Camera Identification on YouTube

Yannick Scheelen & Jop van der Lelie
Universiteit van Amsterdam
System and Network Engineering (MSc)
(Dated: February 12, 2012)

Identifying the camera with which a video has been taken can under the right circumstances be done using the Photo Response Non-Uniformity (PRNU), which acts as a unique fingerprint for cameras. In this paper we use the wavelet filter from Lukáš et al [1] to extract the PRNU pattern from a video file that is re-encoded using the Advanced Video Codec. Different video resolutions and encoding settings were used to investigate the influence of this re-encode on the PRNU pattern. It is shown that even after a video is re-encoded it is still possible to link some videos to their original camera.

I. INTRODUCTION

Identifying the camera that was used to capture a video can be very interesting from a forensics point of view. For example, when a camera is seized in a child-abuse case, a forensic expert wants to see if this camera can be linked to a photo or video from a known database of incriminating video material.

With the increasing popularity of video sharing due to the growing amount of online video upload services, like YouTube or Vimeo, finding the true origin is challenging. Especially because, for the video services to keep up with this enormous amount of data, each video is re-encoded (compressed) after it is uploaded. This compression is needed to reduce the bandwidth usage by reducing the file size and thus increasing streaming speeds. By re-encoding the video, the video is altered which makes it harder to identify the source.

Although most cameras store metadata like the camera's serial number and some newer ones even GPS coordinates within the file, this information can easily be removed. However, another way to identify the origin remains. Another 'tag' or imprint the camera leaves in the output, which is not easy to recognize, is 'noise'. When a camera processes the signal, there are multiple factors that can cause noise. For instance Fixed Pattern Noise (FPN). This noise may be caused by dead pixels, if the camera has any, but a better source is the Photo Response Non-Uniformity (PRNU). The PRNU is introduced by the image sensor [2, 3] and is always present in the output. This PRNU is proven to be unique [4] per sensor which makes it ideal for source identification.

Because this PRNU pattern is in the images itself, the quality of this pattern decreases when more compression is applied. At the time of writing the most commonly used video codec is the Advanced Video Codec (H.264/MPEG-4) whose influence on the PRNU has not yet been researched. The latest research in this field was in 2008 by van Houten et al [5], who looked at low resolution videos compressed with the XviD and Windows Media 9 video codec. We will extend this research and investigate if the PRNU can be used to identify the original camera after the video has been re-encoded using the Advanced Video Codec (H.264/MPEG-4).

II. THEORY

This theory section focusses on camera identification on still images and not videos, as videos can be seen as a sequence of still images.

A. Photo Response Non-Uniformity

When a forensic expert wants to identify the original camera of an image he will look for imprints introduced by the camera and compare these to a set of images of unknown origin. These imprints come from various sources where the most important one is the image sensor. Other imprints can come from color interpolation (interpolation artefacts) [6]) or signal processing (quantization tables)[7]).

For camera identification from images it is important to have an imprint that is present in multiple images, the same pattern has to be present in both the reference material as well as the unknown material. This returning pattern is known as fixed pattern noise (FPN) and is caused by the CCD (charged coupled device) or CMOS (complementary metal oxide semiconductor) sensor that processes the input signal (light) and converts it to a digital signal. In [8] by Geradts et al, it is shown that defect pixels in the sensor can be used to identify the original camera. The FPN can be retrieved by filtering out other noise sources by averaging multiple images as these other noises are not present in all images. However the pattern noise from defective pixels is not always present in the output as many cameras filter this out post-processing.

In [1] by Lukáš et al, a better method for camera identification based on pattern noise is shown. The origin of the source is not from defective pixels but comes from the fact that the light sensitivity of each pixel is non-uniform. When all pixels are exposed to the same amount of light, each pixel registers a slightly different response. This is called Photo Response Non-Uniformity (PRNU) and is also considered pattern noise. This pattern noise is always present due to device and construction imperfections and is proven to be unique per sensor [4]. Once extracted from the video, it can act as a fingerprint for a particular camera and can be used to verify the existence of that pattern in other images. First, the reference pattern noise has to be obtained from the camera which can then be used to verify its presence in images from unknown origin.

To verify if video $\boldsymbol{V}$ was taken with camera C, its reference pattern $\mathbf{P}_C$ has to be obtained. The presence of $\mathbf{P}_C$ in $\boldsymbol{V}$ will be shown with their correlation value. Because most cameras do not allow access to the raw sensor data, only an approximation of the pattern noise can be obtained by averaging multiple frames from the video.

The video $\boldsymbol{V}$ is first cut into individual frames $\boldsymbol{I}_i$ where $i = 1, ..., N$ with N the total amount of frames in $\boldsymbol{V}$. An approximation of the actual pattern noise is retrieved by averaging $\boldsymbol{I}_i$. Ideally, each frame should only contain pattern noise and no scene content during this process. Therefore a denoising filter F is used to filter out scene content per frame $\boldsymbol{I}_i$ leaving only the noise residue $\boldsymbol{n}_i$ that will be used to obtain the pattern noise.

$$\boldsymbol{n}_i = \boldsymbol{I}_i - F(\boldsymbol{I}_i) \tag{1}$$

The higher the amount of frames I_i in $\boldsymbol{V}$, the better the approximation pattern noise $\boldsymbol{P}_V$ for the video will be. An $N > 50$ is recommended [1].

$$\boldsymbol{P}_V = \overline{\boldsymbol{n}} \tag{2}$$

Now to verify if video $\boldsymbol{V}$ was taken with camera C the correlation ρ_C between the obtained pattern $\boldsymbol{P}_V$ and reference pattern $\boldsymbol{P}_C$ is calculated:

$$\rho_C(\boldsymbol{V}) = corr(\boldsymbol{P}_V, \boldsymbol{P}_C) = \frac{(\boldsymbol{P}_V - \overline{\boldsymbol{P}}_V) \cdot (\boldsymbol{P}_C - \overline{\boldsymbol{P}}_C)}{\|\boldsymbol{P}_V - \overline{\boldsymbol{P}}_V\| \|\boldsymbol{P}_C - \overline{\boldsymbol{P}}_C\|} \tag{3}$$

where the bar above a symbol denotes the mean value. Depending on the denoising filter used, better correlations can be found. In [1] by Lukáš et al, a wavelet based filter is presented which performed better than, for example, the Wiener or median filter. The latter two usually misinterpret areas around edges. For a full description on how this denoising filter works we refer to the following papers: [1], [8].

The entire algorithm is implemented in the open-source tool PRNUCompare [9] created by van Houten et al. This is used to obtain the PRNU patterns and calculating the correlations between videos.

