Underwater Image Preprocessing for Automated Photogrammetry in High Turbidity Water

An Application on the *Arles-Rhone XIII* Roman Wreck in the Rhodano River, France

Amine Mahiddine, Julien Seinturier, Daniela Peloso
Jean-Marc Boï, Pierre Drap, Djamel Merad
LSIS umr CNRS 6168
Centre National de la Recherche Scientifique
Marseille, France

Luc Long
DRASSM
Département des Recherches Archéologiques Subaquatique et Sous-Marines
Marseille, France

Abstract—**ROV 3D project aims at developing innovative tools which link underwater photogrammetry and acoustic measurements from an active underwater sensor. The results will be 3D high resolution surveys of underwater sites. The new means and methods developed aim at reducing the investigation time in situ, and proposing comprehensive and non-intrusive measurement tools for the studied environment.**

In this paper, we apply a pre-processing pipe line to increase the SIFT and SURF descriptors extraction quality in order to solve the problem of surveying an underwater archaeological wreck in a very high condition of turbidity. We work in the Rhodano river, in south of France on a roman wreck with 20 centimeters visibility. Under these conditions a standard process is not efficient and water turbidity is a real obstacle to feature extraction. Nevertheless the mission was not dedicated to an exhaustive survey of the wreck, but only a test to show and evaluate the feasibility.

The results are positive even if the main problem seems now to be the time processing, indeed the poor visibility increase drastically the number of photographs

Keywords—SIFT Correlation, Underwater photogrammetry, Underater archaeology,Water turbidity

I. Introduction

ROV3D1 project goal is to develop automated proceedings of 3D surveys, dedicated to underwater environment, using both acoustic and optic sensors. The acoustic sensor allows acquiring a great amount of low resolution data, whereas the optic sensor (close range photogrammetry) allows acquiring a low amount of high resolution data. In practice, a 3D acoustic scanner produces a range wide scan of the scene, and an optic system allows a high resolution restitution (larger scale) of different areas in the scene.

The Rhône Archaeological Map excavations carried out by the DRASSM at Arles (Bouches-du-Rhône Department, southern France) over the course of the past 20 years have been seriously hampered by the river's current and poor visibility. In fact, the river water is opaque below six meters, and the visibility, even with good illumination, hardly ever surpasses 20 centimeters. In addition, pollution, the constant passage of canal boats and attacks by large catfish further hinder divers' concentration and the quality of the work.

Despite extremely poor conditions, the study of these river wrecks has allowed us to record very important elements of archaeological heritage and offers an exceptional opportunity to develop new recording methods. This is particularly true of our experiments with photogrammetric and acoustic techniques, always performed in very difficult set-up and recording conditions. In 2011, however, combining these techniques proved to be an interesting alternative for rapid evaluation of a new Roman wreck discovered in the river. This collaboration between archaeologists and photogrammetrists in the Rhône's turbid waters will eventually lead to the development of information-gathering procedures that will speed up measurement, study and interpretation of sites in situ. Even if poor visibility prevents global perception of a site, the acoustic record provides the missing information. Moreover, generating a 3D representation from a high-resolution digital model of complex sites provides archaeologists an overall notion of their configuration. This had never been possible before in the Rhône. Thus, initial trials on the Arles-Rhône 13 wreck serve as a test of these techniques. The encouraging results foreshadow regular collaboration in the Rhône and represent a turning point for fluvial archaeology.

The Arles-Rhône 13 wreck, discovered on the right bank at PK 282.700 in five to six meters of water (between the Arles-Rhône 14 and 17 sites) is of major interest. It represents the first time a seagoing vessel has been found in the river at Arles. All twelve ancient wrecks previously recorded near Arles are flat-bottomed river or fluvio-maritime vessels [11]. However, the hull shape and assembly method of the Arles-Rhône 13 wreck are typical of maritime, rather than fluvial construction. This vessel likely sank during the third or fourth century CE. River action exposed about 10 m2 of the poorly-preserved hull, which is of typical mortise-and-tenon construction. The hull remains lie bottom-up; the keel has been swept away by the current. The typically elegant concave garboard shape of a seagoing vessel is clearly visible on the exposed portion of the hull. Segments of two overturned floors pierced by limber holes are visible within the hull. They are still fastened to the

[1] http://www.rov3D.eu

garboard and adjacent strakes. The goals of our short mission were to map the exposed parts of the vessel, record two transverse sections, take augur sediment samples and wood samples for radiocarbon analysis. However, recording by photogrammetry and acoustic methods represents an important and innovative step. It remains difficult to estimate the vessel's dimensions. Scantlings of the three meters of exposed hull suggest an overall length not less than 18 or 20 meters. The calibrated 14C date falls between 231 cal CE and 381 cal CE. Thus, this vessel sank during the Late Empire, perhaps at a time when the city of Arles, was recovering from third-century invasion damage and reestablishing significant economic activity linked to international trade with the Mediterranean under Emperor Constantine [10].

In 2009, mixed media artist Daniel Zanca had immersed a ravishing ceramic Venus head framed in industrial steel not far from the Arles-Rhône 13 wreck. It was rediscovered, still in place, at the same time as the Roman wreck. Zanca's creation was recorded using the same combination of photogrammetry and acoustic sonar techniques used for the wreck. This yielded a stunning image of the piece lying just 20 meters from the vessel, but separated from it by nearly 2000 years.

One of the most important issues in this work is to obtain an image quality for analysis, measurement, and extracting points of interest to optimize image processing and their orientation for the photogrammetric use.

II. UNDERWATER IMAGE PRE-PROCESSING

The underwater image pre-processing can be addressed from two different points of view: image restoration techniques or image enhancement methods.

Image restoration techniques need some parameters such as attenuation coefficients, scattering coefficients and depth estimation of the object in a scene. For this reason in our works, the preprocessing of underwater image is devoted to image enhancement methods, which do not require a priori knowledge of the environment.

Bazeille et alii [2] proposed an algorithm to enhance underwater images, this algorithm is automatic and requires no parameter adjustment to correct defects such as non-uniform illumination, low contrast and muted colors.

In this algorithm which is based on the enhancement, each disturbance is corrected sequentially. The first step is to remove the moiré effect is not applied, because in our conditions this effect is not visible. Then, a homomorphic filter or frequency is applied to remove the defects of non-uniformity of illumination and to enhance the contrast in the image.

Regarding the acquisition noise, often present in images, they applied a wavelet denoising followed by anisotropic filtering to eliminate unwanted oscillations. To finalize the processing chain, a dynamic expansion is applied to increase contrast, and equalizing the average colors in the image is being implemented to mitigate the dominant color. Figure 1 shows the result of applying the algorithm Bazeille et alii.

To optimize the computation time, all treatments are applied on the component Y in YCbCr space. However the use of homomorphic filter changes the geometry, which will add errors on measures after the 3D reconstruction of the scene, so we decided not to use this algorithm.

(a) (b)

Figure 1. Images before (a) and after (b) the application of the algorithm proposed by Bazeille et alii.

Iqbal et alii have used slide stretching algorithm both on RGB and HIS color models to enhance underwater images [6]. There are three steps in this algorithm (see Figure 2).

First of all, their method performs contrast stretching on RGB and then it converts the result from RGB to HSI color space. Finally, it deals with saturation and intensity stretching. The use of two stretching models helps to equalize the color contrast in the image and also addresses the problem of lighting.

