3D Moving Target Reconstruction

From images taken with a stationary camera

POLITECNICO
MILANO 1863

Matteo Frosi, 875393
Master program in Computer Science Engineering

matteo1.frosi@mail.polimi.it

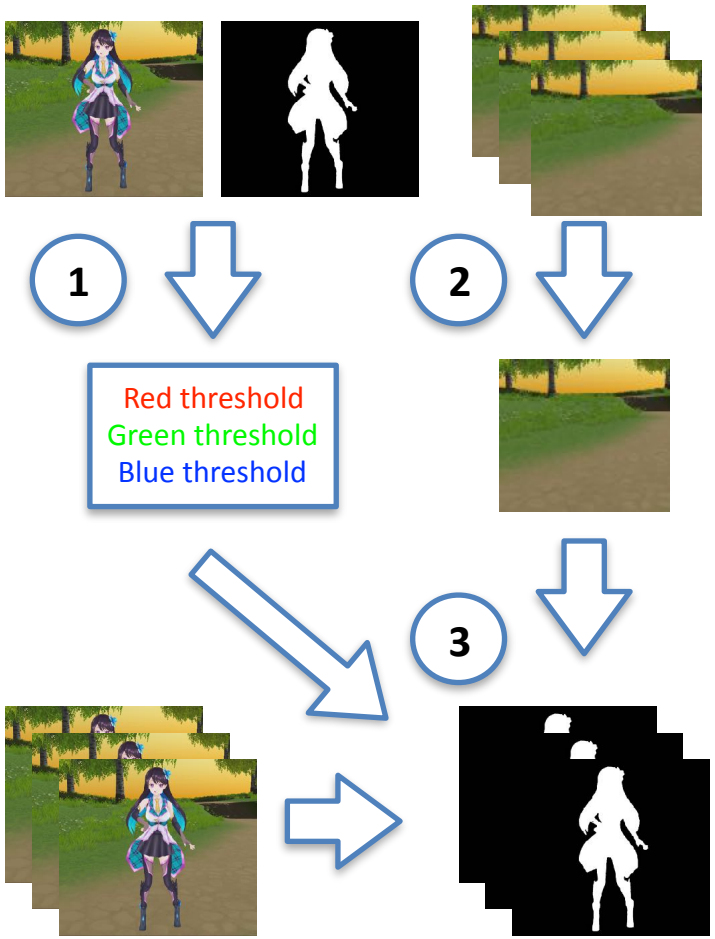
Introduction

The main purpose of this project consists in a volumetric reconstruction (application of texture is not required) of a moving target, not necessarily human. A stationary camera records the target movement and provides a sequence of images describing the scene.

The procedure consists in three consequential phases:

1. Silhouette extraction. We are only interested in the moving target, so we want to identify its silhouette in the image. To do so, we exploit background subtraction techniques, since the camera is stationary and films the same scene for long periods.
2. Pose estimation. Each taken frame of the target correspond to a different position in the scene.
3. Volumetric reconstruction. Using silhouettes and poses, we can project the firsts in a discretised version of the space, composed by small cubes, voxels. The whole projections carve the model of the target.

Silhouette extraction - RGB thresholding with Genetic Algorithm



A ground truth silhouette is taken as model frame and is confronted with one image of the sequence. Exploiting a genetic algorithm we **select the best set of thresholds** to be applied at each color channel of an image to obtain the closest result to the ground truth silhouette [1] . The number of iterations and chromosomes, each composed by three genes corresponding to the three color thresholds, is user defined.

Meanwhile, a simple **model of the background** is computed [2]: the mean of all the images is more than sufficient to describe it.

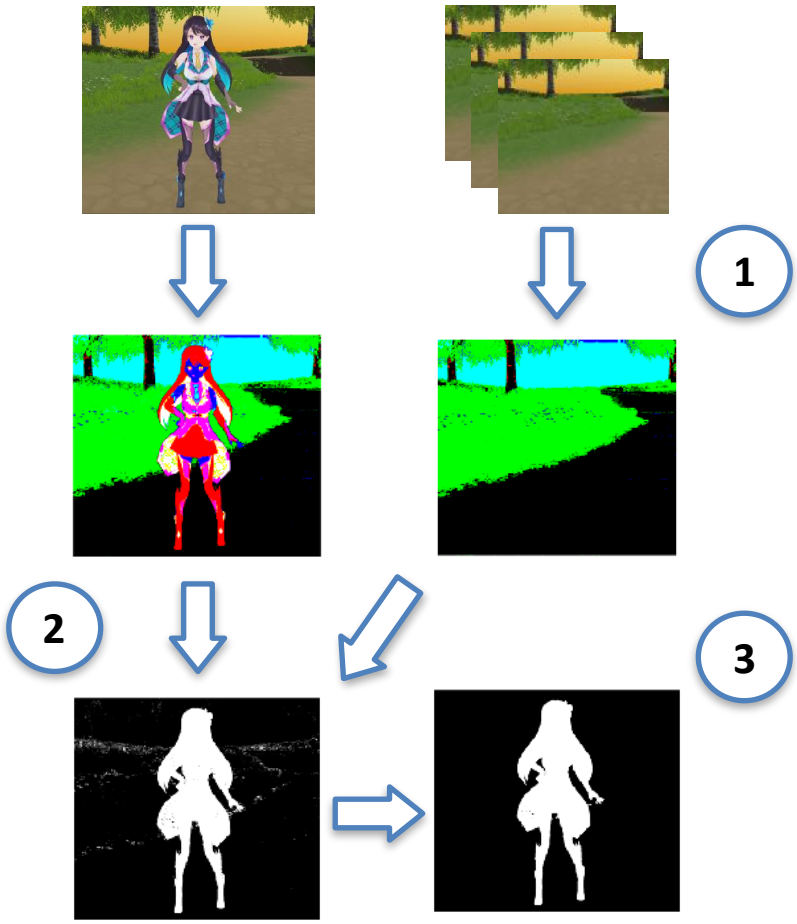
Lastly we adopt a classic technique of background subtraction, the **thresholding** [3]. For each image of the sequence each channel is thresholded, with the computed values, against the corresponding channel of the background model. We obtain then three masks for image, that ,put together, make the silhouette.

Silhouette extraction - HSV thresholding

It has been noticed that image segmentation in different color spaces than RGB provides better results. We use the HSV color space, that is more robust towards external lightning changes. As before, the background is modelled as a simple mean of all its available images, then it is converted in HSV color space [1].

Each frame of the sequence containing the target is first converted, then confronted with the background model, using a bit-wise XOR comparison, giving positive output if two pixels differ [2].

The output is then positive thresholded and smoothed applying a median filter. Lastly, we do denoising, clustering similar regions in the binary image [3]. The last two steps, median filtering and denoising, are done in order to "correct" the errors of the thresholding that has been done without any model of the variance of the background.



Silhouette extraction - Statistic modeling

In the previous slides, we described background modelling as a simple average over all samples. This is a fast technique, but lacks in robustness and does not capture any information about ambience changes, such as variation of illumination or weather factors such as rain or wind.

In the 1999, Horprasert proposed a statistic approach to do background subtraction. First a model of the background is built, considering both brightness and color distortions, then it proceeds to classify each pixel of a target image in four classes: original background, shaded background or shadow, highlighted background or moving foreground object. We are interested only in the moving foreground mask.

Another approach, more efficient than the one above, uses robust statistical descriptors to model the background image, obtaining a noise estimate. Foreground pixels are extracted, and another statistical approach combined with geometrical constraints are adopted to detect and remove shadows.

Silhouette extraction - Adaptive Background Mixture Model

We can model each background pixel by mixture of K Gaussian distributions (usually from 3 to 5). Different Gaussians are assumed to represent different colours.
The weight parameters of the mixture represent the time proportions that those colours stay in the scene. The background components are determined by assuming that the background contains B highest probable colours. These are the ones which stay longer and more static.
Static single-color objects trend to form tight clusters in the color space, while moving ones form widen clusters due to different reflecting surfaces during the movement. To allow the model to adapt to changes in illumination a scheme based upon selective updating is used.

