

# Visual Feedback and Learning for Optimal Velocity of Robotic Visual Quality Inspection

Andrej Gams  
Jozef Stefan Institute  
Ljubljana, Slovenia  
andrej.gams@ijs.si

Simon Reberšek  
Jozef Stefan Institute  
Ljubljana, Slovenia  
simon.rebersek@ijs.si

Aleš Ude  
Jozef Stefan Institute  
Ljubljana, Slovenia  
ales.ude@ijs.si

## ABSTRACT

Robotic learning can effectively be applied for industrial applications. In this paper we show one such example, with a learning algorithm applied to reach the optimal velocity of robotic motion for visual quality inspection. If such learning is performed before the start of the production, even if it takes a lot of repetitions, it can achieve faster cycle times and thus greater productivity. The described approach is general and can be used with different types of learning and feedback signals. In the paper we analyze the appropriate feedback signal and show the results of learning for a standard area-scan camera.

## KEYWORDS

robotic learning, visual feedback, focus measure, robotic quality control

## 1 INTRODUCTION

Many operations are performed with autonomous robots in factories, and many more are expected in the factories of the future. Often, visual feedback is used to provide the trajectory of the robot. [16]. However, various vision techniques, such as time of flight, structured light, laser triangulation, RGB cameras, stereo vision, etc. are used for quality control processes in the industry [4, 9]. Quality control can take different modes. For example, discrete checking of an object from a few viewpoints and comparing the acquired images to predefined templates [11]. Another option is to continuously acquire images with either moving the in-hand camera, or moving the object in front of the camera. A plethora of advanced methods for image processing for quality inspection have been proposed, including deep learning methods [17].

For effective vision-based operations, the machine vision hardware needs to be properly set-up and tuned. In large-scale automated production, it is typically set-up once, and then it remains in the same configuration throughout its life cycle. Consequently, machine vision hardware is often designed in a way that some adjustments can only be carried out manually. Many lenses thus have a fixed focal length and manual adjustment of the iris and focus [1]. However, even if the vision-hardware is set up only once, the process still constitutes a tedious and demanding task. For example, in continuous visual inspection, e. g., for visual inspection of weld



**Figure 1:** The robot cell composed of a UR-10 robot, a Basler acA1300-60gm area scan camera, a dedicated light source (not shown) and the dummy flat object at a calibrated distance from the robot.

seams [18], requires the robot to follow the seam with the camera at the end-effector. The image has to be sharp in all the positions and at all velocities. Thus, for such continuous visual quality control, the operator has to define the correct robot path, but also the correct speed, because too fast motion in front of the camera might result in a blurry image.

The demands of the industry typically culminate in having to move as fast as possible in order to reach high cycle times. [19]. Thus, when programming robot motion for quality control, the path can be properly configured by exporting the object CAD data and appropriate robot-to-object calibration, but the speed of robot motion is typically left to the operator, who spends a considerable amount of time hand-tuning it. However, this tuning could be left to an autonomous learning algorithm with proper feedback. In this paper we briefly analyze possible visual filters for appropriate feedback, and demonstrate how hand-tuning can be automated by employing learning algorithms.

### 1.1 Problem Statement

We investigate learning of motion speed for continuous visual quality inspection of products with a robot using an in-hand camera. The system should:

- follow a predefined path,

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IS2019, October 7–11, 2019, Ljubljana, Slovenia, Europe  
© 2019 Copyright held by the owner/author(s).

- autonomously optimize motion velocity using learning, so that
- the velocity of motion does not introduce blurring, i. e., a reduced focus measure.

We also analyze appropriate visual feedback filters to determine image sharpness.

The following assumptions hold. i) A CAD system provides an accurate robot path (trajectory) consisting of required positions and orientations; ii) proper robot-object calibration can be achieved; iii) The system operates under constant lighting and camera conditions.

