Chapter 2 Developing Content-Driven Superpixels	Chương 2 Xây Dựng Siêu Pixel Điều Khiển Theo Nội Dung
Existing superpixel approaches all require some form of initialisation. This is either in the form of specifying the number of regions, or a parameter that controls the variance within the superpixels. The result is that either some images are overrepresented, containing more superpixels than necessary, or under-represented, containing fewer superpixels than necessary. In addition, many algorithms are unstable as changing the initialisation can drastically alter the result [Tuytelaars and Mikolajczyk, 2007].	Tất cả các phương pháp siêu pixel hiện nay đều đòi hỏi một số dạng khởi tạo. Những dạng này có thể là chỉ định số vùng, hoặc chỉ định một tham số điều khiển variance trong các siêu pixel. Kết quả là một số ảnh có hiện tượng overrepresent, tức là chứa số siêu pixel nhiều hơn cần thiết, hoặc under-represent, chứa số siêu pixel ít hơn cần thiết. Thêm vào đó, nhiều thuật toán không ổn định vì thay đổi khởi tạo có thể ảnh hưởng đáng kể đến kết quả [Tuytelaars and Mikolajczyk, 2007]. variance : sự sai lệch, không ăn khớp, phương sai
To combat this, the superpixels developed here are allowed to evolve through the image in order to develop into the 'best' superpixel representation without constraining them through initialisation. The overall scenario for the CDS approach is that a set of seed points is initialised on an image, shown in Figure 2.1. Given this initialisation, the aim is to determine the superpixels which are derived by content driven analysis. On a blank image there is no content and the result would be a number of superpixels equal to those used for initialisation. For an image with content, the superpixels will adapt in order to represent the content faithfully; large structures (areas of similar colour) aim to be represented by a small number of large superpixels and small structures to be represented by a large number of smaller superpixels. Accordingly, the new approach evolves from the initialisation to the final representation. This requires a growth stage when the areas of similar content are to be determined. Given that the representation adapts to content, this predicates a division stage when change in	[redacted]

content is encountered. The choice of the number of seeds and their location does not to significantly affect the result. This is demonstrated in Section 2.6.2 .	[redacted]
FIGURE 2.1: Initialisation of CDS on an image, shown in red. A superpixel is defined as the set of pixels over which it has grown. Growth adds new pixels to this set; division creates new superpixels by making new sets that correspond to the newly segmented pixels. The algorithm is seeded in a regular grid pattern, where a single pixel is used to initialise each superpixel.	[redacted]
2.1 Growth Phase Growing a set of superpixels independently is a difficult task as there is no information passed between each superpixel. With no information, each superpixel will grow to fill the same space as another superpixel, which is obviously inefficient. Several methods are explored below to solve this problem. These are presented along with the chosen method: the Distance Transform. 2.1.1 Active Contour Model The work by [Kass et al., 1987, Cohen et al., 1990] on parametric active contours is suitable to adapt for superpixel growth. Cohen and Cohen devised a method to make active-contours grow outwards by inserting a normal force (which they called a balloon force) into the snake evolution equation, given in Equation 2.1, that would suit superpixel growth. The response of the snake to the elasticity and curvature control mechanisms are represented by a(s) and (s)	[redacted]

respectively, pn determines the normal force and v(s) describes the contour.

[redacted]

Figure 2.2 illustrates the calculation of the normal force of each point. Contour points s0 and s2 are used to calculate the force on s1, which then acts outward from s1.

One modification must be made to the original snake to use it for superpixels. This is an additional term that relates to the proximity of other superpixels, to avoid the superpixels overlapping as they grow. This additional constraint is shown in Equation 2.2, where Esp represents the combination of all other superpixel boundaries in the image; analogous to image edges. This could be combined with the image energy however including an additional energy term allows different weighting to be applied. The edges of each superpixel are treated as image edges such that each superpixel will attempt to adhere to the edges of neighbouring superpixels.

