

Clothes Perception and Manipulation

## D7.2

# First Project Demonstrator and its Validation

Version: 0.7

Last Update: 5 - 2 -2013

Distribution Level: PU (Public)

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement *n*<sup>o</sup> 288553



<b>Project Number</b>	: FP7-288553
<b>Project Title</b>	: Clothes Perception and Manipulation

<b>Deliverable Number</b>	: D7.2
<b>Title of Deliverable</b>	: First Project Demonstrator and its Validation
<b>Nature of Deliverable</b>	: Report
<b>Dissemination Level</b>	: Public
<b>Version</b>	: 0.7
<b>Contractual Delivery Date</b>	: 31 - 1 - 2013
<b>Actual Delivery Date</b>	: 5 - 2 -2013
<b>Contributing WP</b>	: WP7
<b>Author(s)</b>	: Jiri Trdlicka (NEO), Gerardo Aragon-Camarasa, Paul Cockshott, Susanne Oehler, J. Paul Siebert (UG), G. Cannata, S. Denei, M. Jilich, A. Landini, L. Le, P. Maiolino, R. Molfino, D. Zlatanov, M. Zoppi (UNIGE), Vaclav Hlavac, Libor Wagner, Vladimir Smutny, Pavek Krsek, Vladimir Petrik, Martin Meloun (CVUT), Andreas Doumanoglou, Dimitra Triantafyllou, Dimitris Gorpas, Christos Kampouris, Ioannis Mariolis, Sotiris Malassiotis (CERTH)
<b>Reviewer(s)</b>	: Sotiris Malassiotis, Libor Spacek

### Abstract

This deliverable describes the first demonstrator of the CloPeMa project. Its purpose is to present the state of the project and to make a preliminary evaluation of the performance of the individual subsystems. The demonstrator shows the components of the system separately but a section is presented which demonstrates the first integration and interaction of at least some of the components.

### Keywords

Demonstrator, Validation

## Revision History

Version	Description	Author(s)	Date
0.1	First draft	Jiri Trdlicka	8 Jan 2013
0.2	Source code separation, CERTH integration	Jiri Trdlicka	25 Jan 2013
0.3	Integration of the UNIGE contribution	Jiri Trdlicka	29 Jan 2013
0.4	Integration of the CVUT and UG contributions	Jiri Trdlicka	29 Jan 2013
0.5	List of attachments and conclusion added	Jiri Trdlicka	29 Jan 2013
0.55	CVUT minor changes	Jiri Trdlicka	31 Jan 2013
0.6	new cls file, spell check	Jiri Trdlicka	1 Feb 2013
0.7	Internal Review and Corrections	Libor Spacek	5 Feb 2013

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Two Cooperating Manipulators</b>	<b>2</b>
2.1	Description . . . . .	2
2.2	State of the realization . . . . .	3
2.3	Demonstration . . . . .	3
2.4	Performance . . . . .	4
<b>3</b>	<b>CloPeMa Hand and Tactile Sensing</b>	<b>5</b>
3.1	Introduction to the task . . . . .	5
3.2	Tactile sensorization . . . . .	5
3.3	Experimental determination of the grasping and rubbing forces . . . . .	7
3.4	Gripper rationale, concept and motion requirements . . . . .	10
3.5	Gripper architecture . . . . .	11
3.6	Gripper configurations . . . . .	13
3.7	Gripper static modeling . . . . .	17
3.8	Gripper mechanical design . . . . .	17
3.9	Sensorization of the first finger prototypes . . . . .	18
3.10	Further developments of the gripper . . . . .	19
3.11	Demonstration . . . . .	20
3.11.1	Demo cell layout . . . . .	20
3.11.2	Gripper . . . . .	20
3.11.3	Tactile sensors . . . . .	20
3.12	Performance . . . . .	21
3.12.1	Gripper . . . . .	21
3.12.2	Tactile sensors . . . . .	21
<b>4</b>	<b>Binocular Robot Head, Computer Vision and Depth Sensing</b>	<b>22</b>
4.1	Description . . . . .	22
4.1.1	Robot installation . . . . .	23
4.1.2	Parallel matcher . . . . .	23
4.2	State of the realisation . . . . .	23
4.2.1	Robot head Hardware . . . . .	23
4.2.2	Robot Head Software . . . . .	24
4.2.3	C3D in ROS . . . . .	25
4.2.4	Delivered matching algorithm. . . . .	26
4.2.5	Dynamic Calibration . . . . .	27
4.2.6	Parallel matcher . . . . .	29
4.3	Demonstration . . . . .	30
4.3.1	Main interface of the RH . . . . .	31
4.3.2	Examples of range models of clothes . . . . .	31
4.4	Performance . . . . .	32
4.4.1	Parallel matcher . . . . .	33



<b>5</b>	<b>Photometric Stereo System</b>	<b>36</b>
5.1	Description . . . . .	36
5.2	State of the realization . . . . .	36
5.3	Demonstration . . . . .	36
5.4	Performance . . . . .	37
<b>6</b>	<b>World Representation, Perception and Action Planning</b>	<b>38</b>
6.1	Perception . . . . .	38
6.2	Garment recognition application . . . . .	38
6.2.1	Description . . . . .	38
6.2.2	State of realization . . . . .	39
6.2.3	Demonstration . . . . .	39
6.2.4	Performance . . . . .	39
6.3	Features extraction from a hanging garment . . . . .	41
6.3.1	Description . . . . .	41
6.3.2	State of realization . . . . .	41
6.3.3	Demonstration . . . . .	41
6.3.4	Performance . . . . .	41
<b>7</b>	<b>Demonstration of Integrated Motor-Sensory Guidance of the Robot</b>	<b>43</b>
<b>8</b>	<b>Conclusion</b>	<b>45</b>
<b>9</b>	<b>List of Attachments</b>	<b>46</b>

# 1 Introduction

The deliverable D7.2 describes the first demonstrator of the CloPeMa project. It demonstrates the different components of the system, without their interaction and integration. The basic structure of the deliverable follows the Description of Work [1] of the project.

The purpose of the deliverable is to present the state of the project and to make a preliminary evaluation of the performance of the individual subsystems in accordance with the pre-agreed scenario of operation which has been described in the deliverable D1.1 [2].

This document is divided into several sections according to the DoW [1]. Each section consists of a short description of the subsystem, state of its realisation, demonstration of the function, and performance evaluation. Some of the subsystems are described in more detail in special deliverables.

At the end of this document is a section demonstrating the first integration and interaction of some of the components.

## 2 Two Cooperating Manipulators

### 2.1 Description

The basic mechanical and electrical construction of our dual arm manipulator, as well as its software setup, was described in the previous deliverable D7.1 [4]. Our system is built on top of ROS (Robot Operating System). ROS is running on a PC computer which is connected to the robot controller. The connection is facilitated by the module MotoROS and MotoPlus on the software level. The employed hardware and software ensure the basic functionality of the testbed.

The robot kinematics has to be known in order to control the manipulator, detect collisions and perform path planning. The kinematics is a ROS module, which solves the direct and inverse kinematic task. The module written at CVUT is used after extensive experimentation, as described in [5]. The module is able to compute the position of the gripper based on joint positions and/or on proposed joint positions according to the required gripper position. Kinematics is used by the collision detection and planning modules.

The collision detection approach is based on a simplified robot model (Figure 1) and robot kinematics. The collision detection is used for checking all movement instructions before passing them to the robot. When a collision is detected, the instruction is not executed and the module that sent the instruction is informed. This mechanism introduces a low level safety lock, which protects the robot and surrounding equipment. Collision detection is also employed by the planning algorithm.

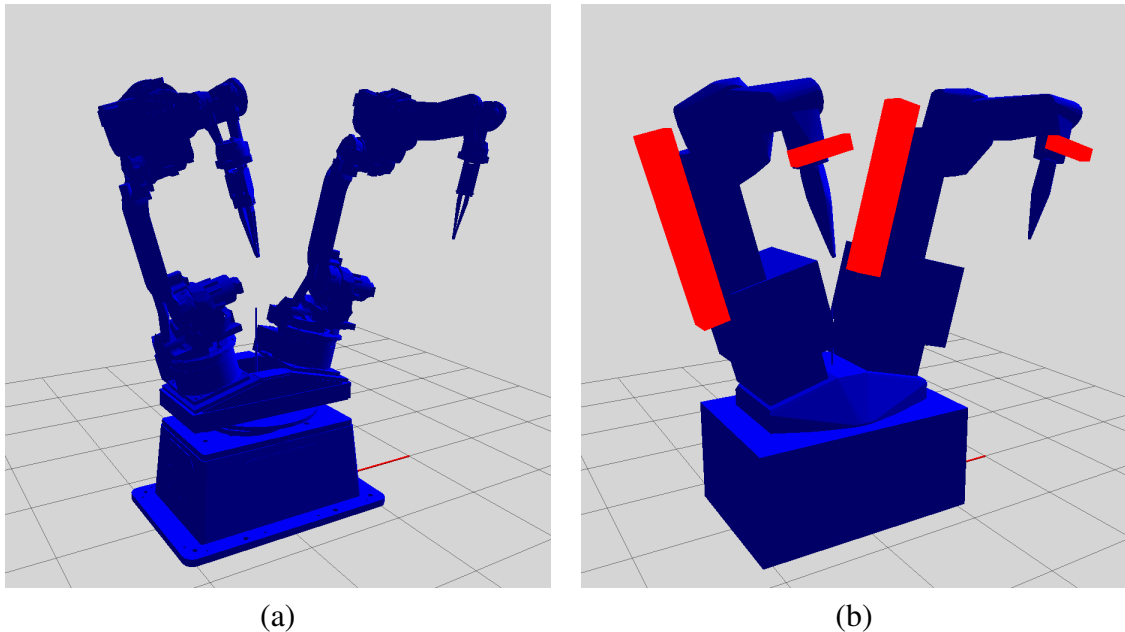


Figure 1: Visualization of detailed robot model (a) and simplified model (b) which is used for collision detection.

The planning algorithm seeks the path which can be followed by the robot. The input to the algorithm consists of the required future position of the robot (both arms and the

turning base). First, the position of the robot is transformed by inverse kinematics into joint coordinates. Next the planning is defined as search for the path in the joint coordinate space, with obstacles defined by collisions. More specifically, we use the planning algorithm based on Rapid-exploring Random Trees from Open Motion Planning Library [10], which is a package for ROS. Planning is extended to take into account the cooperation of both arms while manipulating an object together.

The collision detection and the path planning are performed jointly for the entire robot, i.e. for both arms and the rotating base. Collisions are also checked for the entire robot in each position. More details about collisions handling, planning and calibration can be found in Deliverable 2.1 [5].

## 2.2 State of the realization

We prepared and implemented blocks for robot kinematics, collision detection and path planning. We implemented the algorithm for moving the arms in such a way that the grippers are moved on an equidistant spatial curve. A substantial part of our work in Year 1 was the calibration of the complete system, which is described in Deliverable 2.1 [5].

The testbed is prepared for control of the robot on the middle level, that is where the user or higher-level software subsystems only suggest the goal position of the robot. Movement is planned and safely performed by our software blocks. We continue developing the planning and collision detection methods to enable manipulation of highly flexible objects, such as garments.

## 2.3 Demonstration

We demonstrate the functionality of path planning and collision detection on three short video clips. The timing of clips corresponds to real time. The clips are:

**Plan\_cd.1.avi** – this video shows how the position change is entered by the user in the ROS model visualization window. After that, the trajectory is planned and executed by the robot. During all the steps, the actual position of the robot is shown in the ROS visualization environment.

**Plan\_cd.2.avi** – this video shows the realization of a more complex trajectory. Both arms are moving. One moving arm represents the obstacle for the other moving arm. The goal position is entered manually by the user. The planning process is fully automatic.

**Keep\_dist.1.avi** – this video documents the basic cooperation of the system parts. The movement of the robot is driven automatically by measurements provided by the Asus XtionPRO range finder mounted on the arm. The robot tries to maintain a constant distance between the robot wrist and a moving obstacle (person). The position of the robot arm is planned according to the distance between the robot wrist and the central part of the depth map measured by XtionPRO. The used part of the measurement (the obstacle) is shown as red points in the ROS visualization environment.

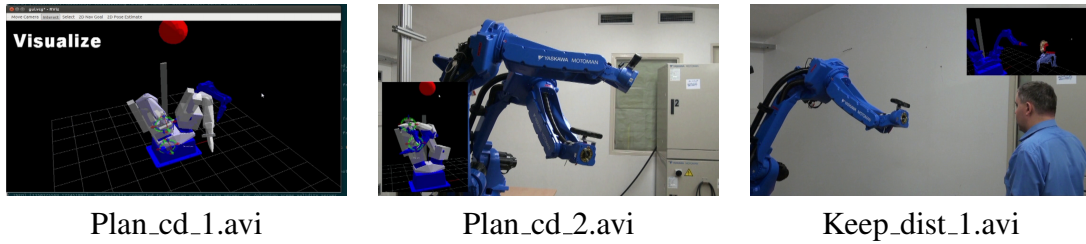


Figure 2: Snapshots from video clips.

## 2.4 Performance

The collision detection has been performed with 100% reliability in our experiments. Results of planning depend on the complexity of planned trajectory. Exceptionally, it can be the case that the planning algorithm does not find the path when the actual and desired position are too distant in joint coordinate space and when the arms obstruct each other. The planning is reliable for short distances between the current and goal positions in a joint coordinate space.

The collision detection algorithm and path planning is realized on the main ROS computer. This computer is equipped with Intel Core i7-2600 CPU running at 3.4 GHz. The collision detection is relatively fast. One position is checked in between 1 and 2 milliseconds. Planning is a more complex task. Its computing time depends on the trajectory complexity. Currently it takes from one to three seconds.

### 3 CloPeMa Hand and Tactile Sensing

The demonstration of CloPeMa hand can be separated in two parts: the gripper itself (also called year-1 hand in the following), and the tactile sensing.

#### 3.1 Introduction to the task

The ClopeMa gripper performs precision grasping (index-thumb finger grasping, [11]) to extract the garment from a heap lying on a hard surface, and it supports the cognition system in classifying the garment through tactile perception.

The visual information available to the handling system is complemented by the tactile information provided by sensors incorporated in the fingers. In order to acquire tactile information, the hands can perform a rubbing motion similar to the one of a human hand assessing the properties of a textile. This motion mimics the human fingers in exploring an object, as described by Klatzky and Lederman [12]. Typical polar pairs of fabric properties can be smooth-rough, soft-stiff, tight-loose. Mapping those properties to garment categories is a critical step for the cognition process: it determines the precision of garment classification by matching the handled piece with the most likely category. The database, organizing key factors for each clothing category, can be prepared in advance during the experiments and training and it is continuously enriched during the operation while the system learns and improves.

During the folding, the garment is picked several times at edges lying over the hard surface of the table. For each such grasp, one finger of the hand must glide close over the surface until it reaches the garment and then it must slide below the edge of the garment lifting it up, while a second finger closes over the first one completing the grasping of the edge. The gliding of the fingertip over the table is made much easier if the attachment of the finger to the palm is compliant.