Using $\rho_C(\boldsymbol{V})$ it is now possible to see if camera C has been used to create video $\boldsymbol{V}$. Obtaining the reference pattern for camera C is done by creating a reference video. This is a video that has no scene content and ideally close to uniform illumination. Using the tool PRNUCompare, the reference pattern noise for camera C can be extracted. The suggested setting of the σ in the wavelet filter is 5 [5].

Because the wavelet filter processes each color channel of the image separately, $\rho_C(\boldsymbol{V})$ will result in 3 (RGB) correlation values. The sum of these values will be used to link the video to a camera.

B. Advanced Video Codec

A video codec is used to compress a video file. This process is always a trade-off between quality and file size. The better the quality of your video, the larger the file. By compressing the video file, the file size is reduced which cuts down bandwidth usage and increases streaming speed.

Today, the standard video codec to encode high definition videos is the AVC encoder. Which is used by multiple online video services, such as YouTube and Vimeo.

AVC is based on two standards, the H.264 standard and the MPEG-4 AVC standard and is implemented in the library 'libx264' for FFmpeg. Various settings can be used to change the compression applied but the easiest one to use is the Constant Rate Factor setting. This aims at a certain quality for the output video. When a CRF setting of 0 is used this will result in a lossless compression. A compression between 18 and 20 will still have high quality video but is smaller than the original file.

Using this CRF encoding setting the quality level of the output video can be easily influenced.

III. EXPERIMENTAL METHODS

A. Restrictions

The supported video resolutions for cameras get more and more standardized. New video cameras are capable of recording in High Definition (HD), which are either 1280x720 or 1920x1080 pixels. These cameras often have an option to switch to a lower resolution of 640x480 pixels to save disk space. This smaller resolution is still being used in video cameras in mobile phones, webcams or surveillance cameras. In the research of van Houten et al. [5] only low resolution cameras where used: 176x144, 320x240, 352x288 and 640x480. A logical step forward is to look at the new HD resolutions and how they compare to the lower ones. In our research we restricted our video resolution to 640x480 and 1280x720. The full HD resolution of 1920x1080 will not be included as there are not many cameras available that record in this resolution.

Van Houten et al. [5] calculated that, in general, 200 or more flatfield frames should be sufficient for extracting a reliable pattern. To guarantee that our patterns are reliable, we trimmed all the videos to 30 seconds. The number of frames a 30 second video contains will vary slightly depending on the cameras' frame rate, ranging between 750 and 900 frames. In previous research[5], conclusive results were drawn with videos of 30 seconds of length which is why we used the same length.

B. Experimental Set-up

1. Cameras

In our experiment we used cameras that were capable of recording videos in both 480p and 720p.

We tested three popular camera brands; 2 Panasonic cameras, 2 Canon cameras and the Apple iPhone 4. To improve the reliability of our data, five cameras of the same model were tested. This would greatly reduce the chance of possible manufacturing faults and would also make it potentially possible to come to a consensus about the effectiveness of PRNU pattern comparisons for these camera models.

The iPhone 4 was specifically chosen because of its popularity with the general public. It's one of the most sold smartphones and an abundance of videos uploaded on YouTube nowadays comes straight from the iPhone. Investigating this smartphone's camera gave us a perfect chance to link our research to topicality.

Table I provides an overview of our cameras, their supported resolutions and encoding possibilities. The Panasonic FP-7 and and the Panasonic FZ-45 both encode their videos using the Motion JPEG codec. However, the Panasonic FZ-45 is capable of shooting 720p videos using the AVC (H.264/MPEG-4) codec, too. The other cameras all shoot both resolutions using the ubiquitous AVC (H.264/MPEG-4) codec.

The frame rate varies per camera. This specifies how many frames per second the camera can record. Since we trim our videos to 30 seconds, this frame rate value will determine the final frame count of our reference and natural videos.

Our research focused on the PRNU pattern we extracted from videos after they have been re-encoded using the AVC (H.264/MPEG-4) codec, meaning that it's not of great importance with which codec they were pre-encoded. We did however shortly discussed the differences these codecs have on videos after the second layer of compression was applied.

	FP-7	FZ-45	SX210 IS	Ixus 220HS	iPhone 4
640x480[a]	30	30	-	-	-
640x480[b]	-	-	29.97	29.97	-
1280x720[a]	24	30	-	-	-
1280x720[b]	-	25	29.97	29.97	VBR 30
x					

[a]Motion JPEG
[b]AVC

TABLE I: Frame rate per supported video resolution per camera

2. Preparing the videos

To decide whether or not the correlation between PRNU patterns is still a reliable method to find the original camera of videos, we needed to verify that this applies for our cameras. We first labelled each camera per model from 1 through 5. We then extracted the patterns from our reference and natural videos before the videos were re-encoded. Afterwards, the pattern of each reference video was compared to the pattern of the other five natural videos from that same model. This was done on both 480p and 720p resolutions, when possible. For example, we extracted the PRNU pattern from the reference video made with the Panasonic FP-7_1. We did the same for the natural videos made with the Panasonic FP-7_1 through 5. The correlation between the pattern of each reference video and that of the 5 naturals was then calculated by PRNUCompare. The correlation between the matching cameras is indicted by ρ_m and the highest correlation of the first mismatching camera by ρ_{mm}.

For each camera we created a reference video and a natural video for both the 480p and 720p resolutions as described in section II. The reference videos where taken under lab conditions to guarantee that the extracted PRNU pattern was reliable. The reference videos were all shot using the same method; by recording a white surface for 30 seconds while slowly moving the camera.

Every video was then trimmed to 30 seconds using the video editing tool Avidemux [10]. Avidemux allowed us to trim video files without re-encoding the video using the 'copy stream' function. First, the tool demuxes the video stream. Next, it rewrites the headers with new lengths so the actual video stream is not altered. Because the frame rate per camera is different we calculated the amount of frames that would correspond with approximately 30 seconds and cut the video accordingly.

3. Video encoding

With all videos in the proper length we now created a re-encode of each natural video with different quality settings. We used the CRF option in the libx264 codec as described in section II to aim at an approximite quality of the video. Each video was encoded with CRF values ranging from 18 till 39 in incremental steps of 3, which resulted in 8 re-encoded videos. Using the tool PRNUCompare, the PRNU pattern of each video was extracted.

4. YouTube encoding

To compare our re-encoded videos to those of YouTube, we uploaded all natural videos and downloaded them again afterwards. Because YouTube now allows both 640x480 as 1280x720 resolution videos none of the videos were resized by YouTube. Every video was compressed by YouTube using the AVC (H.264/MPEG-4) codec with unknown settings. The PRNU pattern of each video was extracted with PRNUCompare.

