



A Manipulation Control Strategy for Granular Materials Based on a Gaussian Mixture Model

Carlos Mateo, Juan Antonio Corrales Ramon, Youcef Mezouar

► To cite this version:

Carlos Mateo, Juan Antonio Corrales Ramon, Youcef Mezouar. A Manipulation Control Strategy for Granular Materials Based on a Gaussian Mixture Model. Conference Proceedings of "Robot 2019: Fourth Iberian Robotics Conference", Springer, pp.171-183, 2020, 978-3-030-36149-5. 10.1007/978-3-030-36150-1_15 . hal-02379943

HAL Id: hal-02379943

<https://uca.hal.science/hal-02379943>

Submitted on 25 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Manipulation Control Strategy for Granular Materials based on a Gaussian Mixture Model

Carlos M. Mateo, Juan Antonio Corrales, and Youcef Mezouar

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000
Clermont-Ferrand, France,
`cmateoagul@sigma-clermont.fr`

Abstract. In the context of metal additive manufacturing, one of the most attractive tasks to be robotized is the cleaning process of metal powder after the printing operations. This task presents a challenging scenario for most of robot manipulation approaches in the literature. In this paper we present an approach, marker-less and real time affordable, which address the cleaning problem like a shape manipulation control problem. This control strategy is designed as an optimization problem. The error function is written as a lagrangian function using an objective function based on Gaussian Mixture Model (GMM). The local optimization is performed by a gradient descent and a global optimization process is used to avoid local minima.

Keywords: robot control, image processing, manipulation, granular material, optimization, non-rigid registration

1 Introduction

Robotize cleaning process in metal additive manufacturing nowadays is a hot topic because this process is still handmade, regardless, the manipulated material is an unhealthy substance for people. Metal powder can be easily inhaled and absorbed by cardiovascular system, producing serious health disorders. In addition, this material also presents explosive behavior in contact with oxygen, making the manipulation of this material unsafe [1]. Moreover, during the printing process of pieces, it is wasted more than half of the source product since just a few parts of the powder material is melted. Therefore, we make safer and ecological additive industry by robotizing this task since we avoid involve human operators in risky situation and the rates of re-usability and recycling are increased.

We define the cleaning task as a close-loop form control strategy to drive the robot to the parts to be cleaned while avoiding obstacles. Specially, we consider the operation of removing powder as a soft manipulation task where the powder bed plus the printed piece plus the powder is a whole object (Fig. 1). So, here the goal is to transform the current object shape (cube of powder with the object inside) into a desired one (the object without any powder). During last years

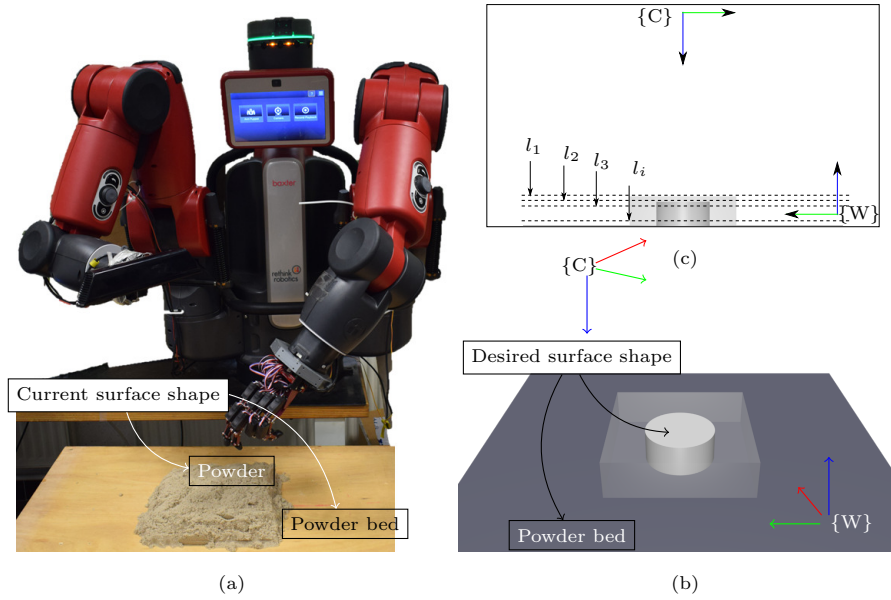


Fig. 1: Platform setup where (a) shows the real robot in front of a table with one object occluded inside the powder. Image (b) represents the virtual scenario with object, solid cylinder, which is covered by the powder, transparent cube. (c) Front view of the virtual scenario.

