International Journal of Mathematics and Soft Computing
Vol.3, No.3. (2013), 115 - 128.

ISSN Print : 2249 – 3328
ISSN Online: 2319 – 5215

Using a Level-Set Model to Generate Artwork from Digital Images

Joseph W Wilder
Department of Mathematics,
The University of Akron,
Akron, OH 44325, USA.

Abstract

The use of mathematics and computers in the generation of art is an ever growing field. This work concerns the development of an algorithm based on the solution of a partial differential equation that can be used with digital images to generate art-like drawings. This modified model uses a conceptual framework that includes local and global perspectives to determine which aspects of the objects in the image should be included in the drawing. This novel inclusion of both of these perspectives results in drawings with enhanced art-like qualities compared with those utilizing a single perspective.

Keywords: Level set; image processing; Hamilton-Jacobi equations.

AMS Subject Classification (2010): 68U10.

1 Introduction

Image processing has been a highly active field of research for over four decades. The vast majority of the techniques employed in this area that have focused on image recognition and computer vision have their roots in the methods devised by Sobel and/or Canny (for a good general introduction to standard edge detection techniques, see [8] and the references therein). These and many associated methods depend on local changes in the image intensity to indicate the presence of an edge or boundary related to some object in the image. Most of these techniques utilize image gradients (first derivatives) or Laplacians (second derivatives) to identify object boundaries in the image. By determining all locations where edges are perceived, one can reconstruct the outline of the objects contained in the image. Usually, there are multiple processes applied to the image data to improve the final, reconstructed, outline. For example, once the edge “strength” is estimated (using the gradient or Laplacian), thresh-holding is applied to determine which of the potential edges will be interpreted as “real” edges. Following this step, methods such as edge thinning are applied to remove spurious points on the edge of the perceived objects. In addition, the image is often filtered to remove noise before the edge strength is even calculated. Different approaches to these three phases (filtering, edge strength estimation, and thresh-holding) and their combinations have led to the large number of different techniques that can be applied to extract edges from digital images. Many of the programs available today for working with photos and other types of images include tools that utilize these (or very similar) techniques to transform the original image into a line drawing. For example, Figure 1 shows an example of applying the edge detection tool of the Paint.net software to the image on the left to generate the line drawing shown on the right. As can be seen in Figure 1, such tools are quite good at the task of identifying edges in this type of image. In examining these results, it is also clear that the output is easily recognizable as a computer generated line drawing based on the original photograph: it would never be mistaken for the work of a human artist. While there may be differences in opinion about what constitutes "art", most would agree that such computer generated line drawings would not qualify for this designation. One possible reason for this is that human artists use their skills and perception of the image as a whole to select those portions that are in some way key to conveying the essence of the image, without getting lost in reproducing the minutia that software packages typically tend to focus on. To anthropomorphize the process, such tools lack any perspective from which they can judge the relative importance of one feature versus another, and so can only reproduce all of the features in minute detail. A human artist, on the other hand, would seek to include the key elements, often ignoring some small features while including others, leading to suggestions of details that would catch the human imagination, allowing the observer to interpret the work as a whole instead of a large number of small pieces. For a computer program to mimic such behavior, it would also need some way to assess the relative importance of features: it would need to examine each potential edge in the context of the global image instead of just utilizing local information to identify all potential edges. One approach to edge detection that has been proposed in recent years is based on global assessments, and has at its core the use of differential equations to seek out and find these globally important edges. These methods are called "active contour models". This work presents a modification of one of these models to incorporate both global and local information to allow for the generation of line drawings that appear to have some of the same qualities as those one would expect in the work of a human artist.

