

The Good, the Bad and the Ugly: droplet recognition by a “shootout”-heuristics

Hans-Georg Matuttis¹, Silvia Holler², Federica Casiraghi², Johannes Josef Schneider³, Alessia Faggian², Rudolf Marcel Fuchsli^{3,4}, and Martin Michael Hanczyc^{2,5}

¹ Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Chofu Chofugaoka 1-5-1, Tokyo 182-8585, Japan

hg@mce.uec.ac.jp

² Laboratory for Artificial Biology, Department of Cellular, Computational and Integrative Biology, University of Trento, Via Sommarive, 9-38123 Povo, Italy

³ Institute of Applied Mathematics and Physics, School of Engineering, Zurich University of Applied Sciences, Technikumstr. 9, 8401 Winterthur, Switzerland

⁴ European Centre for Living Technology, S.Marco 2940, 30124 Venice, Italy

⁵ Chemical and Biological Engineering, University of New Mexico, MSC01 1120, Albuquerque, NM, 87131-0001, USA

Abstract. We explain how to optimize the image analysis of mixed clusters of red and green droplets in solvents with various degrees of sharpness, brightness, contrast and density. The circular Hough Transform is highly efficient for separated circles with reasonable background contrast, but not for large amounts of partially overlapping shapes, some of them blurred, as in the images of our dense droplet suspensions. We explain why standard approaches for image improvement fail and present a “shootout” approach, where already detected circles are masked, so that the removal of sharp outlines improves the relative optical quality of the remaining droplets. Nevertheless, for intrinsic reasons, there are limits to the accuracy of data which can be obtained on very dense clusters.

Keywords: Image processing, Hough transform, cluster of droplets in fluids

1 Introduction

It is relatively straightforward to obtain the full information about the geometry of droplet clusters in simulations[10] to verify existing theories with respect to connectivity and packing characteristics [6,9,8], e.g. with applications to chemical reactors of micro-droplets[1,12]. In contrast, experimental verification is difficult due to potentially lost information in the original image. Nevertheless, experimental understanding of the clustering it is indispensable for theoretical predictions concerning the use of DNA technology for DNA-directed self-assembly of oil-in-water emulsion droplets [2]. While the final aim will be to analyse three-dimensional data from confocal microscopy, in this paper, we will restrict ourselves to the analysis of two-dimensional conventional images. We

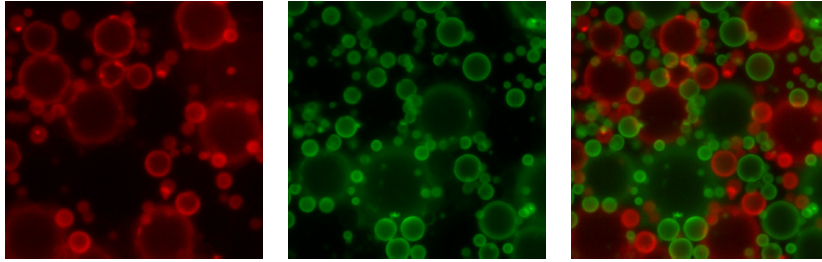


Fig. 1. Image details of droplets with red (left), green (middle) and sum of both red and green (right) color values.

will explore the possibilities of extracting as much as possible information about droplet clusters (radii and positions) with the circular Hough transform. It is a standard technique[7,4,5] for recognising shapes in computer vision and image processing (for it's checked history, see Hart [3]). The circular Hough transform works well for separate circles in two dimensions with good background contrast. This is unfortunately not the case for our graphics: We have several thousand droplets in red and green, partially hidden by each other, with radii from five pixels to several hundred, some out of the focal plane, with varying contrast and sharpness. The red and green droplets have been recorded as tif-RGB-images, with color values either in the red or in the green channel, see Fig. 1. Depending on the choice of color models, interactive programs like Apple's "Digital Color Meter" may indicate mixed colors in the screen display even if the droplets in the original tif image have values only in the red or the green channel. Unfortunately, especially droplets which are larger and out of the focal plain contain different levels of salt and pepper noise (random white, respectively dark pixels). Together with blurred large droplets, there may be small droplets with sharp surface reflections of much higher brightness and scattered

