university of groningen

University of Groningen

Colour Stabilisation of Endoscopy Video

Bachelor Thesis

Author:
Sibren van Vliet
(s2754606)

Supervisors:
André Sobiecki
Alexandru C. Telea

SVCG Group
In collaboration with ZiuZ Visual Intelligence

19 July 2017

Contents

1 Introduction **3**

2 Related work **5**
2.1 Anbarjafari, Iterative n^{th} Root and n^{th} Power Colour Image Equalisation 6
2.2 Vig et al, Guided Filter Sub-Image Histogram Equalisation 6
2.3 Purushothaman, Hue and Intensity Differential Histogram Equalisation 7
2.4 Gautam et al, Adaptive Gamma Correction Range Limited Bi-Histogram Equalisation 7
2.5 González et al, Multi-Scale Retinex Chromaticity Improvement 8
2.6 Moradi et al, Contrast-Limited Adaptive Histogram Equalisation 8
2.7 Vazquez-Corral et al, Colour Matrix Transformation Video Tone Stabilisation 8
2.8 Wang et al, Colour State Smoothing Video Tone Stabilisation 9
2.9 Farbman et al, Colour Adjustment Map Video Tone Stabilisation 10
2.10 Bassiou et al, Absolute Discounting Back-Off Histogram Equalisation 10
2.11 Summary 10
2.12 Conclusion of literature study 11

3 Method **12**
3.1 Proposed algorithm 12
3.2 Software tool 14

4 Results and Discussion **16**
4.1 Qualitative evaluation 16
4.1.1 Effects of various algorithms and parameters 16
4.1.2 Survey 23
4.1.3 Effects on multiple frames 26
4.1.4 Conclusions of qualitative analysis 27
4.2 Quantitative evaluation 27

5 Conclusions **31**
5.1 Limitations 31
5.2 Stimulus for future work 32

References **33**

Appendices **34**

A Colour models **34**

B Software documentation **35**
B.1 Technology stack 35
B.2 Software architecture 35
B.3 Implementation of requirements 36

C Survey instructions **37**

1 Introduction

In the field of medicine, a practitioner may need to see inside of a patient's gastrointestinal tract in order to screen, diagnose, locate or treat a range of medical conditions, including gastrointestinal bleeding, inflammatory bowel disease, celiac disease, polyps, and certain types of cancer. Endoscopy, the practice of looking inside of a patient, is traditionally performed using a small camera on the end of a thin and flexible tube, inserted into the mouth and guided through the tract. However, while this works well for viewing the throat and stomach, it cannot easily traverse the many tight bends of the intestines. A more recent technology is the pill camera, a small capsule-shaped device equipped with a camera and lights. After the patient swallows the pill, it records 8 to 12 hours of video at three frames per second, with a resolution of 320×320 pixels.

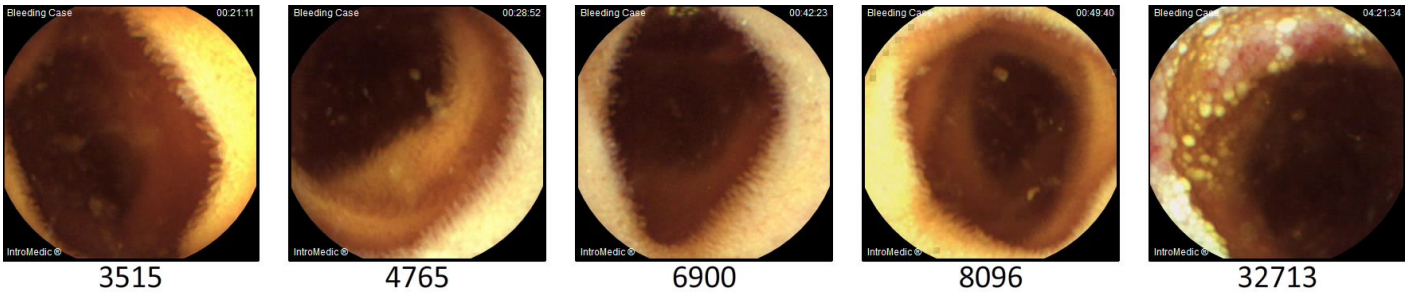


Figure 1: A selection of frames from endoscopy pill camera footage. The numbers below the images indicate the frame numbers.

Figure 1 shows five example frames from a video recorded using the MiroCam® pill camera. Each frame contains a circular picture surrounded by black borders, and some white text is overlaid, showing titles and a timer. We observe that, within each individual frame, there are large differences in the brightness of different areas of tissue. Areas close to the camera are very bright, while areas far away from the camera are very dark. This is due to the difference in distance from the capsule's lights.

Consider the image of frame 4765. In reality, all tissue in this shot has the same colour, but it is not displayed as such. Furthermore, realise that this is a frame of video, and that in the next frame, the camera pill has moved ahead through the gastrointestinal tract. As a result, the area that is moderately lit in the centre of frame 4765, will be too bright in the bottom right of frame 4766. Likewise, the area that is too dark in the top left of frame 4765 will be moderately lit in the centre of frame 4766. This means that, throughout time, the same area of tissue is shown in differing colours and brightnesses.

Because these inconsistencies can make it more difficult for the practitioner to interpret the video (for example, as details are hidden in dark areas), we want to stabilise the brightness in images. We will thus want to answer the question:

- **Research question 1:** How to brighten dark areas and darken bright areas?

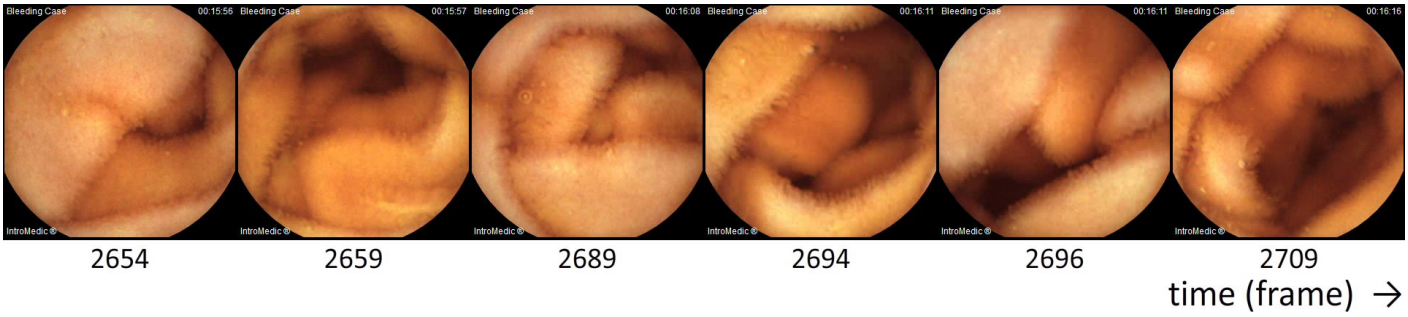


Figure 2: A series of frames from endoscopy pill camera footage, showing how the colour tone fluctuations throughout time.

Another problem is observed in Figure 2. The tissue shown in these images should all have the same colour, but because the camera automatically adjusts its colour balance, the overall colour tone of the frames fluctuate throughout time. At frame 2654, we observe a pink colour tone, while frame 2659 shows a more orange colour tone. Frame 2689 appears pink again, frame 2694 appears orange again, then frame 2696 shifts to pink again, and 2709 has an orange tone again.

When watching the endoscopy video, the practitioner may be distracted by these sudden colour tone fluctuations, even though there is not actually any information of interest in them. In order to allow practitioners to focus better on watching for medical anomalies, we want to smoothen out any sudden colour tone changes. We will thus also want to answer the question:

- **Research question 2:** How to develop a robust algorithm to normalise colour and brightness in video?

When applying the mentioned corrections to video, it is important that the manipulations do not damage the video, i.e. remove information or introduce information that is not actually in the source footage. Especially considering that this is a medical application, a practitioner should never be set on the wrong foot. We specify two types of false information: noise and relative pixel values.

Noise is likely to be introduced when correcting the brightness of very dark and very bright areas. Because of the underexposure and overexposure of these areas, the available information in the image is limited, and we have to work with what we have. We will have to perform a thorough evaluation on various algorithms and parameters values, in order to find out which techniques produce results with acceptable levels of noise.

The relation between pixel values refers to the fact that, while brightening a dark area, it should not become brighter than areas that are brighter than that dark area in the original image - and vice versa. Otherwise, false information is introduced as the areas that are clearly far away from the camera in the original image, may appear closer to the camera in the ouput image, or vice versa. To be more general and exact, we define this property, the relative pixel value rule, as follows: An algorithm takes an input image I and produces and output image O. The values of two pixels in I, say I_a and I_b, should have the same relation in O. For example, if the algorithm is applied to the brightness channel, and I_a is brighter than I_b, then O_a should also be brighter than O_b. However, in order to accommodate for integer rounding, it may occur that $O_a = O_b$ while $I_a \neq I_b$. Of course, this should only be the case if the values of I_a and I_b are very close. To summarise:

- $I_a < I_b \rightarrow O_a \leq O_b$
- $I_a = I_b \rightarrow O_a = O_b$
- $I_a > I_b \rightarrow O_a \geq O_b$

Through evaluation, we will find the optimal combinations of algorithms and parameters for stabilisation with minimal damage, thus answering the question:

- **Research question 3:** Which techniques should be used to perform stabilisation without damages?

In order for practitioners to see and use the techniques, we will build a user-friendly computer program. The practitioner can use this to preview the results of applying certain techniques to a video of endoscopy footage. The requirements for the program are as follows:

- The user can load a video file.
- The user sees the original video next to the video with corrections applied to it.
- The user can play, pause, and seek in the video.
- The user can select which algorithms to apply to the video, and sees the results of changing settings in real time.
- The user can change the parameters of each algorithm, and sees the results of changing settings in real time.
- The user can render a video with the chosen techniques and parameters applied to it.

The intention is that the practitioner uses the program in the order of these requirements. Once they have chosen the algorithm settings and the preview is looking satisfactory, the output video can be rendered, producing the full video with corrections applied. This should be achieved in real time, because in the scenario of a serious medical condition or emergency, waiting a long time is unacceptable. Furthermore, if the same techniques were to be applied to live video (such as traditional endoscopy cameras or advanced endoscopy pills which wirelessly transmit video images), the enhanced video should also appear in real time.

The rest of this thesis is structured as follows. In Section 2, we perform a literature study on related works proposing algorithms of our interest. After choosing an algorithm, we propose an algorithm to complement its shortcomings in Section 3. In Section 4, the results of applying the algorithms to endoscopy video are shown and discussed, and in Section 5, we will draw our conclusions from this.

2 Related work

In order to find algorithms to apply to the endoscopy imagery, we performed a literature study. Using IEEE Xplore[1], ScienceDirect[2] and ACM Digital Library[3], we searched for articles proposing algorithms for stabilisation, correction, normalisation and equalisation of colours, brightness and histograms, as well as techniques for improving the quality of endoscopy video in general. We found the following ten articles proposing algorithms we consider relevant to this thesis.

Algorithm paper	Relevance to the project
[4] (Anbarjafari)	A single image algorithm based on the n^{th} root and n^{th} power equalisation technique. When applied to the HSI representation of an image, it is a powerful algorithm for lighting up the colours in very dark areas of the image and shading the colours in very bright areas of the image, which is a valuable enhancement for the unevenly lit endoscopy footage.
[5] (Vig et al)	A single image algorithm based on histogram equalisation. It is specifically aimed at increasing the brightness of dark (underexposed) areas of an image, so areas that are too bright are not darkened. This algorithm can be useful for dark frames in video footage, but is less useful in overexposed frames.
[6] (Purushothaman et al)	A single image algorithm based on Differential Histogram Equalisation for Color Images (DHECI). It increases the contrast of colour images in order to make the colour information more visible to the human eye, but as a result, brightly lit areas of the image may become even brighter, losing potentially valuable information in endoscopy footage.
[7] (Gautam et al)	A single image algorithm based on histogram equalisation. It increases the contrast in dimly lit areas of an image, while not brightening properly lit areas, thus not losing any information there. However, areas that are too bright are not darkened.
[8] (González et al)	A single image algorithm based on the Multiscale-Retinex Algorithm. It is very powerful at brightening dark areas of an image and thus revealing rich colour information hidden to the human eye. However, properly lit areas of an image may become too bright with this algorithm, thus losing information in endoscopy footage close to the camera.
[9] (Moradi et al)	A single image algorithm based on histogram equalisation. This algorithm is specifically aimed at endoscopy capsule images, which suffer from low contrast and noise distortion. The algorithm increases the contrast and removes noise, but does not appear to be very powerful for lighting up dark areas and darkening overly bright areas.
[10] (Vazquez-Corral et al)	A multiple image algorithm based on using a reference image. It is suitable for images from several cameras, or from the same cameras of which the automatic white balance or automatic exposure changes over time. Given a reference frame with 'correct' colouration, it will stabilise the colours in other frames to match those of the reference frame. This can be used in endoscopy video to remove variations in brightness or colouration as the camera automatically adjusts its settings. A challenge, however, is figuring out which frame to choose as the reference frame.
[11] (Wang et al)	A multiple image algorithm based on smoothing. It smooths out differences in video tone by taking the colour state of neighbouring frames into account, functioning like a running average through time. The algorithm is configurable in the way of where the threshold lies between small jitter that should be smoothed out, and extreme tone differences that should be kept in order to not lose information.
[12] (Farbman et al)	A multiple image algorithm based on using a reference image. It is very similar to the earlier listed algorithm using this technique, but unfortunately the code appears to be closed-source and patented, so using it for this project could prove troublesome.
[13] (Bassiou et al)	A single image algorithm based on histogram equalisation. It uses multi-level smoothing in order to correct images in the HSI colour space, using the probability density functions of the saturation and intensity components while leaving the hue unchanged. This algorithm appears to successfully brighten dark areas of images while darkening overly bright areas, which is desirable in endoscopy footage.

During this literature orientation, we have learnt that Global Histogram Equalisation (GHE) is very widely used, mostly due to its simplicity and the large improvement it already makes on imagery with bad contrast. Due to its simplicity, however, it is not very powerful and can cause artefacts, and colour information in very bright and very dark areas remains unpronounced. An improved algorithm is Differential Histogram Equalisation (DHE), but in recent papers this too is considered very basic. Therefore, we think that implementing GHE and DHE is not very worthwhile, considering that many existing algorithms show better results. The ten articles presented in the table above all build and improve upon these basic algorithms, or even upon algorithms which are already superior to GHE and DHE.

Because we do not have the time and resources to implement all ten algorithms, we rate the ten papers on four properties, so that we can make a well-founded decision on which algorithm(s) to implement. The four properties we evaluate are:

- Validation: How well the algorithm is tested by the authors in the paper, and how promising the results are looking for this project;
- Availability: Whether the source code is available on the Internet, or if not, whether the algorithm is described in enough detail for us to reproduce it;
- Complexity: Whether the algorithm could be executed fast enough to be close to real time;
- Usability: Refers to the number of parameters and their meaningfulness. A large number of parameters is bad for ease of use, and it should be easily understandable what a parameter affects and what the result of adjusting it will be.

For the complexity rating, we make an estimation of the complexity of the algorithm, for performance comparison, based on the formulas presented in the papers. We will use the notations w and h for the width and the height of each image, and the notation n for the number of images it needs to be applied to in order to process a whole video. This way, single-image and multiple-image algorithms can be compared fairly.

2.1 Anbarjafari, Iterative n^{th} Root and n^{th} Power Colour Image Equalisation

The algorithm proposed in [4] is a simple method applied to a single channel of a single image. Each pixel in that channel has a value in $[0, 1]$, and the mean of all pixel values in the channel is μ_c ('current mean'). The goal of the algorithm is to iteratively manipulate the pixel values in the channel, so that the mean reaches a desired value μ_g ('goal mean'). To do this, in each iteration the current pixel value mean μ_c is calculated, and compared to the goal mean μ_g. If $\mu_c \approx \mu_g$, the image channel has the desired mean, and the algorithm is done. Otherwise, a value Θ is calculated as:

$$\Theta = \frac{\ln \mu_g}{\ln \mu_c}$$

Then each pixel in the channel is raised to the power Θ, which will increase the value of pixels with a value smaller than μ_g, and decrease the value of pixels with a value greater than μ_g. The relative pixel value rule is adhered because every pixel is raised to the same power. The iterative process then loops. In order to brighten dark areas and darken bright areas in images, the algorithm should be applied to the intensity channel if the image is provided in the HSI colour space. For more information on colour spaces, see Appendix A.

Validation: In the paper, the author shows the results of using various algorithms to improve two images with bad contrast. For both images, the proposed method yields the best result, where too dark areas are brightened, too bright areas are darkened, and the colour remains rich.

Source code: The source code, provided by the author of the paper, is available in the MATLAB language on the MathWorks website[14].

Complexity: All steps of the algorithm are operations on each pixel of the image, so have a complexity of $\mathcal{O}(wh)$. The iterative power/root calculation of the image is repeated an unknown number of times x. When testing the provided code, x was near 10. Because this algorithm is performed on each image separately, the complexity for colour-correcting a video is $\mathcal{O}(whnx)$ with $x \approx 10$.