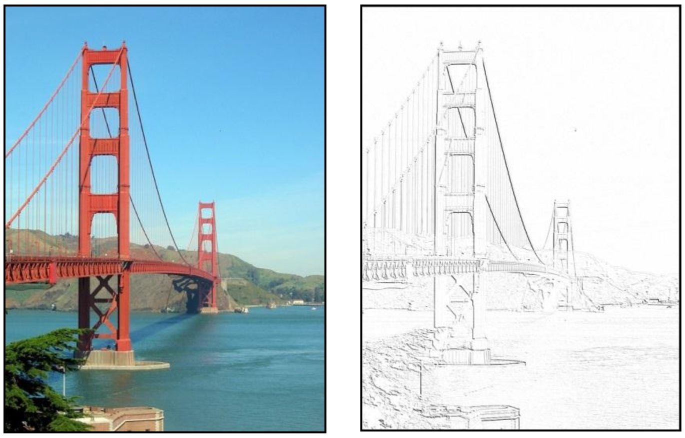

Figure 1: *A photo of the San Francisco Bridge and the result of standard edge detection.*

2 Background

As mentioned above, a relatively new approach to edge detection involves what are known as active contour models (often referred to as "snakes"). In this approach, a curve is laid on top of the image and evolved using criteria determined by the image properties. Based on pre-determined criteria, parts of the curve that come in contact with edges of objects in the image stop moving, while others continue to migrate until all the edges have been found. This method, developed in [6] has been successfully applied in various settings. The original method, as introduced by Kass et al incorporates an edge detector based on gradients that stops the curve evolution when an edge is encountered (the details of how methods like the one used in this work identify edges will be discussed in more detail below). In a series of papers beginning by Chan and Vese [1-5] a different model was proposed that used the Mumford & Shah [7] segmentation technique to stop the evolution when an edge was encountered (instead of the originally proposed use of the gradient of the image at a specific point, Mumford Shah techniques are based on using global image data, as will be discussed in more detail below). The advantage of such a model is that edges not associated with gradients (for example those arising as the result of very smooth objects or ones that have discontinuous boundaries) can also be detected.

Mathematically, one way to formulate the problem of determining the location of the edges in an image is to frame it in terms of a functional that needs to be minimized [6]. Put simply, minimizing the functional (which is merely a function that has a vector (the colors at each pixel) as its argument and which returns a scalar) results in the selection of those points that are determined to lie on an edge based on some predetermined set of edge-detection criteria (which are used to build the terms of the functional). In the approach proposed by Chan and Vese [2], the functional to be minimized (called the "energy") is constructed using a level-set formulation. Level set methods have been applied to many areas in recent years (for a good reference/introduction to these topics, see [9]). These methods are used in problems where there is a curve (a so-called "implicit surface") that evolves over time. When these methods are applied to edge detection, the curve represents the previously mentioned "snake" that is moving around in the image seeking out edges. Once it finds an edge, the part of the curve (snake) that is in contact with the edge stops moving. In this approach, an implicit function (ϕ) is defined everywhere in the domain of interest (the entire image), with the interface/boundary of interest (the edges in the image) defined as the locations where $\phi = 0$. As discussed in the seminal work of Osher and Sethian [10], the motion of the implicit surface is governed by an associated partial differential equation (PDE), solutions of which are found by using Hamilton Jacobi methods. In general, the level set problem involves solving PDE's of the form:

$$\frac{\partial \phi}{\partial t} + \vec{V} \cdot \vec{\nabla} \phi = 0 \qquad (1)$$

where $\vec{V}$ is the velocity field that drives the motion of the implicit surface ($\phi = 0$). The form of $\vec{V}$ varies greatly with the problem of interest, and often depends on ϕ, making the differential equation nonlinear. In the edge detection case, the form of this velocity depends on the choice of the "energy" that needs to be minimized. The details of the development of the level-set formulation for the edge detection problem can be found in the literature [2-5] and will not be reproduced here. Instead, we will present this formulation as it was originally presented, and then discuss its use and further modification to make it more useful in the generation of art-like depictions based on digital images.

