Automatic hypercube acquisition with high spatial and spectral resolution using a HSI linescan camera

Robin Pape, Stefan Patzke, Sebastian Zaunseder, Jörg Thiem

Angaben zur Veröffentlichung / Publication details:

Pape, Robin, Stefan Patzke, Sebastian Zaunseder, and Jörg Thiem. 2021. "Automatic hypercube acquisition with high spatial and spectral resolution using a HSI linescan camera." *Current Directions in Biomedical Engineering* 7 (2): 811–14. https://doi.org/10.1515/cdbme-2021-2207.

Robin Pape*, Stefan Patzke, Sebastian Zaunseder, Jörg Thiem

Automatic Hypercube Acquisition with high spatial and spectral resolution using a HSI Linescan Camera

Abstract: Push-broom HSI-cameras have a high spectral resolution but require scanning and stitching to get a correct spatial and spectral representation of a scene. The conventional approach is to calculate a scanning velocity that matches the camera's frame rate so that the stitching becomes trivial. For this to work one needs to be able to apply the velocity steadily and know the distance between camera and subject. This contribution presents a feature-based approach that directly determines the distance the subject was moved between the images and uses it to correctly stitch the images. It is used in a self-contained environment consisting of a light-protected box with a light source inside and a linear guide unit that moves the object under examination. The distance between camera and object can be altered to increase the spatial resolution for small objects. It is shown that our stitching approach works even if the velocity changes during scanning. If the distance of the camera is altered, e. g. to scan a larger object, the stitching is adjusted automatically without the need for manual intervention.

Keywords: hyperspectral imaging, stitching, push-broom, image acquisition

https://doi.org/10.1515/cdbme-2021-2207

1 Introduction

Unlike RGB-cameras, which capture only three wavelength bands (red, green and blue) to represent the spectral range of visible light, hyperspectral cameras can capture multiple visible and also non-visible parts of the spectrum like near-infrared. They are thus able to reveal additional information in conventional images, making them an interesting technology for different research areas and applications. Hyperspectral imaging (HSI) has already been used successfully in medical applications like cancer detection [1], measuring haemoglobin oxygen saturation [2] or detecting retinal diseases [3].

There are several types of HSI cameras which differ in the way they acquire the images. Snapshot cameras take an image of the scene by capturing spatial and spectral information for each pixel at once. For every pixel of the resulting image, the sensor itself needs to have as many pixels as there are wavelength bands being captured. As the amount of pixels on a sensor is limited, there is a trade-off between spectral and spatial resolution. This trade-off can be resolved by using either spectral or spatial scanning. Using spatial scanning, the images need to be stitched afterwards. If the distance between camera and subject is known and does not change, one can calculate a scanning velocity that matches the cameras frame rate in a way that the resulting images only need to be stitched together without any overlap, so the stitching becomes trivial. However, under real conditions, it may not be possible to apply this velocity perfectly or the distance between the subject and the camera may not be known exactly or change frequently. This contribution proposes an automatic feature-based stitching algorithm that can handle deviations in the scanning velocity and changes in the distance between camera and object, making it easier to acquire hyperspectral images under non-optimal conditions in the field.

2 Methodology

We used a VRmagic HSI camera with an imec LS150 linescan sensor which is partitioned into 192 filter zones, each of which is 5 pixels tall and 2048 pixels wide. The bandpass filters in front of the pixels of the sensor only let light of a specific wavelength pass. As some of the filter zones overlap, it captures 150 different wavelengths. An image of the camera therefore always contains the whole scene and every wavelength of the captured spectrum, but each wavelength is

***Corresponding author: Robin Pape:** University of Applied Sciences and Arts, Dortmund, Germany, e-mail: robin.pape001@stud.fh-dortmund.de
Stefan Patzke, Sebastian Zaunseder, Jörg Thiem: University of Applied Sciences and Arts, Dortmund, Germany

captured only in a 5 pixel tall "stripe" which depicts a different part of the scene spatially. Therefore, it is necessary to either move the camera or the subject so that every part of the scene is captured in every wavelength.
The camera is mounted inside a light-protected box, facing down to the subject which is placed on a board that can be moved by a linear guide unit (LGU). The distance between camera and subject can be varied to make it possible to get a high resolution image from small objects by moving the camera closer while still being able to capture bigger objects, too. The subject is illuminated by a single halogen lamp.

