of the novice string and the length of the expert string. Then, the final level of novice success is measured by Equation 3, where $|LCS|$ is the number of coincidences between the final novice string of length *n* and the final expert string of length *m* (bottom-right corner of the final ***R*** matrix).

$$Success\ Level = \frac{|LCS|}{\max{(m,n)}} \cdot 100 \qquad (3)$$

Therefore, as it can be easily deduced, the oLCS algorithm allows novel surgeons to know, at any time, their level of success through an objective value, the success level.

The computation of Suggested Movements

In addition to the real-time feedback, oLCS obtains the common movements between expert and novice by using a back-tracking algorithm, typically used on LCS to reconstruct the common string. It also keeps trace of the position of the expert sequence where the following gestures would lead to a better alignment, which could be used to suggest the next movements to perform by the novice. Consequently, this algorithm has been adapted for acting as an advanced guide during the surgical training sessions.

The string of Suggested Movements is built from the oLCS coincident and non-coincident movements between expert and novice. The algorithm considers the first *n* characters of the expert string ***sb_E*** and novice strings ***sb_N***. In order to determine the movements the algorithm looks for non-coincident characters. In case of insertions (extra actions), the algorithm tries to compensate them (e.g., actions L and R compensate). In case of deletions (missing actions), the algorithm reminds that this action still has to be done. Figure 3 shows the construction of the string of Suggested Movements applied to the example that appears in Table 4. In this case, the novice string ***sb_N*** has a missing C and an extra R. Both of them are detected by the "NC" (non-coincident) module of the algorithm. Next, since the action R is an insertion (it is extra), the algorithm looks for compensation (an action that can revert the action R). In this case, there is an action that reverts R, which is L. Also, since there is a missing C, this action is also added to the string of suggested movements, which is finally LC.

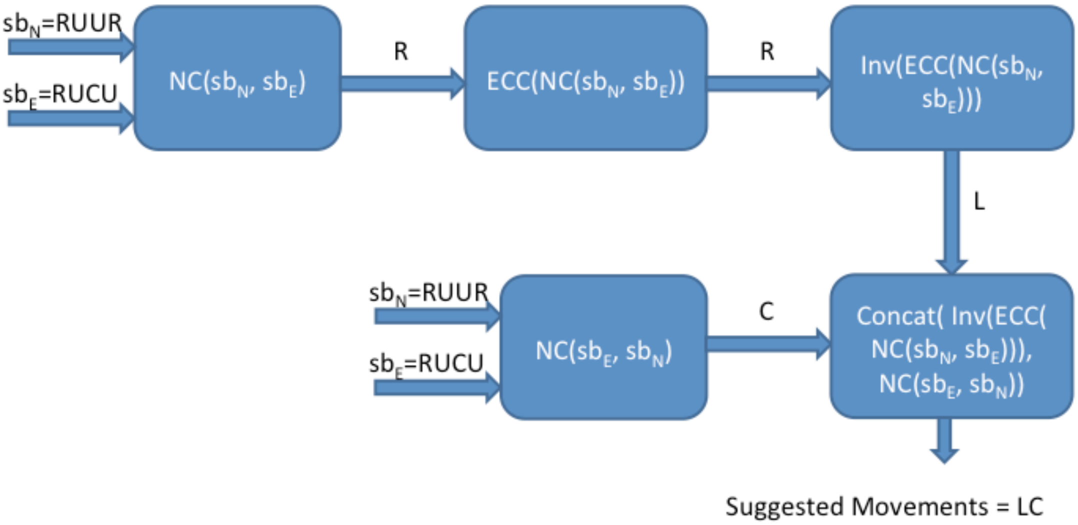

Fig. 3 Building the string of Suggested Movements for the example in Table 4. The final Suggested Movements are LC, which means that the novice must move 2 mm in X (negative direction) and then 2 mm in Z (negative direction) to reach the position of the expert.

3.5 Pilot study design

The goal of this simulator, including the components described above, was to assess the feasibility of our approach, and serve as a reference for future commercial simulators using our method. In order to evaluate this feasibility, we designed a study on our prototype simulator according to this goal.