3 The Original Model

Consider a set of gray-scale image data $u_0(\vec{x})$ where $\vec{x}$ is the pixel location (inside the domain Ω that defines the image) and u_0 the color value at the indicated point. In order to determine the edges in this image using the level-set formulation, one needs to pick an energy functional to be minimized that will stop the motion of the surface (or "snake") when it runs into points in the image that satisfy the desired characteristics. The energy used by Chan and Vese differs from that known as the Mumford-Shah functional. If we consider the curve C to be some contour in the image, this new energy is of the form [2]

$$F(C,c_1,c_2) = \mu\big(length(C)\big)^p + \nu \cdot area(inside\ C)$$
$$+\lambda_1 \int_{inside(C)} |u_0 - c_1|^2 dx \ + \ \lambda_2 \int_{outside(C)} |u_0 - c_2|^2 dx \tag{2}$$

where the integrals in Equation (2) are over the two dimensional areas interior/exterior to the curve C. The first term is included to allow for applications where one is concerned with constraining the total length of the bounding curve (with the value of p being unity except with very noisy images, in which case one can use larger values to heighten the importance of constraining the total curve length). The second term in Equation (2) can be used in cases where one wants to ensure that the snake only moves in the direction of the area inside (or outside) of that bounded by C. For example, one might only be interested in finding the edges inside/outside of the initial curve used as the starting point of the snake, and so would want to restrict its motion to the desired direction. The last two terms In Equation (2) are such that the minimization of the functional requires a C that minimizes F when c_1 and c_2 are the averages of the color values inside and outside, respectively, of the area bounded by C (the nature of these terms is further explored in a simple example discussed following Equation (4)). As derived in [2], the specific form of this functional in the level set formulation where $\phi = 0$ is the desired boundary (or in this application, outline of an object) is given by:

$$F(C,c_1,c_2) = \mu\left(\int_\Omega \delta_\epsilon(\phi)\left|\vec{\nabla}\phi\right| d\,\vec{\mathrm{x}}\right)^p + \nu \int_\Omega H_\epsilon(\phi) d\,\vec{\mathrm{x}}$$
$$+\lambda_1 \int_\Omega |u_0 - c_1|^2 H_\epsilon(\phi) d\,\vec{\mathrm{x}} \ + \ \lambda_2 \int_\Omega |u_0 - c_2|^2 (1 - H_\epsilon(\phi)) d\,\vec{\mathrm{x}} \tag{3}$$

where c_1 and c_2 are the average u_0 values in those regions where $\phi \geq 0$ and $\phi < 0$ (respectively), and are given by:

$$c_1 = \frac{\int_\Omega u_0 H_\epsilon(\phi) d\,\vec{\mathrm{x}}}{\int_\Omega H_\epsilon(\phi) d\,\vec{\mathrm{x}}} \quad \text{and } c_2 = \frac{\int_\Omega u_0 (1 - H_\epsilon(\phi)) d\,\vec{\mathrm{x}}}{\int_\Omega (1 - H_\epsilon(\phi)) d\,\vec{\mathrm{x}}} \tag{4}$$

Note that in the above equations, $H_\epsilon(\phi)$ is any C^2 regularization of the standard Heavyside function and $\delta_\epsilon(\phi) = H'_\epsilon(\phi)$, while $p, \mu, \nu, \lambda_1,$ and λ_2 are non-negative model parameters (which will be used to select certain desired characteristics of the to-be-identified edges).

To better understand the nature of the last set of terms in Equation (3), consider the following simplified situation. Assume that we are not interested in constraining the length of the curve, nor its direction of motion. Also assume that we are dealing only with one color channel, and that we select $\lambda_1 = \lambda_2 = 1$. In this simplified case, Equation (3) becomes:

$$F = \int_{inside(\phi \geq 0)} |u_0 - c_1|^2 dx + \int_{outside(\phi < 0)} |u_0 - c_2|^2 dx \quad (5)$$

