CHALMERS


# Source Camera Classification and Clustering from Sensor Pattern Noise

## Applied to Child Sexual Abuse Investigations

*Master of Science Thesis in Intelligent Systems Design*

JOSEF EKLANN
JOHN LUNDBERG

Department of Applied Information Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2012
Report No.2012:008
ISSN: 1651-4769

Source Camera Classification and Clustering from Sensor Pattern Noise
Applied to Child Sexual Abuse Investigation
JOSEF EKLANN & JOHN LUNDBERG



Technical report no 2012:008
Department of Applied Information Technology
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Göteborg, Sweden 2012

Source Camera Classification and Clustering from Sensor Pattern Noise
Applied to Child Sexual Abuse Investigation

JOSEF EKLANN & JOHN LUNDBERG
Department of Applied Information Technology
Chalmers University of Technology

## ABSTRACT

In police investigations concerning child sexual abuse crimes, the most important evidence is often digital images. Due to the large number of images in such cases, tools are needed to reduce the amount of manual work. A desirable feature currently not available to law enforcement personnel is the ability to reliably tell if an image was captured by a specific camera or not. Another useful feature is the ability to cluster images based on the source camera. These goals can be achieved by the use of Sensor Pattern Noise and in this study methods to extract the noise from images will be evaluated. Furthermore different clustering methods and new clustering heuristics, such as pre-clustering based on camera model, is evaluated. To improve clustering results correlation between reference patterns constructed from already clustered images is studied.

The evaluation of the denoising algorithms concluded that the color decoupled version of the Mihçak denoising filter was superior to the other tested methods. The correlation between reference patterns from clusters of images was concluded to be highly dependent on the number of images in the clusters. The introduction of pre-clustering based on if two images where from the same camera, using features from the image and noise and a trained classifier, decreased the time consumption of the clustering algorithms considerably, thus making the clustering methods more feasible when the amount of images is large. By merging the noise from clusters into reference patterns more images were grouped together than when only single image noise patterns were compared to each other.

Keywords: Child sexual abuse, Source camera identification, Source camera clustering, Sensor pattern noise

Source Camera Classification and Clustering from Sensor Pattern Noise
Applied to Child Sexual Abuse Investigation
JOSEF EKLANN & JOHN LUNDBERG
Institutionen för tillämpad informationsteknologi
Chalmers tekniska högskola

## SAMMANFATTNING

I utredningar angående sexuella övergrepp på barn, är digitala bilder ofta de viktigaste bevisen. På grund av det stora antalet bilder som ofta förekommer i dessa fall behövs verktyg för att hålla nere andelen manuellt arbete. En användbar funktion som polismyndigheter ännu inte har tillgång till är möjligheten att med hög konfidens kunna avgöra om en digital bild är tagen av en specifik kamera eller inte. Ytterligare en användbar funktion är möjligheten att gruppera bilder baserat på vilken kamera de kommer från. Dessa funktioner kan implementeras med hjälp av sensorbrusmönster. I denna studie utvärderas olika metoder att extrahera detta brusmönster. Vidare utvärderas metoder att gruppera bilder, samt nya tillvägagångssätt så som för-gruppering baserat på kameramodell. För att förbättra resultaten vid gruppering utforskas effekten av antalet bilder vid jämförelser av grupper.

I utvärderingen av brusextraheringsmetoder fastslogs att den bäst lämpade metoden var färg-separerad Mihçak-filtrering. Korrelationen mellan grupper av bilder visade sig vara starkt beroende av antalet bilder i vardera grupp. Introduceringen av för-gruppering baserat på kameramodell, klassificierat av ett tränat neuralt nätverk, reducerade tidsåtgången för gruppering avsevärt. Detta gör den presenterade grupperingsmetoden mer lämpad för stora datamängder än tidigare metoder. Genom att sammalslå brusmönster från bilder i samma grupp kunde fler bilder grupperas jämfört med metoder då endast enskilda brusmönster jämförts.

Nyckelord: Kameraidentifikation, Klustring, Sensorbrusmönster

# ACKNOWLEDGEMENTS

We would like to thank everyone at NetClean Technologies where the thesis work has been carried out. Specifically we want to thank Johann Hofmann, our supervisor, and Mattias Shamlo for their inspirational support, their time and their enthusiasm in this thesis. We would also like to thank our examiner Claes Strannegård.

# CONTENTS

# ABBREVIATIONS

ANN – Artificial Neural Network
CFA – Color Filter Array
CFS – Correlation based Feature Selection
CSA – Child Sexual Abuse
EXIF – Exchangeable Image File Format
FPN – Fixed Pattern Noise
IQM – Image Quality Metrics
PNU – Pixel Non-Uniformity
PRNU – Photo-Response Non-Uniformity
RNP – Residual Noise Pattern (Noise extracted from at least one image)
SNR – Signal to Noise Ratio
SPN – Sensor Pattern Noise (The actual noise pattern of a sensor)

# 1 INTRODUCTION

In investigations concerning child sexual abuse (CSA), digital images are central in the body of evidence, and the analysis of such images is therefore of substantial interest to law enforcement. However, seized hardware often contains image sets so large that manual analysis of the material is not feasible. Software, facilitating the processing power of computers, is of help in the process of manual analysis, and can also be capable of analysis that cannot be done by hand at all.

With larger image sets confiscated each year, easier distribution of CSA material is one of the downsides of the continued increase in Internet connectedness across the world. Not only is access to the material faster than ever, the cost of storing becomes cheaper and cheaper. In these sets of images, the size of which can be in the order of hundreds of thousands [1], each picture depicts a crime scene and as in other criminal investigations, being able to pin a person or an object to a crime scene can prove vital in trial. One way of doing this is by visible birthmarks or other visual characteristics of the offender [2], but such body marks may not be present.

In the software that is used in CSA investigations today, content based techniques are used to recognize objects that are visible in different images, thus pairing crime scenes together when the rest of the visuals of the scene might be changed. This gives structure in the vast amounts of collected data and can possibly increase the likelihood of connecting an offender to several images if he or she can be successfully identified in one of the connected images [3]. This is a good example where computers can give structure to seized material in a way that cannot be done manually because of the vast amounts of data.

In digital cameras, the scene is captured by a sensor that consists of a matrix of photocells that reacts to light. Due to imperfections in the manufacturing of these sensors, each photocell has a slightly different sensitivity to light. This, together with other imperfections, gives rise to noise in the image, called sensor pattern noise (SPN). Recent studies have shown that it is possible to classify the source camera from digital images, by comparing the SPN extracted from the images [4]. This technique not only makes use of the processing power of the computer to analyze more pictures than a human could do but also uses intrinsic features of images that are not visible to the human eye, therefore providing entirely new possibilities.

By implementing and extending these methods in software used by the law enforcement, it is the ambition and motivation of this thesis to aid in the identification of perpetrators. This can be accomplished by associating source cameras to images and assist in associating different crime scenes by clustering images that have been taken by the same camera, even when the camera is not available to the investigators. This would provide new means of identifying and helping victims.

## 1.1 Purpose and Scope of the Thesis

The thesis is carried out in cooperation with the company NetClean Technologies with the goal of providing a thesis result suitable for incorporation in the product NetClean Analyze, though the actual integration will be beyond the scope of the thesis. The software suite Analyze is used by police authorities of several countries to support in CSA investigations.

In order to classify the large amount of images this thesis must provide novel, more time efficient, solutions that builds upon the research done by others but scales better when applied to the vast data sets that is handled in CSA investigations today.

This thesis compares and discusses efficiency and performance of different denoising filters that are used to extract the SPN, as well as different measurements and features derived from the SPN and image that are useful when classifying the source camera of a digital image.

In the process of extracting noise from images using denoising filters, there will be no attempts to design entirely new denoising algorithms. Instead the focus will be to evaluate and improve on existing methods.

Cropping, resizing, heavy lossy compression and similar image operations, while common, distorts the SPN and thus makes recognition of this pattern much harder or infeasible, therefore the effort of the project will go into developing techniques to recognize the noise pattern of original images that has been taken by the camera's maximum resolution and that has not been modified in any of the mentioned manners. Furthermore, since the results of this thesis should be applicable to the amounts of images that are seized by the police in CSA investigations, only the middle $512 \times 512$ pixels of each image will be used.

There are two typical use cases that are of interest and will be considered in this thesis:

- From a set of images and access to a camera, determine which of the images in the set were taken with the given camera. In this thesis referred to as source camera classification.
- From a large dataset, form clusters of images taken by the same camera. In this thesis referred to as source camera clustering.

# 2 THEORETICAL FRAMEWORK

Camera source identification evolved as a research field after the success of the digital camera, but identification using noise in images is a relatively new field, with Fridrich et al proposing the first methods in 2006 [4]. The noise in a digital image originates from a number of different sources (See Figure 2.1). The noise components that differ from image to image is collectively called *shot noise*, similarly the noise components that remain approximately the same from image to image is called *pattern noise* [4].

The shot noise consists of a number of components [5], but is not interesting from a forensic point of view since its random nature makes it unfit for any identification.

The pattern noise consists of *fixed pattern noise* (FPN) and *photo-response non-uniformity* (PRNU). The FPN is caused by dark currents, and the signal to noise ratio (SNR) will vary with exposure times, light intensity and temperature. Furthermore the FPN is often suppressed by modern digital cameras, and it is thus not useful for camera source classification [4]. The PRNU is caused mainly by *pixel non-uniformity* (PNU) that is imperfections in the sensor leading to different sensitivity to light at each pixel. The PNU component has a constant SNR which makes it well suitable for forensic applications. The remaining component of the PRNU is *low frequency defects* such as light refractions on dust particles in the camera, optical imperfections in the lens etc.

Due to the random nature of the shot noise, it will be suppressed when averaging the noise component from multiple images. The pattern noise on the other hand will be enhanced when averaging multiple noise components as it is the only part contained in all residual noise patterns. The dominant part of the PRNU is PNU, also the low frequency defects can differ due to zoom, lens cleaning etc. Hence the estimated PRNU is also a good approximation of the PNU. PNU in the context of camera source identification is often referred to as *sensor pattern noise*, SPN.

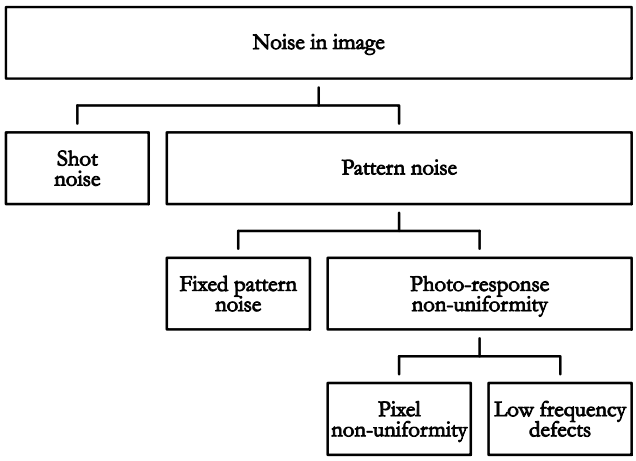


Figure 2.1 A hierarchical view of the noise components in an image captured by a digital camera

## 2.1 Source Camera Classification

The task of identifying which camera in a finite set has produced a given image, often called source camera identification, has been well researched [4], [6], [7], [8], [9], [10]. Even though this is not what this thesis is aimed to, the problem is similar. The most important difference between earlier studies and this thesis is that earlier studies have made the assumption that the source camera of each image is in a small set of known cameras. In the context of experiments made in studies this assumption is correct, but in a real situation this assumption cannot be made.