Silhouette extraction - Comparison between techniques

	Advantages	Disadvantages
RGB thresholding with GA	• Fast in finding threshold values • Better results in indoor scenes	• Needs a ground truth image • Inaccurate results if the scene is moving (wind, rain, clouds)
HSV thresholding	• Performs better when the colours of the target differ from the background ones	• Inaccurate results if there is low chromatic variation in the scenes • Based on clustering
Statistic modeling (Horprasert)	• Provides an accurate model of the background	• Needs tuning to avoid inaccuracies caused by almost not existent chromaticity and brightness distortions • Very slow background modelling
Efficient statistic modeling	• Provides an accurate model of the background • Automatic denoising and shadow removal	• Inaccurate results if the scene is moving • Shadow removal is often inaccurate due to brightness variations
Adaptive Background Mixture Model	• Multiple gaussian mixtures give robust results against changing scenes • Very fast background modelling	• Fails when the scene is interested by a strong illumination change (target occluding the source) • Needs to be finely tuned

Pose estimation - Problem reformulation (1)

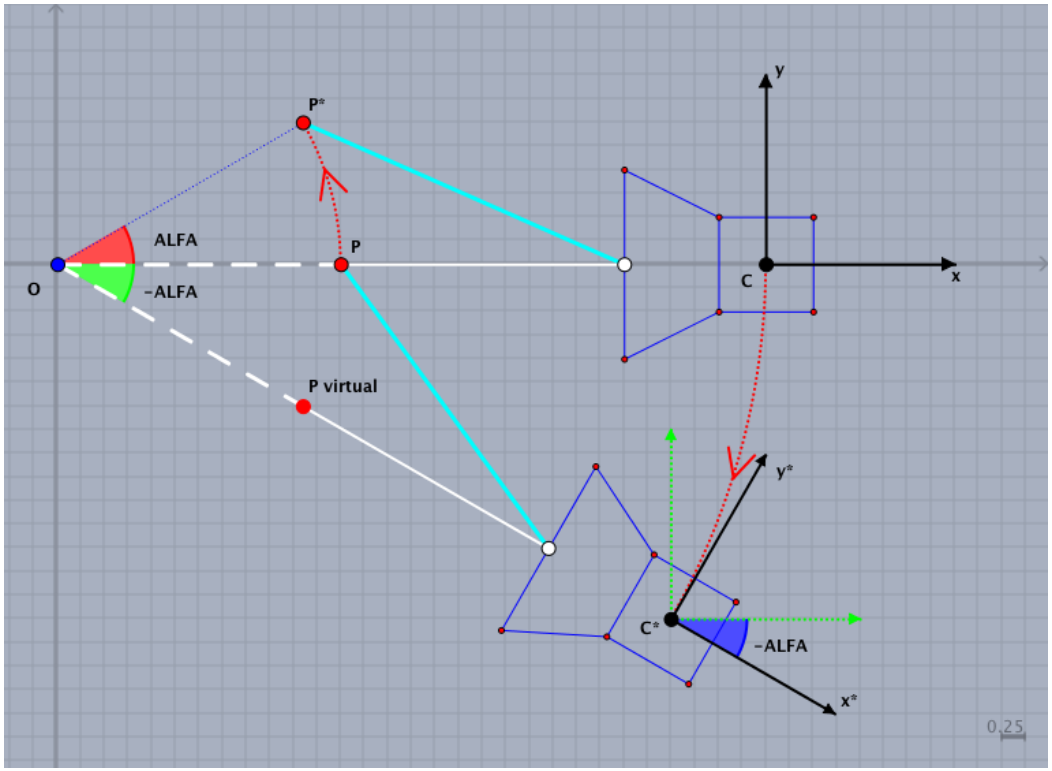
Instead of considering the problem of estimating the pose of the moving target in the scene, we can reformulate it in the camera pose estimation problem.
An intuitive explanation. Suppose we are given a set of images representing a human or an object from different directions and distances, taken by a single camera (this is the only knowledge that we have). It is trivial to see that we are not able to tell if the target or the camera moved in taking such image. In the first figure at the side the camera moved, while in the second the model moved.

Consider an image point u, corresponding to a world point X', obtained rotating around the origin a point X. Let Q be the rotation matrix describing the movement and R_C and C be respectively the rotation matrix and the location in world coordinates of the camera taking the image of point X'. Consider then an image point u', corresponding to world point X, but taken with a camera which rotation matrix and location are transformed by Q^T. Then u and u' are the same image points.

$$u = KR(\widetilde{X}' - C) = KR(Q\widetilde{X} - C) = KR(Q\widetilde{X} - QQ^TC) = K * (RQ) * (\widetilde{X} - Q^TC) = u'$$

$$(RQ) = R_C^{*^T} \Rightarrow R_C^* = Q^T R^T = Q^T R_C$$

Pose estimation - Problem reformulation (2)



The figure on the side gives a simple description of the geometric meaning of what stated before. Point **P** is rotated anticlockwise of an angle **ALFA** and the camera with center in **C** takes its image. Also, the camera rotation is described by **R** and is represented by the black axes **y** and **x**. As stated before, the image of point **P*** taken by such camera, is not distinguishable by the image of point **P** taken by the same camera but which rotation and location of the center are rotated clockwise of the same angle **ALFA** (e. g. rotate anticlockwise by -**ALFA**).

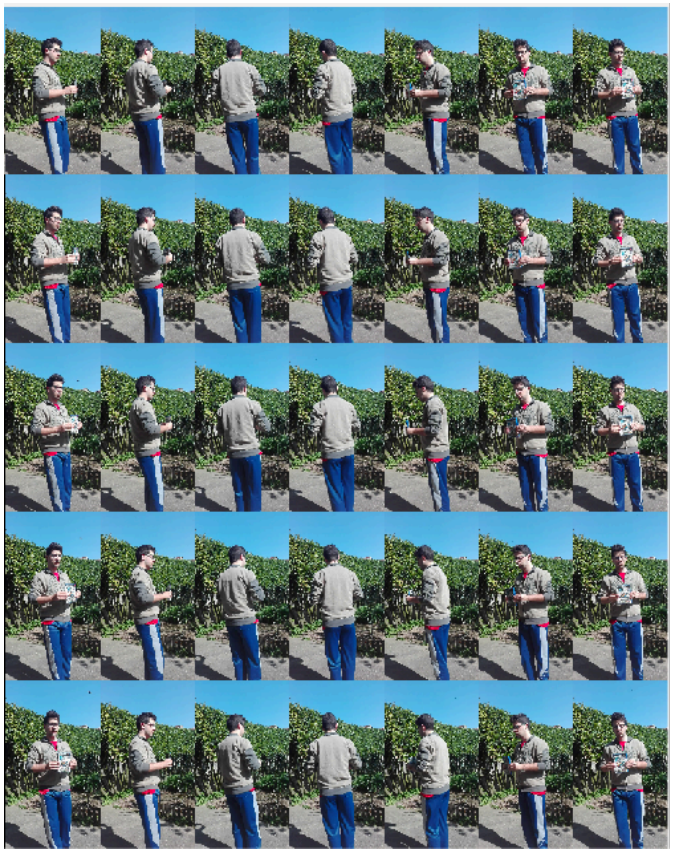
The intuition comes from observing two parts of the figure. Consider the camera mentioned first, points **P** and **P*** and the rotation angle. Consider then the last mentioned camera, points **P virtual** and **P** and the rotation angle. Separately, without any information about the world frame, they represent the same situation. Moreover, it is trivial to see that a simple rotation of the world frame changes the first scenario into the second one.

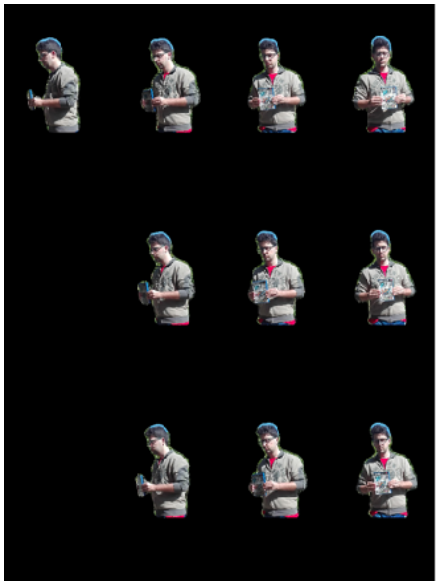
Pose estimation - Sequence preprocessing (1)

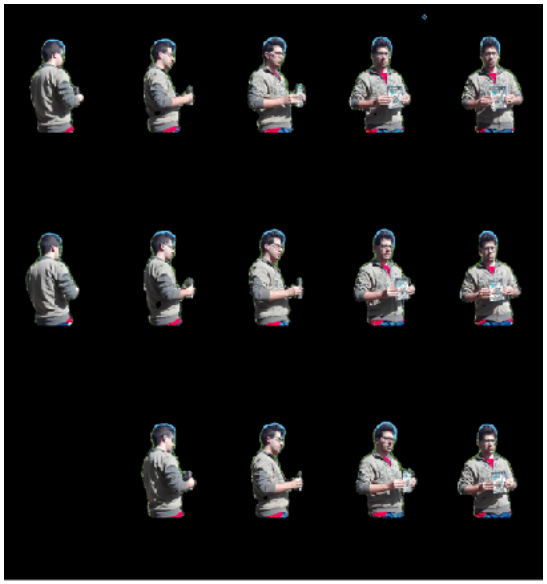
Usually, when dealing with camera pose estimation, we expect to find a huge number of correspondences, since the whole scene is captured from different angles. In other words, feature matching between images consists in both features from the foreground and background. For target pose estimation this is not so: since the background is static with respect to the camera taking images, the background provides no information about the movement of elements in the scene, and only features extracted from the foreground can be used in pose estimation algorithms.
It is correct and intuitive to assume that the number of features extracted from only the foreground (moving target) is much less than the number of features of the whole scene. This leads to incorrect results when dealing with simple matching algorithms.
We perform some preprocessing on the sequence of frames to improve the accuracy and reduce computational time.