2.1 Feature Detection

In every raw image from the HSI camera, which still contains all wavelength bands, we detect SURF features (Speeded Up Robust Feature) and match them to the previous image. SURF features are points in the image, that can reliably be detected as they stand out in their surroundings [4]. To be able to later calculate the distance between the features, we need such a feature detector that returns single points. The features also need to have a dedicated descriptor vector, which makes it possible to find matching points in multiple images. We compared four popular feature detectors that fulfill the requirements, namely SURF, ORB, FAST and BRISK, by detecting and matching features for 600 images of a beef steak (see 2) with default settings for each one in MATLAB. As we intend to use our HSI-environment mainly for human tissue, the beef steak is a reasonable substitute for this. Among the tested feature detectors, only ORB surpassed SURF in terms of detected matches (46332 inliers in all 600 images combined compared to 26430). However, detecting and matching took nearly five times as long (159.6s compared to 33.7s). While taking only 18.6s, FAST only detected 3748 matched inliers. BRISK was slowest and also detected the least matches (167.9s and 2252 inliers). In this comparison, SURF therefore provided the best trade-off of duration and matched features.
From the matches we can compute the distance the feature points moved between the images. In the ideal case where all points lie in a plane and there are no false matches, this is the distance the object was moved in the image, which can then be used to stitch the images together so that the matching points from both images are at the same coordinate in the resulting image. In reality, we need to perform some basic outlier rejection based on prior knowledge to remove false matches. As the LGU only moves the object in the v-direction of the image, we can remove all matches where the points do not lie at the same u-coordinate. Also, all matches with a negative distance are removed, as there is no backward movement. The remaining matches may still have different distances if they do not lie in a plane. We found that choosing the median distance of all remaining points gives the best results as it suppresses the influence of possibly remaining outliers.

2.2 Stitching

The raw images contain each of the captured wavelengths as a 5-pixel tall stripe. The stitching process involves taking the stripes and inserting them into new images, each of which will then contain only the information from one single wavelength. As we determined the distances between the images earlier, we now know how much the stripes need to overlap so that corresponding image features are at the same position in the resulting image. For every filter zone, we get one resulting stitched image ("single filter zone image").

2.3 Stacking

Stacking the single filter zone images to a hyperspectral cube is trivial. The stripes capturing the individual spectrums on the sensor are always 5 pixels tall and directly adjacent to each other. As the images have all been stitched using the same distances, the single-spectrum images are therefore always shifted 5 pixels relative to the previous image, meaning the second image has an offset of 5 pixels to the first one, the third image has an offset of 10 pixels to the first one and so forth. In the stacking process, this offset is accounted for. Finally, the parts of the stacked image that do not include all of the wavelengths are cut off.

3 Verification and Results

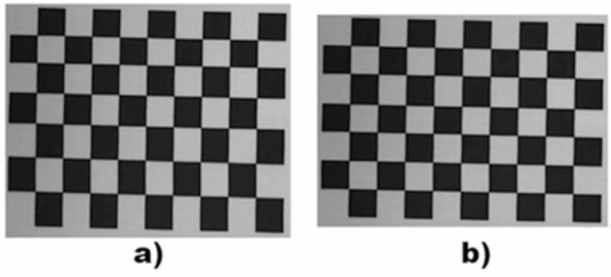

Figure 1: Stitched image of checkerboard with changing scanning velocity. a) Using the conventional static approach. b) Using our dynamic stitching approach. The conventional approach cannot compensate for the changing velocity, so some of the boxes get stretched.

To show the effectiveness of the method, a checkerboard pattern is scanned. We simulate the case of a fluctuating scanning velocity by starting the linear guide unit with the correct velocity which matches the camera's frame rate and then changing it during the scanning process to first 90% and then 80% of the initial velocity. The correct velocity is calculated using the formula

$$v = \frac{h_f \cdot f}{m(z)} \quad (1)$$

which was taken from [5].
h_f is the height of the filter zones on the sensor, f is the camera frame rate and $m(z)$ is a model function that calculates the image scale dependent on the focal length f_B, the pixel-size d_{pix} and the distance z between camera and subject:

$$m(z) = \frac{f_B}{d_{pix}} \cdot \frac{1}{z - f_B} \quad (2)$$

We found, however, that the results using the velocity calculated with this formula were not exact. According to [5], the main reason for this is an increasing image distance due to focusing when the object distance gets smaller, which the model function does not account for. We multiplied the calculated velocity by a factor of 1.1381 to get correct results. The factor was found by looking at the mean height and width of the scanned checkerboard boxes. As the mean height was 1.1381 times the mean width, the speed had to be multiplied by this factor to make the height and width equal.