Esnake = / Eint(v(s)) + Eimage(v(s)) + Eran(v(s)) + Esp(v(s))ds (2.2)
Js=0

[redacted]

As the Cohen balloon forces the contour to expand, avoiding superpixel overlap becomes more difficult. The points on the snake become further apart and so there is an increasing amount of information between these points that gets ignored, shown in Figure 2.3, where the point in red appears within another shape. To maintain stability with large contours, new points are added to the snake as it grows. The main problem with this approach is that it is slow to check for superpixel overlap at an increasing number of points and with an

[redacted]

increasing number of superpixels. FIGURE 2.3: Illustrating the effect of contour overlap (a) t (b) t + St FIGURE 2.4: Overlapping snakes by dual-superpixel growth Figure 2.4 shows a problem that occurs when growing two superpixels, labelled red and blue, close to one another with their joint boundary shown in purple. Despite the additional term in Equation 2.2 the superpixels do not identify each other as edges and so they willingly grow into one another. This is an inherent problem with this style of active contour and is difficult to overcome without exhaustively checking for the existence of other adjacent contours; increasing the computational load of the algorithm. Many people have moved away from this implementation for segmentation due to its inability to adhere to edges when the initialisation is too far from the desired contour. Like all explicit contour methods the Kass snake suffers from topological changes, which is the major argument for using more developed, and complex, methods such as Gradient Vector Flow (GVF) [Xu and Prince, 1997], level-set implementations [Osher and Sethian, 1988] and geodesic active-contours [Caselles et al., 1997]. 2.1.2 GVF field One way to alleviate the problem of contour overlap could be to extend the capture range of the contours. The most well-established method for doing so is Gradient Vector Flow (GVF) [Xu and Prince, 1997]. GVF replaces the image force Eext with a vector field that pulls the snake toward strong features from a long distance. The snake evolves according to Equation 2.3 where v represents the vector field, x represents the	[redacted] [redacted]

contour and a and P are consistent with the parameters in Equation 2.1. The vector field can be solved using a finite-difference method [Xu and Prince, 1998].

$x_t(s,t) = ax (s,t) + Px (s,t) + v(s)$ (2.3)

There are however some changes to be made to the original formulation if this is to be used for superpixels. Instead of contours growing to find objects, contours will grow freely unless in the presence of another contour. To achieve this, the vector field only acts outward from the contours, such that any overlap with other contours is seen as an edge during evolution. Figure 2.5 and Figure 2.6 illustrate this. To do this efficiently and without heuristic derivations, the normal gradient to the contour is computed for all points on each contour, then the subsequent gradient field is diffused, to increase the capture range of the field. This process is given in Equation 2.4. Here, u and v represent the two components of the vector field at iteration n. The rate of diffusion is controlled by r and should not exceed the Courant-Friedrichs-Lewy limit [Xu and Prince, 1998]. u0,v0 represent the initial conditions, that is, the sum of components i, j of the normal nk to each superpixel Ck.

When computed for all superpixels, this has the effect of imposing reduced growth in the presence of additional contours and unimpeded growth otherwise. This method replaces the balloon force that was previously required for contour growth.

FIGURE 2.6: Zoom on the superpixel interface

GVF fields work well in theory however in practice there is one flaw. The fields created

[redacted]

[redacted]

by each superpixel have to perfectly balance out else one superpixel will ‘push against’ the other causing it to retreat. What occurs then is that the superpixel that advances then finds image variation and divides, creating a new superpixel. Were this to be used, this process would then repeat, causing numerous superpixels to occur erroneously.

2.1.3 Distance Transform

2.1.3.1 Introduction

When controlled by image properties, as has been the case so far, superpixel growth is difficult to control and produces initialisation problems. If the superpixel is designed to halt growth on reaching image features, it is unlikely to cover the entire image without at least estimating the number of superpixels required prior to computation. The minimum number of superpixels is also limited by the initialisation. Hypothetically, one would need to know the number of superpixels and an estimate of optimum location (see, for example, [Levinshtein et al., 2009]). Therefore, the best way to cover the image without requiring a-priori knowledge is not to consider image properties during superpixel growth. By only considering superpixel boundaries, each superpixel can grow unimpeded unless in the presence of another superpixel.

To grow superpixels without considering the image, the distance transform is con-sidered. The distance transform [Borgefors, 1986] can be considered equivalent to binary dilation from mathematical morphology but can be computed significantly faster. In morphology, the object is successively eroded until it disappears, with the value of each pixel corresponding to the number of iterations until that particular pixel disappeared.

