Jacquard: A Large Scale Dataset for Robotic Grasp Detection

Amaury Depierre, Emmanuel Dellandréa, Liming Chen

Jacquard: A Large Scale Dataset for Robotic Grasp Detection

Amaury Depierre[1,2], Emmanuel Dellandréa[2] and Liming Chen[2]

***Abstract*— Grasping skill is a major ability that a wide number of real-life applications require for robotisation. State-of-the-art robotic grasping methods perform prediction of object grasp locations based on deep neural networks. However, such networks require huge amount of labeled data for training making this approach often impracticable in robotics. In this paper, we propose a method to generate a large scale synthetic dataset with ground truth, which we refer to as the Jacquard grasping dataset. Jacquard is built on a subset of ShapeNet, a large CAD models dataset, and contains both RGB-D images and annotations of successful grasping positions based on grasp attempts performed in a simulated environment. We carried out experiments using an off-the-shelf CNN, with three different evaluation metrics, including real grasping robot trials. The results show that Jacquard enables much better generalization skills than a human labeled dataset thanks to its diversity of objects and grasping positions. For the purpose of reproducible research in robotics, we are releasing along with the Jacquard dataset a web interface for researchers to evaluate the successfulness of their grasping position detections using our dataset.**

I. INTRODUCTION

Despite being a very simple and intuitive action for a human, grasp planning is quite a hard task for a robotic system. Detecting potential grasp for a parallel plate gripper from images involves segmenting the image into objects, understanding their shapes and mass distributions and eventually sending coordinates to the robot's actuator. As the whole trajectory of the arm and its end position depend on these coordinates, precision is critical and an error of one pixel in the prediction can make the difference between success and failure of the grasping. Because of these difficulties and despite the progress made recently, performance for this task is still far from what we could expect for real-case applications.

State-of-the-art methods to predict a grasping position for a parallel plate gripper from visual data rely on deep neural networks trained either to directly predict a grasp [1] or to evaluate the quality of previously generated candidates and select the best one [2]. These methods rely on supervised training based on labeled data, which may be obtained through one of the following techniques: human labeling, physical trials with robots [3] [4], analytic computation where a model is used to predict the effect of external forces applied on the model [5] and physics simulation for which the grasp is performed in a computer simulation of the real world [6]. The first two methods, despite being the most accurate, are very time-consuming and therefore cannot be easily used to generate very large datasets. The last two, on the other hand, can be used quite easily to generate millions of labeled data, but generally require to match the CAD model to the position of the object in the image to be efficient.

Fig. 1. Jacquard dataset contains a large diversity of objects, each with multiple labeled grasps on realistic images. Grasps are drawn as 2D rectangles on the image, darker sides indicate the position of the jaws.

In this paper, we present an approach to automatize the generation of labeled images for robotic grasping by simulating an environment as close as possible of a physical setup. With this environment, we created the Jacquard dataset containing more than one million unique grasp locations on a large diversity of objects. Fig. 1 shows some examples of annotated images from Jacquard dataset. We also introduce a novel criterion, namely simulated grasp trial (SGT), to judge the goodness of a grasp location prediction based on physic grasp simulation. This criterion comes in contrast to the distance-based metrics traditionally used for the evaluation of grasp prediction and sticks with the fact that a single object can depict a large number of grasping positions, including those which are not necessarily previously annotated. Using three different evaluation metrics, including SGT and assessment through a real grasping robot trials, we show that this novel dataset, despite being synthetic, can be used to train a deep neural network (DNN), for grasp detection from a simple image of the scene and achieve much better prediction

[1]Siléane, Saint-Etienne, France a.depierre@sileane.com
[2]University of Lyon, Ecole Centrale de Lyon, LIRIS, CNRS UMR 5205, France {emmanuel.dellandrea, liming.chen}@ec-lyon.fr
Preprint version, accepted at IEEE/RSJ IROS 2018

of grasp locations, in particular for unseen objects, than the same DNN when it is trained using a grasp dataset with human labeled data.

This paper is organized as follows. Section II overviews the related work. Section III states the modelisation we used to describe a grasp. Section IV presents the method used to generate Jacquard dataset. Section V discusses the experimental results using the Jacquard dataset in comparison with the Cornell dataset. Section VI concludes the paper.