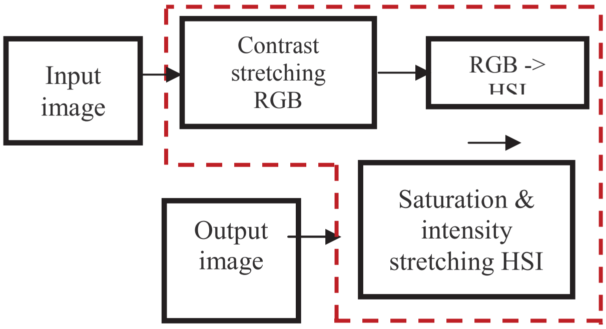
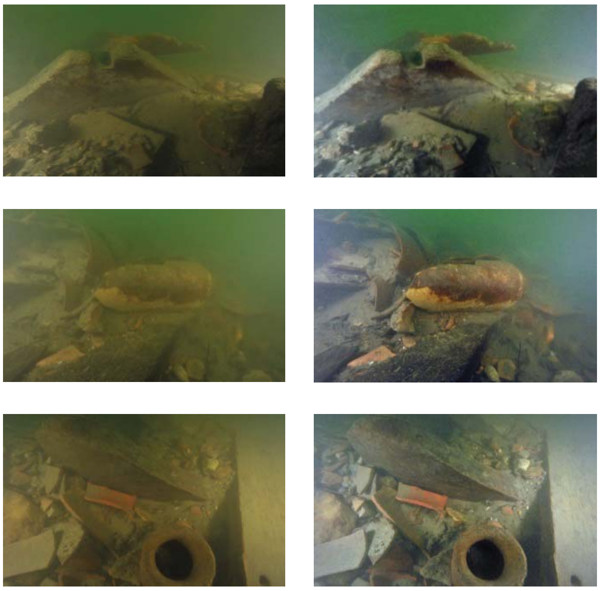


Figure 2. Algorithm proposed by Iqbal et alii [6].

Chambah et alii proposed a method of color correction based on the ACE model [14]. ACE "Automatic Color Equalization" is based on a new calculation approach, which combines the Gray World algorithm with the Patch White algorithm, taking into account the spatial distribution of information color. The ACE is inspired by human visual system, where is able to adapt to highly variable lighting conditions, and extract visual information from the environment [3].

This algorithm consists of two parts. The first one consists in adjusting the chromatic data where the pixels are processed with respect to the content of the image. The second part deals with the restoration and enhancement of colors in the output image [13]. The aim of improving the color is not only for better quality images, but also to see the effects of these methods on the SIFT or SURF in terms of their feature points detection. Three examples of images before and after restoration with ACE are shown in Figure 3.



(a) *(b)*

Figure 3. Phtographs of the wreck Arles-Rhône 13, before (a) and after (b) the enhancement by ACE method. [3].

Kalia et alii [8] investigated the effects of different image pre-processing techniques which can affect or improve the performance of the SURF detector [1]. And they proposed new method named IACE 'Image Adaptive Contrast Enhancement'. They modify this technique of contrast enhancement by adapting it according to the statistics of the image intensity levels.

If Pin is the intensity level of an image, it is possible to calculate the modified intensity level Pout with equation (1).

$$P_{out} = \frac{(P_{in} - c)}{(d - c)} \times (b - a) \qquad (1)$$

where a is the lowest intensity level in the image and equal to 0, b is its corresponding counterpart and equal to 255 and c is the lower threshold intensity level in the original image for which the number of pixels in the image is lower than 4% and d is the upper threshold intensity level for which the number of pixels is cumulatively more than 96%. These thresholds are used to eliminate the effect of outliers, and improve the intrinsic details in the image while keeping the relative contrast. However, Pout values must be in the interval [0, 255], therefore we used the following algorithm:

```
if Pout < 0
    Pout = 0;
else if Pout >255
    Pout  = 255;
end if
```

The results of this algorithm are very interesting. One can observe that the relative performance of IACE method is better than the method proposed by Iqbal et alii in terms of time taken for the complete detection and matching process.

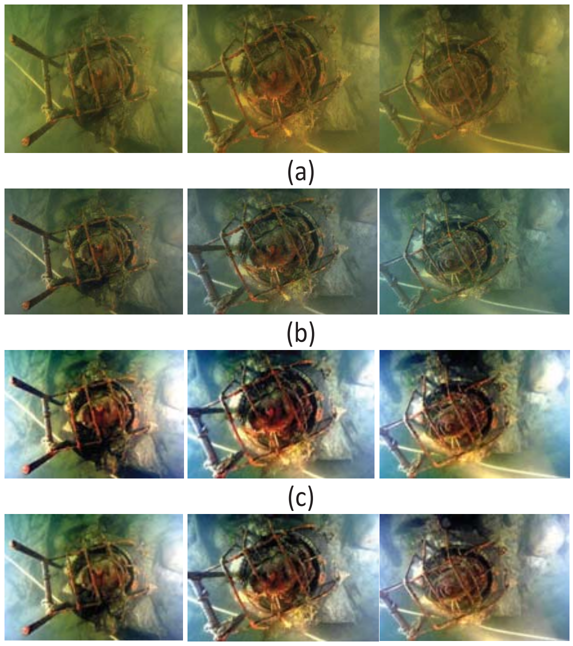


Figure 4. Phtographs of the wreck Arles-Rhône 13, (a) original images, (b) results by ACE method, (c) results by IACE method « Image Adaptative Contrast Enhancement », (d) results by the method *proposed by Iqbal et alii.* [6].

III. Feature extraction and matching

The purpose of preprocessing is improving the quality of images to enhance the detection of interest points. Thereafter, these points of interest will be matched and used for 3D reconstruction of the scene.

There are several methods for extracting interest points such as Edge detector, Corner detector [5]. Juan et alii [7] made a comparison between SIFT, PCA-SIFT and SURF.

In our work, we decided to use two methods most robust in terms of invariance to the transformation and distortion of images: Scale Invariant Feature Transform "SIFT" and Speeded-Up Robust Features "SURF".

Scale-invariant feature transform "SIFT" is a detector and descriptor at the same time proposed by Lowe [12]. it is a method of extracting points of interest that are invariant to changes during image acquisition, these points of interest are local maxima or minima of the difference of Gaussians.

Each point has detected a descriptor vector which is the norm and direction of the gradient in the region around the point of interest.[9]

Speeded-Up Robust Features "SURF" proposed by [1] is a descriptor invariant to change of scale, rotation and image, this method is divided into two parts, the first part is devoted to the detection of points of interest, where in each scaling the local maxima are calculated using the Hessian matrix. From these local maxima, we choose the candidate points that are above a given threshold which will subsequently be invariant to scaling.

The purpose of the second part of this algorithm is to find a descriptor that will make the points detected invariant to rotation, the SURF descriptor is much faster but less robust than SIFT and can therefore be used in applications for real time processing.

After using SIFT and SURF to extract features from images, we implemented three methods for measuring distances, SAD Sum of Absolute Distances, SSD (Sum of Squared Differences) and its normalized version NSSD (Normalise Sum of Squared Differences). Frequently these methods are used to compute the level of dissimilarity between two pixels. In our work we use these methods to compute the distances between each feature (point) obtained with SIFT or SURF descriptor in the first photograph with all points in the second where the best result corresponds to the minimum value obtained after the computation.

We also added the method proposed by Lowe, this method is based on the K-Nearest Neighbour algorithm (KNN) with a modified search using the kd-tree to optimize the calculation time and find corresponding points using Euclidean distance.

IV. EXPERIMENTS

In our experiments, we took a set of 14 images taken by photographer Olivier BIANCHIMANI of VENUS (one of the artworks of international artist Daniel ZANCA), where we reduced the resolution of these photographs to 639 x 425 pixels to reduce computation time (see Figure 4). The choice of this object was to work on an underwater scene where water is very turbid and to test the robustness of SIFT and SURF to extract features.

