



Image-Driven Decision Making with Application to Control Gas Burners

Ewaryst Rafajlowicz, Wojciech Rafajlowicz

► To cite this version:

Ewaryst Rafajlowicz, Wojciech Rafajlowicz. Image-Driven Decision Making with Application to Control Gas Burners. 16th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM), Jun 2017, Bialystok, Poland. pp.436-446, 10.1007/978-3-319-59105-6_37. hal-01656234

HAL Id: hal-01656234

<https://inria.hal.science/hal-01656234>

Submitted on 5 Dec 2017

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.



Distributed under a Creative Commons Attribution 4.0 International License

Image-driven decision making with application to control gas burners

Ewaryst Rafajłowicz and Wojciech Rafajłowicz

Ewaryst Rafajłowicz and Wojciech Rafajłowicz are with the Faculty of Electronics,
Wrocław University of Technology, Wrocław , Poland.

`ewaryst.rafajlowicz@pwr.wroc.pl`

+

Abstract. Our aim is to propose a model-free approach to decision making that is based on the direct use of images. More, precisely, a content of each image is used – without further processing – in order to cluster them by the K-medoids method. Then, decisions are attached to each cluster by an expert. When a new image is acquired, it is firstly classified to one of the clusters and the corresponding decision is made. The approach is conceptually rather simple, but its success in on-line applications depends on the way of organizing learning and decision phases. We illustrate the approach by the example of a decision-making system for industrial gas burners.

Keywords: Camera in the loop, image-based control, gas burner

1 Introduction

Most decisions made by human beings and animals are based on images provided by the eyes. In most cases these decisions are made on an intuitive level as when we are walking or grasping a cup of tea. In somewhat more complicated situations, e.g., when we see a red light at a crossroads, our decisions are more consciously made. Even more image interpretation work is necessary when a policeman is governing the traffic at a crossroads¹.

The number of cases when processing images and their interpretation in decision making is so broad that it is impossible to consider all of them in one paper. We concentrate on image-driven decision making in industrial applications. At least the following three classes of problems can be distinguished:

1. simple signals given by workers to each other (start, stop, go on etc.) and safety monitoring by monitoring the presence in dangerous areas – we skip discussing these topics, since they are relatively simple and well-developed,

¹ By the way, it would be of interest to know how researchers working on autonomous vehicles take such cases into account. More broadly, one can imagine an emergency situation (road accident or a fire), when warnings are even less standardized (hand waving, using a flashlight or a handkerchief).

2. quality monitoring of produced items,
3. on-line control of continuously running industrial processes.

Quality monitoring of a produced item is also relatively well developed (see [3] and extensive bibliography cited therein and [10], [11] for more recent contributions). However, there are still large areas of potential applications of machine vision techniques in quality control both of item by item and continuously running industrial processes. In particular, the approach proposed in this paper can also be used for these purposes. The distinguishing feature of machine vision techniques in quality monitoring is that the resulting decision are in the most cases binary – confirms requirements or not. The second feature that differentiates them from problems of on-line control with camera in the loop is that in quality monitoring usually there is no automatic feedback in order to improve a production process. Instead, experts analyse past data, trying to find reasons of poor quality.

In this paper we concentrate on item 3. of the above list. Advanced applications of control systems with camera in the loop are described rather rarely (see [2], [12], [5], [5], [6] for several more recent examples). Approaches to control systems with a camera in the loop can roughly be classified as follows:

model-based – a mathematical model of a system to be controlled is known and a camera provides information that is inaccessible by classic sensors (see [9]),

model-free – a mathematical model of a process is not available and images provided by a camera are either:

intensively processed in order to extract problem-specific features usefull in decision-making [8], [14]

OR

roughly clustered using only a general (dis-)similarity measure between images in order to cluster them into system states that require the same (or similar) decision (action).

The latter approach is the main topic of this paper. We shall call this approach model-free, image-driven decision making (IDDM). As far as we know, the approach that is completely model-free decision making, which is solely based on images, but without their intensive processing, does not have its counterpart in the literature. For this reason we shall not provide comparisons of the IDDM with other methods that require much more a priori information.