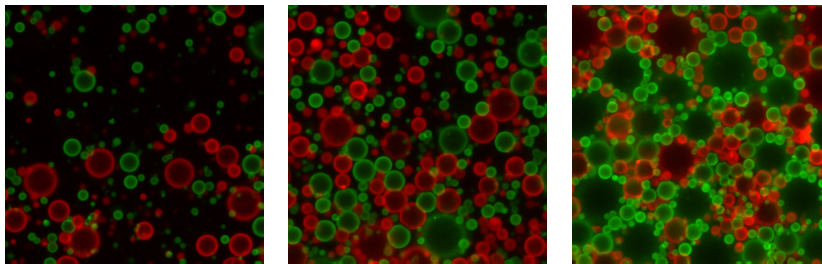


Fig. 2. Image details of "good" (left), "bad" (middle) and "ugly" (right) clusters of droplets (sum of red and green color values); from left to right, the number and density of droplets with too different image quality (light and dark, sharp and fuzzy, with and without noise) increases, so that it becomes more difficult to separate individual droplets and determine their radii and positions correctly.

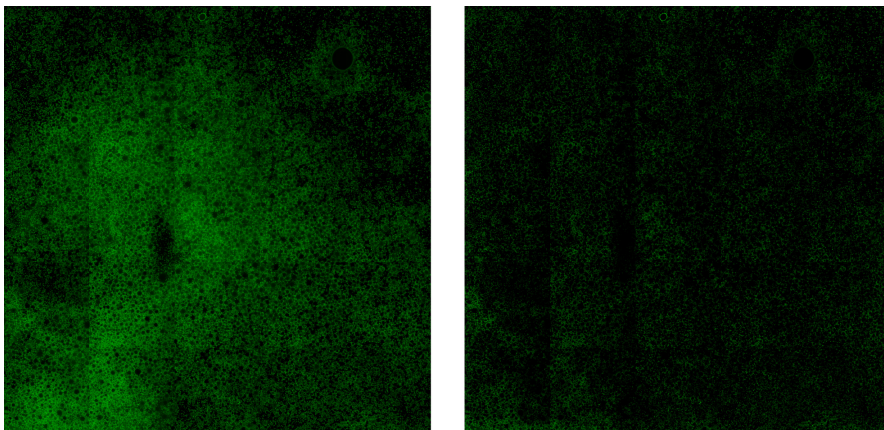


Fig. 3. Composition of 5×5 images (green channel), the original (left) with varying brightness, and the homomorphic colouring (right), with homogeneous, but unfortunately reduced brightness and resolution of details.

reflections inside larger, blurred droplets. In general, very small droplets cannot be discriminated from circular scattered reflections, in Fig. 1. From the experiment we have to take the pictures as they are and cannot scan through various focal planes, as is possible with computer tomography or magnetic resonance imaging. All in all we have to deal with three kinds of image qualities as in Fig. 2, which can be catchily expressed as “the Good” (left, reasonable contrast, good separation), “the Bad” (middle, fuzzy contrast, no clear separation) and “the Ugly” (right, deep stacking of droplets to dense clusters with bad contrast, some large droplets only recognisable as “black holes” acting as spacers between smaller droplets). In other words, we have far from perfect input data and when we extract the size and (two-dimensional) position of the droplets, we cannot expect a perfect geometrical representation, only a “best effort” is possible. In this research, we explain how as much information as possible can be extracted for such droplet clusters with methods originally developed for much better resolved circular objects, and which other methods of image processing may help, which may not, and why.

2 Basic image processing approach

2.1 Attempts at preprocessing to balance exposure

Initially, our ambition had been to process the whole of the various sets of 5×5 tif-frames, each with resolution of 2048×2048 . As can be seen in Fig. 3 (left), it turned out that the camera automatically had selected very different individual exposure of each frame, with sufficient time lag between each frame so that lines appeared where the individual tifs were joined together. It is possible to

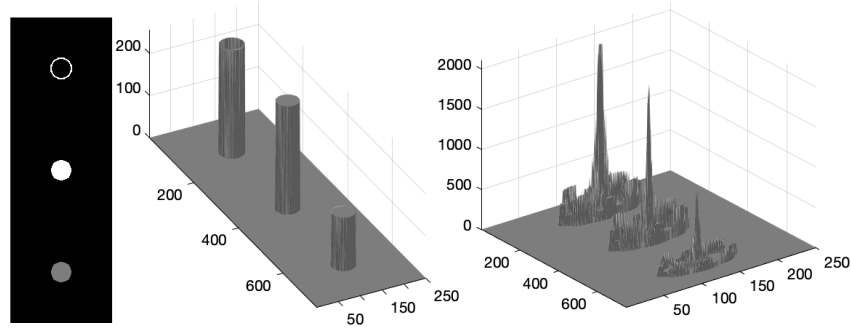


Fig. 4. For an image (left) with a circular ring, a full circle with full (255) and half color saturation (125) - all with the same radius - the corresponding 3D-Graph (middle) as well as the peaks of the circular Hough transform (right).