in the literature, we find a few approaches aimed to resolve concrete problems in object (granular material) manipulation tasks, such as [2]. Others like, the work [3] by Sanchez et al. makes a review of advances in robotic manipulation task of non-rigid objects. Works [4, 5] by Mateo et al. present a method for understanding how the surface changes during manipulation task, implementing a Dijkstra-based method to model the deformation of the object. The authors also present a method in [6] to predict when the object surface changes drastically during manipulation tasks to achieve dexterous manipulation and prevent damages in objects. The previous cited works are focused mainly in the perception of the objects, but works more related to the control of robots for the manipulation of elastic objects are [7, 8] by Navarro-Alarcon, in which an strategy to control the shape of objects with robots, using Fourier transform as features, is presented. In this line, works like [9, 10] realize similar tasks, where Cherubini in [9] presents a study to understand how to address the dexterous manipulation of granular materials using robot hands without a previous planning. At the same time Schenck et al. in [10] present a deep learning strategy to teach robots to address the problem presented by Cherubini.

The contribution of this work lies in the proposal methodology (strategy) to resolve the problem of powder removal in metal additive manufacturing. Being

the goal of this strategy the transformation of the current shape of the scenario to the desired one. The methodology combines visual perception techniques, optimization methods and robot control schemes. The robot should be moved to minimize the disparity between the current shape and the target shape. We decided to use a strategy to drive the robot to those surface points with maximum disparity and at the same time closest to the current robot position. Here, the disparity is computed using a registration point set method. In the literature there are works which address the problem of registering two different non-rigid surfaces: Coherent Point Drift (CPD) [11]; Volume Deformation [12]; or Killing Fusion [13]. All these works follow the same key concept, retrieve the transformation (deformation) which map one surface to another as rigidly as possible. The control of the robot is formalized using a gradient descent algorithm to optimize the robot velocity [14, 15], expressed in \mathbb{R}^3 manifold.

2 Strategy Definition

This approach is devoted to defining the control policy for robotic manipulation tasks in cleaning operations. This method continuously controls a robot until the task is done, without stops for sensing neither re-planning. We consider finished the task when the current scenario shape matches with a reference (or target) scenario. The diagram of the method is pictured in Fig. 2. As is illustrated in the flowchart, the framework splits the approach into three different blocks: perception; and control; level planning. We use the visual perception part to generate the control references (attractors points) used in the robot control. These emerge from the disparity map \mathbf{W} , computed in the block (e) in Fig. 2, by using the reference \mathcal{D}^* and current \mathcal{D} depth images of the scenario.

We use a 3D model to define the target scenario, i.e. a mesh composed by the platform (powder bed) where the printed piece will be built and the piece itself (Fig. 1 (b) flat surface and cylinder). Using the 3D model, we create a virtual depth image \mathcal{D}^* . By a ray tracing technique using a virtual camera placed in the same pose than the real one, a top-view configuration. Where the camera z-axis (optical view axis) is aligned with the z-axis of the world reference frame (Fig. 1 (c)). We use as world frame the bottom-right corner of the scenario (Fig. 1 (b)).

Once we have the set of projected points $\mathbf{x} \in \mathcal{D}^*$ and $\mathbf{y} \in \mathcal{D}$ in image space, we re-project this set into the euclidean space as the set of N points $\mathbf{X} \in \mathbb{R}^{N \times 3}$ and $\mathbf{Y} \in \mathbb{R}^{N \times 3}$. This is done by the well-known re-projection equation $\mathbf{X} = \mathbf{P}^+ \mathbf{x}$ where the projection matrix $\mathbf{P} = \mathbf{K} [\mathbb{I}_{3 \times 3} \mid \mathbf{0}_{3 \times 1}]$ is composed by the extended intrinsic camera parameters matrix $\mathbf{K}_{4 \times 3}$, and a matrix 3×4 , extrinsic camera parameters, where the first block is an identity matrix $\mathbb{I}_{3 \times 3}$ and the last column $\mathbf{0}_{3 \times 1}$ (a zero column). The Moore-Penrose pseudo-inverse projection matrix is computed as $\mathbf{P}^+ = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T$. Equally for the current projected points $\mathbf{y} \in \mathcal{D}$ to obtain \mathbf{Y} . To both point sets (point clouds) \mathbf{X} and \mathbf{Y} , we execute a pass-through filter $f_p : \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}^{M \times 3}$, where $M < N$, (Fig. 2 (a) and (b)) to select those points which lie in the current depth layer l_i (Fig. 1 (c)), according with,

$$f_p(\mathbf{X}, z_{low}, z_{up}) = \{X \mid z_X > z_{low} \wedge z_X < z_{up}\}, \quad (1)$$