Its main advantages of the IDDM are the following:

- the IDDM approach is relatively fast, since the learning (clusterization) phase can be separated from an on-line decision making,
- the decision-making phase can be implemented as a low-cost micro-processor system,
- a low level of a priori knowledge is necessary in order to design a IDDM system.

The IDDM is based on K -medoids clustering, but other clustering methods might also be used. It is worth mentioning that K -medoids clustering in image processing was used mainly for clustering pixels of one image (see [1] and the bibliography cited therein), while the proposed version aims to cluster images as whole entities.

The paper is organized as follows:

- in the next section we provide a general description of the IDDM approach,
- then, this approach is applied to design a decision-making system for control industrial gas burners using a camera,
- finally, we discuss possible extensions of the IDDM approach.

We remark that the problem of decision-making for industrial gas burners using a camera is an area of recent active research: [14], [8], [16], [13], but the approach proposed here is different. Namely, we do not use time-consuming image processing operations other than verifying to which cluster acquired images belong.

2 Image-driven decision making – a general idea and algorithm

The proposed approach consists of the following two phases:

- off-line clustering phase,
- on-line decision making.

Below we describe these phases in more details.

2.1 Off-line clustering phase

The first step of this phase is collecting a large number of images that are representative for the system states. We shall call these images the set of representative images (SRI). This step is crucial for a proper functioning of a decision system. The SRI must contain images that represent all important system states. The number of images illustrating each group of important states should be sufficiently large.

A strategy of clustering images and decisions. The second step is to cluster the set of images into an appropriate number of clusters (see below for details). The third step is to attach decisions (control actions) to each cluster. We discuss steps two and three in common, since one can consider the following two strategies of attaching (linking) decisions and images.

Cluster and attach decisions. The SRI is firstly clustered into, say, $K > 1$ disjoint sets of similar states represented by images. Then, an expert attaches control actions (decisions) to each of them. We shall use this strategy in the present paper.

Attach decisions and cluster. This strategy is more laborious for the expert, since he/she has to attach a decision to each image in the SRI. Then, pairs $(image, decision)$ are clustered, using a (dis-)similarity function (metric) that takes into account both the similarity of images as well as the similarity of decisions. This requires that decisions can be in some sense ordered or their closeness can be defined. After clustering pairs $(image, decision)$ one may find some inconsistencies, since it may happen that two images that are attached to the same cluster have different decisions attached. Thus, before using this approach it is necessary to unify decisions. This can be done by the majority voting among decision labels in each cluster. However, if

in a given cluster we have, e.g., two decision labels in almost equal proportions, then it is expedient to consider splitting this cluster into two and further join it with another cluster (sub-cluster) with the same decision label. As one can notice, this strategy needs to be carefully elaborated in order to avoid decision clashes.

Distance functions for clustering images. Images can formally be clustered in the same way as other objects. The only specificity is in selecting a similarity measure between images that should take into account both the correspondence of pixels in space as well as different contents and ways of coding images. We require that a similarity function $\rho(A, A')$ is a metric (distance function) that is defined on the Cartesian product of an appropriate space of images A, A', \dots , as it is illustrated by examples listed below.

It is assumed that all images in the SRI are of the same type (binary, gray levels or color) and of the same dimensions $I \times J$, say. Concerning color images we additionally assume that all of them use the same color coding scheme (e.g., RGB or HSI).

Color RGB images. Color images, coded as RGB (red, green, blue) channels, are represented as: $C = [c_{ij\gamma}]$, $C' = [c'_{ij\gamma}]$, where $\gamma \in \{R, G, B\}$. Again, $c_{ij\gamma}$ and $c'_{ij\gamma}$ are usually restricted to $[0, 1]$ or to $[0, 255]$ for each channel. Later on we shall use the following distance function for RGB color images:

$$\rho_C(C, C') = \sum_{i=1}^I \sum_{j=1}^J \sum_{\gamma \in \{R, G, B\}} (c_{ij\gamma} - c'_{ij\gamma})^2. \quad (1)$$

Clearly, gray-level and binary counterparts of (1) can also be used.