For illustration purposes, let us further assume that the image of interest is made up of regions that have one of two color values (such as the black and white regions in Figure 2a). If we think of putting the curve $\phi = 0$ inside (but not on the boundary of) the black region of Figure 2a, then (assuming "black" has a color value of $u_0 = 0$ and "white" has one of $u_0 = 1$), the average color value inside the $\phi = 0$ contour will be 0, while that outside of it will be some value between zero and one (due to that region containing a mixture of black pixels and white pixels). Looking at the first term in Equation (5), we see that this integral is over points inside and on the $\phi = 0$ contour, and so $u_0 = 0$ at all of the relevant points because these pixels are all black. Similarly, because c_1 is the average color value inside and on the contour, we see that $c_1 = 0$. As a result, $|u_0 - c_1|^2 = 0$ everywhere inside and on this contour, and so the first integral in Equation (5) would be zero. Looking at the second integral, however, we see that (because there are some black and some white areas outside the $\phi = 0$ contour) c_2 is neither zero nor unity, but rather some value in-between these two extremes. Since at each point outside the contour the color value is either zero or unity (and therefore never equal to c_2), we see that $|u_0 - c_2|^2 > 0$ at all of the points relevant for this integral. Therefore, the second integral is positive. Combining the two integrals we find that the value of *F* would be greater than zero if we select the $\phi = 0$ curve to be completely contained in the black region of Figure 2. Had we selected the location of the $\phi = 0$ contour to be entirely in the white region, the situation for the two integrals would have been reversed, with the result that *F* was once again positive. Only if the $\phi = 0$ contour lies exactly on the boundary of the black and white regions will *both* integrals in Equation (5) be zero, leading to *F* = 0. Thus we see that minimizing the functional F corresponds to selecting the $\phi = 0$ contour to be at the boundary of the differently colored regions in this simplified example. While we only examined a very simple example, similar analysis in the more general case also leads to the conclusion that methods that are designed to minimize this functional will lead to the identification of the boundaries of the objects in the image.

If one wishes to consider color images (ie., those with more than just a single color channel), u_0 can be considered to be made up of 3 separate channels (red, green, and blue), and the last two terms in the energy functional can be replaced with:

$$\sum_{i=1}^{3} \left(\lambda_{1,i} \int_\Omega \left|u_{0,i} - c_{1,i}\right|^2 H_\epsilon(\phi) d\vec{x} + \lambda_{2,i} \int_\Omega \left|u_{0,i} - c_{2,i}\right|^2 (1 - H_\epsilon(\phi)) d\vec{x} \right) \quad (6)$$

where $c_{1,i}$ is the average value of channel i in the region where $\phi \geq 0$, while $c_{2,i}$ is the average value in the region where $\phi < 0$. Minimization of the resulting regularized functional yields an Euler-Lagrange equation for ϕ (see [5] and references therein for details). This new equation can be used as the basis for an evolution (in time) equation where the $\phi = 0$ level set curve seeks out the edges in the image. As a result of the stopping criterion built into the equation, the steady state solution of the PDE given in Equation (7) is also the solution of the original Euler-Lagrange equation for ϕ and therefore minimizes the desired functional (see [2] for details):

$$\frac{\partial \phi}{\partial t} = \delta_\epsilon(\phi)\left[\begin{array}{c} \mu p\left(\int_\Omega \delta_\epsilon(\phi)\left|\vec{\nabla}\phi\right| d\,\vec{x}\right)^p \vec{\nabla}\cdot\left(\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}\right) - \nu \\ + \sum_{i=1}^{3}\left\{-\lambda_{1,i}\left(u_{0,i} - c_{1,i}\right)^2 + \lambda_{2,i}\left(u_{0,i} - c_{2,i}\right)^2\right\} \end{array}\right] \tag{7}$$

