Online Human Assisted and Cooperative Pose Estimation of 2D cameras[1]

Gaetano Manzo[a], Francesc Serratosa[b] & Mario Vento[c]

[a]University of Bern, Bern, Switzerland.
[b]Universitat Rovira i Virgili, Tarragona, Catalonia, Spain
[c]University of Salerno, Salerno, Italy.
(a)gaetano.manzo@students.unibe.ch,
(b)francesc.serratosa@urv.cat,(c)mvento@unisa.it

Abstract. Autonomous robots performing cooperative tasks need to know the relative pose of the other robots in the fleet. Deducing these poses might be performed through structure from motion methods in the applications where there are no landmarks or GPS, for instance, in non-explored indoor environments. Structure from motion is a technique that deduces the pose of cameras only given only the 2D images. This technique relies on a first step that obtains a correspondence between salient points of images. For this reason, the weakness of this method is that poses cannot be estimated if a proper correspondence is not obtained due to low quality of the images or images that do not share enough salient points. We propose, for the first time, an interactive structure-from-motion method to deduce the pose of 2D cameras. Autonomous robots with embedded cameras have to stop when they cannot deduce their position because the structure-from-motion method fails. In these cases, a human interacts by simply mapping a pair of points in the robots' images. Performing this action the human imposes the correct correspondence between them. Then, the interactive structure from motion is capable of deducing the robots' lost positions and the fleet of robots can continue their high level task. From the practical point of view, the interactive method allows the whole system to achieve more complex tasks in more complex environments since the human interaction can be seen as a recovering or a reset process.

Keywords: 2D pose estimation; Robot cooperation; human-robot interaction; social robots; structure from motion.

1. Introduction

We present a method to obtain the pose of several 2D cameras that has two novel features. The first is that the human can assist in mapping salient points between pairs of images when one estimates the error is too high. The second is the ability of the method to deduce the poses in a cooperative manner. The system always tries to automatically deduce the poses of the whole cameras meanwhile the human visualises the deduced poses together with the estimated errors in a human-machine interface but, when the human considers appropriate, one can asynchronously interact on the system. Thus, the supervisor can partially modify the point-to-point mapping between two images, which increases the quality of the deduced homography between these

[1] This research is supported by Spanish projects DPI2013-42458-P & TIN2013-47245-C2-2-R.

images. This interaction decreases not only the poses error of the involved two cameras but also the pose errors of the whole set of cameras.
Figure 1 shows a schematic view of our method based on a module, which we have called interactive pose estimator, and a human-machine interface. In this example, cameras are embedded on the robots. The input of the general system is a set of 2D images and the output is their relative poses and the estimated errors (as the GPS does it). The human-machine interface receives the 2D-images from the cameras, their current relative poses and the mapped points per pair of images from the interactive pose estimation module. The human-machine interface only outputs the user operation, in other words, the point-to-point mapping impositions to the interactive pose estimation module. Besides, the interactive pose estimation also receives the 2D images and then deduces and sends the relative poses estimation and the regression errors to the system that controls the robots.
The human-machine interface is as follows. On the left side of it, the user visualises the deduced current poses of the cameras (2D position on the land and robot orientation). In the middle of the interface, the number of mapped points between any pair of cameras is shown. The lowest values of the number of mappings are highlighted in bold to attract the attention of the user. On the right side of it, the user visualises the 2D images of two manually selected cameras together with the imposed point-to-point correspondences. The user can visualise any pair of 2D images by selecting one of the cells in the middle of the human-machine interface. Thus one can update the imposed correspondence by erasing or creating mappings between points. The two robots (or cameras) in the left panel and the pose errors in the central panel that correspond to the current images in the right panel are highlighted in red.

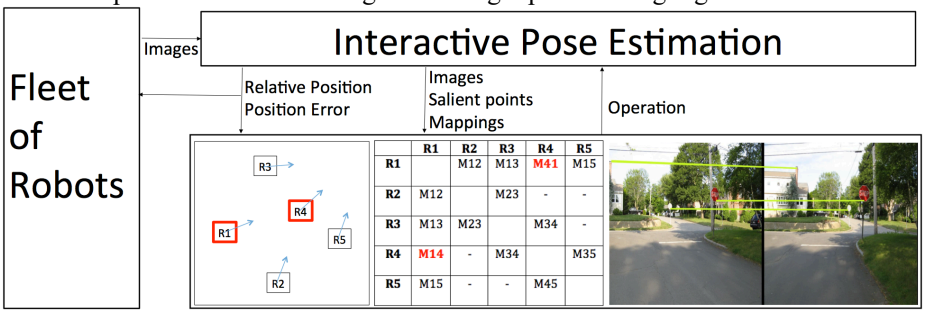


Figure 1. Basic scheme of our method composed of our interactive pose estimation module and the human-machine interface.

In robotics or camera surveillance, properly estimating the pose of cameras is considered to be a crucial low-level task. In the first case, cameras are embedded on the robots and therefore have mobile positions. By contrast, in the second case, we have static cameras. Nevertheless, in both cases, without properly setting the camera poses, the method is not able to deduce the position, direction, speed or acceleration of the objects or humans that are in the surroundings.
The method we present is part of a larger project in which social robots guide people through urban areas (http://www.iri.upc.edu/project/show/144). We have analysed the relation between humans and robots and also the behaviour of humans when social robots move closer to them (Garrell & Sanfeliu, 2012). We have also presented a tracking method that follows people, which allows occlusions and mobile cameras (Serratosa, Alquézar & Amézquita, 2012) and a robot navigation system (Ferrer, G.,

Sanfeliu, A., 2014). Moreover, we have presented some results on structure from motion. In other words, given several 2D cameras, the method reconstructs the 3D position of the cameras (Rubio et. al., 2015).

One important aspect of this project is the human-machine interaction. Human interaction has been applied to classify objects (Ferrer, G., Garrell, A., Villamizar, M., Huerta, I. & Sanfeliu, A., 2013) and also to deduce the pose of the robots.

Several levels of interaction could be considered to deduce the pose of the robots. The highest level could be to impose the position of a camera resulting from this knowledge has been acquired through another method. Our proposal is related to the lowest level. A human is very good and fast at mapping points on two different scenes, independently of the intrinsic or extrinsic characteristics of the images. Thus, what the user is asked to do is simply to select a salient point on one of the images and map this point on another image. Note this action is performed asynchronously to the process of deducing the pose and the supervisor tends to perform it when robots have to stop because the system is unable to deduce the pose in a completely automatic way. Therefore, the human interaction is a mechanism which allows the robot to continue in extreme situations.