II. RELATED WORK

Early research in grasp prediction assumed the robot to have a perfect knowledge of its environment and aimed to plan grasps based on a 3D model of the object [7] [8]. Using this technique, Goldfeder et al. [9] created the Columbia Grasp Database, containing more than 230k grasps. With this type of approach, the notion of image is not present, only the CAD models of the gripper and objects are used. At test time, a query object is matched with an object within the database and a grasp is generated using the similarity of the CAD models. With this approach, both the model and the position of the object have to be known at test time, which is generally not the case for real-world applications.

Recent development of deep learning [10] and more particularly of the Convolutional Neural Networks (CNN) have inspired many researchers to work directly on images instead of 3D CAD models. The simultaneous apparition of cheaper sensors as the Kinect, also helped by providing additional depth information to the RGB image. This led to the development of datasets based on physical trials. In [11] a method to share the knowledge of different robots was developed in order to collect a large collection of data, in [3] a Baxter robot has been used to collect 50k data, while in [4] the authors collected over 800k datapoints (image, grasp position and outcome) using 14 robotic arms running during two months. In the last two cases, a CNN was successfully trained to detect grasp positions from the collected data. However, these approaches are either material or time consuming or and can not be fully automatized: human intervention is still needed to position the objects in front of the robot. Moreover, these methods only generate one single grasp annotation whereas there are generally several positions which could be good for robotic grasping.

To overcome the issue of time-consuming data generation, Mahler et al. [2] created Dexnet-2.0, a synthetic dataset with 6.7 millions depth images annotated with the success of the grasp performed at the center of the image. They trained a Grasp Quality CNN with these data and achieved a 93% success rate when predicting the outcome of a grasp. The GQ-CNN has good performance, but it can not be trained end-to-end to predict grasp positions: it only takes grasp candidates generated by another method as an input and rank them.

In [12], Johns et al. used a similar approach: they simulated grasp attempts on 1000 objects and trained a neural network to predict a score over a predefined grid of possible positions for the gripper. The network's input was a depth image, but they did not release their data publicly. In comparison, our Jacquard dataset contains more than 11k objects with both RGB and realistic depth information created through stereo-vision.

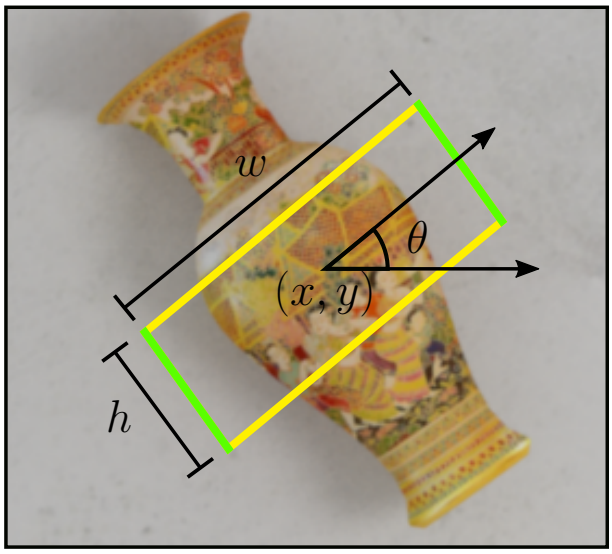

Fig. 2. Parametrization of a grasp for a parallel-plate gripper. A grasp is described as a five dimensional vector: two values for the position of the center, two for its size and one for its orientation with respect to the horizontal axis. Green sides represent the inner sides of the parallel jaws, yellow sides show the opening of the gripper.

The dataset most similar to our work is the Cornell Grasping Dataset [1]. It is composed of 885 RGB-D images of 240 different objects with 8019 hand-labeled grasp rectangles. As shown in [1], this dataset enables the training of a neural network to detect grasps in images. However a dataset with 885 images is quite small compared to the ones traditionally used in deep learning and may lead to bad performance when generalizing on different images or object configurations. Human labeling can also be biased to grasps that are easily performed with a human hand but not necessarily with a parallel plate gripper. Comparatively, the proposed Jacquard dataset is more than 50 times bigger with various objects and grasps sizes and shapes. A summary of the properties of public grasp datasets can be found in Table I.