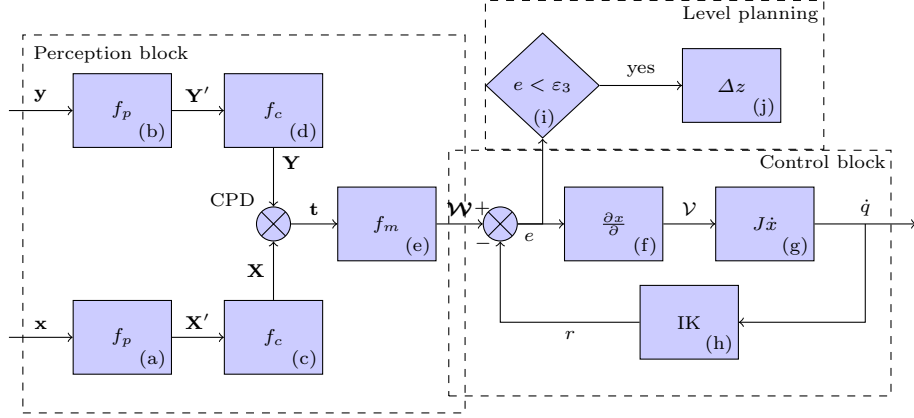


Fig. 2: Method flowchart. The perception task follows an open-loop scheme where is being compared current and target scenario. Control part is designed as a close-loop for minimizing the error between current and target robot position.

where z_X is the z component of any point in $\{X_1, X_2, \dots, X_M\} \in \mathbf{X}$.

We use the Canny edge filter f_c [16] to extract the contours of the silhouette, in the image projection of the survivor points \mathbf{X}' and \mathbf{Y}' in equation 1. Function f_c (Fig. 2 (c) and (d)) works in image space, thus \mathbf{x}' and \mathbf{y}' are projected before to apply this filter. We use $\mathbf{x}' = \mathbf{P}\mathbf{X}'$ (and $\mathbf{y}' = \mathbf{P}\mathbf{Y}'$) relation to project back the 3D points into image plane. Thus, final \mathbf{X} and \mathbf{Y} point sets are composed by the point re-projection of the sets $f_c(\mathbf{x}')$ and $f_c(\mathbf{y}')$. We compute the error between the two final point sets \mathbf{X} and \mathbf{Y} using Coherent Point Drift (CPD) operator [17, 11] (Fig. 2). CPD is a 3D registration algorithm aimed to resolve the problem of 3D non-rigid alignment. This operation returns a set of transformations $\mathbf{t} \in \mathbb{R}^{3 \times N}$, that represent the translations between each pair of correlated points.

2.1 Map Function f_m

Disparity field \mathbf{W} is used in this work to balance the importance of each boundary point \mathbf{Y} according to the robot pose r and the visual feedback \mathbf{t} . So, we purpose the following mapping function $f_m : \mathbb{R}^{N \times 3} \rightarrow \mathbb{R}^N$ subject to the robot pose r and \mathbf{t} to get \mathbf{W} (Fig. 2 block (e)). Thus, the fundamental idea of this function is to define a reachability rank of points. That is, weight with the maximum likelihood the point with better ratio between: distance to r ; and maximum disparity between current $Y \in \mathbb{R}^3$ and the desired $X \in \mathbb{R}^3$ position. Note that here, we keep using the correlation space between \mathbf{X} and \mathbf{Y} obtained after applying CPD algorithm beside robot position r .

Mathematically, we can express the mapping function $f_m(\mathbf{Y} \mid r, \mathbf{t})$ as a multiplication of two different weights terms $\mathbf{W} = \mathbf{W}_t \mathbf{W}_d$. Where \mathbf{W}_d is the vector of weights according to the robot pose r (i.e. distance between point in the

current surface and the robot) and \mathcal{W}_t with the visual feedback (i.e. distance between current and desired surfaces). Then, \mathcal{W}_t is formally expressed like the normalized vector according with infinity norm, i.e. the set of $\hat{\mathbf{t}}$ is weighted with respect to the maximum $\hat{\mathbf{t}}$. This is because we emphasize those points with more transformation, formally this is as follows,

$$\hat{\mathbf{t}} = \|\mathbf{t}\|_2, \quad \mathcal{W}_t = \frac{\hat{\mathbf{t}}}{\|\hat{\mathbf{t}}\|_\infty} \quad (2)$$