#### 3.2 Tactile sensorization

The employed tactile sensing technology is described in [8, 13] and it has been developed during the EU Project ROBOSKIN (FP7 2009/2013, grant agreement 231500). The tactile sensor used is modular and suitable to cover large areas of a robot body like an artificial skin. The adopted sensing mode is based on the capacitive transduction principle.

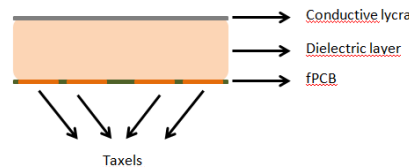


Figure 3: The layered structure of the capacitive tactile sensor.

A capacitive transducer (i.e., a tactile element, or *taxel*) is organized in a layered structure, Fig.3:

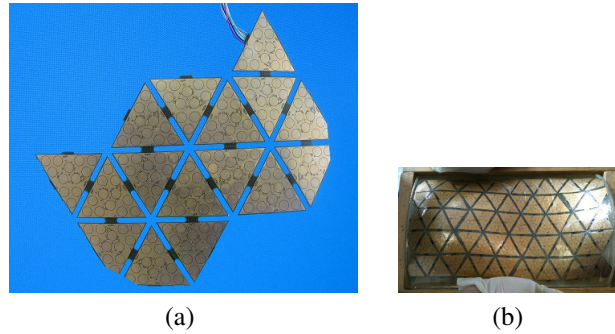


Figure 4: The reference skin system: (a) a skin patch; (b) robot skin conformed to a curved body part.

- The lower layer (which is made of a flexible Printed Circuit Board – fPCB) is divided in a number of interconnected 3 cm side triangular modules (Fig.4a). Each module hosts 12 taxels of 4 mm diameter as well as (on the side not visible in the figure) the read-out electronics for converting capacitance values to 16 bit digital signals (this is accomplished by the small Capacitance to Digital Converter chip AD7147 from Analog Devices, running at 250 kHz, which can measure variations in capacitance within the range 4 to 30 pF with a resolution of 0.32 fF)
- The middle layer consists of the elastomer, which constitutes the compliant dielectric medium for the capacitive sensor. In this work, different candidate elastomers are analysed and compared
- The upper layer is made up of a ground plane, which is used to implement the capacitance. In particular, an electrically conductive Lycra fabric is glued to the elastomer and connected to ground. This configuration allows us to assume the same sensor response when objects with different electrical conductivity are in contact. Furthermore, a reduction of the electronic noise is achieved

When a pressure is applied on the tactile surface, the conductive Lycra layer gets closer to the taxel pads on the fPCB and the distance between the two layers is decreased. The CDC chip of each module measures the variation in capacitance of all the 12 taxels in the module itself. Each measurement is then sent to a central processing unit through a serial  $I^2C$  bus.

The triangular shape is selected because triangular polygons provide a good approximation of a surface. The fPCB triangular modules enforce coverage compliance with respect to robot body parts with varying curvatures (Fig.4b). As a matter of fact, in order to cover robot body parts, the general procedure, explained below, must be followed [13]; during the ROBOSKIN project the tactile sensor technology has been integrated on different robots and in particular on a Schunk robot arm; the pictures that show the integration process are courtesy of Istituto Italiano di Tecnologia, IIT.

The integration procedure involves the following steps:

1. A CAD-like representation of the robot part to be covered (for example like in

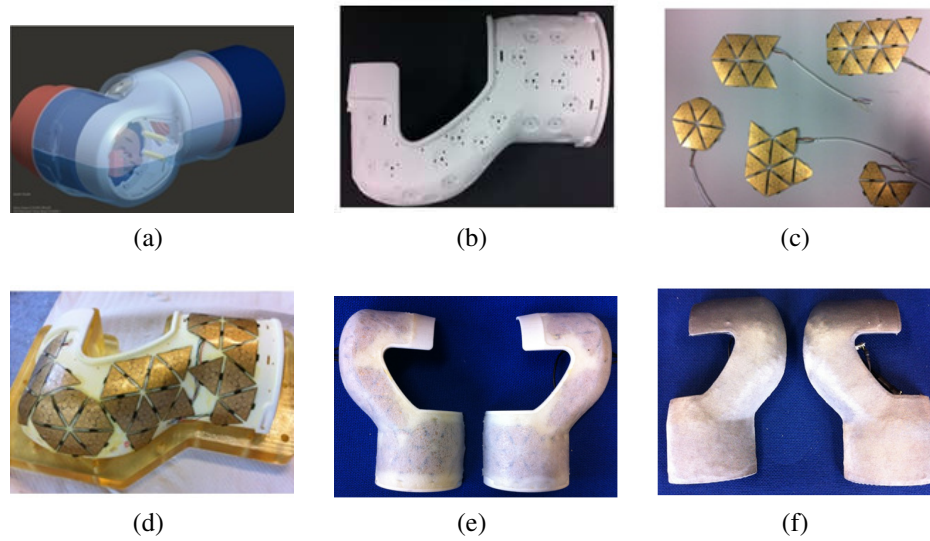


Figure 5: The manufacturing steps for a link of the Schunk robot arm. The pictures presented (courtesy of Istituto Italiano di Tecnologia, IIT) show the integration that has been performed during the EU project ROBOSKIN

- Fig.5a) must be available. If this is not the case, it must be acquired through a 3D laser scanner
2. The robot part (or a suitable cover) is manufactured using a 3D printer, see Fig.5b. Small round holes must be designed to provide space for the CDC chip and the other electronics components which are soldered on the fPCB
  3. The patches of fPCB and the associated  $I^2C$  bus wiring, which are needed to cover the part, must be identified, as shown in Fig.5c
  4. The fPCB patches are bonded to the part using bi-component glue and a vacuum system, see Fig.5d
  5. The fPCB is covered with the elastomer layer, as shown in Fig.5e, where So-maFoama from Smooth-on has been used. To this end, a purpose-built mould is used
  6. The elastomer layer is covered with conductive Lycra as a ground plane, in order for the Lycra to adhere as best as possible to the elastomer, as shown in Fig.5f

The sensorized Schunk robot arm is shown in Figure 6

### 3.3 Experimental determination of the grasping and rubbing forces

Two components of the internal force between the fingertips are of interest for the design of the gripper: the normal force to the fingertip surfaces in contact and the tangential force. We need to know the maximum values of these two forces in two cases: during the





Figure 6: The capacitive tactile technology developed during ROBOSKIN project, integrated on the Schunk arm

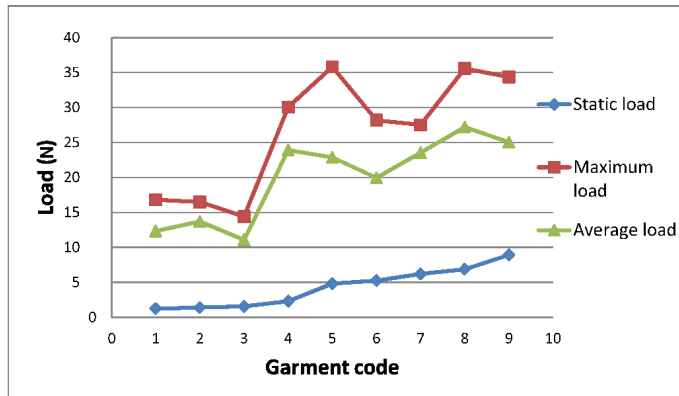
extraction of a cloth from the heap, and during the performance of a rubbing motion. The former is required for the sizing of the grasping actuation; the latter for the development of the rubbing actuation. Both are used to design the links of the gripper. In both cases, the tangential force depends on the normal force and on the friction properties of the materials covering the fingertips; it depends also on the friction properties of the textile grasped.

Consider first the extraction from heap. The maximum of the tangential force depends on the mass of the cloth picked and on the entanglement of the cloth in the heap. The entanglement is generally predominant: the force to extract the cloth depends on the number of items in the heap and on their wrapping; it depends also remarkably on the direction of extraction. Since the modeling of a heap and cloth during the extraction is too complex and would in any case require experiments to verify the results, we have decided to proceed immediately by experiments.

Two sets of experiments were performed: extraction of the heaviest garment (of the selected reference set) from heaps composed of the same number and type of cloths, and extraction of the heaviest garment from heaps composed of different numbers and types of cloths. The results of the first set of experiments are shown in Fig.7 for a cloth of 5 N weight: the cloth is picked 10 times out of a heap of nine pieces; the maximum extraction force measured is 35 N. The second set of experiments illustrates how the size of the heap affects the extraction force of the cloth. It turns out that the extraction force for the same item is roughly proportional to the number of items in the heap, Fig.8.

The normal force necessary to reach 35 N tangential force is 30 N for the materials considered for the fingertips.

The rubbing motion is performed with the garment already extracted from the heap and, either, lying on a surface or differently suspended: the gripper performing the rubbing is not expected to sustain also the weight of the cloth. Consequently, the normal force at the fingertips can be different from the one required for the extraction and it can be adjusted to optimize the sensing of the textile. The best reference is the gesture performed by a human that is assessing a textile with the thumb, the index and medium fingers. The experimental setup developed for the measurement of these forces comprises a glove with the thumb, index and medium fingertips covered by force sensible pads. Gloves with force sensing are present on the market but none is for the range of forces and with the accuracy desired, so we have proceeded to a dedicated development. Capacitive sensors and force-sensitive resistors have been used and the results were compared in order to confirm the correctness of measurements. The fingertips have been covered with the same materials as in the gripper fingers and following the same procedure:



(a)

Code	Garment type
1	short pants
2	T-shirt
3	T-shirt
4	shirt
5	short pants
6	hoodie
7	jeans
8	jeans
9	wool sweater

(b)

Figure 7: (a) Variation of garment load. (b) List of reference set of clothes.

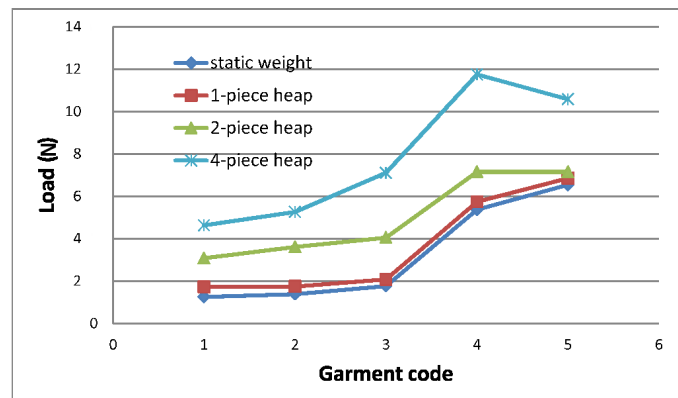


Figure 8: Load for lifting garment from heaps of different sizes.

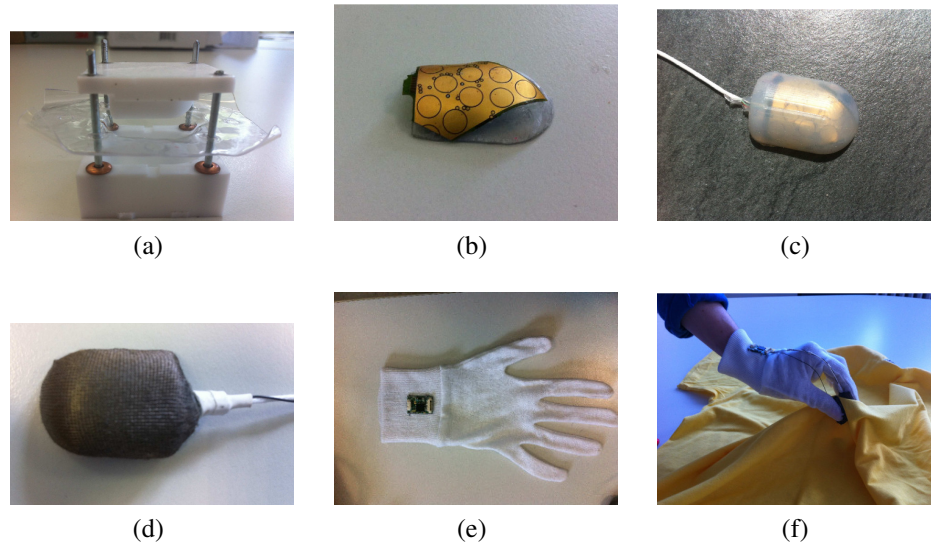


Figure 9: Steps of the manufacturing of the sensorized glove

- a PET-G sheet is thermally formed on a mould to produce the sensor covers (Fig.9a)
- for each fingertip, a triangular module is cut and glued over the cover surface using an epoxy glue (Fig.9b)
- a layer of the silicone elastomer Ecoflex (by Smooth-on) is spread over the PCB board using a dedicated mould (Fig.9c)
- a conductive lycra layer is glued with Sil-poxy glue (by Smooth-on) over the silicone layer (Fig.9d)
- the sensor, so structured, is glued to the fingertip and connected to a dedicated microcontroller board fixed on the back of the glove (Fig. 9e and 9f).

Five subjects have been enrolled for the experiments, three female and two male. Each subject was asked to perform the complete set of gestures, from the picking of a cloth from the heap to its rubbing. Each subject has performed the experiment 10 times over three different cloths: the heaviest of the heap, the most slippery (lowest ratio between tangential and normal force) and the one with higher roughness (highest ratio between tangential and normal force). Incidentally, the roughness of the heaviest cloth is somewhat intermediate between the maximum and the minimum of the cloths in the reference heap.

### 3.4 Gripper rationale, concept and motion requirements

The industrial grippers available on the market for use as robot peripherals are designed for rigid items and they generally provide high grasping force and short strokes; as they are, they cannot be used with textiles. On the other hand, industrial grippers have the advantages to be robust, very cost-effective, and easy to mount on any industrial arm. The

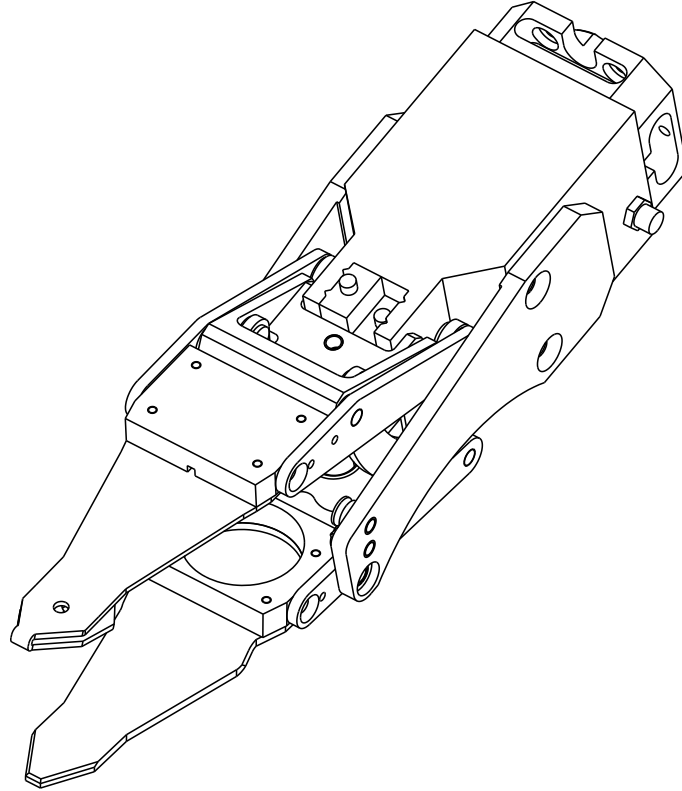


Figure 10: Assembled view of the gripper (4R in neutral configuration; 3R in open configuration)

CloPeMa gripper is designed as an elaboration of an industrial gripper: it uses the translation of the fingers of the industrial gripper as actuation and the frame of the industrial gripper serves also as a base frame of the new gripper.