III. MODELLING ROBOTIC GRASP

In this work, we are interested in finding a good grasp from a RGB-D image of a single object laying on a plane. A grasp is considered good when the object is successfully lifted and moved away from the table by a robot with a parallel-plate gripper. As shown in Fig. 2, a grasp can be described as:

$$g = \{x, y, h, w, \theta\} \tag{1}$$

where (x, y) is the center of a rectangle, (h, w) its size and θ its orientation relative to the horizontal axis of the image. This representation differs from the one of seven

[1] http://pr.cs.cornell.edu/grasping/rect_data/data.php

TABLE I

SUMMARY OF THE PROPERTIES OF PUBLICLY AVAILABLE GRASP DATASETS

Dataset	Number of objects	Modality	Number of images	Multiple gripper sizes	Multiple grasps per image	Grasp location	Number of grasps	Automatized generation
Levine et al. [4]	-	RGB-D	800k	No	No	Yes	800k	No
Mahler et al. [2]	1500	Depth	6.7M	No	No	No	6.7M	Yes
Cornell	240	RGB-D	1035	Yes	Yes	Yes	8019	No
Jacquard (ours)	11k	RGB-D	54k	Yes	Yes	Yes	1.1M	Yes

dimensions described in [13] but Lenz et al. show in [14] that it works well in practice. The main advantage of this representation is that the grasp can be simply expressed in the image coordinates' system, without any information about the physical scene: z position of the parallel plates and approach vector are determined from the depth image. When the grasp is performed by a real robot, h and w are respectively fixed and bounded by the shape of the gripper.

IV. JACQUARD DATASET

To solve the problem of data starvation, we designed a new method to get images and ground truth labels from CAD models through simulation. Then we applied this process to a subset of ShapeNet [15], namely ShapeNetSem [16], resulting in a new dataset with more than 50k images of 11k objects and 1 million unique successful grasp positions annotated. These data are made available to the research community [2]. The main pipeline we used for data generation is illustrated on Fig. 3. Physics simulation were performed using pyBullet library [17] and Blender [18] was used to render the images through its Cycles Renderer.

A. Scene creation

Scenes are all created in the same way. A plane with a white texture is created, the texture being randomly rotated and translated to avoid constant background. Then we select an object from a pool of CAD models. As the objects in ShapeNet have a wide range of scales, we rescale the model so the longest side of its bounding box has a length between 8 and 90 cm. We also give the object a mass depending on its size (80 g for a 8 cm object and 900 g for a 90 cm one) before dropping it from a random position and orientation above the plane. Once the object is in a stable position, the scene configuration is saved.

This scene description is sent to two independent modules: one to render the images and one with a physics simulator to generate the grasp annotations. For the Jacquard dataset, we created up to five scenes for each object. This number was chosen in order to have different views of the objects, but can be increased without any change in the process if necessary.

B. Image rendering

RGB and true depth images are rendered with Blender. To stick as close as possible to real scene images, instead of adding Gaussian noise to the perfect depth image as in [12], we rendered two more RGB synthetic images with a projected pattern and applied an off-the-shelf stereo-vision algorithm [19] on them, giving a noisy depth. This approach has been shown to produce depth images very close to real ones in [20]. A binary mask separating the object and the background is also created.

[2] http://jacquard.liris.cnrs.fr

C. Annotation generation

To generate grasp annotations, we used the real-time physics library pyBullet. As for the rendering module, the object model is loaded into the pyBullet environment, however, to speed up calculations, collisions are not computed directly on the mesh but on a volumetric hierarchical approximate convex decomposition [21] of it. Different grippers with parallel-jaws are simulated. They all have a max opening of 10 cm and a jaw size in $\{1, 2, 3, 4, 6\}$ cm. The different jaw sizes for the gripper combined with the varied scales of objects ensure that our simulated gripper can perform grasps in a wide range of different configurations.