The implementation was run on an Intel Core i7 CPU 980 at 3.33 GHZ with 12GB of RAM under Windows 7 operating system. We studied the effects of different methods that can affect or improve the performance of repeatability of a descriptor. Initially, we noticed improvements in color quality and we also see that the algorithm proposed by Iqbal et alii gives the best visual results.

Our approach is to detect points of interest on all images using SIFT or SURF descriptors. Subsequently, images are matched two by two with one of methods of distance measurement mentioned previously. For each matched pair of images, the relative orientation is computed using the 5 points algorithm proposed by [15].

From these orientations, an approximate value of orientations and coordinates of object points are calculated. Then a bundle adjustment is applied for optimal estimation of orientation parameters and 3D.

We cannot give the results for all tests because of the space limitations. In the Table 1, we present some results obtained after several tests. This table summarizes the tests performed with SURF and SIFTS descriptors on the original images and preprocessed images, the purpose of these tests as a first step is to find the best preprocessing algorithm in terms of color correction and preprocessing time and which mainly increases the repeatability of descriptors. In a second step, we seek to find the most appropriate method for calculating distances with the type of images that we used in our work which will give more points matched and remove outliers.

We judge the quality of these descriptors according to the number of image pairs oriented, the number of corresponding points and the reprojection error calculated both with the Root Mean Square (RMS) and the average error methods.

The results obtained with images from the three preprocessing methods are better in comparison to results obtained with original images. However, the ACE took an hour and 35 seconds, for the same image IACE took 0.13 seconds and the method proposed by Iqbal et alii took 0.15 seconds almost the same time as the IACE method.

Before choosing the best method of preprocessing to be used in our future work, we started first by the choice of method of measuring distances where it was found that the method used by Lowe which is based on the algorithm KNN performed best in terms of points matches and computation time, otherwise the SSD method and its normalized version NSSD also produce good results in terms of matched points and the number of pairs oriented but requires more time for the computation.

Finally after several tests, we found that the IACE and the method proposed by Iqbal et alii are quite efficient in terms of preprocessing time and number of matched points. However we cannot make a choice between these methods because the results depend on image quality and nature of objects which are located in the scene. In Table 1 we presented the results, where the Iqbal et alii method with SIFT and SUFT descriptors gave the best result. However, IACE method is also effective with other photos of underwater scene.

		SURF	SIFT
ORIG 630x425 pixels	Points homologues	*72*	*344*
	Pairs /91	*7*	*11*
	Time (s)	*32*	*134*
	RMS	*0.04*	*0.01*
	Average error	*3.11*	*0.1*
ACE 1h35	Points homologues	*218*	*701*
	Pairs /91	*14*	*16*
	Time (s)	*75*	*214*
	RMS	*0.03*	*0.007*
	Average error	*0.2*	*0.03*
IACE 0.13s	Points homologues	*363*	*413*
	Pairs /91	*15*	*15*
	Time (s)	*104*	*269*
	RMS	*0.02*	*0.01*
	Average error	*0.24*	*0.1*

IQBAL 0.15s	**Points homologues**	*331*	*671*
	Pairs /91	*14*	*16*
	Time (s)	*96*	*260*
	RMS	*0.02*	*0.009*
	Average error	*0.14*	*0.09*

Table 1 Test on a set of photographs of a scene with amphora and stones.

The photogrammetric approach used here provides results such as 3D points seen on at least three photographs. After the orientation step, we produce dense cloud of 3D-points using patch method proposed by Furukawa and Ponce [4].

The point clouds generated are scaled and geo-referenced on a system of reference and, if necessary, aligned with each other to represent a total object.

It is easy to calculate the scale using a synchronized multi-cameras system (Stereo or Tri-focal system) by knowing the baseline between cameras. However, in high turbidity water, we have to be too close to the measured object and to use of synchronized cameras is not possible. In this mission we used a Nikon D700 camera with spherical housing and 14mm focal length lens. Therefore, to calculate the scale with a single camera, we need scale bars or known object on site.

As we use photographs for their determination and that the photographs are oriented in space we also obtain color information for each point. These point clouds contain no semantics and their density can only help the user to recognize the measured object and eventually, to measure the visible parts (See Figure 5, Figure 7 Figure 8).

We are currently working on merging archaeological knowledge (see Figure 6) to these measured points by automatic clustering and learning methods in framework of ROV3D project.

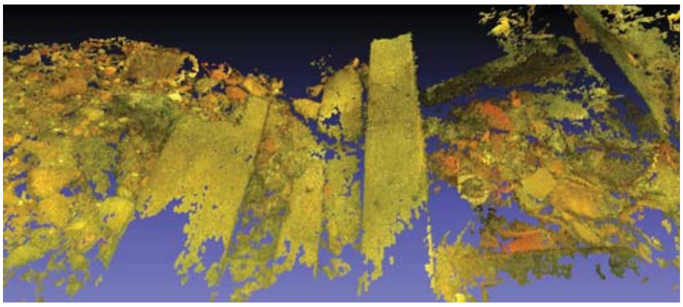


Figure 5 Arles-Rhone 13, Partial of rendered band using 246 photographs oriented in a single block.

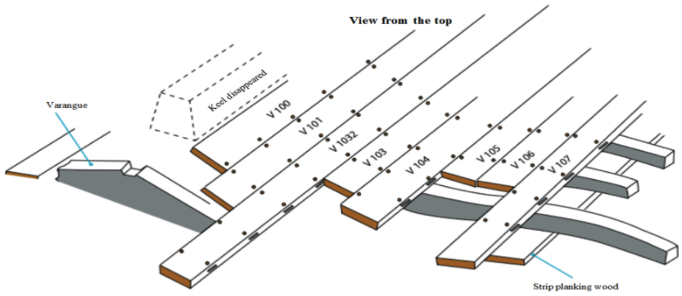


Figure 6 Axonometric plan of the wreck

Figure 7 To the right, one of 253 photographs made on Arles-Rhône XII wreck, and in to the left the corresponding 3D points cloud.

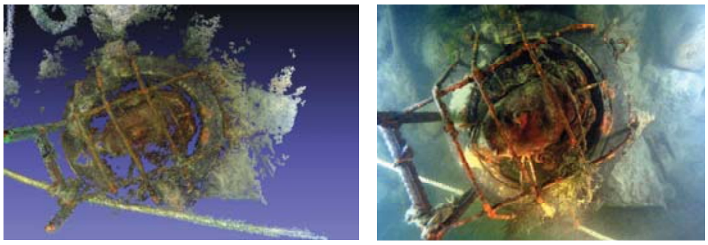


Figure 8 To the right, one of 45 photographs made on the Daniel Zanca immerged statue, close to the wreck, and in to the left the corresponding 3D points cloud.

V. Conclusion & Future Work

In this paper, we studied three preprocessing methods whose purpose was to improve color and contrast of underwater images and increase repeatability of descriptors compared to original images. We have also presented some methods for measuring distances where we found that the IACE method and the method proposed by Iqbal et alii give almost the same results in terms of computation time and repeatability of SIFT and SURF descriptors.

The use of one of these methods as an initial method of preprocessing with the KNN method for distance measurements gives good results in terms of computation time and the reprojection error compared to results obtained with images without preprocessing. Nevertheless, the ACE method is very slow in terms of preprocessing time, however, we observed an improvement of color contrast and a brightness correction. For this reason, we plan to use the images obtained as texture after the full 3D reconstruction of the underwater scene.

Finally we have applied our developments on a real case during the excavation of a roman wreck in Rhodano river, south of France. This approach here is of big interest because of the really poor visibility (less than 20 cm). In effect in such conditions a standard process is not efficient and can't give any results.

Our approach allows us to process a large set of photograph with some very encouraging results. The main problem, in order to satisfy archaeologist needs remains to obtain a complete survey of the wreck which should imply to process around 10000 photographs. Currently we don't know if such a quantity of images (taken with 20 cm visibility) can be oriented together.