Usability: The algorithm as proposed in the paper does not have any parameters, because it always assumes $\mu_g = 0.5$. However, we think that the algorithm can be more powerful if the user is allowed to choose the goal mean within the range $[0, 1]$. The goal mean is also a very meaningful parameter: if the algorithm is applied to the intensity channel, a goal mean of 0 makes the image as dark as possible, a goal mean of 0.5 makes the image moderately bright, and a goal mean of 1 makes the image as bright as possible.

2.2 Vig et al, Guided Filter Sub-Image Histogram Equalisation

The algorithm proposed in [5] is built as a pipeline. Given an imagine in HSI, first, the exposure of the image is calculated, which can be used to divide the image into overexposed and underexposed regions. Next, the saturation and intensity sub-images are transformed as follows: $I_{new} = \lambda V^{\beta}$ where λ is an intensity constant and $\beta < 1$; and $S_{new} = \sigma S^{\alpha}$ where σ is a saturation constant and α is slightly greater than 1. Next, histogram equalisation is performed on I using

$I_{new} = 1/(1 + ((1 - I)/I)^{0.5})$. After performing Histogram Matching and the ESIHE techniques, the CDF and PDF of sub-images are used for histogram equalisation, before the sub-images are compiled into one image again. Finally, the guided image filter algorithm is applied.

Validation: In the paper, the authors show the results of performing the algorithm on two very dark photographs. The resulting corrected images restore the colours very impressively. However, the algorithm is only shown brightening dark areas, so areas that are too bright, are not darkened, and some colour information may actually be lost there.

Availability: The source code of this algorithm is not directly available, but the code of techniques it is based on, such as Guided Image Filtering, can be found elsewhere. However, the authors do not mention all steps of the algorithm in detail, so reproducing the correct algorithm might prove difficult.

Complexity: All components of the pipeline appear to be in $\mathcal{O}(wh)$, and since the algorithm is to be applied to each image separately, the complexity of the algorithm applied to a video is in $\mathcal{O}(whn)$.

Usability: The algorithm has four parameters for the colour transformation $(\lambda, \beta, \sigma, \alpha)$. The values for these parameters bear a relation to the exposure value of the input image, but they are not very meaningful.

2.3 Purushothaman, Hue and Intensity Differential Histogram Equalisation

The algorithm proposed in [6] combines two existing techniques: intensity differential histogram equalisation (IDHE) and hue differential histogram equalisation (HDHE). An RGB image is converted to HSI, and the IDHE process O_I is performed on the intensity channel, and the HDHE process O_H is performed on the hue channel (of the original image). Then, the combined algorithm for finding the enhanced value of pixel i, j of input image I is given by $O(I(i,j)) = \beta \cdot O_I(I(i,j)) + (1 - \beta) \cdot O_H(I(i,j))$ where β is a weighing parameter with range $[0, 1]$.

Validation: In the paper, the authors show the results of applying their algorithm to two images with unfavourable contrast. The dark areas of the images are brightened so that the colours are more visible. However, bright areas of the images remain very bright or become oven brighter, risking the loss of colour information there.

Source code: The source code does not appear to be available on the Internet. However, the paper provides all the necessary mathematics for implementing IDHE and HDHE and for combining the two techniques, so reproducing the algorithm should not be difficult.

Complexity: The mathematics in the algorithm are quite heavy, because the calculation for each pixel's enhancement takes all other pixels into account, which is in $\mathcal{O}((wh)^2)$. For both algorithms, each of these algorithms is also performed x times where $x \in [0, 255]$, so on average $x \approx 128$. And because the algorithm needs to be applied to each individual frame, the resulting complexity is in $\mathcal{O}((wh)^2 nx)$ with $x \approx 128$.

Usability: The algorithm has one parameter, β, configuring the weight between the IDHE and HDHE components. While it is not very intuitive, it has a clear range of $[0, 1]$, so it does not negatively impact the usability.

2.4 Gautam et al, Adaptive Gamma Correction Range Limited Bi-Histogram Equalisation

The algorithm proposed in [7] seeks to offer high-level visual quality with low computational cost, by combining two existing techniques: range limited bi-histogram equalisation (RLBHE) and adaptive gamma correction (AGC). First, the image is converted from RGB to HSV, then RLBHE determines a threshold such that the image is divided into two sections, the foreground and the background. Based on this threshold, the upper and lower bounds for the histogram equalisation are calculated, and then this equalisation is applied to the V component of the HSV sub-images. Next, AGC is performed on the V component as well, before the result is converted from HSV back to RGB.

Validation: In the paper, the authors show the results of applying the algorithm to 5 images with unfavourable contrast, showing good enhancement. While bright areas in the input are darkened in order to show the true colours better, dark areas appear to become even darker in some examples, which could lead to loss of colour information.

Source code: The source code does not appear to be available online. Reproducing the implementation from the paper might prove moderately difficult, because while a lot of the involved mathematics are given, not much intuition is provided for them.

Complexity: None of the mathematics in the papers appear to be greater than $\mathcal{O}(whn)$, and as the authors mention, this algorithm is intended to boast low computational complexity.

Usability: The algorithm does not have any parameters, because the threshold for histogram equalisation and the gamma for adaptive gamma correction are automatically found by the presented techniques.

2.5 González et al, Multi-Scale Retinex Chromaticity Improvement

The algorithm proposed in [8] performs the MSR algorithm on each pixel of the image. This algorithm is defined by the formula $R_{MSR_i} = \sum_{n=1}^{N} \omega_s R_{n_i} = \sum_{n=1}^{N} \omega_s \left[\log\left(\frac{I_i(x,y)}{I_i(x,y) \otimes G_s(x,y)}\right)\right]$ where N is the size of the area of surrounding pixels to take into consideration, and G is the Gaussian filter $G_s(x,y) = \frac{1}{\sigma_s\sqrt{2\pi}} \exp\left(-\frac{x^2+y^2}{\sigma_s^2}\right)$.

Validation: In the paper, the authors show the results of applying the algorithm to 6 example images. The results are very good, impressively recovering a lot of colour from extremely dark images. This, however, is the main purpose of the algorithm, so it is not designed to darken areas that are too bright.

Source code: The source code does not appear to be available online, and reproducing the algorithm from the paper could prove difficult, because some symbols in the math remain unexplained. However, since Multiscale-Retinex is a well-known technique, several implementations can be found on the Internet, including those in C++.

Complexity: From what can be gathered from the mathematics shown in the paper, equations are applied to each pixel and take an area of size N surrounding pixels into consideration for finding the necessary value. Because the algorithm needs to be applied to each image in the video individually, the overall complexity of the algorithm is likely to be in $\mathcal{O}(whNn)$.

Usability: The algorithm has 3 parameters: N, the size of the surrounding area that should be taken into account for the correction calculation of each pixel; σ_s, the scale of the Retinex model; and ω_s, the weight of the lightning estamation. While the first parameter is meaningful enough, the latter two are less intuitive.

2.6 Moradi et al, Contrast-Limited Adaptive Histogram Equalisation

The algorithm proposed in [9] combines several existing technologies in order to enhance capsule endoscopy imagery. First, the Wiener filter is applied, which removes blur and noise from the image. Next, contrast-limited adaptive histogram equalisation is performed on each of the RGB channels, and finally, some blue is added to all pixels as a means of colour correction.

Validation: In the paper, the authors present the results of applying their algorithm to 4 images of capsule endoscopy footage. While the presented results show good improvements, they are most suitable for video pictures that have not been pre-processed at all. Considering that the footage provided for this project has basic contrast stretching applied to it, this algorithm may not provide the best results.

Source code: The source code does not appear to be available online. The paper is also not very descriptive of the implementation of the algorithm, making it difficult to reproduce.

Complexity: Given the few mathematics and existing techniques mentioned in the paper, the algorithm is probably in $\mathcal{O}(whn)$ when applied to each frame of the footage individually.

Usability: The paper does not clearly indicate the parameters. However, for Wiener filtering, the noise intensity to be removed should be given as a parameter in case it is not detected automatically, neither of which is described in the paper. Furthermore, there is no clear suggestion for the intensity of the shade of blue that the authors suggest adding to the resulting image, which could be given as a parameter as well.

2.7 Vazquez-Corral et al, Colour Matrix Transformation Video Tone Stabilisation

The algorithm proposed in [10] is based on the knowledge that colour processing inside the camera was actually originally designed for easy colour correction. On two images of the same scene, containing a common point p, this point is shown

on the first image as $\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{p1} = \left(A_1 \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{p} \right)^{\gamma_1}$ and on the second image as $\begin{bmatrix} R \\ G \\ B \end{bmatrix}_{p2} = \left(A_2 \begin{bmatrix} R \\ G \\ B \end{bmatrix}_{p} \right)^{\gamma_2}$, where γ_1 and γ_2 are the gamma values of the camera, and A_1 and A_2 are yet unknown transformation matrices. Then, if $\begin{bmatrix} r \\ g \\ b \end{bmatrix}_{p1}$ and $\begin{bmatrix} r \\ g \\ b \end{bmatrix}_{p2}$ are the colours of $p1$ and $p2$ with gamma correction undone, $\begin{bmatrix} r \\ g \\ b \end{bmatrix}_{p1} - H \begin{bmatrix} r \\ g \\ b \end{bmatrix}_{p2} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ with $H = A_1 A_2^{-1}$ is true for all pairs $p1, p2$. In order to undo gamma correction, the gamma values of the two images must be found first. This is done by using SIFT to find corresponding pixels between the two images, and repeating this for several pixel pairs until a gamma is found that results in the lowest error margin. Next, more accurate pixel correspondences between the linearised images are found, using SIFT, RANSAC and a method to discard matches that suppose a too large angular motion. Then, the matrix H is calculated by solving a system of equations, using mean average and first SVD-eigenvector of neighbourhood values in order to make the process more robust against variations. Finally, the gamma correction is re-applied.

Validation: In the paper, the authors present 24 examples of how the algorithm performs. Given an input frame and a reference frame, the colours of the input frame are corrected to match the colours of the reference frame. This appears to work very well in the shown results.

Source code: The source code does not appear to be available online, and the description of the algorithm workings in the article misses some crucial points, such as how exactly RANSAC is used in order to find a transformation for colour change, when it is usually used for location change.

Complexity: The algorithm consists mostly of matrix algebra and existing techniques such as SIFT and RANSAC, leading to believe that the algorithm is in $\mathcal{O}(whn)$. The authors mention that, because realtime implementations for SIFT and RANSAC exist, the proposed algorithm could also be made to work in realtime.

Usability: The authors advice on certain values to use for the parameters of the existing algorithms they use (SIFT, RANSAC, et cetera), resulting in only two parameters to configure: the half-width size ω and the threshold ratio T.

2.8 Wang et al, Colour State Smoothing Video Tone Stabilisation

The algorithm proposed in [11] stabilises the colour tone of a video across time, by representing each frame as a colour state, and smoothing the states within a range of successive frames. In order to know which pixels should have the same colour, corresponding pixels are found using pyramidal Lucas-Kanade, and the images are aligned by computing a homography. Then, pairs of pixels on the same coordinates should have the same colour. However, pixels near surface edges (using edge detection), under- and over-exposed pixels, and other outliers (using RANSAC) are excluded because their colours may be misleading. Furthermore, bilateral filtering is used to reduce noise. The pixels that are not excluded, are used in the colour transformation formula $\frac{\omega_c}{|\mathbf{X}|} \sum_x \|J_{x'} - \widehat{A}_i^j(I_x)\|^2 + \|A_i^j - I_{4\times4}\|^2$, where ω_c is a parameter, which the author give a value of 2000. After the colour states have been found for each image, their progression through time can be plotted. If this plot contains jitters, these should be smoothed out, until the plot consists of only static, linear and quadratic sections. This is achieved using L1 optimization and derivatives. This involves the parameters ω_1, ω_2, ω_3 and ω_s, of which the authors mention that ω_s is the most important.

Validation: In the paper, the authors present 7 examples of video clips in which the colour tone changes due to the camera automatically adjusting itself to changing surroundings. These results appear to nicely remove sudden jumps in brightness and colour balance.

Source code: The source code does not appear to be available online. The article provides a detailed description of the algorithm, but due to its difficult nature, reproduction could prove difficult.

Complexity: The algorithm employs matrix multiplications and existing techniques such as pyramidal Lucas-Kanade and RANSAC, but nothing of greater complexity than $\mathcal{O}(whn)$ appears to be involved.

Usability: The algorithm contains 5 parameters (although the authors claim that using given default values for four of them works well). The parameters all act as weights for components, but a meaningful explanation of them lacks.

2.9 Farbman et al, Colour Adjustment Map Video Tone Stabilisation

The algorithm proposed in [12] stabilises the tone of video using a reference frame, i.e. the colour tones of many frames is adjusted to match that of the reference frame. The idea is that a frame f_{i+1} can be colour-corrected using adjustment map A_{i+1}, which can be found when the previous frame f_i and its adjustment map A_i are given. The algorithm finds corresponding pixels between two frames by creating a robust set, i.e. the set of pixels of which the luminance only changed a very small amount between frames. Then, the map of robust signals is added to A_i, filling in the adjustment map A_{i+1} at robust pixels. Because the non-robust pixels need to be corrected as well, the adjustment level for these is found with a fast scattered data interpolation scheme, derived from Shepard's method. This way, the algorithm can be boiled down to a formula with two matrix formulas, but because of the computational cost involved in those, a faster method using eigenvalues is also shown.

Validation: In the paper, the authors present 5 examples of video clips enhanced using the algorithm. While the tone is indeed nicely stabilised every time, the colours do appear rather washed out in the first example.

Source code: No source code is provided online. Furthermore, when looking at the webpage of the project, it appears that the authors have patented the algorithm, so implementing it might present (legal) difficulties.

Complexity: The authors report that their MATLAB implementation of the algorithm processes HD video at 1.5 seconds per frame, but that there was still room for optimisation in their code. Because the algorithm involves matrix operations and existing algorithm, it is likely linear in $\mathcal{O}(whn)$.

Usability: The authors boast that the algorithm has only one tuneable parameter, σ_c which is the diffusion affinity of the Gaussian function. It is quite understandable how it should be used: a small value for very clear scenes with slow movement; a high value for noisy scenes with fast movement.

2.10 Bassiou et al, Absolute Discounting Back-Off Histogram Equalisation

The algorithm proposed in [13] works in the HSI colour space, and equalises the saturation and intensity channels simultaneously, using the common formulas for equalisation. However, this has the disadvantage that a sparse histogram is the result of equalisation: some bins are never filled, resulting in gaps. To solve this, one of many existing probability smoothing algorithms can be used. The authors use discounting back-off in their work, but a different model may suit a different image better. After this, the equalisation is complete, but there is a chance of a gamut problem, where the equalisation in certain pixels has resulted in incorrect colours because the result of the algorithm was outside of the colour gamut. This problem can be solved by converting the equalised HSI image to CMY first and applying an operation before converting back to RGB.

Validation: In the paper, the authors present 5 examples of images of which the colours are enhanced using the proposed algorithm. In all examples, colours appear to become more natural, and dark areas are brightened in order to reveal colours hidden by that darkness. In the first example, a very bright area is darkened to reveal otherwise lost colour information.

Source code: The source code is not provided by the author. However, a user on GitHub has uploaded an implementation of the algorithm in Python[15]. We experimented with this code, but it never resulted in the desired output, because it produces images with pixel values dramatically outside of the permitted value range. After implementing the algorithm ourselves, using the formulas from the paper, we ran into similar problems.

Complexity: The algorithm uses equalisation calculations and one of several probability smoothing techniques. Nonetheless, the final algorithm is likely to still be linear in $\mathcal{O}(whn)$.

Usability: The proposed does not have any parameters, except for the probability smoothing. Not only must the probability smoothing technique itself be chosen, but each has its own set of parameters.

2.11 Summary

The table on the next page summarises the ratings for the requirements of all algorithms. After comparing these, we rated their qualities in relation to each other (symbolised as $--$ for very bad, - for bad, +/- for indifferent, + for good and ++ for very good).