Summarizing, the required motions are: (1) a motion of closure to perform the grasping; (2) a motion of rubbing, with a portion of a finger surface in relative motion with respect to the garment grasped; (3) a compliant motion to accommodate for contacts between the bottom finger and the table when the gripper picks a cloth from a table. The effective architecture selected to have these three motions uses a 4-bar linkage: the motion of the finger blocks of the industrial gripper relocates one of the base revolute joints of the 4-bar and consequently the distance between a pair of points located on the two links used as fingers increases or decreases.

### 3.5 Gripper architecture

The gripper comprises a driver, a flange to connect the driver body to the flange of a robot arm, a mechanical assembly called fingers assembly, two mounting plates, Fig.10 and 11.

The driver comprises a body (numbered 11 in Fig.11) and two driving blocks (numbered 12 and 13) translating in opposite directions along the same guide. The motions of the blocks are mirrored with respect to the mid plane of the gripper body: if one block shifts by a certain amount in a direction, the other block shifts an identical amount in the

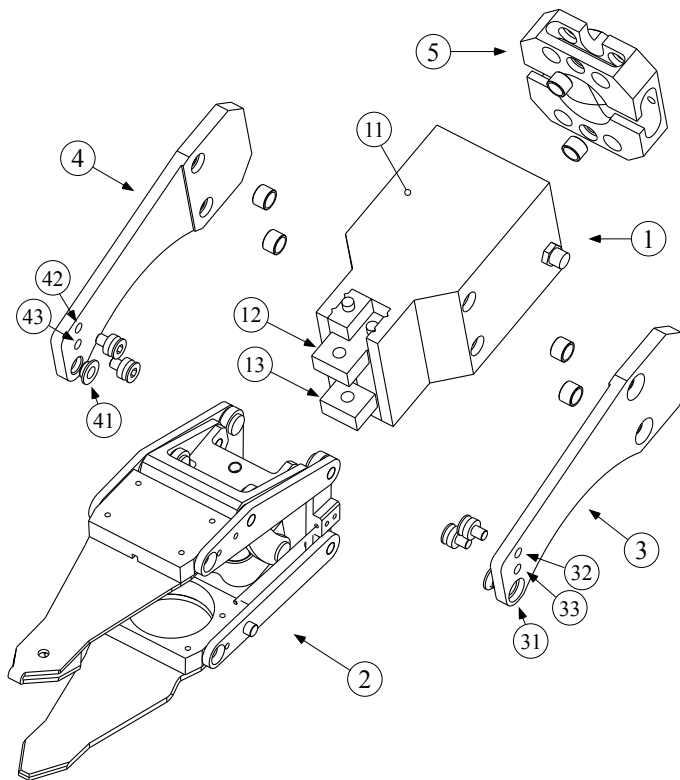


Figure 11: Exploded view of the gripper with driver (1), left (3) and right (4) mounting plates, arm mounting flange (5), fingers assembly (2) (4R in neutral configuration; 3R in open configuration)

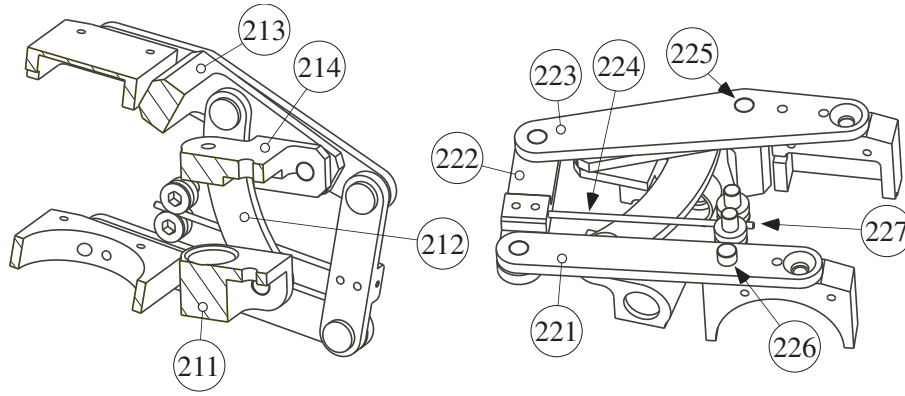


Figure 12: Detailed views of the 3R and 4R sub-mechanisms of the fingers assembly

opposite direction. This is the typical motion of the industrial robotic grippers with parallel motion. The driving blocks stick out of the driver body. Each driving block presents a fixing hole and surfaces in geometric and dimensional tolerance for the alignment of the driven body.

Consider Fig.11. The mounting plates are fixed to the two lateral sides of the driver using alignment pins. Each plate presents, at the free extremity, a hole for the pin of a revolute joint (numbered 31 and 41) and two threaded holes used to retain an extremity of a bending spring (numbered 32, 33 and 42, 43).

The fingers assembly comprises a sub-mechanism with three revolute joints (called 3R) and a sub-mechanism with four revolute joints (called 4R).

Consider Fig.12. The 3R sub-mechanism comprises four bodies (numbered 211, 212, 213, 214). When the fingers assembly is mounted to the driver, the two extremal bodies (211, 214) of the 3R sub-mechanism, visible also in Fig.13, are fixed to the driving blocks of the driver and form with the driver body a 5-bar linkage of type PRRRP. One of the two links with two revolute joints of the 3R sub-mechanism (number 213) is connected to the 4R sub-mechanism by a revolute joint (numbered 225): we call this link the output link of the 3R sub-mechanism (and of the 5-bar linkage when considering the whole gripper).

Consider Fig.12. The 4R sub-mechanism comprises three bodies (number 221, 222, 223). The first (221) and third (223) body are the fingers of the gripper. The first body (221) is connected by two coaxial revolute joints (226) to the two mounting plates of the gripper; it rotates about an axis fixed to the body of the driver. The third body (223) of the 4R sub-mechanism is connected by two coaxial revolute joints (225) to the output link (213) of the 3R sub-mechanism. When the 3R sub-mechanism is locked in any feasible configuration, the 4R sub-mechanism is equivalent to a 4-bar linkage whose fourth body is the ensemble of the driver body with mounting plates, driving blocks and 3R sub-mechanism.

Further details about the design and assembly of the bodies and joints of the 3R and 4R are provided in the exploded views in Fig.14.

### 3.6 Gripper configurations

Based on the task, we distinguish different sets of configurations of the gripper.

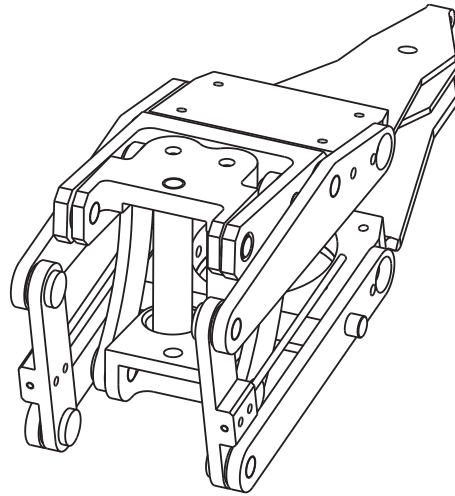


Figure 13: Back view of the fingers assembly with the connections to the driver visible

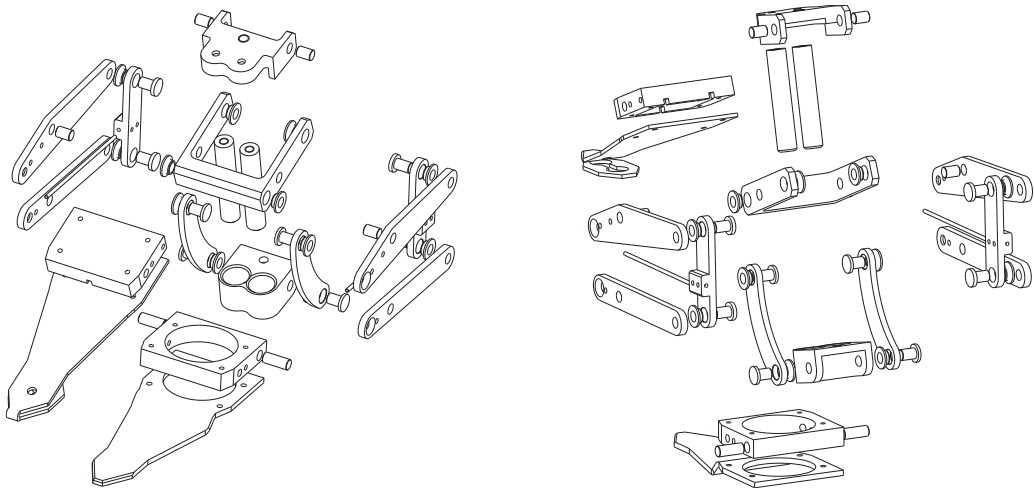


Figure 14: Exploded views of the fingers assembly without the mounting plates and the driver

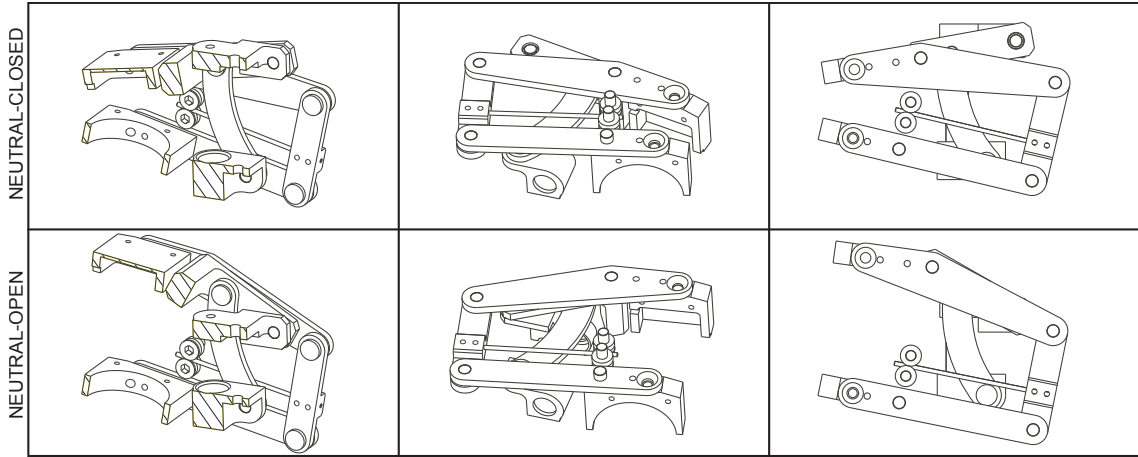


Figure 15: Sections and side views of the fingers assembly mechanism without finger plates (4-bar in neutral configuration; 5-bar in open and closed configuration)

We call *closed* every gripper configuration with the 5-bar linkage in the configuration with minimum distance between the driving blocks. In this configuration, the distance between the fingertips is the minimum for every angle of rotation of the 4R sub-mechanism. Similarly, we call *open* every gripper configuration with the 5-bar linkage in the configuration with maximum distance between the driving blocks. In this configuration, the distance between the fingertips is the maximum possible. Figure 15 shows in detail the positions of the links in the fingers assembly in *closed* and *open* configurations.

There is an intermediate position of the driving blocks, between the *closed* and *open* configurations, such that the four revolute joints of the 4R sub-mechanism of the fingers assembly are along the edges of a right prism with parallelogram base. We call this configuration of the gripper *parallel*. In a *parallel* configuration, the 4-bar mechanism formed by the 4R sub-mechanism with the 5-bar mechanism locked is a parallelogram and the rubbing motion of the finger tips can be performed with the opposite faces of the fingertips (in contact with the material grasped) translating one with respect to the other one.

For any configuration of the 5-bar linkage between the *closed* and *open*, there is an angle of the 4R sub-mechanism of the fingers assembly such that the bending springs connecting the 4R to the mounting plates are undeformed. We call this configuration *neutral*. When no external load is applied to the fingers, the 4R rests at the *neutral* configuration.

The drawings in Figures 16 and 17 show the different sub-mechanisms of the gripper in *closed*, *parallel*, and *open* configuration.

The design of the links of the fingers assembly limits the angle that the 4R sub-mechanism can span without mechanical interferences. We call *up* and *down* configurations the configurations of the 4R sub-mechanism corresponding to the extreme angles that it can reach for a given configuration of the 5-bar linkage (between *closed* and *open*). The neutral configuration is somewhere midway between the *up* and *down* configurations. The drawings in Fig. 18 show the combinations of *up* and *down* configurations with the *closed*, *parallel*, and *open* configurations.