Two papers have been presented that has be seen as necessary previous research in order to achieve the current paper. In the first, the homography between 2D images is computed (Cortés & Serratosa, 2015) and in the second, the 3D positions of the robots are deduced given 3D cameras (Cortés & Serratosa, 2016). In (Cortés & Serratosa, 2015), the influence of some human point-to-point mapping impositions while deducing the homographies that convert one image into the other was analysed for the first time. That paper concluded that with very few impositions, the end-point error generated by the semi-automatically deduced homographies is much lower than the error committed by the automatically deduced homographies. With this knowledge, in (Cortés & Serratosa, 2016), the poses of some robots were deduced in a semi-automatic way. Nevertheless, to avoid the process of estimating 3D poses from 2D images, known as structure-from-motion (Xu, C., Tao, D., and Xu, C., 2015), the paper assumes robots have 3D and 2D cameras. The 3D images are used to compute the 3D point-to-3D point correspondences and the 2D images are shown to the human. Since both cameras have been calibrated, when the human imposes 2D point-to-2D point correspondences, these are easily converted to 3D point-to-3D point correspondences and used to deduce the final poses in a semi-automatic way.

In this paper, we move one step further since the robots only have 2D cameras and the method obtains the 3D pose of the cameras performing structure from motion with human interaction. Note that this is the first time that an interactive structure from motion method has defined.

Figure 2 (left) represents three robots performing guiding tasks. Robots fence the visitor group to force them to follow a specific tour. Robots need to work in a cooperative manner to keep a triangular shape in which people have to be inside. In these cooperative tasks, it is crucial to have a low-level computer vision task so that images extracted from the three robots and some static cameras in the surrounding are properly aligned to correctly deduce their relative poses. In this environment, there is a human that, through our human-machine interface, gives orders to the robots and controls their tasks. What we propose in this paper is that the human can also visualise the images of the cameras and interact in a low level task through asynchronously imposing point-to-point mappings. Figure 2 (right) shows the images taken from the first two robots and three mappings that the human has imposed. Note

that the system deduces the poses automatically but, when the human asynchronously imposes a mapping, then it takes into consideration this mapping and it continues its process automatically.

Figure 2. A representation of three robots performing guiding tasks and images taken from the cameras on the first two robots.

Besides, we say it is a cooperative model since the relative pose of two robots is deduced through the other robots when these robots do not share any part of the scene they visualise. Figure 3 shows the pose of three robots. Robot 1 and Robot 2 can deduce their relative pose but this is not possible between Robot 1 and Robot 3 since they do not share any part of their images. This problem is solved through the cooperation of the robots. Robot 2 deduces its relative pose with respect to Robot 3 and shares this information with Robot 1.

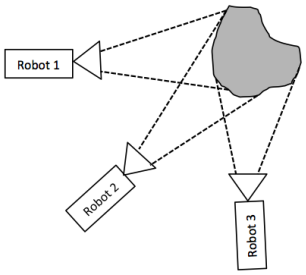


Figure 3. Three robots visualising the same scene.

The rest of the paper is organized as follows. In the next section, we first summarise the state of the art related to human-robot interaction. Then, we explain how to deduce the relative pose of an object with respect to another one given an affine 3D homography. We also comment on the four basic methods used in our robotic system, namely salient point extractor, feature matching, structure from motion and human interaction. In section 3, we present our model. We first describe the main scheme and then we concretise on two specific parts: the human assisted matching estimator and the cooperative pose estimator. Note that the human-machine interface has been described in the introduction. In section 4, we experimentally validate our model. In a first sub-section we show that with few human interactions, the accuracy of the estimated pose drastically increases. In the other two sub-sections, we apply our method to automatic robot positioning and camera calibration. We conclude the paper in section 5.

2. State of the art and basic methods

In recent years, interaction between robots and humans and also cooperation between robots has increased rapidly. Applications in this field are very diverse, ranging from developing automatic exploration sites (Trevai, Ota, Fukazawa, Yuasa, Arai & Asama, 2004) to using robot formations to transport and evacuate people in emergency situations (Casper & Murphy, 2003), assembly lines (Unhelkar, Shah, 2015) or simply vehicle positioning (Ifthekhar, Saha, Jang, 2015). Within the area of social and cooperative robots we have (Kim, Taguchi, Hong & Lee, 2014; Garcia, Cena, Cardenas, Saltaren, Puglisi & Santonja, 2013) and in hospital care application we have (Jeong et. al., 2015). As commented in the introduction, interactions between a group of people and a set of accompanying robots have become a primary point of interest in (Garrell & Sanfeliu, 2012).

In the next two sub-sections, we sum up the following methodologies: Salient point extractor, feature matching, structure from motion and human interactivity, which are used by our method.

These methodologies, which involved pose estimation, usually assume the transformation between two 3D images or two sets of 3D points is modelled as an affine transformation in the 3D space. Thus homography $H_{i,j}$ is defined as follows,

$$H_{i,j} = \begin{bmatrix} a_{i,j} & 0 & 0 & x_{i,j} \\ 0 & b_{i,j} & 0 & y_{i,j} \\ 0 & 0 & c_{i,j} & z_{i,j} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $a_{i,j} = S_{i,j} \cdot cos(\beta_{i,j}) \cdot cos(\gamma_{i,j})$, $b_{i,j} = S_{i,j} \cdot cos(\alpha_{i,j}) \cdot cos(\gamma_{i,j})$ and $c_{i,j} = S_{i,j} \cdot cos(\alpha_{i,j}) \cdot cos(\beta_{i,j})$. Parameter $S_{i,j}$ is the scale and $\alpha_{i,j}$, $\beta_{i,j}$ and $\gamma_{i,j}$ are the three orientation angles of one robot with respect to the other. Besides $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$ is the translation of one robot with respect to the other.

Thus, given the affine homography $H_{i,j}$, the relative positions are simply values $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$. Moreover, one combination of $\alpha_{i,j}$, $\beta_{i,j}$ and $\gamma_{i,j}$ angles is deduced by solving the following equations,

$$cos(\beta_{i,j}) \cdot cos(\gamma_{i,j}) = \frac{a_{i,j}}{S_{i,j}}$$

$$cos(\alpha_{i,j}) \cdot cos(\gamma_{i,j}) = \frac{b_{i,j}}{S_{i,j}}$$

$$cos(\alpha_{i,j}) \cdot cos(\beta_{i,j}) = \frac{c_{i,j}}{S_{i,j}}$$

Errors in poses are considered as described in (Huynh, D.Q., 2009). The position error is computed as the Euclidean distance between the computed position in 3D and the ground truth position. The angle error is computed as $\left\| I - R_{i,j}^{sys} \cdot \left(R_{i,j}^{grt}\right)^T \right\|_F / 2\sqrt{2}$ where $\|.\|_F$ is the Frobenius norm, $R_{i,j}^{sys}$ and $R_{i,j}^{grt}$ are the obtained and ground truth rotations between images i and j and $(.)^T$ represents the transpose matrix. The range of this expression is $[0,1]$ since it is demonstrated in (Huynh, D.Q., 2009) that the maximum value of $\left\| I - R_{i,j}^{sys} \cdot \left(R_{i,j}^{grt}\right)^T \right\|_F$ is $2\sqrt{2}$.