In the typical setting of a real-world application given a set of images and a camera, there is no knowledge as to if there are images in the set from the camera. The aim is to find out if this is the case. Therefore the goal must be to determine if an image was taken with the camera or not, rather than identifying the camera in the set most likely to have captured the image.

Previous studies have pointed out that an SPN estimation of multiple images is more accurate than that of only one image. Ideally one can by having access to the camera produce images to make a reference pattern, but there might also be situations where existing images are known to be from the camera, but more images cannot be produced. Therefore the proposed method must be able to use an arbitrary number of images as a reference pattern.

### 2.1.1 Extracting Sensor Pattern Noise

In order to extract the noise from the image, the general approach is to apply a denoising filter, and subtract the denoised image from the original image. The result of this operation is the residual noise, which is an estimation of the SPN. One can get a more accurate estimation by using multiple images from the same device, as shown in [6]. This can be done by averaging the estimates.

The accuracy of the SPN estimation depends on a number of parameters. For example images with high luminance gives estimates with lower variance, many denoising filters also performs better on images with smooth scenes. The most influential parameter is however the choice of denoising filter. In earlier studies a number of different filters, called Mihçak's, Argenti's, CBM3D and PCAI respectively, have been proposed and tested for the purpose of camera identification. [11], [12], [13], [14].

Studies have shown that use of CBM3D is slightly more accurate than Mihçak and Argenti, while Mihçak and Argenti have comparable accuracy when using noise correlation to determine the source of an image [7]. CBM3D is very slow in comparison to the other denoising algorithms but it can be set to use a faster approximate approach.

A recent study [14] introduced the use of the PCAI filter in the context of camera source identification and the results appear very promising, suggesting that the performance of the PCAI filter exceeds both CBM3D, Mihçak and Argenti filtering.

Another approach to extract the SPN is presented in [8] where the image is decoupled into four subimages prior to the denoising. This approach was reported to outperform the standard approach. In most modern digital cameras each photocell only captures one of the red, green and blue colors. To produce the missing values in each color channel the sensor values are interpolated in a demosaicing process. The filter that determines which color is captured by which photocell is called a Color Filter Array(CFA), as illustrated in Figure 2.2, and it is often defined in a $2 \times 2$ matrix with two green, one red and one blue cell [15]. By dividing each channel into four subimages, one (or two in the case of the green channel) of the four images will contain only true measurements while the others are interpolated. Thus the denoising of the subimage with original values will not be affected by any demosaicing algorithm.

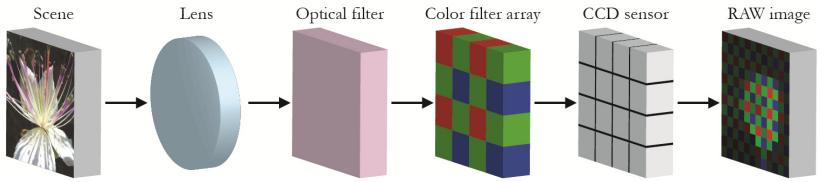

Figure 2.2 Digital camera pipeline

Since both CBM3D and PCAI has been shown to give better results than the Mihçak filter these two will be considered in the thesis, together with the Mihçak filter with and without color decoupling. CBM3D will be used with the fast profile. Using the Argenti filter did not improve the performance in the paper that it was introduced and it will not be discussed further in this thesis. Additionally a new variant of the color-decoupled Mihçak filter described below will be considered.

#### 2.1.1.1 Details of new Version of Decoupled Mihçak's Filter

When decoupling the image, the neighborhood of each pixel is affected. Pixels in the decoupled image are closer than in the original. The neighboring pixel at radius $r$ from pixel $x$ in a subimage is at radius $2r$ from $x$ in the original image. This motivates the choice of a smaller neighborhood to estimate the local variance in the subimages. This can also be a way to reduce the time consumption of the denoising algorithm, while hopefully maintaining a good performance.

As illustrated in Figure 2.3, all the pixels contained in the neighborhood of size $9 \times 9$ around pixels $(4,4)$, $(5,4)$, $(4,5)$ and $(5,5)$, in the original image is after the decoupling either contained in the neighborhood of size $5 \times 5$, or in another subimage. By first

decoupling the image, and then using the window size radius $W = \{1,2\}$ instead of $W = \{1,2,3,4\}$ to estimate the local variance, the time consumption (compared to the standard Mihçak filter) can be reduced, while the quality of the filter remains close to the same as when the larger neighborhoods where used.

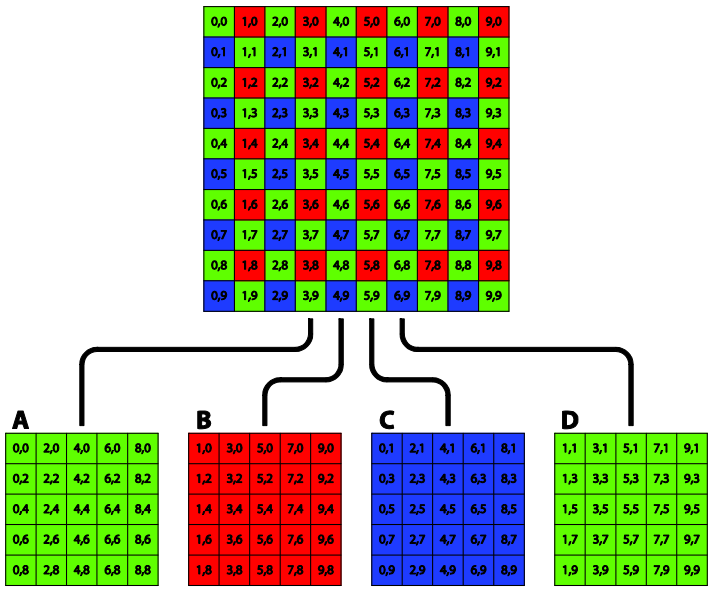


Figure 2.3 Color-decoupling of an image

### 2.1.2 Using SPN to Determine Source Camera Device

When SPN was introduced to identify the source camera, the means of comparison was correlation between two patterns which was shown to work well when the pattern of an image is compared to the reference pattern of a device [6]. The formula that was used to calculate the correlation between two noise patterns is:

$$Corr(n,m) = \frac{\sum_{x=1}^{w}\sum_{y=1}^{h}(n_{xy}-\mu_n)(m_{xy}-\mu_m)}{\sqrt{\sum_{x=1}^{w}\sum_{y=1}^{h}(n_{xy}-\mu_n) * \sum_{x=1}^{w}\sum_{y=1}^{h}(m_{xy}-\mu_m)}}$$

Where $w$ is the width of the images, $h$ is the height of the images, $i_{xy}$ is the value of image $i$ at pixel $(x, y)$, and $\mu_i$ is the mean of all pixels in image $i$.

When the device is unknown, and thus a device reference pattern is not present, one needs to compare the noise from different images, which ideally would give a high correlation if they have been acquired using the same device, or a low correlation if the opposite is true. The scene of the image and random shot noise also affects the correlation, and might cause misleading correlation values.

## 2.2 Source Camera Clustering

An algorithm for clustering of noise patterns was proposed by Caldelli et al. [16] that begins by calculating a comparison matrix, where each noise pattern is correlated with all other noise patterns. The noise patterns used in the paper by Caldelli et al are modified by a method that is supposed to enhance the noise. This method is patented in Europe and has thus not been considered in this thesis, instead the clustering algorithm worked on the extracted noise, without any post processing applied. From the beginning of the algorithm each image is considered as a cluster. The algorithm then iteratively picks the cluster pair between which the correlation is the highest. These two clusters are merged and the correlation values from these two clusters to other clusters are averaged. After each iteration a silhouette coefficient is calculated. This coefficient measures how well the clusters are separated from each other, as well as measuring how tightly the images in each cluster are connected to each other. One coefficient is calculated for each cluster by subtracting the average of the correlations within the cluster from the average of the correlations to other clusters. This coefficient is calculated and averaged over all clusters to get the silhouette coefficient of the entire clustering.

When all images have been merged to one cluster, the iteration with the lowest silhouette coefficient is chosen. Since correlation is a costly operation, the fact that the correlation between all images is calculated is a drawback of this algorithm and one would not want to do this on a big set of images. To decrease the time of the clustering process this method must be complemented by other approaches.

An interesting aspect of this problem is that one gets a more reliable correlation result between two noise patterns if they have been averaged over many images, thus a good clustering algorithm for this problem could preferably take advantage of this.

Since only images in their maximal native resolution will be considered a possible heuristic for clustering could be to start by dividing the images into subsets based on the sizes of the images. This reduces the size of each set to be clustered, and thereby the number of necessary comparisons and the time consumption. Since this thesis however is focused on the classification and clustering of noise patterns, this approach will not be considered during the evaluation of clustering algorithms in the thesis.

### 2.2.1 Model Clustering

To decrease the amount of correlations that needs to be calculated one idea is to do an initial clustering, based on something else than the correlation of noise. A natural idea is to base this clustering on what camera model, instead of unique device, that has captured the image. If one can achieve a fast comparison between two images, that can fairly separate different models, this can be used as a first step to reduce the number of correlations to compute.

Before SPN was recognized as a means of identifying the source device of digital images, attempts was made to use Image Quality Metrics and other measurements on the image and on different transformed representations of the image, for identification of the source [9]. These features have later been used as a complement to measurements that can be derived from the SPN when determining which camera device or model a specific image was taken with. More features that were specifically intended to indicate the camera model was tested by Filler et al. [10]. The features are generally statistical measures on the image and/or noise, or correlations between the noise and modified versions of the noise. There are existing methods that aim to find out from an image what CFA the capturing camera had [17], and this was also used as a candidate feature. All tested features are described in detail in Appendix A. While it could take some time to calculate the features of each individual image, the comparisons between the feature sets of the different images can be significantly faster than calculating the correlation, depending on the used classifier.

#### 2.2.1.1 Feature Selection

To help machine learning algorithms to make sense of training data it is sometimes beneficial to reduce the feature space by removing redundant or irrelevant features. Decreasing the number of features naturally also decreases the time needed to calculate all features of an image. To select which features that are indicative of the label of test data there are several existing methods. Correlation based feature selection (CFS) is fast and takes into account if the features correlate with the already selected features, but the drawback is that locally indicative features might be lost in the process. To counteract the drawback of CFS, after a base set of features has been selected using CFS, one can select more features using a wrapper selector, which is a slower but more exhaustive and precise method [18].

The idea behind the wrapper selection algorithm is to incorporate the classifier in the algorithm that selects the features, since some classifiers can make use of some features, and others not. To select more features one tests all, or a subset of all, possible feature subsets and determines if adding a subset of the remaining features increases the performance of the classifier [18].

#### 2.2.1.2 Classification by Features

The input to the classifier that should determine if two images were captured by the same camera model will be the result of a comparison between two feature sets, a vector of high dimensionality (the same number of dimensions as the number of features). The comparison methods of different features are further described in section 3.6.1.

One common type of classifiers are artificial neural networks (ANN) which is a classifier inspired by the human brain. It consists of a set of neurons linked together, which each performs a simple calculation. A common configuration is to organize the neurons in layers where the output of one layer is linked to the next. The links in the network each have a weight, and by setting these weights, one affects the behavior of the network. To adapt a neural network to the problem at hand supervised learning is often used. More specifically one can train a neural network by feeding it data, and altering the links when its prediction is wrong, using the backpropagation algorithm. The stochastic version of the backpropagation algorithm handles the training data in a random order and updates the network weights after each considered training instance [19].