Algorithm	Validation	Source code	Complexity	Usability
[4] (Anbarjafari)	+ (2 tests show very good results)	++ (MATLAB code provided)	+ ($\mathcal{O}(whnx)$ with $x \approx 10$)	++ (0 parameters originally, 1 meaningful parameter in our version)
[5] (Vig et al)	+/- (2 tests show good results, but only for brighting dark areas)	- (No code provided, reproducing might prove difficult)	+ (Likely $\mathcal{O}(whn)$)	- (4 parameters, not very meaningful)
[6] (Purushothaman et al)	+/- (2 tests show good results, though mainly for brightening dark areas)	+/- (No code provided, but implementation clear and easy to reproduce)	- ($\mathcal{O}((wh)^2nx)$ with $x \approx 128$)	+ (1 parameter, not very meaningful but easy to understand)
[7] (Gautam et al)	+/- (5 tests show good results, although dark areas might become undesirably darker)	- (No code provided, reproducing might prove moderately difficult due to lack of intuition given for mathematics)	+ (Likely $\mathcal{O}(whn)$)	++ (0 parameters)
[8] (González et al)	+/- (6 tests show good results, although all focus only on brightening dark areas)	+/- (No code provided, reproducing might be moderately difficult)	+ (Likely $\mathcal{O}(whNn)$ where N is the size of the area around each pixel used for colour enhancement)	+/- (3 parameters, one meaningful, two not very meaningful)
[9] (Moradi et al)	+/- (4 tests show good results, but focusus on video without pre-processing)	- (No code provided, reproducing may prove difficuly due to vague description)	+ (Likely $\mathcal{O}(whn)$)	? (Not clearly described by the paper, probably 2, not very meaningful)
[10] (Vazquez-Corral et al)	++ (24 tests show good results)	+/- (No code provided, algorithm explanation leaves out some important details)	+ (Likely $\mathcal{O}(whn)$ (authors mention that realtime implementation is feasible))	+/- (2 parameters, not very meaningful)
[11] (Wang et al)	++ (7 tests show good results)	- (No code provided, reproducing could be difficult)	+ (Likely $\mathcal{O}(whn)$)	- (5 parameters, not very meaningful)
[12] (Farbman et al)	+ (5 tests show good results)	- (No code provided, algorithm patented by authors)	+ (Likely $\mathcal{O}(whn)$)	+ (1 parameter with clear use instruction)
[13] (Bassiou et al)	+ (5 tests show good results)	+/- (Third party code and formulas produce undesired results)	+ (Likely $\mathcal{O}(whn)$)	+ (Which probability smoothing method to use, and that method's parameters)

In the above table, the algorithm proposed by Anbarjafari (to which we will refer to as 'the Anbarjafari algorithm' for the remainder of this thesis) receives the most favourable rating, while all other algorithms present difficulty in some areas. For the sake of further research, we also attempted to implement the algorithms proposed by Vazquez-Corral et al and by Bassiou et al, but both attempts failed when we ran into problems with crucial details omitted from their respective papers.

2.12 Conclusion of literature study

After studying and comparing ten algorithms proposed in related works, we have chosen to implement and use the Anbarjafari algorithm. This algorithm brightens dark areas and darkens bright areas in individual images, but it does not stabilise the colour tone of video. In order to solve this shortcoming, we will propose a complementary algorithm in the next Section.

3 Method

In the previous Section, we researched ten existing algorithms relevant to the project, and selected the Anbarjafari algorithm for implementation. As explained in Subsection 2.1, the Anbarjafari algorithm brightens dark areas and darkens bright areas in single images. However, we also want to smoothen out colour fluctuations throughout time. Because the Anbarjafari algorithm does not do this, in Subsection 3.1 we propose a complementary algorithm, that is to be applied to endoscopy video in combination with the Anbarjafari algorithm. Then, in Subsection 3.2, we develop a software tool in order to test, present and use the two algorithms on endoscopy video.

3.1 Proposed algorithm

The complementary algorithm (to which we will refer as 'the proposed algorithm' for the remainder of the thesis) aims to smoothen out fluctuations in an image channel throughout time, by detecting large variances between the pixel value histograms of the frames within a time window, and manipulating the image channel's pixel values in such a way that the histogram is compressed accordingly. By compressing the histogram, differences between pixel values are made smaller, and when applied to all frames within a time window, the compression rate should progress gradually, in order to smoothen out sudden differences. This technique could be applied to any image channel, but in order to smoothen out colour tone fluctuations, we apply it to the saturation channel of the HSI colour model.

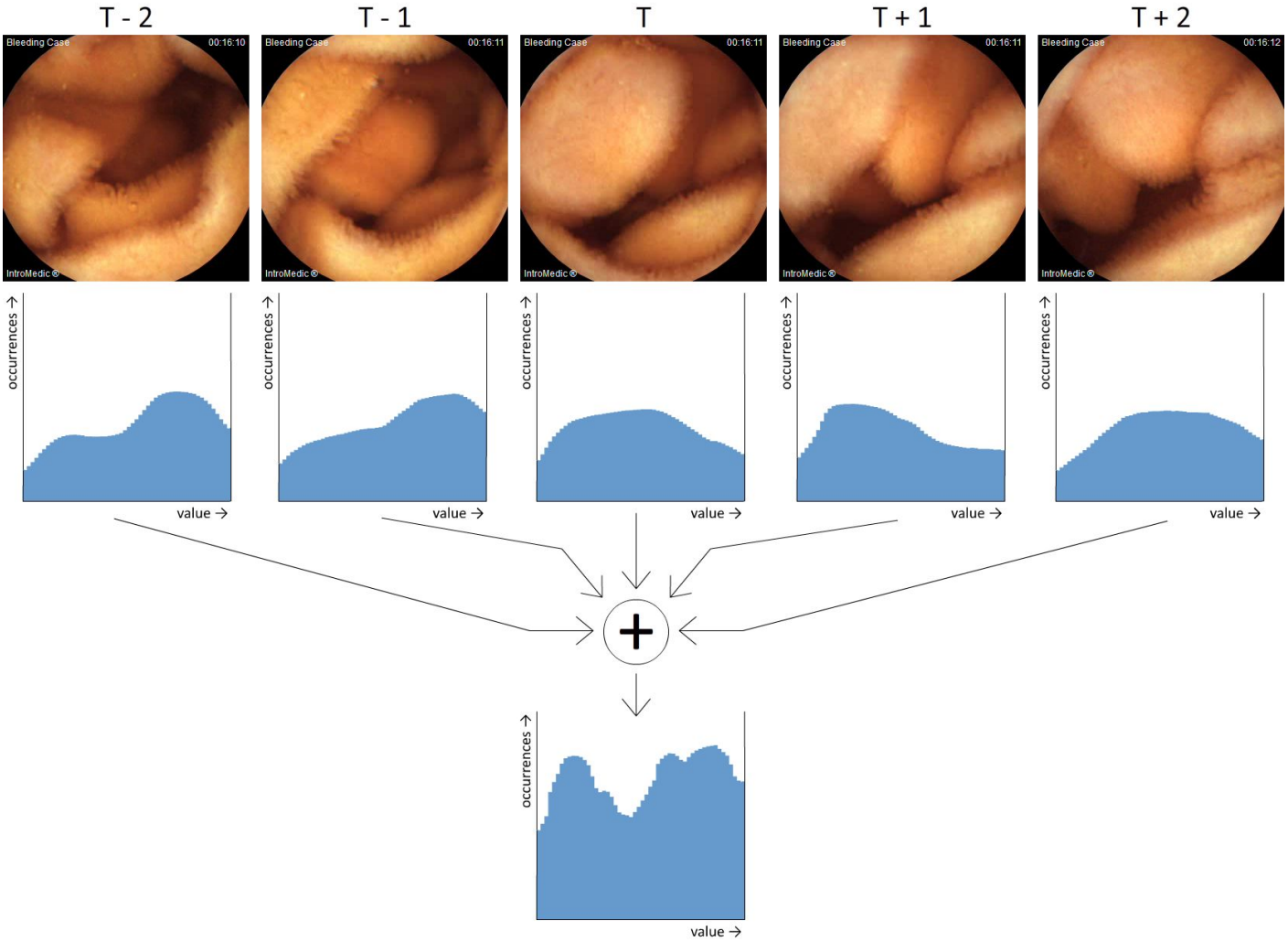


Figure 3: Five successive frames of video in which colour tone fluctuation occurs (from orange tone to pink tone). Below each frame, a histogram of pixel values in the frame's saturation channel is shown. (The histograms are mock-ups for demonstration purposes.) Summing these histograms results in a cumulative histogram on which measurements can be performed.

Figure 3 shows a frame T along with the two frames before and after it. Below each frame, a representation of the histogram of the saturation channel for that frame is shown. In the frames, we observe a sudden colour tone shift from orange to pink, which is undesired because we know that this tissue is all the same colour. We also observe a distinct change of shape of the saturation histograms.

We want to have a way of measuring the 'shape difference' between these histograms. The proposed method accumulates the given histograms, meaning that each bin H_x^C in the cumulative histogram H^C is the sum:

$$H_x^C = \sum_{i=-k}^{k} H_x^{T+i}$$

where H^{T+i} is the histogram of the desired channel of frame $T + i$. Note that the histograms of frames $T - k$ up to and including $T + k$ are taken into account, therefore the enhancing of a frame is influenced by the histograms of $2k + 1$ frames, including itself. k is a parameter of the algorithm, called the time window reach.

When the cumulative histogram is produced, its mean μ and variance σ^2 can be calculated. The latter is taken as the desired measurement of 'shape change' between the histograms within the time frame. A small variance indicates a small fluctuation, meaning that very little compression is necessary. A large variance indicates a large fluctuation, meaning that more compression is necessary in order to smoothen it out. Note that 'small' and 'large' are vague terms; we will use a parameter during later research in order to find more concrete boundaries.

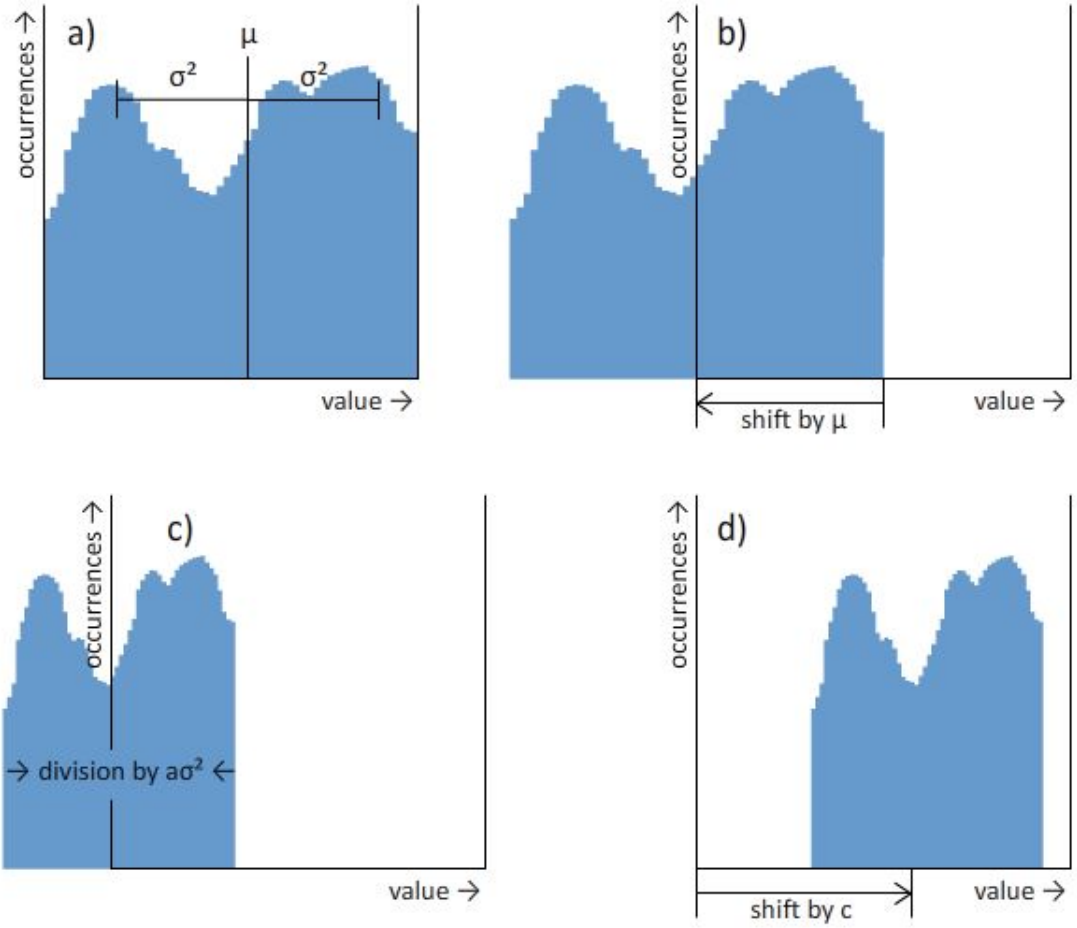


Figure 4: Visualisation of the process of histogram compression. a) The mean and variance of the histogram are measured. b) The histogram is shifted μ bins to the left, so that the mean is on the y-axis. c) By performing division by $a\sigma^2$ on all pixels, the histogram is compressed. d) A correction value c is added to all pixels, yielding the histogram of the output image.

Figure 4 visualises the steps of compressing an image's histogram. In the case of 8 bits per colour channel, the histogram has 256 bins. See Figure 4a. These bins and their contents signify the number of occurrences of a certain pixel value. With the mean value μ and variance σ^2 of a histogram calculated, we have a numeric description of how the pixel values are distributed. For example, in Figure 4a, observe that the mean pixel value is approximately 140, and that the variance is very wide.

As seen in Figure 4b, if μ is subtracted from all pixel values in the image, the mean of the histogram is on the y-axis. Then, dividing all pixel values by $a\sigma^2$ will result in a compressed histogram. See Figure 4c. a is a parameter of the algorithm, signifying the aggressiveness of the division. The range of a is $[0, 1]$. When $a = 0$, all pixel values are divided by 1, so no compression occurs. When $a = 1$, all pixel values are divided by σ^2, so that the histogram is compressed by an amount proportionate to the variance. During the testing phase, we will find a suitable value for a, providing the best results.

At this point, part of the histogram is in the negative space because of the shift in step 4b. Because we subtracted μ from all pixels, it appears intuitive to add μ to all pixels again, however the result may not always look satisfying to the human eye. More precisely, all pixel values may appear higher or lower than desired. Therefore, the user of the algorithm should be able to adjust the amount of shift, if a shift by μ is unsatisfying. This is done with the parameter c (for 'correction'), having the range $[0, 1]$. If the user chooses to correct with $c = 0$, then all pixel values are shifted so that the lowest value is in bin 0 of the histogram. If $c = 1$, then all pixel values are shifted so that the highest value is in bin 255 of the histogram. Any other correction causes the shift to move linearly between these extremes.

The relative pixel value rule is maintained with compression of the histogram. All pixel values are subtracted, divided and added with the same numbers, so that the shape of the histogram is not affected, as illustrated in Figure 4.

3.2 Software tool

In order to experiment and use both the Anbarjafari and the proposed algorithm on endoscopy video footage, we built a software tool according to the requirements in Section 1. For a detailed documentation of the implementation, see Appendix B. This is omitted here because the focus of this thesis is on the image processing, not so much the software engineering. What follows is a brief explanation of how the practitioner is expected to use the program.

The software tool is shown in Figure 5. The I/O functionality of the program is in the bottom-right corner, where the user can load a video file from the file system. Optionally, a black-and-white mask image can be opened as well. Then, the algorithms will only take pixels in account that overlap with white pixels in the mask.

The top half of the screen shows the current frame of the endoscopy footage, the original on the left, and the processed output on the right. Below this are the video controls, which allow the user to start and pause the video playback, seek through the video, and go to the previous frame, next frame, or a specific frame number.

The bottom-left and bottom-centre of the screen contain the algorithm controls. With the checkboxes on the left, the user chooses which algorithms to apply to the footage. The output picture will update as a result.

For each active algorithm, the user can adjust the parameters and see the results automatically. For the Anbarjafari algorithm, the goal mean parameter can be adjusted using either the slider or the text box. Note that values in the textbox are in percent. For the proposed algorithm, the aggressiveness parameter can be set in the same way. For the correction parameter, a checkbox allows the user to choose whether or not to perform manual correction. When unchecked, the histogram is automatically shifted by μ as explained in the previous Subsection. When checked, a slider and textbox allow the user to adjust the correction parameter manually. Finally, the time window reach parameter can be set in the textbox on the left.

The user can take a snapshot of the video preview, meaning that the image on the right hand side is saved to the file system as a PNG file. Once the practitioner is satisfied with the video preview result of the currently chosen algorithms and parameters, the processed footage can be saved to the file system as an AVI file. Using the textboxes below the render button, the user can optionally specify only a certain part of the video to be processed.