5. Evaluation criteria

The PRNU pattern comparison we did correlates the difference in RGB values between two videos. Each color channel can correlate to a maximum of 1 or a minimum of -1. The sum of the three correlations is the final value we will use to conclude whether or not the videos are a match. The sum has a maximum of 3, indicating a 100% match between the 2 videos. This is achieved by comparing two of the same videos.

To confirm the values returned after a PRNU comparison, a Student's T-Test is performed on the results as well. The T-test will increase the significance of the data and, together with the PRNU correlation values we retrieved, will conclude the reliability of the pattern.

IV. RESULTS

A. Pre-encoding

The first set of results was obtained by using PRNUCompare to extract and compare the PRNU patterns from our videos using the extraction method as described in section III.

As table II shows for the Panasonic FP-7, the highest match is found by comparing the PRNU pattern of the Panasonic FP-7_1's reference video, with its natural video. For the videos shot at a resolution of 480p; values $\rho_m = 2{,}021$ and $\rho_{mm} = 0{,}054$ are calculated. Table III shows the results of the same camera but with the videos shot at a resolution of 720p. There is a minor increase in ρ_m yet a much higher mismatch is found compared to the 480p videos. $\rho_m = 2.129$ and $\rho_{mm} = 0.481$.

The same calculations have been made for each camera, resulting in tables II through XI.

B. Second layer of compression

After the PRNU patterns were extracted before the re-encoding, we proceeded with extracting the PRNU patterns of all the encoded videos and compared these values to the reference videos of the same model, the same way we did with our original video files. In tables XII through XXI the specifications per video along with ρ_m and ρ_{mm} are shown. Only results of the first camera per model have been included in the appendix.

As table XII shows for the Panasonic FP-7_1, with a constant rate factor of 18, $\rho_m = 1.456$ (opposed to 2.021 without encoding[IV A]) and $\rho_{mm} = 0{,}034$ (opposed to 0.054 without encoding[IV A]) is achieved. The stream size is 15.2MB and the bitrate is 4063kbps (opposed to 38.4MB and 10.7Mbps for the original video).

As the compression increases with CRF setting 18 through 39, the PRNU correlations at a CRF 39 decrease to $\rho_m = 0.068$ and $\rho_{mm} = 0.023$. The stream size is just 930kB and the bit rate drops to 246kbps.

The values for the YouTube videos are also included in these tables. For the Panasonic FP-7_1, $\rho_m = 0.306$ and $\rho_{mm} = 0.019$. The stream size of the YouTube video is 4.23MB and the bitrate measures 1129kbps.

V. ANALYSIS

A. Pre-encoding

The PRNU pattern correlations for all the video files were calculated. We took ρ_m and ρ_{mm} of the values in tables II through XI and calculated the mean $\overline{x}$ of these values.

Our data set is finite and thus the standard deviation is calculated on the ρ_m and ρ_{mm} PRNU pattern correlations of tables II through XI to identify the spread of the data.

The formula of the standard deviation, where x_i is the set of data and $\overline{x}$ the mean:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}} \tag{4}$$

Figure 1 represents the calculated PRNU values of the videos, before the second layer compression is applied.

To measure the significance of our data, a Student's T-Test is done on the correlation values of the PRNU patterns of our videos on all encoding settings.

In testing the null hypothesis that the population mean is equal to a specified value μ_0, we use the statistic:

$$p = \frac{\overline{x} - \mu_0}{s/\sqrt{n}} \tag{5}$$

where $\overline{x}$ is the sample mean, s the standard deviation of the sample and n the sample size. Applied on our data, this means that if the value p is around a threshold of 0.05 with a deviation of 0.05, the significance of the difference between ρ_m and ρ_{mm} is rejected and thus we can say that the PRNU correlation between the two videos is not reliable.

What graph 1 shows, is that comparing PRNU patterns is an effective method for detecting the origin of a camera for **some** camera models. The Panasonic FP-7, the Panasonic FZ-45 and the iPhone have clear differences between the matching value ρ_m and the highest mismatch, ρ_{mm}. The PRNU values for the Canon PowerShot SX210 IS videos on 720p are also still considered as reliable. Looking at the differences in graph 1, it shows that the differences between ρ_m and ρ_{mm} for the Canon

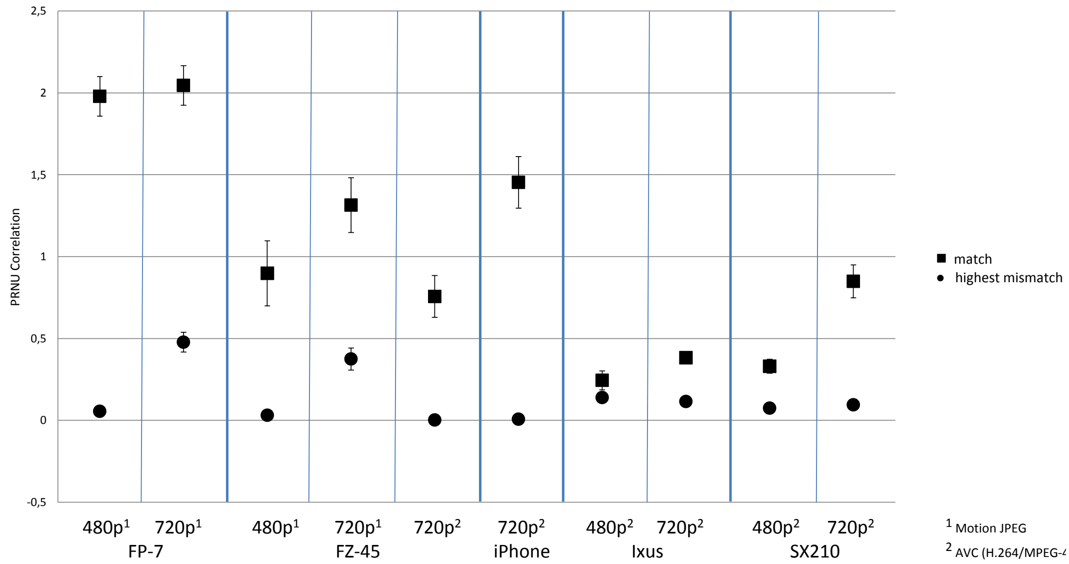

FIG. 1: Averages of the pre-encoded PRNU pattern correlation values of all tested cameras

Ixus 220HS and the Canon PowerShot SX210 IS at 480p are barely visible.

We took a closer look at the different PRNU values of the videos on their respective resolutions, to determine which are accurate enough to be considered reliable.

1. *Panasonic FP-7*

This camera uses Motion JPEG to encode its videos. On 480p, ρ_m equals to 1.979 and ρ_{mm} equals to 0.054. This is a difference of factor 36 and is considered excellent. The 720p videos have $\rho_m = 2.045$ and $\rho_{mm} = 0.477$, which is a difference of factor 4.2, which is considered good enough, especially because of the high correlation of the matching value (knowing that the maximum correlation is 3). Graph 1 clearly depicts this difference in values.