To achieve seamless velocity modulation, we applied Dynamic movement primitives (DMPs) developed by Ijspeert et al. [10]. We used a variant of DMPs called Cartesian Space Dynamic Movement Primitives [21] for the trajectory encoding. Other trajectory encoding approaches could easily be applied, for example Gaussian Mixture Models [3]. For the learning we applied Iterative Learning Control (ILC) [2, 6]. Again, other methods, such as reinforcement learning [5, 13] could be applied.

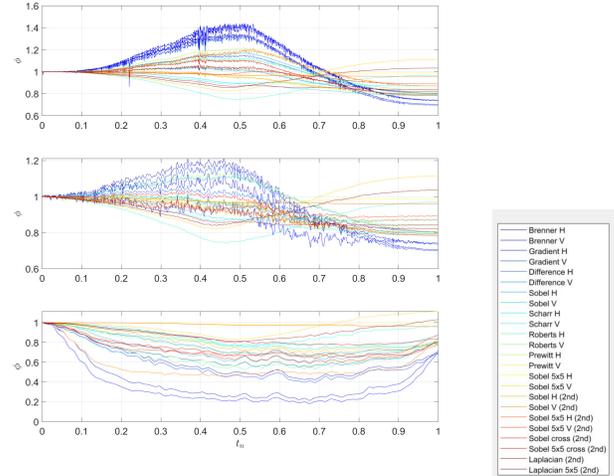
## 2 FOCUS MEASURE

Visual quality inspection requires sharp, focused images. Only a few industrial camera/lenses on the market provide autofocus, with little information about how focus is determined in these cameras [1].

We first used robot-driven autofocus as described in [1] to set our fixed-focus camera at the right distance from the object for inspection. To do this we used squared horizontal gradient focus measure, as suggested by [1]. This focus measure has a distinct bell-shape characteristics, with the best focus achieved at the peak. The robot moves the camera perpendicularly to the object of inspection, away and towards the object. After detecting the peak value (the focus measure begins to decrease), the robot reverses its motion and travels in the other direction at a slower speed, again until crossing the peak value. These movements are repeated until the accurate position resulting in peak focus measure  $\phi$  is obtained. Details of this method and results showing that the achieved focus measure is higher than the one achieved by manually positioning the camera, are presented in [1].

Using this approach we can set the camera into focus for one point, for example above the starting point of the path of inspection. We assume that the desired inspection path has been extracted from a CAD model of the inspected object. To obtain the reference values  $\phi(t)$  for all points on the inspection trajectory, the robot moves along the desired inspection path. However, the question is whether speed has an effect on the focus measure, and furthermore, out of many focus measures that exist, which will be most effected by the speed.

Focus measures are based on different orders of differentiation (first or second), image histogram, correlation and data compression [14]. Methods employing first-order gradients use different operators, such as squared gradient, Sobel (horizontal, vertical, combined), Laplacian, Scharr, and others. We tested several possible focus measures. We moved the robot with an in-hand camera over a dummy object at three different velocities, completing the motion in 3s, 20s, and 60s. Figure 2 shows the relative focus measure as a function of normalized time (phase), going from 0 to 1, for different



**Figure 2: Different focus measures at different speeds of robot motion (top: 60s, middle: 20s, bottom: 3s), for normalized time. The measures were normalized to the initial value.**

measures. The label states the measures used. The feedback focus measures were normalized to the initial value. As we can see, a higher velocity indeed decreases the focus measure, and the effect is different for different focus measures.

Figure 3 shows filtered values of relative difference between slow and fast motion for the top 10 focus measures. We can see that squared horizontal gradient focus measure is the most reactive to change of velocity. Is is provided by

$$\phi = \sum_{x=0}^{M-1} \sum_{y=0}^{N-2} (I(x, y+1) - I(x, y))^2. \quad (1)$$

Here the image is sized  $M \times N$ , with  $I(x, y)$  the intensity values at pixels  $(x, y)$ .