It is typically used for skeletonisation of

[redacted]

[redacted]

[redacted]

[redacted]

image objects where the skeleton is described by transforming each object such that it displays the distance of object pixels to its boundary with the background.

The algorithm transforms an image to show the distance to a specified colour. This can be used in binary images by showing the distance to either state. The transform also returns an array that determines the closest location in the image that matches the specified colour or state.

2.1.3.2 Use of the distance transform on superpixels

To grow a superpixel, a distance transform of every superpixel that exists is taken. The superpixels are not transformed individually to eliminate superpixel overlap. This is illustrated later in Figure 2.8.

The algorithm transforms each superpixel S such that the set of pixels at each location (i, j) within the superpixel display the distance D to the background (in

FIGURE 2.7: Binary Dilation operator on a superpixel shape. The grey pixels indicate where the superpixel will grow.

this case, the region in which superpixels have yet to form). Superpixel edges therefore have a distance of one from the background. This is shown in Equation 2.5, where the locations (k, l) form the set of points that constitute the background. The image I used to calculate the distance transform is a binary image where True denotes that a superpixel covers this point in the image and False otherwise. The background is therefore all the False points in this image. The same image is used to individually grow each superpixel.

Equation 2.6 shows the iteration, t, of the superpixel to include the background

[redacted]

[redacted]

[redacted]

[redacted]

location (k, l) that is adjacent to the superpixel edge. By only considering the pixels that have a distance of one from the background, superpixel overlap is handled implicitly. Any pixel inside the superpixel or adjacent to another superpixel is not connected to the background and hence the superpixel cannot grow at these locations. As the distance transform is stable for any shape, the superpixels can grow from any initial size and shape. The stopping criterion is a direct consequence of the algorithm; terminating once the superpixels cannot grow any further. This occurs when they are completely bordered by other superpixels or image boundaries.

……………………………………………

Figure 2.7 shows how the binary dilation works on a superpixel. The black pixels

FIGURE 2.8: Dilation operator on two neighbouring superpixels. The red and blue pixels show the two different superpixels. The lighter pixels show the new pixels for each superpixel. The outline shows the current envelope of the superpixel area.

represent the original shape, the grey pixels are the parts to which it grows in the next iteration and the white pixels are the background.

Figure 2.8 demonstrates how this extends to two neighbouring superpixels. The two superpixels shown in opposing colours, red and blue, only grow where they are not bordered by another superpixel. The outline shows the current envelope of both of the superpixels. There is a single purple pixel in the image to show the potential overlap of two superpixels in close proximity. As each superpixel grows independently, both superpixels will occupy this location after growth. Post processing is performed to remove these overlapping pixels from one of the conflicting superpixels.

[redacted]

[redacted]

[redacted]

[redacted]

2.2 Division Phase

As superpixel growth does not take the image information into account, it is required that the superpixel division method must take the image into account in order to fully describe the image. What follows is a comparison of several methods of segmentation available to divide superpixels.

[redacted]

2.2.1 Graph Partition

Using the N-cuts algorithm it would be possible to partition the superpixel using a pixel-based graph created within the superpixel. Weights are applied to graph edges according to the Euclidean colour distance between nodes. The minimum cut through the edges can be taken in order to segment the superpixel. However, segmentation will always occur if there is not a minimum weight, so a superpixel containing one colour will divide into two superpixels of the same colour. This is an undesirable property. If at each iteration the superpixel area increases and is subsequently halved, it will be very difficult to achieve much less than pixel resolution.

[redacted]

2.2.2 Watershed

The watershed algorithm [Meyer and Beucher, 1990] ‘floods’ an image from a chosen number of local minima until the sources meet. The boundaries of these independent floods become the segmentation boundaries. Usually this would require knowledge of the number of regions required, however binary segmentation would be possible given two sources. This method would always create two regions irrespective of image content, so a way of sensing variation or a bi-modal distribution would have to be produced in order to trigger the segmentation. The

[redacted]

[redacted]

location of the initialisation would also significantly affect the result for example if the two minima were close together or near the edge of the superpixel.