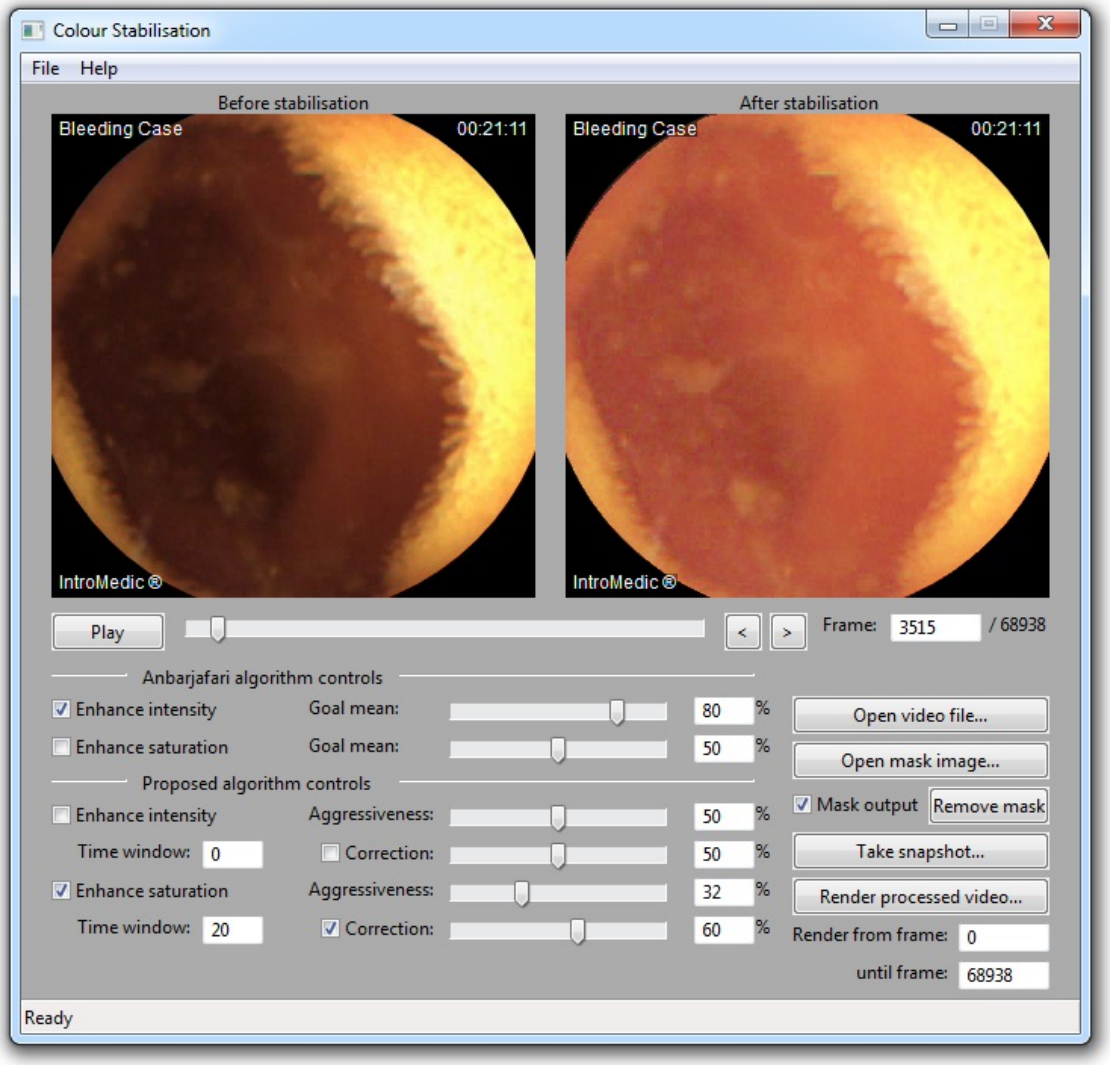


Figure 5: The software program built for experimentation, demonstration and use of the algorithms.

An important requirement is that the software can operate in real-time. The following performance results were recorded on a laptop with a 2.3 GHz Intel quadcore processor and 4 GB of RAM.

- Calculating histograms is a slow process, and this process is called every time the user seeks in the video, or changes the value of the time window reach parameter. When one of the time window parameters is set to 20, there are 41 histograms to be calculated, which takes about 3 seconds.
- Changing any of the other parameters is nearly instantaneous, taking about a tenth of a second. This is the time it takes to completely recalculate all algorithms and their parameters applied to the image.
- With algorithms applied, the video preview in the program runs smoothly at three frames per second, the real-time input video speed.
- Rendering a fragment of video of 1000 frames takes about three minutes. Considering that the input video is at three frames per second, this means that five and a half minutes of footage is processed in three minutes, which is faster than real-time.

From the above, we conclude that the requirements for the software program, both in functionality and performance speed, are satisfied.

4 Results and Discussion

In this Section, we evaluate whether the results of applying the implemented algorithms to endoscopy video satisfy the quality improvements we seek. In Subsection 4.1, we present a qualitative evaluation, where the results of the various combinations of algorithms and parameters that are possible, are observed and judged. In Subsection 4.2, a quantitative evaluation is presented, where we support several observations from the qualitative evaluation in numbers.

4.1 Qualitative evaluation

The aim of this project is to improve the brightness and colour tone of endoscopy video footage, so that it is easier for practitioners to watch and scan for medical anomalies. The nature of this is very qualitative; that is to say, it is not easy to measure how much 'better' one image is than another in numbers. In fact, it is important to note that there is no ground truth: we have no knowledge of what exactly defines the optimal image quality, what its parameters are, and how to achieve it. It is more relevant for the images to be compared and judged by human eyes, as we will do thoroughly in this user study. Because our own observations might be biased, we also conducted a survey, in which people were asked to choose the most favourable results, thus rendering the results more representative.

4.1.1 Effects of various algorithms and parameters

We have chosen the following five frames from the endoscopy video, in order to demonstrate the results of applying the algorithms:

- Frame 3515 (Figure 6), containing a dark area directly bordering a very bright area;
- Frame 4765 (Figure 8), containing a very dark area separated from a very bright area by a moderately illuminated area;
- Frame 6900 (Figure 10), containing a very dark area directly bordering bright areas on either sides;
- Frame 8096 (Figure 12), containing a dark area separated from very bright areas by moderately illuminated areas;
- Frame 32713 (Figure 14), containing a very dark area directly bordering a brightly lit area of varied colour and structure.

On the following pages, each aforementioned Figure of an original frame is followed by a Figure showing the results of processing that image with various combinations of algorithms and parameters. The rows indicate which algorithms are performed (for example, on the fifth row, the Abarjafari algorithm is used on the intensity channel and the proposed algorithm is used on the intensity histogram).

The columns indicate the parameters used for the processing. Because there is such an enormous number of possible combinations that an exhaustive demonstration is not practical, we restrict the values of the mean goal in the Anbarjafari algorithms to $\{60\%, 70\%, 80\%, 90\%, 100\%\}$ and the values of the aggressiveness in the proposed algorithms to $\{2\%, 4\%, 8\%, 16\%, 32\%\}$. The latter is chosen as such, because the parameter determines the divisor, which is not a linear function, but describes a hyperbola.

The notation $A = x$ means that if an image in that column has the Anbarjafari algorithm applied to it, the mean goal is $x\%$. The notation $P = y$ means that if an image in that column has the proposed algorithm applied to it, the aggressiveness parameter is $y\%$. For example, the following algorithms are thus performed to the image in the bottom-right corner: the Anbarjafari algorithm on the intensity channel with a goal mean of 100%, the Anbarjafari algorithm on the saturation channel with a goal mean of 100%, the proposed algorithm with an aggressiveness of 32% on the intensity histogram, and the proposed algorithm with an aggressiveness of 32% on the saturation histogram.

All other parameters are constant. The time window range for the proposed algorithms is 20 frames, and the proposed algorithms use correction with a value of 90%.

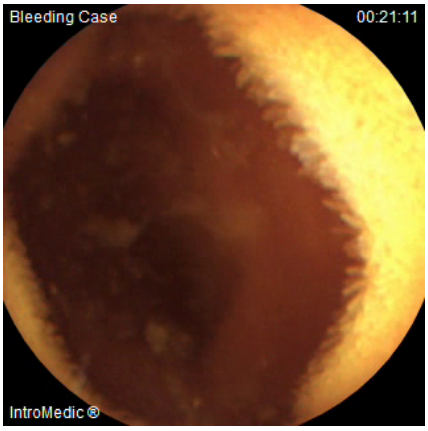


Figure 6: Original image of frame 3515.

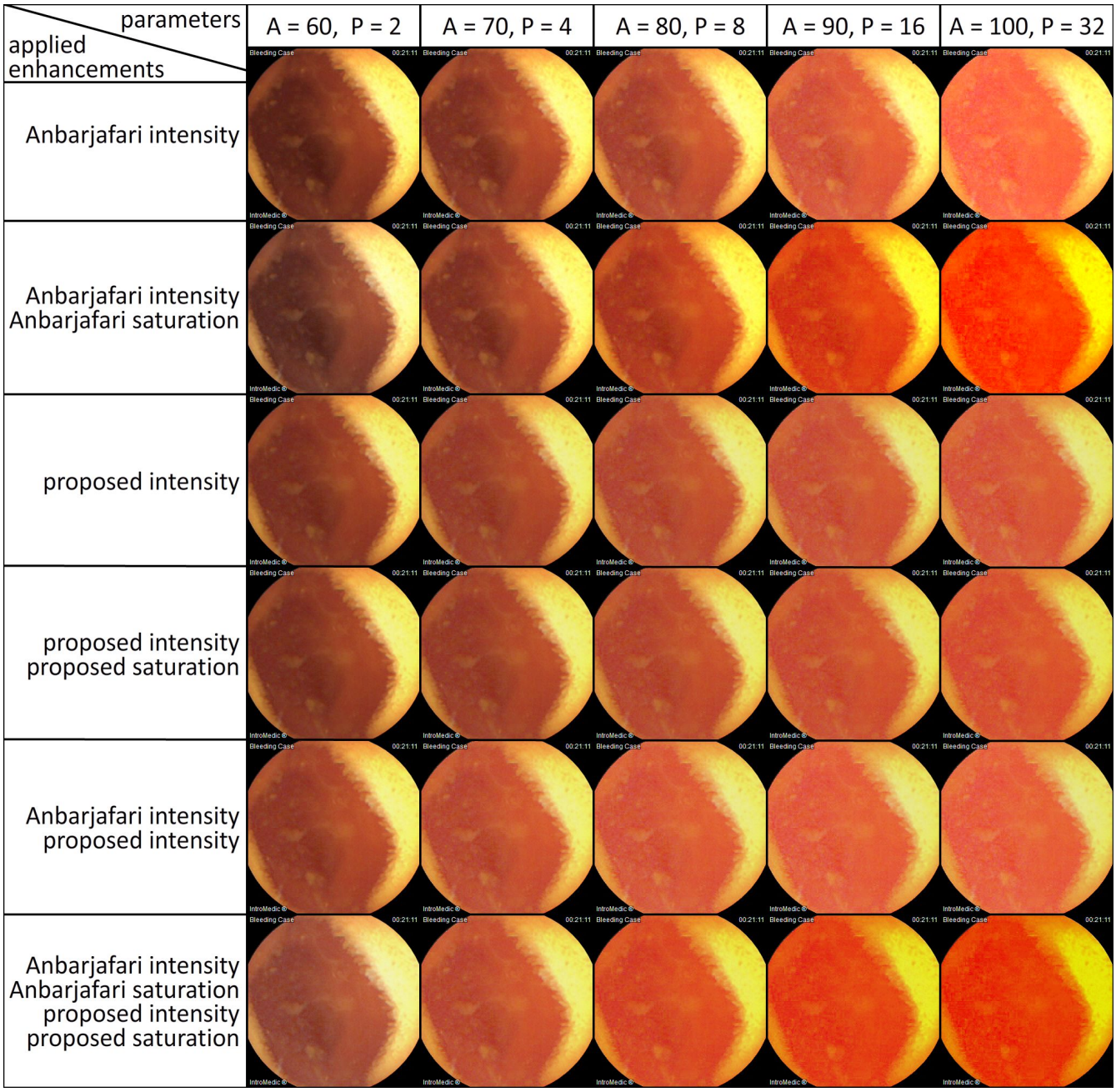


Figure 7: Frame 3515 processed with various combinations of algorithms and parameters.

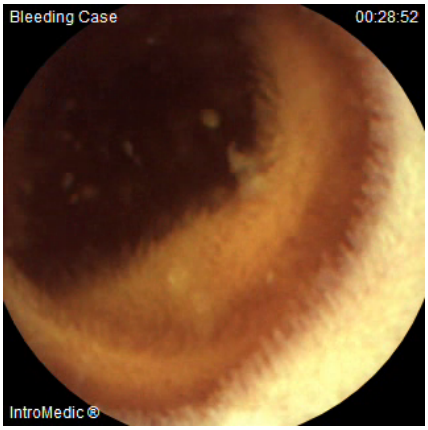


Figure 8: Original image of frame 4765.

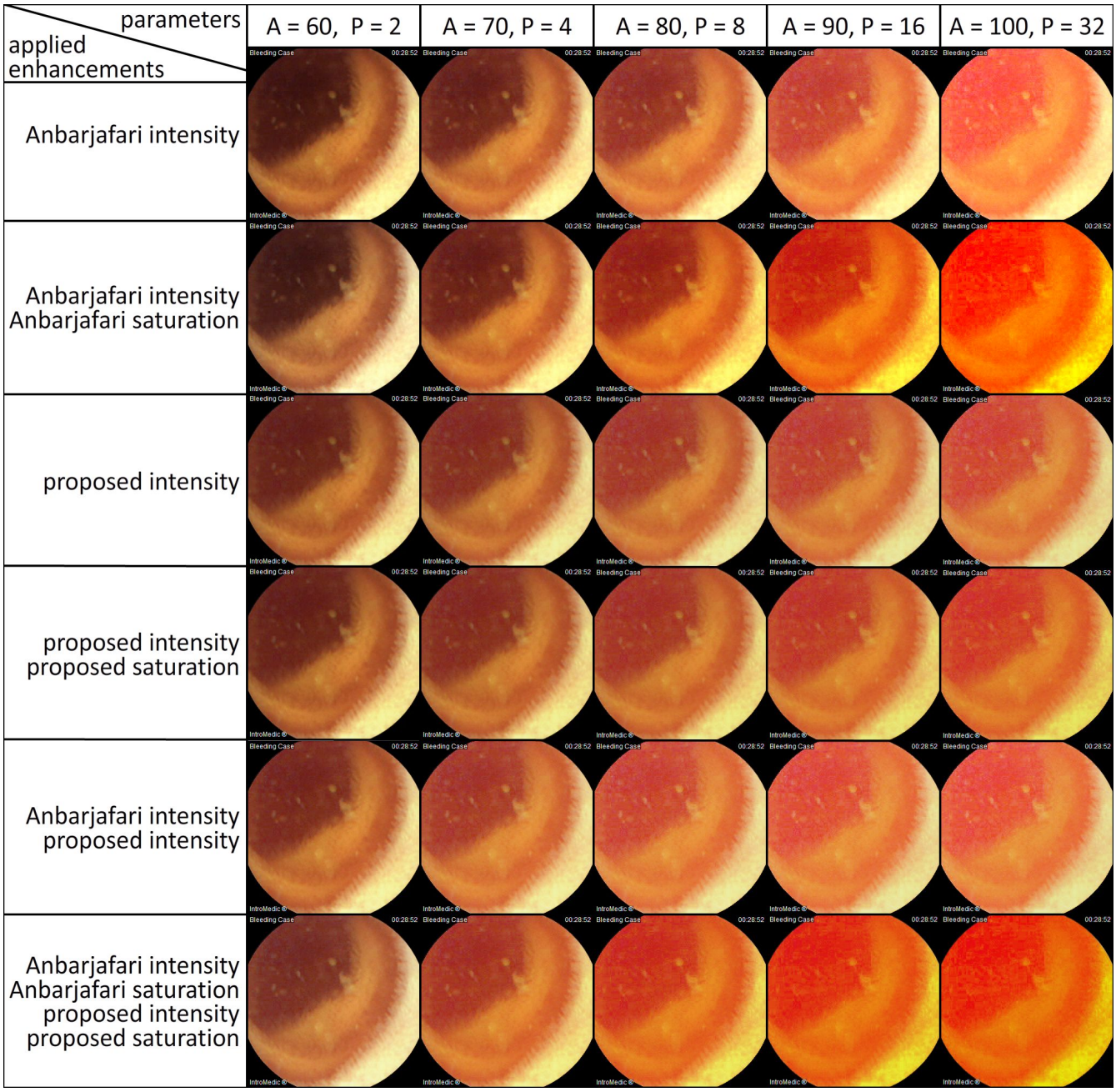


Figure 9: Frame 4765 processed with various combinations of algorithms and parameters.

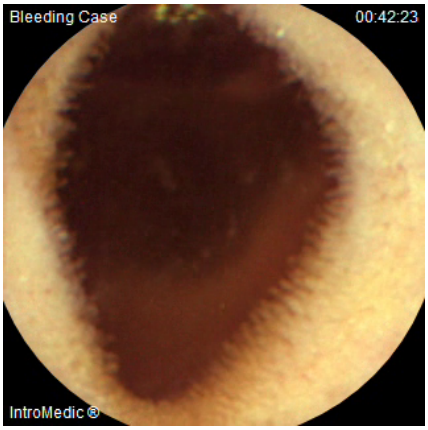


Figure 10: Original image of frame 6900.