The values would be the same for the vertical gradient if the camera were rotated  $90^\circ$ . The vertical horizontal gradient is calculated by

$$\phi = \sum_{x=0}^{M-2} \sum_{y=0}^{N-1} (I(x+1, y) - I(x, y))^2. \quad (2)$$

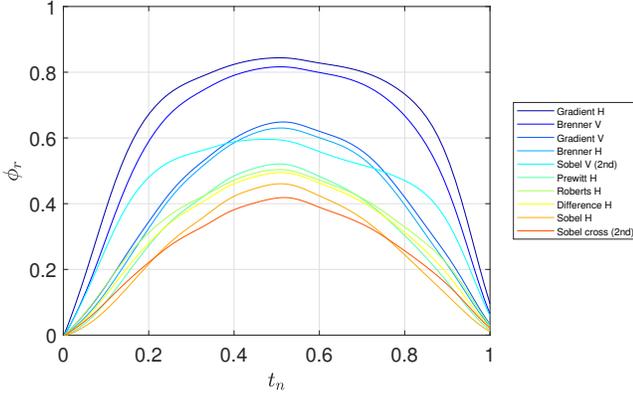
Brenner vertical and horizontal filters provide similar values. They are defined by

$$\phi = \sum_{x=0}^{M-1} \sum_{y=0}^{N-3} (I(x, y+2) - I(x, y))^2. \quad (3)$$

for the horizontal and

$$\phi = \sum_{x=0}^{M-3} \sum_{y=0}^{N-1} (I(x+2, y) - I(x, y))^2. \quad (4)$$

for the vertical filter.



**Figure 3: Filtered relative change of feedback values for 10 different focus measures.**

### 3 TRAJECTORY ENCODING

In this paper we used the original formulation of Cartesian DMPs from [21], expanded with temporal scaling, as originally proposed for standard DMPs in [15].

The following parameters compose a CDMP: weights  $\mathbf{w}_k^p$ ,  $\mathbf{w}_k^o \in \mathbb{R}^3$ ,  $k = 1, \dots, N$ , which represent the position and orientation parts of the trajectory, respectively; trajectory duration  $\tau$  and the final desired, goal position  $\mathbf{g}^p$  and orientation  $\mathbf{g}^o$  of the robot. Variable  $N$  sets the number of radial basis functions that are used to encode the trajectory. The orientation is in CDMP represented by a unit quaternion. In this paper we only consider the positions.

$$v(s)\tau\dot{\mathbf{z}} = \alpha_z(\beta_z(\mathbf{g}^p - \mathbf{p}) - \mathbf{z}) + \mathbf{f}_p(s), \quad (5)$$

$$v(s)\tau\dot{\mathbf{p}} = \mathbf{z}, \quad (6)$$

Variable  $v(s)$ , as a function of the phase, provides temporal scaling. Parameter  $\mathbf{z}$ , denotes the scaled linear velocity ( $\mathbf{z} = \tau\dot{\mathbf{p}}$ ). The nonlinear parts, termed also forcing terms,  $\mathbf{f}_p$  and are defined as

$$\mathbf{f}_p(s) = \mathbf{D}_p \frac{\sum_{k=1}^N \mathbf{w}_k^p \Psi_k(s)}{\sum_{k=1}^N \Psi_k(s)}, \quad (7)$$

Forcing terms contain parameters  $\mathbf{w}_k^p \in \mathbb{R}^3$ . They have to be learned, for example directly from an input Cartesian trajectory  $\{\mathbf{p}_j, \dot{\mathbf{p}}_j, \ddot{\mathbf{p}}_j, t_j\}_{j=1}^T$ . The scaling matrix  $\mathbf{D}_p \in \mathbb{R}^{3 \times 3}$  can be set to  $\mathbf{D}_p = \mathbf{I}$ . Other possibilities are described in [21]. The nonlinear forcing terms are defined as a linear combination of radial basis functions  $\Psi_k$