2. *Panasonic FZ-45*

The Panasonic FZ-45 records its 480p videos in Motion JPEG but has the possibility to make 720p videos in either Motion JPEG or AVC (H.264/MPEG-4). This provided us with a nice opportunity to distinguish a preliminary difference in how M-JPEG and AVC affect the PRNU values. As graph 1 shows, there is sufficient difference in the matching and mismatching videos. The height of the correlation of the matching values and the low correlation of the mismatching values prove to be significantly different to correctly determine the camera's origin.

It also shows how, in our case, the AVC (H.264/MPEG-4) codec has a substantially lower mismatching value compared to the 720p video recorded with Motion JPEG (0.006 for AVC opposed to 0.374 for the Motion JPEG). However, it must be noted that ρ_m of the 720p AVC is 0.756, which is almost twice as low as ρ_m of the 720p M-JPEG (1.131).

If the factor ratios are compared further, the 720p AVC trumps the 720p Motion JPEG with a difference of factor 729 compared to factor 3.5 for the Motion JPEG videos. Even though the Motion JPEG has a fairly low factor difference, because the matching PRNU value correlation is so high, it is considered as reliable.

3. *Apple iPhone 4*

The Apple iPhone 4 shows very good results. $\rho_m = 1.45$ and $\rho_{mm} = 0.006$. This is a difference of factor 235. This high factor combined with a high ρ_m value and a very low ρ_{mm} make it an exemplary camera for comparing PRNU patterns.

4. *Canon Ixus 220HS*

The Ixus was the first camera for which the PRNU values did not satisfy. Both on 480p and 720p, the matching values are considered too low to be reliable and as graph 1 illustrates, the correlation of the mismatches lays very close to the matching values. Also, when we take the standard deviation of the 480p video in closer consideration, there is almost no difference visible in the values; $\rho_m = 0.196$ and $\rho_{mm} = 0.169$.

With the resolution at 720p, ρ_m increases a little bit

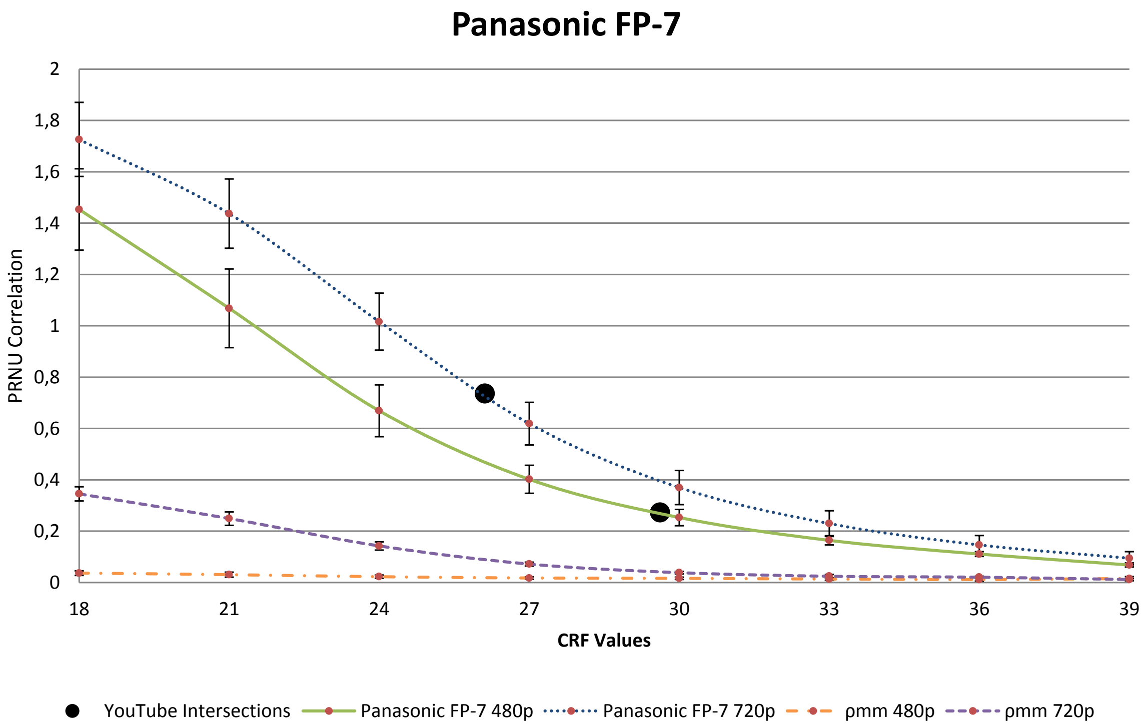

FIG. 2: The effect of encoding on the PRNU values of videos shot by the Panasonic FP-7

to 0.382 and lowers ρ_{mm} to 0.114, but this is still far too low a correlation to be reliable.

5. Canon PowerShow SX 210IS

This camera has the same outcome as the Canon Ixus 220HS for the 480p videos: a very low matching correlation and a relatively high mismatching correlation. Equally, it is not considered reliable and should not be used for PRNU comparisons.

The 720p videos were proven to work much better though, possibly because of the greater amount of pixels the higher resolution is able to capture.
With $\rho_m = 0.849$ and $\rho_{mm} = 0.093$, a difference of factor 9, it is adequate for PRNU comparisons.

B. Second layer of compression

After analyzing the pre-encoding values, we were able to show the effect on the PRNU correlation values after our second layer compression was applied. Since we pointed out for which cameras PRNU comparisons still works, we provided graphs of two cameras; the Panasonic FP-7, which was considered as reliable, and the Canon PowerShot SX210 IS, wich was considered partially reliable. The results for the other cameras are provided as graphs 3, 4, 5 in the appendix and will not be discussed in detail.

1. Panasonic FP-7

Graph 2 proves that the PRNU comparison technique is perfectly applicable on videos shot by the Panasonic FP-7, even after a high compression is applied on the videos. We determined that up until a compression of constant rate factor 33, the PRNU pattern correlations were high enough to be distinguished and thus to be reliable.

After downloading the videos from YouTube, we determined that the extracted PRNU patterns were comparable of those videos compressed with a CRF value of around 27. As seen in table XII, YouTube maintains a slightly higher bit rate and stream size when compared to the 480p videos we manually compressed, but an almost identical bit rate and stream size when compared to the 720p videos. Since the PRNU correlation of the YouTube videos is as high as it is for this camera, it is possible to determine the original camera with which the YouTube video is created.

The T-test is equally positive for the Panasonic FP-7 at 480p; With a CRF of 18, $p = 0,001$ and at CRF 39, $p = 0,001$. YouTube gets $p = 0,001$. All significantly below the 0.05 threshold.