As the conventional (static) stitching approach relies on the correct velocity being applied, the stitching fails as soon as the velocity changes (see Figure 1:a). In the band that is shown, the first two rows are still fine, but afterwards the velocity was lowered and the boxes get stretched. This only applies to the band shown. For other bands, the stretching happens at another point which makes the problem even more complex. Our dynamic approach however detects the change in velocity and adjusts the stitching accordingly by increasing the overlap of the stripes (see Figure 1:b). We calculated the deviation of width and height of the checkerboard squares, which ideally should be 0, at five wavelengths (500nm, 599nm, 700nm, 800 nm, 899nm) and took the mean. In the static case, the mean deviation is 18.1 pixels. With our dynamic stitching approach, the mean deviation of width and height of the checkerboard squares is only 1.0 pixels.

To show that our dynamic stitching approach can also handle varying distances between the camera and the object, we changed the distance from initially 315mm to 415mm without increasing the velocity of the LGU (which is necessary to get correct results with the static stitching approach) and scanned the checkerboard again. With the static stitching, this, as expected, resulted in stretched boxes (see Figure 2a) with a mean deviation of width and height of 20.7 pixels. In contrast, our dynamic stitching approach is able to correct for the changed distance so the boxes stay square with a mean deviation of width and height of 0.6 pixels (Figure 2b).

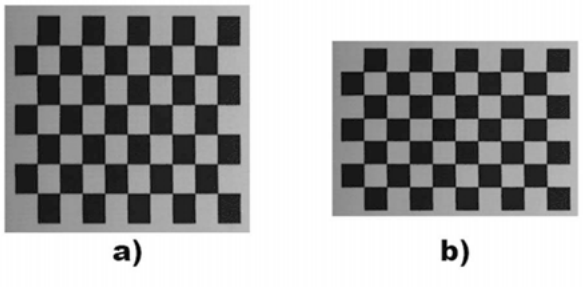

Figure 2: Stitched image of checkerboard with not matching distance and velocity. a) Using the conventional static approach. b) Using our dynamic stitching approach. The dynamic stitching approach adjusts the stitching to the new distance.

The velocity calculated using equations (1) and (2) is not only the velocity that needs to be applied in case of the static stitching approach, but also the upper limit for our dynamic approach. Anything above would result in our approach also failing, as the object then moves faster than the camera takes images and not every part of the object will be captured in every band. To be on the safe side, the velocity should be adjusted slow enough to comply to this upper limit even if the camera is placed at the nearest possible distance to the object. In this case, the stitching process will work for greater distances, too. In theory, there is no lower limit for the velocity for our approach to work, but practically for very slow moving objects very many images will be taken which will be computationally too expensive.

As a real-world example and a first test of our HSI environment we captured images of a beef steak to see if we are able to distinguish muscle/flesh, bone and fat in the spectrum after using our HSI environment and the automatic stitching method. For each of the three we manually picked a 10x10 pixel sized patch that only contains the one specific tissue type from an evenly lit part of the image. We then calculated the normalized, uncorrected mean intensity values for every wavelength. Figure 3: shows the results. The largest difference can be seen around 770nm where the intensity of fat is significantly higher than the intensity of flesh, which is still slightly higher than the intensity of bone. The same can be observed for the wavelengths around 717nm and 690nm. Fat also shows higher intensity than flesh and bone at 650nm to 665nm. Otherwise there are not any distinctive features of either tissue type in this case.

Using the k-means algorithm with all bands to segment the image into three classes showed that distinguishing fat from flesh worked mostly well, whereas bone was often misclassified as flesh (see Figure 4: b). This agrees with the findings in the spectrum, where fat was distinguishable from flesh and bone, whereas the latter were much closer together. However, very bright parts of bone were also misclassified as fat. The most likely reason for these misclassifications is the non-uniform illumination by the single light we used and the fact that the differences between flesh, fat and bone in the spectrum our camera captures are relatively small.