2.2.3 Region Growing

Similarly to the Watershed algorithm, Region Growing involves growing a single pixel from local minima until no new pixels can be added. It will only add pixels that are similar to pixels already contained in the region. However, Region Growth can suffer from initialisation variation. One initialisation can give a vastly different result to another. This can be seen in Figure 2.9. As the algorithm only adds similar pixels, growth stops once the colour boundary is reached. The result is that the algorithm is susceptible to noise. One advantage over the Watershed algorithm is that it would only produce two superpixels if there was a boundary created within the superpixel. If the algorithm were to reach the superpixel boundary without stopping then the superpixel is homogeneous with respect to the splitting criterion.

(a) Image under test (b) Top left (c) Centre (d) Bottom right

FIGURE 2.9: Region growing initialisation problem, with the seed point marked in red. Only the centre seed gives the correct result.

2.2.4 Local Active Contours without Edges

As superpixel division (via segmentation) is occurring at a small scale, more sophisticated segmentation algorithms become viable. Using, for example, a kernel based on Mumford and Shah [1989] leads to a situation where region-based segmentation algorithms can be used to generate new superpixels. Active Contours Without Edges (ACWE) cannot normally be used in complex images due to its creation of only two regions: object and background.

[redacted]

[redacted]

[redacted]

However in this case, region- based segmentation becomes an ideal solution.	[redacted]
A benefit of ACWE is the addition of localised smoothing introduced by the approach. This helps to restrict superpixel division; a necessary requirement due to the greedy nature of the algorithm. In addition, division will not occur if the colour of the superpixel is uni-modal.	[redacted]
2.2.4.1 The Basics of ACWE	[redacted]
Active Contours Without Edges (ACWE) aims to partition an image u0 into two piecewise-constant intensities of distinct values u and u2. These piecewise regions are separated by a boundary c0 such that Equation 2.7 is minimised. ………………………….	[redacted]
where F describes the force from inside and outside the contour and ci, c2 are the averages of the regions inside and outside the contour. It can be easily seen from this that if the boundary lies outside c0, then F1(C) > 0 and F2(C) =0. If the ………………………	[redacted]
FIGURE 2.10: Illustrating the evolution of the boundary [Chan and Vese, 2001]	[redacted]
boundary is inside c0 then F\(C) = 0 and F2(C) > 0. This is shown in Figure 2.10.	[redacted]
Next some regularising terms are added to control the length of the contour and the area of the region inside C. This can all be represented by an energy function F in Equation 2.8, where p, v, Ai, A2 are weighting co-efficients. …………………..	[redacted]

2.2.4.2 Level Set Method

ACWE can be solved using level sets. Equation 2.9 introduces 0 as a function of the image, and defines the contour C as the points in the image H where 0 = 0. The points inside the resulting contour are denoted by u.

C = 5u = (x, y)eH : 0(x, y) = 0 inside(C) = u = {(x,y) E H : 0(x,y) > 0} outside(C) = H/u = {(x,y) E H : 0(x,y) < 0

Using the Heaviside function H, and the Dirac 5 function, we can write F as in Equation 2.10 where u0 is the image and c1, c2 are the averages as described in 2.11. ^,v and A are positive parameters.

…………

where

(2.11)

2.2.4.3 ACWE for Vector Valued Images

ACWE can easily be extended [Chan et al., 2000] to any size of vector for each pixel by averaging over the vector length. For example, over the three colours in RGB space. This is shown in Equation 2.12.

………..(2.12)

2.2.4.4 Adaptation for Local Area

Chan-Vese segmentation of superpixels, shown in Figure 2.11, works by considering two regions u, v that form the positive and negative parts of a signed distance function, . A force F iteratively updates the distance function (Equation 2.13) such that each pixel is 'moved' toward the region it best matches by adding the force F to the surface 0. The new superpixels, Cu, Cv, are taken to be the positive and negative parts of 0.

In this application, the problem is further simplified by considering only a subset of the image: the area within the superpixel.

[redacted]

[redacted]

[redacted]

Considering these smaller regions makes the problem tractable as an iterative algorithm. This is achieved by the inclusion of a binary function S(x,y) that is greater than zero when inside the superpixel and zero otherwise. In addition, the length and area constraints have been removed as the smaller area of the superpixel does not require them. Both weighting parameters A are set to one to give equal weight to either side of the contour.

[redacted]

The final force equation is given in Equation 2.13, where t denotes the iteration of 0. The updated average calculation for c1,2 is given in Equation 2.14, where the same constraint on S(x,y) is applied.