The reason that ANNs was favored over other forms of classifiers, most notably support vector machines (SVM), was the combination of good classification performance and a very good execution time. The number of comparisons needed when comparing every instance to every other instance grows in $O(n^2)$, where $n$ is the number of instances. The time it takes for the classifier to give a result is thus very important.

# 3 METHOD

After the first method of identifying a source camera from the SPN was proposed, many research papers on how to improve the identification using other, possibly improved, approaches, has been published. Therefore a large portion of the work was focused on implementing and testing these suggested ideas as well as trying new ones, in order to find the optimal method in the setting of the thesis.

## 3.1 THE OUTLINE OF THE STUDY

The work was divided into two main problems based on the two features previously described as useful in CSA investigations. The first area of research was to assess existing ways of determining if two images were taken by the same unique device, and apply this to the problem of identifying images in a set taken by a given camera. The second area was to use these methods to cluster images, where one cluster should represent one camera device.

## 3.2 TEST IMAGES

The images used in this study were collected from image-sharing portal *Flickr.com*, using their API. Each candidate image was checked so that it fulfilled the following criteria:

- The image contains EXIF data about manufacturer and model
- The image is in the highest native resolution of the model
- The image has not been digitally zoomed, according to EXIF data
- The EXIF data does not indicate that photo manipulation software has been used on the image.
- The image is in landscape orientation. This guarantees that the orientation of the sensor is the same for all images.

Many images were collected in this manner for use in the study. The images were separated into sets of different cardinality, so that quick evaluation could be done on the smaller sets, while more important or less time consuming tests could be carried out on the larger image sets. Cameras are distinguished by the camera model and Flickr user account.

An image in a lower resolution than the highest native to the source camera will have a noise component that is equally scaled down. This means that it cannot be successfully compared to a noise pattern from an image of the native resolution without further processing. It can however be compared to other noise patterns extracted from images of the same size. The classification performance will likely be lower in this case since the noise has been downsampled.

## 3.3 TEST EVALUATION

When testing different denoising filters or classification methods etc. the means of comparison used is the so called F-Measure, unless otherwise stated. The F-Measure is a measure of how good a classification of data is, and in this study it is used to measure how separable a dataset is when hard thresholding is applied to classify the data.

The F-measure is also known as $F_1\ score$ and is calculated as:

$$F_1 = \frac{2 * true\ positives}{2 * true\ positives + false\ positives + false\ negatives}$$

The classification performance of the clustering algorithms was assessed using two constructed measurements. The first is the number of constructed clusters divided by the number of clusters that the test data actually consisted of, thus giving a ratio above 1 if the algorithm outputs too many clusters and below one if it groups the images to much. This measurement gives however no indication of the number of correctly grouped images.

To give an indication on how well the clustering algorithm manages to avoid clustering images from different sources together, each image in each cluster that was not from the same source as the majority in the respective cluster was counted. This number was divided by the total number of images to give an error rate.

## 3.4 SOURCE CAMERA CLASSIFICATION

With a more accurate SPN estimation one would naturally expect a higher rate of correct classifications, therefore different tests was carried out to find the best denoising filter and parameters for the task.

In the scenario that one wants to determine if an image was taken with a specific camera available for examination, one typically uses the following approach: Produce new images from the camera, and make a reference pattern from them. Then compare this reference pattern to the image in question by correlation. The reference pattern can be computed in different ways, for example by averaging the patterns.

To decide if a comparison was a match or not one method is to use a hard threshold, and regard results above the threshold as a match, and results below it as a mismatch. One can adjust the threshold for example to yield a low overall error rate, or to yield a low false positive rate.

### 3.4.1 Choice of Denoising Filter

The denoising filters considered in this thesis was compared by measuring how well one could separate comparisons made between an image and a reference pattern from the same camera and comparisons made between an image and a reference pattern from a different camera. This was measured with the F-Measure in the following way:

A set of 12 500 images from 198 different cameras, from 50 different models, was denoised by each denoising filter, giving one residual noise pattern (RNP) for each image. The dimension of the images was $512 \times 512$, cropped out from the center of the image. This part is less likely to contain fully saturated pixels, making the RNP a better estimation of the SPN [20]. The number of images used to create a reference pattern,$n$, was varied from 5 to 50, in intervals of 5. For each value of $n$, $n$ RNPs were selected randomly from the set of images belonging to a camera that in total had more than $n$ images. These RNPs was used to create a reference pattern which was correlated with the remaining images of that camera, and the same number of images randomly picked from other cameras, which also had more than $n$ images. In this manner the number of comparisons where the image and reference pattern came from the same camera will be the same as the number that came from different sources.

To get a measure on how well the denoising algorithms separated the instances coming from the same source from the ones coming from different sources, a threshold was adjusted so that the obtained F-measure was maximized.

To compare the time consumption of denoising filters the following test was conducted; from a dataset of 100 images collected from *Flickr.com*, all images were cropped down to sizes $256 \times 256$, $512 \times 512$ and $1024 \times 1024$. Denoising filters were then applied to all the images in the three cropped sets, and the mean time consumption to denoise a single image was calculated.

### 3.4.2 Denoising Filter Parameters

The Wiener filter, used in all denoising filters considered in this thesis takes one parameter, namely the estimation of the standard deviation of the noise, $\sigma_0$. In previous studies this parameter is commonly set to 5 [4]. In this thesis different values of $\sigma_0$ has been evaluated in order to make a more informed choice.

A set of 490 images from 12 cameras of 10 different models was randomly selected from the total test image dataset. The noise residuals of these images were then extracted. The noise residuals was then correlated to each other, and a decision threshold for the correlation was set to predict if the images were taken with the same camera or not. The threshold was chosen so that the F-measure of the prediction was maximized. This process was repeated for different values of $\sigma_0$. The results are shown inFigure 4.2.

### 3.4.3 Number of Images Needed for Reference Patterns

As described in section 3.4, camera reference patterns are made from a number of images known to be taken by the camera. To determine how many images that are needed to create a reference pattern of high quality the results from the same test that was defined in section 3.4.1 was evaluated.

## 3.5 Source Camera Clustering

In this thesis, three different methods of clustering are used. In each method one starts with a number of clusters (where each cluster consists of at least one image) and aims to further cluster the data, merging groups of clusters into new clusters. First a method called chain clustering is used to reduce number of clusters which are then fed to the Caldelli clustering method described in section 2.2. Finally the remaining clusters are clustered using a method called simple clustering. Chain clustering and simple clustering are new approaches presented in this thesis.

### 3.5.1 Chain Clustering

In this method all the images are only compared to the previous and the next image, meaning that the number of comparisons is $O(n)$ instead of $O(n^2)$. In each step one merges the clusters that has the highest correlation into one (by averaging the residual noise patterns of the images in the cluster), and then re-computes the correlation to its neighbors. This is iterated until no correlation is above a certain threshold. See Figure 3.1.

This method makes use of the assumption that images from the same camera are adjacent to each other in some ordering. This ordering can for example be by file names or timestamps. Also random ordering gives a high probability of neighboring images being from the same camera, as long as the number of cameras in the dataset is small.

For example, on a dataset with 10 images from one camera and 10 from another camera, the expected number of images from the same camera adjacent in random ordering is approximately 9.47. If the chain clustering finds 9 matches, the number of comparisons needed to be computed for the fast clustering is reduced from 190 to 55 (and 10 of them are already calculated in the chain clustering).

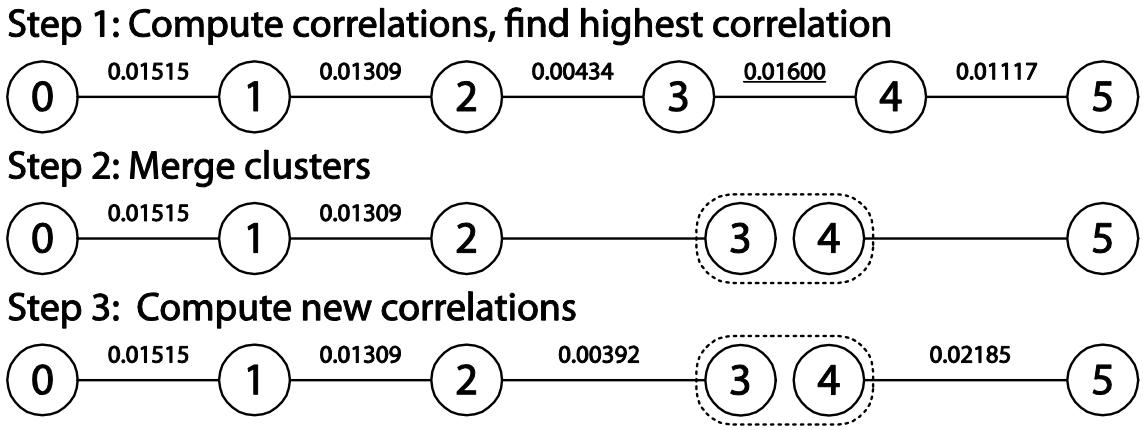

Figure 3.1 Explanation of chain-clustering

### 3.5.2 Caldelli Clustering

This clustering algorithm is described briefly in section 2.2, for the full description see [16]. In this study we use a slight modification, namely that only clusters with a correlation above a threshold are merged. When the algorithm is finished the residual noise of the clusters are merged by averaging, and new correlations are calculated.

### 3.5.3 Simple Clustering

This final clustering step once again uses the correlations between all the remaining clusters, and in each step merges the clusters with the highest correlation by averaging the residual noise. Then the correlations to this new cluster are updated and the procedure is iterated until no correlation is above a threshold.

### 3.5.4 Correlation Between Clusters of RNPs

With the proposed methods of clustering described in previous sections, correlations will be calculated between noise patterns averaged from clusters of RNPs. In order to investigate how the correlation values depend on the number of images in the involved clusters, reference patterns were built and correlated from the set of 12 500 images from 198 cameras of 50 different models. Different numbers of images were used in the clusters and the cluster patterns were correlated with each other. For every combination of differently sized clusters, a threshold was set so that the F-measure was maximized. By analyzing how the correlation depends on the sizes of the clusters, the threshold used in the clustering algorithms can be dynamically adjusted to compensate when the sizes of the clusters varies. A model for this was generated by applying curve fitting to the data points in the software Wolfram Mathematica.

### 3.5.5 Clustering Methods Evaluation

Three different purely correlation based clustering approaches will be evaluated on a set of 500 images from 12 different cameras collected from Flickr.com, in addition one approach also incorporating camera model clustering will be evaluated. The first approach is only using the Caldelli clustering algorithm as it was defined in [16]. The second approach is to do the chain clustering, Caldelli clustering and the simple clustering, in that order. The third approach is to divide the 500 images into four sets of

125 images, carry out approach 2 on each of the sets and then combine them as described in section 3.6.4. The fourth approach is to first cluster the images using camera model clustering, which is described further in section 3.6, then clustering each individual model cluster with the correlation based clustering methods and combining the clusters as described in section 3.6.4. To reduce any unfair advantage that the chain clustering might gain from the order of the images, the order was randomized in all evaluated clustering approaches.

## 3.6 Camera Model Clustering by Features

In a similar way as described in section 3.5, clustering on camera model can be done by using the comparison between features as input to an artificial neural network (ANN). While two feature vectors could be merged to a "reference feature vector", for example by averaging the values of the two vectors, comparisons of two merged feature vectors was not part of the classifiers training set, nor validation set. It is thus not certain how well the classifier would cope with such modified feature vectors and therefore only the Caldelli clustering algorithm will be used together with the classifier.