$$\Psi_k(x) = \exp\left(-h_k(x - c_k)^2\right). \quad (8)$$

Here  $c_k$  are the centers and  $h_k$  the widths of the radial basis functions. The distribution of weights can be, as in [20],  $c_k = \exp\left(-\alpha_x \frac{k-1}{N-1}\right)$ ,  $h_k = \frac{1}{(c_{k+1} - c_k)^2}$ ,  $h_N = h_{N-1}$ ,  $k = 1, \dots, N$ . The time constant  $\tau$  is set to the desired duration of the trajectory, i. e.  $\tau = t_T - t_1$ . The goal position is usually set to the final position on the desired trajectory, i. e.  $\mathbf{g}^p = \mathbf{p}_{t_T}$ . Detailed CDMP description and auxiliary math are explained in [21].

Temporal scaling  $v(s)$  provides a trajectory that defines a speed profile of the motion. It is composed of a weighted combination of kernel functions

$$v(s) = \frac{\sum_{k=1}^R \mathbf{w}_k^v \Psi_k(s)}{\sum_{k=1}^R \Psi_k(s)}. \quad (9)$$

Here  $R$  defines the number of kernel functions, given in (8), for temporal scaling. For simplicity, this number can be the same as  $N$  in (7). The weights  $\mathbf{w}_k^v$  need to be learned in the same manner as the weights for position trajectories.

### 4 IMPROVING SPEED OF QUALITY CONTROL WITH LEARNING

Focus measure is repeatable, and there is a clear difference in  $\phi$  for different motion speeds, as evident from Fig. 2. Therefore, we can use  $\phi$  as the feedback for learning.

The goal of learning here is to achieve a fastest possible velocity profile, where there will be only little or even no degradation of the focus measure. Thus, the motion will be executed as fast as possible, and the sharpness of the image, used for quality inspection, will not degrade.

It should be noted that with the chosen parametric speed profile representation, different means of learning open up, as was shown in [5], or in [12]. In this paper we have chosen one of the variations of iterative learning control. The advantage of using a learning control method is that it requires very few iterations to improve results. However, such methods never truly converge, but only asymptotically approach the target value [2].

The chosen learning algorithm for learning was previously applied for coaching of robot motion through human intervention [7]. A short recap is provided for completeness of the paper. Its basis is learning of weights of CDMPs, but in this case it is used for the learning of the weights of the velocity profile  $v$ . The weights of the velocity profile  $\mathbf{w}^v$  are iteratively updated (for 1DOF) with

$$\mathbf{w}_{i,j+1}^v = \mathbf{w}_{i,j}^v + \Gamma_{i,j+1} P_{i,j+1} r e_j \quad (10)$$

$$P_{i,j+1} = \frac{1}{\lambda} \left( P_{i,j} - \frac{P_{i,j}^2 r^2}{\lambda + P_{i,j} r^2} \right) \quad (11)$$

$$e_j = f_{\text{target},j} - \mathbf{w}_{i,j}^v r. \quad (12)$$

Here  $j+1$  stands for the next time sample and  $i$  for the selected weight.  $P_i$ , is the inverse covariance of  $w_i$ ,  $r$  is the amplitude gain. To apply this algorithm for modifying the speed profile based on the focus measure  $\phi$ , we replace (12) with

$$e_j = k * (\phi_{\text{slow}} - \phi_{\text{fast}}). \quad (13)$$

here  $k$  is a positive constant gain. The whole algorithm is described in procedure of Fig. 4. The learning takes place until a predefined threshold of  $e_j$  is reached. This threshold can be determined empirically.