The same significance can be calculated for the 720p videos; At CRF 18, $p = 0,001$ and at CRF 39, $p = 0,001$. The YouTube video calculates at $p = 0,001$.

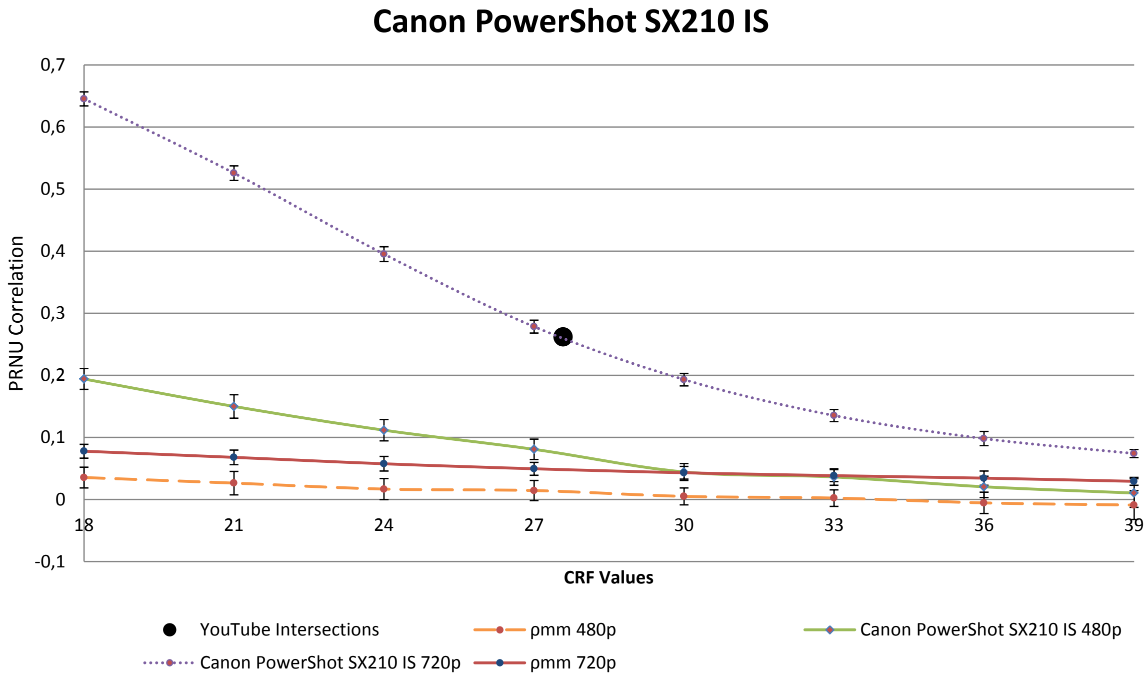

FIG. 3: The effect of encoding on the PRNU values of videos shot by the Canon PowerShot SX210 IS

2. *Canon PowerShot SX210 IS*

As mentioned before, we determined that even before a second layer of compression is applied, not all videos can be linked back to their original camera. This is for example the case with the Canon PowerShot SX210 IS V A 5.

As shown in graph 3, the correlation values of the 720p video were adequately high. We determined that up until a compression with a constant rate factor of 30, we were still able to correctly identify the matching camera.

The T-test for these videos backs up our statement; for the videos encoded at CRF 18, $p = 0,001$ and at CRF 39, $p = 0,001$. The YouTube video's compression compared to that of the video we encoded with CRF 27 and thus, it's possible to correctly identify the original camera. A T-test value of $p = 0,001$ is calculated for the YouTube video.

However, since the PRNU value correlations of the 480p videos were insufficient even before our second layer of compression, it would show that compressing the video would not improve the correlation values at all.

Even if we consider the highest constant rate factor we tested on, 18, ρ_m and ρ_{mm} are far too low to be successfully compared. We calculated the PRNU correlation values for all the CRF settings and they got worse and worse as the compression rate got higher.

The T-test confirms this, at CRF 18 $p = 0,001$, which statistically can be considered as a significant result. However if we take into consideration that the average correlation of the video patterns on this encoding level is only 0.19 (out of the maximum of 3) and the standard deviation 0.03, we decided that this significance value has to be disregarded. At CRF 30, $p = 0.06$ and crosses the set threshold of 0.05. All the values obtained with a higher encoding become statistically insignificant.

The PRNU patterns are so unreliable, that the pattern we extracted from the video after it was uploaded to YouTube, doesn't even come close to any of our predefined CRF settings. After some additional testing we determined that a CRF of 45 is necessary to reduce the PRNU correlation to the one we got from YouTube. Needless to say, the video quality was completely destroyed. In terms of stream size and bit rate, the YouTube video does correlate with the video encoded at a constant rate factor of 27, similarly to the other videos we've tested on other cameras. The T-test is also good for the YouTube video; p equals to 0.001 which is considered significant by our pre-defined standards.

This proves that YouTube did something to the PRNU pattern that we can not distinguish or reproduce, and further proves that the Canon cameras we tested are not suitable for PRNU comparisons, apart from the 720p video shot with the Canon PowerShot SX210.

VI. CONCLUSION

From our results we can conclude that using a comparison of pattern noise values to link videos to their camera's origin works in some cases. It all depends on the brand of the camera, the resolution and the amount of compression that is done on the video. The reliability of the PRNU pattern decreases exponentially if the compression becomes higher.

We proved that the robustness of the PRNU pattern is camera brand-specific. The Panasonics and the Apple iPhones we've tested performed exemplary. Before re-encoding the videos, a distinguishable pattern could already be extracted and there was a clear link between the video and the camera. It was subsequently shown that applying the encoding with different CRF values had a destructive effect on the original pattern noise. Graphs 2 through 5 show that, as the compression increases, the pattern values decrease. By comparing the YouTube pattern noise values to the values of the re-encoded videos, we conclude that YouTube applies a level of compression which is similar to a constant rate factor between 27 and 30. The pattern noise extracted from those videos made it possible to link these videos to their original camera.

The Canon videos have proven to be much more unreliable. On a resolution of 480p, it is impossible to adequately find the original camera with the pattern noise extracted from the videos. For 720p, we can conclude that it depends on the specific camera, since a pattern noise comparison only worked 50% percent. However, the same exponentially decreasing trend of the pattern noise is found as with the other cameras. This means that if the extracted pattern noise before the re-encode is sufficient, the chance of linking the videos to their original camera after the re-encode increases. The YouTube videos can not always be compared to the original camera, it definitely depends on the resolution, where a resolution of 720p proves to be sufficient, even for the Canon cameras. With the videos recorded at 480p, none of the patterns were preserved enough to be successfully compared with the original camera.