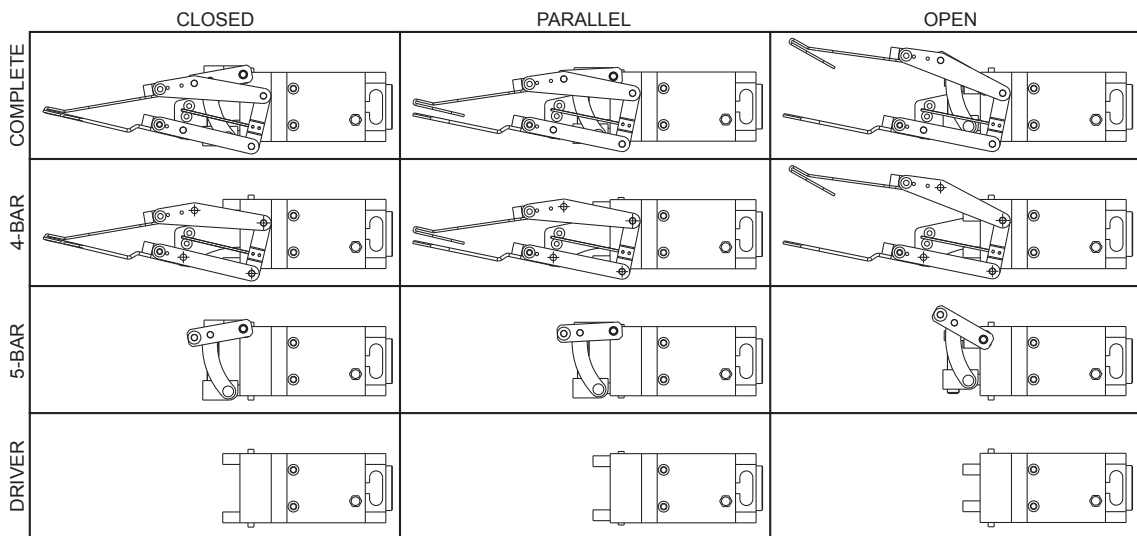


Figure 16: From bottom to top, side views of: driver; driver with the 5-bar assembled; driver with the 4-bar (5-bar not shown); complete gripper (driver, 5-bar, 4-bar). The mounting plates are removed to see the mechanisms. The 4-bar is in neutral configuration; the 5-bar is in open, parallel and closed configuration

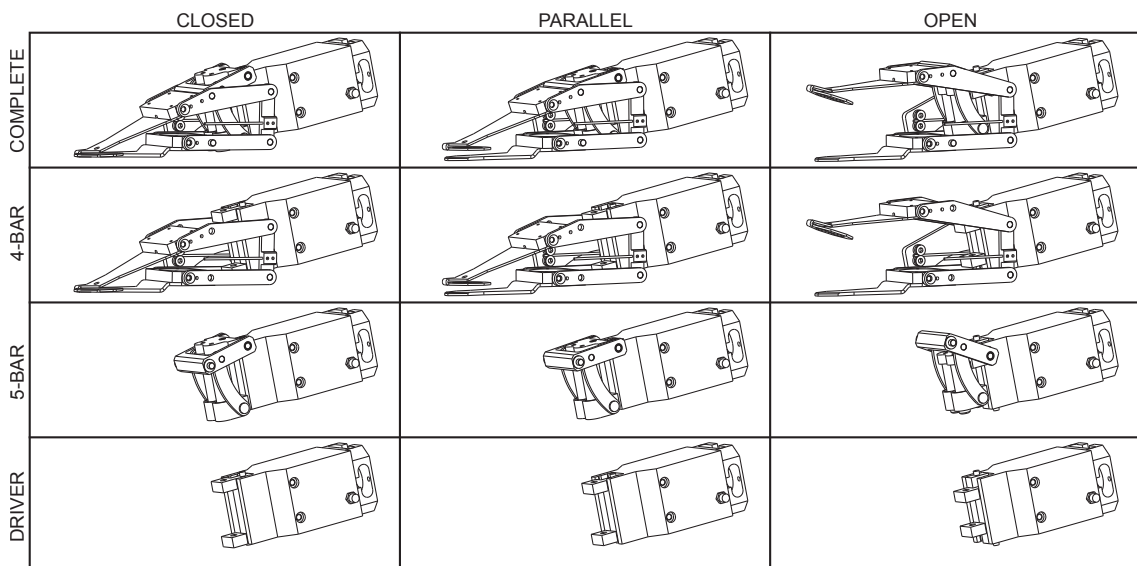


Figure 17: From bottom to top, axonometric views of: driver; driver with the 5-bar assembled; driver with the 4-bar (5-bar not shown); complete gripper (driver, 5-bar, 4-bar). The mounting plates are removed to see the mechanisms. The 4-bar is in neutral configuration; the 5-bar is in open, parallel and closed configuration

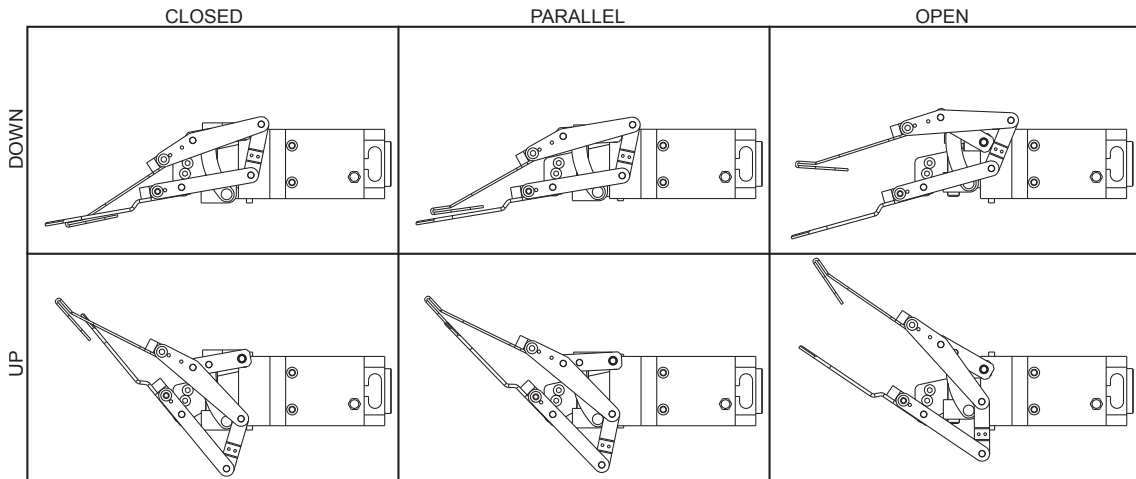


Figure 18: Up and down configurations of the 4-bar (maximum and minimum angle of the fingers) with the 5-bar in open, parallel and closed configuration

### 3.7 Gripper static modeling

The statics of the gripper has been modeled and the model has been used for the selection of the geometry, the sizing of the actuation, and the structural design. The model is not reported in this deliverable.

### 3.8 Gripper mechanical design

The industrial gripper selected as driver is the model MEG64 commercialized by the company Schunk. It is a translational gripper with maximum closing force (between finger blocks) of 140 N and 10 mm stroke per finger. The mechanism of the new gripper is assembled on the body of the MEG64 using the fixing holes present on its frame; the finger blocks are connected to the linkages of the new mechanism. All links are realized in aluminium with anodized surfaces; plastic bushes are used as bearings. The whole design is light and robust, with minimum mechanical plays.

The fingers are realized using bent metal sheets, cut and shaped. This has two advantages: each finger can be replaced without disassembling the mechanism, making it possible to customize the finger geometry; the fingers are compliant, they bend when they apply a pressure to the grasped item thus evening out the applied pressure along the whole area of contact. The compliance in the fingers helps making the grasping force independent from the thickness of the material grasped.

The MEG64 is designed to apply the maximum force both during the opening and the closing motions of the finger blocks. In the new gripper, the force is required only during the closing motion. An effective solution, allowing to have almost twice the maximum force of the MEG64 on the gripper mechanism is to add springs that are compressed when the finger blocks are open. When the gripper closes, the force of the springs adds to the force of the MEG64, increasing the total output force of the gripper. This allows the effective actuated force to increase to about 186 N; the corresponding maximum grasping force at neutral and parallel configuration is of about 33 N.

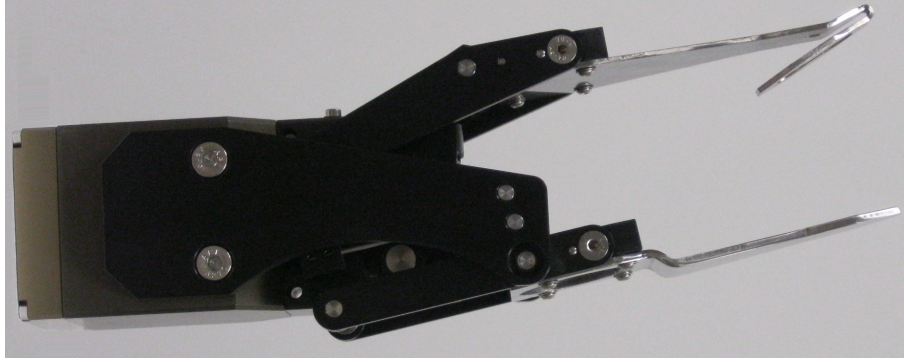


Figure 19: The final gripper.

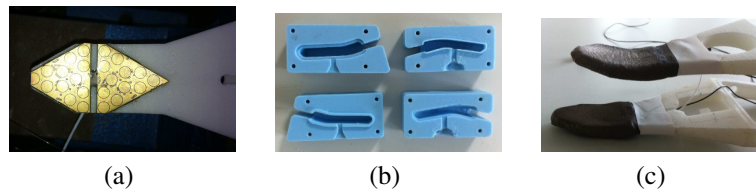


Figure 20: The first fingertips prototype: (a) the fPCB glued on a fingertip;(b) the specific designed moulds for the deposition of the silicone elastomer (c) the finished sensorized fingertips prototype.

The springs are well embedded in the complete design. The whole mechanism is made in aluminium except the flexible metal bar (Fig.19) with the same function as in version 0.3 Maximum opening of the gripper is 50 mm and it can apply up to 30-35 N of gripping force. The gripper has been tested with the typical laundry heap to validate its performance. However, more experiments and modifications are necessary to validate and enhance all the functions of the gripper.

Grippers are sent to all partners for their experiments. The assembly instruction and the manual for the gripper is prepared.

### 3.9 Sensorization of the first finger prototypes

The first fingers prototype has been made by rapid prototyping process and, from the tactile sensor point of view, this solution allowed to glue the fPCB directly on the finger surface without the use of insulating layer; for this first prototype, 2 triangular modules have been used for the sensorization of each fingertip (see Fig. 20a).

A layer of a very soft silicone elastomer (Ecoflex by Smooth-on) has been deposited over the fPCB with the help of specific designed mould (see Figure 20b). Finally, the conductive lycra has been glued with a specific adhesive (sil-poxy by Smooth-on) on top of the silicone elastomer. Figure 20c shows the first sensorized prototype of fingertips.

In the final design, the fingers are made of steel plates. In order to insulate the sensor from the fingers, an insulating layer made with kapton has been put over the fingertips surface. In order to simplify the procedure for the deposition of the silicone elastomer, a new solution has been adopted: a neoprene sheet 3 mm thick has been cut with the same

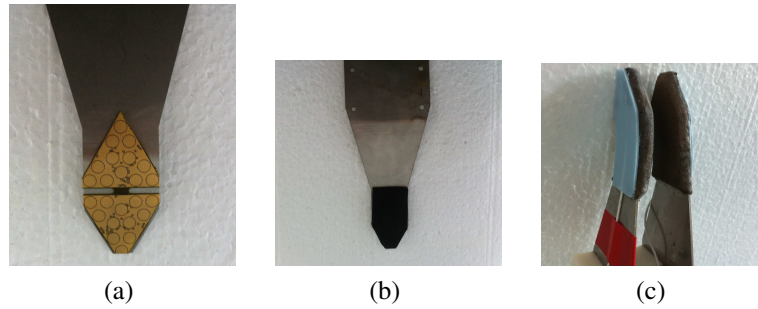


Figure 21: The second fingertips prototype: (a) the fPCB glued on a fingertip;(b) the neoprene, used as dielectric layer, glued on the fingertip (c) the finished sensorized fingertips prototype.

shape as the fingertips and it has been glued with a cyanoacrilate glue. Finally, as the last step, the lycra has been glued on top of the neoprene layer.

The two fingers have been mounted on the gripper for testing during pinching operations. Further adjustments have been done to increase the tangential force available upto the target value.

### 3.10 Further developments of the gripper

The version of the year-1 hand under testing is not equipped with tactile sensors (demonstrated separately on a dedicated bench) and the motion of rubbing is not actuated (passive rotation of the 4-bar with the springs to determine a configuration of elastic equilibrium). The development of the hand will continue with the incorporation of one tactile pad (a new version of the capacitive tactile sensor is under development) in the upper finger and with the actuation of the rubbing motion.

The tactile pad will replace the dummy version that is currently present in order to recreate same grasping conditions; the modularity of the finger design will make this replacement immediate; the electronics of the sensor (a new microcontroller board for the data acquisition is under development) will be positioned in a room already present in the finger body.

The actuation of the rubbing motion will be realized with an additional actuator that will be fixed to the upper hand body. There are two architectural schemes currently under evaluation: a standard position actuation and a variable stiffness actuation. The first option is of easier implementation since it requires a replacement in the couplers of the 4-bar gripper mechanisms and the placement of the actuators on the gripper body. The implementation of a variable stiffness actuation is more complicated. We are currently oriented to the use of two linear actuators on the gripper body operating a fluidic transmission which would generate the rubbing motion of the fingers.

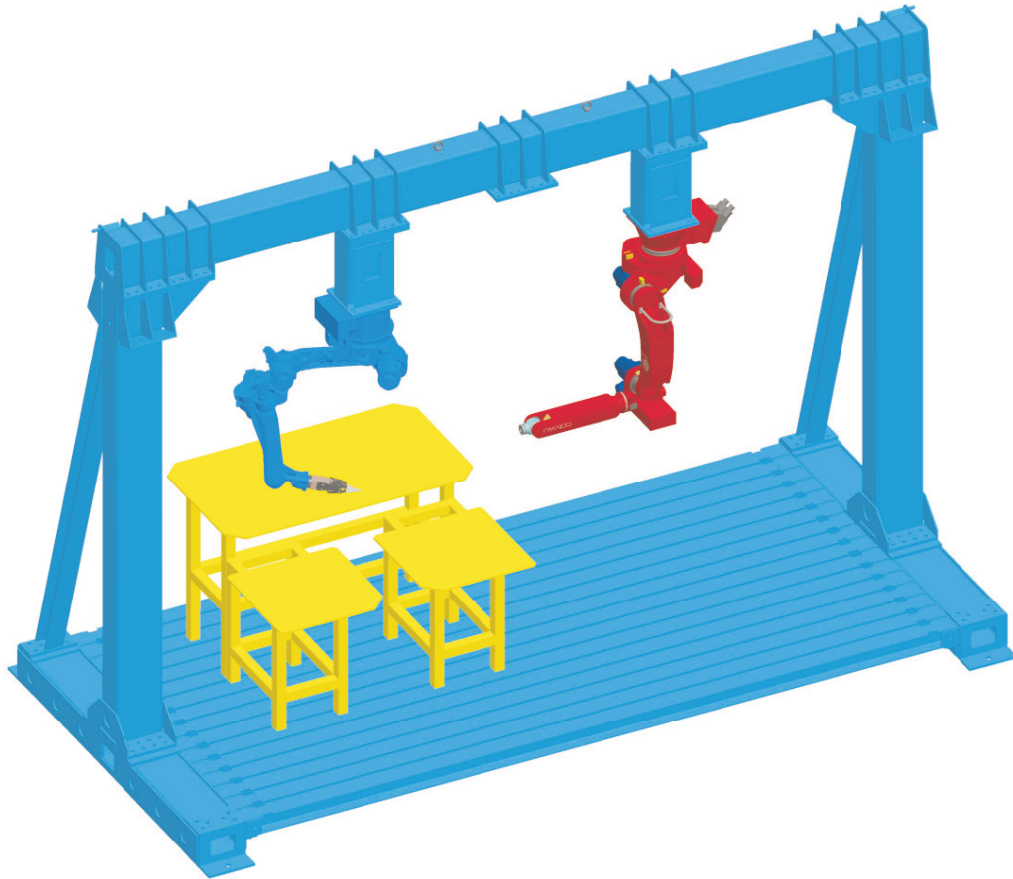


Figure 22: Layout of the cell for the development and testing of the gripper

## 3.11 Demonstration

### 3.11.1 Demo cell layout

A testing layout has been designed and is under construction. The robot is suspended upside down to a portal for a better use of the workspace, Fig.22; the hand is fixed to the arm, Fig.23.