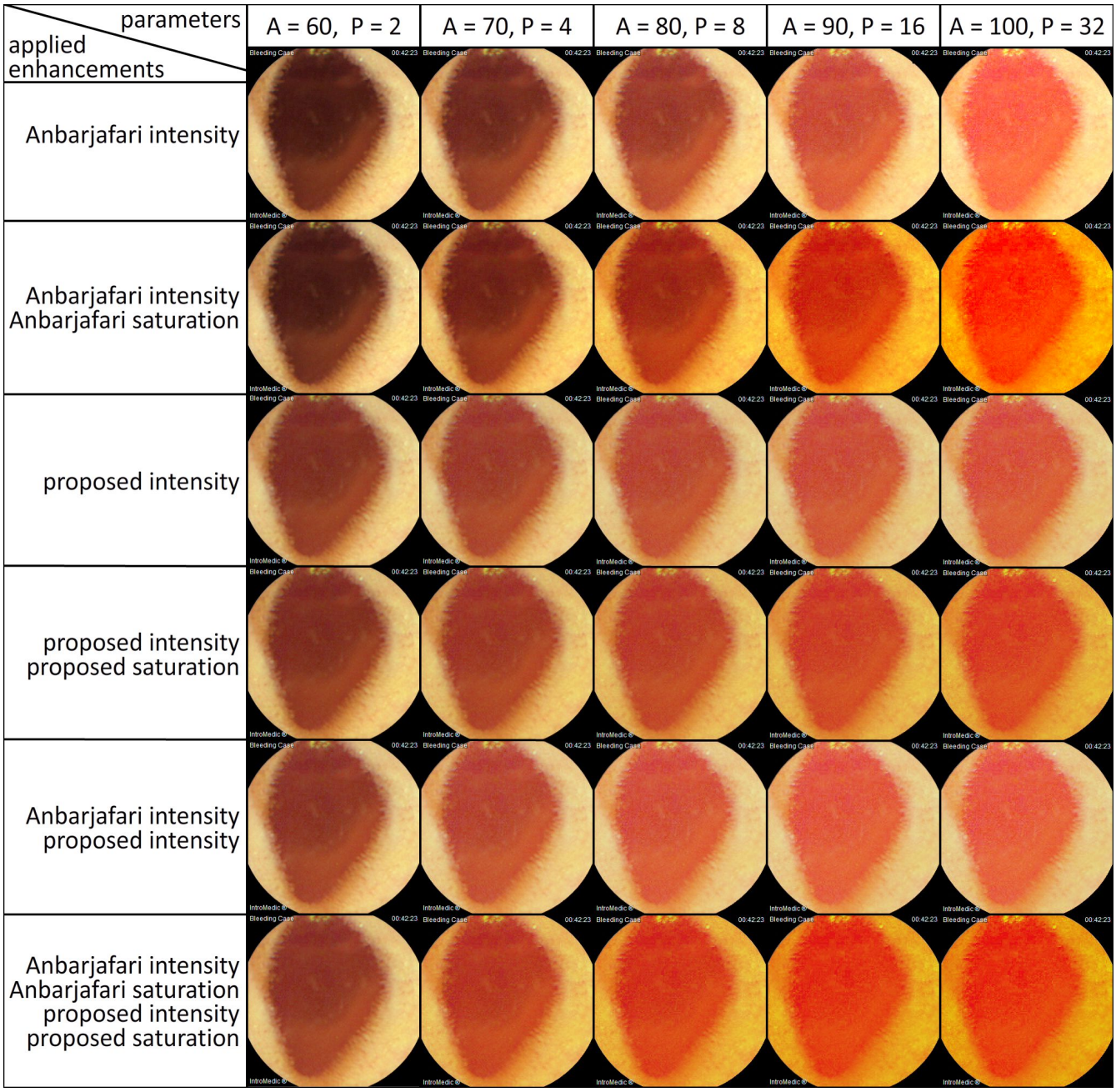


Figure 11: Frame 6900 processed with various combinations of algorithms and parameters.

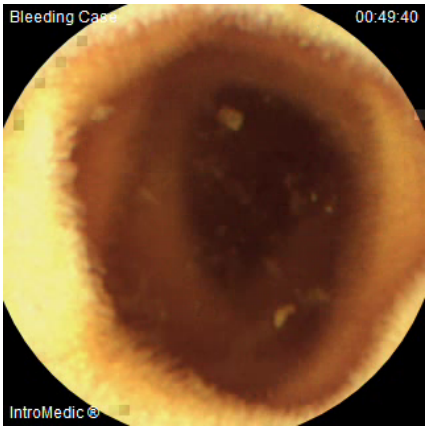


Figure 12: Original image of frame 8096.

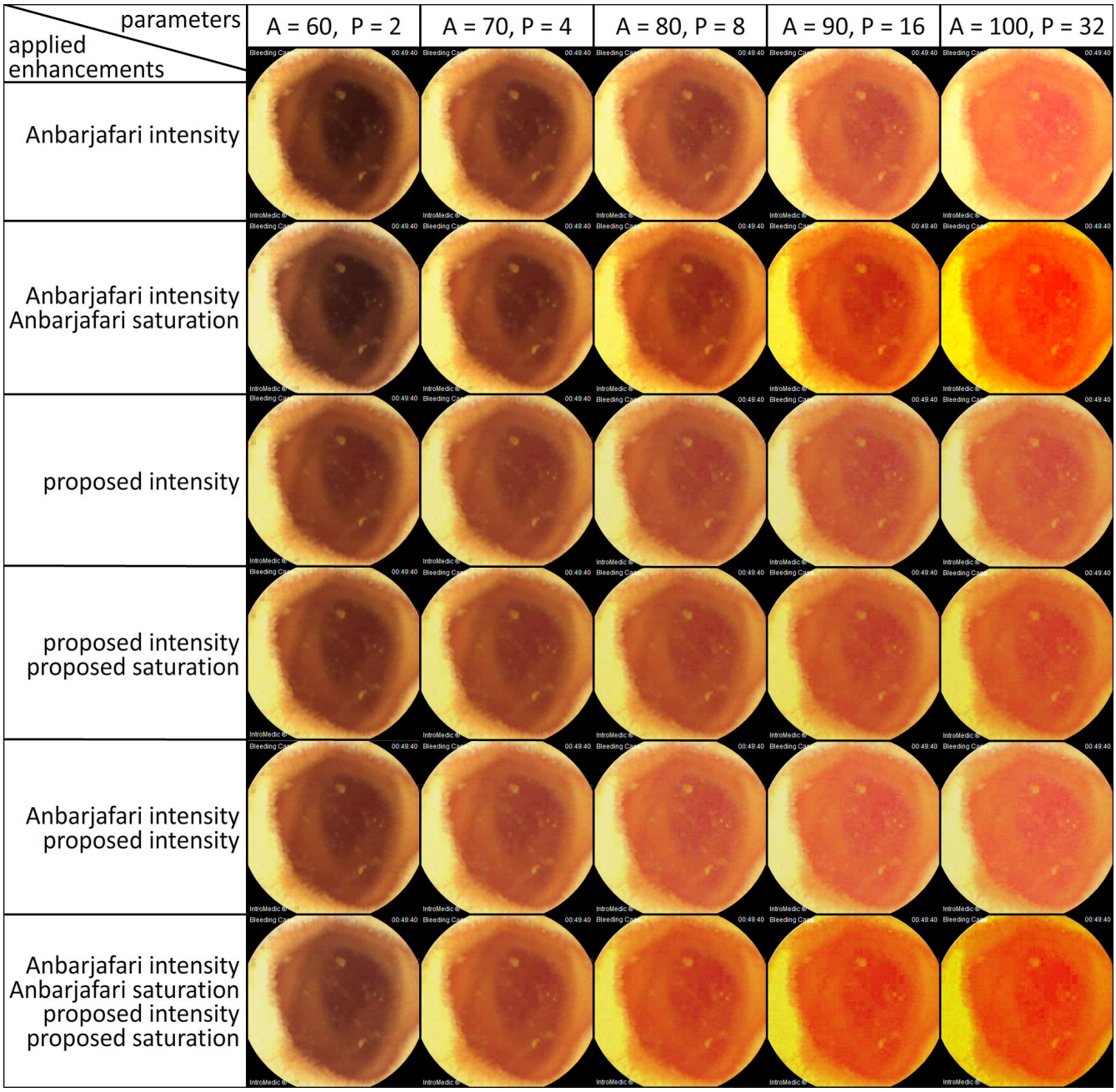


Figure 13: Frame 8096 processed with various combinations of algorithms and parameters.

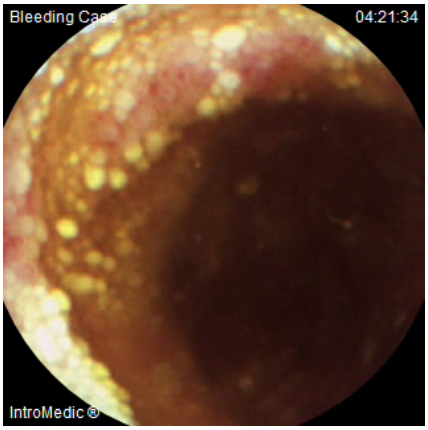


Figure 14: Original image of frame 32713.

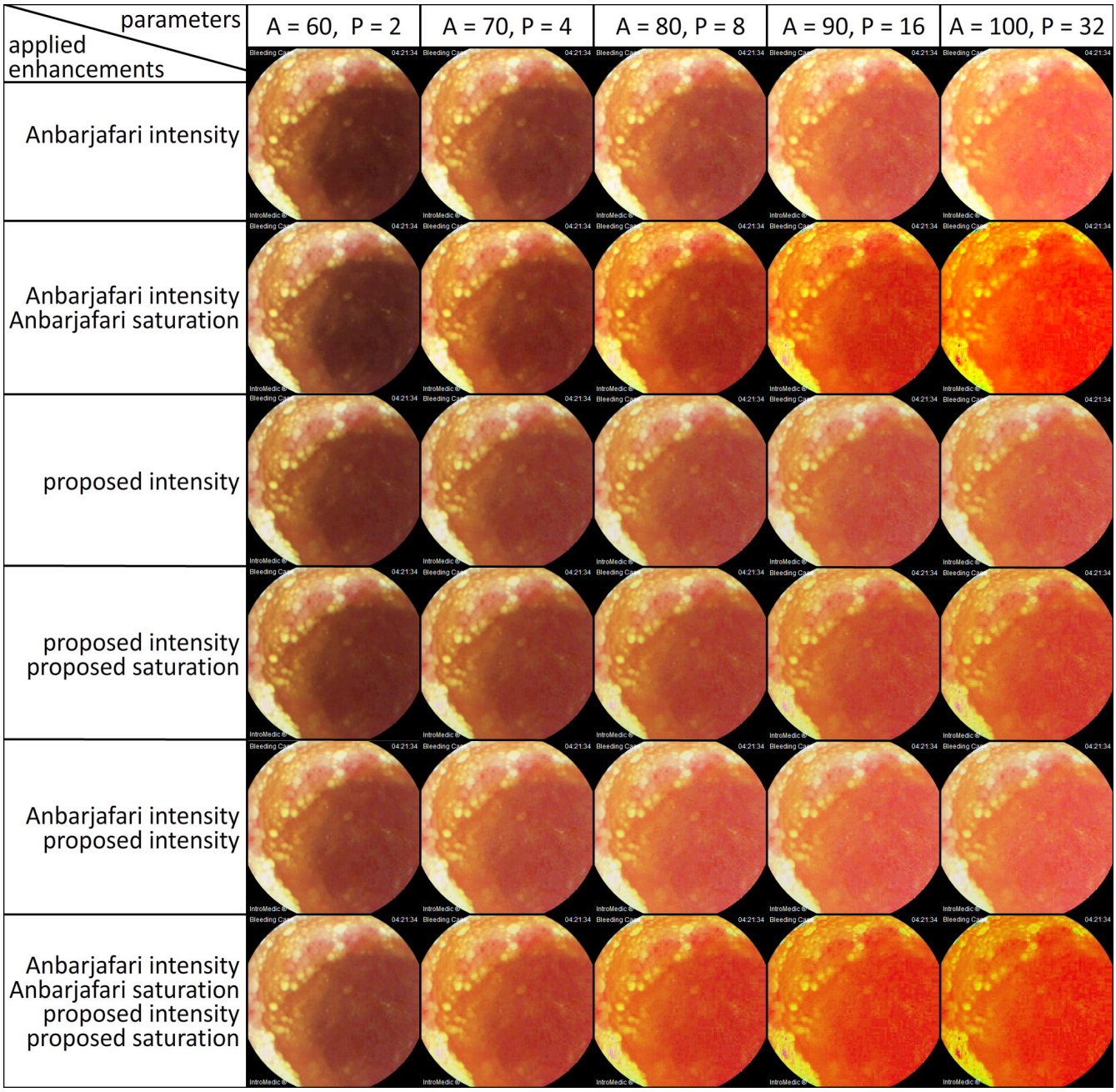


Figure 15: Frame 32713 processed with various combinations of algorithms and parameters.

For each unique combination of algorithms, i.e. the rows of the image tables, we make the following observations.

Anbarjafari algorithm applied only to the intensity channel: We observe that, as the parameter is increased, dark areas in the image are brightened, such that colours and details are more easily visible to the human eye. For all five frames, we judge that a goal mean of 80% results in the greatest increase of brightness without too much loss of information. When a goal mean greater than 80% is used, the images become too noisy, and because some detail (such as edges) are defined only by their brightness and not so much their colour, brightening those dark areas actually makes those details more difficult to see.

Anbarjafari algorithm applied to both the intensity and saturation channels: The expectation is that, like how the intensity enhancement mainly brightens dark areas, the saturation enhancement makes the colour of lowly saturated areas more pronounced. In the results, we observe that, as the parameters are increased, dark areas in the image are brightened, and the colours of the image become more saturated quite dramatically. Again, we would judge that a goal mean of 80% is the maximum for increasing the saturation so that colours are more easily visible, while a goal mean greater than 80% results in loss of colour information as the entire image becomes shaded in the dominant colour. However, it should be noted that, while the saturation enhancement can make colours easier to recognise, it may also result in a different interpretation of the contents of the image. Therefore, especially in a medical application, the saturation-enhanced image should be interpreted with great care.

Proposed algorithm applied only to the intensity channel: Similarly to the results of applying the Anbarjafari algorithm only to the intensity channel, we observe that dark areas become brighter as the aggressiveness parameter is increased. However, details in the dark areas are lost earlier than in the Anbarjafari results. We observe that with an aggressiveness of 32%, the overall brightness of the proposed algorithm's output is lower than the Anbarjafari algorithm's output with a goal mean of 100%. However, details and edge information in the proposed algorithm's output are less clearly visible at an aggressiveness of 8%, than they are in the Anbarjafari algorithm's output with a goal mean of 80%. In a way, this is understandable, because the goal of the proposed algorithm is not to enhance single images per se, but to smoothen severe and sudden changes in video. Because the proposed algorithm works by compressing the histogram of the intensity channel, the differences in intensities that may define edges or other details may become less visible or lost completely.

Proposed algorithm applied to both the intensity and saturation channels: In addition to the previously discussed effect on the intensity levels, the colours of the image become more saturated as the aggressiveness parameter increases. Interestingly, the saturation is not increased as aggressively as in the outputs of the Anbarjafari algorithm. Again, this is because the proposed algorithm does not try to increase the saturation of the image to a certain level, but is meant to smoothen out sudden differences in the saturation histograms compared to neighbouring frames. This is why, as we will present later, the proposed algorithm for saturation enhancement is better for enhancing video rather than individual images.

Anbarjafari algorithm and proposed algorithm both applied to only the intensity channel: In all tables, we observe that these results are nearly identical to those of applying the proposed algorithm to only the intensity channel. We explain this by the fact that the proposed algorithm compresses the histogram after the Anbarjafari algorithm enhanced the brightness, which for the most part undoes the enhancements the Anbarjafari algorithm made. As a result, the output images suffer from the same problems we observed when only applying the proposed algorithm to the intensity channel: loss of details due to the compression of the histogram.

Anbarjafari algorithm and proposed algorithm both applied to both the intensity and saturation channels: In all tables, we observe that these results are very similar to those of applying the Anbarjafari algorithm to both the intensity and saturation channels, but the saturation is less extremely increased. We explain this by the fact that, after the Anbarjafari algorithm has made the saturation extremely high, the proposed algorithm then compresses the saturation histogram, making the vibrance of the colours less extreme.

From these observations, we make the following evaluation. Because the Anbarjafari algorithm is designed to enhance single images while the proposed algorithm is designed to smoothen sudden changes in video, when applied to individual images the Anbarjafari algorithm results in more favourable output. Furthermore, while saturation enhancement in the Anbarjafari algorithm can make colours more apparent to the human eye, it may introduce false colour information and does not smoothen sudden changes in the saturation in video. As a result, we consider saturation enhancement to be a task better performed by the proposed algorithm, while intensity enhancement is better performed by the Anbarjafari algorithm. Furthermore, when the goal mean for the intensity channel is too high, the enhancement becomes too extreme, resulting in noise, and loss of details and information in the image. Therefore, we would recommend enhancing separate images by applying the Anbarjafari algorithm on the intensity channel with a goal mean no greater than 80%.