(a) Before (b) Contour plot show- (c) Contour plot showing (d) After
ing the initialisation of the the final distance function distance function

[redacted]

FIGURE 2.11: Illustrating the mechanism of division. The superpixel is divided by ‘moving’ the distance function causing pixels to be either positive or negative.
(a) Before labelling
(b) After labelling

[redacted]

FIGURE 2.12: The split in the segmentation is corrected by using connected component labelling. The green superpixel is the new superpixel.

2.2.4.5 Choice of Initialisation

ACWE is initialised by setting the distance function 0 to be a function in the range of ±1. When considering only a small region of the image, it is expected that initialisation will be important, and that the faster it is possible to arrive at the best solution, the faster the algorithm will perform. To select

[redacted]

the best initialisation, a set are tested on a modulated cosine signal (Figure 2.13(a)) to observe the amount of incorrect labelling.	[redacted]
As ACWE can be realised in N-dimensions, a 1D signal is used to observe the results in a concise way. The chosen initialisations to test are: • 0[x] = 0 (Figure 2.13(b)); • 0[x] = 1 (Figure 2.13(c)); • 0[x] = —1 (Figure 2.13(d)); • 0 is alternately ±1 (Figure 2.13(e)); • 0 is the signal, normalised to the range ±1 (Figure 2.13(f))	[redacted]
As the chosen signal varies uniformly around zero, the resulting segmentation should be to separate the signal into two signals either side of zero. The orientation of the result does not matter (some are flipped), what is important is that the two extracted regions (shown in different colours) correspond to all of the correct points in the signal. It is important to make the distinction that the values of 0 are not the image values, they are the values of the distance function that are used to segment the signal. Figure 2.13 shows the results of this test where red circles denote error. The only two results without error are the cases where 0 = 0 and where 0 relates to the signal. However, normalising the image over the range ±1 will force large changes in 0 that should not necessarily exist in like-pixels. This could force unnecessary superpixels and remove the smoothing effect of ACWE. Consequently, the initialisation will be 0 = 0.	[redacted]
2.3 Control	[redacted]
CDS is designed not to require controlling parameters as the idea is to create the best possible reconstruction of the image.	[redacted]

However, one can still influence the result in the following two ways. Firstly, the level of detail detected by the algorithm can be controlled by smoothing the image using a simple Gaussian filter with standard deviation a. This still retains the larger image variation however details such as facial features are missing. Figure 2.14 shows this effect. Adding the smoothing removes all facial features and treats the face as a single superpixel. Also, the brim of the hat still exists but the rest of it is merged with the grass behind it. The clothes are also merged into a single superpixel. Secondly, the number of superpixels can be controlled by the initialisation of the superpixel seeds. The number of seeds is the minimum possible number of superpixels that are to be generated. This has a small effect on the result as the final shape and distribution might be different, however the same features will be detected regardless of the initialisation, and the reconstruction quality is unaffected.	[redacted]
2.4 Final method By using a modified Distance Transform in conjunction with localised ACWE, there are numerous benefits. The Distance Transform allows implicit handling of an arbitrary number of superpixels of any size and any shape. Allowing them to grow unimpeded by image properties ensures total image coverage from any initialisation. New superpixels are handled from within the superpixel, as they	[redacted]
(a) Original image with no smoothing, a = 0 (b) Original image with smoothing, a = 2 (c) Reconstruction, a = 0 (d)	[redacted]

Reconstruction, a = 2 FIGURE 2.14: Showing the effects of Gaussian based control	[redacted]
are formed when the superpixel is no longer uniform in colour, irrespective of size. The combination of these two algorithms is such that they can produce accurate superpixels that are stable and will not overlap under any condition.	[redacted]
A pseudocode implementation of the algorithm is given in Appendix B.	[redacted]
2.5 A Discussion on Region Merging There is a temptation once the superpixels have been generated to remove the artificial boundaries produced where superpixel seeds meet. This could be achieved by merging them where the colour distance between neighbours is small. This is perhaps useful to further reduce the ‘resolution’ of the superpixels.	[redacted]
FIGURE 2.15: Borders that exist due to seeds meeting, with some of the superpixels that could be merged marked with green dots The ability to merge regions will not be developed for several reasons. The first is that if one reduces the number of superpixels, the superpixel representation no longer reflects the image content as there would have to be a threshold in order to merge superpixels. The second reason is that this would only really affect the borders between seed superpixels. Theoretically, borders only exist between distinct superpixels and that merging them again appears counter-intuitive to the original reasoning behind the algorithm.	[redacted]
Removing borders between seed superpixels, shown in Figure 2.15, actually removes a very small percentage of superpixels. The final reason is that this step is tantamount to clustering,	[redacted]

and if this is the desired effect then it could easily be achieved via k-means or other such algorithms, using the superpixels as a pre-processing step. An initialisation of one superpixel would solve this minor problem, but it will require additional processing time.