### 3.11.2 Gripper

The gripper is mounted on the arm. The current plan is to operate the arm manually from remote. Two tasks are considered: picking a cloth from the heap and lifting a cloth from a table, both performed manually.

### 3.11.3 Tactile sensors

The tactile sensors are demonstrated separately on dedicated benches. The aim of the demonstration is to show that they can be operated and used to distinguish different types of fabric based on their texture.

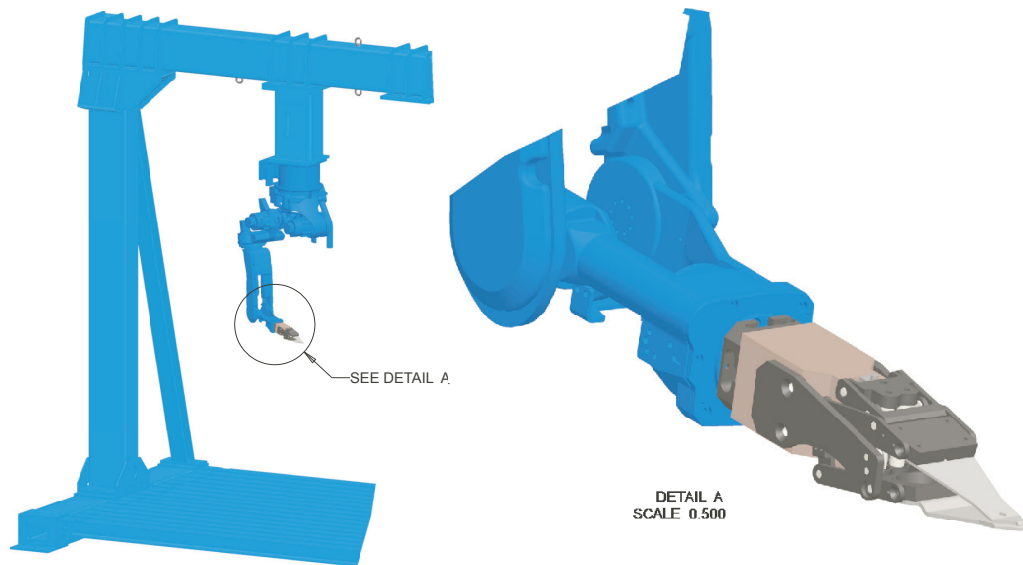


Figure 23: Details of the robot mounting and hand-arm assembly

## 3.12 Performance

### 3.12.1 Gripper

The efficacy of the gripper is verified experimentally by performing manipulation tasks and by measuring the force between the fingertips.

### 3.12.2 Tactile sensors

Sets of experimental trials are carried out, the data are processed and the correspondence between inferred and real properties is checked and assessed.

## 4 Binocular Robot Head, Computer Vision and Depth Sensing

University of Glasgow (UG) developed three binocular robot heads comprising a stereo-pair of actuated cameras, each camera mounted on accurate pan-tilt actuators (suitable for photogrammetric measurements) and the assembly mounted on a commercial-off-the-shelf (COTS) camera mounting frame.

As stated in Deliverable 1.1 [2], the aim of the CloPeMa binocular robot head was to integrate high resolution imaging for 3D mapping and range images and low resolution imaging for real-time gaze control and feature extraction. These operational modes will enable the high-level reasoning engine working in conjunction with CloPeMa dual-robot arm and tactile and RGBD sensors to exploit sensed visual information for the tasks of manipulation and interaction with garments.

Range capture was supported based on the above head using the UG C3D technology integrated in ROS to provide high resolution full field-of-view (FOV) range maps. We have investigated accelerated codes for processing range maps in order to reduce range map acquisition and processing times very substantially. Camera calibration was provided based on the standard C3D (using multiple observations of a planar target). Likewise, we have also explored the integration of camera calibration under dynamic camera motion using standard OpenCV camera calibration routines. Therefore, the calibration system will maintain the calibration of the camera system under extrinsic camera movement.

Accelerated SIFT feature extraction from each camera in the binocular head was provided based on a freely-available GPU implementation [16]. SIFT features extracted from each camera of the stereo-pair are employed to maintain camera vergence using inter-camera correspondences in order to maximise the overlap of the field of view of the stereo-pair cameras for range map computations. The above facilities were interfaced within the ROS environment.

### 4.1 Description

Under the remit of this deliverable, UG has investigated suitable low-cost camera and actuator configurations to implement three fully programmable binocular heads, one for each experimental testbed, that are suitable for maintaining calibration by means of photogrammetry and actuator position feedback. Therefore, three basic stand-alone binocular robot heads have been delivered (as detailed in Deliverable 4.1 [3]). These integrated binocular robot heads are capable of:

1. Directing their gaze under program control & acquiring stereo-pair images in full- (16 megapixels) and low-resolution (VGA resolution)
2. Delivering calibrated range data based on UG's C3D stereo-photogrammetry package
3. Delivering SIFT features in full and low-resolution modes

Likewise, we have investigated accelerated codes for range map construction under Task 4.2 Accelerated Range Sensing, the next phase of this research (Section 4.4.1).



### 4.1.1 Robot installation

The robot at University of Glasgow will be installed by the end of February. Due to the weight, mobility constraints and operating distance of the robot it has been difficult to find an adequate room for the robot, which would fulfil UK health and safety regulations. The School of Engineering has provided four rooms which have been assessed for work-space size, health and safety and for accessibility for robot installation. Two of the rooms fulfilling most criteria were surveyed by an independent structural engineer for floor strength. One room was later on discounted as its floor was not sufficiently strong to allow the robot to operate safely.

The room selected is located on the seventh floor of the University's Boyd Orr building that was constructed in 1968. It has required a substantial amount of refurbishment, including strengthening of the floor to accommodate the weight of the robot. During the preparation of the refurbishment asbestos containing material had been found and required removal. This caused a considerable delay as the governmental policies for asbestos removal are very strict, requiring notice being served to the governmental health and safety executive. The asbestos has been removed but the main refurbishment is still on going and is due to be completed shortly after the 2nd of February.

The Boyd Orr building is in heavy use by different Schools of the University and consequently much of the work has to be done at week-ends.

The refurbishment costs for the CloPeMa project are being fully subsidised by the College of Science Engineering and the School of Computing Science own funds. This amounts to £21,631.24 from the College funds and approximately £11,021.66 from the Schools own research funds. Although safety considerations and refurbishment has caused a delay in installing of the robot, the University has been generous in providing a total £33,652.90 (39,307.93) to support the installation.

### 4.1.2 Parallel matcher

Because the existing 3D matcher software was originally intended for batch applications and because it is legacy code of a considerable age (dating back some 20 years) it became clear at the first tests using the new cameras that it was going to be too slow for real time use. We thus decided to advance the work originally intended for the second year and to develop a multi core version re written from scratch, but following the same algorithmic design as the current matcher. We report results of this in section 2.

## 4.2 State of the realisation

### 4.2.1 Robot head Hardware

Figure 24 depicts the integrated robot head delivered in [3]. Below, a brief description of the selected hardware configurations for the CloPeMa project is given.

As detailed in [3], the cameras for the CloPeMa robot head were low-cost, off the shelf and supported by the software operating system. UG selected the **Nikon D5100**<sup>1</sup>

<sup>1</sup><http://www.nikonusa.com/en/Nikon-Products/Product/Digital-SLR-Cameras/25478/D5100.html>





Figure 24: Integrated robot head.

model (Figure 24) as it fulfils the requirements for the CloPeMa project. The selected camera model features have been described in [3].

The mechanical configuration and design of the CloPeMa robot head followed a previous active binocular robot head [7] developed at the Computer Vision and Graphics Laboratory (CV&G Lab) at the University of Glasgow. For the CloPeMa project, UG therefore adopted this configuration for its modularity and ease of integration with other robotic hardware. Thus, the above Nikon cameras were mounted on two motorised Pan-and-Tilt Units (PTU) which are designed for high-speed, accurate positioning of camera, laser, antenna, or other payloads. Features of the most cost-effective and delivered units were detailed in [3].

#### 4.2.2 Robot Head Software

The CloPeMa robot head, as specified in [3] and Section 4.1, is able to verge its cameras on a point in order to improve range image computation, to compute disparity maps for range computation by disparity dynamic range minimisation and also by maximising the overlap of the field of view of the stereo-pair cameras. Software functionalities were implemented as ROS nodes. As detailed in deliverable [3], hardware was selected in order to be compatible with Ubuntu and, in consequence, cameras and motor controller drivers were implemented as ROS nodes. In [3], UG described the implementation and development of ROS nodes for the CLoPeMa head functionality. For completeness, a brief description of these ROS nodes is given below.

- Feature Extraction: SIFT features in the Graphics Processor Unit (GPU) [16] for real-time operation for gaze control, object recognition and binocular vergence
- Cameras drivers: these provide the required image capturing capabilities
- Pan and tilt drivers: These units were interfaced and controlled by a C++ ROS node

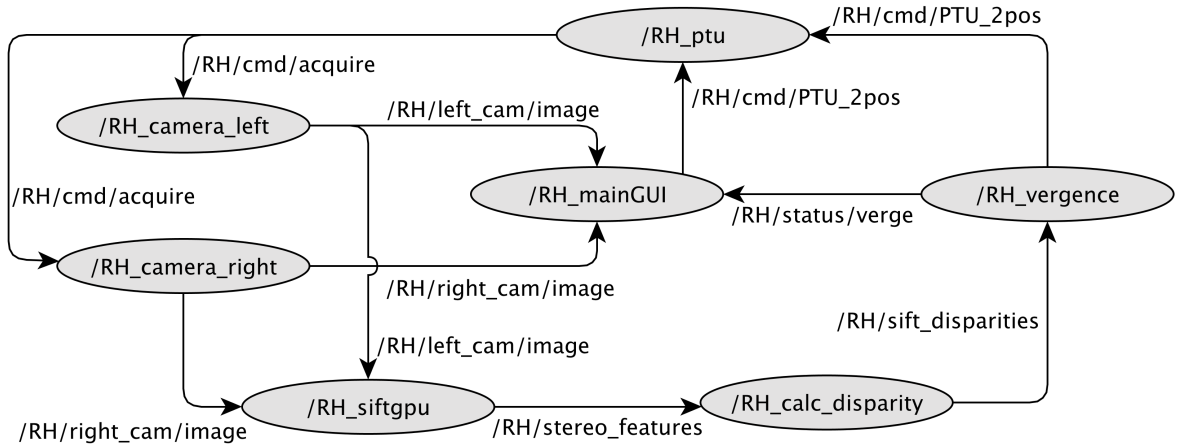


Figure 25: ROS graph architecture of the CLoPeMa robot head.

- Binocular vergence: C++ ROS node in order to verge the cameras towards a point. The vergence algorithm is described in [7]
- Gaze control and object recognition: A set of ROS nodes that perform detection and localisation of single non-deformable objects in the robot's environment. These functions are C++ translations of the MATLAB algorithms detailed in [7]
- Graphic User Interface (GUI): C++ ROS node that allows an user to control the gaze of the robot head and provides a graphical front end for demonstration purposes

Figure 25 depicts a ROS graph comprising the above software functionalities in terms of ROS nodes. This graph also depicts the information flow through the ROS architecture.

#### 4.2.3 C3D in ROS

The robot head has been equipped with software support for stereo-pair image matching and stereo-camera calibration based on the integration of the University of Glasgow's proprietary stereo-photogrammetry package, C3D, in two ROS nodes of the system. The C3D matcher node finds stereo-correspondences between the left and right camera images, which are outputted as horizontal and vertical disparity maps, and then used to create a range image aligned with the left camera imaging plane.

Due to the real-time requirements of the robot system, C3D has been configured to match the images at quarter resolution, reducing the processing time from approximately 20 minutes for 16 Megapixel resolution stereo-pair images to just under one minute for 4 Megapixel resolution images. To enable C3D to produce accurate range images, C3D requires calibration parameters for both cameras of the robot head. C3D own calibration software has therefore been integrated in a second node. The calibration will produce two calibration files, one for each camera, `lefcac.tcl` and `rigcal.tcl`. These files contain the extrinsic and intrinsic parameters for each camera in form of rotation and translation parameters, principal point, focal length, and pixel size in meters in the x and y direction. When range images require to be produced, these calibration files are automatically loaded into the C3D matcher node.

A limitation of the current ROS nodes is that the camera's calibration is not maintained if the robot head changes its internal and external parameters. For this purpose, the following section describes a dynamic calibration routine in order to allow the robot head to maintain external calibration parameters.

#### 4.2.4 Delivered matching algorithm.

The C3D stereo-matcher takes two images, as input and outputs a horizontal disparity matrix, a vertical disparity matrix and a confidence matrix of the same size as the input images. The algorithm used is as followed:

1. First a pyramid representation with a given number of pyramid levels for the left and right image is produced. For the smallest level of the pyramid the horizontal disparity, vertical disparity and confidence matrix are initialised to zero
2. Starting at the smallest pyramid level Step 2 will be repeated for each pyramid level:
  - (a) Step 2a to 2h is repeated for each iteration on the current pyramid level: The sampling step (*sampStep*) is initialised to one by default, unless *totIter* > 1 and *curIter* < 7 . The sampling step is calculated as follows:  

$$smpStep = (totIter - curIter - 1) * \frac{0.9}{totIter-1} + 1$$
 where *totIter* is total iteration and *curIter* is the iteration counter
  - (b) Step 2b to 2h is repeated twice for each iteration (1 iteration = 2 match cycles): Five different warped right image instances are created by applying bi-linear interpolation using the horizontal and vertical disparities with and without sampling step shift. The first is created using the horizontal and vertical disparities without shifts. For the other four, horizontal and vertical  $\pm smpStep$  is added to the disparities before warping
  - (c) For the left image and each of the five warped images a standard deviation matrix is calculated by multiplying the images with themselves, i.e. squaring them, and convolving the result horizontally and vertically with a Gaussian kernel
  - (d) With the same Gaussian kernel, covariance matrices are calculated multiplying the warped images with the left and then convolving
  - (e) Based on the standard deviation and covariance matrices the correlation coefficients are calculated
  - (f) Having obtained five correlation coefficient matrices, 2nd order polynomial maximisation is applied to the corresponding elements of the matrices. In case of a confidence value greater than one and the zero point correlation coefficient ( $c_0$ ) greater than zero it readjusts the disparity (*disp*) to be:  $disp = disp * \frac{1-c_0}{conf-c_0}$  followed by overwriting the confidence with 1. Should C3D fail to fit a polynomial, it sets the disparity to zero and the corresponding confidence to 0.4

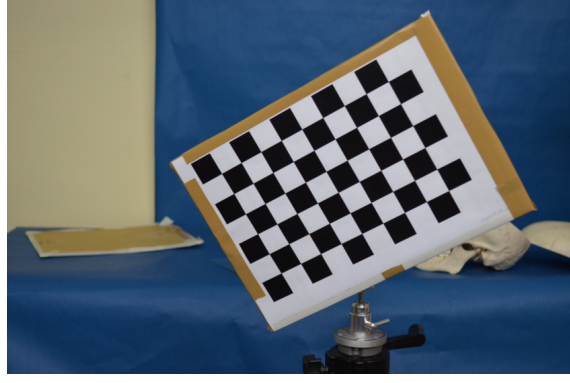


Figure 26: OpenCV's chessboard target used for calibrating the CLoPeMa binocular robot head.