2.1. Salient point extractor and feature matching

Salient points are image locations that can be robustly detected among different instances of the same scene with varying imaging conditions and they play the role of parts of the image to be matched. Several feature extractors have been presented to detect salient points such as FAST (Rosten, E., Reid Porter, R., Drummond, T., 2010), HARRIS (Harris, C., Stephens, M., 1988), MINEIGEN (Jolliffe, I.T., 2002), SURF (Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008), SIFT (Lowe, D.G., 2004), BRIEF (Calonder, M., Lepetit, V., Strecha, C., Fua, P., 2010), BRISK (Leutenegger, S., Chli, M., Siegwart, C.R, 2011) and Alahi, A., Ortiz, R., Vandergheynst, P., (2012). There is an interesting comparison in (Kashif, M., Deserno, T., Haak, D. and Jonas, S., 2016) and a previous evaluation of the most competent approaches in (Mikolajczyk & Schmid, 2005).

When salient points have been detected, several correspondence methods can be applied that obtain the alignment (or homography) that maps one image into the other (Zhang, Z., 1994), discards outlier points (Fischler & Bolles, 1981) or characterises the image into an attributed graph (Sanromà, et al., 2012a; Sanromà, et al., 2012b; Serratosa & Cortés, 2015; Serratosa, 2014; Serratosa, 2015a; Serratosa, 2015b; Solé, et al., 2012). Typically, these methods have been applied to 2D images but recently 3D shape retrieval methods have appeared (Lia, B., 2015).

Some correspondence methods consider a rigid deformation from one image to the other one and other ones consider a non-rigid deformation. In the first case, it is assumed the whole image (and the extracted salient points) suffers from the same deformation and therefore the image alignment parametres are applied equally to the whole salient points or image pixels. Some examples are (Sanromà, et al., 2012a; Luo & Hancock, 2003; Rangarajan, Chui & Bookstein, 1997; Gold, Rangarajan, 1996). In the second case, each salient points suffers a different projection and there are different alignment parametres applied to each salient point or image region. Examples include (Myronenko, Song, 2010; Chui, Rangarajan, 2003). Usually, the rigid strategy is applied to detect objects on outdoor images in which the deformation is mostly due to the change of the point of view. The non-rigid strategy is mostly applied to object detection or matching in medical or industrial images since it is assumed that objects suffer from deformations although the point of view is the same.

2.2. Structure from motion and human interactivity

Structure from motion refers to the process of estimating 3D structures from 2D image sequences (Xu, C., Tao, D., and Xu, C., 2015), which may be coupled with local image features. Finding structure from motion presents a similar problem to finding structure from stereovision. In both instances, the correspondence between images and the reconstruction of 3D object needs to be found. To find the correspondence between images, local features commented in the previous sub-section are tracked from one image to the next.

There are several approaches to solve structure from motion (Yi, G., Jianxin, L., Hangping, Q., Bo, W., 2014; Gui, J., Tao, D., Sun, Z., Luo, Y., You, X. and Tang, Y. 2014). In incremental structure from motion, poses are solved by adding one by one to the collection. In global structure from motion, the poses of all cameras are solved at the same time. A somewhat intermediate approach is out-of-core structure from motion, where several partial reconstructions are computed that are then integrated into a global solution.

Humans are very good at finding the correspondences between local parts of an image regardless of the intrinsic or extrinsic characteristics of the point of view. Human interactivity on feature matching has been applied to medical images (Pfluger & Thomas, et al., 2000; Pietrzyk, et al., 1994; Khader & Ben Hamza, 2012). Moreover, two patents have been presented (Von & Neitzel, 2010; Gering & David, 2010). These methods are specific to certain medical environments and for this reason cannot be applied to our problem. In (Pfluger & Thomas, et al., 2000), they show a comparison of 3D images in MRI-SPECT format and they concretise on images from the brain. In (Pietrzyk, et al., 1994), authors present a method to validate the 3D position given 3D medical images. Finally, in (Khader & Ben Hamza, 2012), the aim is to solve the feature matching given similar medical images extracted from different sensors or technologies. Patent (Von & Neitzel, 2010) defines a system for registration of thorax X-Ray images given that it does not depend on bony structures. Finally, patent (Gering & David T., 2010) defines a multi-scale registration for medical images where images are first aligned at a course resolution, and subsequently at progressively finer resolutions; user input is applied at the current scale. Another typical application of human interaction is semi-automatic video annotation (Bianco, Ciocca, Napoletano & Schettini, 2015).

The method we present in this paper puts in common structure from motion and human interactivity. Given a correct (or almost correct) salient point correspondence between some images, completely automatic methods are very good at deducing the 3D pose of cameras but it is a difficult task for the humans. By contrast, finding a correct correspondence of several images can be a very elaborate task for an automatic method but humans perform it very well and quickly. This method takes advantage of the qualities of the automatic and human behaviours since the automatic deduceion of the pose is done by the automatic method but the human can help on deducing the point correspondences when the automatic method fails to do it.

The strength of this method is that more accurate cameras poses are achieved than the ones deduced through the current structure from motion method (Xu, C., Tao, D., and Xu, C., 2015). When our method is applied to autonomous robots, the increase in pose accuracy means the robots miss their position fewer times. Moreover, when the method is applied to camera surveillance, the pose is deduced more accurately and so it does the objects' position estimations. On the other hand, the weakness of the method is the need for a human per se. Moreover, a human spends a minimum of 3 seconds to map two points. Thus, where a human interaction is needed, there is a clear slow down in the system. When our method is applied to autonomous robots, it means the system has not recovered immediately. This delay is not so important when the method is applied to camera surveillance since it is supposed that the process is performed while tuning the system.

3. The proposed model

Figure 4 shows our method in schematic terms. We suppose there is a controlling system that needs to know the poses of some 2D cameras. As commented, given only the images from these cameras, our system deduces their poses together with an error and sends this information to the controlling system.

The human-machine interface, which is explained in the introduction section, shows the position of the whole cameras and also the selected images to the user. It is a friendly interface to impose point-to-point mappings between pairs of images.

The first module of our system is the scheduler that selects which N cameras are going to be considered and returns their current images $I_i, I_j, \ldots, I_k$. Several strategies can be used depending on the application. If the human does not interact in the human-machine interface, the strategy could be to select the cameras that the current number of mappings is the largest and also depending on the deduced distances between them. Nevertheless, when the human selects a pair of cameras, these ones are scheduled with the closest ones.

The next two steps are composed of salient point extractors and feature matching methods. Several methodologies summarised in section 2 can be used depending on the application. The first step generates a set of points per image, $P_i, P_j, \ldots, P_k$ and the second generates the $N \cdot (N-1)/2$ correspondences $M_{i,j}$, $M_{i,k}, \ldots, M_{j,k}$ between these sets (we suppose $M_{i,j} = M_{j,i}$). The aim of the next module, called human-assisted matching estimator, is to adapt the automatically deduced set of point-to-point mappings to the mappings imposed by the human. We call the updated set of correspondences $M'_{i,j}, M'_{i,k}, \ldots, M'_{j,k}$. We explain this module in detail in the section 3.1. The structure from motion module deduces the homographies $H_{i,i}, H_{i,j}, \ldots, H_{i,k}$ in the 3D space that convert the pose of the first selected camera to the rest of the selected ones. To do so, it needs the input to be the set of point-to-point mappings in the 2D space. Several methodologies have been summarised in section 2.3.