Grasp annotations are generated in three steps. First, we generate thousands of random grasp candidates covering the whole area under the camera. Then, all these grasp candidates are tested through rigid body simulation using a gripper with a jaw size of 2 cm. And finally all the successful positions of the previous step are tested again with all the gripper sizes. The result is a set of successful grasp locations, each having between 1 and 5 jaw sizes.

To perform simulated grasps, the approach vector is set to the normal at the center of the grasp and the orientation and opening of the gripper are defined by the rectangle coordinates as described in section III. A grasp is considered successful if the object is correctly lifted, moved away and dropped at a given location by the simulated robot. Once all the random candidates have been tested, a last pass is performed on good grasps to remove the ones which are too close from each other. This last step is necessary to ensure that all the grasps are annotated only once.

As the number of possible grasps for one image is very large, we used a non-uniform probability distribution: candidates are generated more frequently in the most promising areas. Theoretically, candidates could be generated with a uniform distribution, but in this case many grasps would fall in an empty area without the object. For the Jacquard dataset, we used a simple heuristic looking for aligned edges in the image and generating the probability distribution from the density of such edges. However, our experiments showed us that any reasonable heuristic lead to a similar final grasps distribution in the image, at the cost of more random trials. With this method, we can reduce the number

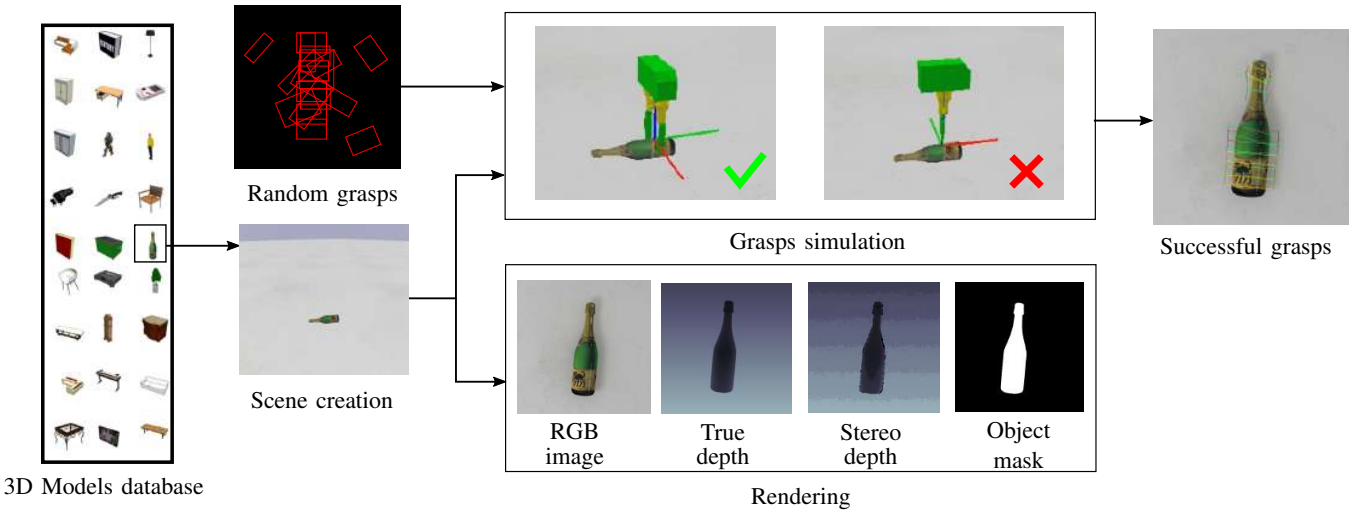

Fig. 3. The pipeline we used to generate annotated images from 3D models. Random grasps are generated from a probability map obtained with a simple heuristic algorithm before being tested in the simulation environment. In the rendering part, a synthetic camera renders the different images.

of grasp attempts necessary to annotate a scene by orders of magnitudes, while keeping a diversity in grasp locations. Such a diversity is very important for deep learning oriented methods.