4.1.2 Survey

Because the qualitative evaluation is subjective and could therefore be influenced by our own bias, we conducted an online survey in order to compile an analysis by a wider group of people, thus obtaining a more representative evaluation. This survey was conducted as follows:

The survey consisted of five pages, with each page containing all processed images for a certain frame, as seen in the Figures on the previous pages. Each page contained six image sets, corresponding to the rows in the aforementioned Figures. Each image set contained six images: the original image, followed by the five outputs of processing the image with various outputs, corresponding to the columns of the aforementioned Figures. Thus, one page contains 36 images.

For each image set, the participant was asked to pick the image that they thought was the best, in terms of enhancing the information in the brighter and darker areas of the image, without introducing too much noise or losing information. Then, at the end of each page, the participant was asked to review the six images they picked, and to pick the best one among these, resulting in a best-of-the-best image.

The survey was conducted using Google Forms. Participants were encouraged to look at each of the 30 image sets for roughly 10 seconds, so that the survey could be finished in about 5 minutes. However, the participants could spend any time they desired, and were allowed to go back to previous pages to review or change their answers. Note that participants saw only the images, without mention of which algorithms and parameters were applied to which. The exact survey instructions can be found in Appendix C.

18 people participated in the survey. These participants are employees of ZiuZ Visual Intelligence, working in the fields of image processing and computer science. The demographics are between the ages of 20 and 50, and the majority is male. Even though participants were invited too send any textual feedback or comments on the survey or the images, none chose to do so.

The first five tables on the next page present the results as follows: For each image, the rows and columns of the table indicate the images participants could vote for, laid out the same way as in the Figures on the previous pages. Each cell contains two numbers, separated by a slash. The number before the slash indicates how many times an image was chosen as the best in an image set (i.e. best of 6), and the number after the slash and printed in boldface indicates how many times an image was chosen as best of the page (i.e. best of 36). Notice that the numbers in boldface do not necessarily add up to the number of participants. This is because several images on a page may look identical, and if this image is chosen as the best, each of the instances gets a point.

The winner algorithm is decided for each of the five input frames, by seeing which individual processed image is chosen as best-of-the-best most often. From the first five tables on the next page, we see that the winners are as follows:

- For frame 3515, the Anbarjafari algorithm applied to the intensity and saturation channels with a mean goal of 70% is the winner with 5 votes.
- For frame 4765, there is a tie between the Anbarjafari algorithm on intensity and saturation with a goal mean of 80%, and the proposed algorithm on intensity with aggressiveness 4. We break this tie by considering the total number of best-of-the-bests per row. Anbarjafari on intensity and saturation has a total of 6, while proposed on intensity has 5, so we declare Anbarjafari the winner.
- For frame 6900, the Anbarjafari algorithm applied to the intensity channel with a mean goal of 70% is the winner with 6 votes.
- For frame 8096, the Anbarjafari algorithm on the intensity and saturation channels with a mean goal of 70% is the winner with 4 votes.
- For frame 32713, the Anbarjafari algorithm on the intensity channel with a mean goal of 60% is the winner with 4 votes.

From these results, we conclude that the majority of participants consider the Anbarjafari algorithm to produce the best results. Depending on the content of the image, the Anbarjafari algorithm should be applied to only the intensity channel or both the intensity and the saturation channels, with mean goals in the range of 60% to 80%.

The images that were processed with parameters $A = 90, P = 16$ and $A = 100, P = 32$ were almost never chosen as best-of-the-best, indicating that the manipulation of these images is too aggressive, thus resulting in too much noise or loss of information.

A table on the bottom of the next page shows the numbers of votes accumulated for all five frames. After all, our goal is to find a set of parameters that yields good results for any input frame, and because the relation between algorithm / parameter choice and output image quality appears quite consistent so far, it should also make sense that the survey votes are consistent. In the accumulative table, two choices are clear outliers in the number of votes for best-of-the-best results:

- The Anbarjafari algorithm on the intensity and saturation channels with a mean goal of 70%, with 17 votes.
- The Anbarjafari algorithm applied to the intensity channel with a mean goal of 70%, with 12 votes.

This supports the earlier observation that the Anbarjafari algorithm is most favoured for the enhancement of individual images.

Survey results for frame 3515:

	original	A=60,P=2	A=70, P=4	A=80, P=8	A=90, P=16	A=100, P=32
Anb. I	1 / **1**	5 / **2**	9 / **0**	3 / **0**	0 / **0**	0 / **0**
Anb. I, Anb. S	0 / **0**	5 / **2**	13 / **5**	0 / **0**	0 / **0**	0 / **0**
prp. I	2 / **1**	9 / **0**	6 / **2**	0 / **0**	1 / **1**	0 / **0**
prp. I, prp. S	1 / **1**	8 / **2**	7 / **1**	2 / **0**	0 / **0**	0 / **0**
Anb. I, prp. I	1 / **1**	12 / **2**	4 / **0**	1 / **0**	0 / **0**	0 / **0**
Anb. I, Anb. S, prp. I, prp. S	1 / **1**	13 / **2**	4 / **0**	0 / **0**	0 / **0**	0 / **0**

Survey results for frame 4765:

	original	A=60,P=2	A=70, P=4	A=80, P=8	A=90, P=16	A=100, P=32
Anb. I	1 / **1**	1 / **1**	7 / **1**	8 / **1**	1 / **0**	0 / **0**
Anb. I, Anb. S	1 / **1**	0 / **0**	12 / **2**	5 / **3**	0 / **0**	0 / **0**
prp. I	2 / **1**	3 / **1**	7 / **3**	5 / **0**	1 / **0**	0 / **0**
prp. I, prp. S	1 / **1**	4 / **0**	8 / **0**	3 / **1**	2 / **0**	0 / **0**
Anb. I, prp. I	1 / **1**	6 / **1**	4 / **1**	7 / **0**	0 / **0**	0 / **0**
Anb. I, Anb. S, prp. I, prp. S	1 / **1**	6 / **0**	11 / **2**	0 / **0**	0 / **0**	0 / **0**

Survey results for frame 6900:

	original	A=60,P=2	A=70, P=4	A=80, P=8	A=90, P=16	A=100, P=32
Anb. I	1 / **1**	0 / **0**	11 / **6**	6 / **1**	0 / **0**	0 / **0**
Anb. I, Anb. S	1 / **1**	2 / **0**	10 / **3**	5 / **0**	0 / **0**	0 / **0**
prp. I	2 / **1**	11 / **2**	4 / **0**	1 / **0**	0 / **0**	0 / **0**
prp. I, prp. S	2 / **1**	9 / **1**	7 / **0**	0 / **0**	0 / **0**	0 / **0**
Anb. I, prp. I	2 / **1**	10 / **0**	5 / **1**	1 / **1**	0 / **0**	0 / **0**
Anb. I, Anb. S, prp. I, prp. S	1 / **1**	15 / **3**	2 / **1**	0 / **0**	0 / **0**	0 / **0**

Survey results for frame 8096:

	original	A=60,P=2	A=70, P=4	A=80, P=8	A=90, P=16	A=100, P=32
Anb. I	1 / **1**	3 / **0**	8 / **3**	6 / **1**	0 / **0**	0 / **0**
Anb. I, Anb. S	1 / **1**	2 / **0**	12 / **4**	3 / **0**	0 / **0**	0 / **0**
prp. I	3 / **2**	7 / **0**	7 / **0**	1 / **0**	0 / **0**	0 / **0**
prp. I, prp. S	3 / **2**	9 / **2**	5 / **0**	1 / **0**	0 / **0**	0 / **0**
Anb. I, prp. I	2 / **2**	10 / **3**	6 / **1**	0 / **0**	0 / **0**	0 / **0**
Anb. I, Anb. S, prp. I, prp. S	2 / **2**	11 / **2**	5 / **2**	0 / **0**	0 / **0**	0 / **0**

Survey results for frame 32713:

	original	A=60,P=2	A=70, P=4	A=80, P=8	A=90, P=16	A=100, P=32
Anb. I	2 / **2**	9 / **4**	5 / **2**	1 / **0**	1 / **0**	0 / **0**
Anb. I, Anb. S	2 / **2**	5 / **3**	8 / **3**	3 / **0**	0 / **0**	0 / **0**
prp. I	2 / **2**	11 / **1**	4 / **1**	0 / **0**	1 / **1**	0 / **0**
prp. I, prp. S	2 / **2**	11 / **2**	3 / **1**	2 / **0**	0 / **0**	0 / **0**
Anb. I, prp. I	2 / **2**	12 / **0**	2 / **0**	2 / **0**	0 / **0**	0 / **0**
Anb. I, Anb. S, prp. I, prp. S	3 / **2**	12 / **0**	1 / **1**	2 / **0**	0 / **0**	0 / **0**

Survey results accumulated for all frames:

	original	A=60,P=2	A=70, P=4	A=80, P=8	A=90, P=16	A=100, P=32
Anb. I	6 / **6**	18 / **7**	40 / **12**	24 / **3**	2 / **0**	0 / **0**
Anb. I, Anb. S	5 / **5**	14 / **5**	55 / **17**	16 / **3**	0 / **0**	0 / **0**
prp. I	11 / **7**	41 / **4**	28 / **6**	7 / **0**	3 / **2**	0 / **0**
prp. I, prp. S	9 / **7**	41 / **7**	30 / **2**	8 / **1**	2 / **0**	0 / **0**
Anb. I, prp. I	8 / **7**	50 / **6**	21 / **3**	11 / **1**	0 / **0**	0 / **0**
Anb. I, Anb. S, prp. I, prp. S	8 / **7**	57 / **7**	23 / **6**	2 / **0**	0 / **0**	0 / **0**

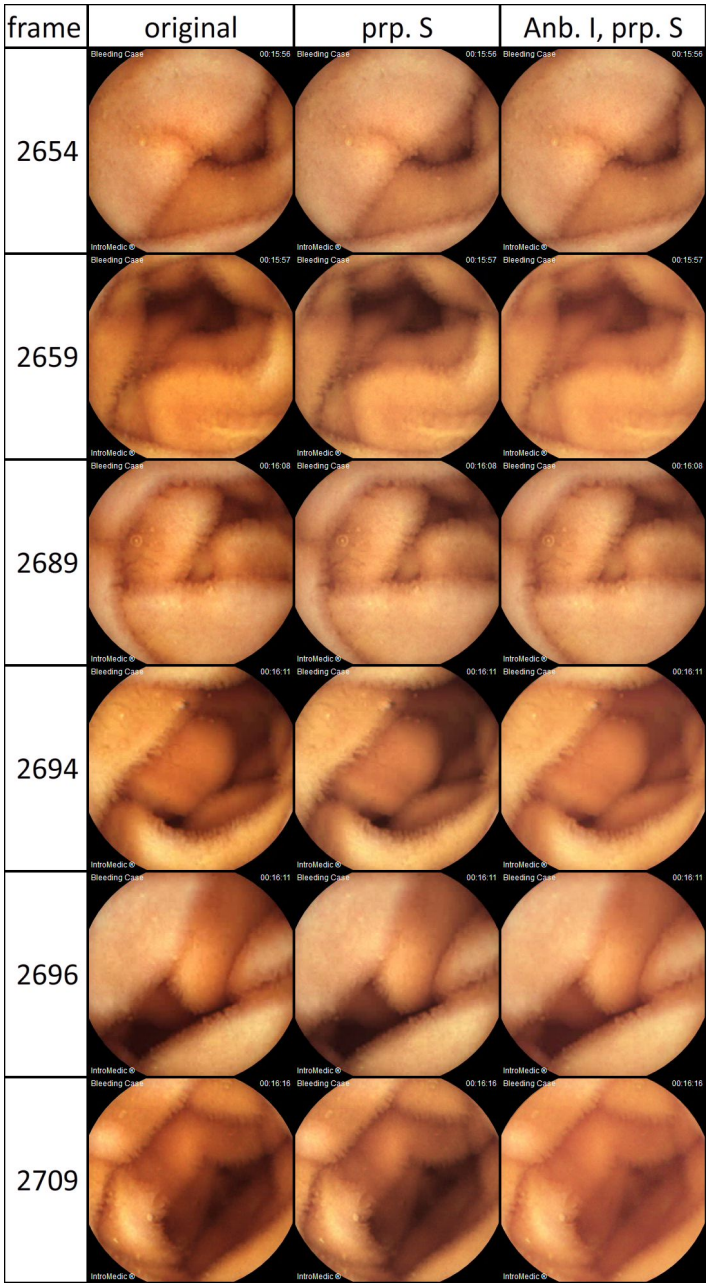


Figure 16: Selected frames from a short clip of video, demonstrating the result of using the proposed algorithm to smoothen sudden changes of colour tone.

4.1.3 Effects on multiple frames

With the results from the survey, we conclude that the Anbarjafari algorithm is powerful against the problem of stark brightness differences in endoscopy video frames. For the problem of colour fluctuations throughout time, we proposed a new algorithm earlier.

Figure 16 shows a selection of frames from a clip of the bleeding case video, between frame 2654 and frame 2709. In the left column, frames of the original video are displayed. Recall that these frames were also shown in Section 1, to introduce the problem of colour tone fluctuations.

In order to resolve this issue, we apply the proposed algorithm to the saturation histograms of this video clip. We choose a frame window reach k of 40, which should be wide enough to smoothen out sudden changes in colour tone. The aggressiveness parameter a is 32, and the correction parameter c is 40. The results are presented in the middle column of Figure 16.

In the middle column, we observe that the sudden changes in colour tone have indeed been smoothened out. All frames have a pink-ish colour tone, and when watching the video clip, the issue of distracting changes is resolved.

Lastly, we demonstrate in the right column of Figure 16 how the proposed algorithm can be combined with the Anbarjafari algorithm applied to the intensity channels of the individual frames, in order to brighten the dark areas seen on some frames. The used parameter is a goal mean for the intensity channel of 80%. While this video does not have particularly large dark areas, we judge it a successful combination of the two algorithms.

4.1.4 Conclusions of qualitative analysis

The strong point of the Anjarbafari algorithm, besides its speed and simplicity, is that it is very good at brightening dark areas of an image without overly exposing areas that are already bright enough. It is not very useful for saturation stabilisation, because it only works on one frame at a time, without knowing anything about any other frames. The weak point of the algorithm is that some edge information and detail is partially lost during intensity correction, although the loss is relatively small.

The strong point of the proposed algorithm is its ability to smoothen the saturation of video transitions throughout time, thus removing bumps or sudden fluctuations in the colour tone of the video. The weak point of the algorithm is that is has not proven useful for intensity correction, because too much valuable information such as edges, details and significantly different colours, are lost in the process.

4.2 Quantitative evaluation

During the qualitative evaluation of the various possible combinations of algorithms and parameters (Subsections 4.1.1 and 4.1.2), we observed that the relation between algorithm/parameter choice and output image quality was fairly consistent, across the five input images. We observed that the image manipulation would consistently get too extreme with very high parameter values, that the proposed algorithm caused slightly washed out colours when compared to the Anbarjafari algorithm's output, and that survey participants tended to prefer the quality of images in the top-left region of the tables of choices (i.e. output images where only the Anbarjafari algorithm was applied with a moderately high mean goal parameter value).

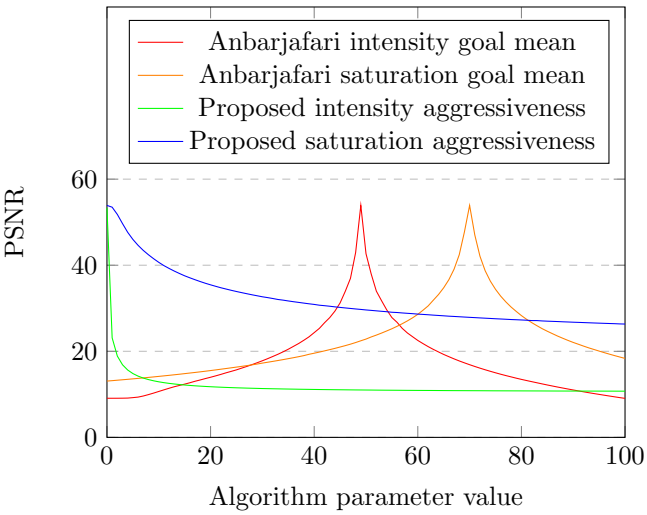
A main goal of this project is to find the combination(s) of algorithms and parameter values which provide the best results. Consistency is very important, because if a certain set of options works well for one image, we would like it to work well on other images too. We can evaluate the consistency of the algorithm results with a quantitative measurement.

In image processing, quantitative evaluation is usually performed using similarity measurements, in order to express how similar a processed image is to its original. This is especially useful information when comparing different methods of processing. We will use the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) methods in order to measure the similarity between the original image and a processed image.