The last module is the cooperative pose estimator. The aim of this module is to return the M^2 3D homographies between the whole set of cameras, $H_{1,1}, H_{1,2}, \ldots, H_{M,M}$ and their estimated error. On the one hand, each time the whole process is computed, only a sub-set of cameras are selected and therefore, only a sub-set of 3D homographies are deduced by the structure from motion module. On the other hand, if the point of view of some cameras is such that it is not possible to share any point in the 3D space, then, it is impossible for the structure from motion module to deduce their respective 3D homographies. This module is explained in detail in section 3.2.

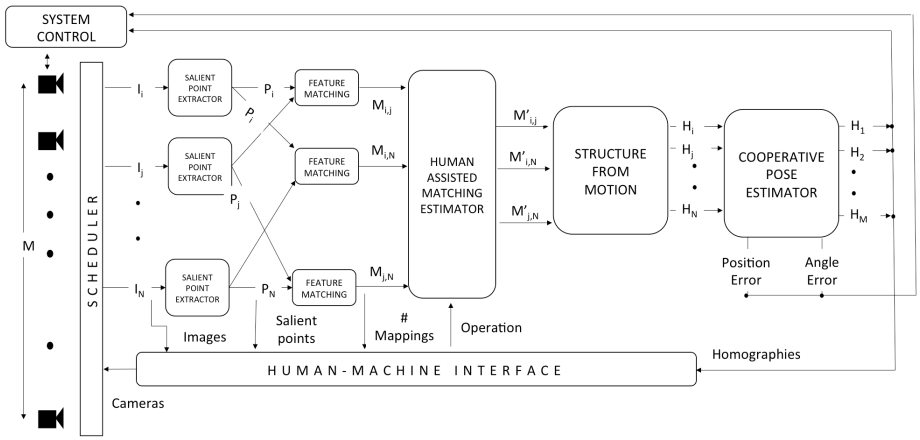


Figure 4. Scheme of our human assisted and pose estimation method.

3.1. Human-assisted matching estimator

Through the human-machine interface, the user can perform three different operations that we call: $False\ Mapping$, $True\ Mapping$ and $Set\ Mapping$. It is usual for the feature matching algorithms not only to return a set of mappings but also a goodness

of each mapping. This goodness is related to the distance between the mapped features extracted at the points. Thus, these three operations influence a specific point-to-point mapping and also the goodness of this mapping. The user interacts on the human-machine interface in a non-synchronous way. For this reason, the human-assisted matching estimator can receive only one user operation or some of them. The formats of these operations are as follows.

$False\ Mapping(p_i^a, p_j^b)$: The user considers the mapping from point a on image i to point b on image j has to be deleted since it is not correct. $True\ Mapping(p_i^a, p_j^b)$: The user is completely sure that the mapping from point a on image i to point b on image j is correct. $Set\ Mapping(p_i^a, p_j^b)$: The user imposes a new mapping from point a on image i to point b on image j.

When the human assisted matching estimator receives the action $False\ Mapping(p_i^a, p_j^b)$, it checks the existence of this mapping and, if it exists, deletes it. Moreover, when this module receives the action $True\ Mapping(p_i^a, p_j^b)$, it checks the existence of this mapping and, if it exists, increases its goodness parameter to the maximum. Finally, action $Set\ Mapping(p_i^a, p_j^b)$ makes the module impose this new mapping with the maximum goodness.

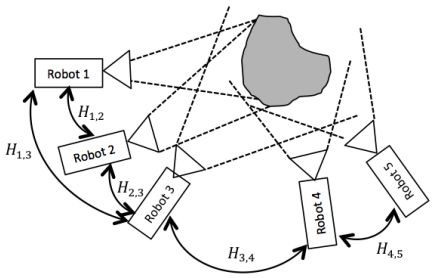
This module has the $Matching\ Matrix$, or MM for short, in which there is the last mapping between a pair of images in each cell. The user impositions directly update this matrix.

3.2. Cooperative pose estimator

The cooperative pose estimator receives a set of 3D homographies from the structure from motion module. As commented, the homographies in this set are only related to pairs of images selected by the scheduler. The cooperative pose estimator has an M x M matrix, that we call $Direct\ Homographies$ or DH for short. In each cell, there is the last computed homography by the structure from motion module: $DH[i,j] = H_{i,j}$. Thus, this matrix is updated each time the cooperative pose estimator receives new homographies. Note the diagonal of this matrix is filled with the identity homography. When the application initialises, the whole matrix is empty and it is being filled when the scheduler begins to output images. Nevertheless, the cells that cameras do not share any point always remain empty. Since we wish to output the homographies between the whole cameras, we define an M x M matrix, that we call $Computed\ Homographies$ or CH for short. The non-empty cells in the DH matrix are copied to this new matrix when new data is available but the rest of the cells are deduced in a cooperative manner.

Figure 5 (left) shows a scene where five robots visualise an object and there is a high overlapping between images of the first three robots but Robot 4 only shares part of the image with Robot 3 and Robot 5. Initially, they do not know the relative position of the other robots since robots do not have any system to locate themselves in the scene and therefore they do not know which robot is the closest to another. Figure 5 (right) shows DH matrix (I represents the identity). Robot 1 can deduce the relative pose of Robot 2 directly (through homography $H_{1,2}$) but it is not able to deduce the relative pose of Robot 4 or Robot 5. Nevertheless, in the case of Robot 4, this information can be indirectly estimated through Robot 3 as the product of homographies $H_{1,4} = H_{1,3} \cdot H_{3,4}$. Another option could be to compute this homography so that Robot 2 takes part of it, $H_{1,4} = H_{1,2} \cdot H_{2,3} \cdot H_{3,4}$. Since we do not

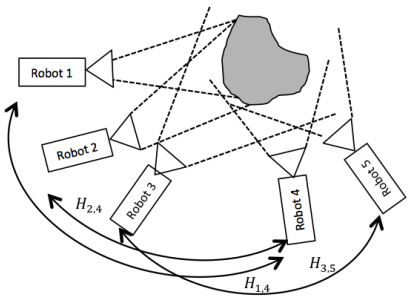
know which is the best way to deduce the relative pose, we estimate the mean error is zero and we compute the empty cells in CH through the average of angles and positions of the available homographies in DH.



	R1	R2	R3	R4	R5
R1	I	H_{12}	H_{13}	-	-
R2	H_{21}	I	H_{23}	-	-
R3	H_{31}	H_{32}	I	H_{34}	-
R4	-	-	H_{43}	I	H_{45}
R5	-	-	-	H_{54}	I

Figure 5. Five robots looking at an object and the $Direct\ Homographies$ matrix, DH.