REFERENCES

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, & Luc Van Gool. (2008). Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.,* Vol.:110, pp.346-359, isbn/issn:1077-3142.

[2] Stéphane Bazeille, Isabelle Quidu, Luc Jaulin, & Jean-Phillipe Malkasse. (2006, 16 - 19 Octobre 2006). *Automatic Underwater Image Pre-Processing.* Paper presented at the CMM'06 - CARACTERISATION DU MILIEU MARIN

[3] Majed Chambah, Dahbia Semani, Arnaud Renouf, Pierre Courtellemont, & Alessandro Rizzi. (2004). *Underwater Color Constancy : Enhancement of Automatic Live Fish Recognition.* Paper presented at the 16th Annual symposium on electronic imaging,, United States.

[4] Yasutaka Furukawa, & Jean Ponce. (2009). Accurate, Dense, and Robust Multi-View Stereopsis. *Ieee Transactions on Pattern Analysis and Machine Intelligence*.

[5] Chenguang Guo, Xianglong Li, Linfeng Zhong, & Xiang Luo. (2009, 21-22 Nov. 2009). *A Fast and Accurate Corner Detector Based on Harris Algorithm.* Paper presented at the Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on. Vol.:2, pp.49-52.

[6] Kashif Iqbal, Rosalina Abdul Salam, Azam Osman, & Abdullah Zawawi Talib. (2007). Underwater Image Enhancement Using an Integrated Colour Model. *IAENG International Journal of Computer Science,* Vol.:34.

[7] Luo Juan, & Oubong Gwon. (2009). A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP),* Vol.:3, pp.143-152.

[8] Robin Kalia, Keun-Dong Lee, Samir B.V.R., Sung-Kwan Je, & Weon-Geun Oh. (2011, 9-11 Fev.). *An analysis of the effect of different image preprocessing techniques on the performance of SURF: Speeded Up Robust Feature.* Paper presented at the 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV). pp.1-6.

[9] Andrea Lingua, Davide Marenchino, & Francesco Nex. (2009). Performance Analysis of the SIFT Operator for Automatic Feature Extraction and Matching in Photogrammetric Applications. *Sensors,* Vol.:9, pp.3745-3766, isbn/issn:1424-8220.

[10] Long. (2010). Commerce maritime et fluvial à Arles dans l'Antiquité, l'apport des fouilles de Camargue et du Rhône, in L'Archéo Théma. *Revue d'archéologie et d'histoire,* pp.13-18.

[11] Long, & Duperron. (2011). Le mobilier de la fouille de l'épave romaine Arles-Rhône 7. Un navire fluvio-maritime du IIIème siècle de notre ère. *in SFECAG, Actes du congrès d'Arles,* pp.37-56.

[12] David Lowe. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision,* Vol.:60, pp.91-110.

[13] Frédéric Petit. (2010). *Traitement et analyse d'images couleur sous-marines : modèles physiques et représentation quaternionique.* Doctorat, Sciences et Ingénierie pour l'Information, Poitier.

[14] Alessandro Rizzi, & Carlo Gatta. (2004). From Retinex to Automatic Color Equalization: issues in developing a new algorithm for unsupervised color equalization. *Journal of Electronic Imaging,* Vol.:13, pp.75-84.

[15] Henrik Stewenius, C. Engels, & David Nister. (2006). Recent developments on direct relative orientation. *Isprs Journal of Photogrammetry and Remote Sensing,* Vol.:60, pp.284-294, isbn/issn:0924-2716.

Point Clouds Indexing in Real Time Motion Capture

Dario Mazzanti, Victor Zappi, Andrea Brogni and Darwin Caldwell
(Department of) Advanced Robotics,
Istituto Italiano di Tecnologia,
via Morego, 30, 16163 Genova, Italy.
dario.mazzanti@iit.it, victor.zappi@iit.it, andrea.brogni@iit.it, darwin.caldwell@iit.it

***Abstract*—Today's human-computer interaction techniques are often gesture-inspired and thus pushed towards naturalness and immediateness. Their implementation requires non-invasive tracking systems which can work with little or no body attached devices, like wireless optical motion capture. These technologies present a recurrent problem, which is to keep a coherent indexing for the different captured points during real time tracking. The inability to constantly distinguish tracked points limits interaction naturalness and design possibilities. In this paper we present a real time algorithm capable of dealing with points indexing matter. Compared to other solutions, the presented research adds a computed indexing correction to keep coherent indexing throughout the tracking session. The correction is applied automatically by the system, whenever a specific configuration is detected. Our solution works with an arbitrary number of points and it was primarily designed for fingertips tracking. A Virtual Reality application was developed in order to exploit the algorithm functionalities while testing its behavior and effectiveness. The application provides a virtual stereoscopic, user-centric environment in which the user can trigger simple interactions by reaching virtual objects with his/her fingertips.**

I. Introduction

In the last decade, motion capture systems became very common in different fields, from movie industries and Virtual Reality to medical protocols, both for off-line analysis and real time interactions.

Optical tracking of passive markers is often used within Virtual Reality environments to provide non-invasive tracking of the required features. Known, fixed configurations of multiple markers can be unambiguously recognized by this tracking technology, but recognition is not possible for free markers: passive optical markers are not distinguishable one from another. Passively tracked features are therefore lacking any sort of identification: the recognition of a specific marker among a point cloud is not possible. The inability to recognize and distinguish markers especially affects most of the real time applications. The absence of identification for tracked features, and thus the lack of their indexing, makes it impossible to recognize specific points inside non-rigid configurations. Despite being more comfortable for the user, passive markers technologies do not provide reliable tools when dealing with tracked features recognition. This obviously limits real time interaction design.

This work suggests a real time solution suitable for indexing three dimensional point clouds, allowing real time identification of passive markers, together with an indexing correction feature. The development of the solution will be addressed, followed by the description of an interaction-oriented Virtual Reality application relying upon the proposed indexing algorithm. Some of the data recorded during a short experimental session is then provided in order to evaluate the proposed indexing solution. Conclusions and possible applications will be addressed.

II. Background

The need for a real time indexing solution comes from the interest towards natural interaction within Virtual Reality environments. Compared to the technologies available on the market, optical motion capture represents a functional and non-invasive technique for this purpose, but it has some limitations due to the lack of real time distinction between tracked markers.

A. Natural Interaction

Interaction is a core feature for every user-oriented computer application. As interaction strictly depends on the performed action, specialized devices can increase user satisfaction and/or tasks execution speed. Mice and keyboards, for example, are well known devices, mostly used for desktop interaction: they are simple and can cover the majority of the tasks, but they are not always suited to obtain fast and natural interactions. Specific computer applications are diffused in almost any field, from graphics to gaming, from typewriting to projects design and management. Potentially, every application could have a specific device optimally handling its interface and functions. This explains why research continues to push towards novel techniques and devices, and why there is an increasing interest in the application of human real-life interaction and communication modalities into human-computer interaction [17].

Hand and body gestures and motion represent a reflection of human actions in real world, since they normally indicate what individuals are doing, such as pointing or manipulating an object, moving around or communicating. Motion tracking systems allow computers to interpret these gestures and actions through sensory data. Many attempts towards the development of tracking devices resulted in the realization of cumbersome devices, usually carrying many cables and thus implying a reduction of gestures naturalness for the average user [13]. Recent diffusion of non-invasive tracking systems like passive optical motion capture or, more recently, markerless systems is leading to the avoidance of annoying body devices, increasing

978-1-4673-2565-3/12/$31.00 ©2012 IEEE

the swiftness and usability of the devices themselves. Gaming controllers like Nintendo's Wii Remote [12] and Microsoft Kinect [6] are well known for allowing players to trigger real time interactions by moving their hands and bodies, while wearing little or no interaction devices at all. The huge interest raised by the potential of these [5] [7] and other devices confirms the globally shared push to explore human-computer interaction possibilities and applications.