Consequently, a set of experiments were carried out in order to test whether our approach was able to distinguish between different levels of dexterities (also, indirectly, and in a more subjective way, it was also used to see whether the interactive indications recommended by our application helped surgeons in their learning). For that, the reference expert string corresponding to each scenario shown in Section 3.2 was obtained from ten strings corresponding to ten different experts. The ten experts have demonstrated their knowledge in the handling of laparoscopic instruments and the use of new technologies. The string with lowest mean distance to the rest of strings was used as the reference string.

Once the reference string was obtained for each scenario, 15 students were chosen to perform the experiments. They were divided into three groups according to their surgical knowledge: high, intermediate and novice. All persons included in this study gave informed consent prior the study. The “High” knowledge group consisted of students with some experience in laparoscopic surgery and the use of new technologies. The “Intermediate” knowledge group consisted of students that have little experience in laparoscopic surgery. Finally, the “Novice” group consisted of student without experience in laparoscopic surgery. Each group had 5 students who performed 4 repetitions in each scenario.

4 Results

From the previous experimental setting, 20 samples were obtained for each group, 60 samples for each scenario and 120 samples in total. Figures 4 and 5 show the experimental results. The performed experiments showed differences between the mean level of success for each group and, as expected, with greater levels of success for high level group and lower for novices. Figure 4 shows the evolution of each group in each repetition. Since the repetitions were performed over the same scenarios, the students learned and achieved better results as the number of repetitions increased. Applying the Tukey's Test, significant differences were found between novices and high-level group ($p<0.05$).

Finally, Figure 5 shows the success level for each group per second during the training process.