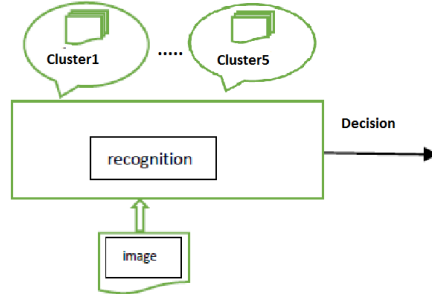


Fig. 1. A general scheme of on-line, image-driven decision making after the learning phase that provides clusters.

Details of clustering images – K -medoids. Having selected a distance function between images, one can cluster SRI using the well known K -means algorithm. However, for our purposes the method of K -medoids

(see [4], [7] for recent implementations of this algorithm) seems to be better suited. The reason is in that it returns – as its output – K images which are present in SIR. This would be of special interest when the second strategy (attach decisions to images and cluster) would be used. However, also when the first strategy is preferred (as in our case study), the K -medoids approach works better than the K -means for the following reasons:

1. the K -medoids method is more robust to noise and outliers (as it has features similar to the one-dimensional median),
2. the distances between images are calculated only once (important for large images),
3. for a long sequence of images the K -medoids method can be faster than the K -means method.

Denote by $C^{(n)}$, $n = 1, 2, \dots, N$ the images in the SRI and let \mathcal{C} be the collection of them. An image $\hat{C} \in \mathcal{C}$ is called the medoid of this set (see, e.g., [15] and the bibliography cited therein), if it minimizes the following distance function:

$$\sum \rho_C(\hat{C}, C^{(n)}), \quad (2)$$

where the sum in (2) is taken over all $C^{(n)} \in \mathcal{C}$ such that $\hat{C} \neq C^{(n)}$. We shall use this definition also for subsets of \mathcal{C} .

For arbitrarily selected $C_k \in \mathcal{C}$, $k = 1, 2, \dots, K$ define clusters \mathcal{C}_k , $k = 1, 2, \dots, K$ by attaching to C_k all the images $C^{(n)}$, $n = 1, 2, \dots, N$ such that they are closer to C_k than to other C_l , $k \neq l$ in ρ_C distance. Define the total distance as follows:

$$L(C_1, C_2, \dots, C_K) = \sum_{k=1}^K \sum_{C^{(n)} \in \mathcal{C}_k} \rho_C(C_k, C^{(n)}). \quad (3)$$

We are looking for $\hat{C}_k \in \mathcal{C}$ and \hat{C}_k , $k = 1, 2, \dots, K$ that minimize (3).

Below we provide a skeletal version of the K -medoids algorithm that is adapted to split all images from \mathcal{C} into K disjoint clusters. It can also serve as the definition of the K -medoids method.

Algorithm 1 – K -medoids algorithm for clustering images.

Step 0 Calculate and store a $N \times N$ distance matrix $D = [d_{nm}]$, where $d_{nm} = \rho_C(C^{(n)}, C^{(m)})$. Select initial medoids C_k , $k = 1, 2, \dots, K$ by drawing them at random from \mathcal{C} and insert them into a collection, \mathcal{CCM} say, of considered candidates for medoids.

Step 1 Form clusters \mathcal{C}_k , $k = 1, 2, \dots, K$ by attaching to C_k all the images from \mathcal{C} that are closer C_k than to other temporary medoids C_l , $k \neq l$.

Step 2 Calculate the total distance L between images and their temporary clusters as follows:

$$L(C_1, C_2, \dots, C_K) = \sum_{k=1}^K \sum_{C^{(n)} \in \mathcal{C}_k} \rho_C(C_k, C^{(n)}), \quad (4)$$

where $\rho_C(C_k, C^{(n)})$ is read out from the appropriate element of matrix D .

- Step 3** Select at random an image, say $C^{(*)}$, from \mathcal{C} , which is different than those in \mathcal{CCM} . $C^{(*)}$ is a candidate for a new medoid center. Using distances from matrix D find the medoid $C(j)$, say, that is closest to $C^{(*)}$ among all current medoids C_1, C_2, \dots, C_K .
- Step 4** Calculate the total distance function with $C(j)$ replaced by $C^{(*)}$:

$$L_{trial} \stackrel{def}{=} L(C_1, C_2, \dots, \underbrace{C^{(*)}}_{j-th}, \dots, C_K)$$

in the same way as in (4).