The possibility of using every day body and hand movements to interact with computer applications opens new perspectives on the matter of human-computer interaction [4]. The use of physical and spontaneous actions surely adds new dimensions to the relationship between humans and machines, narrowing the distance between the conceiving of an action and the performing of the action itself. These concepts may provide tools for enhancing all kinds of interfaces, in particular those which are designed to emulate reality, like Virtual Reality environments.

B. Motion Tracking

Human motion analysis is a discipline involving many branches of science and a wide range of applications. *"A Survey of Computer Vision-Based Human Motion Capture"* by Moeslund et al. [11] and the following work [10] provide a comprehensive survey of motion capture approaches, techniques and advances up to 2006.

Virtual Reality is known for its wide use of different motion capture techniques, required by interfaces which are typically too complex to be simply handled with mouse and keyboard [9]. Virtual and Augmented Reality computer-simulated environments use real time motion capture data to implement and improve 3D objects interaction and visualization [15] [3] [18], to enhance subjects experience through perspective correction and to achieve user-centered viewpoint change based on head tracking [2] [16]. Passive sensing is often preferred over active sensing, especially when wire-free interaction oriented setups are required. A perfect example of this approach is optical tracking of passive markers. This technology is well suited for real time interaction within Virtual Reality environments: it is light for the user and no wires are involved. Another interesting technique is computer imaging, which allows 3D markerless tracking by analyzing optical input, even with affordable and diffused devices. Computer imaging systems work better with large motions and they require high computational resources. Therefore extremities like fingers are difficult to be tracked with these techniques. At the present time, this makes optical tracking a convenient, non-invasive and precise solution. On the other hand, by not providing constant identification of free tracked markers, this technique introduces markers indexing problems, which can strongly limit interaction design.

Real time indexing of moving features is an already addressed issue [1] [8] which likely benefits from application specific approaches. More specifically, existing literature says little about passive markers indexing matter, being that markers indexing is more precisely done by using active markers, like the A.R.T. Fingertracking device[1] or different non-optical solutions do. The paper titled *"Human Movement Tracking and Analysis With Kalman Filtering and Global Optimization Techniques"* [14] inspired this project's first steps towards passive markers indexing. The authors approached the problem of tracking human movement along image sequences by using a Kalman filter to perform estimation and correction of feature points' movement. The main criteria was to establish the best global correspondences between the group of estimate points in each image and the new points positioning data. This approach has been tested with positive results on feature points fixed upon the legs of a filmed walking person. The solution presented by the authors also considered occluded markers managing and recognition.

III. Point Clouds Indexing

The realization of a functional indexing solution required the design and development of two instructions sets, capable of achieving real time indexing and managing indexing correction for optically tracked markers. Also, many tests were done at different stages of development to evaluate the tracking system and to test the indexing logic behavior.

A. Motivation

Optical tracking of passive markers has many advantages over other techniques, such as being light-weighted and allowing larger capture areas, but tracked markers references may swap during real time tracking. These errors can be found and corrected off-line at the end of the tracking sessions, but applications making real time use of captured data obviously do not allow this approach. Real time optical tracking is widely used in the Virtual Reality field for interaction purposes and it has a potential in natural interaction achievement. Anyway, the difficulty to distinguish each marker strongly limits the possibilities of interaction design for optical tracking systems. This may lead to the use of rigid bodies or other cumbersome devices which surely can help in keeping constant references to each tracked marker, but also add complexity to the needed setup, thus keeping a distance from the wanted natural and comfortable setting.

B. Objective

The main goal of this research was to design and develop an algorithm capable of maintaining correct real time indexing of free markers using every tracked point position as input, and testing its benefits on real time interaction design. The algorithm is made of two instructions sets: the first one maintains markers indexing, and the second one corrects that indexing by using session-specific constraints as a reference.

C. Preliminary Tests

A preliminary study was necessary in order to observe raw indexing behavior and related errors. An Optitrack FLEX:100 system composed by 12 infrared cameras has been used to perform dislocated multipoint optical motion capture. It is

[1] http://www.ar-tracking.de

possible to track hands and fingers using small, light-weighted passive markers. Multiple tracking sessions were recorded and analyzed using a simple application to retrieve and visualize data. A straight bar working as a support for three markers was moved inside the tracking volume. Movements included the ones that may be responsible for indexing errors, such as covering and uncovering markers, or moving markers while some other ones were covered.

Following an example of a sequence of movements known to always generate errors: we have three distinctly separate markers A, B, C, respectively numbered by the system as 1, 2, 3, as showed in Fig. 1

Fig. 1. A (red), B (green), C (blue), respectively numbered as 1, 2, 3.

After hiding marker A, the resulting indexing will be B = 1, C = 2: this because the first appearing marker (A = 1) is no longer visible. The results can be seen in Fig. 2. If we uncover marker A the resulting configuration will be indexed as A = 3, B = 1, C = 2 (Fig. 3).

Fig. 2. Markers B and C with their new, wrong labeling.

Fig. 3. Markers A, B, C. Wrong indexing.

Indexing errors can also be due to the possibility for two markers to touch each other: when that happens, the system detects a single marker. When the markers are separated again, the indexing of the two may be swapped. This specific behavior appeared since the beginning as a delicate matter, likely of difficult solution.

After these observations we were able to notice that the two main indexing errors are due to:

- markers occlusion, even if happening for short time intervals
- markers touching each other

The first point includes different sub-causes, which will be addressed later.

D. Algorithm Development

While the approach to occluded markers seen in Sec. II [14] inspired our indexing algorithm, the Kalman filter solution described in the article appeared to be too specific. Estimates computed through the filter may give good results only when linear movements occur. These constraints were too strong and they badly described usual motion capture characteristics.

A first version of the algorithm included a Kalman filter feature which was used to estimate occluded markers movements. We verified that the filter was badly influencing indexing behavior, especially when nonlinear and sudden movements occurred. Movements of these kinds are a typical feature of natural interaction contexts. Being so specific and far from the purposes of the project, Kalman filtering was excluded from the presented solution.

Using the mentioned [14] indexing and occluded markers approach as a starting point and ignoring Kalman estimates, we started following our main goal: realizing an algorithm capable of keeping track of every single marker position and to correctly manage markers occlusion.

1) Main Indexing Algorithm: The main algorithm is designed to index any specified number of points. This is done by storing and keeping markers univocal reference and their visibility status, the latter indicating if a marker is currently visible or occluded. The algorithm receives and elaborates markers coordinates from the available tracking system. The algorithm consists of a series of instructions which periodically elaborate each marker position. It's structure is made of two states: an initialization state and an indexing state (Fig. 1).

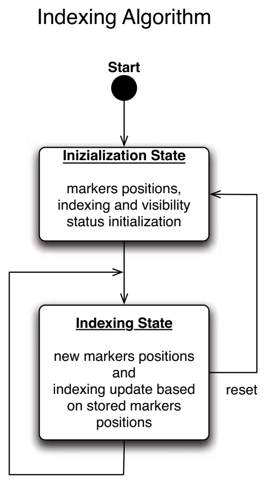


Fig. 4. Indexing algorithm finite state machine

Initialization state instantiates the variables needed to store and index the data of a specified number of markers. Once initial setup is done, the algorithm cyclically manages incoming markers positions data in order to maintain the most coherent indexing. This is done by associating new markers positions to previously recorded positions. The criteria used to match an incoming point position with a stored one is based on euclidean distances. Every stored marker is associated to the nearest marker selected among the current tracking caption. In order to find the correspondence for a single marker, the algorithm computes the distances between the marker position and each new captioned position. A customizable threshold is included. The computed distances are then compared between them: the marker picked by the procedure is the one which generates the minimum value.