### 3.6.1 Comparing Feature Vectors

When the features presented in Appendix A has been calculated from an image one wants to compare the values to the same features derived from another image to produce an array of values suitable as input to a classifier such as an ANN. Since the representation of the features differ, a number of methods to provide a quantitative measure of the difference between two feature values has been used.

If a feature represented an array of values in between which the relation was more interesting than the magnitude of the values, correlation between the arrays derived from different images was used to compare them. The features that used this kind of comparison were the features derived from the linear pattern cross-correlation (see "Noise pattern statistics and characteristics" in Appendix A) and the six frequency signal features indicating periodicity in the demosaicing. Also the three color channel energy pairs was combined as one feature which was compared with correlation.

Furthermore the three color channel energy pairs was compared between images by calculating the euclidean distance between the three variables.

The Color Filter Array determination suggests which of the possible Bayer filters that exists in the source camera of an image, and thus when this feature is compared between images, it was simply determined if the same green, red or blue pattern was used, and indicating this with a one or a zero for each color channel. During the determination of the color filter array, another feature, the majority of the vote, is derived and this feature is compared between two images by multiplying the two ratios. Thus a high ratio in

both images is needed to give a high value and if the result of the comparison between the CFA-patterns is reliable this majority feature will indicate this.

To compare the positions of max- and min-values among the demosaicing artifacts between two images, a high number of pixels in the intersection between the "max-value" pixels and in the intersection between the "min-value" pixels, and thus also a low number of pixels in the intersection between the "max-value" set of one image and the "min-value" pixel set of the other image was taken as an indication towards the fact that two images were taken by the same camera, and vice versa.

Just calculating the absolute value of the difference of the values is perhaps the simplest way of comparing two features and this was applied in all other cases when the feature only consisted of a single value.

### 3.6.2 Feature Selection

To find the feature subset that best determines if two images were taken by cameras of the same model, when used as input to a neural network, the CFS algorithm, as implemented in machine learning software WEKA [21], was used to find an initial base set. After that, one feature was added at a time by evaluating which additional feature that increased the performance of the neural network the most, so called forward best first selection, inspired by wrapper selection which can be used as a complement to the CFS algorithm [18]. To calculate the performance when adding a new feature, eight neural networks were trained using stochastic backpropagation, and the error percentage of the different networks on the validation set was averaged. The feature that gave the lowest average error percentage was added and the process was iterated until adding more features did not increase performance. If adding a feature meant that the error percentage became one percentage point higher the feature was discarded and would not be considered during later iterations. This was done to decrease the time needed by the feature selector.

In addition to the classification performance of a feature, the time it takes to derive the feature from the image was considered in order to keep the time consumption of the final solution to a minimum.

### 3.6.3 ANN Learning Parameters

The values of the parameters that the ANN used during its learning phase was determined with cross validation, by finding the setting that had the lowest error percentage when the training and validation sets consisted of an equal number of positive (same camera model) and negative (different camera models) instances. To get an equal number of positive and negative instances, negative instances was discarded at random until this was achieved. The reason that negative instances was discarded was that the error percentage otherwise would have favored classifying negative instances

correctly to the point that the algorithms might classify all instances as negative to get the best solution, in terms of error percentage.

### 3.6.4 Divide and Conquer Approach

By dividing the image sets into clusters containing images from the same camera model, and then dividing these smaller sets into clusters from the same unique camera device the approach can be viewed as the dividing part of a divide and conquer algorithm, using the camera model as a heuristic. However, since the classifier does not have a 100 percent classification success rate there is a risk that images from one model is divided into two or more sets. There is also a risk that images from one unique camera device will end up in different clusters, therefore it is good to merge the model clusters afterwards. Also if the number of images is too big to keep all noise patterns in memory it would be good to consider smaller batches of images and then combining the results.

When two model clusters have been processed and divided into camera clusters the two model clusters are combined by calculating a correlation matrix of the camera clusters. Since the correlations between clusters in the same model set are already known, it is only the correlation between camera clusters from different sets that needs to be calculated. After this the Caldelli clustering algorithm described in section 2.2 and the simple clustering approach described in section 3.5.3 is used to merge clusters with high correlation. When two model clusters have been merged to one set, that set can be further merged with another pair of merged model clusters. By iterating this one finally gets only one remaining set where each cluster of images belonging to a unique device has been compared to each other cluster.

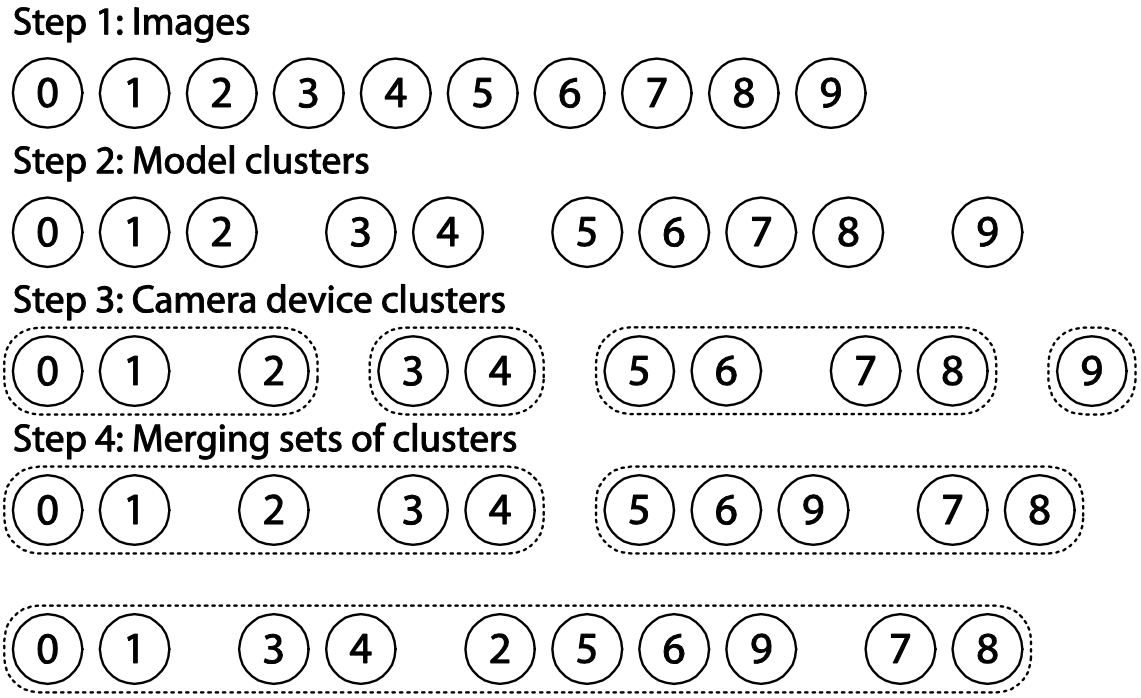


Figure 3.2 Divide and conquer approach to image clustering. In this example the model clustering is not perfect as can be seen when images 2 and 9 are misplaced, this is however corrected in the merge step.

# 4 RESULTS AND DISCUSSION

The test results and discussion has been divided into results that affect the denoising methods and residual noise patterns, and results that affect the clustering procedure. These results will be presented in this chapter.

## 4.1 DENOISING FILTER AND PARAMETERS

A denoising filter that produces a "cleaner" noise, a noise with less artifacts from image objects, could hopefully decrease the number of images needed to create a good reference pattern. Different methods achieve this to varying extent and in this part the results from the evaluation of the different methods will be presented.

### 4.1.1 DENOISING FILTER CLASSIFICATION PERFORMANCE

The different denoising filters mentioned in section 2.1.1 was tested as described in section 3.4.1, and as can be seen in Figure 4.1, the results confirm results from previous studies that stated that the PCAI filter is superior to the standard Mihçak filter (M). Furthermore the CBM3D algorithm performed better than the Mihçak filter if only a small number of images are available for the reference pattern. The algorithm that clearly performs best is however the decoupled version of the Mihçak filter (MD). Noteworthy is that the difference in performance between the MD filter with only the two smaller window size radiuses $W = \{1,2\}$, and the MD filter with all radiuses $W = \{1,2,3,4\}$, is very small.

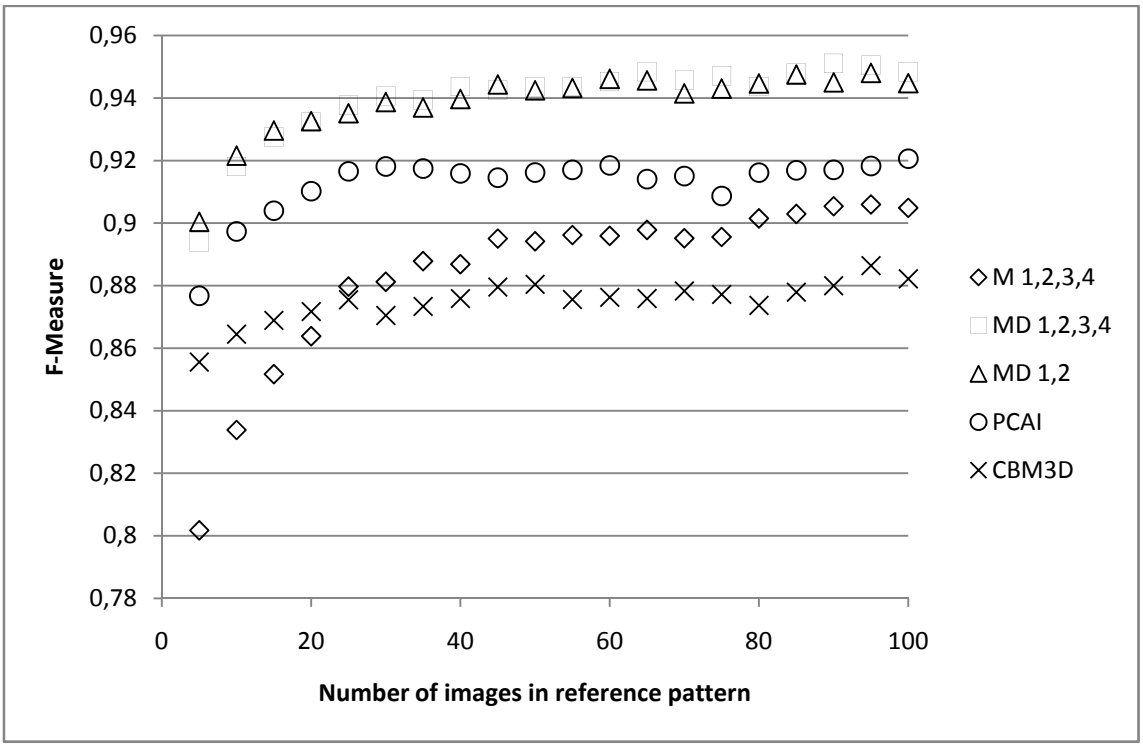


Figure 4.1 Comparison of different denoising filters

### 4.1.2 Number of Images in Reference Pattern

The more images used to create a reference pattern, the more random shot noise and artifacts from the scenes of the images are suppressed to form a noise pattern closer to the actual SPN of the camera. As can be seen in Figure 4.1 some denoising techniques require more images to get a good SPN approximation. The performance increase of the MD filter utilized in this thesis stagnates when the reference patterns are created from more than around 50 images. These results are consistent with earlier research [4]. Since the test images are ordinary photos collected from *Flickr.com*, the amount of scene artifacts is higher than if for instance only images of a clear blue sky were used, and thus the image set, from which the reference pattern is created, could be made smaller if only images with little scene artifacts are considered, as has been pointed out in previous research [6].