1. Sequence ordering. In our project we assume a rotating target, maybe even more than one time around itself, so we want to order the frames w.r.t. the rotation angle.
2. Sequence cleaning. The ordered sequence will probably contain duplicate images, that do not bring any information about the target pose. We remove than by considering colorimetric distances, using the Earth Mover Distance metric (EMD, cross-bin distance), and the average distance between matched features in subsequent images.
3. Frames selection. It may happen that two subsequent images do not share enough matching points to provide an accurate estimation of the camera pose, so we delete them from the sequence.
4. Sequence breaking. To avoid errors accumulating when estimating the camera poses, noticeable when using a lot of images, we break the sequence into two sub-sequences, showing respectively an half-anticlockwise and half-clockwise rotation of the target. The break point, or reference image, is user selected and detected in the sequence exploiting once again the EMD.

Pose estimation - Sequence preprocessing (2)







Silhouette detachment + preprocessing

Pose estimation - Simple matching algorithm

1. Extract SURF features from the first and second frames
2. Match corresponding features
3. Estimate the Essential matrix E_{12} and compute the scale factor between images
4. Extract SURF features from the next frame
5. Match corresponding features with the previous frame
6. Estimate the camera pose exploiting the known poses and triangulated 3D world points
7. If the current frame is a keyframe, do bundle adjustment to optimize poses and triangulated points co-ordinates
8. If the current frame is the last of the sequence STOP, otherwise GOTO 4

Structure from motion from simple matching is very fast, but accumulates error when the number of frames increases, and performs poorly when presented with images with a low amount of features.

Pose estimation - Interrupted KLT tracking algorithm

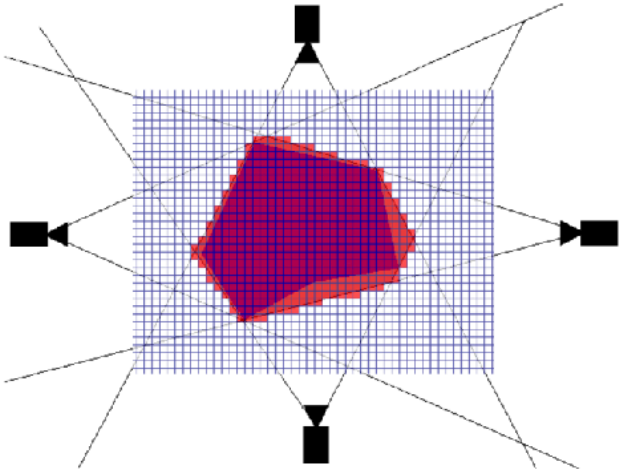
1. Extract SURF features from the first frame
2. Track the previously found features in the second frame
3. Estimate the Essential matrix E_{12} and compute the scale factor between images
4. Extract SURF features from the current frame and add them to the tracked ones
5. Track the previously found features (tracked + extracted) in the next frame, interrupting the old tracking flow
6. Estimate the camera pose exploiting the known poses and triangulated 3D world points
7. If the current frame is a keyframe, do bundle adjustment to optimize poses and triangulated points co-ordinates
8. The next frame becomes the current frame
9. If the current frame is the last of the sequence STOP, otherwise GOTO 4

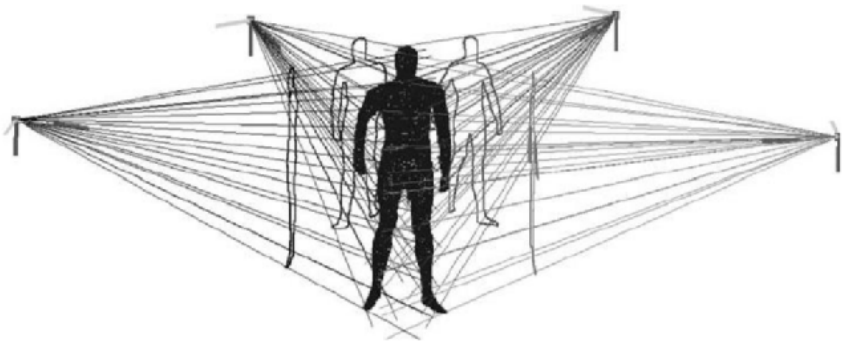
Structure from motion from interrupted tracking is slow compared to the previous method, because the number of features alive at each iteration increases with the frames. For this very reason, it provides also more accurate results, allowing higher confidence when estimating the poses.

Volumetric reconstruction - Voxel carving (1)

Voxel carving is a procedure that consists in the progressive elimination of voxels, starting from a dense 3D discretisation of the space surrounding the target, based on the extracted silhouettes and the corresponding camera poses.

Since we allow some inaccuracy in the silhouette extraction, we model a voxel as a four dimensional object. The first three dimension correspond to the voxel center, while the fourth represents a not normalised fuzzy discrete value for the voxel. The uncertainty of the silhouette extraction is translated into the fact that a voxel does not just belong or not to the target, but it belongs to it with a certain degree of truth.

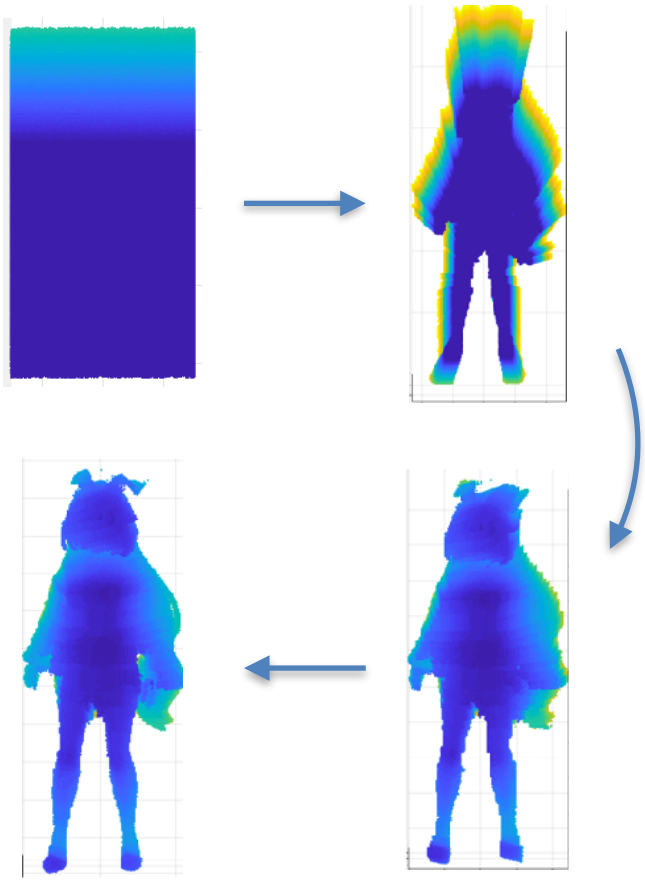




Volumetric reconstruction - Voxel carving (2)

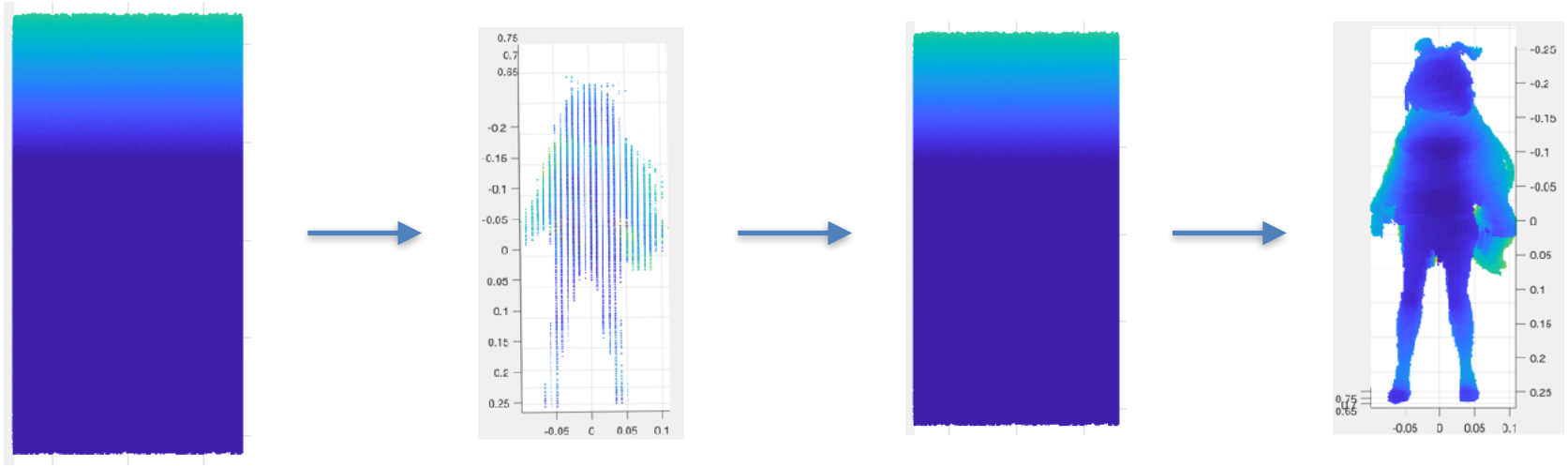
For each available sequence of images, the algorithm works in the following way:

1. Create a grid of voxels which contains the target. It requires the knowledge of the target position in the world with respect to the camera frame. The volume occupied by the voxels influences both the computational time of the algorithm and the precision, in an inverse relation. All the voxels are initialised with a fuzzy value corresponding to the number of images in the whole starting sequence.
2. Consider the first silhouette of the sequence as the current image
3. Project the voxels on the current image using the corresponding camera parameters.
4. If the voxel overlaps the silhouette, it is kept, otherwise its fuzzy value is decremented by 1. If the value is less than a selected threshold, the voxel is discarded.
5. If there is no remaining frame in the sequence STOP, otherwise the current image is the next frame and GOTO 3.