The matching procedure is done using previously stored markers positions, which are updated caption after caption:

- picking a point A among the available stored ones (starting with the first one in the stored positions data structure)
- computing euclidean distances between A and each available position acquired from tracking system
- choosing a point A_{new} as the one at the smallest distance from point A
- A and A_{new} are now matched and will be unavailable for next searches during current matching procedure
- if there is no A_{new} suitable as a match, point A will be considered not visible until next caption
- matching procedure is repeated until all stored points have been updated

This is done to find the new current position for every stored marker.

The threshold added to computed distances, which was previously mentioned, defines a volume surrounding each stored point. When looking for potential new positions for a stored marker, only the ones included in this volume are considered (Fig. 5). Among these, the position nearest to the center (A, the stored position) is considered as the new, current position.

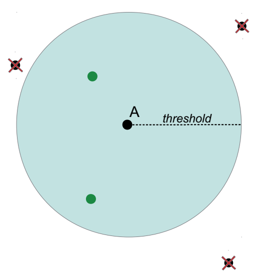


Fig. 5. Indexing lookup volume. Green dots represent coordinates which will be evaluated as possible new positions for point A.

As all points have been considered, the system updates itself overwriting old positions with new ones and updating visibility status variables.

The possible presence of occluded markers was of course a pending problem. The value of the mentioned status variable indicates if a specific marker is currently visible or occluded. Whenever a stored point has no correspondence among the new incoming markers data, its status is set as "not visible" and its new position is kept unchanged. This of course diminishes the accuracy of the following correspondence evaluations for that specific marker. When a marker set as "not visible" matches with an input position it will be set as "visible" again. Its position will be the same as the matching input marker. The algorithm priority is to find correspondence for all markers classified as visible. Only after that the focuses moves on occluded markers correspondences, since their position is less reliable. If a single occluded marker is left to be associated to a new one, and there is only one new position left to assign, every distance evaluation is skipped, and the two points are given an automatic match. If new positions are still available after completing all matches, they are likely to be ghost markers or reflections positions, which are not going to be associated to any indexed marker.

2) Algorithm Tolerance Threshold Observations: As explained, the algorithm is completely based upon each marker's last recorded position, even when markers occlusion occurs. A large number of real time and off line tests conducted with different markers number and configurations proved that the algorithm behavior is strongly influenced by the chosen tolerance threshold. The higher the tolerance value, the bigger is the volume of correspondence research considered around each incoming point. A low threshold value leads to a more precise correspondence between old recorded markers positions and new markers positions. This helps the algorithm to narrow its searches, but suffers from rapid markers movements: a restricted correspondence research volume could make the algorithm ignore correct indexing correspondences. A higher tolerance allows to follow rapid movements more precisely, but the enlargement of the research volume may lead to indexing errors, especially for those markers which happen to be at close range, as explained in the following section.

After these considerations, it is reasonable to say that an application and system specific threshold setting my be needed to adapt the algorithm to the available system tracking resolution and acquisition rate, and to match the movements expected by the application.

3) Algorithm Limitations: At this stage, the algorithm left some open issues related to a number of possible behaviors.

Coexistence of rapid movements with close ranged markers setups is difficult: two or more rapidly moving markers, at close range, may be swapped. That is because the algorithm considers minimum euclidean distances between stored and new markers positions. Incoming new positions received during quick motions may be misleading.

Hidden markers reappearing far from their disappearing position can not be recognized as the same marker. Correct indexing under those circumstances only works when only a marker is occluded and then reappears, even at a completely different position.

Also, two different markers disappearing and reappearing at similar positions may result to be swapped. This usually happens when two markers touch each other or when two close markers are covered simultaneously.

4) Secondary Correction Algorithm: As seen in the previous paragraph, the indexing algorithm still suffers from some limitations due to the variety of possible tracking situations which could lead the algorithm to false detections. Some of these behaviors are non predictable and thus cannot be avoided. The main purpose of this work was to find a reliable real time indexing solution. Once an indexing error occurs in the main algorithm, there is no control over it, no correction is possible. A useful set of instructions would be one that could, with a certain occurrence, reset markers to their correct indexing order while being completely transparent to the main algorithm.

Starting from these purposes, a second support algorithm was designed with the objective to add an indexing correction feature to the main algorithm.

One of the main fields we thought the algorithm for was the interaction with virtual objects in immersive environments. For a natural interaction, we wanted to track the fingertips of the user's hand and trigger interactions considering their positions. Imagine a moving human hand with trackable markers fixed upon its fingertips. In that situation markers are forced to move in a peculiar manner and during a tracking session it is likely to capture typical, recurring configurations. One of these recurring configurations may be stored during a session-specific calibration phase, and considered as a known indexing configuration. This is what the secondary correction algorithm does: a calibration posture is recorded at the beginning of the tracking session and used as a reference. Whenever markers reach this known configuration the system forces an overwriting, meaning that the current indexing will be changed to match the calibration one. So, if for any reason the indexing happens to be wrong it can be corrected by reaching the calibration configuration. This could also happen involuntarily if the calibration posture is a recurring and comfortable posture for the user.

This secondary correction algorithm is also represented by a two-state finite state machine, made of a calibration and a recognition state.

5) Calibration and Recognition Phases: A calibration phase was introduced at the beginning of the indexing procedure (Fig. 6). During this state markers must be set to the desired "reset configuration". This phase lasts for a customizable amount of time to allow the user to specify a desired pose. This permits to store user specific constraints which represent the chosen reset configuration. When the calibration state ends a recognition phase begins. During this state the algorithm continuously observes indexed markers configuration. Current tracked points inter-distances are computed and compared to the ones stored during the calibration. When the difference between the two is under a certain threshold, the current pose is recognized as a reset configuration. The threshold has been introduced to lower the user accuracy needed to reach this configuration. Whenever the reset configuration occurs, the algorithm forces current tracked points indexing to match the calibration one. The characteristics chosen to memorize (and identify) the configuration are, as mentioned, the distances between each marker. Their order represents the indexing reset condition and the indexing itself. That indexing is to be considered the correct one. Naturally, it is possible to store a chosen calibration for future sessions.

Adding an indexing correction feature improved indexing correctness and reliability over time.

In addition to the described generic recognition procedure, other algorithms were developed to test specific markers numbers and configurations (i.e. pinch recognition, double pinch recognition and hand recognition). These algorithms share the same structure of the generic one: they need a calibration at the beginning of the tracking session, to compare input markers positions with the recorded configuration and trigger indexing correction. The advantage of custom correction solutions is that they can add application specific constraints designed to help calibration posture recognition, while being completely transparent to the main indexing instructions.