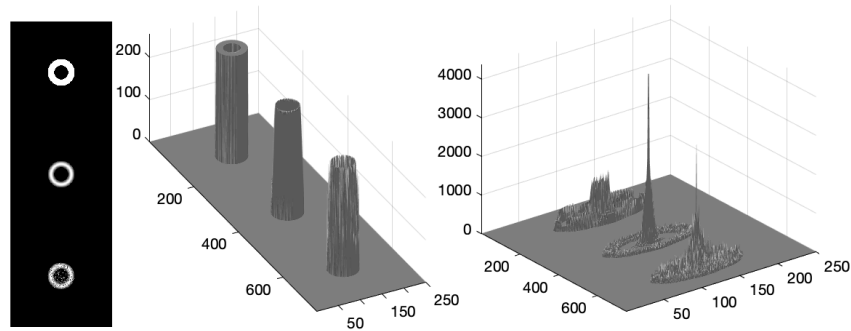


Fig. 5. For an image (left) with a circular ring (above), a circular ring with Gaussian intensity distribution (middle) and the same with 20% salt and pepper noise, the 3D-Graph (middle) as well as the peaks of the circular Hough transform (right).

remove the differences in brightness in different regions of images with so-called homomorphic filters[7] with a suitable combination of Fourier transforms and Gaussian filters. We applied such a filter in Fig. 3 (right), successfully inasmuch as the difference in light and dark regions disappeared. Nevertheless, not only the exposure differences, but also resolution and brightness as a whole were reduced, up to a point where the the shape of circles was destroyed. The original aim of analysing whole sets of 5×5 frames had to be abandoned, so we decided to focus on the analysis of individual tifs with 2048×2048 pixels resolution.

2.2 The circular Hough transform

While the standard Hough transform is a method to extract the information about lines (their location and orientation) from an image, the circular Hough

transform extracts information about circles (their centers and radii). Simply speaking, the circular Hough transform histograms the possible three-point least-squares fits of a circle for a given center (x, y) and radius r and stores the information in the resulting transform matrix. The range of the radii to be investigated is supplied as input parameter. Because there are more possibilities to fit large than to fit small radii, Hough transforms with larger radii take longer than runs with small radii due to the larger set of the input data. A maximum search on the elements of the transform matrix can then be used to determine the centers and radii of the circles in the original image. Because we will have to deal with droplets which may be imaged as rings or circles with varying shading intensity on the boundary, we will compare in the following the Hough transforms of some circular shapes. In Fig. 4, the circular Hough transform is performed on a monochrome image of 750×250 pixels with (from back to front) a circular ring and a full circle with saturation value 255, and in the foreground there is a circle with saturation value 125. All things being equal, circular rings give Hough transforms with higher peak intensity than full circles, and, as expected, the intensity of the Hough transforms decreases with the color intensity of the circles. In Fig. 5, we have shown the Hough transform (right) for an image of three circular rings (left), with constant intensity (above), with an outline of Gaussian intensity (middle) and with additional 20% salt and pepper noise (front). The peak height for the ring of constant intensity is lower, but wider, while the height of the peaks for the Gaussian outlines is comparable, but the noise reduces the width of the peak.