where $\hat{\mathbf{t}}$ is a vector in \mathbb{R}^N stacking all euclidean norm of all elements in \mathbf{t} . On the other hand, \mathcal{W}_d is the normalized distance vector $\mathbf{d} = \mathbf{Y} - r$ according with the negative infinity norm, i.e. the set $\hat{\mathbf{d}}$ is balanced w.r.t. the minimum $\hat{\mathbf{d}}$. This is because we look for the closest point,

$$\hat{\mathbf{d}} = \|\mathbf{d}\|_2, \quad \mathcal{W}_d = \frac{\hat{\mathbf{d}}}{\|\hat{\mathbf{d}}\|_{-\infty}}. \quad (3)$$

Values in Equation 2 and 3 are bounded in the range $(0, 1]$, thus \mathcal{W} is also bounded in the same range. Where the most relevant point $Y_i \in \mathbf{Y}$ is valued with a 1.

2.2 Objective Function Definition

At this point all filter, registration and mapping functions used in the perception block are presented. This subsection is devoted to present the objective function used in the control law which is shown in Fig. 2.

Control here is formalized as an optimization problem where the parameters that minimize the objective function should be found. In our case, the parameter that we want to optimize is the robot velocity. The goal (or objective) is to move the robot r to the maximum-interesting points \mathcal{W} , discussed above. Our objective function is constrained to the fact that the robot does not have to cross over the desired shape \mathbf{X} . This constraint is imposed because \mathbf{X} is the contour of the solid 3D printed object plus the machine support (powder bed Fig. 1), therefore crossing over this area means a collision between the robot and the scene.

To convert a constrained into an unconstrained problem, we use the Lagrange multipliers method [18]. Using this function, we can find the local minimal of our objective function subject to our constraint,

$$\mathcal{L}(r, w) = f(r) - wg(r | \mathbf{X}) \quad (4)$$

where, $f(r)$ is the objective function, $g(r | \mathbf{X})$ is the equality constraint and w (free parameter, in our experimentation $w = \frac{1}{2}$) is a weight to balance the relevance of g . Note that in \mathcal{L} is still differentiable.

Because the intrinsically noisy nature of image sensors, we define the objective function using a Mixture Model (MM) function. Concretely the MM used here is the Gaussian Mixture Model GMM,

$$f(r) = -\log \sum_i^N \alpha_i \mathcal{N}(r | Y_i, \sigma), \quad (5)$$

where $\mathcal{N}(r | Y_i, \sigma)$ is a Gaussian distribution function centered in the point y_i with a spread factor of σ and a peak value equal to α_i . Note that there is a Gaussian for each point in the current contour. Gaussian's peak in GMM is treated as the prior-knowledge (in our case \mathbf{W}) of the i -th distribution and is bounded between $[0, 1]$. In the other hand, constraint equation g is defined in terms of an exponential artificial field,

$$g(r | \mathbf{X}) \equiv \mathcal{E}(r | \mathbf{X}). \quad (6)$$

Note that we do not use the Gaussian Mixture Model to represent the position of the robot but to relate the current robot pose with respect to the points in the current surface shape \mathbf{Y} . Also, MM is used to deal with the intrinsic uncertainty of visual information.

Following section shows more detailed develop about objective function f , constraint g and lagrangian \mathcal{L} .

3 Task-Space Robotic Control

The most straightforward approach to move a robot is to design a velocity controller which relates the error function variations during time \dot{e} with the robot velocity $\mathcal{V} = [v, \omega]$, Chaumette and Hutchinson in [19]. Formally, this is written as $\dot{e} = \nabla e \mathcal{V}$, where ∇e is the gradient of the error function e , $v \in \mathbb{R}^3$ is the linear velocity and $\omega \in \mathbb{R}^3$ the angular velocity, thus the velocity vector is in $\mathcal{V} \in \mathbb{R}^6$.

In this work the error function is equivalent to the Lagrangian equation 4, $e \equiv \mathcal{L}(r, w)$. Considering that the hidden idea is to minimize the error ensuring that the decreasing follows a monotonic exponential function with respect to time $\dot{e} = -\lambda_e e$, being the controller gain λ_e an scalar positive constant, the obtained expression is as follow,

$$\mathcal{V} = -\lambda_e \nabla e^+ e. \quad (7)$$

where ∇e^+ is the Moore-Penrose pseudo-inverse gradient matrix.