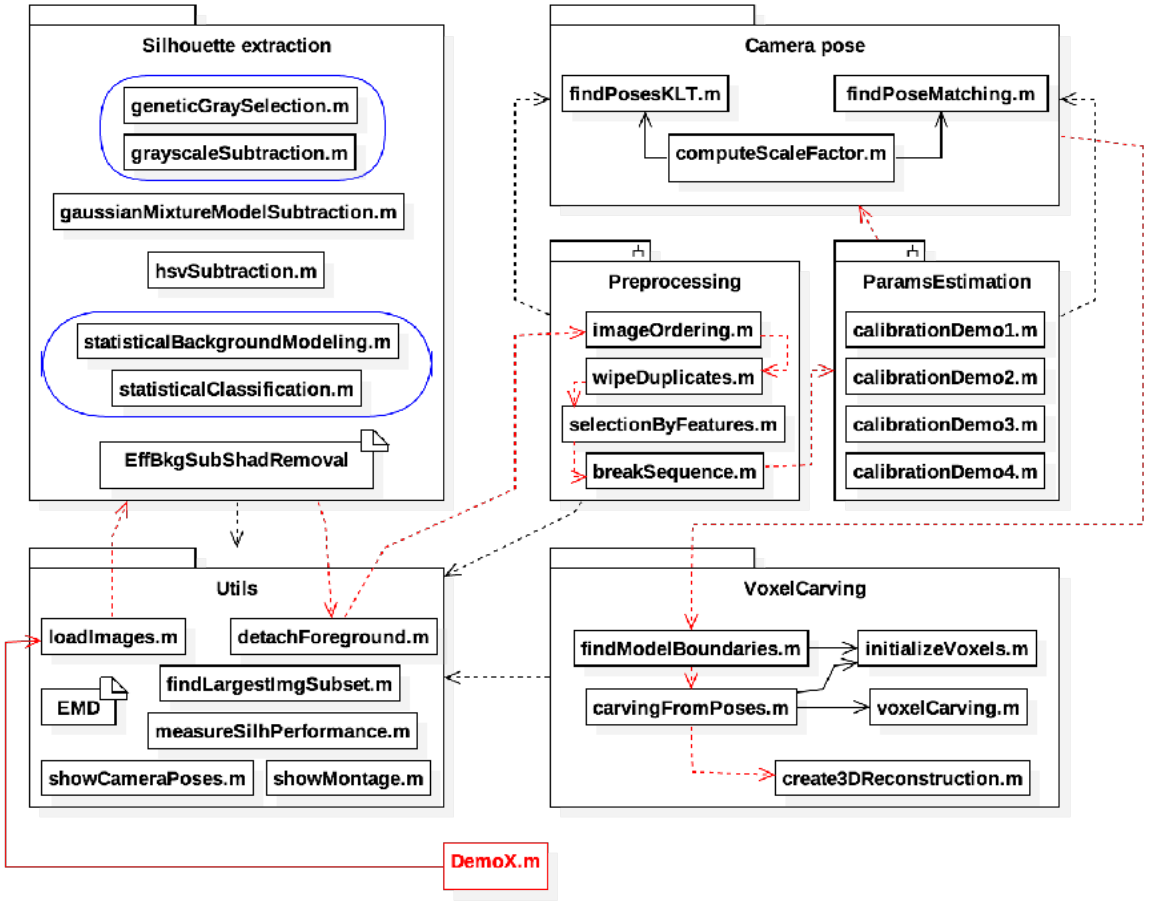


Volumetric reconstruction - Voxel carving (3)

In the first phase of the algorithm we stated that we need the knowledge of the target location in the world, to initialise the voxel grid.
To do this we perform a pre-carving, on a wider space. This volume is created considering the camera positions as the boundaries of this parallelepiped (the height is more than two times the base length) and discretised in 10 million voxels. From the obtained carving, we take the maximum and minimum values for each axes, among all the points of the reconstruction, incrementing them by 10% to avoid errors: they will be the limit values for the dense carving (20 million voxels) described before.



Code architecture overview



The figure represents a high view of the code architecture of the project. The code is grouped in modules, each involving a particular phase described before. In particular we see silhouette extraction, camera pose estimation, voxel carving and utility code as main modules.

Starting from a generic demo file, the red line shows the flow of operation needed to perform volumetric reconstruction.

Results - Demo overview

Each demo presented in the project has its own set of characteristics, including the background type, movement of the target, general conditions and so on. In the following table, we briefly describe such features. Demos are also supposed to increase the difficulty in doing reconstruction.

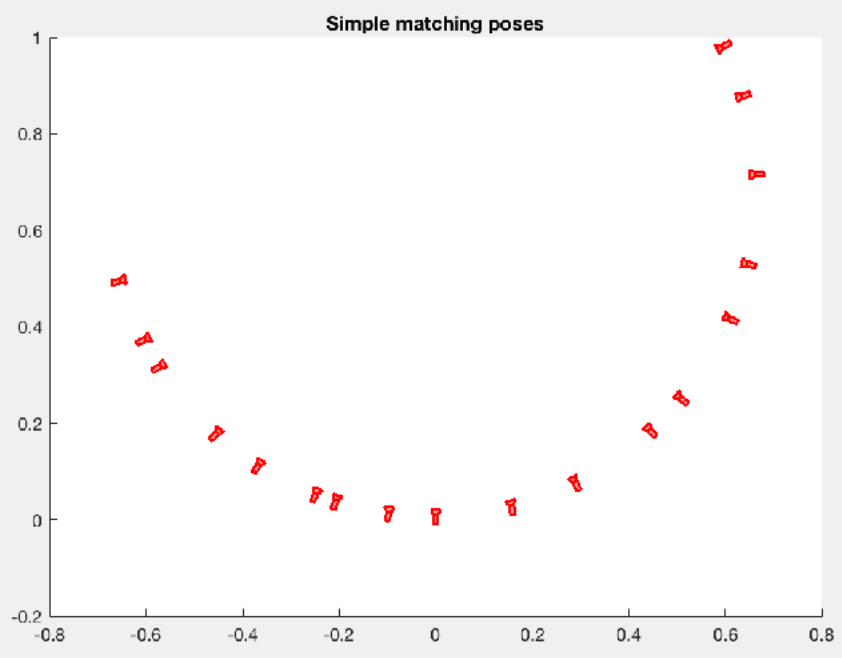
	Positive aspects	Negative aspects
Demo 1	• Almost dichromatic background, with small brightness variations • Rigid target: all body parts move together • Good camera focus	• Low number of features on the sides and back of the target
Demo 2	• Rigid target: all body parts move together • Decent number of features on the sides of the target and high number on the front and back • Good camera focus	• Background subject to environmental conditions: random wind, small illumination changes
Demo 3	• Almost stationary background, with small brightness variations	• Not rigid target: some body parts may move differently • Low number of features on the sides and back of the target • Bad camera focus (images blurred on target edges)
Demo 4	• High number of features • Almost rigid body: only head and torso are considered targets, so it is unlikely that they moved differently w. r. t. each other	• Bad camera focus (images blurred) • Very poor scene brightness