2.3 The MATLAB implementation of the Hough transform

We use the function `imfindcircle` [11] from MATLAB’s image processing toolbox (versions 2017b, respectively 2018b), which adapts the circular Hough transform with additional controls. As input parameters, the preferred range of radii $[r_{\min}, r_{\max}]$, the polarity (“bright” or “dark” circle to be detected), as well as the sensitivity (between 0 and 1) can be selected. Because most droplet reflections, as in Fig. 1,2, are not filled circles, but light circular rims with a darker core, the “bright” option for the circle detection must be used: The relative width of the inner “dark” core varies too much to give meaningful radii for the corresponding droplet. MATLAB’s `imfindcircle` is parallelized, and for graphics with 2048×2048 pixels resolution runs in parallel on 2 to 3 cores on a MacBook Pro. For our data, another necessary input is the sensitivity. Here, bigger is not always better, and a sensitivity of 0.9 is reasonable for our images. Beyond a sensitivity of 0.92, `imfindcircle` starts “seeing things”, in particular circles which are not there, and misidentifies irregularities on the outline of droplets, i.e. deviations from an ideal circle, as circular droplets on its own, as can be seen in Fig. 6. Apart from radius and position, `imfindcircle` outputs also the metric (as it is called in the MATLAB’s function description), i.e. the “quality” of the circles. In the case where spurious circles with smaller radii have been inscribed into larger droplets (like droplet 1 and 3 in Fig. 6), these small circles can be eliminated when `imfindcircle` is run with larger radii and it turns out

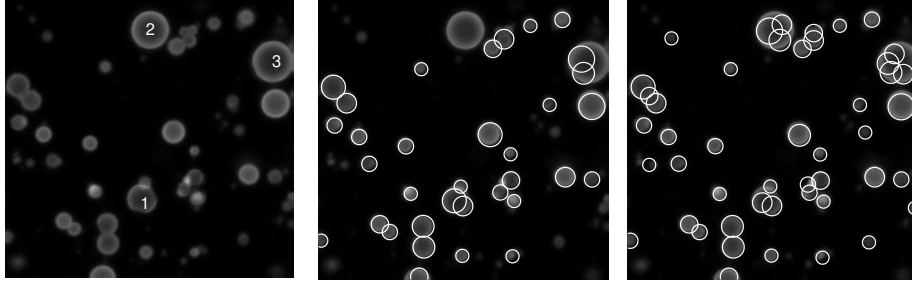


Fig. 6. From left to right: Original picture (left) as well as circles found with `imfindcircle` with sensitivity of 0.90 (middle) and 0.96 (right) with radii between 6 and 12 pixels. While the droplets 1 and 3 (left) are spuriously resolved with two circles at sensitivity of 0.90 (the outline is not circular) for a sensitivity of 0.96 also droplet 2 gets split into three circles.

that several small circles with low metric (quality) are situated inside a larger one with higher metric: In that case, it is clear that the smaller circles were only spurious and can be eliminated.

2.4 Loop over several radius-ranges

For our images, the best performance of `imfindcircle` is obtained if the input radii are in a range less than a factor of two (all lengths in pixels, input-data in angular brackets in typewrite-font to discriminate them from references): For $[r_{\min}, r_{\max}]$, $[8, 12]$ will give reasonable result when $[8, 16]$ may not. Therefore, we run `imfindcircle` with a sequence of increasing radius ranges and start with the smallest radii, because the smaller droplets are usually the clearer ones. Typical sequences are $[5, 8]$, $[8, 16]$, $[15, 26]$, $[27, 35]$, $[36, 56]$. For smaller radii of about 16 pixels, different circles may be recognized for a setting of $[8, 16]$ than for a setting of $[15, 26]$, for larger radii, the algorithm is more tolerant. With different input radii, the same droplet may be found twice with different radius ranges, e.g. with a radius of 12 for input range $[8, 16]$ and with a radius of 16 for an input range of $[15, 26]$: For the smaller input radius, the brightest radius will be detected, for larger radius the largest outer radius may be detected. Therefore, when new sets of positions and radii are obtained with `imfindcircle`, a loop over all old and new radii must be run to eliminate doublets, concentric circles with different radii but the same center (with an error margin of 1/10 of the larger radius).

3 From good to bad clusters

3.1 Failure of standard techniques for image improvement

The Hough transform works for the “good” droplet images with reasonable sharpness, brightness, contrast and separation distance from other droplets. The