We extend the control law (equation 7) to optimize the spread σ of \mathcal{N} while the robot velocity \mathcal{V} . Therefore, our final velocity vector $\bar{\mathcal{V}} \equiv [\mathcal{V}, \dot{\sigma}] \in \mathbb{R}^7$ is composed by \mathcal{V} and the gaussian's spread time variation $\dot{\sigma}$. Consequently, the gradient ∇e is differentiated also with respect to σ , not just w.r.t. the robot pose. This is because by controlling the spread of GMM variance, we are controlling the velocity of the robot. The idea is to situate the robot pose in the slop of the GMM where it is the maximum value variation of the function and therefore the gradient have the maximum value.

3.1 Error definition

As it has been introduced before, removing particle media from a scene can be considered as a surface shape control problem. Then, controlling the difference between desired \mathbf{X} and current \mathbf{Y} shapes can be expressed as the optimization of an error function e . Moreover, the error function e , as we discussed above, is equivalent to the lagrangian operator \mathcal{L} , defined in equation 4.

Objective function f in \mathcal{L} is defined using a GMM in equation 5, where the i -th gaussian distribution $\mathcal{N}(r | Y_i, \sigma)$ is defined as,

$$\mathcal{N}(\mathbf{r} | Y_i, \sigma) = \exp \left(-\frac{1}{2} \frac{\|r - Y_i\|_2^2}{\sigma^2} \right), \quad (8)$$

and the prior-knowledge (guassian's peak) of the i -th gaussian is equivalent to $\alpha_i \equiv \mathcal{W}_i$. The potential field term \mathcal{E} is designed using a sigmoid function and is defined as follows,

$$\mathcal{E}(r | \mathbf{X}) = -\frac{1}{1 + \exp(-\|r - \mathbf{X}\|_{-\infty})} \quad (9)$$

As the equation shows this potential barrier is based on the minimum distance between any point of the desired surface \mathbf{X} and current robot pose of r . Therefore, the potential barrier is going to be governed just by the closest desired contour point X_i to r .

Then, by injecting equation 8 and 9 in 4 we obtain our error function as,

$$e = -\log \left[\sum_i^M \mathcal{W}_i \exp \left(-\frac{1}{2} \frac{\|r - Y_i\|_2^2}{\sigma^2} \right) \right] + w \frac{1}{1 + \exp(-\|r - \mathbf{X}\|_{-\infty})} \quad (10)$$

Equation 10 represents the error function of our method. This formulation can be seen has a problem of regularization where we have two different forces: attraction (first term); and repulsive (second term). As it is expressed in the first part of the equation, as smaller is the difference between robot pose with the points in current surface shape $\|r - Y_i\|_2$, smaller is the energy of the error function. In contrast the second term as smaller is the distance between the robot pose and the point in desired surface shape $-\|r - \mathbf{X}\|_{-\infty}$, bigger is the energy of the error function.

4 Global robot pose Optimization & Level planning

We use the Deterministic annealing algorithm [20] to govern the global behavior of our control scheme, while the control scheme is devoted to walk towards the most interesting points in the current surface according with \mathcal{W} . The global optimization strategy prevents that the system falls in stationary state. These stationary states are reached when the robot position is too far from the influence of f or too close to the GMM's peak. This global optimization is as follows,

$$\sigma_{t+1} = \begin{cases} \lambda_\sigma \sigma_t, & e < \varepsilon_1 \\ \lambda_\sigma^{-1} \sigma_t, & e > \varepsilon_2 \end{cases} \quad (11)$$