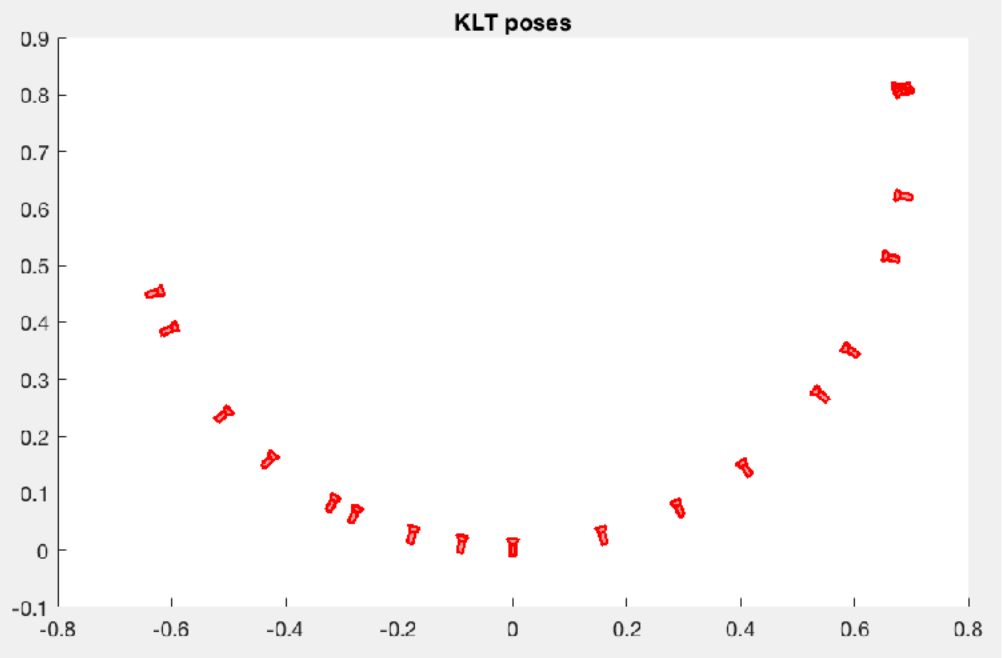
Results - Demo 1 (silhouettes extraction)

Subject: character model Unity-chan
Camera features: Mac OSX screen capture + Unity3D virtual camera
Distance and camera position: about 70 cm from the camera, that is parallel to the ground

	RGB+GA	HSV	Statistical modelling	Efficient statistical modelling	Adaptive Background Mixture Model
Performance (last one considered ground truth)	95,73%	97,21%	99,23%	98,98%	100,00%

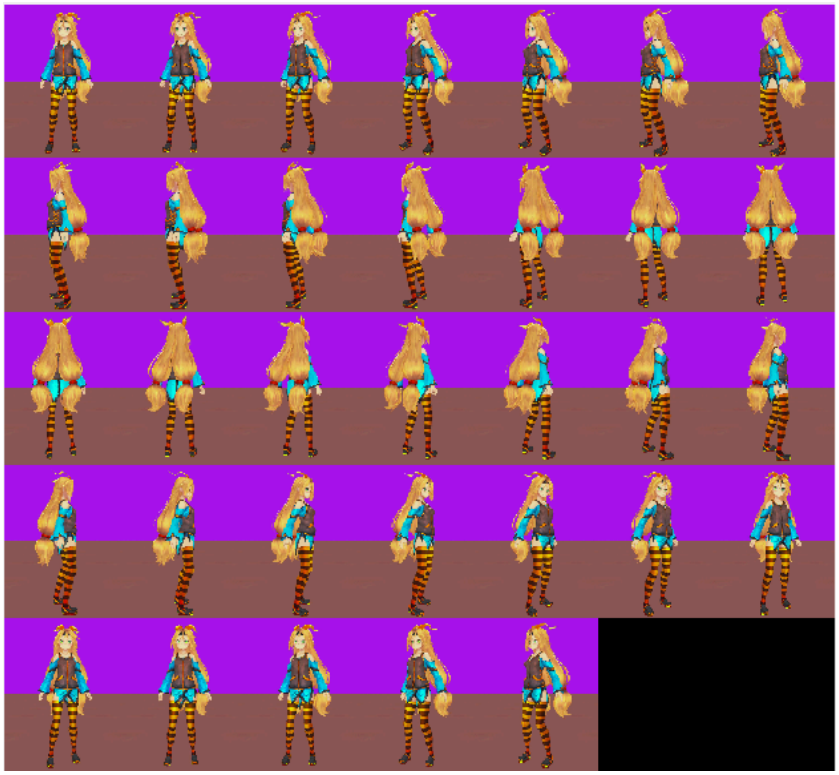
Results - Demo 1 (camera pose estimate)

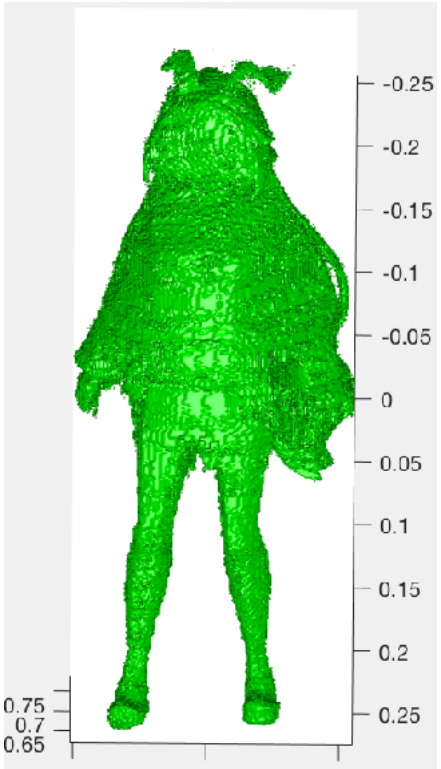


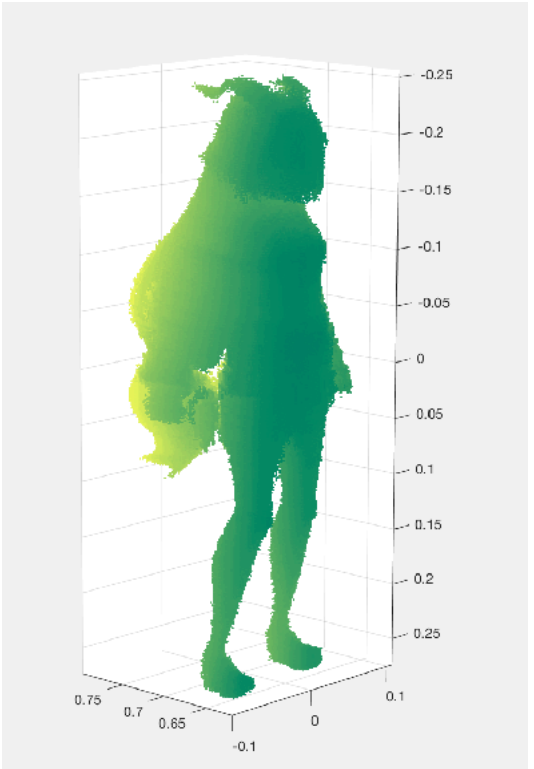


Only a small amount of frames can be used in the pose estimation due to the lack of features. In this demo, both estimation methods provide an accurate results, even if with small difference. When doing voxel carving, we will use the first, since it has a wider range of motion.

Results - Demo 1 (volumetric reconstruction)

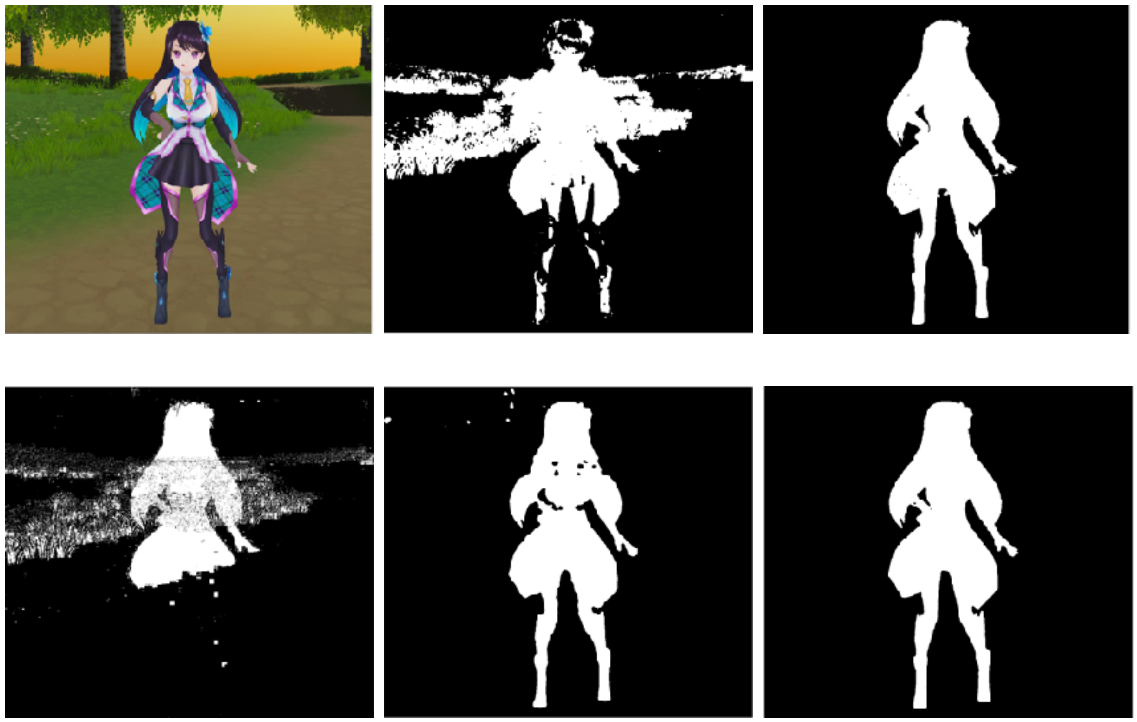






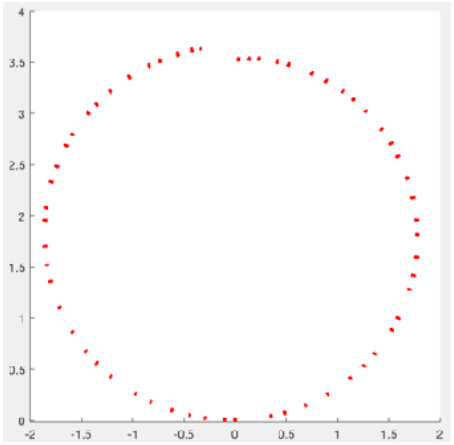
Results - Demo 2 (silhouettes extraction)