Algorithm 1 returns CH matrix and the estimated error of each homography given DH matrix. This algorithm is executed by the cooperative pose estimator module each time new homographies are received from the structure from motion module. It is composed of two main steps. The first updates the cells in CH that the module has received new homographies. In the second step, the empty cells in CH are filled in a cooperative way. The first time the $Loop - While$ is executed, the mean is computed in cases where there is only one extra camera. $H_{1,4}$ and $H_{2,4}$ can only be deduced through Camera 3, $H_{1,4} = H_{1,3} \cdot H_{3,4}$ and $H_{2,4} = H_{2,3} \cdot H_{3,4}$. Moreover, $H_{3,5}$ can only be deduced through Camera 4, $H_{3,5} = H_{3,4} \cdot H_{4,5}$. Figure 6 shows the newly-obtained homographies and CH after the first time the $Loop - While$ is executed.



	R1	R2	R3	R4	R5
R1	I	H_{12}	H_{13}	H_{14}	-
R2	H_{21}	I	H_{23}	H_{24}	-
R3	H_{31}	H_{32}	I	H_{34}	H_{35}
R4	H_{41}	H_{42}	H_{43}	I	H_{45}
R5	-	-	H_{53}	H_{54}	I

Figure 6. Homographies and the $Computed\ Homographies$ matrix (CH) deduced in the first run of the loop in Algorithm 1.

```
Algorithm 1
Input DH
Output CH ERROR
N=length(DH);
%% Step one
for i=1:N
    for j=1:N
        if not(empty(DH[i,j])
            ERROR[i,j] = 0 % Error is 0 if directly computed
            CH[i,j] = DH[i,j]
        endif
    endfor
endfor
%% Step two
do
UPDATE=false
 for i=1:N
     for j=1:N
         if empty(DH[i,j]) % New data does not have info.
             for k=1:N
                 if not(empty(CH[i,k]) & not(empty(CH[k,j]))
                     Hk=CH[i,k]*CH[k,j]
                     Pk=Pose(Hk)
                     UPDATE=true
                 else
                     Pk=empty
                 endif
             endfor
             Pij=mean(P1,P2,..,Pn) % Empty Pi are discarded
             ERROR[i,j]=Standard_Deviation(P1,P2,..,Pn)
             CH[i,j]=Homography(Pij) % Return empty if Pi is empty
           endif
      endfor
  endfor
while UPDATE
End Algorithm
```

In the second run of the loop, the mean is computed in cases where there are two extra cameras. For instance, $H_{1,5}$ can be computed as $H_{1,3} \cdot H_{3,5}$ but since $H_{3,5}$ was deduced in the first loop as $H_{3,4} \cdot H_{4,5}$ then in fact, we have Camera 3 and Camera 4 involved, $H_{1,5} = H_{1,3} \cdot H_{3,4} \cdot H_{4,5}$. Nevertheless, another option is possible, which is $H_{1,5} = H_{1,4} \cdot H_{4,5}$. Therefore, $H_{1,5}$ is computed in the second run of the loop as the mean of the first option $H_{3,4} \cdot H_{4,5}$ and the second option $H_{1,4} \cdot H_{4,5}$.
The pose error is set to zero if the homography is directly computed by the structure from motion module or it is computed in the first run of the loop. In the other runs of the loop, the error is considered to be the standard deviation of the whole poses involved in computing the new homographies.
The algorithm stops when the whole cells are filled or it is not possible to fill them. This last case appears when there are two or more sets of cameras without shearing enough points.

4. Practical Validation

We propose three different databases to validate our method, which are summed up in table 1. In the following sub-sections, we explain each database and the errors committed while estimating the pose of the cameras. Each database is composed of a set of images and the ground truth of the 3D poses of each image. Databases are

available at (http://deim.urv.cat/~francesc.serratosa/databases/). The feature point extractor was SIFT (Lowe, D.G., 2004), the matching feature algorithm was the Matlab function matchFeatures (http://es.mathworks.com/help/vision/ref/matchfeatures.html) and the structure from motion method has been Bundler (http://www.cs.cornell.edu/~snavely/bundler/). The Matlab code of our method is available at (http://deim.urv.cat/~francesc.serratosa/SW/).

Database	# images	# cameras	Static/mobile	Indoor/outdoor
Sagrada Família	360	8	Static	Outdoor
Courtyard	32	2	Mobile	Outdoor
SSAI Lab	8	8	Static	Indoor

Table 1. Mean features of the three used databases.

4.1. First analysis: Camera positioning on open spaces

Sagrada Família database is a sequence of 360 2D images manually taken around the Sagrada Família church in Barcelona (Spain) and looking at the centre of it. The average distance between two consecutive shots is 1.1 metres. Given the whole sequence, we used the method called Bundler (Snavely & Todorovic, 2011) to extract the pose of each camera that we consider is the ground truth. Figure 7 shows the set of 3D points extracted from Bundler method and also the poses of the cameras. More than 100.000 3D points where extracted and for this reason, an I7 processor spent more than 7 hours to deduce the 3D poses of the cameras. The database is composed of 360 images, the 360 poses of the cameras and the set of 2D and 3D salient points. In this experiments, we have only used the set of images to deduce the poses and the cameras' poses as a ground truth.

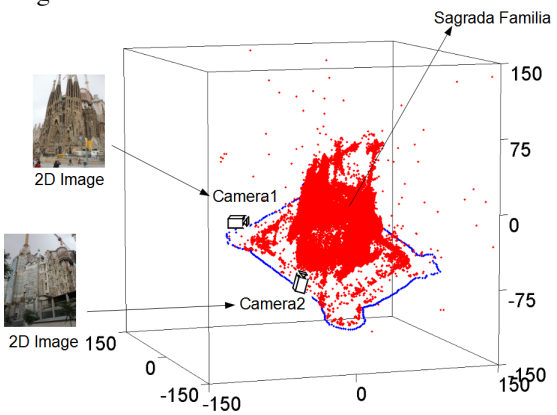


Figure 7. 3D points extracted by Bundler method of Sagrada Família database and the deduced 3D poses of the cameras.

Figure 8 shows the first eight images of this database, the deduced poses of the cameras given only these eight images and the set of points that Bundler has obtained and it considers that appear in the eight images. In this case, the runtime spent to compute the cameras' poses is lower than a second. There is a building on the left hand of these images that appears to be represented as a cloud of points that has a linear shape in front of the cameras. Note the branches of a tree that hinders the proper extraction of salient points in the images.