- Step 4a** If $L_{trial} < L(C_1, C_2, \dots, C_K)$, then: replace $C(j)$ by $C^{(*)}$, insert $C^{(*)}$ into \mathcal{CCM} and go to Step 1.
- Step 4b** $L_{trial} \geq L(C_1, C_2, \dots, C_K)$, insert $C^{(*)}$ into \mathcal{CCM} . If $\mathcal{CCM} = \mathcal{C}$ or if Step 4b was visited a large (prescribed) number of times in subsequent iterations, then STOP and provide current medoids C_1, C_2, \dots, C_K and the corresponding clusters $\mathcal{C}_k, k = 1, 2, \dots, K$ as the output of the algorithm. Otherwise, go to Step 3.

If the algorithm stops at Step 4b, then we consider its output either as the optimal solution $\hat{C}_k \in \mathcal{C}$ and $\hat{C}_k, k = 1, 2, \dots, K$ or its approximation. A faster implementation of K -medoids clustering of images, more in the spirit of the K -means algorithm, can be elaborated using general guidelines provided in [7]. In [4] one can find a survey of approaches to K -medoids clustering that are based on evolutionary algorithms.

2.2 On-line decision making

Medoids and clusters $\hat{C}_k \in \mathcal{C}$ and $\hat{C}_k, k = 1, 2, \dots, K$ form a base for on-line decision making, when a new image, C_{cur} say, is captured by a camera. As it was mentioned earlier, at this stage an expert has to attach decisions, $d_k, k = 1, 2, \dots, K$ say, to each cluster. Thus, we have the sequence $(\hat{C}_k, d_k), k = 1, 2, \dots, K$ at our disposal.

Algorithm 2 – On-line decision making algorithm.

- Step 1** Image acquisition: provides C_{cur} image.
- Step 2** Recognition: find cluster label $k_{cur} \in \{1, 2, \dots, K\}$ such that C_{cur} fits best $\hat{C}_{k_{cur}}$ among all clusters.
- Step 3** Decision: read out the corresponding decision $d_{k_{cur}}$ and go to the step Image acquisition.

An outline of this algorithm is presented in Fig. 1.

Recognition (Step 2) of this algorithm can be formalized in a number of ways. At least three of them are worth mentioning.

- R1)** Features can be extracted from medoids $\hat{C}_k, k = 1, 2, \dots, K$ and compared with the same features extracted from C_{cur} .
- R2** Among medoids $\hat{C}_k, k = 1, 2, \dots, K$ find the one that is closest to C_{cur} in ρ_C metric.
- R3** Calculate representative images, \tilde{C}_k say, for clusters $\hat{C}_k, k = 1, 2, \dots, K$, other than \hat{C}_k 's and find \tilde{C}_k that is closest to C_{cur} .

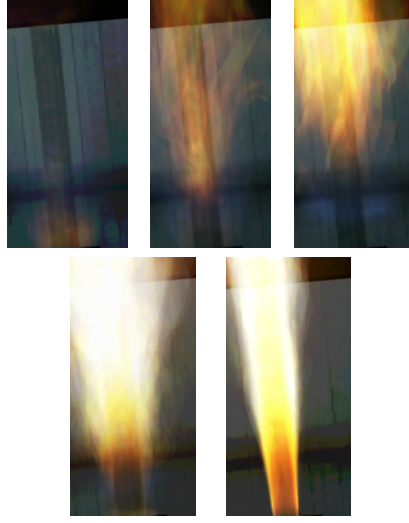


Fig. 2. Representative images – obtained by averaging K -medoid clusters. Classes labelled 1-5, row-wise from the upper left.

In the case study presented in the next section we have selected the arithmetic means of images contained in \hat{C}_k 's as the representative images \tilde{C}_k , $k = 1, 2, \dots, K$. Notice that this way is different than using K -means, since at the first stage we use the more robust K -medoids method for defining clusters and only after that – they are averaged.

For R3) approach (applicable also to R2)), one can estimate the probabilities $\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_K$ of the events: C_{cur} is from k -th cluster. A natural, compatible with ρ_C , way of doing this is as follows.