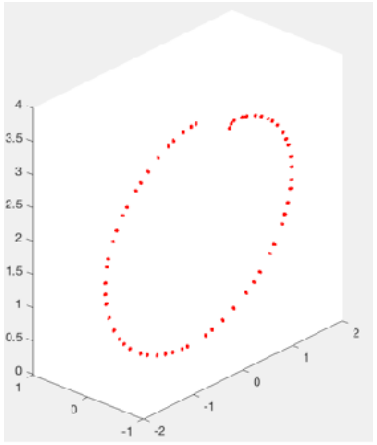
Subject: character model Sapphire-chan
Camera features: Mac OSX screen capture + Unity3D virtual camera
Distance and camera position: about 1.4 m from the camera, that is inclined w.r.t. the ground of 10°

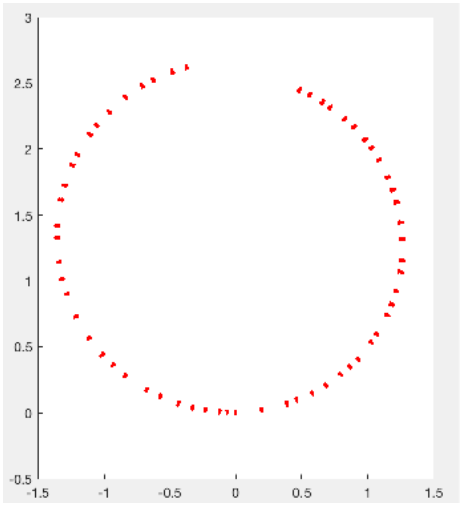


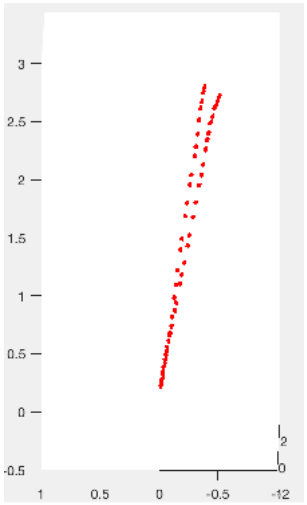
	Performance (last as g. t.)
RGB + GA	95,14%
HSV	99,54%
Statistical	93,48%
Efficient statistical	99,14%
Adaptive	100,00%

Results - Demo 2 (camera pose estimate)







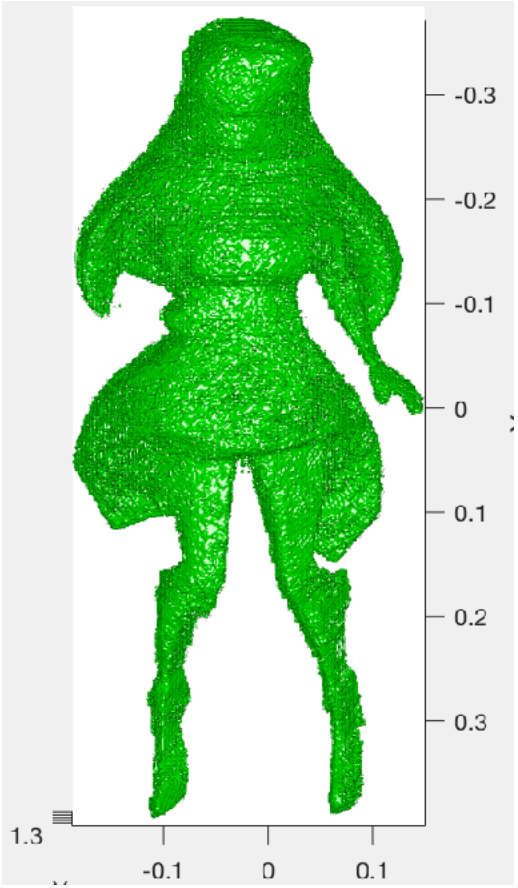


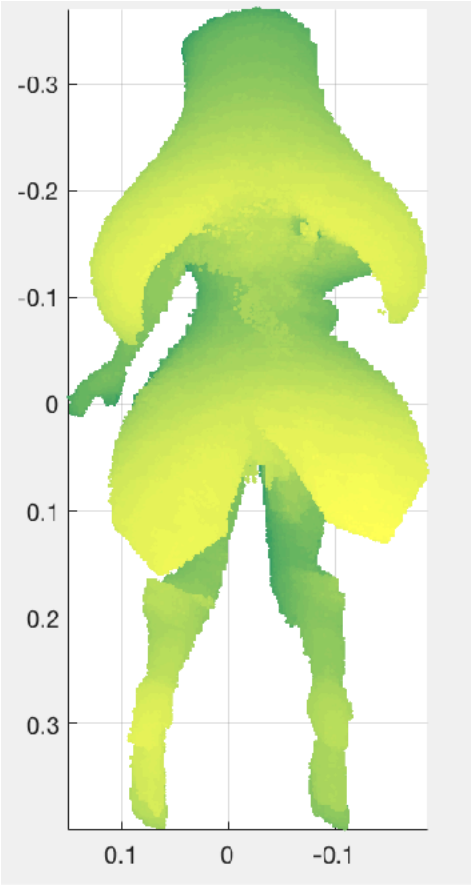
On the top row there are the camera poses estimated with simple matching, wile on the bottom row the ones obtained through KLT tracking.

Both methods correctly estimated the angle of the camera (seen by the oblique cameras position in the space, when the initial camera is considered reference with no rotation and no displacement).

It is noticeable however that the second method results in less displacement and more accurate results, while the first gives an overestimation of the distance and closes, inducing an error, the loop. In fact, the preprocessing leaves an open space within the sequence of images, shown in the KLT pose estimates.

Results - Demo 2 (volumetric reconstruction)





Results - Demo 3 (silhouette extraction)

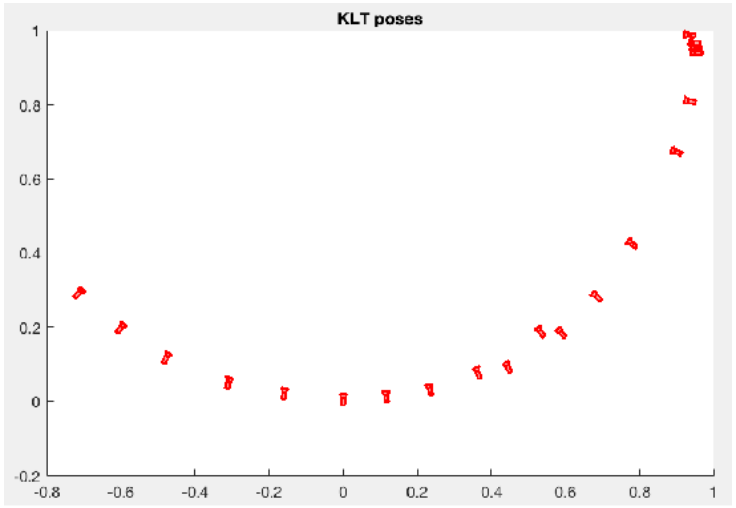
Subject: author of the project, Matteo Frosi
Camera features: Huawei P8 Lite frontal camera, 720x1280 px resolution
Distance and camera position: about 1 m from the camera, inclined w.r.t. the ground