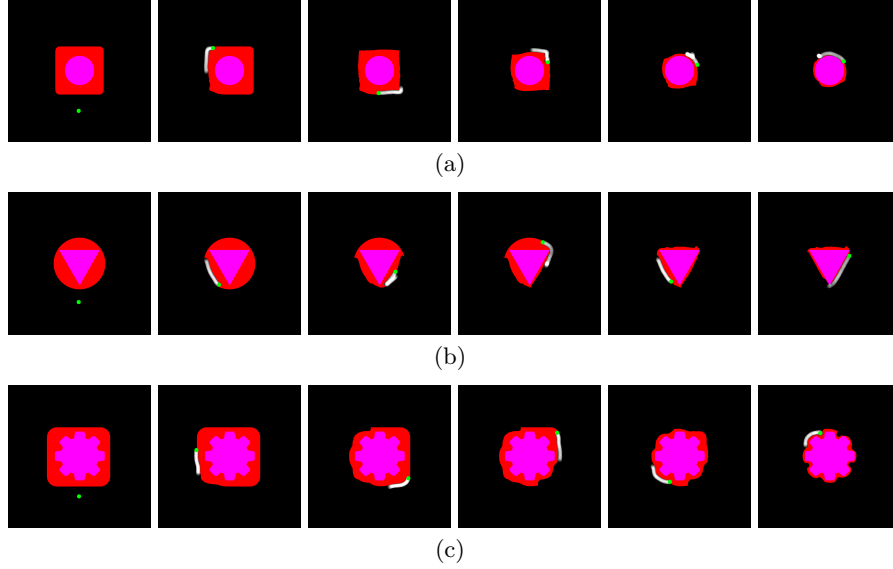


Fig. 3: Experiments over 3 different scenarios. Green dot represents robot pose, path is represented in white color, current surface is the red surface and purple is the desired surface. Note that the purple area is a restricted area for the robot.

where λ_σ is the global gain of the system and ε_1 and ε_2 are the thresholds used for activating the global optimization.

In the other hand, the global evolution of the cleaning task $l_1 \rightarrow l_2 \rightarrow \dots \rightarrow l_i$ evolves by removing layers of powder (Fig. 1). The method flowchart as a level planning block (Fig. 2 block (i)) aimed to move the robot in z -direction in the cleaning task. Then, the depth range are $z_{low} = z_{low} + \Delta z$ and $z_{up} = z_{up} + \Delta z$ (this is a range area surrounding a level l_i) if the differences between current and target surfaces is lower than a desired value $\|\mathbf{X} - \mathbf{Y}\|_2 < \varepsilon_3$.

5 Experiments

Here, a set of 3 different simulations are designed (Fig 3), assuming the robot should clean in z direction, i.e. layer by layer, where each test represents a depth map in a specific layer. This strategy is chosen because it reduces drastically the point set density (because only those points in the silhouette of the shape are considered) in comparison to take all the points in the 3D surface, hence the computational time is reduced proportionally. We first evaluate the characteristics of the proposed method using simulations. And secondly, we have implemented this strategy in a real robot to prove the viability. This approach has 2 different gains: one for the local optimization λ_e ; and the other for the global optimization λ_σ . Moreover, this scheme has three bound parameters ε_1 ,

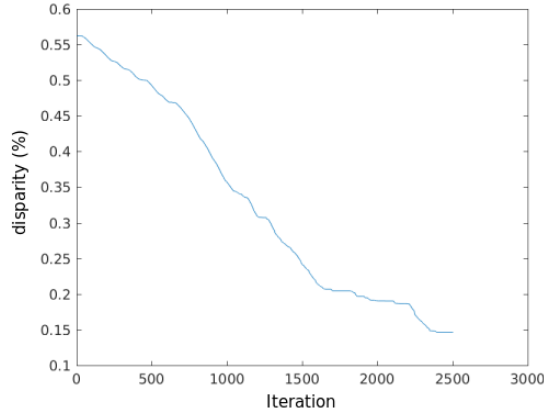


Fig. 4: The disparity between shapes evolution over time. x-axis represents the iteration

ε_2 and ε_3 , two first to keep e in a desired level and the last to know when a depth level is cleaned (or deformed). Due the fact that this strategy comprised a global optimization step to bound the error function inside an error levels, this method cannot be evaluated directly on the study of the error evolution. Therefore, the convergence is analyzed based on the shape similarities evolution between current and target shapes.

We found that the behavior of system is suitable for each test, doing just a single set up of these parameters. The gains for error minimization and variance optimization are $\lambda_e = 0.03$ and $\lambda_\sigma = 200$. The limits for error function in this experimentation are set in the range $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.15$. Additionally, we have initialized the value of GMM spread to $\sigma = 0.005$. One of the advantages of this method is the smooth transformation that generates. This ability can be appreciated in Fig. 3(a) where the shape is transformed from a square to a circle. This kind of behavior is progressively doing the task while avoiding problems of local minimum. In this work, we consider that the robot is in a local minimum when this is stop and still remind powder to remove. Although it is not common, we still have problem with local minima in concave shapes. In other hand, the Fig. 3(b) presents a challenging situation where three unconnected powder areas should be removed. This test shows how the system deals with local tasks to get a global solution. The experiment 3(c)) was designed to demonstrate how the controller manages to convert a simple shape into a much more complex shape (concave shape).