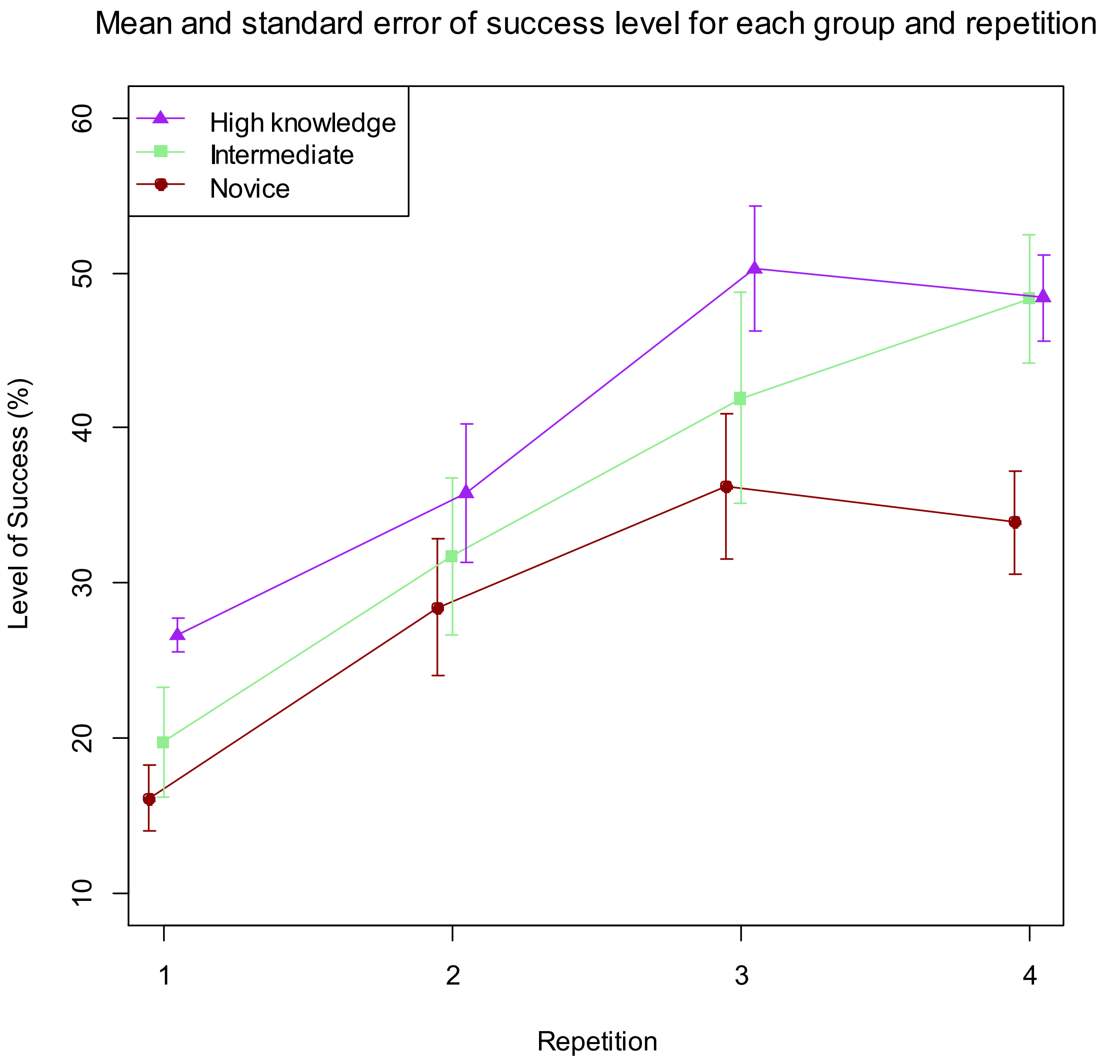

Fig. 4 Mean and standard error of success level for each group and repetition.

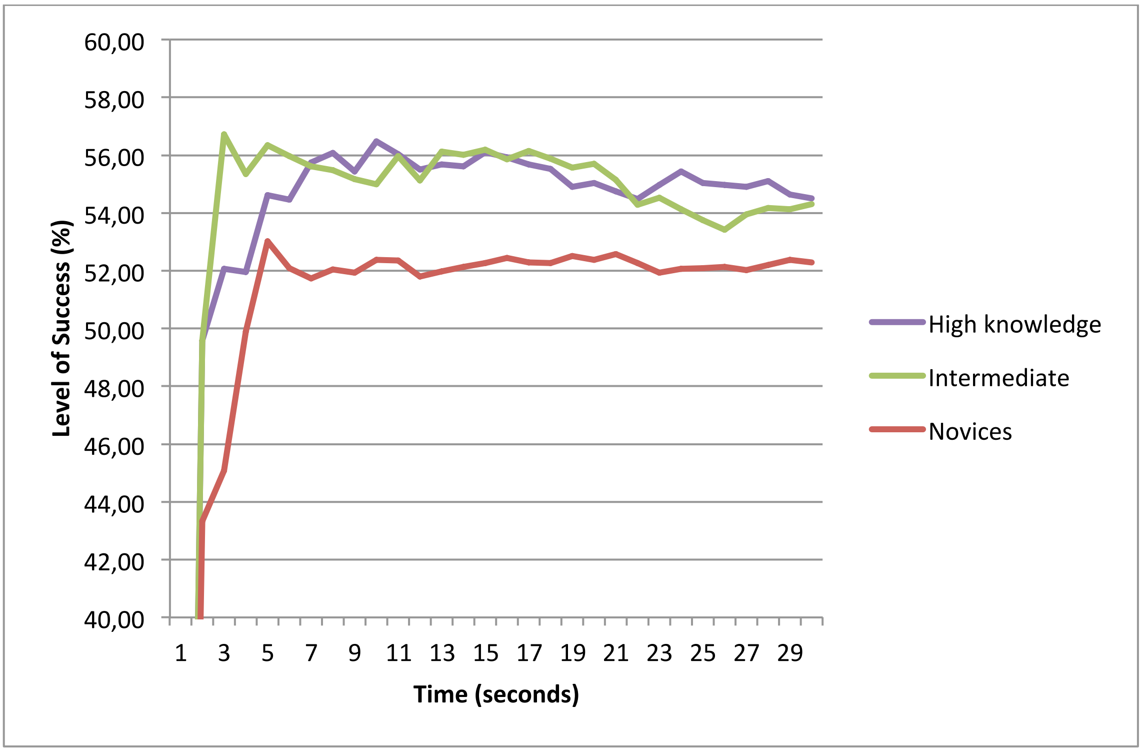

Fig. 5 Real-time success level for each group (each point is the mean of the success level for all the individuals from the group and their four repetitions at a moment during the evaluation session. In order to ease comparison the ten last seconds for each trial are removed as they usually include bogus actions).

In general, these results show that novices are well separated from the rest even when faced upon the same task repeatedly, showing that things like smoothness and speed are important factors that are considered by our string coding. We can also observe in Fig. 5 that the detection of differences can be spotted in less than 10 seconds. This is important, as the evaluation can be interrupted anytime when the purpose is just the evaluation of the trainees, without having to wait until the end of the exercise. This is important when the cost of using the simulator is high.