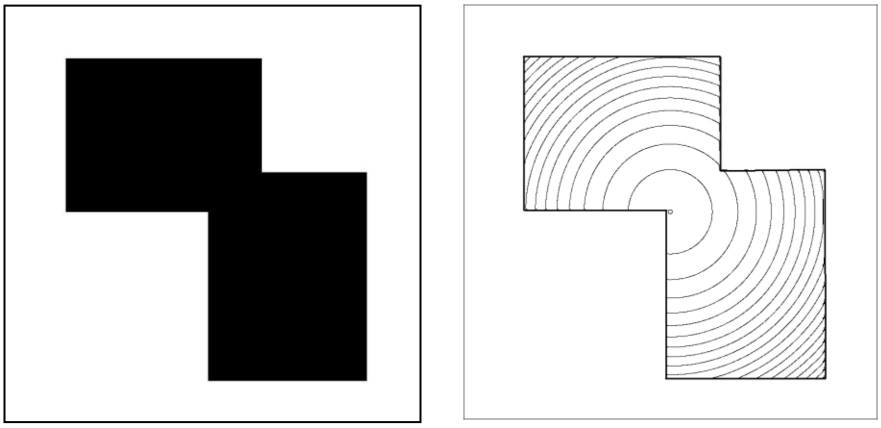

Figure 2: (a) *A simple figure* (b) *Evolution of the $\phi = 0$ starting from the initial small (central) circle and ending with the boundary between the black and white regions of* (a).

Following [2], the choice for the form of the regularized Heavyside and delta functions that will be used in this work are:

$$H_\epsilon(\phi) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\epsilon}\right)\right);\ \ \delta_\epsilon(\phi) = H'_\epsilon(\phi) = \frac{1}{\pi}\left(\frac{\epsilon}{\epsilon^2 + \phi^2}\right) \tag{8}$$

where ϵ is a small parameter (chosen to be the size of the numerical mesh used to solve the PDE for ϕ (Equation 7)). Note that when working with images, the "mesh" size is determined by the dimensions of the image (its height and width in pixels).

Using the finite differencing approach outlined by Chan and Vese [2], Equation (7) can be used along with digital images to search for edges in these images. While this method has proven successful in various applications related to edge detection for image recognition problems, it has not been applied to the use we have in mind in this work: the construction of line drawings from digital images and an assessment of their art-like nature. Before we examine this application, it is useful to review the components of Equation (7) and how they might impact these constructions.

Recall that the evolution of the "snake" represented by the $\phi = 0$ level set is controlled by the PDE given in Equation (7). While this evolution in itself is interesting, and can be used to generate animations similar to the opening of an iris with respect to the viewing of the final image, it is the steady-state solution (the one for which $\frac{\partial \phi}{\partial t} = 0$) that represents the final work that will depict all of the edges found in the image. Therefore, Equation (7) is solved numerically until the solution reaches this steady state, which is controlled by the parameter values used in evaluating the right-hand-side of Equation (7). For example, setting $\mu = 0$ removes any constraint on the total length of the edges, while a nonzero value adjusts the relative importance of the total length of this contour with respect to the other constraints placed on the final figure (we shall examine the effects of this particular constraint below). The second term in Equation (7), as discussed previously, is only used in applications where one wants to control the direction of motion of the $\phi = 0$ level set during the evolution of the solution (we are not interested in doing so in this work, and so we will always use $\nu = 0$). The last set of terms are those (as discussed above) that ensure that the $\phi = 0$ level set found from the steady state solution lies on the boundaries of objects in the image. Because we will be looking at images that involve multiple color channels, we need to recall that the relative size of the λ values controls the relative importance of the separate color channels when identifying object boundaries. For example, an image that contained objects and backgrounds only made up of shades of blue would have no boundaries (edges) as far as Equation (7) is concerned if the λ_i values associated with the blue color channel were all set to zero. As a result, the identification and strength of the edges detected in an image can depend on the selection of the λ_i values. Noting that for each color channel there are two λ_i values (applied to the interior and exterior of the contours), and that we have no reason to favor one region over another, we will use $\lambda_{1,i} = \lambda_{2,i}$. Lastly, we will normalize the relative importance of the color channels such that $\sum_{i=1}^{3} \lambda_{1,i} = \sum_{i=1}^{3} \lambda_{2,i} = 1$. Having made these general remarks about Equation (7), we will now move onto looking at specific cases to see how the model performs.