### 4.1.3 Denoising Filter Time Comparison

While the classification performance of the denoising filters is important, the time the filter uses to denoise images is also relevant.

| Image Size | M 1,2,3,4 | MD 1,2,3,4 | MD 1,2 | PCAI | CBM3D |
|---|---|---|---|---|---|
| $\mathbf{256 \times 256}$ | 1.087 | 0.808 | 0.368 | 0.091 | 1.617 |
| $\mathbf{512 \times 512}$ | 4.397 | 3.387 | 1.482 | 0.329 | 6.708 |
| $\mathbf{1024 \times 1024}$ | 18.452 | 15.174 | 6.004 | 1.579 | 28.838 |

Table 4.1 Comparison of time consumption of different denoising filters, in seconds

The system used to compute the noise and measure time had an Intel Core Duo T2050 processor and while the actual time consumption will vary between different systems, the PCAI algorithm was the fastest of the tested denoising filters. This is quite natural since the PCAI only works in the spatial domain, while the other filters first transform the image to the wavelet domain. The Decoupled Mihçak filter however was the filter with the highest classification performance and since the version with only 1 and 2 as window size radiuses had a time consumption which was deemed to be acceptable, this filter was chosen as the one best suited and thus this is the filter used in the rest of the thesis. This is motivated by the fact that a good classification performance can lead to a better clustering, which in turn results in less calculations of correlation between clusters. Since the denoising procedure only is done once per image, but the number of correlations is in the order of $n^2$, where $n$ is the number of images, reducing the number of clusters is of a higher priority.

### 4.1.4 Setting $\sigma_0$

When evaluating the performance of the denoising filter with different values of the $\sigma_0$ parameter, the highest F-measure observed was at $\sigma_0 = 2$, as can be seen in Figure 4.2, and thus in the rest of the thesis this value is used. One can also see in Figure 4.3 that the noise residual of the example image contains more artifacts from the image when it was denoised using $\sigma_0 = 5$ than $\sigma_0 = 2$.

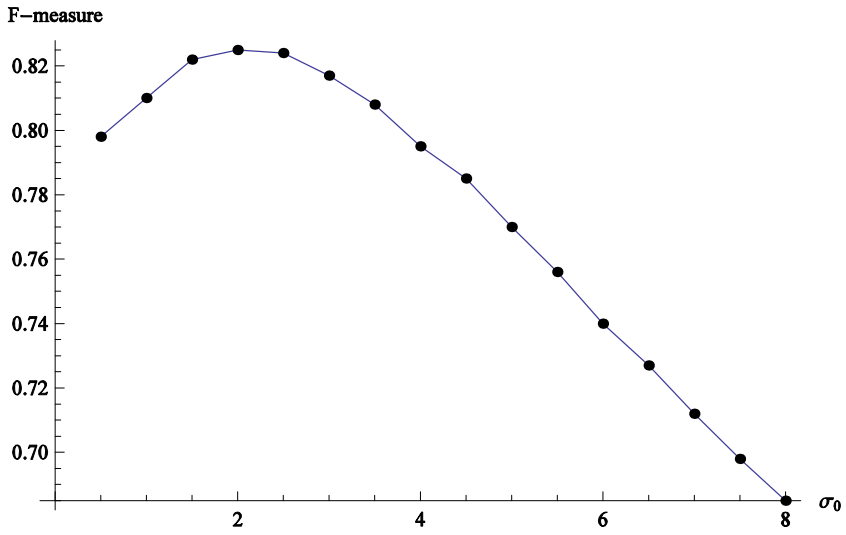


Figure 4.2 Evaluation of different values of $\sigma_0$

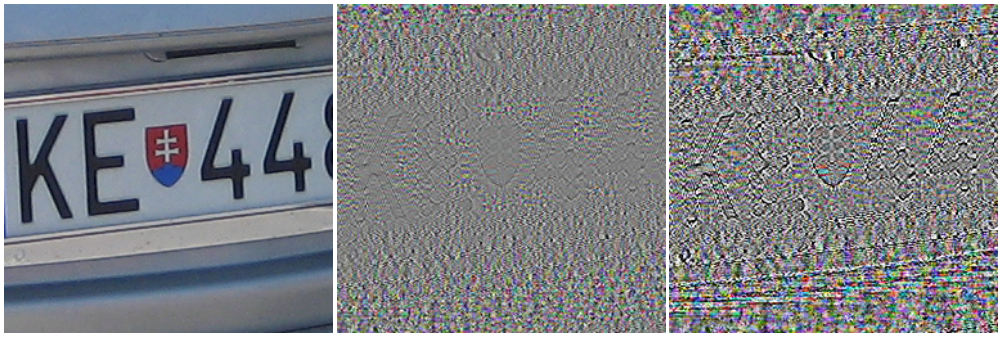

Figure 4.3 Original image, residual noise extracted with $\sigma_0 = 2$, and residual noise extracted with $\sigma_0 = 5$, noise intensity enhanced for better visualization

## 4.2 CLUSTERING

Different clustering methods needed evaluation of the building blocks such as evaluating how the correlation is affected when it is reference patterns that are evaluated, as well as model clustering needed evaluation of what features that gave the best model classification performance. These results together with the evaluation of the different approaches to clustering is presented in this chapter.

### 4.2.1 CORRELATION BETWEEN CLUSTERS OF IMAGES

The data collected in the test defined in section 3.5.4, shown in Figure 4.1, clearly shows that the optimal threshold varies with the number of images used to create the reference patterns.

To model how the threshold varies residual noise patterns from single images was correlated to find the optimal standard threshold, $t$, to use for image clusters of size 1 and 1 (i.e. comparing two images). This was found to be $t \approx 0.002551$. To find a good model that fits the data when the number of images increases, all data points were divided by the standard threshold. Thus the modified data points indicate how much the correlation between two patterns should be adjusted. By letting $(x + 1)$ and $(y + 1)$ be

the number of images in the reference patterns, a function $f(x, y)$ should give a factor that the correlation should be divided by before it is checked against the threshold.

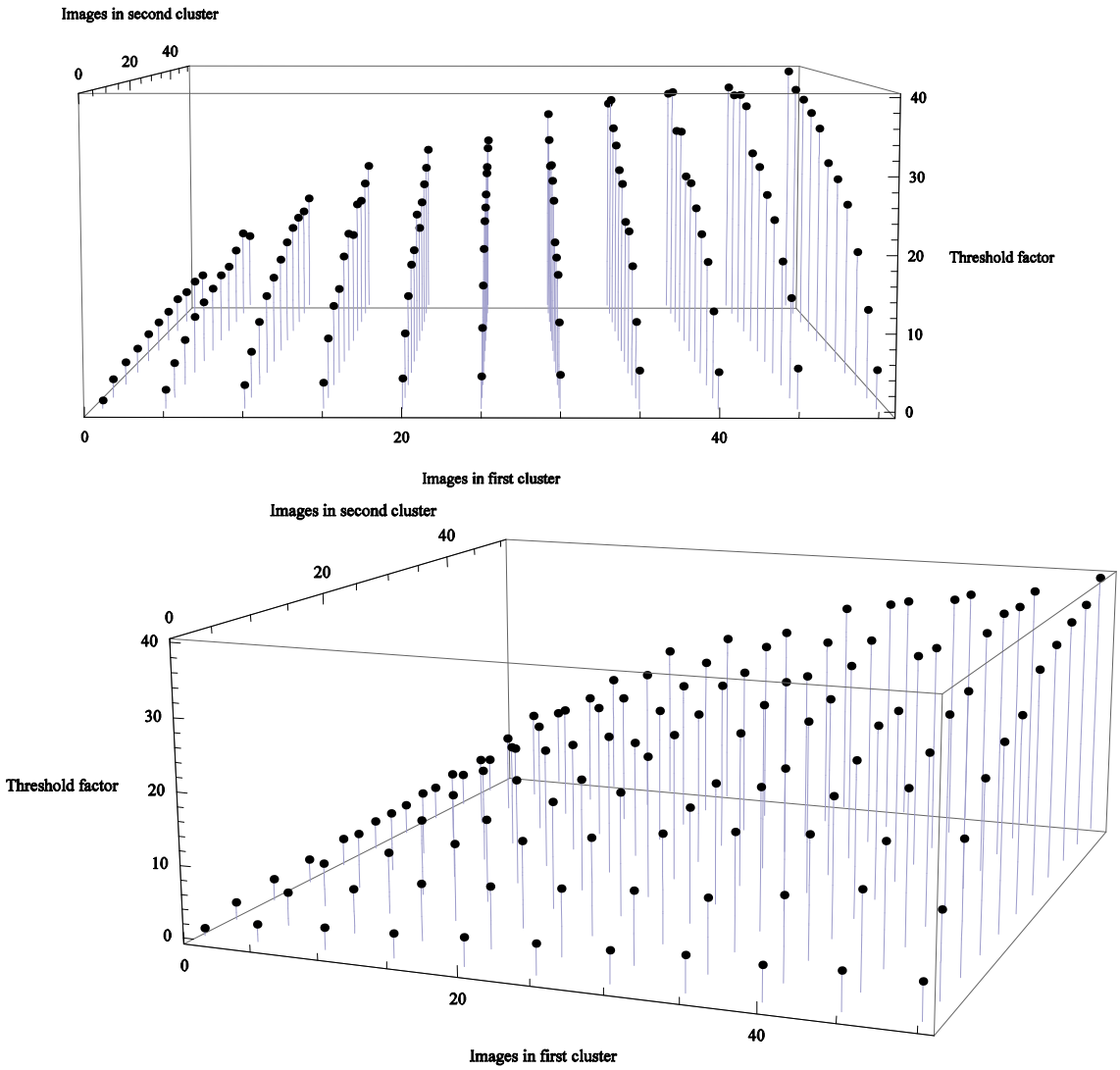


Figure 4.4 Optimal threshold for different cluster sizes, two different angles

Since the random noise and scene artifacts will be fully suppressed after a certain number of images it is reasonable that the function flattens for high values of $x$ and $y$ as well as that the condition $f(0,0) = 1$ holds. Furthermore the function should be symmetric, so that $\forall(x, y): f(x, y) = f(y, x)$ since the correlation should be the same whether pattern $a$ is compared to pattern $b$ or vice versa. Different functions that fulfills these requirements was tested to approximate the increased threshold and the function that was chosen to be fitted to the data was:

$$f(x, y) = 1 + a\left(\sqrt{x} + \sqrt{y}\right) + b\sqrt{xy} + c\sqrt{x + y}$$

The constants $a$, $b$ and $c$ was found using curve fitting in Wolfram Mathematica, and the final function, that adjusts the correlation values, was found to be:

$$f(x,y) = 1 - 1.41429\left(\sqrt{x} + \sqrt{y}\right) + 0.802655\sqrt{xy} + 2.00206\sqrt{x+y}$$

As seen in Figure 4.5 it fits the data well.