5 Discussion

The proposed method has been evaluated in two basic scenarios that are commonly used in commercial virtual laparoscopic simulators [23]. From Figure 5, we can conclude that oLCS can distinguish the level of surgical skills during the training session from the beginning.

While the results of this previous pilot study are encouraging, the major advantage of our method over other approaches is precisely the real-time feedback that our method can continuously be giving to the novice, and the detail of this feedback. This has been shown to be crucial by some experimental studies, such as Dubrowski et al. [5] and also by more extensive bibliographic surveys, such as [4].

The experiments show that an early detection between a novice or an expert surgeon is possible. This early detection may be useful to determine the way in which the feedback can be performed on the novice, for instance, by changing the exercise to another (easier) one immediately or even stop the session if the divergence is too high. This may reduce the training time and hence more students can take advantage of the same simulator. It can be noted that the percentage of success varies between

20% and 50%. These values do not have to be taken as an absolute measure of performance, as some dummy characters may reduce the overall coincidence between a trainee and the reference. As a suggestion, these values could be normalized by taking what the experts do on average as a maximum.

Another advantage of our method is its versatility. By properly changing the gesture alphabet and the coding our approach can be adapted to other surgical settings with more or less degrees of freedom, instruments and sensitivity. While the String Generator module should be changed, the Distance Calculator would not need any relevant modification and reused across several simulators.

In addition, the expert sequences used as reference are totally configurable (they can be just recorded and handled from a repository) and can be adapted to different ways of approaching the same problem, without the need of modifying the method. Moreover, as the calculation is not too expensive, we can use the algorithm to find the similarity to several reference strings at the same time, and look for the best alignment for all of them, suggesting the next move for the closest one.

While we have introduced our method for virtual simulators, it could also be applied to non-virtual objects, with an appropriate coding of the alphabet and the reference sequences. Also, we have focused on *novice* surgeon training, but the approach could also be useful to senior surgeons in the context of continued education. In fact, it is recognised (see [1], chapter 8) that "the accuracy of medical doctors' self-assessments is generally poor and does not accurately identify need for improvements in professional performance". Also, some doctors may be reluctant to receive corrections by other (usually younger) colleagues, but may be more positive if the feedback and corrections are performed by a system, which may look more objective to them.

Naturally, our approach has some limitations. For instance, according to our results it cannot distinguish well between the high level group and the intermediate group. One explanation for this fact stems from the selection of the groups. When the individuals were divided into groups, the difference of knowledge among individuals in the high level group and individuals in the intermediate group was not as clear as the difference of knowledge between high level group and novices.

An aspect to be analyzed is the oLCS behavior once the training session finishes. In our solution, any extra or missed character from the novice string was considered a mistake. However, on some situations this could entail an excessive penalty. A possible solution would be to use a cost matrix for the mismatches between gestures, instead of having a constant cost for all of them. This idea has already been considered partially in our implementation, since the temporal characters W and N have a reduced cost. The introduction of costs associated to gestures or to sets of gestures is a very interesting research line which will allow the adaptation of the string analysis to different training scenarios and to the skill level of the users.

Another aspect to analyze is the recognition of gestures and its coding of movements. Our approach is versatile in the way that many coding schemes can be used. However, while determining the right coding is relatively easy in basic virtual laparoscopic scenarios, it could be complex in more advanced scenarios. This is not only a problem in our method but also in all the objective measure systems previously implemented. A possible solution could be to adapt the gesture-identification method proposed in [9] by observing the ones performed by an expert (or a set of experts).

Related to the above issue, we mentioned that, in our case, the reference string was selected as the string that has the minimum mean distance to the rest of expert strings. However, this could not always be the best way. In order to improve this selection, a high number of expert strings should be registered. From this high number of experts, we could construct clusters of common expert strings based on distances, using, e.g., a K-Means clustering algorithm. In this case, the algorithm would consider all the ways to perform the training and the novice ongoing string would be matched with the cluster that would minimize the string distance. However, this approximation requires a high number of expert strings and it goes beyond the scope of this paper.