Instead of learning directly on the weights, one can also simply generate the velocity profile from the weights and add to it a scaled  $e_j$ ,

$$v_{l+1}(t) = v_l(t) + k e_j(t), \quad (14)$$

where the gain  $k$  is set empirically and  $l$  stands for iteration. The resulting  $v_{l+1}(t)$  is then again encoded into weights, for example

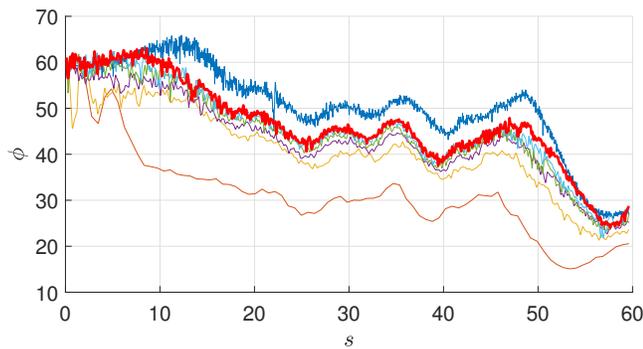
**procedure** LearnProfile

```

record  $\phi$  for slow (practically static) motion;
record  $\phi$  for fast motion with  $w_i^v = \text{const}$ ;
while  $\phi_{\text{latest}} > \text{threshold}$ 
  execute motion with current  $w^v$ 
  calculate new error of  $\phi$  with  $\phi - \phi_{\text{latest}}$ 
  update  $w^v$  using (10), (11) and (13)
end

```

**Figure 4: Procedure for learning the velocity profile using the squared gradient focus measure.**



**Figure 5: Results of velocity learning for a dummy flat object. The top lines shows absolute  $\phi$  for slow 60s motion. The bottom line shows  $\phi$  for fast, 3s motion.  $\phi$  over iterations is shown between, with the final, red line almost the reference, but at 19.12s.**

iteratively using (10) – (12), or with a batch conversion, as shown in [10].

Figure 5 shows the results of the algorithm, applied on a dummy object, using the algorithm described in Fig. 4 and squared horizontal gradient focus measure. Results on a curved object were reported in [8].

## 5 CONCLUSION

Learning algorithms have tremendous potential to improve the productivity of industrial processes today, not only in the future. The results show that autonomous learning algorithms can improve the performance of the robot, and that such algorithms can be effectively applied optimizing production processes. Thus, they can relieve and help operators/engineers. Fine-tuning and calibration of the processes is a tedious, long process, requiring a lot of effort. Time and money can be saved both in the set-up as well as in the improved productivity.

## ACKNOWLEDGMENTS

This research has been funded in part by the GOSTOP programme C3330-16-529000, co-financed by Slovenia and EU under ERDF, and by the EU’s Horizon 2020 IA QU4LITY (GA no. 825030).