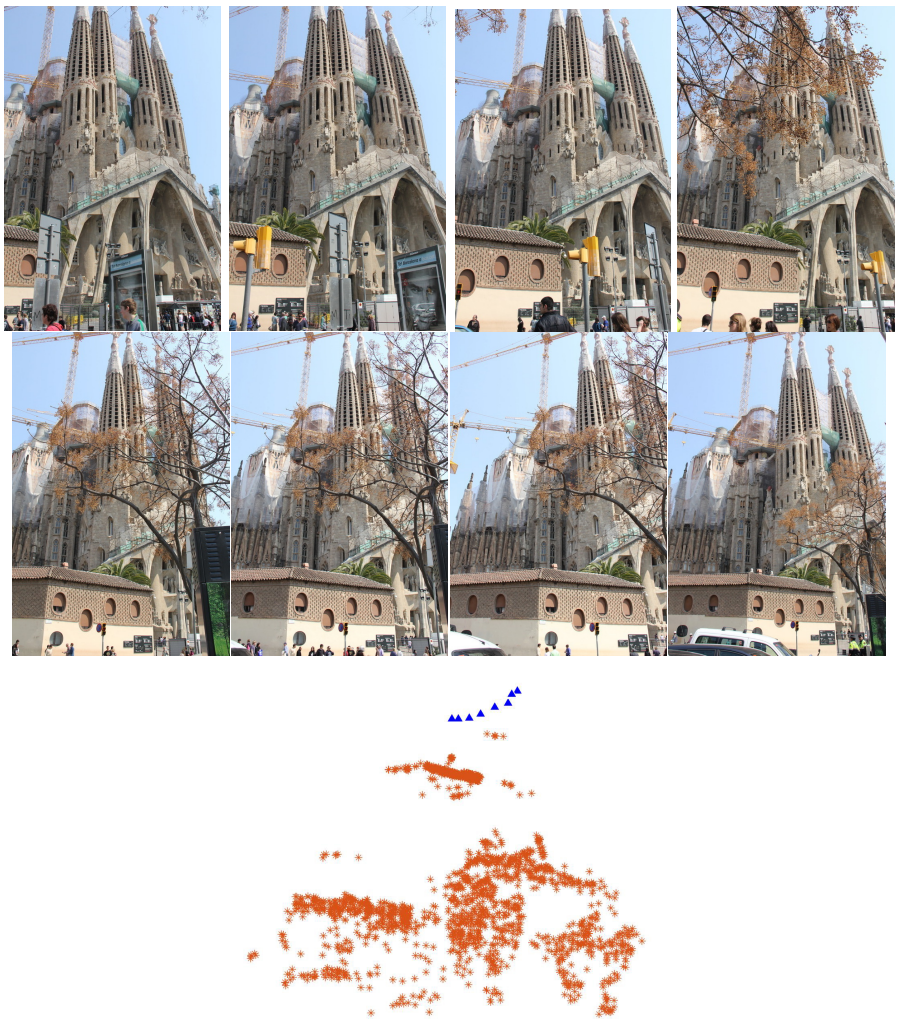


Figure 8. The first eight images of Sagrada Família database and the deduced 3D points and poses of the cameras.

Each result we present in this section is the average of 360 consecutive tests. In each test, we have taken 8 consecutive images (we suppose there are 8 cameras each time) from the 360 images that constitute the database with the angle of separation between images. For instance, test number 350 with angle of separation 2 is composed of images 350, 353, 356, 359, 2, 5, 8 and 11. Figure 9 shows the relative position error (left) and the relative angle error (right) returned by the system with respect to the number of interactions and the angle of separation between images. Moreover, Table 2 and Table 3 show the standard deviation of the position error and the relative angle error without interactions and with one interaction. The standard deviations of more than one interaction are almost the same as with one interaction. We realise errors have little variability and, in general, the variability is in concordance with the mean error.

As it is supposed to be, the further away the images, the larger the error is since less 2D points are shared and also the larger the distortion is between images. It is clear that with one interaction, the error is drastically reduced irrespective of the angle of separation between paired images. The error was only slightly reduced when more than one interaction was imposed. Moreover, we have checked that with angles equal to or larger than 5, the method was not able to deduce the poses without the cooperative module. This is because the angle between the first camera and the eighth one is too large. For instance, with angle 5 the angle between the first camera and the last one is 35 and with an angle of 20, this last angle is 140.

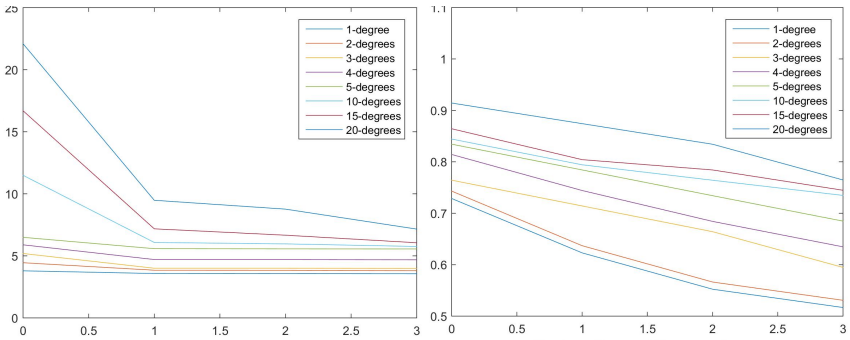


Figure 9. Mean errors on the positions in metres (left) and relative angle error (right) with respect to the number of interactions and the angle of separation between cameras. The relative angle error is the angle error in degrees divided by 360.

Degree	1	2	3	4	5	10	15	20
Non interaction	2	1.8	1.7	0.8	0.7	0.8	0.6	0.3
One interaction	1.1	0.8	0.8	0.7	0.6	0.6	0.4	0.2

Table 2. Standard deviation of the position errors.

Degree	1	2	3	4	5	10	15	20
Non interaction	0.20	0.18	0.16	0.15	0.17	0.16	0.16	0.15
One interaction	0.20	0.17	0.17	0.14	0.15	0.14	0.10	0.09

Table 3. Standard deviation of the relative angle errors.

Given the results in Figure 9, we could recommend interacting a maximum of two times per pair of cameras through all pairs of cameras instead of interacting more than two times in some few pairs of cameras and keeping some pairs of cameras without interaction. We conclude it is better to interact in the pairs of cameras that have the largest pose errors because in these cases the human interaction accentuates the pose errors to decrease. Nevertheless, the ground truth is not available in a practical situation, for this reason, we propose interacting in the pair of images so that the number of point mappings is the lowest one. This information is available at the centre of the Human-Machine Interface (figure 1). This is because, when the number of mapped points decreases, so does the quality of the position and angle estimation.

4.2. Application 1: Robotics positioning

Courtyard database is composed of 32 images taken by two mobile robots (16 images per robot) in a courtyard of Universitat Politècnica de Catalunya. Figure 10 shows two of these images.

Figure 10. Two images in the sequence.

Figure 11 (left) shows the distance between both robots given the 16 shots. In the first nine images, the distance between 8 and 11 metres and from shot ten to thirteen robots get closer until being almost touching. In the last three shots, the robots separate again. Figure 11 (right) shows the error in the robots' position without human interaction and with one human interaction given the 16 shots. In the sixteen shots, the human interactions make to decrees the position error. The position error not only depends on the distance between robots but also on the point of view. In the thirteenth shot, robots are close to each other but they share few salient points. For this reason, the error is larger than the real distance.