It should also be highlighted that expert strings extracted directly from only one expert (like the one used in our experiments) could have “bad habits” integrated in the expert string. This must be considered very carefully because the highly targeted training of the presented method could introduce this “bad habits” in the novice training. This would be avoided in an approximation with clusters and K-Means algorithm like the one mentioned before.

This is a novel approach and there are of course many promising things to analyze. As a pilot study, more refined implementations may follow; we could perform more (and more thorough) experiments about the effectiveness of guiding novices during surgical training. These experiments should obtain the learning curve of students who use simulators with automatic guiding during training (like oLCS) and compare with the ones that show the typical performance metrics at the end of the training as mentioned in the introduction.

In our case, both the coding of the tracking into strings and the string-matching algorithm that we introduce in this paper constitute the “method” for which many specific surgery simulator tools can be implemented. Each of them would require a thorough experimental study, with several groups of surgeons (using the new tool, control groups, etc.). The great effort required to implement the simulator and do the experiments can only be justified if some proof-of-concept is performed beforehand when a new methodology or technique is proposed.

Consequently, the goal of the paper has been to present a novel approach (with the main advantage of providing online feedback) such that the assisted-surgery community could assess its potential previously to deciding whether they decide to apply this approach and adapt it to several scenarios.

6 Conclusion

Currently, the commercial surgical simulators are able to perform reasonably good classifications of surgeons among novice, intermediate and expert. However, they do not provide novice surgeons with an interactive and real-time feedback about their performance during surgical training. In this work, an interactive and real-time evaluation method which incorporates this kind of scoring feedback was proposed.

Each surgeon’s gesture sequence was considered as a character string. In order to detect the deviations between novice and skilled surgeons some adaptations were introduced to make the string matching algorithms able to compare a string (expert sequence) and an ongoing (partial) string (novice sequence). This incremental string matching algorithm was called oLCS algorithm and represents the core of the developed method.

To the authors' knowledge, this is the first time that a virtual laparoscopic environment incorporates a real-time evaluation method that produces detailed feedback about what the trainee is doing (wrong) and how this deviates from the reference patterns. This method not only allows surgeon classification but also a guided learning for novice surgeons. The developed simulator is a prototype of an automated expert supervisor which can represent an important improvement for the surgical training through surgical simulators. However, in order to fully confirm the validity of our approach in general, more simulators (and experiments on them) should be carried out in the future.

Disclosure

Dr. Carlos Monserrat, Alejandro Lucas, Dr. José Hernández, Dr. Mª José Rupérez and Dr. Mariano Alcañiz have no conflicts of interest or financial ties to disclosure.

References

[1] K.A. Ericsson (ed.), Development of Professional Expertise: Toward Measurement of Expert Performance and Design of Optimal Learning Environments, Cambridge University Press: New York, 2009.

[2] W. C. McGaghie, “Research Opportunities in Simulation-based Medical Education Using Deliberate Practice” Academic Emergency Medicine, November 2008, Vol. 15, pp. 995-1001, 2008

[3] K. A. Ericsson, “Deliberate Practice and Acquisition of Expert Performance: A General Overview”, Academic Emergency Medicine, Vol. 15, pp. 988-994, November 2008.

[4] Issenberg SB, McGaghie WC, Petrusa ER, et al. “Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review.” Med Teach. 2005; 27:10–28.

[5] M.C. Porte, G. Xeoulis, R.K. Reznick, A. Dubrowski, “Verbal feedback from an expert is more effective than self-accessed feedback about motion efficiency in learning new surgical skills”, The American Journal of Surgery, vol. 193, pp. 105-110, 2007. DOI: 10.1016/j.amjsurg.2006.03.016

[6] P.A.V. Hall and G.R. Dowling, "Approximate string matching," ACM Computing Surveys (CSUR), vol. 18, no. 2, pp. 381-402, 1980. DOI: 10.1145/356827.356830

[7] N. Stylopoulos et al., "CELTS: a clinically-based Computer Enhanced Laparoscopic Training System.," Surgical Endoscopy, vol. 18, no. 5, pp. 782-789, 2004. DOI: 10.3233/978-1-60750-938-7-336

[8] J. Solis et al., "Quantitative assessment of the surgical training methods with the suture/ligature training system WKS-2RII," in IEEE International Conference on Robotics and Automation, 2009. ICRA '09. , Kobe (Japan), 2009, pp. 4219-4224. DOI: 10.1109/ROBOT.2009.5152314