It is demonstrated for all experiments that the system presents a local and global behavior. All experiments complete their task after approximately 2500 iterations. The time to process a single iteration depends on the amount of points in the silhouette. More concretely the relation between the number of points and the time to perform the task is linear. In these experiments, the

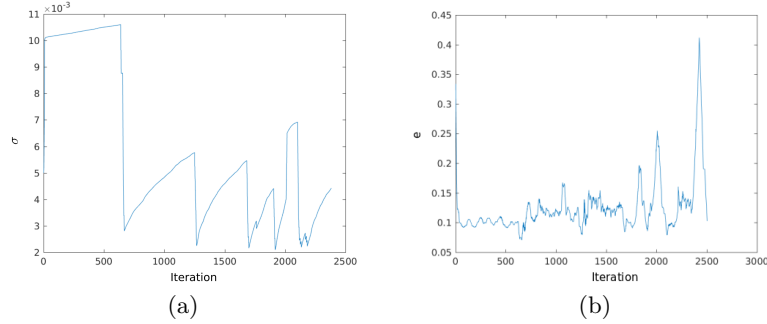


Fig. 5: Evolution over the time (x-axis) of: (a) GMM σ , y-axis; and (b) error e function, y-axis

silhouette is composed by point set of around 300 elements and the time to compute an iteration is approximated 0.01s. As it is mentioned before, the system convergence is achieved when the disparity between surfaces (ranged in $[0, 1]$) is less than a desired value. A desired disparity of 0.015 was set for the experiments. Fig 4 shows the evolution (monotonic decrease) of disparity for experiment 3 (Fig. 3(c)). In the other hand, the associated error evolution is presented in Fig 5(b) and shows how this error falls until a bounded error is achieved and during all the task the system works to keep this error inside the limits. Fig. 5 shows global optimization evolution according to the plot presented in Fig. 5(b). GMM variance evolution is plotted in Fig. 5(a) and it can be observed how the σ is being progressively update using local minimum optimization until, approx. iteration 600, when error value (Fig 5(b)) fall down the low limit. Then the value of σ is globally updated, to recover rapidly the robot velocity, according to the equation 11 (Fig. 5(a)). And again, the GMM variance is locally updated.

Finally, we implement this approach over a real platform to demonstrate its applicability in real scenarios (Fig. 6). We use the collaborative robot Baxter equipped with a Kinect camera in the end effector of the right arm and an AR10 humanoid hand in the left arm. We decided to use a hard actuation (touching) over the material (kinetic sand) to avoid uncertainties produced by soft actuation (vacuum aspiration). Fig. 6 presents a series of key frames of the manipulation task. This experiment is following the described trajectory in the third simulated test (Fig. 3(c)).

6 Conclusions

This work presents a novel approach to manipulate granular materials using a robotic manipulator. Here, it is defined the cleaning tasks problems as a shape deformation problem. The goal is reached using our proposed control scheme based on a Gaussian Mixture Model. One of the major contributions of this work is the novel control scheme which combines local and global optimization

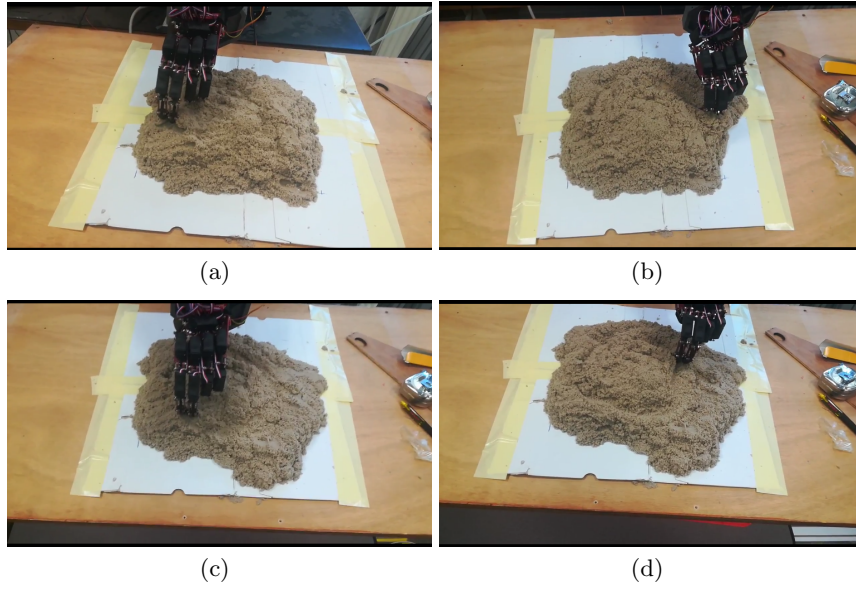


Fig. 6: Real robot experiment. Hard actuation over kinetic sand.

by means of a gradient descent and a Deterministic Annealing method. Another important contribution is that here the task control is not defined in terms of error evolution. The error evolution here is just used to control the robot velocity. In contrast here, task evolution is defined in terms of surface shape disparity (Fig. 4). The experiments show that this method reach solutions in real time when the point density of silhouette contour is low.