[redacted]

Figure 2.15 also shows that the superpixel growth algorithm causes anisotropic boundaries to occur. This is a consequence of the distance transform and does not affect the quality of the reconstruction so it was not altered to produce isotropic boundaries.

2.6 Analysis

2.6.1 Method of Analysis

Superpixels are only useful if they can capture relevant image information. As the focus is on image representation, evaluation must focus on evaluating the ability to reconstruct an image by losing as little information as possible. Reconstruction is defined as the process of replacing the colour of each pixel (x, y) with the mean of the superpixel ^(x,y) it is contained in, given in Equation 2.15

1 (x,y) = ^i(x,y) (2.15)

[redacted]

The results come from the test set from the Berkeley Segmentation Dataset (BSDS) [Martin et al., 2001]. BSDS includes human segmented annotations of the original images, typically five for each image. Given that the labels are much larger than a typical superpixel, each label will contain multiple superpixels. Mode label analysis is introduced in order to identify undersegmentation error, the average proportion of each superpixel that matches the modal annotated label. In addition, the percentage of label boundaries that match superpixel boundaries, the boundary recall rate, is computed. These are weighted such

[redacted]

that borders included by all subjects are stronger than those occurring in one image. What is not computed is the boundary precision rate. This is the percentage of superpixel boundaries that match label boundaries. Using recall rate alone has been considered sufficient in previous research and gives a good result even if many superpixels edges do not occur at label edges. This is because the focus in previous research has been on avoiding undersegmentation rather than oversegmentation.

[redacted]

Oversegmentation: phân đoạn dư, phân đoạn quá nhiều, phân đoạn quá nhỏ

[redacted]

The use of metrics on this database is not sufficient. During the generation of BSDS, subjects were instructed to make sure all labels were of equal importance and size within the image. Consequently, small detail could easily be falsely attributed to incorrect or insufficient labelling. To provide a measure independent of the human labels, the ‘explained variation’ Moore et al. [2008] is calculated, providing a measure of superpixel accuracy, which helps to evaluate superpixels that do not correspond to the human-labelled edges. It calculates how accurately the mean of each superpixel matches the pixels within by calculating the variation about the global mean. This is given in Equation 2.16, where x* represents the pixel value, ^ is the mean of the superpixel containing the pixel x* and ^ is the global mean of the image.

R2 = EniPi ^ (2 16)
E*(x - P)2 (.)

Explained variation can be considered a measure of undersegmentation. If the average colour of a superpixel does not accurately represent the pixels within it, that region is undersegmented.

Many analysis techniques, such as explained variation, favour more rather than less superpixels. While it is recognised that this is often not the main aim of superpixels, results can be improved if more superpixels are present to capture the higher levels of image variation. The extreme of this being of course the case where each superpixel represents a single pixel. Taking oversegmentation into account removes the emphasis on just creating more superpixels to capture more information. The emphasis is then on creating superpixels to capture more information only when required to do so by the image properties. As such, oversegmentation can be considered analogous to a measure of superpixel precision. As CD superpixels split on colour differences, oversegmentation can be measured by the Euclidean distance between the mean colour value of each connected superpixel averaged over all connections. If this value is low, then the average distance between superpixels is small, and therefore superpixels are less distinct in colour, implying that oversegmentation exists. This measure is given in Equation 2.17 where (r, #,6)e[0,1] represent the colour of connected superpixels i, j. C represents the sum of all superpixel connections and c represents a single connection.