Applying a second layer of compression using the Advanced Video Codec has a clear negative effect on the pattern noise. The video resolution, camera brand and level of encoding plays an integral role in the reliability of comparing PRNU patterns. The process of matching a re-encoded video, be it encoding from an online video service or after manually applying the compression, with its original camera, is case dependent.

VII. SUGGESTIONS FOR FURTHER RESEARCH

Since our study focussed on videos shot with a resolution of 480p and 720p, it would be interesting to conduct a similar study with higher resolution videos; ie. 1080p or 2304p.

The difference in reliability of the pattern noise has proven to be camera brand specific. More cameras can be researched, be it from the same brand or from a more diverse set of brands, and a possible link between them could be researched.

Acknowledgments

We would like to thank the Netherlands Forensics Institute for presenting us with the opportunity of conducting our research at their facilities and making their resources available to us. Specially, we would like to thank Zeno Geradts for assisting us on-site and supervising the project.

We offer our gratitude to Marcel Worring, who assisted us with writing the final report and for giving us invaluable input needed for analyzing the data.

Ewald Snel provided us with a modified version of PRNUCompare, making a command line interface possible. His work made it possible for us to greatly increase the speed in processing our data.

Lastly, we would like to thank Cees de Laat for making the research projects in their current form possible.

[1] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. 2006.

[2] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Determining digital image origin using sensor imperfections. 2005.

[3] Mo Chen, Jessica Fridrich, and Miroslav Goljan. Digital imaging sensor identification (further study). 2007.

[4] Miroslav Goljan, Jessica Fridrich, and Tomáš Filler. Large scale test of sensor fingerprint camera identification. 2009.

[5] Wiger van Houten and Zeno Geradts. Using sensor noise to identify low resolution compressed videos from youtube. 2009.

[6] Alin C. Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. 2005.

[7] Hany Farid. Digital image ballistics from jpeg quantization. 2008.

[8] Zeno Geradts and Thomas Gloe. D6.8b: Identification of images. 2009.

[9] `http://prnucompare.sourceforge.net/`.

[10] `http://www.avidemux.org/`.

APPENDIX A: PRE-ENCODING RESULTS TABLES

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**2.021**	**2.041**	**1.849**	**2.150**	**1.831**
ρ_{mm}	0.054	0.071	0.048	0.048	0.047

TABLE II: Panasonic FP-7. Natural videos encoded with Motion JPEG in 640x480 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**2,128**	**1.771**	**2.136**	**1.957**	**2.233**
ρ_{mm}	0.480	0.363	0.527	0.528	0.484

TABLE III: Panasonic FP-7. Natural videos encoded with Motion JPEG in 1280x720 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**0.772**	**0.690**	**1.201**	**0.758**	**1.065**
ρ_{mm}	0.031	0.025	0.020	0.040	0.031

TABLE IV: Panasonic FZ-45. Natural videos encoded with Motion JPEG in 640x480 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**1.071**	**1.222**	**1.349**	**1.347**	**1.578**
ρ_{mm}	0.348	0.419	0.304	0.480	0.316

TABLE V: Panasonic FZ-45. Natural videos encoded with Motion JPEG in 1280x720 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**0.639**	**0.666**	**0.840**	**0.666**	**0.967**
ρ_{mm}	0.016	0.046	-0.026	0.003	-0.034

TABLE VI: Panasonic FZ-45. Natural videos encoded with AVC (H.264/MPEG-4) in 1280x720 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**1.758**	**1.436**	**1.339**	**1.398**	**1.331**
ρ_{mm}	0.007	0.007	-0.007	0.013	0.009

TABLE VII: Apple iPhone 4. Natural videos encoded with AVC (H.264/MPEG-4) in 1280x720 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**0.194**	**0.287**	**0.333**	**0.219**	**0.181**
ρ_{mm}	0.092	0.164	0.146	0.176	0.111

TABLE VIII: Canon Ixus 220HS. Natural videos encoded with AVC (H.264/MPEG-4) in 640x480 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**0.172**	**0.431**	**0.468**	0.114	**0.366**
ρ_{mm}	0.0948	0.140	0.133	**0.124**	0.120

TABLE IX: Canon Ixus 220HS. Natural videos encoded with AVC (H.264/MPEG-4) in 1280x720 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**0.360**	**0.256**	**0.319**	**0.383**	**0.331**
ρ_{mm}	0.069	0.085	0.083	0.056	0.071

TABLE X: Canon PowerShot SX210 IS. Natural videos encoded with AVC (H.264/MPEG-4) in 640x480 resolution. Frame averaging = 10, $\sigma = 5$

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
ρ_m	**0.968**	**0.788**	**0.959**	**0.824**	**0.706**
ρ_{mm}	0.080	0.098	0.094	0.089	0.107

TABLE XI: Canon PowerShot SX210 IS. Natural videos encoded with AVC (H.264/MPEG-4) in 1280x720 resolution. Frame averaging = 10, $\sigma = 5$

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	15245266	4063	900	0.441	**1.455**	0.033
21	9084877	2420	900	0.263	**1.052**	0.031
24	5467827	1456	900	0.158	**0.656**	0.025
27	3527106	938	900	0.102	**0.404**	0.023
30	2402719	639	900	0.069	**0.262**	0.014
33	1701471	452	900	0.049	**0.163**	0.009
36	1241775	329	900	0.036	**0.117**	0.010
39	930918	246	900	0.027	**0.068**	0.023
YouTube	4234560	1129	900	0.123	**0.306**	0.019

TABLE XII: Panasonic FP-7. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 640x480 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	39152267	10439	720	0.472	**1.737**	0.322
21	23866348	6363	720	0.288	**1.403**	0.231
24	13838272	3688	720	0.167	**0.957**	0.135
27	8065925	2149	720	0.097	**0.584**	0.071
30	4931987	1313	720	0.059	**0.262**	0.014
33	3182543	847	720	0.038	**0.210**	0.034
36	2166939	576	720	0.026	**0.141**	0.026
39	1564410	415	720	0.019	**0.068**	0.023
YouTube	11139543	2971	720	0.134	**0.631**	0.054

TABLE XIII: Panasonic FP-7. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 1280x720 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	12437142	3317	900	0.360	**0.411**	0.018
21	7658856	2042	900	0.222	**0.288**	0.018
24	4834486	1289	900	0.140	**0.196**	0.014
27	3164089	844	900	0.092	**0.129**	0.011
YouTube	3745965	999	900	0.108	**0.122**	0.021
30	2145344	572	900	0.062	**0.089**	0.016
33	1505498	401	900	0.044	**0.055**	0.012
36	1096322	292	900	0.032	**0.032**	0.008
39	818389	218	900	0.024	**0.024**	0.006