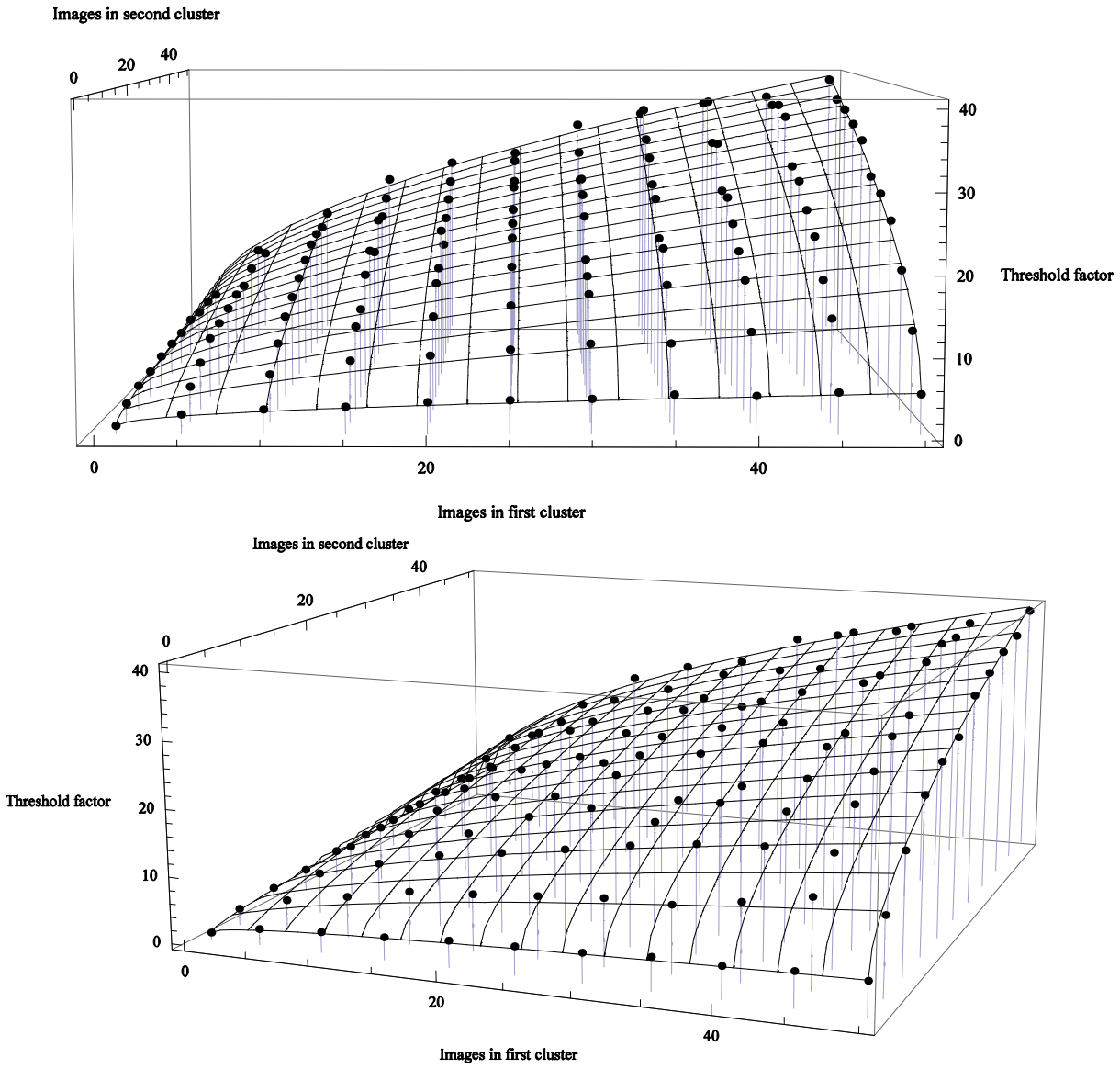


Figure 4.5Fitted function to approximate optimal threshold for given size of clusters, two different angles

Since the optimal threshold is much higher when the number of images in the clusters grow, it is clear that none of the chain clustering or the simple clustering procedures, which both rely on merging residual noise patterns into reference patterns, would work without a dynamically set threshold.

### 4.2.2 Clustering by Feature Vectors

If a classifier is trained to recognize images taken by cameras of the same model one can do an initial clustering and divide the big set of images into smaller sets of images taken by the same camera model. The algorithm described in section 3.5 can then be used to cluster these smaller sets of images.

#### 4.2.2.1 Selected Features

A first set of 19 features was selected by the CFS selection algorithm from the features described in Appendix A. The selected features were:

- The skewness of the noise, that is the third centralized statistical moment, in the green color channel
- The euclidean distance between the channel pair energy vectors, the linear-pattern cross-correlation of the noise in both the rows and columns of all color channels (6 features)
- The correlation between the noise of two color channels, all three possible pairs (3 features)
- The correlation between the noise in the red color channel and the same noise shifted three steps both horizontally and vertically
- The correlation between the noise in the green color channel and the same noise shifted one step horizontally
- If the images are thought to have the same configuration of the green part of their CFA as well as the majority with which this is determined (2 features).
- The average of the diagonal subband of the first level of the wavelet decomposition of the noise in the three different color channels (3 features).
- The average of the horizontal subband of the first level of the wavelet decomposition of the noise in the blue color channel.

Eleven more features were selected one after another by the best first selection algorithm decreasing the error percentage as seen in Figure 4.6.

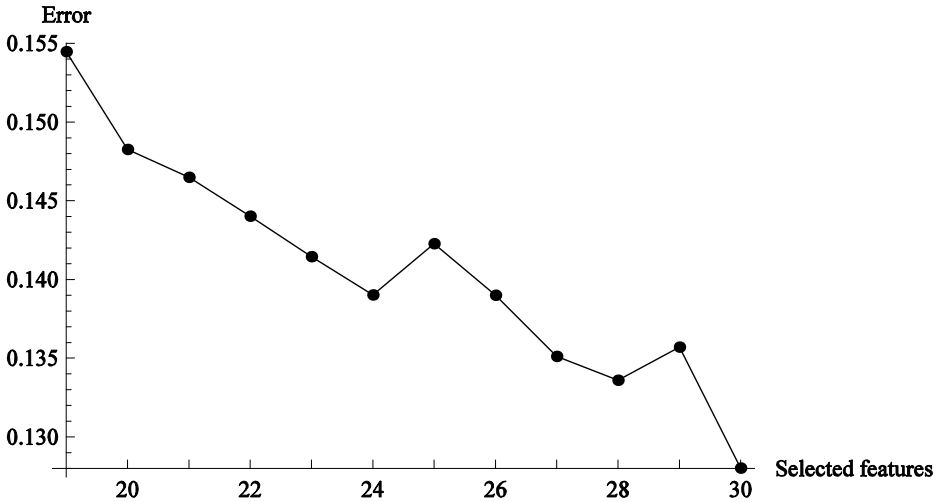


Figure 4.6 Wrapper selection progress

The features added after the 19 features selected by the CFS algorithm were, in the following order:

20. The correlation between the noise in the red color channel and the same noise shifted three steps vertically.
21. The average of the horizontal subband of the second level of the wavelet decomposition of the noise in the green color channel.
22. The average of the diagonal subband the second level of the wavelet decomposition of the noise in the blue color channel.
23. The majority of the red channel CFA-pattern vote
24. The IQM-feature structural content of the red color channel.
25. The correlation between the noise in the red color channel and the same noise shifted one step horizontally and three steps vertically.
26. The average of the vertical subband of the second level of the wavelet decomposition of the noise in the blue color channel.
27. The average of the diagonal subband of the third level of the wavelet decomposition of the noise in the green color channel.
28. The correlation between the noise in the red channel and the same noise shifted eight steps horizontally.
29. The mean square error of the noise in the blue color channel
30. Variance of the noise in the green color channel.

It was decided that the increased performance of feature 23, the CFA majority vote of the red color channel, was not enough to justify the time it took to calculate that feature and thus it was discarded. In total 29 features was selected to be used with the ANN. This gave an error rate of less than 0.13 on the validation set.

#### 4.2.2.2 Time Comparison Between Noise- and Feature Based Classification

The time taken to compare two descriptors with an ANN is approximately 1.32ms compared to the time taken to correlate two noise patterns, 446ms. However the descriptor takes more time to compute, approximately 4990ms compared to 1482ms for the noise. These measurements were made from an average of 100 computations on an Intel Core Duo T2050, no parallelization used. By using feature-based image classification one can thus perform many more comparisons between images. There is no guarantee that the model clustering results in smaller clusters, the images could all be from cameras of the same model. As can be seen in Figure 4.7 already when the image set consists of 17 images, even if the model clustering puts all images in the same cluster, performing the initial clustering would only double the needed time to compute all noise patterns and correlate each RNP with each other RNP. As more images are added only a fraction of the total time would be devoted to model clustering. How long it takes to combine the different model clusters is however very dependent on each specific instance of the problem.

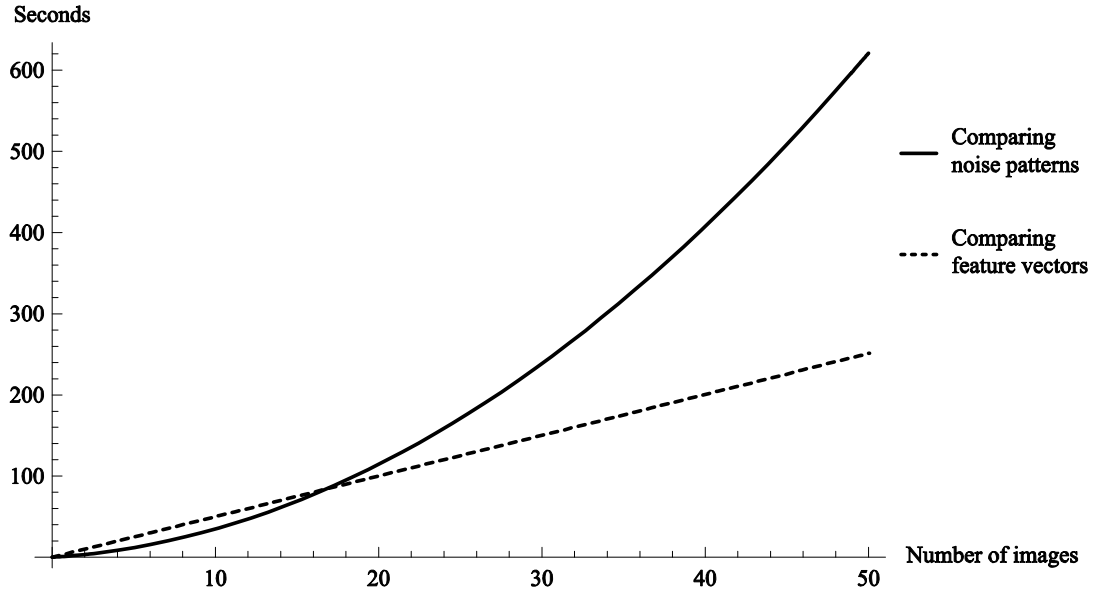

Figure 4.7 Comparison of worst case time consumption of different comparison methods for clustering, in the time estimation both computation of the noise pattern/feature vector and comparison between them are included.

#### 4.2.2.3 Performance of Different Clustering Methods

Performance testing of the clustering algorithms was carried out as described in section 3.5.5. All calculations were done on an Intel Core i7-2630QM processor and no parallelization was used when running the algorithms. Thus the presented time consumption results should only be used for comparison between the different approaches, since the time could be cut considerably if the algorithms were slightly modified to incorporate parallelization.

Approach 2, 3 and 4, the ones using chain clustering and simple clustering, uses a correlation threshold. In section 4.2.1 the threshold that separated positive from negative instances best, i.e. the one with the highest F-measure, were found to be 0.002551. When clustering however, it is important not to group images if they do not belong together, since that could possibly corrupt the reference pattern. Thus the threshold was increased to decrease the risk of wrong classifications. A value of the threshold that was found to give good results with this aspect taken into consideration was 0.006667, and this threshold is used when clustering based on correlation.