E /(ri - r,)2 + (gi - g,)2 + (b, - b,)2

d = - C (2.17)

The results generated in this chapter are compared against the algorithms in [Felzenszwalb and Huttenlocher, 2004](FH) and [Ren and Malik, 2003](N-Cuts) as these are well established techniques. As our new algorithm does not directly control the number of superpixels, all comparisons are achieved by using the output from CD superpixels to specify the equivalent

[redacted]

[redacted]

parameters in the other algorithms. To assess the quality of our algorithm, results are generated on each image using varying levels of Gaussian smoothing. This allows a comparison to be drawn as the number of superpixels changes.	[redacted]
Finally, the compression ratio is defined as the ratio of pixels to superpixels. Plotting this ratio instead of the explicit superpixel numbers helps to put the result into perspective irrespective of the image size. A low compression ratio indicates	[redacted]
(a) Initialisation A: 9 superpixels; result: 8871 super-(b) Reconstruction using initialisation A. pixels.	[redacted]
(c) Initialisation B: 36 superpixels; result: 7248 super-(d) Reconstruction using initialisation B. pixels.	[redacted]
FIGURE 2.16: Illustrating the difference between two different initialisations arranged in an evenly spaced grid. Despite the difference in the initialisation the reconstruction is hardly affected.	[redacted]
a high number of generated superpixels. As shown in Figure 2.16, even though the output can be directed by the initialisation, the reconstruction is largely unaffected despite the difference in resulting superpixels. In addition, all results are similarly affected by smoothing the image. For this reason, the initialisation will not be changed while testing the effects of smoothing. However, the invariance to initialisation is tested. This invariance is tested on a single image using random Gaussian perturbations of the grid pattern at varying levels of standard deviation.	[redacted]
The algorithm is also assessed when introducing increasing amounts of Gaussian noise.	[redacted]

Compression Ratio

FIGURE 2.17: Showing the difference in colour between superpixel neighbours as a function of superpixel compression.

2.6.2 Results

Figure 2.17 shows the colour difference between neighbouring superpixels as a function of the number of superpixels in the image over the test set of images from BSDS. A lower value indicates that the colour difference is smaller, or in other words, that the regions are more likely to be oversegmented as two neighbouring superpixels could be represented as one. N-cuts performs badly on this test because as the number of superpixels increases, the oversegmentation increases. This is not unexpected as superpixels generated in this way are designed to be of similar size, meaning that large regions of one colour will contain the same number of superpixels as a much more complex region. CD superpixels, however, has an almost uniform response despite the number of superpixels. This means that as the number of superpixels increases and the representation of the image improves, the colour difference between superpixels remains constant. When specifying lower numbers of superpixels for N-Cuts and FH, the latter performs best. Again this could be due to the implicit nature of superpixels generated by Felzenswalb.

Figure 2.18 shows the relationship between oversegmentation and undersegmentation. N-cuts and FH both show, to varying degrees, that as the explained variation of the result increases (how well is the image reconstructed by superpixels), the difference in colour between those superpixels decreases, leading to oversegmentation.

FIGURE 2.18: Showing the difference in colour between superpixel neighbours as a

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

function of explained variation. CD superpixels remains almost constant, irrespective of the quality of reconstruction. This means that increasing the quality of result for CD superpixels does not lead to large amounts of oversegmentation as would normally occur.	[redacted]
A sample image is taken from all three algorithms at the same Explained Variation value denoted by the dotted line in Figure 2.18. These three images are shown in Figure 2.19. N-Cuts clearly contains too many superpixels, particularly in the central area surrounding the person. FH does a better job in the centre of the image. The windows in the right hand tower are well detected, yet the trees contain significant undersegmentation, as does the person in the centre. CDS performs well, particularly in the trees and the person in the centre yet, like N-Cuts, does not detect the windows on the right.	[redacted]
Figure 2.20 illustrates three metrics: recall rate; explained variation; and mode label. In Figures 2.20(a) and 2.20(b), CD superpixels perform well, but only at high numbers of superpixels (low compression). Recall rate (Figure 2.20(c)) clearly shows N-cuts to perform best by almost 20% at low compression. Soon after, all three algorithms are largely equivalent until high compression at which point CD superpixels suffers.	[redacted]
FIGURE 2.19: Example images all shown at the same value for Explained Variation shown in Figure 2.18	[redacted]
FIGURE 2.21: Showing how recall rate, mode label and explained variation all vary as the initialisation is perturbed by a Gaussian random variable of standard deviation a. The ability of CD superpixels to provide a constant measure of oversegmentation throughout is an interesting property of the	[redacted]

algorithm. It is thought that it can be explained by the use of ACWE as a splitting algorithm. As ACWE uses colour differences to divide, and almost all neighbouring superpixels have occurred through division, the constant difference in colour must be attributed to how ACWE separates a region. The benefit of this is improved stability when generating superpixels.