D. Assessment criterion of successful grasp predictions

With the Cornell Grasp Dataset, the criterion used to determine whether a grasp prediction is correct or not is a rectangle-based metrics. With this criterion, a grasp is considered to be correct if both:

- The angle between the prediction and the ground-truth grasp is smaller than a threshold (a typical value is 30°)
- The intersection over union ratio between the prediction and the ground-truth grasp is over a threshold (typically 25%)

This criterion can however produce a lot of "visually" false-positives, *i.e.*, grasps that, from our human expertise, look bad, but that the rectangle metrics predict as good, as well as false-negatives, *i.e.*, grasps that, from our human expertise, look good, but that the rectangle metrics predict bad. Fig. 4 shows some examples of such misclassifications.

With the Jacquard dataset, we propose a new criterion based on simulation, subsequently called simulated grasp trial-based criterion (SGT). Specifically, when a new grasp should be evaluated as successful or not, the corresponding scene is rebuilt in the simulation environment and the grasp is performed by the simulated robot, in the exact same conditions as during the generation of the annotations. If the outcome of the simulated grasp is a success, *i.e.*, the object is successfully lifted and moved away by the simulated robot using the predicted grasp location, the prediction is then considered as a good grasp. This novel SGT criterion is much closer than the rectangle metrics to real-world situations where a single object can have many successful grasp locations, including successful grasp locations which are not previously annotated. For the purpose of reproducible research, we are releasing along with the dataset a web interface allowing researchers to send grasp requests on our simulator and receive the corresponding grasp outcome.

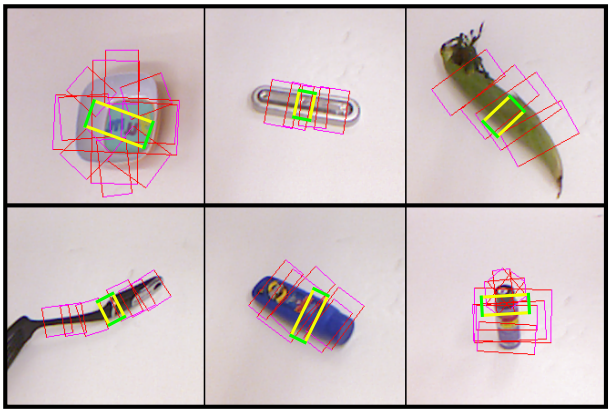

Fig. 4. Example of misclassifications with the rectangle metrics. Prediction is in yellow and green, ground truth is in red and purple. Top row shows false positives, bottom row shows false negatives.

V. EXPERIMENTS AND RESULTS

In order to evaluate the effectiveness of the proposed simulated Jacquard grasp dataset, we carried out two series of experiments: 1) cross-dataset grasp prediction with the Cornell and Jacquard datasets (section V-B); 2) evaluation of grasp predictions using a real grasping robot (section V-C). We start by explaining the training setup.

A. Training setup

In all our experiments, we used an off-the-shelf CNN, *i.e.*, AlexNet[22]. The network's convolution weights have been pre-trained on ImageNet [23] while the fully connected layers are trained from scratch. To use AlexNet with RGB-D, we simply normalize the depth image to get values close to color channels and duplicate the blue filters in the first pre-trained convolution layers. The network is trained through Stochastic Gradient Descent algorithm for 100k iterations

TABLE II

ACCURACY OF THE NETWORK TRAINED ON DIFFERENT DATASETS

Training Dataset	Rectangle Metrics		SGT
	Cornell	Jacquard	Jacquard
Cornell	86.88% ± 2.57	54.28% ± 1.22	42.76% ± 0.91
Jacquard (ours)	81.92% ± 1.95	74.21% ± 0.71	72.42% ± 0.80

with a learning rate of 0.0005, a momentum of 0.9 and a weight decay of 0.001. The learning rate is set to 0.00005 after the first 75k iterations. To compute the error of the network, the Euclidean distance between the prediction and the closest annotation is used:

$$\mathcal{L} = \min_{g \in \mathcal{G}} \|g - \hat{g}\|^2 \quad (2)$$

Where $\mathcal{G}$ is the set of all the annotations for the image and $\hat{g}$ is the network prediction.

Before training, we perform data augmentation by translating, rotating and mirroring the images. For synthetic data, we also use the object's mask to replace the default background with different textures (cardboard, paper, wood, grass ...) to generate more variabilities.