The results are presented in Table 4.2. Clustering time indicates the time the algorithms use and total time includes the time it takes to denoise the images, as well as calculating feature vectors in approach 4. Approach 1, only applying the Caldelli algorithm, resulted in many clusters only containing one image and thus the “clusters per image”-value was very high. It did however not group any images that were not from the same camera. Approach 2 increases the time consumption, which can be considered as an argument against chain and simple clustering. The idea behind the chain clustering however is that the images might already be ordered by name or in a file system

hierarchy and chain clustering would take advantage of that. Since the order of the images is randomized in approach 2 the gain of the chain clustering will be limited.
Approach 2 also introduces a small error. 1.8% of the images are put in clusters where the majority of the images are from another camera. On the positive side the number of clusters per camera is much smaller than for approach 1. Both approach 3 and 4 lowers the time consumption considerably. Approach 4, when the images was clustered on model first, had a lower error rate than approach 3 as well as having the lowest "clusters per camera"-value. While tests on other datasets can be carried out to see how the performance of the different approaches varies with different data it is still clear from these tests that clustering on models provides a significant speed increase compared to not dividing the images into sets at al.

| | **1: Caldelli** | **2: Chain, Caldelli, Simple** | **3: Divide and Conquer** | **4: Model Divide and Conquer** |
|---|---|---|---|---|
| Error per image | 0 | 0.01871 | 0.03367 | 0.03197 |
| Clusters per camera | 33.75 | 6.236 | 4.75 | 4.083 |
| Clustering time | 5:34:35 | 8:01:43 | 1:39:29 | 1:01:09 |
| Total time | 5:44:01 | 8:11:09 | 1:48:55 | 1:37:00 |

Table 4.2 Comparison of different clustering approaches, time on h:mm:ss format

The Caldelli clustering algorithm picks the next pair of images to group by finding the pair with the highest correlation between them. Therefore it is natural that approach 3 and 4 gets a higher error rate than the other two approaches. Dividing the sets of images means that the Caldelli algorithm has fewer comparisons to look at and there is thus a higher risk that the max value is a false positive. Since the number of true positives hopefully is high when the image set has been divided based on the model, as in approach 4, it is also natural that the probability of the max value being a true positive is higher than in approach 3. It seems reasonable that a higher threshold could lower the error rate when smaller sets of images are considered. How this affects the number of clusters per camera has however not been evaluated.

For comparison, Approach 3 was also tested with all files already ordered on what camera that took the images, to see how chain clustering can improve the results if the circumstances are the best possible. The error rate dropped to 0.0102 and the number of cluster per camera were 5.25. Those results were achieved in 45 minutes total time and thus it is clear that chain clustering can provide a significant speed up in these circumstances. Also situations with few cameras and seemingly random order will benefit from this, since more images will be adjacent than in this example with 12 different cameras.

# 5 CONCLUSIONS

The focus of the thesis has been to evaluate the existing research in an environment where the number of images that should be analyzed is very high. Furthermore new methods that could make the clustering procedure more efficient in an applied environment has been introduced and tested.

In the research field of source camera identification Mihçak's denoising filter has been the standard filter to compare new denoising filters against since the first papers on the subject used that filter. This has however led to new denoising methods rarely being compared to each other, and this is something that has been done in this thesis. The outcome was that color decoupling improved the Mihçak filter to the extent that the color decoupled version should be considered standard rather than the original version.

Another aspect of the denoising filter that has been standard since the first papers on the subject of source camera identification was assuming the noise had a standard deviation of five. When testing what assumption of standard deviation that gave the best classification results in this thesis however, the choice of five had no justification. Instead a standard deviation of two was favored.

It was concluded that when one correlates two reference patterns, the correlation will depend on how many images that was used to create the patterns. This is an aspect of the correlation classification approach that will need to be considered if one wants to take advantage of the reduced amount of random shot noise in reference patterns.

A number of different clustering approaches were tested and the ones that divided the images into smaller sets that were clustered individually, and then combined, were much faster than the methods that tried to cluster all images at once. By dividing the images on model instead of dividing the images randomly in equally sized subsets, the error rate could be lowered a bit while the number of clusters per camera also was lower.

A number of different clustering approaches were tested and the ones that divided the images into smaller sets that were clustered individually, and then combined, were much faster than the methods that tried to cluster all images at once. By dividing the images on model instead of dividing the images randomly in equally sized subsets, the error rate could be lowered a bit while the number of clusters per camera also was lower. Furthermore an initial chain clustering can speed up the clustering procedure if the images are ordered by camera, as can be the case if the files have not been renamed or if images from the same set lie in their own folders.

This is still a relatively new research field and there are still many aspects to improve and explore. New denoising algorithms developed specifically with source camera classification in mind, such as color decoupled denoising, improve the methods. The existing research is often evaluated on toy examples and to apply the methods to real world examples more work can still be done.

# 6 FUTURE WORK

During the course of the thesis potential improvements and extensions of the work has been identified. Some of the ideas could be the basis of another master's thesis while all topics could be areas of interesting research.

## 6.1 JPEG QUANTIZATION TABLES

In JPEG images, the compression algorithm uses a quantization table to set the level of compression in the image. Many camera manufacturers have their own quantization tables [22], making this a good candidate feature to use for initial clustering. Since it is defined in the JPEG-file itself, it is fast and easy to obtain.

Often the quantization tables are multiples of a base table, where the scaling factor determines the magnitude of the jpeg compression [23]. If one were to compare two quantization tables using correlation, the scaling would not affect the calculation and thus a correlation close to one would indicate that the cameras that produced the images used the same base table, and hence they might also be of the same model. By including the correlation between quantization tables one could hopefully increase the performance of a classifier designed to find images of the same model. Since it however is a characteristic of the file format and not the image data itself, this was not considered in this thesis.

## 6.2 VIDEO CAMERA IDENTIFICATION

In this thesis only still image photography cameras have been considered. It is tempting to say that the solution would work as well on the identification of video cameras from the frames of a video sequence, or perhaps even better since a single second of video could give 30 images and thus a good device reference pattern, but there exists difficulties not present during normal photography.

One additional difficulty is block-like artifacts that are often present across several frames due to compression [24]. Another feature that is more common when dealing with video cameras than ordinary cameras is digital image stabilization (DIS). A video camera that utilizes DIS has a larger sensor than output resolution and can thus shift the captured frames to give a less shaky final output [25]. This means that averaging the residual noise pattern over several frames might suppress the SPN instead of enhancing it. Thus it is evident that in order to adapt the solution to video, further studies and subsequent modifications need to be carried out.

## 6.3 MERGING FEATURE VECTORS

When one uses residual noise patterns for classification one gets a better performance if the noise patterns have been derived from a set of images and averaged into a reference noise pattern. Since the calculation of features also is affected by random shot noise and

artifacts from the scene of the image, the model clustering performance could possibly also be increased if feature vectors could be merged into a reference version. It is however not as simple as with the residual noise patterns since the ANN is a non-linear classifier working on a high dimensional space. Just because two feature vectors have been classified as being from the same model does not necessarily mean that they are close to each other in the high dimensional space. It could thus be of interest to further study if there is a way to merge feature vectors that decreases the error rate of the ANN.

## 6.4 Color Decoupled Denoising

Since color decoupling the image before using the Mihçak denoising filter increased classification performance significantly as presented in Figure 4.1, it would be interesting to see if also other denoising filters could benefit from color decoupling. The PCAI filter is much faster than the other filters and if color decoupling could improve the PCAI approach as much as it improved the Mihçak filter, PCAI could possibly be a better filter to use than the decoupled version of the Mihçak filter.

## 6.5 Non-Binary Classification

One problem is that while comparing images by correlation of the RNPs, the instances of true and false matches overlap a bit in the correlation values. This is why a 100% rate of classification cannot be reached by only considering the noise of two images. Instead of using hard thresholding, one could assign a probability that a comparison is a match, given the correlation and the number of images that was used in each of the RNPs. This can for example be accomplished by modeling the probability density of true and false matches as functions, and for each comparison compute what the probability density is for true and false at that point. In this way one could distinguish the predicted matches that are that are very likely to be true from those that are just above 50% probability to be true.

# 7 REFERENCES


[1] BBC, ”Lincolnshire Police seizes millions of indecent child images,” 2012. Available at: http://www.bbc.co.uk/news/uk-england-lincolnshire-17023530, Accessed 27 March 2012.

[2] SpyBlog, “Operation Algebra child rape convictions in Scotland: open WiFi tracking, digital camera image forensics,” 2009. Available at: http://spyblog.org.uk/blog/2009/05/09/operation-algebra-child-rape-convictions-in-scotland-open-wifi-tracking-digi.html, Accessed 10 May 2012.

[3] J. Hofmann, “Video & Image forensics,” Digital Forensics Magazine, vol. 8, pp. 60-64, August 2011.

[4] J. Lukas, J. Fridrich, and M. Goljan, “Digital camera identification from sensor pattern noise,” IEEE Transactions on Information Security and Forensics, vol. 1, no. 2, pp. 205–214, June 2006.

[5] Gerald C. Holst. “CCD Arrays cameras and displays, Second edition,” Winter Park, FL: JCD Publishing; 1998.

[6] M. Chen, J. Fridrich, and M. Goljan, "Digital imaging sensor identification (further study)," Proceedings of SPIE Electronic Imaging, vol. 6505, January 2007.

[7] V. Conotter, and G. Boato: "Analysis of sensor fingerprint for source camera identification," Electronic Letters, vol. 47, no. 25, pp. 1366-1367, December 2011.

[8] Y. Li, and C.-T. Li, “Digital camera identification using colour-decoupled photo response non-uniformity noise pattern,” in Proceedings of IEEE International Symposium on Circuits and Systems, pp. 3052-3055, June 2010.

[9] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," International Conference on Image Processing, vol. 1, pp. 709-712, October 2004.

[10] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," In Proceedings of International Conference on Image Processing, pp. 1296–1299, October 2008.

[11] M. K. Mihçak , I. Kozintsev, and K. Ramchandran, "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 6, pp. 3253-3256, March 1999.

[12] L. Alparone, F. Argenti, and G. Torricelli, "MMSE filtering of generalised signal-dependent noise in spatial and shift-invariant wavelet domain," Journal of Signal Processing; vol. 86, no. 8, pp. 2056–2066, August 2006.

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," IEEE Transactions on Image Processing; vol. 16, no. 8, pp. 2080–2095, August 2007.

[14] G. Wu, X. Kang, and K. J. R. Liu, "A context adaptive predictor of sensor pattern noise for camera source identification," IEEE International Conference on on Image Processing, October 2012.

[15] C. Popescu, and H. Farid, “Exposing digital forgeries in color filter array interpolated images,” IEEE Transactions on Signal Processing, vol. 53, no. 10, pp. 3948–3959, October 2005.

[16] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast image clustering of unknown source images," IEEE International Workshop on Information Forensics and Security, pp. 1-5, December 2010.