[9] Z. Lin et al., "Objective evaluation of laparoscopic surgical skills using Waseda bioinstrumentation system WB-3," in IEEE International Conference on Robotics and Biomimetics (ROBIO), Tianjin (China), 2010, pp. 247-252. DOI: 10.1109/ROBIO.2010.5723335

[10] M.K. Chmarra, S. Klein, J.C.F. Winter, F.W. Jansen, and J. Dankelman, "Objective classification of residents based on their psychomotor laparoscopic skills," Surgical Endoscopy, vol. 24, no. 5, pp. 1031-1039, 2010. DOI: 10.1007/s00464-009-0721-y

[11] H.C. Lin, I. Shafran, D. Yuh, and G.D. Hager, "Towards automatic skill evaluation: detection and segmentation of robot-assisted surgical motions," Computer Aided Surgery, vol. 11, no. 5, pp. 220-230, 2006. DOI: 10.3109/10929080600989189

[12] J. Rosen, J.D. Brown, L. Chang, M.N. Sinanan, and B. Hannaford, "Generalized approach for modeling minimally invasive surgery as a stochastic process using a Discrete Markov Model,"

IEEE Transactions on Biomedical Engineering, vol. 53, no. 3, pp. 399-413, 2006. DOI: 10.1109/TBME.2005.869771

[13] V. Lahanas, C. Loukas, N. Nikiteas, D. Dimitroulis, and E. Georgiou, "Psychomotor skills assessment in laparoscopic surgery using augmented reality scenarios," in 17th International Conference on Digital Signal Processing (DSP), Corfu (Greece), 2011. DOI: 10.1109/ICDSP.2011.6004893

[14] G. Megali, S. Sinigaglia, O. Tonet, and P. Dario, "Modelling and evaluation of surgical performance using Hidden Markov Models," IEEE Transactions on Biomedical Engineering, vol. 53, no. 10, pp. 1911-1919, 2006. DOI: 10.1109/TBME.2006.881784

[15] J.J. Leong et al., "HMM Assessment of Quality of Movement Trajectory in Laparoscopic Surgery," in International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06), 2006, pp. 752-759. DOI: 10.3109/10929080701730979

[16] J. Huang, S. Payandeh, P. Doris, and I. Hajshirmohammadi, "Fuzzy classification: towards evaluating performance on a surgical simulator," Studies in Health Technology and Informatics, vol. 111, pp. 194-200, 2005.

[17] I. Hajshirmohammadi and S. Payandeh, "Fuzzy set theory for performance evaluation in a surgical simulator," Presence: Teleoperator and Virtual Environments, vol. 16, no. 6, pp. 603-622, 2007. DOI: 10.1162/pres.16.6.603

[18] E. Ukkonen, "Algorithms for approximate string matching," Information and Control, vol. 64, no. 1-3, pp. 100-118, 1985. DOI: 10.1016/S0019-9958(85)80046-2

[19] G. Navarro, "A guided tour to approximate string matching," ACM Computing Surveys (CSUR), vol. 33, no. 1, pp. 31-88, 2001. DOI: 10.1145/375360.375365

[20] F.J. Damerau, "A technique for computer detection and correction of spelling errors," Comunications of the ACM, vol. 7, no. 3, pp. 171-176, 1964. DOI: 10.1145/363958.363994

[21] L. Bergroth, H. Hakonen, and T. Raita, "A Survey of Longest Common Subsequence Algorithms," in Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00), A Coruña (Spain), 2000, p. 39. DOI: 10.1109/SPIRE.2000.878178

[22] Z. Zhang, "A flexible new technique for camera calibration," IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 11, pp. 1330-1334, November 2000. DOI: 10.1109/34.888718

[23] Simbionix^{TM}, Lap Mentor^{TM}, simbionix.com, August 2012 http://simbionix.com/simulators/lap-mentor/library-of-modules/basic-skills/, (31 Gen. 2013).

[24] R.A. Wagner and M.J. Fischer, "Algorithms for approximate string matching," Journal of the ACM, vol. 21, no. 1, pp. 168-173, 1974. DOI: 10.1016/S0019-9958(85)80046-2

[25] D.S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," Communications of the ACM (CACM), vol. 18, no. 6, pp. 341-343, 1975. DOI: 10.1145/360825.360861