B. Cross-dataset evaluation

This series of experiments aims to show that: 1) our Jacquard grasp dataset, despite being synthetic, can be used to train DNNs to predict grasp locations on real images; 2) The diversity of objects and grasp locations is important for a trained CNN to generalize on unseen objects. For this purpose, the Cornell dataset with its 885 RGB-D images on 240 objects and 8019 hand labeled grasp locations is used along with a portion of Jacquard which contains 15k RGB-D images on a selected 3k objects and 316k different grasp positions. To highlight 1), Alexnet is trained on Jacquard and tested on Cornell; for 2) it is trained on Cornell and tested on Jacquard. For comparison, we also display a baseline performance with Alexnet trained and tested on the same dataset, *i.e.*, Cornell or Jacquard. For this purpose, we performed training and testing of the network with 5-fold cross validation for Cornell or Jacquard, leading to 5 variants of Alexnet with slightly different accuracies on each dataset. Each variant trained on Cornell (Jacquard, respectively) is then tested on the whole Jacquard dataset (Cornell, respectively) to evidence 1).

Table II summarizes the experimental results evaluated by both rectangle metrics and SGT criterion. As can be seen from Table II, when Alexnet is trained on our simulated Jacquard dataset and tested on Cornell, it achieves a grasp prediction accuracy of 81.92% which is quite close to the baseline performance of 86.88%. Furthermore, we also noticed that the networks trained on synthetic data tended to predict grasps which were visually correct despite being classified as wrong by the rectangle metrics. Typical examples are shown on the bottom line of Fig. 4.

In contrast, when Alexnet is trained on Cornell and tested on Jacquard with a much wider diversity of objects and grasps, it depicts a grasp prediction accuracy of 54.28% which records a performance decrease of 20 points in comparison with its baseline performance. As for the other training, part of this gap could be explained by the misclassifications of the rectangle metrics. However, this performance decrease is confirmed by our criterion based on simulated grasp trials (SGT): Alexnet trained on Cornell only displays a grasp prediction accuracy of 42.76% which is 30 points behind the 72.42% accuracy of the same CNN trained on Jacquard.

All these figures thus suggest that Jacquard can be used to train CNN, for an effective prediction of grasp locations. Furthermore, thanks to the diversity of objects and grasp locations, Jacquard enables a much better generalization skills of the trained CNN.

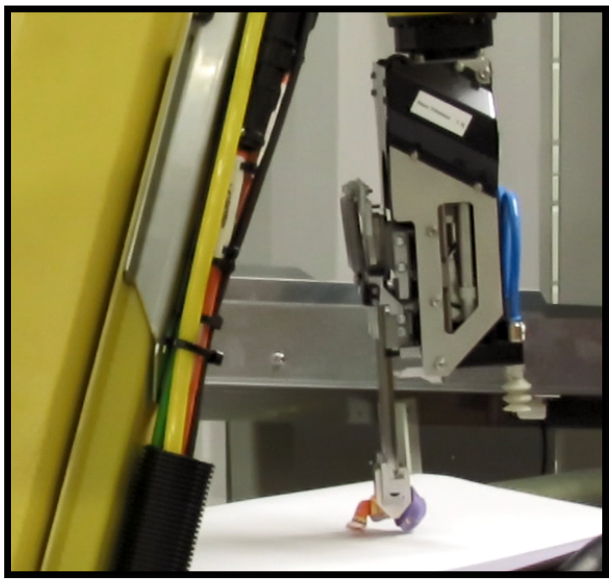

Fig. 5. Our physical setup to test grasp predictions. The camera is located above the grasping area.

C. Evaluation of grasp predictions using a real grasping robot

How good is a grasp predicted by a trained deep neural network, in real? To answer this question of possible reality gap, we used a parallel plate gripper mounted on a Fanuc's M-20iA robotic arm and a set of various objects. To ensure a wide variability in shapes, materials and scales, we used 15 everyday objects (toys and furnitures) and 13 industrial components. Fig. 5 shows the robot performing a predicted grasp on one of the testing objects. Our criterion of a successful grasp was the same as in the simulator but this time using the aforementioned real grasping robot instead of the simulated one: the grasp of an object is considered successful only if the object is lifted, moved away and correctly dropped. For this test, we compared Alexnet trained on the Cornell dataset and the same network trained on a subset of 2k objects from the Jacquard dataset.