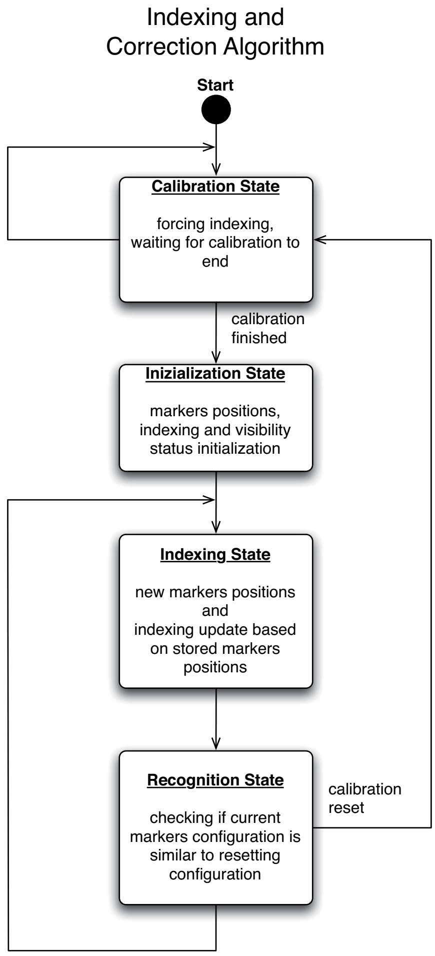


Fig. 6. Indexing and correction algorithm finite state machine

Fig. 7 shows a simple application of the generic correction feature: ten markers are placed on a wire, tracked and indexed as a whole cloud. The calibration configuration is recorded while keeping the wire straightened. The user can freely manipulate the wire and move it in space. As a consequence, markers are likely to often touch each other and move while hidden from the cameras, thus introducing errors. Every time the wire is straightened, the correction algorithm recognizes the calibration configuration and corrects the errors which may have occurred.

IV. Evaluation Study

No objective data on the algorithm behavior and efficacy was available, so we conducted a short session of experiments, as described in this section.

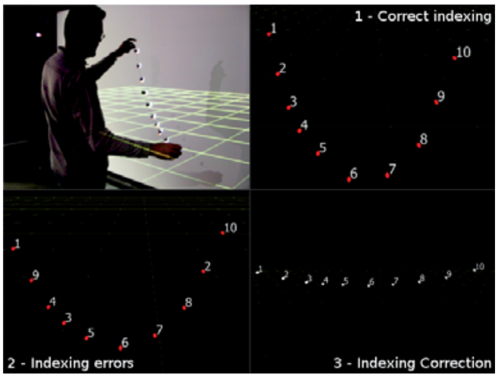


Fig. 7. Indexing of ten markers on a wire.

A. *Objective*

The main objective of this evaluation study was to test the algorithm inside a real indexing-based application while recording data on its behavior. This also required an application capable of detecting indexing errors by comparing the algorithm indexing with a reliable indexing.

B. *Materials*

The available hardware and software setup consists of a 4x2 m^2 powerwall, projected by two Christie Mirage S+ 4000 projectors, synchronized with StereoGraphics CrystalEyes active shutter glasses. The main software development platform is VRMedia XVR[2], used to handle graphics, scene behavior and input/output data sending. Devices and software exchange data with the main application is made possible through XVR internal modules, written in C++. XVR natively supports tracking systems: the hardware includes also an Intersense IS-900 inertial-ultrasonic motion tracking system used to track user's head position. Errors detection needed a genuine indexing to be compared to the one provided by the algorithm. A CyberGlove device was used: thanks to a resistive bend-sensing technology it transforms hand and finger motions into real-time joint-angle data. Given those angles and using a tracked rigid body placed over the glove, we were able to reconstruct hand-related fingertips positions by means of inverse kinematics. These positions were compared with glove-related markers positions in order to detect and record indexing errors.

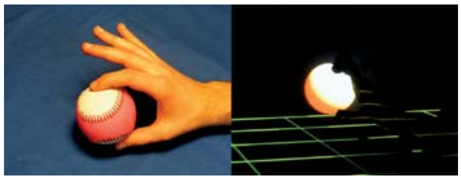
C. *Experiment*

An indexing-based Virtual Reality environment was created and provided with an indexing error detection feature. The application recorded indexing correctness data while users executed interaction tasks. The algorithm is capable of indexing an arbitrary number of markers. The interactive environment was designed to trigger interaction by using the positions of three markers fixed upon thumb, index and middle fingers, as in Fig. 8.

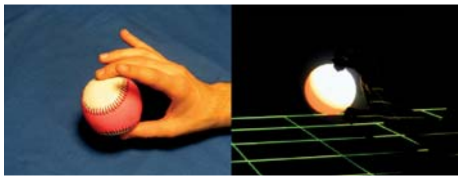
[2]http://www.vrmedia.it/Xvr.htm

Fig. 8. The simple markers setup used within the interactive environment.

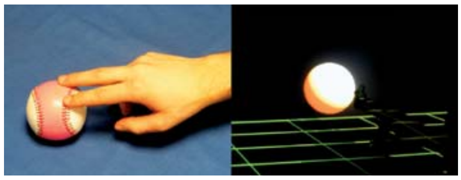
1) Experiment Design: The subjects experienced a three dimensional immersive environment. Subject's head position inside the room was tracked. Within the virtual environment a table, a shelf and a frame were shown (Fig. 10). The table was the nearest object to reach for the subjects. A small number of simple procedural objects were placed over the shelf. Objects section varied from 9 to 10 centimeters. In the background, a frame was showing combinations of procedural objects. During the experiment the subjects had to recreate these combinations over the table. That was possible by copying, moving and deleting a number of provided objects. These interactions took place whenever the subject "touched" with his/her fingers one of the virtual objects:



(a) To move an object within the virtual environment the user pinches it with the index and thumb fingers.



(b) A movable duplicate of an object can be created by touching it with middle and thumb fingers.



(c) Index and middle fingers both touch an object to delete it.

Fig. 9. Possible object interactions.

- By pinching an object with the thumb and the index fingers, it could be moved (Fig. 9(a)).
- By doing it with the thumb and the middle finger a duplicate of the object was created, and then moved to be placed anywhere (Fig. 9(b)).

- By inserting the index and middle finger into an object it was deleted (Fig. 9(c)).

The four shapes over the shelf were not movable or erasable: they could only be duplicated.

2) Procedure: The subjects, all right handed, were asked to wear the CyberGlove and to follow the calibration procedure needed by the device. Then, three passive markers were placed in proximity of the glove thumb, index and middle finger fingertips. The subject could wear the shutter goggles. The object interaction methods and the task required were explained so that it was clear to the subject how to interact with the virtual objects. The subject was then required to keep the three tracked fingers in front of him/her for the correction algorithm calibration. The fingers had to be detached in a comfortable way in order to achieve a functional calibration. The subject was trained to open the hand in the proper way during the experiment whenever he/she thought the interaction with the objects was not triggering the expected behavior. The subject was then asked to reproduce five objects combinations shown inside the frame, starting from the four objects placed over the shelf. The combinations must be reproduced over the table in front of the subject (Fig. 10). No feedback was provided during the experiments to signal if or when correction was happening.

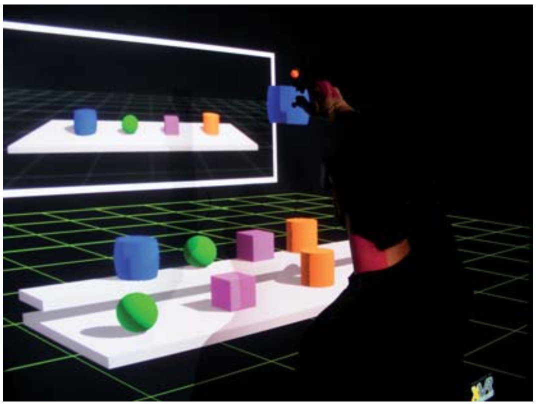


Fig. 10. A shot showing a subject experiment. Little red spheres represented the user fingertips to help interaction. Picture taken while tracking the camera position instead of the user's head.