- (g) The newly computed horizontal and vertical disparities matrices are multiplied by the sampling step and added on to the initial horizontal and vertical matrices. To obtain again one confidence matrix it computes  $Conf_{HV} = Conf_H \otimes Conf_V$  weights this against the initial confidence ( $InitConf$ ) :  $InitConf = Conf_{HV} + 0.75 * (InitConf - Conf_{HV})$
  - (h) The disparity matrices are then weighted by the confidence matrix  $Disp_{xy} = Disp_{xy} \otimes InitConf$ . This is followed by smoothing using the local confidence matrix as a convolution kernel. The same process is then applied to the confidence matrix The weighted smoothing process (2h) is repeated twice. If the steps 2b to 2h have been run twice the process will either step to the next iteration, starting at 2a or in case the total number of iterations for the pyramid level has been reached move to the next pyramid level (3)
3. After the matching iterations the resulting horizontal and vertical disparity matrices and the confidence matrix are smoothed once more with a 3x3 average filter, before being expanded to the size of the next pyramid level. OR
  4. The final (largest) level of the pyramid is returned as output

#### 4.2.5 Dynamic Calibration

The aim of the dynamic calibration routines is to provide ready-to-use calibration routines based on OpenCV and ROS implementations. Camera calibration is provided under intrinsic and extrinsic camera calibration parameters for each camera by sampling the camera's parameter space using a chessboard target (Figure 26).

In order to provide the required geometric transformation to maintain extrinsic calibration under different camera motions, it is required to find the forward kinematic model of the CloPeMa robot head. Figure 27 depicts the coordinates frames in the "HOME" position of the robot head (H subscript in the figure). We use Sharkey et. al. [14] notation to express the forward kinematic model of the CloPeMa robot head. Table 1 defines the coordinate frames (based on [15]). We assumed that the world reference frame,  $\{W\}$ , lies on the base,  $\{B\}$ , of the robot head and  $y_{CS_0} = z_{CS_0} = \theta_C = 0$ . Sharkey et. al. [14]

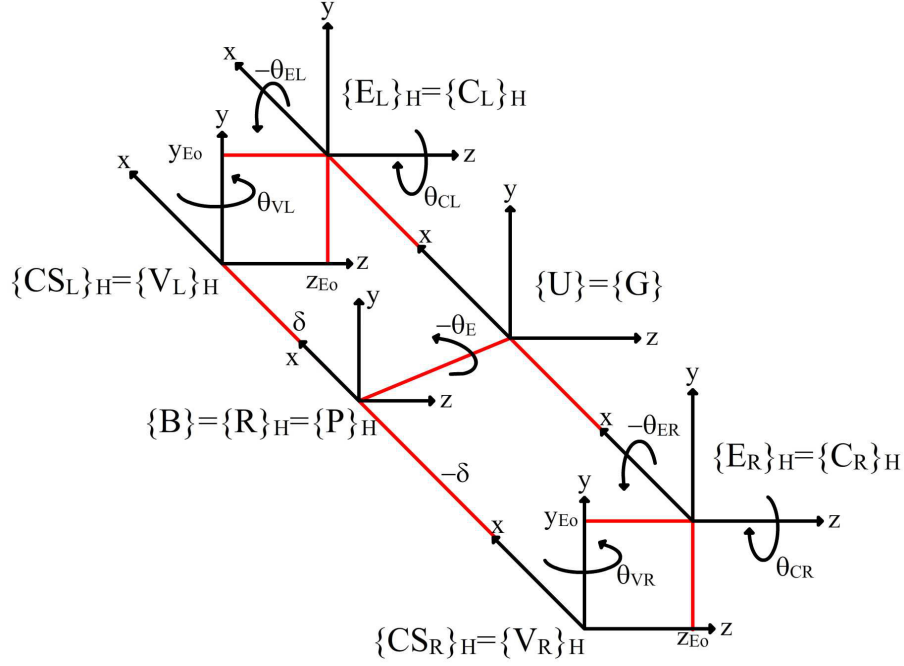


Figure 27: Coordinate frames in the HOME position of the CLoPeMa robot head.

established that if the cameras are targeting the same point in  $\{W\}$ , then both open kinematic chains for each camera will be closed around the target. Therefore, the forward kinematics for the open kinematic chain from the base to the target position is found as:

$${}^bD = {}^W T_T \times \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} s\delta + d_T \cos(\theta_E) \sin(\theta_V) + z_{E_o} \sin(\theta_V) \\ d_T \sin(\theta_E) + y_{E_o} \\ d_T \cos(\theta_E) \cos(\theta_V) + z_{E_o} \cos(\theta_V) \\ 1 \end{bmatrix} \quad (1)$$

where  ${}^bD$  is the kinematic chain,  $s$  indicates if the camera separation is for the left,  $s = 1$ , or right,  $s = -1$ , camera; thus, the forward kinematic chain holds true for the left and right camera.

The above forward kinematic chain is not complete as we assume that the tilt reference frame is aligned to the optical image plane. In the CLoPeMa robot head, this assumption does not hold true and it is required to find the geometric transformation from the tilt reference frame to the principal point of the camera's image plane ( $\{E\}$  and  $\{C\}$  in Table 1, respectively). In order to find this transformation, we implement and carry out Tsai's hand-eye calibration method [15].

The complete camera-hand eye calibration routine implemented is depicted in Figure 28. The strategy adopted consists of capturing images of the chessboard target with both cameras in a static position. The target is presented to the cameras at different positions and orientations in order to sample the camera parameter space adequately. After capturing these images (e.g. 10 images for each camera), the ROS node then calibrates each camera in isolation and outputs the calibration parameters. Thereafter, the node starts capturing images again over different camera positions while the chessboard target is static

Table 1: Coordinate frames definition.

$\{W\}$	World
$\{B\}$	Base
$\{R\}$	Roll (Not considered)
$\{P\}$	Pan (Not considered)
$\{CS_{R/L}\}$	Camera Separation (left / right) Offsets : $(0, y_{CS_0}, z_{CS_0})$ ; and from $\{B\}$ is given by $+ / - \delta$
$\{E_{R/L}\}$	Independently Elevated (left / right)
$\{V_{L/R}\}$	Vergence (left / right) Offsets wrt to $\{E\}$ : $(0, y_{V_0}, z_{V_0})$
$\{C_{L/R}\}$	Cyclotorsion (left / right) (Not considered)
$\{O_{L/R}\}$	Optical (left / right) Offsets wrt $\{C\}$ or $\{V\}$ (for systems without $\{C\}$ : $(\alpha, \beta, \gamma)$ and $(u, v, w)$
$\{T\}$	Target object. For simplicity, both centres represent a single point in $\{W\}$ Offsets wrt $z_{OL/OR}$ : $d_{TL}$ and $d_{TR}$

(e.g. 10 images). For each camera movement, it is computed the forward kinematic chain for each camera in the current camera position. Finally, it is carried out the hand-eye calibration for each camera in order to obtain the geometric transformation that relates  $\{E\}$  and  $\{C\}$ .

We are currently working on linking the above calibration routines in order to be compatible with C3D calibration reference frames formats 4.2.3. Similarly, we are currently investigating how to maintain dynamic intrinsic calibration (i.e. change of focus). We will also validate the extrinsic dynamic calibration in terms of drift errors and accuracy of the computed range maps.

#### 4.2.6 Parallel matcher

The aim has been to write a version of the matcher that uses the same basic algorithm as we have used in the past but which can be compiled to a variety of platforms with different levels of parallelism to achieve the same semantic effect without altering the source code. The previous system is coded in Java, the new system has been written in Pascal and is compiled with the Glasgow Pascal Compiler which auto-parallelises across SIMD and multi-core architectures. In the compiler allows all operators to be overloaded so that they can operate on arrays and vectors as well as scalars.

Each array assignment statement is evaluated by the compiler for parallelisation in three ways:

1. If it is a two dimensional array and it uses only basic arithmetic operations or pure functions ( side effect free) on the right hand side, then the work on rows of the array is interleaved between different  $n$  processors so that processor  $j$  handles all rows  $i$  such that  $i \bmod n = j$ .
2. If the right hand side contains no function calls and operates on adjacent array elements, then the compiler generates SIMD code.
3. If the expression on the right is a conditional one with no function calls it is evaluated in parallel using boolean masks to allow SIMD execution.

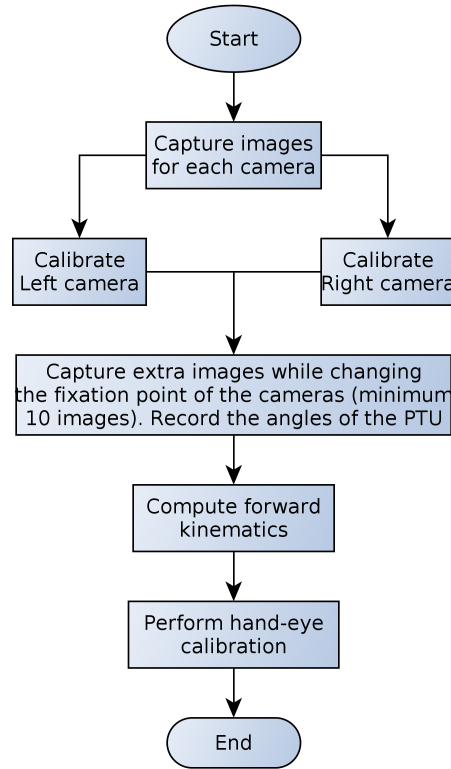


Figure 28: Flow diagram of the camera-hand eye calibration ROS node.

The recoding involved converting the algorithm from one using explicit nested loops to one that is expressed in a data parallel way in terms of operations on entire image arrays. This technique will be familiar to Matlab coders, but instead of running interpretively as on Matlab the operations are compiled.

### 4.3 Demonstration

University of Glasgow (UG) has designed and delivered 3 working binocular robot heads and provided a graphical user interface that supports, stereo-pair image capture, camera pan/tilt control and automatic vergence control either by physically converging or diverging the cameras or by shifting the offset between the captured stereo-pair images. Similarly, we integrated range capture based on the UG C3D technology and this was integrated in ROS in order to provide high resolution full field-of-view (FOV) range maps.

During the M12 demonstrator, UG will therefore present:

- A video demonstrating the graphical user interface of the CLoPeMa binocular robot head that shows stereo image capture, pan and tilt control, real-time feature extraction and automatic vergence while a user selects a fixation point on the interface provided. A brief description on the operation of the robot head is provided in Section 4.3.1
- A set of image examples that depicts the quality of the range maps captured (Section 4.3.2)



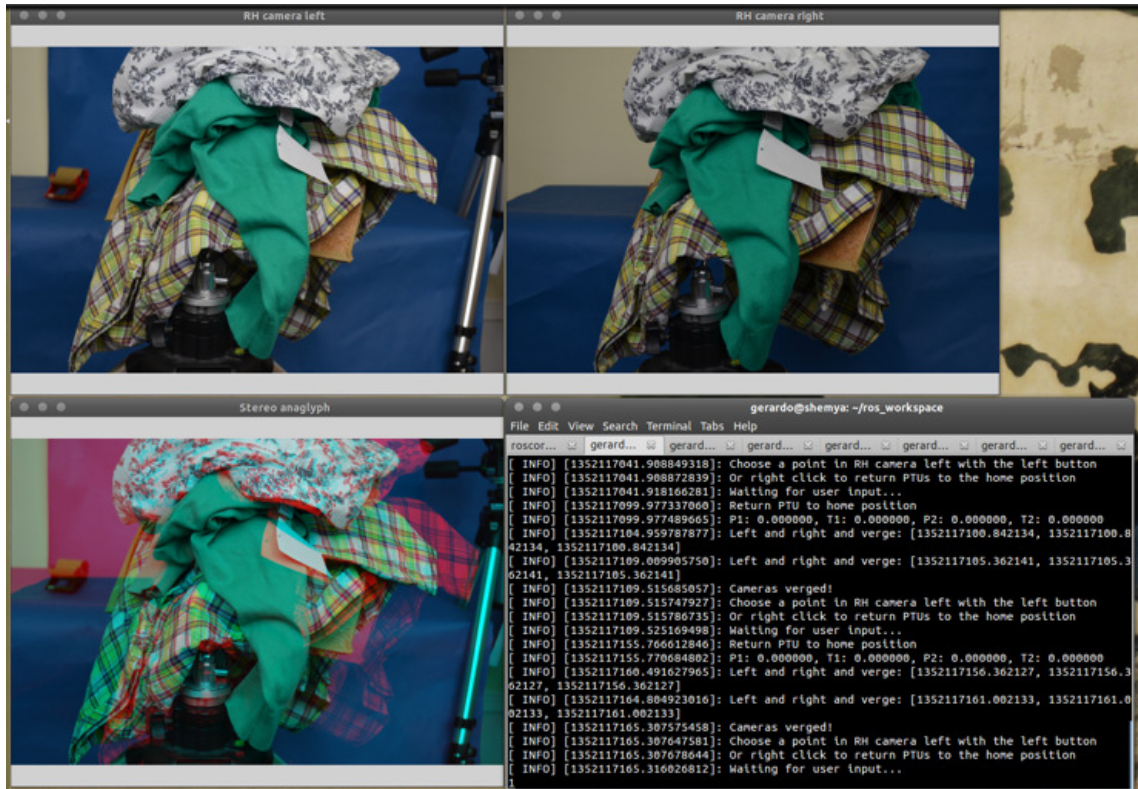


Figure 29: Working session of the software of the CloPeMa robot head.

#### 4.3.1 Main interface of the RH

The control of the robot head consists of 3 windows displaying the left and right camera images and an anaglyph depicting the stereo registration between both cameras while the system verges on a point. Figure 29 shows a working session with the robot head under Ubuntu. In order to operate the system, it is needed to start the ROS manager (e.g. *roscor*, issued in a terminal window) and the corresponding nodes for controlling the robot head. A video is supplied demonstrating the above functionality (see <http://youtu.be/AIFgjJupL8> ).