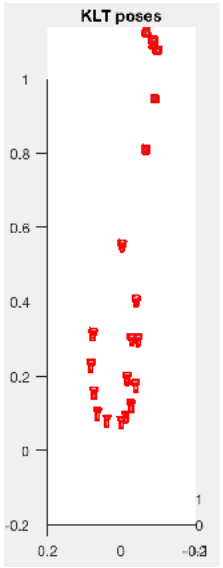


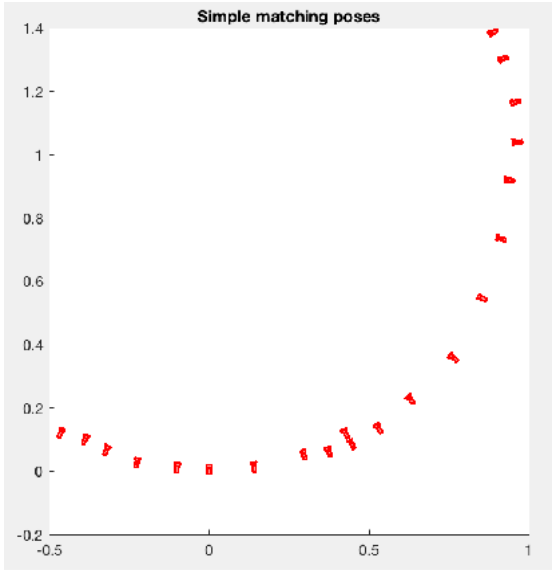


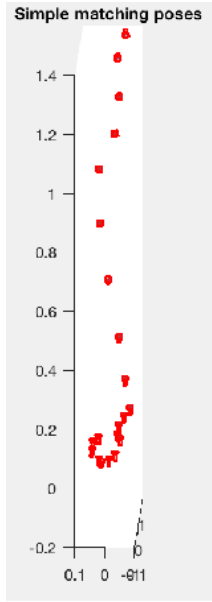

	RGB+GA	HSV	Statistical modelling	Efficient stat. modelling	Adaptive B. M. M.
Performance (last one considered ground truth overestimating)	86,87%	95,91%	85,77%	89,38%	100,00%

Results - Demo 3 (camera pose estimate)



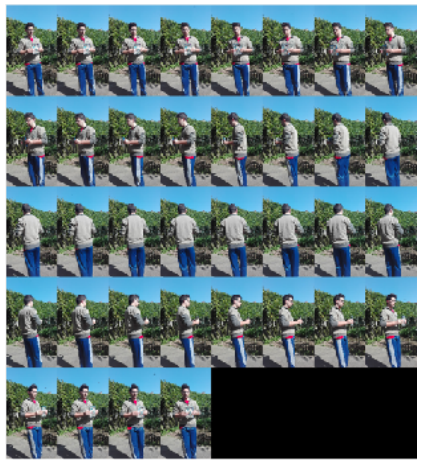


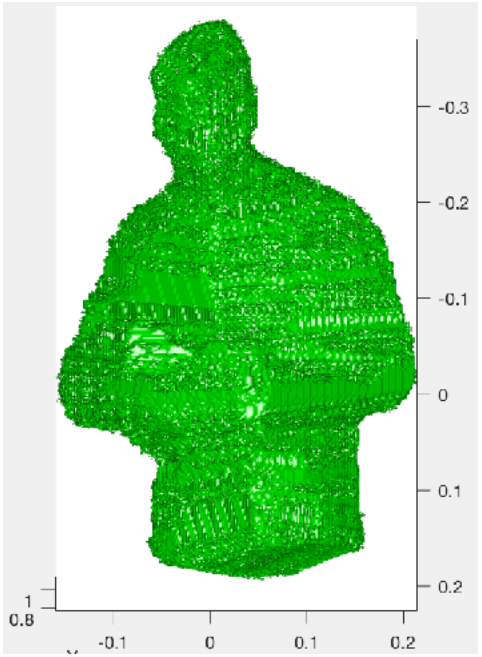


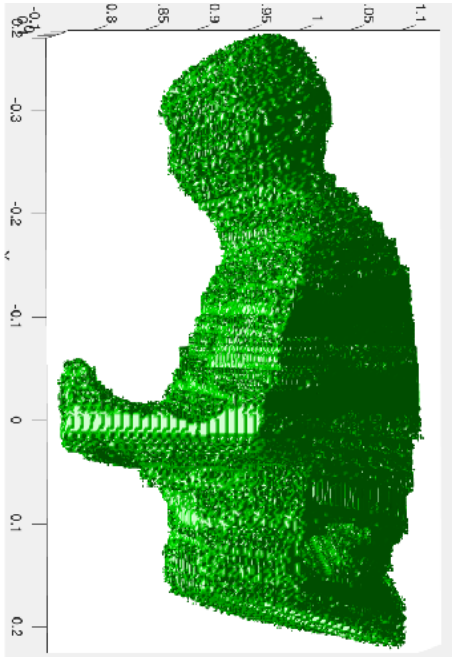


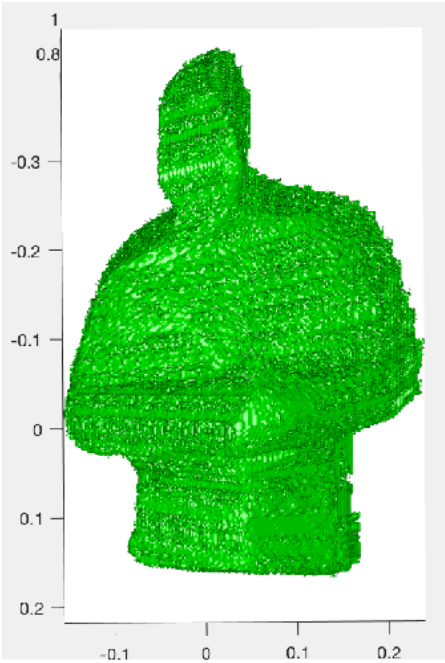
Interrupted KLT provides more accurate results in a short range of motion, describing also the movement of the target while rotating (up and down when moving the feet). Simple matching covers a wider range of motion, but it does so incorrectly, making an overestimation of the poses. In both cases they provide an advantage and a disadvantage in the volumetric reconstruction, as it will be seen in the next slide.

Results - Demo 3 (volumetric reconstruction)







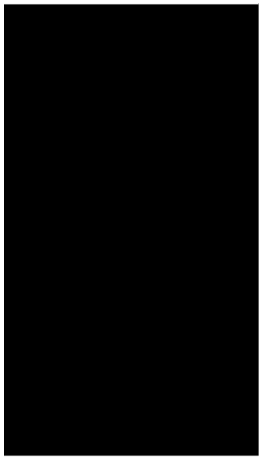


The **left and central reconstructions** are the one done with the poses estimated via tracking, while the **rightmost one** is done with the simple matching poses. The reconstruction by tracking is more smooth and accurate, especially on semi-sharp edges, but it is imperfect since it derives from a small range of poses, not fully carving the space (e. g. around arms or covered parts). The other, gives a rough shape of the model, but it shows the inaccuracy in estimating the poses.
In this demo, we demonstrate that tracking leads to less carving but more accuracy, while simple matching has wider range of motion but has poor precision.

Results - Demo 4 (silhouettes extraction)

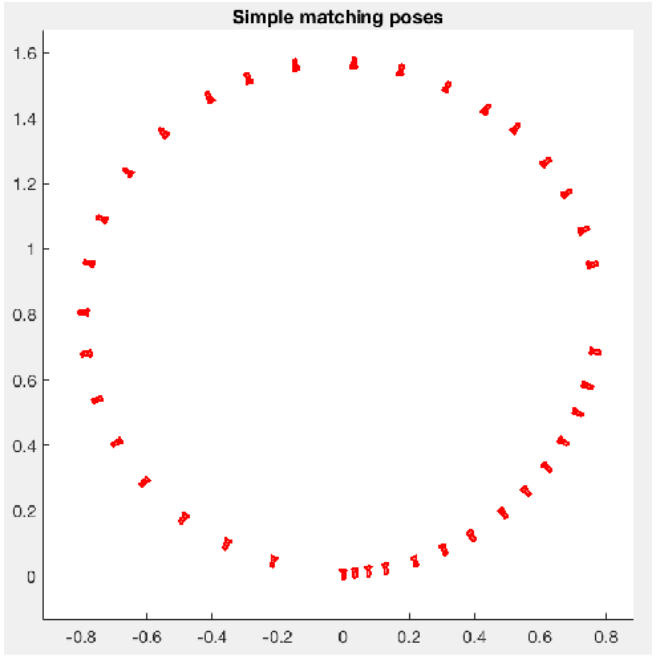
Subject: author of the project, Matteo Frosi
Camera features: Huawei P8 Lite frontal camera, 1088x1980 px resolution
Distance and camera position: about 80 cm from the camera, slightly inclined w.r.t. the ground