D. Analysis and Results

1) Variables: Data recorded during the experiments includes each marker correctness and visibility status, global indexing correctness and calibration posture recognition. From these recorded values the following variables are extracted for each experiment:

- the number of indexing errors occurrences
- the number of markers occlusions occurrences
- the number of posture recognitions
- average time between each indexing error
- average time between each marker occlusion
- average indexing correction time value
- correction time standard deviation

N.	n_e	n_r	n_o	$\frac{T}{n_e}$ [s]	$\frac{T}{n_o}$ [s]	$\overline{T_c}$ [ms]	σ_{T_c}	T_{tot} [s]
1	17	103	50	16.445	5.591	13.82	7.67	279.574
2	7	7	15	32.048	14.956	12.71	7.74	224.338
3	13	71	123	17.897	1.892	17.69	5.78	232.668

TABLE I
THREE SESSIONS EXPERIMENTAL DATA (n_e: INDEXING ERRORS OCCURRENCES n_o: MARKERS OCCLUSIONS OCCURRENCES, n_r: POSTURE RECOGNITION OCCURRENCES, $\frac{T}{n_e}$: AVERAGE TIME BETWEEN INDEXING ERRORS, $\frac{T}{n_o}$: AVERAGE TIME BETWEEN MARKERS OCCLUSIONS, $\overline{T_c}$: CORRECTION TIME AVERAGE VALUE, σ_{T_c}: CORRECTION TIME VALUES STANDARD DEVIATION, T_{tot}: SESSION TOTAL DURATION).

- experiment duration

Occlusions and indexing errors were automatically recorded by the application. This was done by comparing the indexing algorithm data with the data provided by the CyberGlove application.

2) Analysis: All subjects successfully completed required tasks. Some of them used the correction feature better than others, while some found certain moments of the experience a bit frustrating. This occurred especially when they were not able to immediately correct indexing, thus compromising interaction with the objects. When this occurred, a recalibration was proposed to allow the subject to find a more comfortable indexing correction posture.

Tab. I shows data computed from three experiments data logs. For each experiment the table shows indexing errors and markers occlusions occurrences (n_e, n_o), and the number of posture recognitions (n_r). $\frac{T}{n_e}$ and $\frac{T}{n_o}$ respectively represent average time between each indexing error and markers occlusion within the same session. Indexing correction average time and standard deviation values are displayed ($\overline{T_c}$, σ_{T_c}). The field labeled as T_{tot} indicates the total duration for each session.

3) Results: Posture recognition occurrence rate varies from subject to subject due to personal response to the recognition feature. After watching different sessions it also seems strongly dependent on the initial calibration posture. By analyzing the second subject's data it can be seen that recognition only occurred when it was strictly necessary: in other words, it probably occurred only after each indexing error, voluntarily triggered. During the other two subjects' experiments recognition occurred way more frequently: a sign that the correction posture chosen by the subjects was a more recurring and maybe natural pose for their fingers.

Even if in different proportions, the number of occlusions have always been greater than the number of indexing errors. This tells us that an approximate value of $n_o - n_e$ potential errors had been avoided during each session. The efficacy of occluded markers handling can also be extracted by watching the difference between $\frac{T}{n_e}$ and $\frac{T}{n_o}$ values. During the third experiment, for example, the recorded $\frac{T}{n_o}$ is less than 2 seconds while $\frac{T}{n_e}$ is almost 18 seconds. Without the algorithm the average time between each error could have been exactly the same as the average time between occlusions.

V. Conclusions

The proposed algorithm represents an effective indexing technique that may be applied whenever real time points distinction is required, especially for human-computer interaction purposes. We had the chance to test its efficacy by providing a Virtual Reality application with simple indexing-based interactions, also maintaining a lightweight markers setup.

A small video, showing test and application, can be seen at http://youtu.be/r2GsxXblD8s*http://youtu.be/r2GsxXblD8s*. The video has been uploaded for this submission only.

Experiment session data shows that the algorithm can not completely avoid indexing errors occurrences, but it can decrease their rate, sometimes drastically. The work done towards configuration recognition and indexing correction confirmed that real time indexing algorithms can be improved by introducing session-specific constraints. The indexing algorithm can be used with any motion capture application to introduce real time indexing for the tracked features. This may allow the introduction of new possibilities in terms of motion capture data usage. Since the algorithm development was mainly designed for Virtual Reality interaction, its features are better suited for this purpose, especially the posture recognition and correction technique. The Virtual Reality environment seen in Sec. IV is an example of indexing enhanced Virtual Reality interaction. The same application would have not been possible without the algorithm indexing and its correction feature: their lack could have led to frequent and non predictable changes in the interaction protocol, making the interface almost impossible to be used. The algorithm could also prove to be effective in the reduction of data correction time for non-real time applications, but since the goal of this research was mainly set towards real time, interaction-oriented motion capture, no tests have been done in this direction.

VI. Future Work

An optimization technique could be introduced to speed up the main algorithm matching procedure, and to achieve a lower indexing errors occurrence rate. Defining as displacement the distance between a point and its associated next position, the solution could be enhanced through the minimization of global displacement during correspondences research.

The applications created to test the algorithm took advantage of a generic version of the recognition feature, but the algorithm is capable of indexing an arbitrary number of points. This work could inspire the exploration of other natural interaction techniques, and additional studies could be addressed in order to create new environments using more specific and advanced versions of the correction feature. An increase of the recognition occurrence rate could help keeping a correct indexing. This could be done by recording more than one calibration posture per session, or by finding suitable model-based constraints describing, for example, typical hands, fingers or body behaviors. A more accurate investigation of session-specific constraints will be required in order to search for a more precise model-based recognition feature. Such feature could substitute current distance-based recognition and increase recognition rate. Depending on tracked markers configurations and available hardware and software, more complex hybrid approaches may further increase the indexing correctness stability and capabilities by taking advantage of modern computer vision techniques.

References

[1] F. Berard. The Magic Table: Computer Vision Based Augmentation of a Whiteboard for Creative Meetings. In *IEEE International Conference in Computer Vision*, 1993.

[2] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *SIGGRAPH'93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, 1993.

[3] K. Dorfmuller-Ulhaas and D. Schmalstieg. Finger tracking for interaction in augmented environments. In *IEEE and ACM International Symposium on Augmented Reality (ISAR01)*, 2001.

[4] W. T. Freeman, D. B. Anderson, P. A. Beardsley, C. N. Dodge, M. Roth, C. D. Weissman, W. S. Yerazunis, H. Kage, K. Kyuma, Y. Miyake, and K. ichi Tanaka. Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications*, 18:42–53, 1998.

[5] J. C. Lee. Hacking the nintendo wii remote. *IEEE Pervasive Computing*, 7:39–45, 2008.

[6] Microsoft, http://www.xbox.com/kinect/.

[7] Microsoft Research, http://research.microsoft.com/en-us/.

[8] Letessier J. and Berard F. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST 04)*, 2004.

[9] T. B. Moeslund. Interacting with a virtual world through motion capture. In *Interaction in Virtual Inhabited 3D Worlds, chapter 11*. Springer-Verlag, 2000.

[10] T. B. Moeslund and E. Granum. A survey of computer vision-based. *Computer Vision and Image Understanding*, 2006.

[11] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 2006.

[12] Nintendo, http://wii.com/.

[13] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. In *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 1997.

[14] R. Pinho, J. R. S. Tavares, and M. V. Correia. Human movement tracking and analysis with kalman filtering and global optimization techniques. In *II International Conference on Computational Bioengineering*, September 2005.

[15] H. Schepers. *Ambulatory Assessment of Human Body Kinematics and Kinetics*. PhD thesis, Universiteit Twente, 2009.

[16] M. J. Tarr and W. H. Warren. Virtual reality in behavioral neuroscience and beyond. *Nature Publishing Group*, 2002.

[17] C. von Hardenberg and F. Bérard. Bare-hand human-computer interaction. In *Proceedings of the 2001 workshop on Perceptive user interfaces*, PUI '01, pages 1–8, New York, NY, USA, 2001. ACM.

[18] V. Zappi, A. Brogni, and D. Caldwell. Passive hand pose-recognition in virtual reality. In *Proceedings of IEEE International Symposium on 3D User Interfaces*, 2010.