CD superpixels clearly suffer with regard to recall rate. As the control mechanism comes from smoothing the input image, the edges and detail in the image degrade with more smoothing. This reduces the likelihood of an image boundary matching a superpixel boundary. This problem occurs for all metrics plotted in Figure 2.20 however colour difference does not suffer for the same reason described above.

One further important point is that the human labelling used for recall rate and mode analysis is subjective. Labels are drawn on the assumption that they are equally important to the user. While this is partially accounted for by the averaging, some important image information is ignored. For this reason, superpixel quality should not be exclusively assessed by low-resolution labelling. The use of explained variation is intended to address this issue.

Figure 2.21 shows the results of perturbing the initialisation by a random Gaussian variable of increasing variance. Explained variation and modal label vary little, having standard deviations of 0.3 and 0.01 respectively. Recall rate is the only metric that varies by more than one percentile at 2.4. This is still a small variation and is attributed to one result only, at a = 2. Colour difference is not assessed as the previous experiment has shown it is almost constant.

The results in Figure 2.21 show there is high stability in the algorithm. As CD superpixels

[redacted]

[redacted]

[redacted]

are parameterised on image properties, the number of superpixels is forced to adhere to the image and consequently, there is little possibility of the superpixel arrangement deviating. This helps to resolve one major problem of superpixel algorithms: that they are unstable due to initialisation parameters.

Figure 2.23 shows the effect of noise on the algorithm. The noise was generated by using a Gaussian random variable of increasing variance centred on the value of each pixel. This will make the image tend more toward a completely random image where no structure is present. The variance is controlled between a =10 and a = 70 such that the image is still at least partially visible.

This shows that the reconstruction accuracy reduces as the image tends toward being completely random.

One would expect that as the image becomes less structured that it is more difficult for CDS to extract regions of uniformity. CDS has an inherent averaging process, which in most images has little effect. Using images where the colour changes frequently and unpredictably clearly affects the reconstruction quality. However, the explained variation is still 75% in the presence of significant noise. This can be seen in Figure 2.22 where the visual quality of the result appears to have been improved by the presence of noise. The reason for this is that there are more superpixels triggered by the presence of more variance within local areas. So the result is more visually pleasing, yet the oversegmentation has been increased.

2.7 Conclusion

This chapter has developed and demonstrated an algorithm that can successfully and, more importantly, reliably reconstruct an image using superpixels. The results also show that the instability of

[redacted]

[redacted]

[redacted]

previous superpixel algorithms has been reduced by parameterising superpixels not by number but by image complexity.

(b) Reconstruction at a2 = 0

(c) Image at a2 = 70

(d) Reconstruction at a2 = 70

FIGURE 2.22: Examples of the effect of noise on an image and its reconstruction.

FIGURE 2.23: Quality of reconstruction as noise increases

This improves the invariance to initialisation as all metrics used vary by less than 3%.

The performance of Content-Driven Superpixels is dependant on image complexity. The more complex an image is the better the performance of CDS. This might seem counter-intuitive yet it is linked to the use of ACWE as a division method. As the colour information within the superpixel is averaged, bigger superpixels are more likely to contain larger colour differences. A smaller superpixel is more sensitive to small variation in colour as it makes a larger contribution to the average. This averaging also causes problems if two areas vary significantly yet their means are similar. This manifests itself when dealing with noisy images, as shown in Figure 2.23. Superpixels will not divide under this circumstance and this reduces the reconstruction quality.

In addition, this chapter developed the previously overlooked concept of explicitly measuring oversegmentation to better evaluate superpixels. It was subsequently shown that CDS oversegments less than other well-used algorithms.

In general, superpixels remain to have their uses truly explored. The ability to reduce image complexity whilst retaining high-level features is highly desirable in many areas of

[redacted]

[redacted]

computer vision and the next few chapters explore some potential applications.	[redacted]