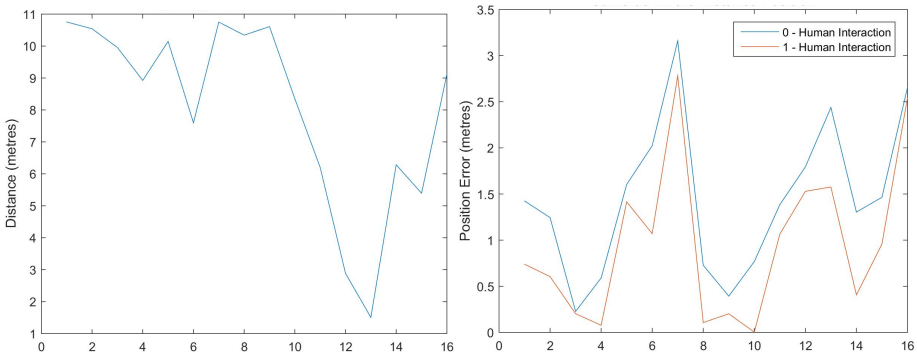


Figure 11. Distance between both robots and position errors throughout the 16 shots.

4.3. Application 2: Camera calibration

Salerno Lab database is a sequence of five images taken from a lab at Salerno University. The aim of this database is to calibrate the cameras to later perform surveillance and human recognition tasks. Figure 12 shows the five images and the real pose of the cameras.

Figure 12. Salerno lab database composed of 5 images and the cameras' poses.

We first tried to deduce the cameras' poses with the five images and without human interaction. In the first round, the method was not able to deduce the homographies of the fifth camera with respect to cameras 1, 2 and 3 due to the large differences between the fifth image and the rest of the images (Figure 13 left shows the direct homographies matrix). Nevertheless, the scheduler realised that the matching modules were able to deduce 45 mappings between the fourth and the fifth image (Figure 13 right shows the matching matrix). For this reason, the scheduler selected cameras 1, 2, 3 and 4 in a second round and selected cameras 4 and 5 in a third round. Then, with the deduced homographies, the cooperative pose estimator was able to deduce the whole homographies. Later, the user imposed a point mapping between the whole pairs of images.

	C1	**C2**	**C3**	**C4**	**C5**
C1	I	H_{12}	H_{13}	-	-
C2	H_{21}	I	H_{23}	-	-
C3	H_{31}	H_{32}	I	H_{34}	-
C4	-	-	H_{43}	I	H_{45}
C5	-	-	-	H_{54}	I

	C1	**C2**	**C3**	**C4**	**C5**
C1	-	75	29	0	0
C2	75	-	66	17	0
C3	29	66	-	64	0
C4	0	17	64	-	45
C5	0	0	0	45	-

Figure 13. Direct homographies matrix and matching matrix.

Figure 14 shows the real pose of the cameras and the deduced poses without interaction and with one human interaction. We can see that the quality of the poses increases with only one human interaction per pair of images.

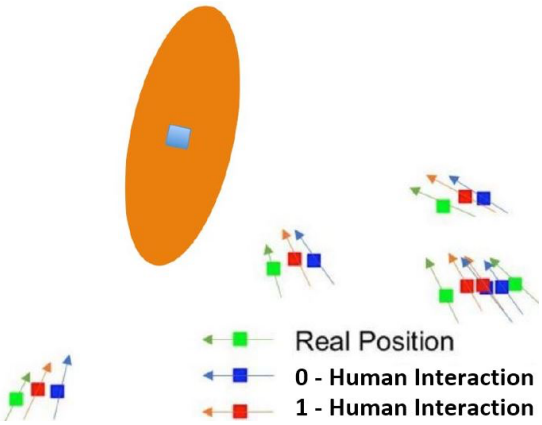


Figure 14. Ground truth and deduced poses (with and with one human interaction).

5. Conclusions and future work

We have presented an interactive and cooperative method to deduce the poses of some cameras given only the 2D images taken by these cameras. It is composed of any algorithm that extracts salient points, aligns them and performs structure from motion to deduce the cameras' poses. Moreover, we have added two new modules. The first makes it possible to deduce the pose of the cameras that do not share any point through a cooperative algorithm. The second adds interactivity to the system allowing the user to impose some point-to-point mappings when one believes it is needed to increase the pose accuracy.

In the experimental section, we have seen that, in some cases, the structure from motion module is unable to deduce the pose of the cameras but when the cooperative method is used, these poses can be estimated. Besides, in the extreme cases that the structure from motion plus the cooperative module is unable to deduce the pose or its error is very high, it is worth letting the human interact by imposing some point-to-point mappings. In these situations, we have seen that it is worth interacting only one mapping in several pairs of images rather than several mappings on few images. In fact, with only one imposition per pair of images, the pose error reduces drastically. When the method is applied to autonomous robots, the human needs a minimum of three seconds to map two points on two images. Thus, the robots do not recover their positions immediately but remain static for a minimum of 4 seconds approximately.

The system is currently being used in a large project, in which social robots guide people through urban areas. Usually, there is a human supervisor that interacts on the fleet of robots and sends them orders. In some cases, robots have to stop since they lose their pose. In these cases, the human interacts by simply mapping pairs of points and the robots recover their pose and therefore they are able to continue their high level task.

As a future work, we want to study in depth the impact of using our method on these high level tasks and how frequent the interaction is needed. Moreover, we will study their behaviour in more difficult frameworks, such as low image quality or several mobile objects interfering in the robots' task.

References

Alahi, A., Ortiz, R., Vandergheynst, P., (2012), FREAK: fast retina keypoint, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., (2008), SURF: speeded up robust features. Computer Vision and Image Understanding (CVIU), Vol. 110 (3), pp. 346—359.

Bianco, S. Ciocca, G. Napoletano, P. Schettini, R. (2015), An interactive tool for manual, semi-automatic and automatic video annotation, Computer Vision and Image Understanding 131, pp: 88–99.

Calonder, M., Lepetit, V., Strecha, C., Fua, P., BRIEF: binary robust independent elementary features, Proc. ECCV 2010 (2010) 778–792.

Casper J.& Murphy RR., (2003), Human–robot interactions during the robot-assisted urban search and rescue response at the world trade centre, IEEE Transactions on Systems, Man, and Cybernetics, Part B. 33, pp: 367–385.

Chui, H., Rangarajan, A., (2003). A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding 89 (2–3), pp: 114–141.

Cortés, X. & Serratosa, F. (2015). An Interactive Method for the Image Alignment problem based on Partially Supervised Correspondence, Expert Systems With Applications 42 (1), pp: 179 - 192.

Cortés, X. & Serratosa, F. (2016). Cooperative Pose Estimation of a Fleet of Robots based on Interactive Points Alignment, Expert Systems With Applications, 45, pp: 150–160.

Fischler, M.A. Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6), pp: 381–395.

Garcia, C., Cena, P. F. Cardenas, R. Saltaren, L. Puglisi, R. Santonja, A (2013). Cooperative multi-agent robotics system: Design and modelling, Expert Systems with Applications, 40, pp: 4737–4748.