[17] M. Kirchner, "Efficient Estimation of CFA Pattern Configuration in Digital Camera Images," Proceedings of SPIE Electronic Imaging, vol. 7541, January 2010.
[18] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," Hamilton: Department of Computer Science, University of Waikato; 1999
[19] S. Haykin, "Neural Networks and Learning Machines, Third Edition," Upper Saddle River, N.J.: Pearson Education, 2009.
[20] C.-T. Li, "Source Camera Identification Using Enhanced Sensor Pattern Noise," IEEE Transactions on Information Forensics and Security, vol. 5, pp. 280-287, June 2010.
[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explorations, vol. 11, no. 1, 2009.
[22] H. Farid, "Digital Image Ballistics from JPEG Quantization," Hanover: Department of Computer Science, Dartmouth College, 2006.
[23] J. D. Kornblum, "Using JPEG quantization tables to identify imagery processed by software," The Journal of Digital Investigation, vol. 5, pp. 21-25, September 2008.
[24] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Source Digital Camcorder Identification Using Sensor Photo Response Non-Uniformity," Proceedings of SPIE Electronic Imaging, vol. 6505, January 2007.
[25] K. Kurosawa, K. Kuroki, and N. Saitoh, "An approach to individual video camera identification," Journal of Forensic Sciences, vol. 47, pp. 97-102, 2002.
[26] International Telecommunication Union, "T.81: Information technology - Digital compression and coding of continous-tone still images - Requirements and guidelines", 1992. Available at: http://www.w3.org/Graphics/JPEG/itu-t81.pdf Accessed 14 May 2012.
[27] N. Khanna, A. K. Mikkilineni, G. T. C Chi, J. P. Allebach, E. J Delp, "Scanner identification Using Sensor Pattern Noise," In proceedings of SPIE, Electronic Imaging, Security, Steganography and Watermarking of Multimetia Contents IX, San Jose, Ca, vol. 6505, pp. 1K - 1, 2007.
[28] E. Bayer, "Color Imaging Array," US Patent, 3 971 065, 1976.
[29] M. Kirchner and R. Böhme: "Synthesis of Color Filter Array Pattern in Digital Images", Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 7254, 72540K, 2009.
[30] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," IEEE Transactions on Image Processing, vol.12, no.2, pp. 221- 229, February 2003.
[31] Y. Hu, C.-T. Li, and C. Zhou, "Selecting forensic features for robust source camera identification," International Computer Symposium 2010, pp. 506-511, December 2010.
[32] C.-T Li, "Source camera identification using enhanced sensor pattern noise," IEEE Transactions on Information Forensics and Security, vol. 5, no. 2, pp. 280–287, June 2010.
[33] H. Farid, "Detecting hidden messages using higher-order statistical models," Proceedings of International conference on Image processing, vol. 2, pp. 905-908, 2002.
[34] S. Bayram, H. T. Sencar, and N. Memon, "Classification of digital camera-models based on demosaicing artifacts," The Journal of Digital Investigation, vol. 5, no. 1-2, pp. 49-59, September 2008.

# APPENDIX A – CANDIDATE FEATURES FOR CAMERA MODEL CLASSIFICATION

Many different features have been used by others to give an indication on what camera that took an image. In the setting of this thesis the features should indicate what camera model, rather than unique device, that captured the image, or more specifically if two images, or clusters of images, were captured by the same model. Thus features defined by others were tested in this new setting and the features implemented and tested are described in detail in this appendix.

## A.1 NOISE PATTERN STATISTICS AND CHARACTERISTICS

After the introduction of SPN for camera source identification Filler et al [10] introduced a number of features calculated from the noise pattern to identify what camera model that captured the original photo. The study by Filler et al suggested that the first three centralized statistical moments of the noise could be used, and since no argument was made by Filler et al as to why only the three first moments was used, it was decided that also the fourth centralized statistical moment where to be tested as a feature.

Furthermore a number of correlation based features was introduced by Filler et al.; for each color channel pair out of the red, green and blue channel, and for each horizontal and/or vertical pixel shift in the interval of zero to three pixels, the correlation between the resulting shifted channel pairs made up 96 extracted values. By using principle component analysis (PCA) Filler et al reduced the 96 values to four features. This approach means that to extract one of these four features from an image; all 96 original correlations must be calculated. Since the reason that features are considered in this thesis is to speed up the clustering process, and the correlation is a costly operation, calculating 96 correlations to extract four features is not a good approach in the setting of the thesis, if only a part of the 96 features are useful or if some of them are redundant. Therefore PCA was not applied to the original correlation values and not all 96 combinations of shifts and channels were used. More specifically the only correlation between different color channels was without any positional shift, resulting in 3 values. Furthermore each color channel was correlated with the shifted versions of itself, adding another 3x4x4 = 48 features.

A third group of features, known as linear-pattern cross-correlations was introduced by Filler et al. The feature is calculated by averaging row and column noise so that one value for each row and channel and one value for each column and channel is obtained. Since the noise in each pixel ideally is a random variable with the zero mean, how much the rows and columns diverge from this can give information on the post processing done by the camera model. One thus gets one vector with the row means and one vector with the column means. By shifting the vectors a number of steps, and correlating the

vectors with the shifted versions of itself, one gets new vectors with one value for each step the vector was shifted, which can then be compared to the respective vectors from other images.

In previous studies the energy ratio between color channels has been used to find the source camera of images [9], the mathematical definition of the energy between channel$A$ and $B$is:

$$\frac{\sum_{x=0}^{w}\sum_{y=0}^{h}A(x,y)^2}{\sum_{x=0}^{w}\sum_{y=0}^{h}B(x,y)^2}$$

Where $A(x,y)$ is the value of pixel $(x,y)$ in channel $A$.

In addition to the features presented, a number of new features, derived from the noise are presented in this thesis. Since Jpeg-compression divides the image into blocks that are eight pixels high and wide during processing [26], a new feature introduced in this thesis was the correlation between each color channel and an eight-step shifted version, horizontally and/or vertically, of the respective channel.

Scanners uses a sensor array, instead of matrix, and therefore when the task is to identify what scanner that captured a digital image, the noise of the different rows in the image is averaged, since all pixels in a row was captured with the same sensor [27]. This approach inspired a number of new features derived from the described row average vector. From this vector the first four central statistical moments was calculated. Four additional features were acquired by repeating the same procedure but considering columns instead of rows.

## A.2 Color Filter Array

Nearly all digital cameras today use a color filter array (CFA) in order to make the camera able to capture colors in the visible spectra. The most common filter is the Bayer filter [15], [28], in which there is a repeating $2 \times 2$ pattern with two green cells and one each of red and blue. Among the four variations of Bayer patterns, all are used in digital cameras. Efforts have been made to be able to tell which pattern was used by the camera that took a specific image [29].

Using the method described in [17], one can compute a prediction for the green channel, and subsequently predictions for the red and blue channel. These three predictions can then be used as features. The algorithm divides the image into blocks and lets each block cast one vote on which of the types of Bayer patterns that is the most likely. The algorithm concludes that the configuration that is the most likely one for the most blocks is the true configuration. In addition to the prediction, the certainty of the

prediction in each channel was considered as features. By letting $v_1$ and $v_2$ be the number of votes for filter one and two respectively, this certainty, called majority overweight, was calculated by the following formula:

$$Majority = \frac{|v_1 - v_2|}{v_1 + v_2}$$

## A.3 Image Quality Metrics

To assess the quality of an image generated by a camera, Image Quality Metrics are a set of mostly statistical image features that for instance has been used to detect if images contain embedded hidden messages [30]. The idea then was that the quality of an image would worsen when a message was embedded, but Kharazzi et al. used them to classify image sources on the premise that different camera models and possibly even different camera devices of the same model produce images of different quality [9]. The work done in [30] concluded that some IQM features had a greater impact than others and therefore only those features are presented here.

### A.3.1 Image Measurements

Mean square error, where error in this case is the noise value in each pixel, is defined as an IQM feature in [30]; one value for each channel is calculated on the premise that a camera device might be more prone to noise in one channel than another.

Furthermore a number of features related to correlation can be calculated by considering both the original image and the denoised image, namely Czekanowski distance, angular correlation, normalized cross correlation, image fidelity and structural content [30]. All but structural content was concluded by Avcibas et al as good characteristics for steganalysis, but it was picked up as a feature again in [31] and thus for completeness it was also included in this study.

### A.3.2 Spectral Features

By transforming the image channels to the Fourier Domain we get coefficients representing different frequencies and these coefficients can be analyzed. A number of spectral features were proposed in [30]. The simplest of the features is the mean square difference of the magnitude of the spectral coefficients. The other proposed spectral features are instead taken by dividing the image into square blocks and performing the Fourier transform on these. The extracted feature is then the median value of the blocks. This seems like a good approach when identifying camera sources since there might be more extracted noise in some areas of the image because of edges in the original and thus this noise is not attributable to the sensor, or there might be less extracted noise if the pixels in the original image are fully saturated [32].

The median features are derived by calculating the root mean squared difference in magnitude and the root mean squared difference in phase. This gives two new features and a third feature is given by, for each block, combining the magnitude and phase features as defined below:

$$J = \lambda J_M + (1 - \lambda) J_\varphi$$

Where $J_M$ is the magnitude error and $J_\varphi$ is the phase error. $\lambda$ is set so that the magnitude and phase error contributes in equal part. Once again the used feature is the median of the block values.

### A.3.3 Human Visual System

By applying a band pass filter in the discrete cosine transform domain one gets an image on which image quality measurements has produced better results when compared to how humans perceive the quality and thus a feature applied to the filtered image might produce better results, since the noise of a camera affects the visual appearance. The feature in the human visual system domain (HVS), defined in [30], called normalized mean square HVS error is calculated between the filtered original and denoised images.

### A.3.4 Wavelet Domain Features

By representing an image in the wavelet domain one can perform analysis on the high frequency components of an image, and since noise differs from pixel to pixel it is clearly a detail that will be represented in the high frequency part of a wavelet decomposition of an image. The wavelet decomposition extracts the high frequency parts into horizontal, vertical and diagonal subbands. By further breaking down the low frequency part of the wavelet decomposition, one gets new high frequency subbands.

The decomposition considers each color channel separately. Simple features such as the first four statistical moments in the different subbands and scales were used by Farid et al. [33]. As when extracting the noise component with Mihçak's filter [11] an 8-tap Daubechies transform was used to decompose the image in four levels.

Additionally [33] suggested defining linear predictors that given neighboring wavelet coefficients predicts the coefficient at position $(x, y)$ in different levels. By calculating the weights that solves the linear prediction optimally the proposed features are to calculate the four statistical moments of the error in each coefficient, and thus one receives again four features for three subbands and three levels of decomposition.

## A.4 Intensity neighborhood distribution center of mass

Intensity neighborhood distribution center of mass is a value that is used to indicate if the sensor is more prone to intercept low or high values in each of the color channels, that is if it is more susceptible to some intensity levels than others. In similar images the distribution is similar as well but shifted in one direction when comparing different cameras and by calculating the center of mass one can find this shift [9]. The first part of the calculation was counting the number of pixels that had each intensity level, thus getting one value for each intensity level. The neighborhood distribution was then retrieved by assigning each intensity level a new value, the sum of the two previously counted number of pixels that had either of the neighboring intensity values. The extracted value was then the middle point where the accumulated number of neighbor intensity values was the same above and below.

A neighborhood distribution center of mass of the noise values was in this study introduced as a complement to the mean noise value, by mapping the noise to the same discrete interval as the image intensity values, that is $[0,255]$, where the minimum noise value in the image, that is the negative noise value of highest amplitude, was assigned the lowest intensity value and the maximum noise value was assigned the highest intensity value. After this scaling and discretization the noise neighborhood center of mass was calculated as for the image and the extracted value is intended to show if the camera is more susceptible to positive or negative noise.

## A.5 Demosaicing Artifacts

In the image, one can calculate the second derivative in each row of the image using the following formula:

$$s_y = p(x+1, y) + p(x-1, y) - 2p(x, y)$$

Where $p(x, y)$ is the value of the pixel at position $(x, y)$. Next, one computes the mean of the rows to form a one-dimensional signal according to:

$$v = \sum_{y=0}^{h} |s_y|$$

Finally, by computing the discrete Fourier transform of this signal, one can see periodicity in the resulting signal. This resulting signal can be computed similarly for the columns in the image. Using both columns and rows one obtains 6 features, two for each color channel. For each of these resulting signals, we located the 32 highest and lowest values, giving 6 more features [34].