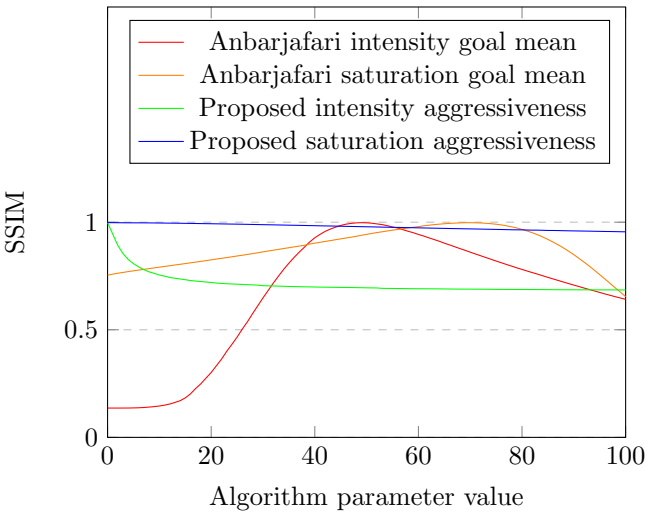
The following figures show, for the five selected frames, the plots of the similarity measurements PSNR and SSIM as a function of the parameter value in the range $[0, 100]$. Each figure shows the following four plots:

1. The similarity measurement as a function of the goal mean for the intensity channel using the Anbarjafari algorithm;
2. The similarity measurement as a function of the goal mean for the saturation channel using the Anbarjafari algorithm;
3. The similarity measurement as a function of the aggressiveness of the intensity histogram compression using the proposed algorithm, with a time window width of 20 and a correction of 90;
4. The similarity measurement as a function of the aggressiveness of the saturation histogram compression using the proposed algorithm, with a time window width of 20 and a correction of 90.

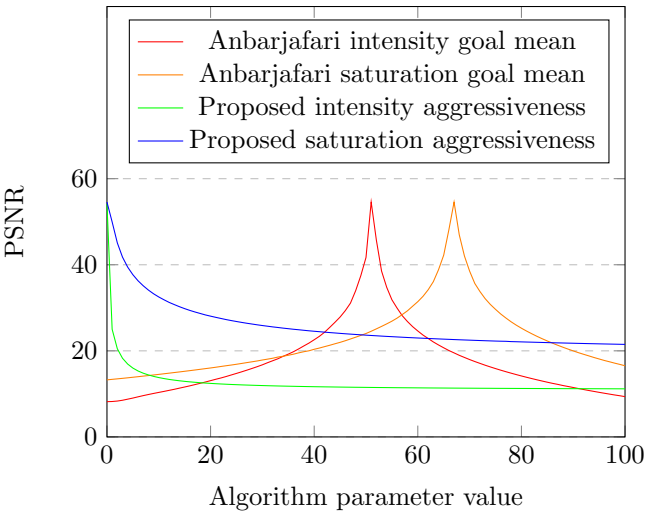
PSNR of frame 3515 with various algorithms



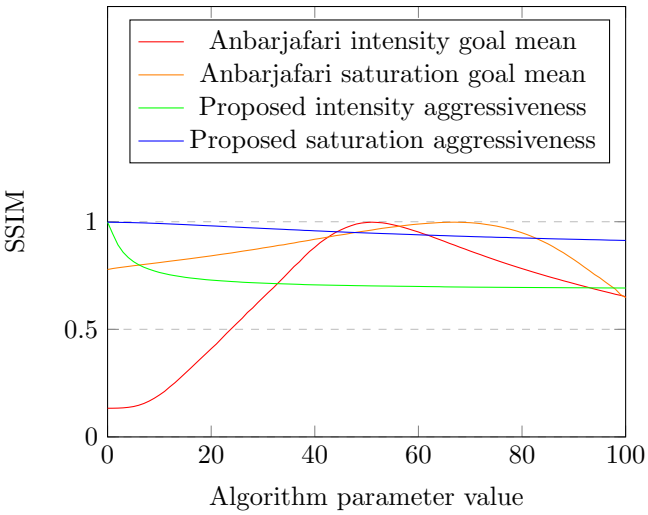
SSIM of frame 3515 with various algorithms



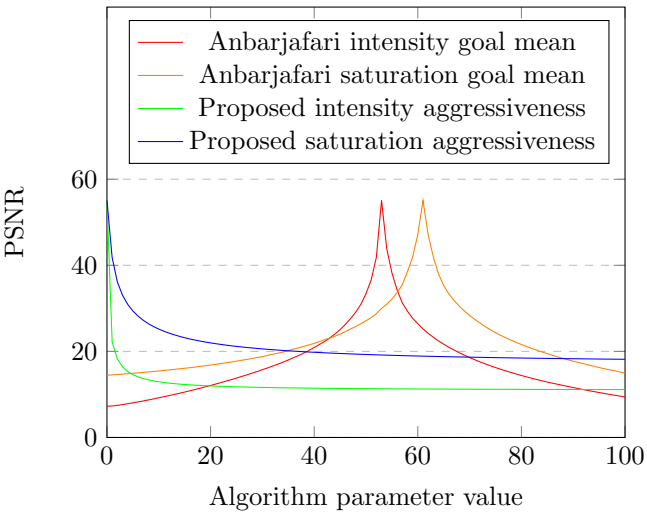
PSNR of frame 4765 with various algorithms



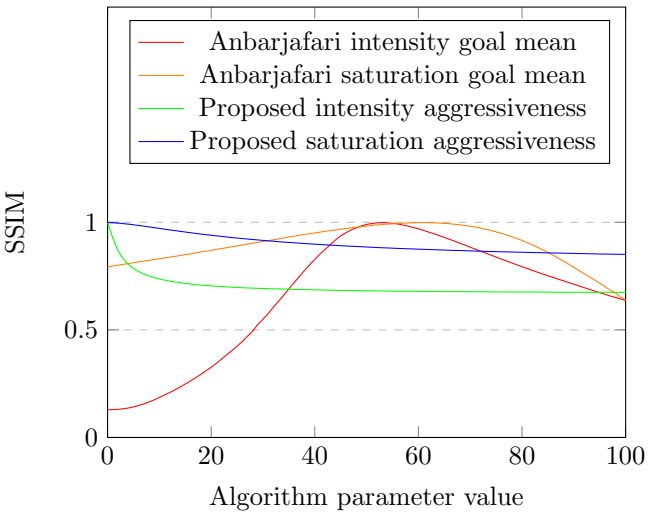
SSIM of frame 4765 with various algorithms

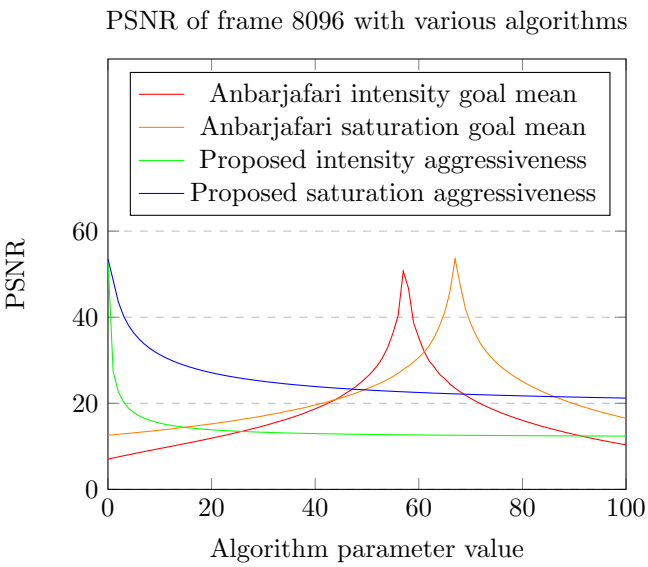


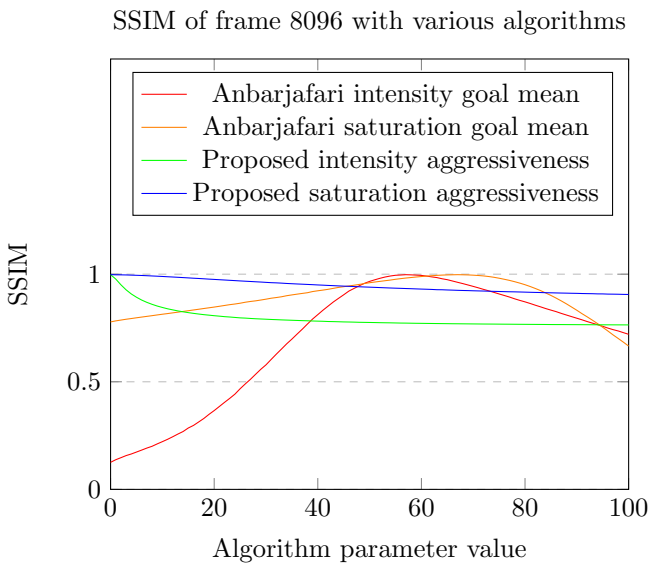
PSNR of frame 6900 with various algorithms

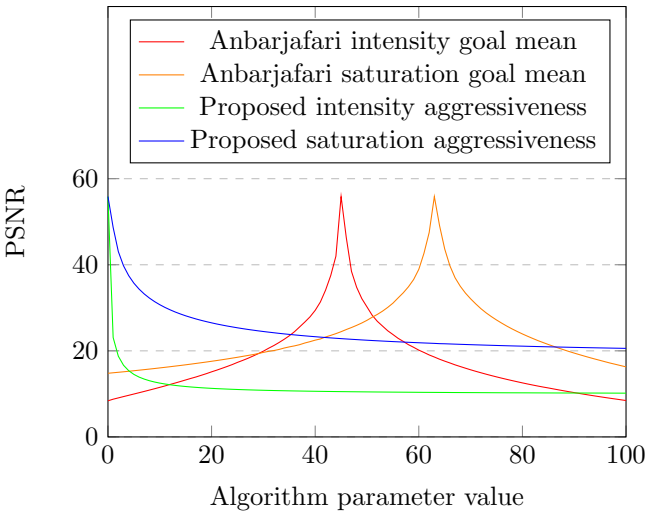


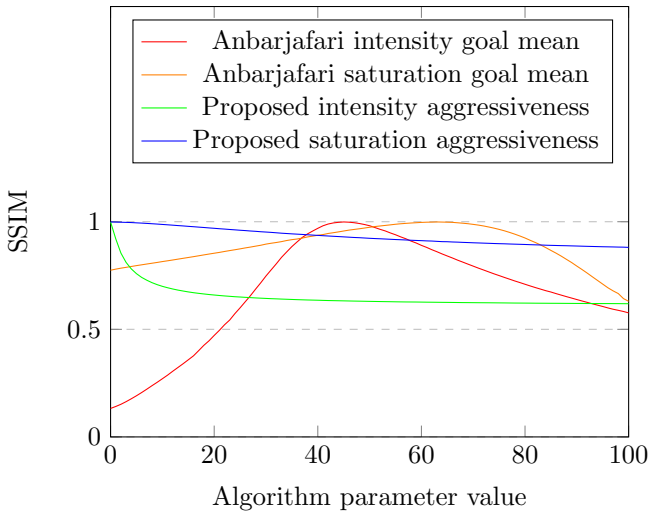
SSIM of frame 6900 with various algorithms




PSNR of frame 8096 with various algorithms
Anbarjafari intensity goal mean
Anbarjafari saturation goal mean
Proposed intensity aggressiveness
Proposed saturation aggressiveness
PSNR
0
20
40
60
0
20
40
60
80
100
Algorithm parameter value


SSIM of frame 8096 with various algorithms
Anbarjafari intensity goal mean
Anbarjafari saturation goal mean
Proposed intensity aggressiveness
Proposed saturation aggressiveness
SSIM
0
0.5
1
0
20
40
60
80
100
Algorithm parameter value


PSNR of frame 32713 with various algorithms
Anbarjafari intensity goal mean
Anbarjafari saturation goal mean
Proposed intensity aggressiveness
Proposed saturation aggressiveness
PSNR
0
20
40
60
0
20
40
60
80
100
Algorithm parameter value


SSIM of frame 32713 with various algorithms
Anbarjafari intensity goal mean
Anbarjafari saturation goal mean
Proposed intensity aggressiveness
Proposed saturation aggressiveness
SSIM
0
0.5
1
0
20
40
60
80
100
Algorithm parameter value

From these plots, we make the following observations:

- The Anbarjafari algorithm plots show peaks at the intensity/saturation channel means of the input image. This is logical, because if the goal mean equals the original mean, no correction needs to be done, so the processed image is no different from the input image.
- The proposed algorithm plots show peaks at parameter value 0. This is logical, because when the aggressiveness is zero, the histogram is not compressed at all, and no correction occurs, so the processed image is identical to the input image.
- For every frame, the lowest similarity is measured when the Anbarjafari intensity parameter nears 0. In this case, the algorithm must manipulate the image such that it has a mean intensity of (nearing) zero, which is total darkness. Because a fully black image has no recognisable structures, it is logical that there are no similarities to be found.
- In the Anbarjafari algorithm, the plots of the intensity and the saturation enhancements are very similar in shape and magnitude, apart from the obvious dip towards zero when the intensity goal mean nears zero. However, in the proposed algorithm, the plot for the saturation is always a magnitude larger than that of the intensity. This means that compressing the intensity histogram has a larger effect on the similarity than compressing the saturation histogram, when using the same compression rates.
- The plots of the proposed algorithm functions are quicker to reach a 'stable' value as the aggressiveness parameter goes from 0 to 100 (indicated by the plot being a nearly horizontal line), while the plots of the Anbarjafari algorithm functions move towards and away from the peaks at steeper inclines. The reason for this is in the differences between what the parameters control. In the Anbarjafari algorithm, the parameter determines what the mean value of a certain channel of the output image should become, and moves linearly between the values of 0% and 100%. In the proposed algorithm, the parameter determines the divisor that the width of the histogram should be divided by. Division does not result in a linear function, but one shaped by a hyperbola, which is why the values $2, 4, 8, 16, 32$ were used during the earlier experiments.
- Across the five frames, each plot for the same algorithm is very consistently shaped and positioned, which supports our earlier observation of a consistent relation between algorithm/parameter choice and output image quality. Because the same algorithm/parameter choice can be applied to different frames with a consistent quality in output imagery, we can say that the methods are fairly robust.
- The plots themselves are smoothly shaped and curved, as opposed to being jittery with random spikes or noise. This is good, because it means that changing the parameters has no sudden, random or unexpected results. This is also an indication of the robustness of the algorithms.

From these observations, we conclude that our earlier suspicion of consistency between algorithm/parameter choice and output image quality is supported. Furthermore, several aspects indicate that the algorithms are robust, without unexpected effects when adjusting the parameters.

5 Conclusions

In this thesis, we have researched and implemented an existing algorithm for enhancing the clarity of images, and proposed a new algorithm for stabilising the colour tone of video using histogram compression. The aim is to use these algorithms to make it easier to see details in capsule endoscopy video, by brightening dark areas and removing distracting sudden changes in colour tone. In order to easily apply the algorithms to video, we developed a user interface program. What follows is a review of the original goals and requirements, and in what proximity we achieved them.

Research question 1: How to brighten dark areas and darken bright areas?
After a literature study, we found the Anbarjafari algorithm as a very powerful and lightweight method for enhancing images with large brightness differences. After a multi-fauceted evaluation, consisting of personal qualitative review, a survey, and quantitative measurement, we conclude that the Anbarjafari algorithm is the favoured and robust method for brightening dark areas in endoscopy video.

Research question 2: How to develop a robust algorithm to normalise colour and brightness in video?
We proposed an algorithm that has shown to successfully smoothen out colour tone fluctuations in video. It operates by measuring the variance in the sum of the histograms within a time window of frames, and compressing the image histograms proportionally, in order to eliminate stark colour tone differences. In a quantitative analysis, the algorithm is shown to be robust, showing no sudden or unexpected effects of adjusting its parameters.

Research question 3: Which techniques should be used to perform stabilisation without damages?
After performing qualitative evaluation and conducting a survey among 18 participants, the results of using various combinations of algorithms and parameters has been thoroughly investigated. For correcting brightness differences in individual frames, the best results are obtained using the Anbarjafari algorithm on the intensity channel, with a goal mean between 60% and 80%. For smoothening out sudden colour fluctuation in video, the proposed algorithm is used to compress the saturation histogram depending on the histograms of neighbouring frames. This was demonstrated to be successful with a time window width of 40, an aggressiveness of 32% and a correction of 40%.

The algorithms can be demonstrated and used in a program we built. The requirements for this software were video file import, movie playback and seeking functionality, selection and adjustment of algorithms and parameters to use, and preview and export of processed video. These functionalities were all successfully implemented. A practical requirement was that video could be previewed and rendered in real-time. Considering that the endoscopy video has a frame rate of three frames per second, this requirement is easily met on modern machines.

5.1 Limitations

The following is a list of aspects where research was limited or information was unknown, with the possible consequence that the conclusions above do not take everything into consideration.