Ferrer, G., Garrell, A., Villamizar, M., Huerta, I. & Sanfeliu, A., (2013). Robot Interactive Learning through Human Assistance. Multimodal Interaction in Image and Video Applications, pp: 185-203.

Ferrer, G. & Sanfeliu, A., (2014) Bayesian Human Motion Intentionality Prediction in urban environments, Pattern Recognition Letters 44: 134-140

Garrell, A. Sanfeliu, A. (2012). Cooperative social robots to accompany groups of people, International Journal of Robotics Research, 31(13): 1675-1701.

Gering, D. T. (2010). Systems and methods for interactive image registration, U.S. Patent 7, pp: 693-349.

Gold, S., Rangarajan, A., (1996). A graduated assignment algorithm for graph matching. IEEE Trans. Pattern Anal. Machine Intell. 18 (4).

Harris, C., Stephens, M., (1988), Proceedings of the 4th Alvey Vision Conference. pp: 147–151.

Gui, J., Tao, D., Sun, Z., Luo, Y., You, X. and Tang, Y. (2014), Group sparse multiview patch alignment framework with view consistency for image classification, IEEE Trans. Image Process., 23 (7), pp: 3126–3137.

Huynh, D.Q., (2009). Metrics for 3D Rotations: Comparison and Analysis. Journal of Math Imaging Vision, 35, pp: 155-164.

Ifthekhar, Saha, Jang (2015). Stereo-vision-based cooperative-vehicle positioning using OCC and neural networks, Optics Communications, 352, pp: 166–180.
Jeong. et. al. (2015). A Social Robot to Mitigate Stress, Anxiety, and Pain in Hospital Paediatric Care, Extended Abstracts of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts, pp: 103-104.
Jolliffe, I.T., (2002), Principal component analysis. Second Edition. Springer.
Kashif, M., Deserno, T., Haak, D. and Jonas, S., (2016), Feature description with SIFT, SURF, BRIEF, BRISK, or FREAK? A general question answered for bone age assessment, Computers in Biology and Medicine, 68, pp: 67–75.
Khader, M., Ben Hamza, A, (2012). An information-theoretic method for multimodality medical image registration, Expert Systems with Applications 39 (5), pp: 5548-5556.
Kim, S. Taguchi, S. Hong, S. Lee, H. (2014). Cooperative behavior control of robot group using stress antibody allotment reward, Artificial life and robotics 19 (1), pp: 16-22.
Lia, B. (2015). A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries, Computer Vision and Image Understanding 131, pp: 1–27.
Leutenegger, S., Chli, M., Siegwart, C.R., (2011), BRISK: binary robust invariant scalable keypoints, IEEE Int. Conf. Computer Vision pp: 2548–2555.
Lowe, D.G., (2004), Distinctive image features from scale-invariant keypoints. IJCV 60 (2), pp: 91–110.
Luo, B., Hancock, E., (2003). A unified framework for alignment and correspondence. Computer Vision and Image Understanding 92 (1), pp: 26–55.
Mikolajczyk, K. Schmid, C. (2005). A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10), pp: 1615–1630.
Myronenko, A., Song, X.B., (2010). Point set registration: coherent point drift. IEEE Trans. Pattern Anal. Machine Intell. 32 (12), pp: 2262–2275.
Pfluger, T. (2000). Quantitative Comparison of Automatic and Interactive Methods for MRI-SPECT Image Registration of the Brain Based on 3-Dimensional Calculation of Error, Journal of Nuclear Medicine 41(11), pp: 1823-1829.
Pietrzyk, U. (1994). An Interactive Technique for Three-Dimensional Image Registration, Validation for The Journal of Nuclear Medicine, 35 (12), pp: 2011-2018.
Rangarajan, A., Chui, H., Bookstein, F.L., 1997. The softassign procrustes matching algorithm. In: Proceedings of the 15th International Conference on Information Processing in Medical Imaging. Springer-Verlag, pp: 29–42.
Rosten, E., Reid Porter, R., Drummond, T., (2010), Faster and better: a machine learning approach to corner detection. IEEE Trans. Pattern Analysis and Machine Intelligence 32, pp: 105–119.
Rubio A. (2015). Efficient monocular pose estimation for complex 3D models, International Congress on Robotics and Automation. ICRA2015: 1397-1402.
Sanromà, G. Alquézar, R. Serratosa F. & Herrera, B. (2012). Smooth Point-set Registration using Neighbouring Constraints, Pattern Recognition Letters 33, pp: 2029-2037.
Sanromà, G. Alquézar, R. & Serratosa, F. (2012). A New Graph Matching Method for Point-Set Correspondence using the EM Algorithm and Softassign, Computer Vision and Image Understanding, 116(2), pp: 292-304.
Serratosa, F. (2015a). Speeding up Fast Bipartite Graph Matching trough a new cost matrix, International Journal of Pattern Recognition and Artificial Intelligence, 29 (2).

Serratosa, F. (2015b). Computation of Graph Edit Distance: Reasoning about Optimality and Speed-up, Image and Vision Computing, 40, pp: 38-48.
Serratosa F. & Cortés, X. (2015). Interactive Graph-Matching using Active Query Strategies, Pattern Recognition 48, pp: 1360-1369.
Serratosa, F. (2014). Fast Computation of Bipartite Graph Matching, Pattern Recognition Letters 45, pp: 244 - 250.
Serratosa, F. Alquézar R. & Amézquita, N. (2012). A Probabilistic Integrated Object Recognition and Tracking Framework, Expert Systems With Applications, 39, pp: 7302-7318.
Snavely, N. Todorovic, S. (2011). From contours to 3D object detection and pose estimation, International Congress on Computer Vision.
Solé, A., Serratosa, F. & A. Sanfeliu (2012). On the Graph Edit Distance cost: Properties and Applications, International Journal of Pattern Recognition and Artificial Intelligence 26 (5), 1260004 [21 pages].
Trevai, C., Fukazawa, Y., Yuasa, H., Ota, J., Arai, T. and Asama, H. (2004). Exploration Path Generation for Multiple Mobile Robots Using a Reaction-Diffusion Equation on a Graph, Integrated Computer-Aided Engineering, pp: 1114-1119.
Unhelkar, Shah, (2015). Challenges in Developing a Collaborative Robotic Assistant for Automotive Assembly Lines, Extended Abstracts of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts, pp: 239-240.
Von B., Neitzel, U, (2010), Interactive image registration, WO 2010134013 A1.
Xu, C., Tao, D., and Xu, C., (2015), Multi-View Intact Space Learning, Transactions on Pattern Analysis and Machine Intelligence 37 (12), pp: 2531, 2544.
Yi, G., Jianxin, L., Hangping, Q., Bo, W., (2014). Survey of Structure from Motion, International Conference on Cloud Computing and Internet of Things, pp: 72-76.
Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces, Int. J. Comput. Vision 13 (2), pp: 119–152.