The experimental results show that the grasp predictor with Alexnet trained on the Jacquard dataset displays a grasp successful rate of 78.43% which is even 6 points higher than

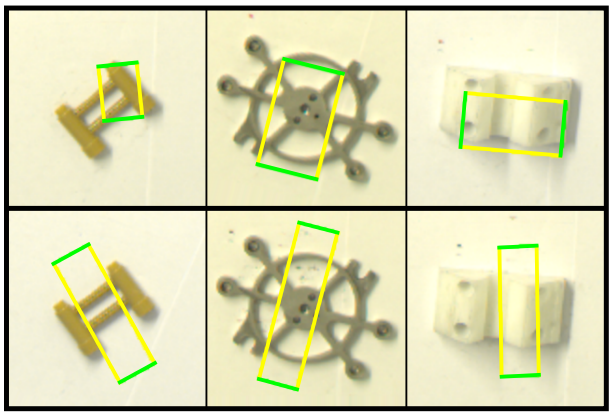

Fig. 6. Samples of grasp predictions on our real setup for the network trained on the Cornell dataset (top row) and the one trained on our synthetic Jacquard data (bottom row).

the grasp accuracy displayed by Alexnet when it was trained and tested on the subset of 3k objects of Jacquard (see Table II) using the SGT criterion. This generalization skill of the trained grasp predictor can be explained by the large diversity of objects and grasp locations in the Jacquard dataset. For most of the failed cases, the grasp was not stable enough: the rectangle in the image was visually coherent and the object was successfully lifted but dropped during the movement of the robot.

Now with the the same network trained on Cornell, the robot succeeded only 60.46% of the predicted grasps, mostly due to bad rectangle localization in the image. Fig. 6 shows some examples of the objects for which the network trained on Cornell failed to predict a good grasp while the one trained on Jacquard succeeded.

VI. CONCLUSIONS

In this work, we presented a method to generate realistic RGB-D data with localized grasp annotations from simulation. Using this method, we built a large scale grasp dataset with simulated data, namely Jacquard, and we successfully used it to train a deep neural network to predict grasp positions in images. The grasp predictor trained using Jacquard shows a much better generalization skill than the same network when trained with a small hand labeled grasp dataset. Our future work will focus on the quality assessment of grasp predictions and on extending this method to more complexe scenes, for example with multiple objects.

ACKNOWLEDGMENT

This work was in part supported by the EU FEDER, Saint-Etienne Métropole and Région Auvergne-Rhône-Alpes fundings through the FUI PIKAFLEX project and in part by the French National Research Agency, l'Agence Nationale de Recherche (ANR), through the ARES labcom project under grant ANR 16-LCV2-0012-01.

REFERENCES

[1] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1316–1322.

[2] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[3] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3406–3413.

[4] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, 2016.

[5] A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 886–900, 2012.

[6] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[7] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1824–1829.

[8] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, 2010.

[9] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 1710–1716.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] J. Oberlin, M. Meier, T. Kraska, and S. Tellex, "Acquiring Object Experiences at Scale," in *AAAI-RSS Special Workshop on the 50th Anniversary of Shakey: The Role of AI to Harmonize Robots and Humans*, 2015, blue Sky Award.

[12] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4461–4468.

[13] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3304–3311.

[14] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[15] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[16] M. Savva, A. X. Chang, and P. Hanrahan, "Semantically-Enriched 3D Models for Common-sense Knowledge," *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality*, 2015.

[17] E. Coumans and Y. Bai, "pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org/, 2016–2017.

[18] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Blender Institute, Amsterdam, 2016. [Online]. Available: http://www.blender.org

[19] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[20] R. Brégier, F. Devernay, L. Leyrit, and J. L. Crowley, "Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2209–2218.

[21] K. Mamou, "Volumetric hierarchical approximate convex decomposition," in *Game Engine Gems 3*, E. Lengyel, Ed. A K Peters, 2016, pp. 141–158.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.