- Confidence in the user study. No contact, feedback or other comments were received from the survey participants. While this was entirely optional, the lack of any textual reaction means that we have no knowledge of whether participants fully understood why or how to fill in the survey. We can only assume that the participants were well competent because they are employees of a computer vision company.
- Studied material. In order to focus time and resources on comparing different algorithms and parameters, a single input video was taken as a constant. We have therefore not evaluated whether using the algorithms on other video files results in surprising or undesirable results. We can only assume that, because of the similarity in the subject matter of endoscopy videos, that using different videos as input files would result in similar quality enhancements.
- The proposed algorithm. Its mathematics are simply designed to translate the intuitive description of shifting and compressing of histograms to formulas, but it is possible that not all special cases are taken into consideration. The effects of its parameters were tested and approved on the entirety of one endoscopy video, but not mathematically proven in any way.
- Professional review. While it was mentioned that the evaluated techniques may erase certain details in endoscopy video, our claims on the impact of this are only assumptions, because the manipulated video was never reviewed by actual practitioners. Comparison with labelled footage was not performed either.

5.2 Stimulus for future work

It follows from the earlier conclusions that the research questions and requirements have been answered satisfactorily, but provides a basis to build improvements upon.

On the scientific aspect, it should be noted that during this project we have worked with one existing algorithm from the related literature, while the original intention was to implement and experiment with several. Therefore, one basis for future work is to implement any of the algorithms that were explored during the literature study, but ultimately not chosen due to time and resource constraints. With this, a broader and more varied research in the possible different combinations of algorithms and parameters can be performed, with the goal of getting better results than those shown in this thesis. Any improvements, adaptations or other alterations of the proposed algorithm can also be experimented with.

On the software aspect, the performance of the program is currently limited due to the time it requires to calculate image channel histograms for each frame. While video rendering is currently possible in real-time, any improvements of the implementation to reduce processing time is enormously valuable, especially due to the medical nature of the application.

References

[1] IEEE Xplore Digital Library,
`http://ieeexplore.ieee.org/` [Accessed 22 April 2017]

[2] ScienceDirect.com — Science, health and medical journals, full text articles and books,
`http://www.sciencedirect.com/` [Accessed 22 April 2017]

[3] ACM Digital Library,
`http://dl.acm.org/` [Accessed 22 April 2017]

[4] Anbarjafari, G. (2014). "HSI based colour image equalization using iterative nth root and nth power," eprint arXiv:1501.00108

[5] Vig, N., Budhiraja, S., Singh, J. (2016, August). "Hue preserving color image enhancement using guided filter based sub image histogram equalization," IEEE, 2016 Ninth International Conference on Contemporary Computing (IC3), (pp. 1-6).

[6] Purushothaman, J., Kamiyama, M., Taguchi, A. (2016, October). "Color image enhancement based on Hue differential histogram equalization," IEEE, 2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), (pp. 1-5).

[7] Gautam, C., Tiwari, N. (2015, May). "Efficient color image contrast enhancement using range limited bi-histogram equalization with adaptive gamma correction," IEEE, 2015 International Conference on Industrial Instrumentation and Control (ICIC), (pp. 175-180).

[8] González, D. M., Ponomaryov, V., Kravchenko, V. (2016, June). "Cromaticity improvement in images with poor lighting using the Multiscale-Retinex MSR algorithm," IEEE, 2016 9th International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter and Submillimeter Waves (MSMW), (pp. 1-4).

[9] Moradi, M., Falahati, A., Shahbahrami, A., Zare-Hassanpour, R. (2015, March). "Improving visual quality in wireless capsule endoscopy images with contrast-limited adaptive histogram equalization," IEEE, 2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA), (pp. 1-5).

[10] Vazquez-Corral, J., Bertalmío, M. (2014). "Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications," IEEE, Transactions on Image Processing, 23(10), 4564-4575.

[11] Wang, Y., Tao, D., Li, X., Song, M., Bu, J., Tan, P. (2014). "Video tonal stabilization via color states smoothing," IEEE, Transactions on image processing, 23(11), 4838-4849.

[12] Farbman, Z., Lischinski, D. (2011, August). "Tonal stabilization of video," ACM, Transactions on Graphics (TOG) (Vol. 30, No. 4, p. 89).

[13] Bassiou, N., Kotropoulos, C. (2007). "Color image histogram equalization by absolute discounting back-off," Computer Vision and Image Understanding, 107(1), 108-122.

[14] Iterative nth power/root image equalization - File Exchange - MATLAB Central,
`http://mathworks.com/matlabcentral/fileexchange/23987-iterative-nth-power-root-image-equalization`
[Accessed 23 April 2017]

[15] Color Image Histogram Equalisation code by GitHub user NoahDragon,
`github.com/NoahDragon/CodeLab/blob/master/ImageProcessing/ImageContrastEnhancement/ColorImageHE.py`
[Accessed 23 April 2017]

[16] OpenCV library
`http://opencv.org/` [Accessed 2 May 2017]

[17] wxWidgets: Cross-Platform GUI Library
`https://www.wxwidgets.org/` [Accessed 22 May 2017]

Appendices

A Colour models

In the field of computer graphics, the colour information in digital images can be saved in various representations, called colour models. Each colour model supports a certain purpose, and if an image is required for a different purpose, conversion between colour models is performed.

Digital colour photo and video is typically recorded in the RGB format, and the same format is used to display colour on televisions and monitors, where each pixel consists of a red, green and blue element. A colour in RGB format is represented by three values signifying how much red, green and blue must be blended together to create a colour.

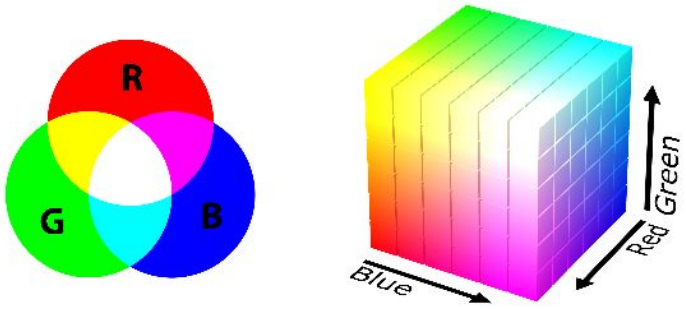


Figure 17: Left: Example of additive colours. Image credit: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/AdditiveColor.svg/1024px-AdditiveColor.svg.png
Right: RGB colour cube. Image credit: https://upload.wikimedia.org/wikipedia/commons/a/af/RGB_color_solid_cube.png

The left image in Figure 17 shows three circles in the primitive colours red, green and blue. When several circles overlap, the colour values are added together, resulting in a new colour. Therefore, it is said that colour blending in RGB is additive. This results in the RGB cube, shown in the right image of Figure 17. Every point in this three dimensional body is a unique possible combination of the values of red, green and blue, resulting in certain colour.

As mentioned, the main purpose of the RGB colour model is capturing and displaying digital colour images and video. However, this colour model does not capture certain properties in a single value. Examples of such properties, which can be recognised by the human eye, are how vibrant and how bright a certain colour is.

The HSV, HSI and HSL colour models are often used in the field of computer vision. The three models are fairly similar: the H component signifies the hue, the S component signifies the saturation, and the third component signifies the brightness of a colour in some way. In HSV, this is the value component; in HSI, this is the intensity component; and in HSL, this is the lightness component.

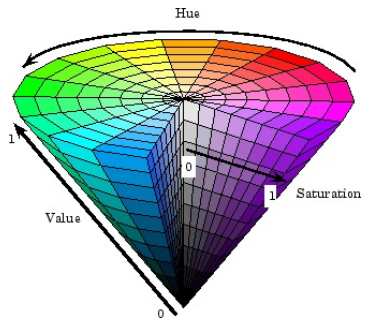


Figure 18: HSV colour cone. Image credit: https://www.mathworks.com/help/images/hsvcone.gif

Figure 18 shows the HSV colour cone. For the purpose of this explanation, the HSV colour model is similar enough to the HSI colour model. The hue component is a 360° spectrum, indicating where the colour is between infrared and ultraviolet. The saturation component indicates where the colour is between grayscale and the colour in full vibrace, and the intensity (or value) component indicates where the colour is between black (i.e. darkness) and the colour in full brightness.

B Software documentation

The software developed for this project is intended to be open for extension and improvement. The following documentation serves to explain how the software was designed, so that people interested in working with the code can understand its design and workings.

B.1 Technology stack

The program is built in the C++ language, using the OpenCV (Open Source Computer Vision) library[16] for the image processing, and the wxWidgets library[17] for the user interface. In terms of the MVC (Model-View-Controller) pattern, OpenCV provides the model, and wxWidgets provides the view and controller. Both libraries are available for use on all mainstream operating systems, so that the software tool is portable.

B.2 Software architecture

The most important design aspects of the architecture for this project are the separation of model and interface, and the modularity of the algorithms. The wxWidgets library provides an application and a frame class, which handle the view and controller responsibilities, detached from the model. The model consists of a manager class and any number of algorithm classes. Algorithm classes can be added, removed or modified easily, thus ensuring that the program is modular enough to allow for the testing of various algorithms on the same video footage.

Figure 19 shows a UML diagram with the classes and their responsibilities, main functionalities and interactions. For easy viewing, the diagram is deliberately on a moderately abstract level.

The ColourStabilisationFrame class defines all GUI components and their accompanying events. These events trigger changes of the model, and these changes are handled by the AlgorithmManager. This class is responsible for all aspects of the model except for the algorithms themselves. The algorithms are in separate classes, and when the AlgorithmManager needs an input frame to be processed, it calls the public function of the desired algorithm class.

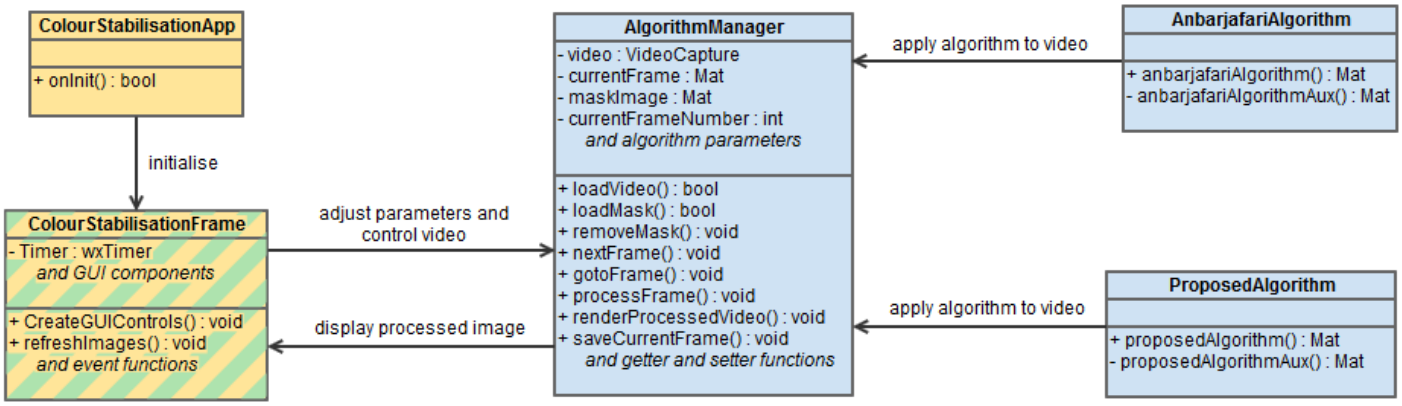


Figure 19: UML diagram of the software tool. The colours of the class signify their role in the MVC pattern: blue for model, yellow for view and green for controller.

B.3 Implementation of requirements

For each of the software requirements listed in Section 1, it is described how the functionality for meeting the requirement is implemented.

The user can load a video file.
The ColourStabilisationFrame contains a LoadVideoButton with LoadVideoButtonClick event that is fired when the button is clicked. In this event, a file dialog provided by wxWidgets opens, allowing the user to browse for an AVI video file to open. A String of the file location of the chosen video is then passed to the AlgorithmManager function loadVideo, which attempts to open the file. If it fails, the user is notified by a message in the status bar of the interface. If it succeeds, the video is loaded and the first frame will be shown. Data such as the number of frames in the video is also collected and stored, and shown in the interface where appropriate.

The user sees the original video next to the video with corrections applied to it.
The ColourStabilisationFrame has two wxStaticBitmap components side by side. Every time the model is instructed to alter the images (either when going to a different frame, be it manually or automatically, or when the selection of algorithms and parameters is changed), the frame class gets the before-image (currentFrame) and after-image (currentProcessedFrame) from the AlgorithmManager before displaying them in the interface. In addition to processing the images, the AlgorithmManager also converts the images from the OpenCV Mat format to the wxWidgets wxBitmap format.

The user can play, pause and seek in the video.
The ColourStabilisationFrame has a set of interface components directly below the image previews, for controlling the video playback. When a video is loaded, it is initially paused on the first frame. When the user clicks the play button, a wxTimer starts to fire an event every 0.333 seconds. The onTimer function that handles this event instructs the AlgorithmManager to advance one frame and provide the new images. This way, video will play at three frames per second. Another component is a wxSlider, allowing the user to seek through the video. When the position of the slider is changed to a certain frame number, AlgorithmManager will process and provide the image of that frame near instantly. Additional buttons for going back or forward by one frame, and a text box for entering a specific frame number to jump to, invoke AlgorithmManager in similar manners.

The user can select which algorithms to apply to the video, and sees the results of changing settings in real time.
The ColourStabilisationFrame contains a checkbox for each algorithm and image channel combination. When a video file is loaded and the user clicks one of the checkboxes, a boolean value is set in AlgorithmManager, signifying whether a certain algorithm is to be used. The image is then updated and shown on the screen again, so the result of enabling or disabling a certain algorithm is seen in real time.

The user can change the parameters of each algorithm, and sees the results of changing settings in real time.
The ColourStabilisationFrame contains a multitude of components for adjusting the parameters of each algorithm. Most can be adjusted with both a slider and a text box. In such a case, not only is the new value reported to the AlgorithmManager which returns the processed output image in real time, but the accompanying interface component is changed to match the input. This means that if the value of a slider is changed, the text in the accompanying textbox is changed to match this new value, and vice versa. In the case that the time window reach parameter of the proposed algorithm is changed, the AlgorithmManager (re)computes the histograms within this new window

The user can render a video with the chosen techniques and parameters applied to it.
The ColourStabilisationFrame has a button for rendering the processed video, and two textboxes that allow the user to set the start frame and end frame if they want to render only a portion of the footage. When the button is clicked, a file dialog allows the user to give the destination and name of the video file to be created. This file location is then sent to the AlgorithmManager function renderProcessedVideo as a String. OpenCV functions are then used to open a VideoWriter at the given file location and prepare this file with the correct specifications (such as frame rate and resolution). Then, the AlgorithmManager will play the video while processing each frame, but instead of displaying the output frames in the interface, they are written to the output video file. When all frames in the desired range are processed and recorded, control is handed back to the user.

C Survey instructions

(This is the exact text that was presented to participants of the survey.)

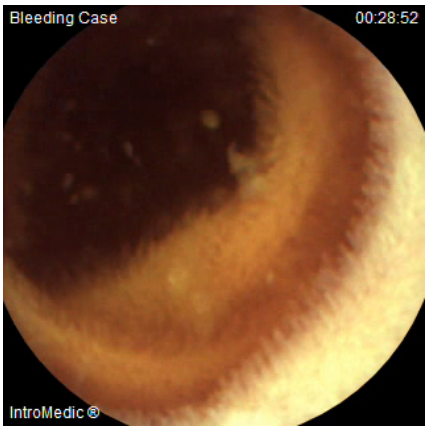
Welcome to the survey for the Bachelor project 'Colour stabilisation of endoscopy video' by Sibren van Vliet at the University of Groningen. The aim of this survey is to gather a qualitative assessment of the clarity of processed images.

The images are from endoscopy video, filmed by a camera travelling through the human gastrointestinal tract. In these images, areas close to the camera may be too bright, while areas far away from the camera may be too dark. (See the example image below.) This might hide details in these areas. The goal of the project is to manipulate the images so that details in these areas are more clearly visible.

During this survey, imagine that you are a doctor, watching the footage to scan for anomalies in the patient. The images in the endoscopy video have bright and dark areas, and you want to be able to see all areas clearly. You will now be presented 30 sets of images, each containing 6 images. (The first image of each set is the original footage, and the next five images are various manipulations.)

For each set of six images, please select the picture where you think that all details are equally clearly visible. This is a qualitative assessment, so there are no wrong answers. Furthermore, you do not need to look at each image set for very long. Simply choose the image that, in your opinion, is not to dark and not too bright, appears to have the least noise and information loss, and then move on to the next set. You should be able to finish this survey in 5 minutes, so please look at each image set for 10 seconds at most. (Please note that it is not your goal to look for medical anomalies, only to judge the image clarity.)

Example image. Notice how the upper left area is very dark, and the lower right area is very bright.



(On each of the following five pages, there are six subsections with an image set (consisting of six images) as explained in Subsection 4.1.2. Each image set had the following instruction.)

In which of the following images do all details, in bright and dark areas, appear clearly visible to you?

(On each of the five pages, after the six image sets, the last question was as follows.)

Now, please compare the images you chose in the last six image sets. Of the images you selected, which one do you find the clearest? (Please select multiple answers if, and only if, the images of your choice look identical to each other.)