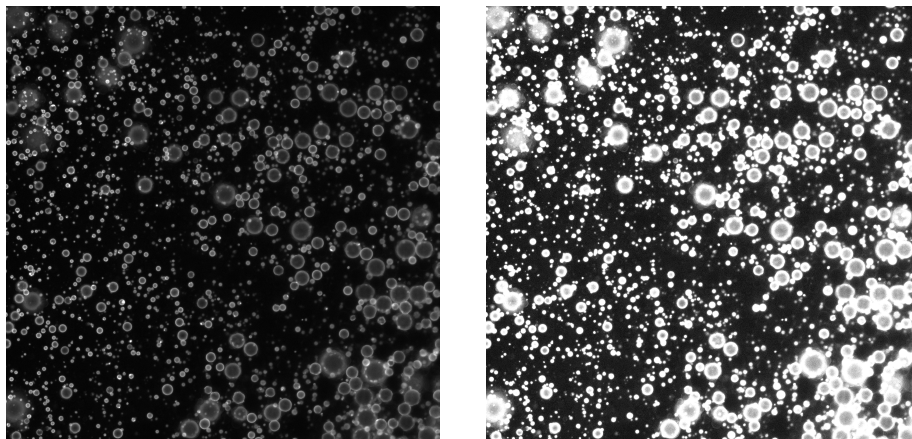


Fig. 7. Original input (left) and result with Laplace filtering using $A_{\text{Lap}2}$ (right).

optical quality of the droplets with small radius and center of mass in the focal plane will always be superior, their signals in the Hough transform always be sharper than for larger droplets or small droplets more distant from this plane. Droplets with less contrast, separation, brightness or sharpness will not be detected with the Hough transform directly, because the corresponding peaks (see sec. 2.2) are too low. When this happens, everybody who has worked with graphics programs will then try some of the standard methods to improve the image, in particular: brightening, increasing contrast and sharpening contours. All of these methods are implemented in MATLAB, and for interesting reasons turn out to be useless for preprocessing images for use with the Hough transform. Brightening of the input-data will brighten all droplets: The peaks for the sharpest and brightest droplets in the Hough transform will be increased too, and relatively speaking, nothing changes, the droplets with less contrast, brightness etc. will still be outshone. The same is true with increased contrast: In brightened images, the originally brighter droplets will always exhibit the better contrast compared to darker ones. In short, global or unified approaches to improve the picture quality will not help. Sharpening will be discussed in the following section.

3.2 Sharpening contours with Laplacian filtering

It is instructive to have a look at what sharpening with Laplacian filtering really accomplishes with a concrete example. Using filters (in digital image processing, i.e. a suitable convolution of an image with a filtering matrix) is a conventional technique to improve image quality. In MATLAB, it is implemented via `imfilter`[11], and the original image and the convolution matrix are input arguments. One of the standard procedures for image sharpening is the Laplace filter. It is derived from the three-point approximation of the Laplacian operator

$\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2$ for a discrete function $f(x, y)$ on a grid with unit spacing (which takes care of the denominator) so that $\Delta f(x_n, y_n)$ becomes

$$f(x_{n+1}, y_n) + f(x_{n-1}, y_n) + f(x_n, y_{n+1}) + f(x_n, y_{n-1}, y) - 4f(x_n, y_n). \quad (1)$$

The details of the approximation of the Laplacian have much less effect for image filtering than for numerical analysis. We used implementations as convolution masks with the matrices

$$A_{\text{Lap}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad A_{\text{Lap1}} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad A_{\text{Lap2}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad (2)$$

where A_{Lap} is the faithful implementation of the Laplace operator eq.(1), while A_{Lap1} [5] and A_{Lap2} and are crude approximations which nevertheless give a brighter image. We processed Fig. 7 (left) with A_{Lap2} , so that in the result Fig. 7 (right) the outlines were homogenised and brightened. Unfortunately, this erased other image details, and already bright droplets became brighter, so the relative intensity of the peaks in the Hough transform did not change.

3.3 Failure of more sophisticated image processing methods

One of the results of Laplace filtering in Fig. 7 (right) was that - while some specific droplets became lighter - in dense clusters, the separation between droplets vanished. To segment pixels between clusters, the `watershed`[11] transform could be used: Unfortunately, for our amount of data, together with the large size distribution in our images, we were not able to find a suitable separation rule. Also the `clearboarder`[11] algorithm could not be used, which can remove frayed out borders around separate objects, but not for partially overlapping objects. The temptation to run edge detection algorithms[7,4,5,11] on filtered images like Fig. 7 (right) is overwhelming. However, the results in Fig. 8, where the standard edge detection algorithms have been used, are rather underwhelming: The Sobel[7,4,5]- Roberts- and Prewitt[5] algorithm do not only produce circular shapes, but also any kind of outlines for connected clusters. The contours obtained from the Prewitt algorithm seem to be the most complete, with the