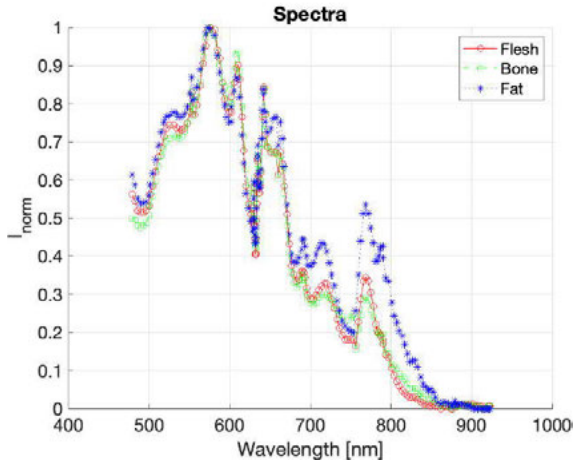

Figure 3: Spectra of flesh, bone and fat. The intensity is the normalized mean value of a 10x10 pixel sized patch we manually selected for each of the tissue types.

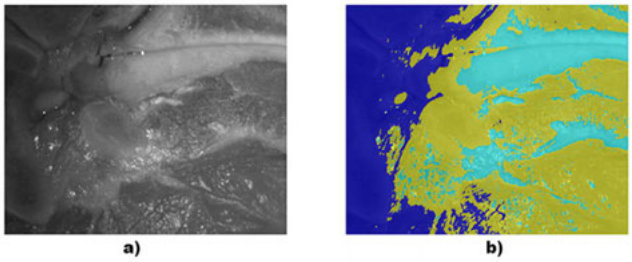

Figure 4: Crop of the steak. a) In one single band. b) k-means segmentation in muscle (yellow), fat (cyan) and bone (blue). Not every pixel was classified correctly.

4 Conclusion

By using our dynamic stitching algorithm, we were able to achieve good results in situations where a steady velocity cannot be applied and when changing the distance between the camera and the subject, e. g. moving the camera farther away to capture a bigger object completely. With the conventional approach one would need to adjust the velocity to the new distance, whereas our approach is able to work with the slower velocity that was applied before and therefore simplifies the process.

When used on a beef steak, the feature-based stitching algorithm is able to detect enough feature matches to work well which suggests that it should also work on human tissue. In a future work, the illumination should be improved to be more homogeneous and cover a wider spectrum than our halogen lamp and a spectral correction method could also be applied to the resulting spectra to make them more comparable to other works [6]. Additionally, other HSI cameras with advantages like a wider spectral coverage could be used to get better results when used at an operational level in a pathological institute.

Author Statement

Research funding: Financial support from Ministry of Culture and Science of North Rhine-Westphalia under the program FH BASIS for purchasing the hyperspectral camera system, is gratefully acknowledged. Conflict of interest: Authors state no conflict of interest. Informed consent: Informed consent has been obtained from all individuals included in this study. Ethical approval: The conducted research is not related to either human or animals use.

References

[1] Martin ME, Wabuyele MB, Chen K, Kasili P, Panjehpour M, Phan M, Overholt B, et al. Development of an Advanced Hyperspectral Imaging (HSI) System with Applications for Cancer Detection. Annals of Biomedical Engineering 2006;34:1061-1068

[2] Zuzak KJ, Gladwin MT, Cannon RO, Levin IW. Imaging hemoglobin oxygen saturation in sickle cell disease patients using noninvasive visible reflectance hyperspectral techniques: effects of nitric oxide. American J. of Physiology - Heart and Circulatory Physiology 2003;285:1183-1189

[3] Johnson WR, Wilson DW, Fink W, Humayun M, Bearman G. Snapshot hyperspectral imaging in ophthamology. Journal of Biomedical Optics 2007;10.1117/1.2434950

[4] Bay H, Tuytelaars T, van Gool L. SURF: Speeded Up Robust Features. Proceedings of the 9th European conference on Computer Vision - Volume Part I 2006; 10.1007/11744023_32

[5] Schwarzer J. Modellbasierte Entwicklung einer Roboter gestützten Führung eines HSI-Zeilenscanners. Project work at University of Applied Sciences, Dortmund, 2018.

[6] Pichette J, Goossens T, Vunckx K, Lambrechts A. Hyperspectral calibration method For CMOS-based hyperspectral sensors. In: Yakov G. Soskind, Craig Olson, eds. Photonic Instrumentation Engineering IV. Vol. 10110. SPIE; 2017:132-14