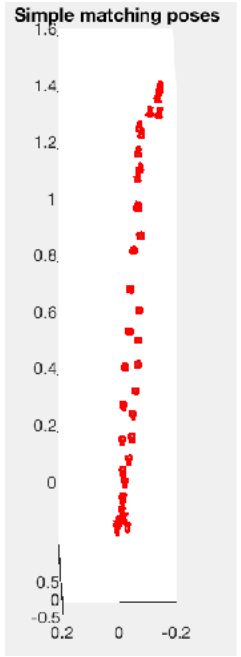


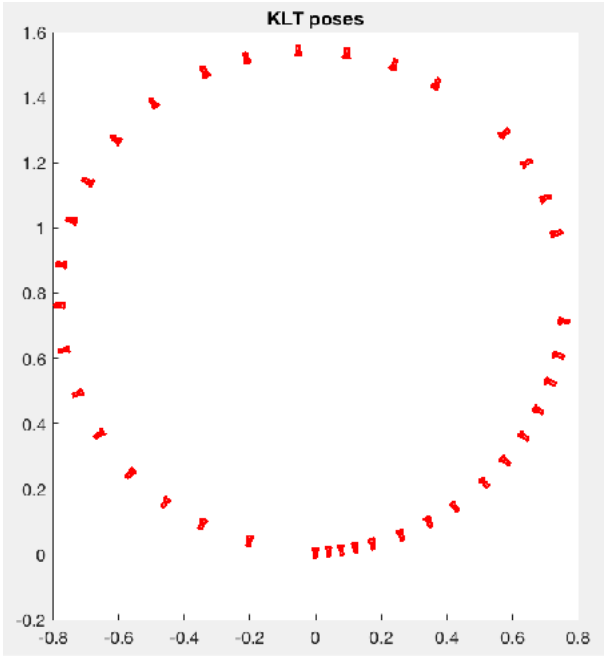



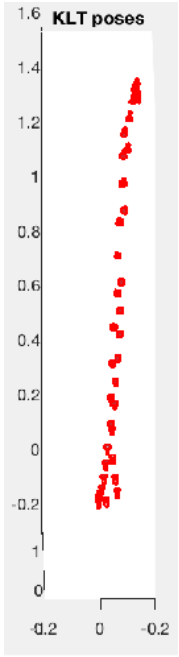
	RGB+GA	HSV	Statistical modelling	Efficient stat. modelling	Adaptive B. M. M.
Performance (last one considered ground truth)	93,97%	94,51%	0%* (difficult tuning)	97,38%	100,00%

Results - Demo 4 (camera pose estimate)



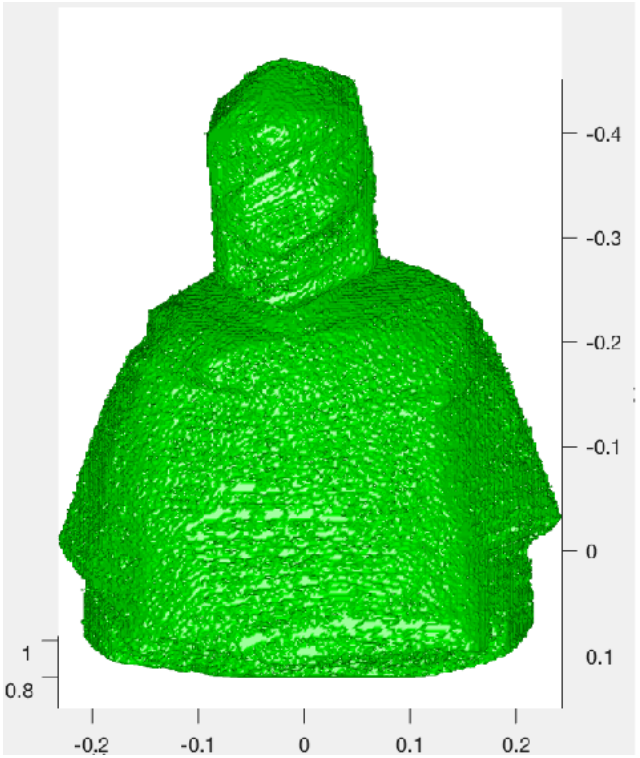


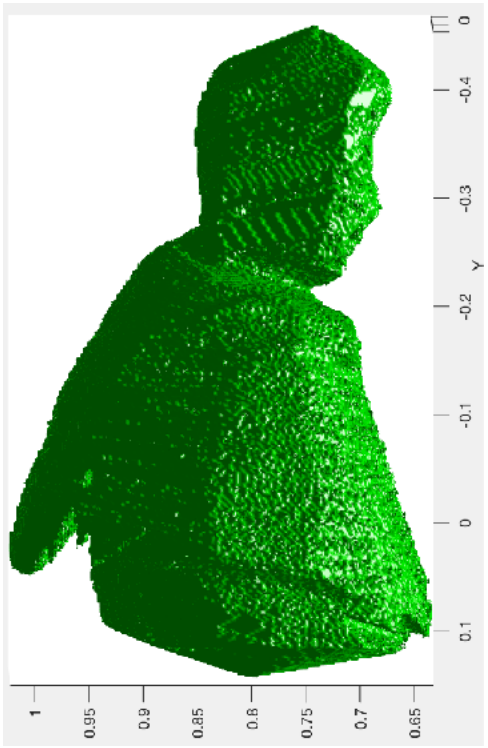




Since in this demo a great number of features is available, both estimation algorithms provide the same results. The up&down behaviour of the poses is due to the small orientation of the camera towards right, visible in the silhouettes. It is noticeable however, that Interrupted KLT captures better this behaviour, making the motion between scenes appear smoother.

Results - Demo 4 (volumetric reconstruction)





The funny setup allows a detailed reconstruction, with a full range of camera poses, however it fails to represent hidden entities, such as the shape of the eyes or nose (covered by the game casings). This problem is intrinsic to voxel carving, as described in the introduction. Notice however that pillows, shirt and flat surfaces are well visible in the reconstruction.

Results - Conclusions

Silhouettes extraction

- HSV thresholding and modelling the background as a mixture of gaussians for each pixel provides the best results in background subtraction, the first slightly worse than the second, but more faster to compute.
- Background modelling as mixture of gaussians performs the best, except some cases. If the source of illumination is placed behind the camera, it may happen that the target occludes the light and changes the background brightness and chromaticity. Since the method relies on training with the background images, the sudden change when the training ends (and the target comes in the scene), makes it fail when classifying the foreground.

Pose estimation

- It is almost unfeasible to pretend that a moving target will maintain its body rigid. This leads to incorrect estimations and a wrong volumetric reconstruction.
- The amount of detected features, and their quality, strongly influences the estimation, leading to poor results, both in accuracy and range of motion.
- Interrupted KLT exploits many features, but it is increasingly slow with the number of analysed frames. Moreover it performs almost as well as simple matching when a lot of features are available, making the last preferable. A solution to the slowness can be a thresholding over the number of features.

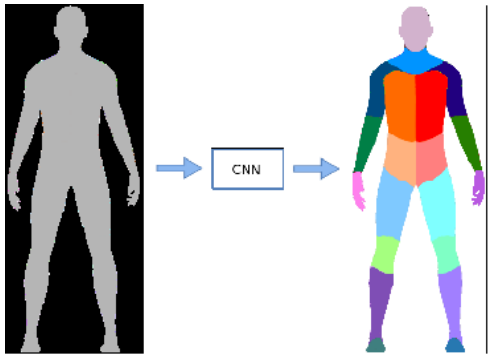
Voxel carving

- It is very easy to implement and work with, and also to manipulate the algorithms related to voxel carving, but the method has problems: it relies on the quality of camera poses and silhouettes and to represent hidden parts requires scenes taken from different angles all around the space surrounding the target (above, sides, below, ...).

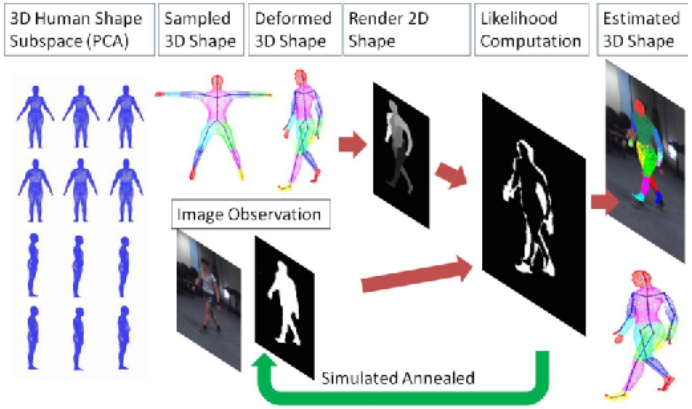
Ideas and proposals

Against non-rigid movements

Instead of performing an overall volumetric reconstruction by voxel carving, we can first break each image of the sequence into small fragments, namely the standard parts of the human body. Then, we perform voxel carving for each part, paying attention to ambiguous situations (e. g. body side image with upper arm overlapping the torso). This procedure may be nice in theory but brings two great disadvantages:



1. Requires a CNN (already available) to classify each image of the sequence, greatly increasing the computational time and resources
2. Since it has sense only for the problem described in this project (static camera – moving target), breaking the foregrounds in small fragments makes even smaller the number of features available for each subset of the sequence, making extremely difficult, if not impossible, to estimate correctly the poses of each chunk of body.



Beyond volumetric reconstruction methods

Instead of forcing volumetric reconstruction over a dense discretisation of the 3D space, we can rely on inferential methods and model based reconstruction techniques. The main advantage w. r. t. voxel carving is that we already start with a prior knowledge of the target shape, that is refined continuously through the sequence of input images. Moreover we have no constraints over the target movement, that can assume every possible position.

Thanks for the attention!

(and maybe texture application can be another improvement)