TABLE XIV: Panasonic FZ-45. Natural video encoded in Motion-JPEG with different crf settings in 640x480 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
crf 18	36573318	9753	900	0.353	**0.784**	0.246
crf 21	21094473	5625	900	0.203	**0.569**	0.152
crf 24	12400886	3307	900	0.120	**0.393**	0.088
crf 27	7574475	2020	900	0.073	**0.244**	0.049
YouTube	11272479	3006	900	0.109	**0.254**	0.034
crf 30	4831772	1288	900	0.047	**0.138**	0.023
crf 33	3246826	866	900	0.031	**0.080**	0.015
crf 36	2301853	614	900	0.022	**0.050**	0.011
crf 39	1683259	449	900	0.016	**0.038**	0.004

TABLE XV: Panasonic FZ-45. Natural video encoded in Motion-JPEG with different crf settings in 1280x720 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
crf 18	25000338	13333	750	0.289	**0.425**	4.37E-04
crf 21	16070373	8571	750	0.186	**0.334**	9.38E-04
crf 24	10224776	5453	750	0.118	**0.248**	-0.002
crf 27	6587042	3513	750	0.076	**0.164**	-0.004
YouTube	5344166	2850	750	0.124	**0.109**	-0.001
crf 30	4348511	2319	750	0.050	**0.098**	0.016
crf 33	2968503	1583	750	0.034	**0.059**	0.002
crf 36	2097618	1119	750	0.024	**0.038**	9.26E-04
crf 39	1524840	813	750	0.018	**0.026**	0.007

TABLE XVI: Panasonic FZ-45. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 1280x720 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
crf 18	42582761	11365	893	0.414	**1.341**	0.012
crf 21	26603542	7100	893	0.259	**1.118**	0.013
crf 24	15342099	4095	893	0.149	**0.762**	0.013
YouTube	11403499	3044	899	0.110	**0.469**	0.013
crf 27	9050514	2416	893	0.088	**0.455**	0.016
crf 30	5704331	1522	893	0.055	**0.284**	0.014
crf 33	3768058	1006	893	0.037	**0.188**	0.012
crf 36	2598929	694	893	0.025	**0.135**	0.014
crf 39	1874824	500	893	0.018	0.091	**0.011**

TABLE XVII: Apple iPhone 4. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 1280x720 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	11348272	3027	899	0.329	**0.181**	0.097
21	7422328	1980	899	0.215	**0.165**	0.094
24	4948911	1320	899	0.143	**0.164**	0.099
27	3394700	905	899	0.098	**0.157**	0.097
30	2384345	636	899	0.069	**0.149**	0.097
33	1695970	452	899	0.049	**0.144**	0.110
36	1231182	328	899	0.036	**0.145**	0.114
YouTube	3165042	844	899	0.092	**0.116**	0.061
39	911394	243	899	0.026	0.064	**0.116**

TABLE XVIII: Canon Ixus 220HS. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 640x480 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	29628720	7902	899	0.286	**0.308**	0.132
21	19799344	5280	899	0.191	**0.274**	0.126
24	13119435	3499	899	0.127	**0.235**	0.120
27	8936600	2383	899	0.086	**0.202**	0.118
30	6349131	1693	899	0.061	**0.180**	0.121
YouTube	10752435	2868	899	0.104	**0.168**	0.103
33	4588012	1224	899	0.044	**0.155**	0.127
36	3407682	909	899	0.033	**0.145**	0.130
39	2599106	693	899	0.025	**0.138**	0.134

TABLE XIX: Canon Ixus 220HS. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 1280x720 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	11636834	3104	899	0.337	**0.195**	0.008
21	7399041	1973	899	0.214	**0.143**	-0.005
24	4771953	1273	899	0.138	0.100	**-0.012**
27	3150379	840	899	0.091	0.063	**-0.012**
YouTube	3598938	960	899	0.104	**0.045**	-0.020
30	2150640	574	899	0.062	**0.041**	-0.016
33	1514425	404	899	0.044	0.019	**-0.021**
36	1106026	295	899	0.032	0.007	**-0.031**
39	831776	222	899	0.024	-0.011	**-0.040**

TABLE XX: Canon PowerShow SX210 IS. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 640x480 resolution.

encoding	streamsize *byte*	kbit /s	frames	bpp	ρ_m	ρ_{mm}
18	35666436	9512	899	0.344	**0.747**	0.064
21	22902661	6108	899	0.221	**0.607**	0.053
24	14474121	3860	899	0.140	**0.452**	0.041
27	9109958	2430	899	0.088	**0.314**	0.035
YouTube	11290964	4233	899	0.109	**0.298**	0.034
30	5827005	1554	899	0.056	**0.212**	0.029
33	3871858	1033	899	0.037	**0.147**	0.026
36	2734769	729	899	0.026	**0.100**	0.018
39	2010834	536	899	0.019	**0.077**	0.022

TABLE XXI: Canon PowerShow SX210 IS. Natural video encoded in AVC (H.264/MPEG-4) with different crf settings in 1280x720 resolution.