#### 4.3.2 Examples of range models of clothes

In figure 30 we can see on the left the range image of a towel as it is produced by the C3D node with matching the images at full resolution and on the right as it is produced at quarter image resolution. In order to counter the delay incurred with the robot installation, UG has been going ahead of schedule with the early development of the parallelised Vector Pascal matcher as described in section 4.1.2. Figure 31 shows the latest range image results of the parallelised matcher. It requires still a little more optimisation to have the same level of accuracy, however the current results are already viable. It is to be expected that the new parallelised version of the matcher will therefore be released ahead of the official deliverable time.

Further range map examples have been attached to this document.



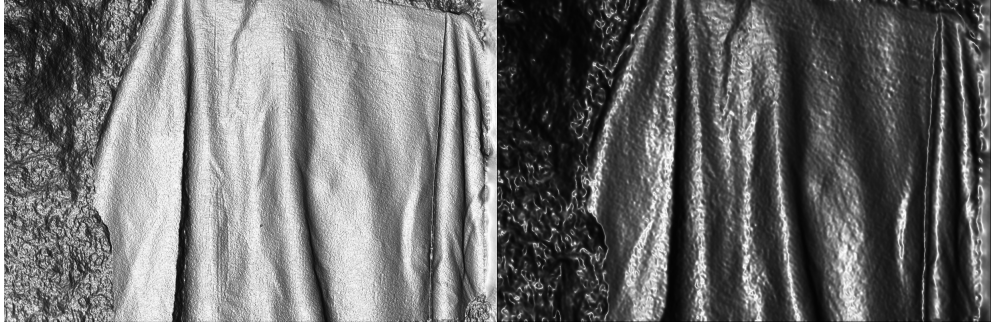


Figure 30: C3D matcher range image results, on the left for full resolution, on the right with quarter resolution.

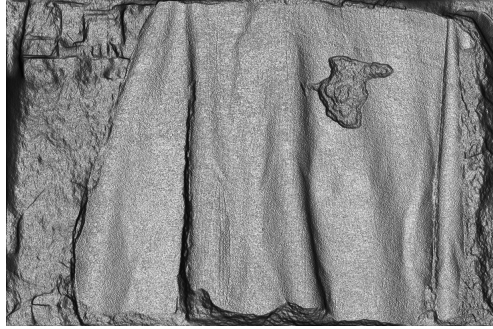


Figure 31: Parallelised matcher range image results.

#### 4.4 Performance

Binocular vergence is implemented as a closed-loop system based on extracting and matching SIFT features from each stereo-pair images captured by the CLoPeMa robot head system. The induced disparity is then calculated by matching SIFT features between the stereo-pair images and histogram these disparities in order to determine the distribution of the disparity magnitudes. The highest peak of this histogram in essence cues the most dense/compact cluster of SIFT-features which are of similar distance as observed by the cameras. The resulting disparity associated with the highest peak is then used to estimate the actuator movement required to rotate the gaze angle of each camera in order to centre this peak at zero disparity in the  $x$  and  $y$  axes. The capture-verge loop halts when the highest disparity peak is below a threshold. We have found that 15 pixels threshold works well in practice, i.e. this represents 0.3% and 0.5% in the  $x$  and  $y$  axes respectively for a 16 Mega Pixel image and 2.3% and 3.1% in the  $x$  and  $y$  axes respectively for VGA image resolution.

In order to measure the performance of the binocular vergence node, the resultant disparity error incurred is recorded after each successful vergence cycle. As there is not a defined vergence point, it is only recorded the residual disparity error (i.e. the disparity associated with the highest peak) in both axes and the number of vergence iterations required to stabilise the algorithm. We thus carry out 50 manual fixation points as demonstrated in Section 4.3.1 and record the resultant disparities. Figure 32 depicts box plots for the  $x$  and  $y$  axes that characterise the performance of the vergence node, whereas Table

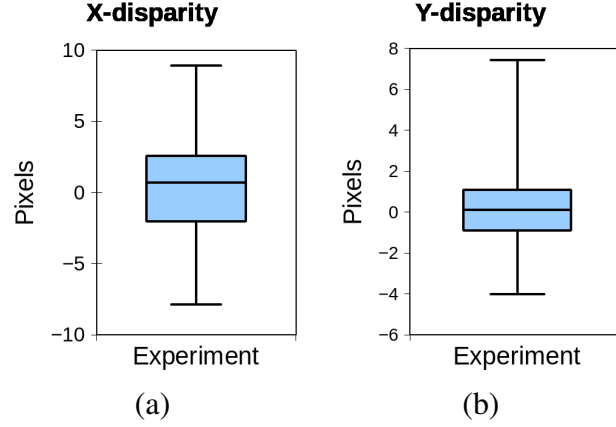


Figure 32: Box-plots of the measured disparity errors in the (a) x- and (b) y-axis.

Table 2: Vergence performance.

	X	Y
Mean	0.46	0.17
Standard error	0.55	0.23
Median	0.72	0.1
Standard deviation	4.07	1.76
Minimum value	-7.85	-4.01
Maximum value	8.93	7.43
First quartile	-2.02	-0.9
Third quartile	2.58	1.09

2 presents the statistic measures used to generate Figure 32.

As observed in Figure 32, the measured disparity errors are within  $\pm 10$  pixels from the statistical zero mean. Therefore, the degree of dispersion achieved is still within practical limits for stereo reconstruction and depth recovery as pointed out in [7]. Likewise, the vergence algorithm takes an average of 1 iteration in order to reduce the level of residual disparity below the terminating threshold.

#### 4.4.1 Parallel matcher

We tested speed and precision for the Pascal matcher and the C3D matcher using fractal-noise image pairs. The fractal noise images (2448x2050 pixels) were generated with different noise frequency patterns and warped using bi-linear interpolation for a set of predefined disparities maps. For the stereo-matching the two original images served as the left and the warped images as the right image. To evaluate the precision of the new matcher and C3D, the output disparities have been compared against the predefined disparities, calculating the root mean square (RMS) error over the differences of disparities. The speed of the parallelised matcher was tested for two different instruction sets, AVX and x87, two different image formats, monochrome and colour, and run in parallel over different number of cores for the three processors presented.

Table 3: Comparative speed of matching for different combinations of processor and image format.  $\mu, \sigma$  stand for mean and standard deviation of the runs respectively.

Processor	Instrs	Procs	Monochrome		Colour	
			$\mu$	$\sigma$	$\mu$	$\sigma$
Intel 4	x87	1	18.42s	0.07s	33.45s	0.25s
Intel 4	AVX	1	8.39s	0.03s	12.73s	0.04s
Intel 4	AVX	4	3.30s	0.01s	5.55s	0.01s
Intel 6 (12)	x87	1	16.23s	0.03s	30.03	0.002
Intel 6 (12)	AVX	1	7.34s	0.02s	11.15s	0.02s
Intel 6 (12)	AVX	6	2.82s	0.01s	4.60s	0.01s
Intel 6 (12)	AVX	12	<b>2.36s</b>	0.04s	<b>3.98s</b>	0.06s
AMD 8	x87	1	28.03s	0.14s	51.48s	0.20s
AMD 8	AVX	1	9.72s	0.07s	15.20s	0.07s
AMD 8	AVX	8	3.98s	0.04s	6.93s	0.06s

**Speed results** We have available 3 architectures: a 6 core Intel Ivy-bridge (3.2GHz, 12MB cache), an 8 core AMD Bulldozer (3.1GHz, 8MB cache) and a 4 core Intel Sandy-bridge (3.1GHz, 6MB cache). The Ivy- and Sandy-Bridge and the Bulldozer utilise the AVX instruction set, capable of operating on vectors of 8 single precision floating point values. The single core x87 instruction set served as the speed benchmark for each processor. The Intel 4 and Intel 6 both run colour matches 2.6 times as fast with AVX. Multi-core parallelisation gives further accelerations of 2.3 and 2.8 respectively. The AMD 8 has a 3.5 times speed-up on colour due to AVX and around 2.2 times due to multicore. However the AMD 8 was overall the slowest. Hyper-threading provided noticeable speed-up. Cache optimisation (elided from tables) of the convolve function to work on cache size image strips gave a further speedup of the overall algorithm of 1.2 times.

C3D was run on the Intel 6 with Java SE7, for 1 iteration at 16.10 seconds and for the standard number of 10 iterations at 38.93 seconds. As C3D can not match colour images, the timings are all monochrome images (see Table 3). Comparing the AVX 12 core version on the Intel 6 with C3D's standard setup, the new matcher is approximately 16 times faster. Even using colour images it is approximately 10 times faster. Considering the MSSM timings using the 1992 SPARCstation LX of 215 seconds over 576x768 pixels, in 20 years we have become 1020 times faster. This is almost exactly in line with Moore's law growth rates.

**Accuracy results** C3D is slightly more accurate than the Pascal version, with an RMS error of 0.41 pixels, compared to the 0.83 pixels (see Table 4). We attribute the difference to the slightly different scale-space pyramids used. However, the new matcher achieves more accurate results over colour images than over monochrome, which is interesting as C3D can only match monochrome images.

Overall, the 6 core Ivy Bridge has shown the best performance in terms of balancing speed and accuracy. The performance of the new matcher demonstrates the potential feasibility of integrating a stereo-robot head into a real-time system. We will investigate whether the differences between our prototype pyramid library in Pascal and the existing Java pyramid library account for the difference in accuracy.

Table 4: Comparative accuracy of matching

Img	Disp	Java C3D		Monochrome		Colour	
		1 Iter RMS	10 Iter RMS	AVX RMS	x87 RMS	AVX RMS	x87 RMS
1	1	0.00	0.01	0.29	0.50	0.09	0.11
	2	0.06	0.02	0.29	0.50	0.09	0.11
	3	0.05	0.02	0.29	0.50	0.09	0.108
	4	0.33	0.32	0.66	1.26	0.32	0.47
	5	0.95	0.92	5.35	6.52	1.51	1.44
	6	1.17	1.15	13.12	13.55	3.0	2.81
2	1	0.01	0.01	0.47	0.541	0.10	0.10
	2	0.06	0.02	0.48	0.55	0.10	0.10
	3	0.04	0.02	0.48	0.55	0.10	0.10
	4	0.38	0.36	0.75	0.94	0.32	0.44
	5	0.97	0.94	1.92	2.11	1.44	1.38
	6	1.19	1.16	4.79	4.39	2.76	2.32
$\mu$	RMS	0.43	<b>0.41</b>	2.41	2.66	<b>0.83</b>	<b>0.79</b>
$\sigma$	RMS	0.47	<b>0.46</b>	3.65	3.75	<b>1.04</b>	<b>0.93</b>

Disparities (1)none, (2), 1 pixel horizontal, (3) 1 pixel vertical, (4) 10 pixels horizontal, (5) sin horizontal, (6) sin horizontal and vertical.

A paper reporting on our preliminary results was presented at the Multicore Applications Research Community Conference in December 2012 [9].

## 5 Photometric Stereo System

### 5.1 Description

The purpose of the photometric stereo system is to reconstruct the 3D shape of a garment surface and to use this information with machine learning techniques to recognize the type of the textile fabric (wool, cotton, silk, etc). The system consists of a small camera synchronized with light sources, algorithms for 3D reconstruction and classification and a graphical user interface. It has small dimensions to mount on the gripper of the robot.

### 5.2 State of the realization

A prototype system has been developed using low cost off-the-shelf components. The camera is a small and light USB webcam with resolution of 640x480 and manual focus. The illumination subsystem is composed of 8 visible 3mm LEDs. The synchronization between the camera and the LEDs is ensured by using an Arduino Nano microcontroller board. The overall dimensions (without the controller) are 40mm diameter and 40mm length.

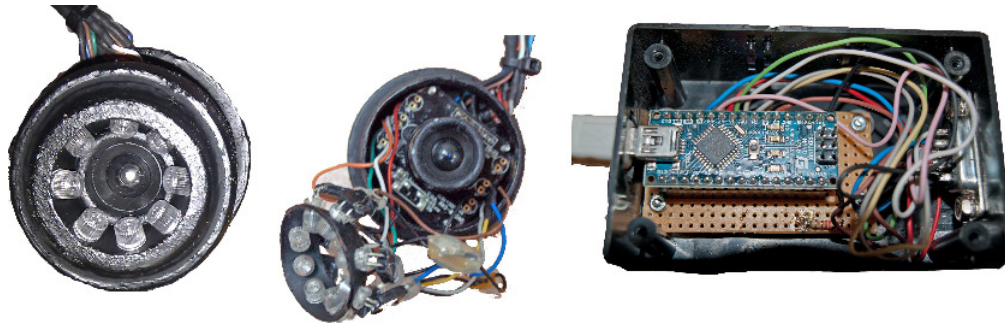


Figure 33: Photometric stereo prototype sensor and controller

Early versions of algorithms for photometric stereo and classification have been developed and tested. So far, all surfaces are assumed to be Lambertian and the illumination uniform. The classification system consists of a support vector machine with a radial basis function as kernel. To overcome the shallow depth of field and to achieve better focus, an algorithm for estimating the optimal image plane position has been tested but not yet integrated into the system. Additionally, a graphical user interface application has been built to visualize the 3D reconstructed surface and the classification result. Implementation of all algorithms is in C++.

### 5.3 Demonstration

The basic functionality of the photometric stereo system will be shown during the M12 demonstration. The sensor will not be placed on the gripper of the robot but it will be manually positioned on top of the unfolded garments lying on a table. The 3D reconstruction of the surface of garments will be shown on the graphical user interface application and

the classification algorithm will predict the garment materials. The focus functionality will be also demonstrated.

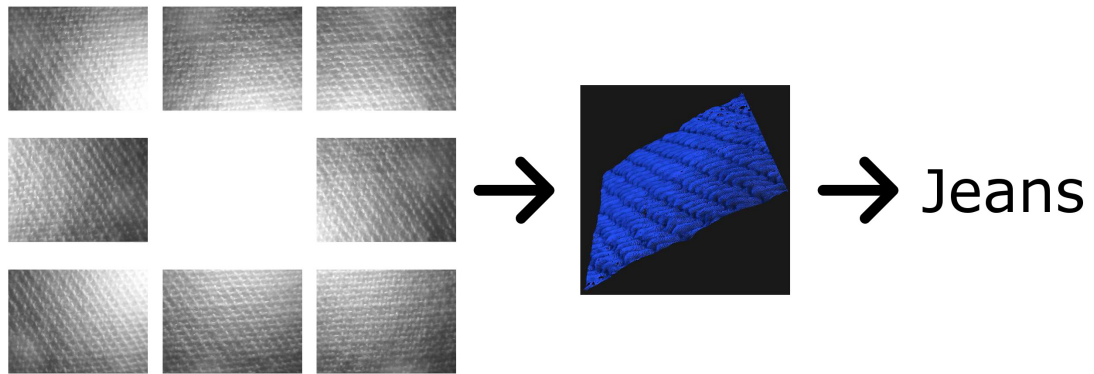


Figure 34: Photometric stereo reconstruction and classification

## 5.4 Performance

The system produces a 3D model of the shape of different textile fabric surfaces. Due to the fact that it is not a gauging device, reconstruction accuracy has not yet been evaluated extensively. The classification system performance has been assessed on the available photometric texture databases (PhoTex, ALOT) with promising results. However, the classification accuracy decreases when we use the photometric stereo sensor due to issues with the camera focus. Nevertheless, the preliminary results of the focus algorithm show that this issue can be dealt with.