But this approach cannot guaranty an optimal solution when desired shape has complex topology (many concave shapes). In this work we initialize the gaussian distribution's spread although this can be discovered by the global optimization algorithm. We do this because as near these values are to the optimal ones, the convergence is faster. This fact opens the possibility to extend this method to predict optimal values of σ by prior clues using learning strategies. Similarly, the definitions of boundaries in the energy function can be improved by using another source of information which predicts best boundaries for each problem.

Acknowledgements Research supported by project CoMManDIA (SOE2/ P1/ F0638) which is cofinanced by Interreg Sudoe Programme (European Regional Development Fun), and by the Chair Industrial of SIGMA Clermont Chaire SIG-AM “Fabrication Additive Multisectorielle”.

References

1. Gibson, I., Rosen, D., Stucker, B.: Additive Manufacturing Technologies (2015)

2. Balatti, P., Kanoulas, D., Tsagarakis, N.G., Ajoudani, A.: Towards Robot Interaction Autonomy : Explore , Identify , and Interact (April), 9523–9529 (2019)
3. Sanchez, J., Corrales, J.A., Bouzgarrou, B.C., Mezouar, Y.: Robotic manipulation and Sensing of Deformable Objects in Domestic and Industrial Applications: A Survey. *Journal of Robotics Research* pp. 1–34 (2018)
4. Mateo, C.M., Gil, P., Torres, F.: Computation of curvature skeleton to measure deformations in surfaces. In: *Lecture Notes in Electrical Engineering*, vol. 383, pp. 197–207. Springer, Cham (2016)
5. Gil, P., Mateo, C.M., Delgado, Á., Torres, F.: Visual/Tactile sensing to monitor grasps with robot-hand for planar elastic objects. In: *47th International Symposium on Robotics, ISR 2016*. vol. 2016, pp. 439–445 (2016)
6. Mateo, C.M., Gil, P., Torres, F.: 3D visual data-driven spatiotemporal deformations for non-rigid object grasping using robot hands. *Sensors (Switzerland)* 16(5) (2016)
7. Navarro-alarcon, D.: A Depth-Based Algorithm for Manipulating Deformable Objects Using Smooth Parametric Surfaces and Energy Minimisation (April), 33–34 (2018)
8. Navarro-alarcon, D.: Visual Shape Servoing of Deformable Objects : The Fundamentals. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. vol. 18, p. 5386 (2017)
9. Cherubini, A., Leitner, J., Ortenzi, V., Corke, P.: Towards vision-based manipulation of plastic materials (2018)
10. Schenck, C., Tompson, J., Fox, D., Levine, S.: Learning Robotic Manipulation of Granular Media. *Proceedings of Machine Learning Research* 78(Conference on Robot Learning (CoRL)) (2017)
11. Myronenko, A., Myronenko, A., Song, X., Song, X., Carreira-Perpiñán, M.Á., Carreira-Perpiñán, M.Á.: Non-rigid point set registration: Coherent Point Drift. In: *Advances in Neural Information Processing Systems* 19. pp. 1009–1016. No. September (2007)
12. Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., Stamminger, M.: Volumedeform: Real-time volumetric non-rigid reconstruction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9912 LNCS, 362–379 (2016)
13. Slavcheva, M., Baust, M., Cremers, D., Ilic, S.: KillingFusion: Non-rigid 3D Reconstruction without Correspondences. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (4), 1386–1395 (2017)
14. Corke, P.: Corke, Peter. *Robotics, vision and control: fundamental algorithms* In *MATLAB®*, vol. 118. Springer International Publishing, 2nd editio edn. (2017)
15. Lynch, K.K.M., Park, F.C.: *Modern robotics : mechanics, planning, and control*. No. May, Cambridge University Press (2017)
16. Canny J.: A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8*(6), 679–698 (1986)
17. Myronenko, A., Song, X.: Point set registration: Coherent point drifts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12), 2262–2275 (2010)
18. Nocedal, J., Wright, S.J.: *Numerical Optimization*. Springer Science & Business Media (2006)
19. Chaumette, F., Hutchinson, S.: Visual servo control. I. Basic approaches. *IEEE Robotics and Automation Magazine* 13(4), 82–90 (dec 2006)
20. Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. *Neural Networks* 11(2), 271–282 (1998)