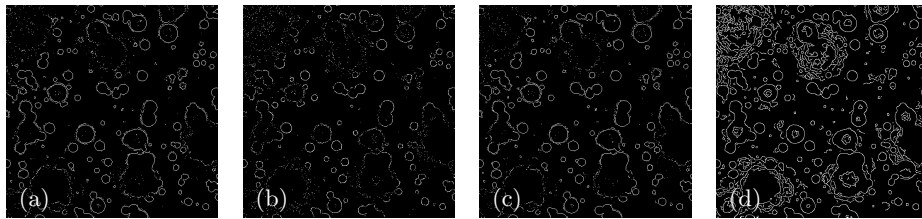


Fig. 8. Edge detection with Sobel (a), Roberts (b), Prewitt (c) and Canny (d) for the filtered data of Fig. 7; shown are the left upper 512×512 of the 2048×2048 pixels.

least amount of gaps. As usual, the Canny[7,5] algorithm produces the highest detail and most subtle contours. Unfortunately, in case of our ambiguous data that means: There are multiple outlines for the same shape, so the usefulness is rather marginal, compared to the rather more robust Sobel method, which usually produces a single outline. When running `imfindcircle` on the graphics with edge detection in Fig. 8 (a) to (d), the accuracy and resolution was actually reduced compared to the result for the original image.

3.4 The shootout method

The recognition of images of mixed optical quality is rather due to the “good” (bright, sharp, well-formed) droplets, because their distinct Hough peaks hide the less-distinct peaks of bad droplets, as in Fig. 9 (a). Consequently, it makes sense to mask already recognized droplets. To “shoot out” such detected circles from the image, we have to overwrite them with a suitable color hue. Choosing the wrong color creates spurious contours, i.e. new problems, so we also have to discuss the most practical choice of colors for overwriting droplets. A self-evident choice would be the average or median color of the whole image, as it should - on average - hide a droplet in the average background color. Nevertheless, the medium color of a whole picture with a lot of bright clusters or dark background to overwrite a droplet will create conspicuously bright or dark circles as in Fig. 9 (b). So it is not advisable to use color hues which are globally computed for the picture. Each “shoot out” should use a color which is inconspicuous relative to the vicinity of the droplet. That leaves a choice of using either all the data inside a droplet or the values on the circumference. We processed only the values on the circumference, as pixels inside a droplet were sometimes influenced by color deviations. As the diameter found with the Hough transform is very often that of the color maximum at the border and does not include the whole “halo”, we used an additional “safety distance” (between 1.1 and 1.2 times the Hough transform-radius) for the shootout. Choosing the darkest pixels on the circumference resulted in spuriously too dark colors as in Fig. 9 (c) with artificially sharp contrasts. The best choice was the average color on the circumference, see

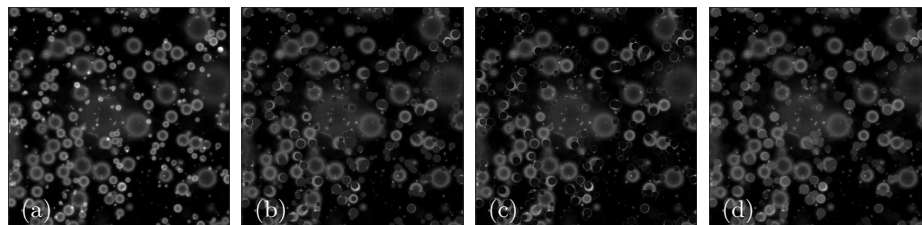


Fig. 9. Original picture (a), shootout of bright dots with average color of the whole frame (b), with darkest color on the circumference (c) and (best choice, to minimize sharp contrasts) with average color of the circumference (d).