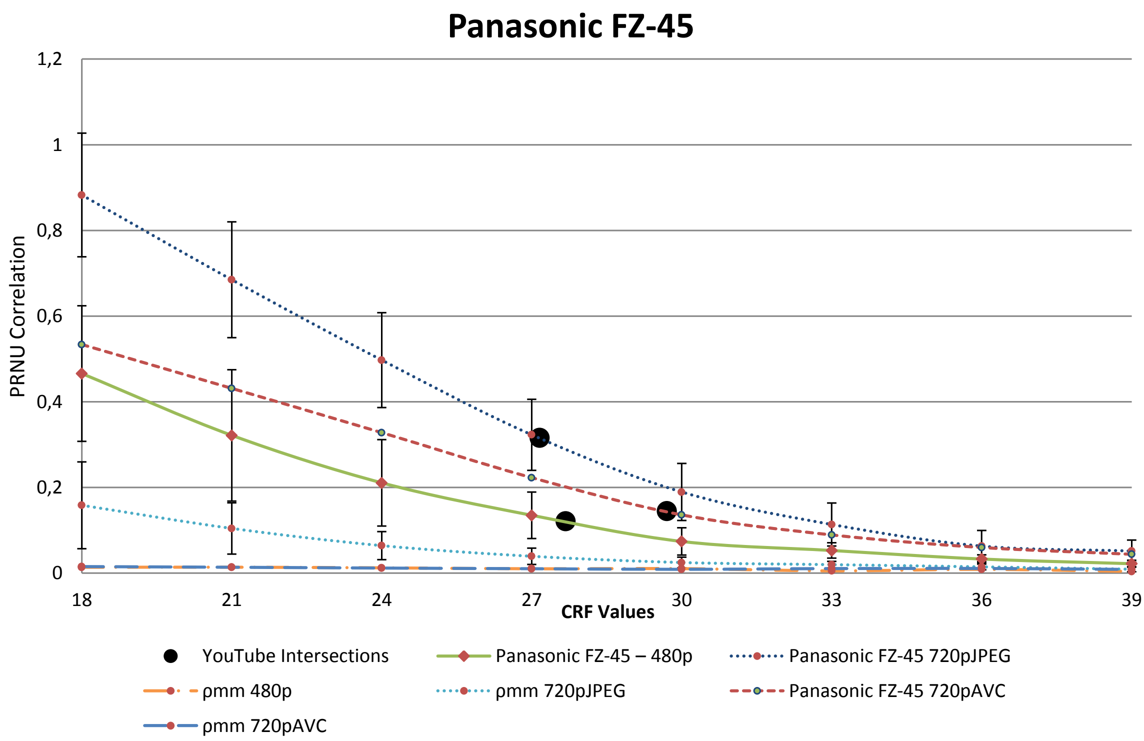

FIG. 4: The effect of encoding on the PRNU values of videos shot by the Panasonic FZ-45

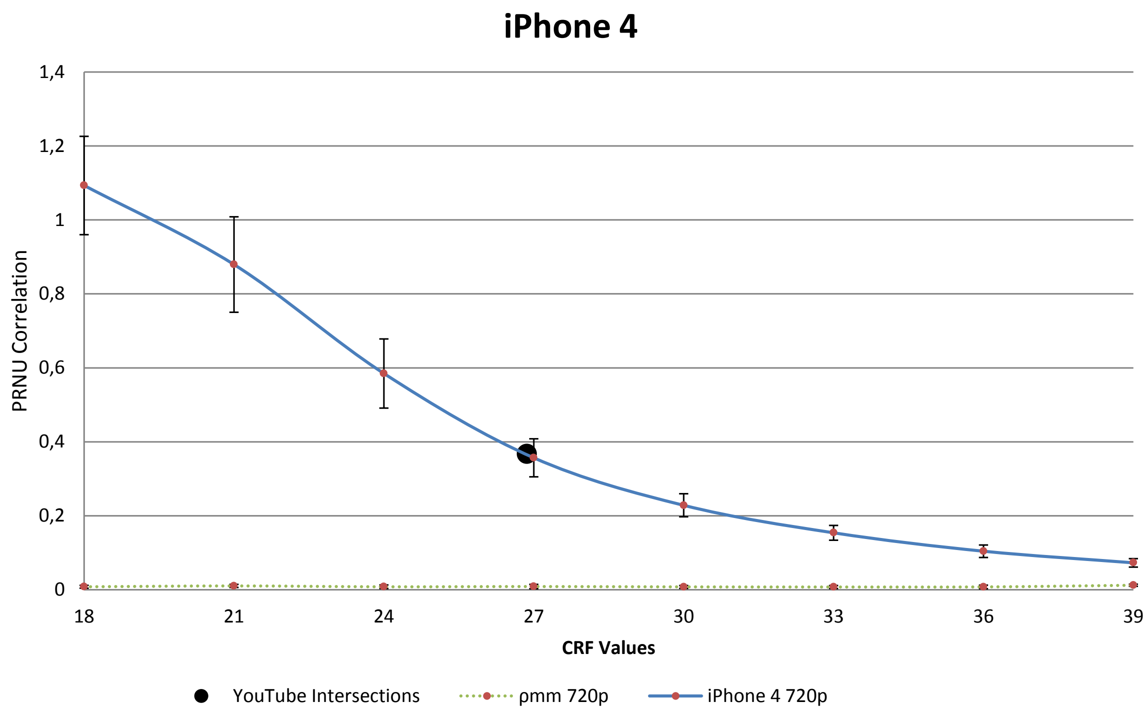

FIG. 5: The effect of encoding on the PRNU values of videos shot with the iPhone 4

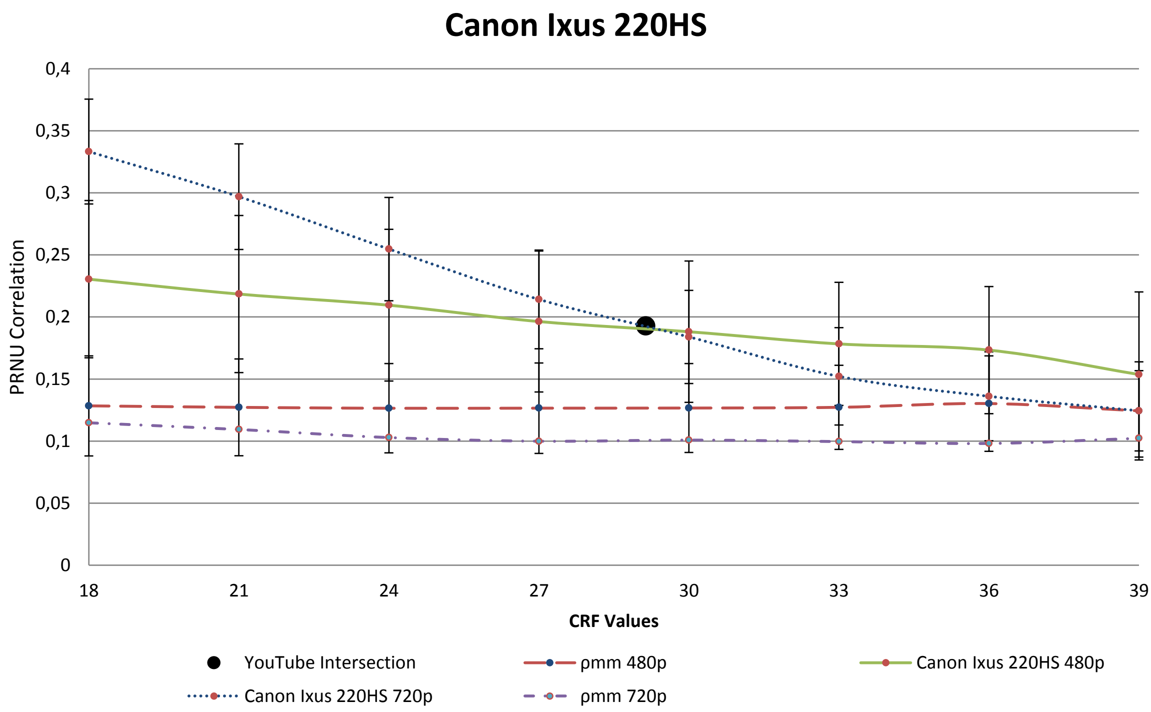

FIG. 6: The effect of encoding on the PRNU values of videos shot with the Canon Ixus 220HS