## 6 World Representation, Perception and Action Planning

### 6.1 Perception

In deliverable D5.1 “Report on the Reasoning Mechanism and the Object Representation”, we presented an architecture for clothes garment recognition based on graph grammars. This architecture can handle the large variability of geometry and appearance by using an AND/OR graph representation. This proved to be quite robust in recognizing clothes assuming a large number of different configurations.

However, the current version of this architecture considers only the shape of a garment and especially its boundary. Furthermore, for simplicity, it is assumed that a garment is lying on a table and most of its parts are unfolded. For the time being, the graph grammar has been designed to handle only trousers and shirts.

During the review meeting, CERTH will present software with the implementation of the graph grammar algorithm. The software will take as input an arbitrary image of a shirt or trousers from CERTH’s database of clothes and display overlayed on their image the extracted primitives, the final inference of the type of garment and the corresponding score for each type. An example is illustrated in Figure 35. These results, although preliminary, seem promising for the successful solution of the clothes perception-recognition problem.



Figure 35: Example of primitive extraction and final rule based score assignment.

### 6.2 Garment recognition application

#### 6.2.1 Description

The purpose of the garment recognition application is to perform real time recognition of manipulated garments. The application is a C++ implementation of the garment recognition method based on inner distance shape contexts, presented in D5.1 [6]. A Graphical User Interface (GUI) has been designed in QT and is integrated with the application. In order to perform fast and robust extraction of the garment silhouettes, an Asus XtionPRO sensor has been employed providing both an RGB image and a depth map of the scene. The communication between the application and the XtionPRO device is accomplished through methods of the OpenNI library.

### 6.2.2 State of realization

Currently, the application has been successfully deployed, presenting the following functionalities. When it is initialized, a collection of binary images is loaded from disk and the contained silhouettes are used as matching prototypes. When the user activates the XtionPRO sensor, the manipulated garment is detected and matched to one of the prototypes. Garment detection is based on depth information, whereas the garment is assigned to the class of the prototype presenting the minimum matching cost.

Since any movement of the garment can alter its configuration, matching results vary over time. In order to present coherent results, recognition is updated only if a lower matching cost is achieved. However, when the matching cost is above a selected threshold, then the match is rejected and the garment is not recognized. The threshold is set to the minimum matching cost between prototypes of different classes.

An installation package for Windows platforms has been constructed and the application has been installed to a Windows 7 64-bit PC, with Intel(R) Core(TM) i7-2700K CPU at 3.50Ghz, 8GB RAM and NVIDIA GeForce GTX 560 display adaptor with 1GB of memory. An Asus XtionRRO device has been also installed to the PC and the resulting system presents real time performance.

### 6.2.3 Demonstration

Screen shots of a running instance of the application are provided in Figure 36. On the top screen is presented the application's initialization, where the prototypes' images are loaded and displayed to the user. On the bottom screen, the XtionPro sensor has been activated and two windows open. In the left window the depth map of the scene is displayed, whereas in the window on the right the detection and recognition results are presented. In the right window a red polygon is overlaid on the detected garment indicating its position and shape silhouette. The matching information is displayed in the top left corner. In this example a towel is successfully detected and recognized, during lowest point manipulation [6] by a human operator.

A short video produced by capturing the screen while the developed application was being tested on various garments can be found in the attached file: "RecognizeClothes-GUI.avi".

During the M12 demonstrator, a live demonstration of the garment recognition application will be performed. Different garments will be handled by a human operator applying lowest point manipulation and the system will provide the recognition results.

### 6.2.4 Performance

The implemented system processes 5 frames per second at VGA resolution, whereas the matching cost between the detected garment and a prototype is calculated in less than 25 ms. The system has been tested in a challenging environment with a noisy background, using different garments and 8 prototypes belonging to 6 different classes. Since the application is not yet embedded in a robotic system, the garments have been manipulated by a human operator. The operator applied lowest point manipulation, and in every case the minimum matching cost resulted in correct recognition.



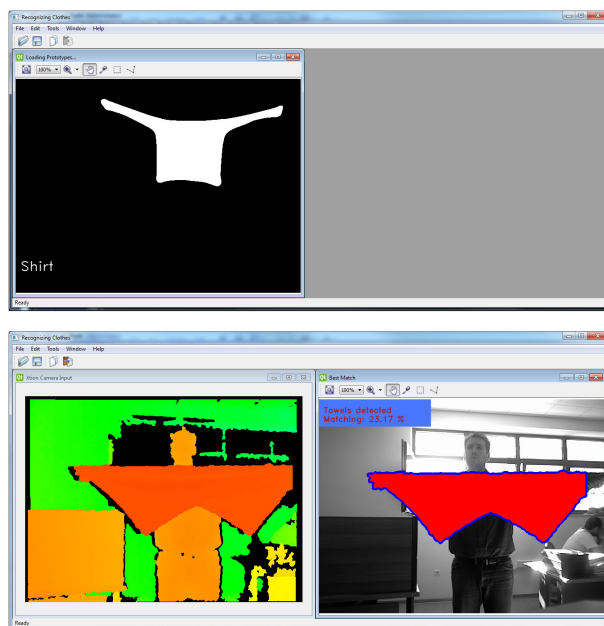


Figure 36: Screen shots of the developed garment recognition application. On launch the prototypes are loaded (top). On XtionPRO activation two windows open (bottom) showing the depth map of the scene (left) and the detection and recognition results (right).

## **6.3 Features extraction from a hanging garment**

### **6.3.1 Description**

The extraction of critical features on a piece of clothing, such as edges and junctions, provides useful information about its configuration and facilitates its manipulation. The feature extraction application, which is described in detail in deliverable D5.1, detects edges and junctions on a hanging clothing article using 3D information acquired from an Asus XtionPRO sensor. So, the input of the application is a depth image and the output a list of lines and points that correspond to edges, folds and corners. The application is implemented in C++, while methods from OpenCV and OpenNI libraries are used for the image processing and the acquisition of the image respectively.

### **6.3.2 State of realization**

At the moment, the application extracts edges and junctions of a hanging article, depicts them on an RGB image and returns a list of the corresponding lines and points. When an article that we wish to analyse is in a hanging position, a depth image is acquired and its edges are detected. According to their inclination, the edges are separated into four categories: 1) vertical, 2) edges that incline to the right, 3) edges that incline to the left, 4) horizontal. Due to the noise of the 3D data and the curvature that the edges of a fabric might have, there is a tolerance on the inclination of the edges of each category. Finally, the edges, according to their inclination, create junctions or are connected to each other to form longer edges. These features are overlaid on an RGB image of the towel in order to evaluate the results and a list of them is returned.

### **6.3.3 Demonstration**

An example of the feature extraction of a hanging towel is presented in figure 37. The detected edges are depicted with different colours according to their inclination and the junctions are marked with black dots.

A video depicting the feature extraction of clothes, manipulated by a human operator so as to be brought into various configurations, will be presented during the M12 demonstration. In order to visualize the results and to facilitate their evaluation, the edges will be highlighted with coloured lines and the junctions will be marked with black dots, as shown in figure 37.

### **6.3.4 Performance**

Although there is a tolerance to the noise of the 3D data during the edge detection, the noisy information acquired from Asus XtionPRO can create difficulties in the correct feature extraction. In addition, the depth difference created by an edge is not always sufficiently large, in order to overcome the limit set by the sensor's accuracy. As a result, edges or their parts might not be detected. The replacement of the Asus XtionPRO with a depth sensor with better precision is expected to improve the performance of the method.

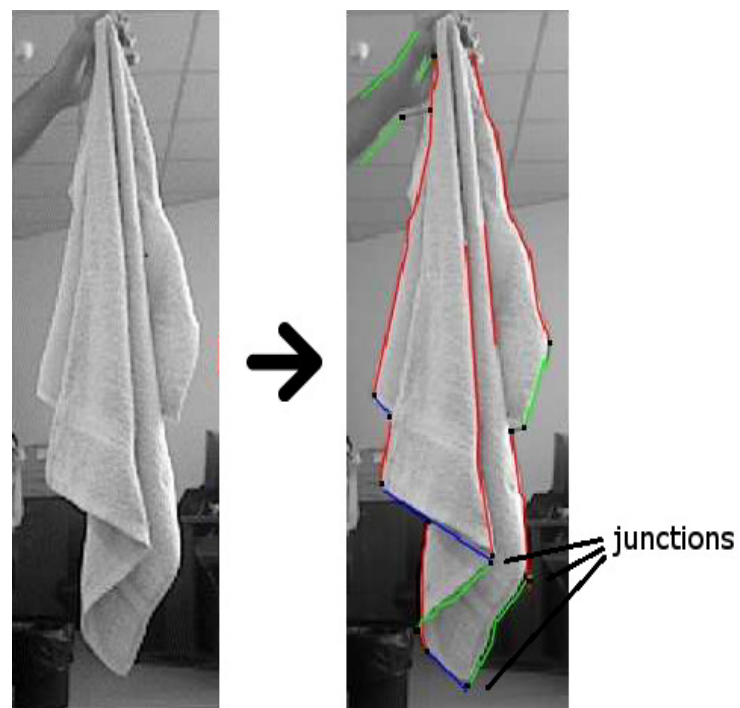


Figure 37: Detection of edges and junctions on a hanging towel

## 7 Demonstration of Integrated Motor-Sensory Guidance of the Robot

The basic mechanical and electrical construction of a dual arm manipulator was described in Deliverable 7.1 [4]. The robot cell was augmented by three Xtion range sensors mounted on the wrist of each arm as well as on the stereo head holder. In addition, the stereo head is also mounted at the top of the stereo head holder, which is fixed to the turning base of the robot. The stereo head is described in Deliverable 4.1 [3]. All subsystems were integrated into ROS on the node level.

The complex functionality of the system is documented by its ability to perform basic garment manipulation. A video clip which shows the ability of the system to grasp a piece of cloth dropped in a random position can be found in the attached file: 'Repeat\_grasp\_garment\_1.avi'. The video clip demonstrates the following functionalities:

- A T-shirt lies on the working table.
- The robot moves over the table and captures the depth map of the working area. Now the depth map is processed.
- The concavity (approximately equal to surface mean curvature) is computed for all points on the depth map. The concavity is displayed by colour in the projected depth map in ROS visualization environment.
- Points of maximal local concavity are selected by a non-maximum suppression algorithm. Principle directions based on principle curvatures are computed for selected points with high concavity.
- Orientation of the fold on T-shirt is estimated using principle directions. The fold defines the position and the orientation into which a gripper should orient while grasping.
- The system plans and realizes the required path. Finally, the gripper is in the grasping position and the gripper is closed.
- The T-shirt is lifted up and held. After a while, the robot drops the T-shirt and the T-shirt falls down on the working table and the process could be repeated.

To enhance the viewer's impression, we throw the T-shirt to the working table at the beginning. It demonstrates that the robot is able to grasp a piece of garment in a random position without making any special arrangements or assumptions.

The video clip demonstrates grasping and lifting a garment in front of the stereo head. This solved task represents a significant step towards the lowest point manipulation which was proposed in Deliverable 5.1 [6]. The result of the lowest point manipulation is grasped garment in specific hanging configuration, which can be used for recognition. The demonstration will be extended by implementing the algorithm described in [6] to realize the lowest point manipulation.

We have not yet reached the stage of folding a towel. However, our main present aim, i.e. demonstration of the robot control and manipulating ability, was reached. The towel folding will be easier when we have more sensory information available.

## 8 Conclusion

According to the Description of Work, the first demonstrator deliverable describes the individual components of the system, the state of their realization and future vision. Moreover, a demonstration of the current state of the system integration has been presented in this document. The system integration is an important part for verification of the components compatibility and it is a first step toward the second demonstrator which is expected in M24 of the project.

Six videos have been presented as demonstrations of the individual components of the CloPeMa project and as a demonstrator of the state of the system integration. The video files are distributed as digital attachments to this document.

## 9 List of Attachments

- Plan\_cd\_1.avi
- Plan\_cd\_2.avi
- Keep\_dist\_1.avi
- robotheadGUI.avi
- RecognizeClothesGUI.avi
- Repeat\_grasp\_garment\_1.avi

## References

- [1] DoW - Description of Work, European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288553, CloPeMa, August 2011.
- [2] Deliverable D1.1, European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288553, CloPeMa, May 2012.
- [3] Deliverable D4.1, European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288553, CloPeMa, November 2012.
- [4] Deliverable D7.1, European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288553, CloPeMa, October 2012.
- [5] Deliverable D2.1, European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288553, CloPeMa, January 2013.
- [6] Deliverable D5.1, European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no 288553, CloPeMa, January 2013.
- [7] Gerardo Aragon-Camarasa, Haitham Fattah, and J Paul Siebert. Towards a unified visual framework in a binocular active robot vision system. *Robotics and Autonomous Systems*, 58(3):276–286, March 2010.
- [8] G. Cannata, M. Maggiali, G. Metta, and G. Sandini. An embedded artificial skin for humanoid robots. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 434–438, aug. 2008.
- [9] W.P. Cockshott, S. Oehler, G. Aragon Camarasa, J. Siebert, and T. Xu. A parallel stereo vision algorithm. In *Many-Core Applications Research Community Symposium 2012*, 2012.
- [10] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. <http://ompl.kavrakilab.org>.
- [11] Mark R. Cutkosky and Robert D. Howe. Human grasp choice and robotic grasp analysis. In Subramanian T. Venkataraman and Thea Iberall, editors, *Dextrous Robot Hands*, pages 5–31. Springer New York, New York, NY, USA, January 1990.
- [12] Roberta L. Klatzky and Susan Lederman. Intelligent exploration by the human hand. In Subramanian T. Venkataraman and Thea Iberall, editors, *Dextrous Robot Hands*, pages 66–81. Springer New York, New York, NY, USA, January 1990.
- [13] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta. Methods and technologies for the implementation of large-scale robot tactile sensors. *Robotics, IEEE Transactions on*, 27(3):389–400, june 2011.



- [14] P.M. Sharkey, D.W. Murray, and J.J. Heuring. On the kinematics of robot heads. *Robotics and Automation, IEEE Transactions on*, 13(3):437 –442, jun 1997.
- [15] R.Y. Tsai and R.K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *Robotics and Automation, IEEE Transactions on*, 5(3):345 –358, jun 1989.
- [16] Changchang Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). <http://cs.unc.edu/ccwu/siftgpu>, 2007.