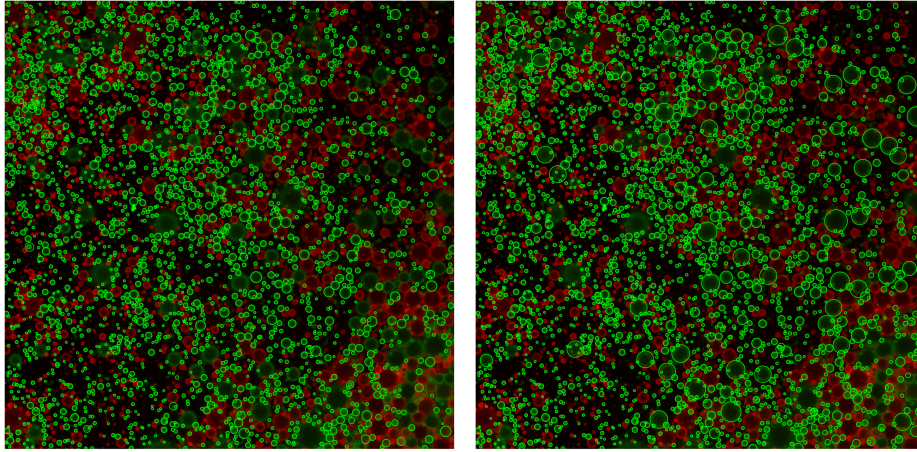


Fig. 10. Droplet recognition for the same image of 2522 green droplets recognized without the shootout procedure (left) and 2719 green droplets recognized with the shootout method (right), with the recognized droplets drawn over the original image.

Fig. 9 (d). The replaced circles were most of the time equivalent to the background color and did not induce spurious new circles. As can be seen in Fig. 10, with the shootout method the number of recognized droplets rises for the green droplets from 2522 to 2719, for the red ones from 1859 to 1999. This may not seem much, but in Fig. 10 (right), one sees that in particular large droplets which got unnoticed without shootout in Fig. 10 (left) can now be recognized. These large droplets with radii between 30 to 60 pixels are practically not detectable with the application of `imfindcircle` alone due to the lack of exposure and contrast.

3.5 Selecting “ugly” clusters according to color value

As the original purpose of our data evaluation had been the determination of neighbourhoods between red and green droplets, the recognition of larger droplets is essential for a meaningful analysis. Up to here, Fig. 10 looks like a “bad” image which can mostly be dealt with by our shootout method. In reality, the approach has failed, as superpositions hide particularly large droplets with very pale hue, sometimes surrounded by rings of other droplets. These clusters are visible in a certain color range (where 255 corresponds to full saturation): For the green channel between 25 and 80 and for the red channel between 15 and 70, “mega-droplets” appear in white in Fig. 11 which are hardly visible in Fig. 10. Clusters with circles inside could be (more often than not) detected by the circular Hough transform. Some clusters are much larger than the circles inside, or so deformed, as in the left upper corner of Fig. 11 (a), that no detection with the Hough transform is possible. Other huge clusters have their boundary and insides decorated with cutouts of other droplets.

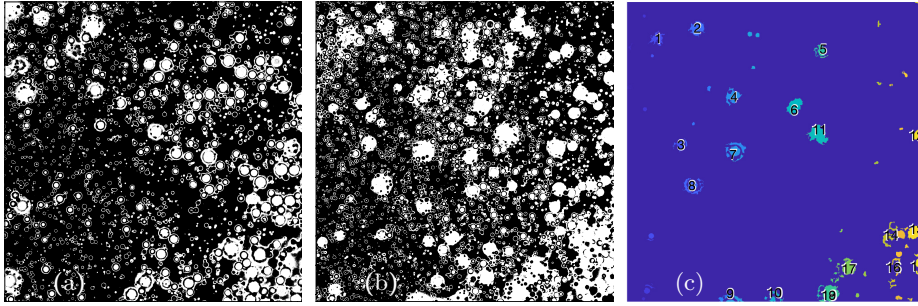


Fig. 11. Very huge droplets for the original image of Fig. 10 for the green channel with color values between 25 and 80 in (a), for the red channel with values between 15 and 70 (b) and semiautomatic recognition of the clusters in (b) shown in (c).