Algorithm 3 – Estimating probabilities

- Step 1** If for a certain cluster label \check{k} , say, we have $\rho(C_{cur}, \tilde{C}_{\check{k}}) = 0$, then set $\tilde{p}_{\check{k}} = 1$ and $\tilde{p}_k = 0$ for $k \neq \check{k}$.
- Step 2** Otherwise, calculate $l_k = 1/\rho(C_{cur}, \tilde{C}_k)$ and set $M = \sum_{k=1}^K l_k$.
- Step 3** Set $\tilde{p}_k = l_k/M$, $k = 1, 2, \dots, K$.

Remark 1 By construction, the largest \tilde{p}_k corresponds to that \check{k} , for which $\rho(C_{cur}, \tilde{C}_{\check{k}})$ is the smallest one. However, values of \tilde{p}_k 's still have diagnostic properties. Namely, if $\tilde{p}_{\check{k}}$ is not too much larger than (a) certain other(s) \tilde{p}_k 's, then we may consider this classification as uncertain. In the next section the following rule is applied: if for at least one k $0.8\tilde{p}_{\check{k}} < \tilde{p}_k$, then the classification is uncertain and – instead of taking decision $d_{\check{k}}$ – one may consider other actions.

3 Image-driven control of a gas burner – a case study

In this section we apply the methodology of image-driven control (Sec. 2) to support the decision on the rate of air supply to a gas burner. We refer the reader to [14], [8] for the problem statement and specific features of industrial gas burners control using a camera. For an alternative approach see [13].

Images of flames of an industrial gas burner ($N = 100$) are clustered into $K = 5$ groups using ρ_C similarity measure between images. The clustering was done according to general rules of K -medoid clustering (see Algorithm 1), implemented in Wolfram's Mathematica ver. 11. Then, representative images for each class were calculated, according to R3) – by averaging. The resulting images are shown in Fig. 2. In order to test the resulting cluster-based classifier a testing sequence containing 54 images (other than those in the learning sequence) was passed as its input. According to the rule described in Remark 1, nine images displayed in Table 1 (right panel) are selected as those that are potentially incorrectly classified. This means that about 84% of them were classified without doubts. A more careful analysis of the probabilities corresponding to potentially misclassified images (see Table 1 – right panel) reveals that six of them have been classified to class labelled as 2, while possible misclassifications were to classes 1 or 3. As we shall see, these potential misclassifications do not lead to essential problems. Thus, we can say that we have a sufficiently reliable and fast tool for the direct recognition of images.

Now, it remains to attach decisions to clusters and then, one can apply Algorithm 2 on-line. The analysis of Fig. 2 indicates that a proper mode of operating the burner is somewhere between Class 1 and Class 2, since Class 1 flames are blue, but it happens that the burner is loudly roaring. Flames from Class 2 indicate a slightly too low air supply. In order to ensure that the decisions are not too frequently changed, we assume that Class 2 is the proper one. Images from Classes 3-5 indicate a too low or much too low air supply. Thus, as decisions attached to each class, one may consider the following:

- d_1 – slightly reduce the air supply rate,
- d_2 – keep the air supply rate at this level,
- d_3 – slightly increase the air supply rate,
- d_4 – largely increase the air supply rate,
- d_5 – highly increase the air supply rate.

The above decisions are of a qualitative nature. Providing quantitative decisions would require much broader experimental data than we had at our disposal. Notice that the reason that leads to changes of decisions stems from fluctuations of methane contents in natural gas. A specific feature of gas burners of a moderate size that made it possible to apply the proposed approach is their almost immediate response for changing the air supply rate.

No	Cl	probabilities of classes					
18.	2.	0.31	0.35	0.13	0.09	0.11	
20.	2.	0.16	0.25	0.24	0.16	0.18	
21.	2.	0.17	0.26	0.22	0.16	0.19	
22.	2.	0.33	0.37	0.12	0.08	0.10	
23.	2.	0.17	0.29	0.23	0.14	0.17	
25.	2.	0.20	0.28	0.25	0.12	0.14	
32.	4.	0.11	0.14	0.19	0.31	0.25	
34.	5.	0.12	0.16	0.19	0.23	0.30	
42.	5.	0.11	0.13	0.18	0.25	0.33	