## REFERENCES

- [1] Robert Bevec, Timotej Gašpar, and Aleš Ude. 2019. Robot-Driven Autofocus Control Mechanism for an In-hand Fixed Focus Camera. In *Advances in Service and Industrial Robotics*, Nikos A. Aspragathos, Panagiotis N. Koustoumpardis, and Vassilis C. Moulaniotis (Eds.). Springer International Publishing, Cham, 551–559.
- [2] D. A. Bristol, M. Tharayil, and A. G. Alleyne. 2006. A survey of iterative learning control. *IEEE Ctrl. Sys. M.* 26, 3 (2006), 96–114. <https://doi.org/10.1109/MCS.2006.1636313>
- [3] S. Calinon. 2015. Robot learning with task-parameterized generative models. In *Proc. Intl Symp. on Robotics Research (ISRR)*.
- [4] Che-Seung Cho, Byeong-Mook Chung, and Moo-Jin Park. 2005. Development of real-time vision-based fabric inspection system. *IEEE Transactions on Industrial Electronics* 52, 4 (Aug 2005), 1073–1079. <https://doi.org/10.1109/TIE.2005.851648>
- [5] M. P. Deisenroth, G. Neumann, and J. Peters. 2013. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics* 2, 1-2 (2013), 1–142.
- [6] A. Gams, B. Nemeč, A. J. Ijspeert, and A. Ude. 2014. Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks. *IEEE Transactions on Robotics* 30, 4 (Aug 2014), 816–830. <https://doi.org/10.1109/TRO.2014.2304775>
- [7] Andrej Gams, Tadej Petrič, Martin Do, Bojan Nemeč, Jun Morimoto, Tamim Asfour, and Aleš Ude. 2016. Adaptation and coaching of periodic motion primitives through physical and visual interaction. *Robotics and Autonomous Systems* 75 (2016), 340 – 351. <https://doi.org/10.1016/j.robot.2015.09.011>
- [8] A. Gams, S. Reberšek, B. Nemeč, J. Škrabar, R. Krhlikar, J. Skvarč, and A. Ude. 2019. Robotic Learning for Increased Productivity: Autonomously Improving Speed of Robotic Visual Quality Inspection. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. 1275–1281. <https://doi.org/10.1109/COASE.2019.8842851>
- [9] Niko Herakovic. 2010. Robot Vision in Industrial Assembly and Quality Control Processes. In *Robot Vision*. IntechOpen, Rijeka, Chapter 26. <https://doi.org/10.5772/9285>
- [10] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal. 2013. Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors. *Neural Computation* 25, 2 (2013), 328–373.
- [11] Tetyana Ivanovska, Simon Reich, Robert Bevec, Ziga Gosar, Minija Tamosiunaite, Ales Ude, and Florentin Wörgötter. 2018. Visual Inspection and Error Detection in a Reconfigurable Robot Workcell: An Automotive Light Assembly Example. In *VISIGRAPP*.
- [12] Jens Kober, J Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274. <https://doi.org/10.1177/0278364913495721>
- [13] J. Kober and J. Peters. 2011. Policy Search for Motor Primitives in Robotics. *Machine Learning (MLJ)* 1-2 (2011), 171–203.
- [14] H. Mir, P. Xu, and P. van Beek. 2014. An extensive empirical evaluation of focus measures for digital photography. In *Digital Photography X (procspie)*, Vol. 9023. 90230I. <https://doi.org/10.1117/12.2042350>
- [15] B. Nemeč, A. Gams, and A. Ude. 2013. Velocity adaptation for self-improvement of skills learned from user demonstrations. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 423–428. <https://doi.org/10.1109/HUMANOIDS.2013.7030009>
- [16] Luis Pérez, Íñigo Rodríguez, Nuria Rodríguez, Rubén Usamentiaga, and Daniel F. García. 2016. Robot Guidance Using Machine Vision Techniques in Industrial Environments: A Comparative Review. *Sensors* 16 3 (2016).
- [17] D. Racki, D. Tomazevec, and D. Skocaj. 2018. A Compact Convolutional Neural Network for Textured Surface Anomaly Detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1331–1339. <https://doi.org/10.1109/WACV.2018.00150>
- [18] D. Schreiber, L. Cambrini, J. Biber, and B. Sardy. 2009. Online visual quality inspection for weld seams. *The International Journal of Advanced Manufacturing Technology* 42, 5 (01 May 2009), 497–504. <https://doi.org/10.1007/s00170-008-1605-3>
- [19] O. Semeniuta, S. Dransfeld, and P. Falkman. 2016. Vision-based robotic system for picking and inspection of small automotive components. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. 549–554. <https://doi.org/10.1109/COASE.2016.7743452>
- [20] A. Ude, A. Gams, T. Asfour, and J. Morimoto. 2010. Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. *IEEE Transactions on Robotics* 26, 5 (Oct 2010), 800–815. <https://doi.org/10.1109/TRO.2010.2065430>
- [21] A. Ude, B. Nemeč, T. Petrič, and J. Morimoto. 2014. Orientation in Cartesian space dynamic movement primitives. In *IEEE Int. Conference on Robotics and Automation (ICRA)*. 2997–3004. <https://doi.org/10.1109/ICRA.2014.6907291>