3.6 The final shootout

As in every good Western-themed undertaking, we will finish with a final shootout. The huge droplets could only be recognised by the color hue set by hand, so for green and red droplets, and for different pictures - depending on exposure and density - various values must be selected. This is not practicable for the automatic image analysis we aimed for. Nevertheless, we will now try other methods for cluster recognition, which do not use the Hough transform. For the red channel of the underlying image in Fig. 10, we removed all circular areas with recognized droplets with a shootout of both red and green circles, and binarised the image: All pixels were set to black except red color values between 25 and 48, which were set to white. Over this black-white image, we ran a sequence of MATLAB algorithms for irregular clusters. With `bwareaopen`[11], connected pixel clusters with a total number of 250 pixels or larger are determined, all smaller clusters are removed to erase the pixelated noise. Next, with `imfill`[11], holes inside connected pixel clusters are occluded. At the end, `bwboundaries`[11] is used to label connected pixel clusters. The result is shown in Fig. 11 (c) with numbers added for the largest droplets to simplify the discussion. The reconstruction of the droplets 1,2,4,6,11 and 13 looks mostly intact: The determination of the center and the maximal pixel diameter could give a somehow realistic location of the droplet. Issues exist with droplets 3,5,7 and 8 (also droplets 9 and 10, though they are only partially in the image), which appear in several unconnected parts. However, an automatic determination of the connection is not possible: An algorithm which would connect the central pixels of droplets 3,5,7 and 8 with their outer rim would also connect areas near droplet 16 and 15, which belong to separate droplets. Finally, the total size of droplets 12,13,14,17 and 18 are unclear, because one cannot determine where the fragments near 13,14,17 and 18 belong to. In the case of droplet 12, it is unclear how large the area is which is hidden or cut off by other droplets. Even with all this effort, and selecting the color range by hand, ambiguities about the largest droplets remain.

4 Conclusions and outlook

Our original aim in processing two-dimensional images of clusters was to obtain automatically an overview over the possible three dimensional contact geometries between droplets from two-dimensional images. Our investigations have clearly shown the limits of this approach: Our shootout-method worked well for droplets with a radius of up to 50 or 60 pixels. It was only possible to extend the recognition to droplets beyond that size by selecting color ranges by hand, and such an approach is futile for the amount of actual data which have to be processed. Information which was absent in the original image with respect to exposure and sharpness cannot be recovered even with the most refined analysis technique. In the future, we will strive for the evaluation of images with higher data content and less noise, from more sophisticated image acquisition methods, in particular from high-resolution fluorescence confocal microscopy.

References

1. Flumini, D., Weyland, M.S., Schneider, J.J., Fellermann, H., Füchslin, R.M.: Towards programmable chemistries. In: Cicirelli, F., Guerrieri, A., Pizzuti, C., Socievole, A., Spezzano, G., Vinci, A. (eds.) *Artificial Life and Evolutionary Computation*, Wivace 2019. vol. 1200, pp. 145–157 (2020)
2. Hadorn, M., Boenzli, E., Sørensen, K.T., Fellermann, H., Eggenberger Hotz, P., Hanczyc, M.M.: Specific and reversible dna-directed self-assembly of oil-in-water emulsion droplets. *Proc. of the Nat. Ac. of Sc.* (2012)
3. Hart, P.E.: How the Hough transform was invented. *IEEE Signal Processing Magazine* 26(6), 18–22 (2009)
4. Jähne, B.: *Digital Image Processing*. Springer Berlin Heidelberg (2005)
5. Marques, O.: *Practical Image and Video Processing Using MATLAB*. Wiley (2011)
6. Müller, A., Schneider, J.J., Schömer, E.: Packing a multidisperse system of hard disks in a circular environment. *Phys. Rev. E* 79, 021102 (Feb 2009)
7. Parker, J.: *Algorithms for Image Processing and Computer Vision*. Wiley (1997)
8. Schneider, J.J., Kirkpatrick, S.: Selfish versus unselfish optimization of network creation. *J. of Stat. Mech.: Theory and Experiment* 2005(08), P08007 (2005)
9. Schneider, J.J., Müller, A., Schömer, E.: Ultrametricity property of energy landscapes of multidisperse packing problems. *Phys. Rev. E* 79, 031122 (Mar 2009)
10. Schneider, J.J., Faggian, A., Holler, S., Casiraghi, F., Li, J., Cebolla Sanahuja, L., Matuttis, H.G., Hanczyc, M.M., Barrow, D.A., Weyland, M.S., Flumini, D., Eggenberger Hotz, P., Füchslin, R.M.: Influence of the geometry on the agglomeration of a polydisperse binary system of spherical particles. *Artificial Life Conference Proceedings*, 71 (2021). https://doi.org/10.1162/isal_a_00392
11. The Mathworks: MATLAB helppage. <https://uk.mathworks.com/help/index.html> (link last followed on July 16, 2021)
12. Weyland, M.S., Flumini, D., Schneider, J.J., Füchslin, R.M.: A compiler framework to derive microfluidic platforms for manufacturing hierarchical, compartmentalized structures that maximize yield of chemical reactions. *Artificial Life Conference Proceedings* (32), 602–604 (2020)