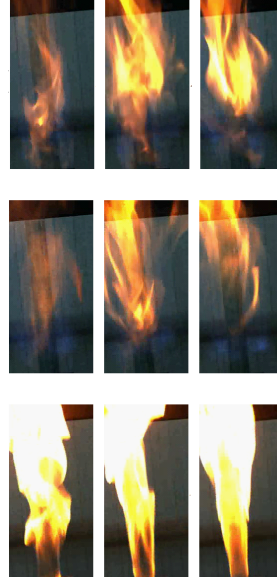


Table 1. Right panel – images selected according to Remark 1 – as potentially misclassified. Left panel – the corresponding probabilities of class matching.

4 Concluding remarks

A direct approach to image-driven decision making has been proposed. It consists of the clustering phase – that can be time-consuming – and a decision-making phase, which is sufficiently fast for on-line applications, since no image processing is necessary. This approach has a wide range of possible applications. One of them, namely, air supply rate control of an industrial gas burner is reported in the paper. The approach is flexible and may have many variants that differ in the way of selecting the method of clustering and the similarity function as well as by forming images representative for classes.

Being widely applicable, the IDDM cannot be universal. Its applicability is limited to cases when relatively large portions of images are changing. It can be extended to certain classes of dynamic systems when one has to take into account that a memory of older states is present in the system, but the extension is outside the scope of this paper.

References

1. Cai Weiling, Chen Songcan, and Zhang Daoqiang. Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern recognition*, 40(3):825–838, 2007.
2. Chapman Kenneth W, Johnson W Carroll, and McLean Thomas J. A high speed statistical process control application of machine vision

- to electronics manufacturing. *Computers & Industrial Engineering*, 19(1):234–238, 1990.
3. Davies E Roy. *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.
 4. Hruschka Eduardo Raul, Campello Ricardo JGB, Freitas Alex A, et al. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155, 2009.
 5. King Tim. Vision-in-the-loop for control in manufacturing. *Mechatronics*, 13(10):1123–1147, 2003.
 6. OLeary Paul. Machine vision for feedback control in a steel rolling mill. *Computers in Industry*, 56(8):997–1004, 2005.
 7. Park Hae-Sang and Jun Chi-Hyuck. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.
 8. Rafajłowicz Ewaryst, Pawlak-KruczekHalina, and Rafajłowicz Wojciech. Statistical classifier with ordered decisions as an image based controller with application to gas burners. In *International Conference on Artificial Intelligence and Soft Computing*, pages 586–597. Springer, 2014.
 9. Rafajłowicz Ewaryst and Rafajłowicz Wojciech. Iterative learning in repetitive optimal control of linear dynamic processes. In *Artificial Intelligence and Soft Computing. ICAISC 2016. Lecture Notes in Computer Science, vol 9692.*, pages 705–717. Springer, 2016.
 10. Rafajłowicz Ewaryst, Wietrzyk Jerzy and Rafajłowicz Wojciech. A computer vision system for evaluation of high temperature corrosion damages in steam boilers. In *Intelligent Systems in Technical and Medical Diagnostics*, pages 391–402. Springer, 2014.
 11. Rafajłowicz Ewaryst, Wnuk Marek and Rafajłowicz Wojciech. Local detection of defects from image sequences. *International Journal of Applied Mathematics and Computer Science*, 18(4):581–592, 2008.
 12. Rigelsford Jon. Industrial image processing: Visual quality control in manufacturing. *Sensor Review*, 21(2), 2001.
 13. Skubalska-Rafajłowicz Ewa. Sparse random projections of camera images for monitoring of a combustion process in a gas burner. In *CISIM*, page this volume. Springer, 2017.
 14. Wójcik Waldemar and Kotyra Andrzej. Combustion diagnosis by image processing. *Photonics Letters of Poland*, 1(1):40–42, 2009.
 15. Xu Rui and Wunsch Donald. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.
 16. Yan Hao, Paynabar Kamran and Shi Jianjun. Image-based process monitoring using low-rank tensor decomposition. *IEEE Transactions on Automation Science and Engineering*, 12(1):216–227, 2015.

Acknowledgements The research of the first author has been supported by the National Science Center under grant: 2012/07/B/ST7/01216.