4 Results using the original (global) model

To begin our exploration of the use of Equation (7) to generate art-like drawings from images, we will first apply it to the simple example of the object in Figure 2a to demonstrate that it performs as expected in this trivial example. Figure 2b shows the time evolution of the initial $\phi = 0$ contour at a time of zero (the small circle near the center of the diagram) into the final (steady state) $\phi = 0$ contour (the outermost curve). Note that these (equally spaced in time) contours migrate outwards until they intersect an object boundary. When they make contact with an edge, the part of the $\phi = 0$ contour that is in contact with the boundary of the object ceases to migrate, while the rest of the surface continues to evolve. Once it has locked onto all of the edges in the image, the contour ceases to change (leading to the final (darker) contour). Clearly, the method performs as expected, and quite easily finds the edges in this simple image.

The next step in testing this model is attempting to utilize the same method with a more complex image, for example the one in Figure (1). Because this image contains multiple color channels, one must decide the relative importance of these channels, and what values of λ to use. Figure 3 shows the results of using each color channel (red, green, or blue) separately, as well as a commonly used mixture when converting color images to greyscale ones ($\lambda_{red} = 0.30,\ \lambda_{green} = 0.66,\ \lambda_{blue} = 0.04$). As can be seen, the relative importance of the color channels makes a difference in the selection of the object boundaries in the image, with some leading to final drawings that contain more information about the objects. While some choices lead to drawings that contain significant detail, none of those depicted in Figure 3 (nor any explored by the author) led to the identification of those boundaries/edges that would result in a line-drawing that would be considered art-like in the sense considered later in this work. It should be noted at this point that, as discussed by Osher and Fedkiw [9], there are more advanced implementations of the level set method that have been designed for applications involving image segmentation. While use of these more complicated models might result in the identification of more edges in the images than the simpler version presented here, none would result in the type of art-like line drawings that are the goal of this work. Why is this the case, and why is the use of Equation (7) inferior (at least in some respects) even to the edge detection method utilized by standard packages (such as that used to generate the image on the right in Figure 1)? The answer to this question is fundamental to our quest to find an edge-detection method useful in the generation of artwork that is based on these level-set methods, and can be found by returning the very nature of these methods.

As the reader may recall, classical edge detection methods are based on local data: the gradient or Laplacian of the image data at a point is used to determine if an edge is located in that region. This potentially results in each and every edge-like location being identified as an object boundary in the image, and in all of them being drawn on the canvas. The current method, however, is based on global data, not local data (recall that c_1 and c_2 are the averages over the entire image inside/outside the zero level-set contour, and so are global rather than local in nature). This use of global measures means that a potential edge must be “strong enough” in a global sense to show up on the final canvas. If, for example, the color change at what our eye might perceive as an edge is small enough with respect to the global changes in that color, the algorithm might not select it as an edge in the global context because it is searching for a solution that minimizes the functional (which, due to being based on integrals, is global, not local, in nature). As a result, selection of the global minimum may require these “weak” edges to be ignored. If edges, by their very nature, are local phenomena, one might then ask the question why we would want a global method in the first place? The answer to this can be seen in Figure 1. The fact that local methods result in nearly every edge being identified results in drawings that, while “photo-realistic” in nature, lack a certain artistic quality than can arise from the judicious inclusion/exclusion of details (which is what artists do based on their assessment of what is key to conveying the essence of the picture). In order to assess which features should be included and which excluded, one needs to examine the image from a global perspective (weighing the value of various pieces with respect to the image as a whole), while not losing sight of the importance of (at least some) local details. The global nature of Equation (7) allows for assessment of the importance of each feature from the global perspective, but it lacks the input of local data. A human artist would look at the image as a whole to assess the key elements of what s/he wants to convey through the canvas, but would then look regionally at areas within the image to assess which local features need to be included to support those key elements. To emulate this inclusion of both global and local influences on the final selection of lines to be drawn on the canvas, we must modify our model to include a regional component, as will be discussed in the next section.

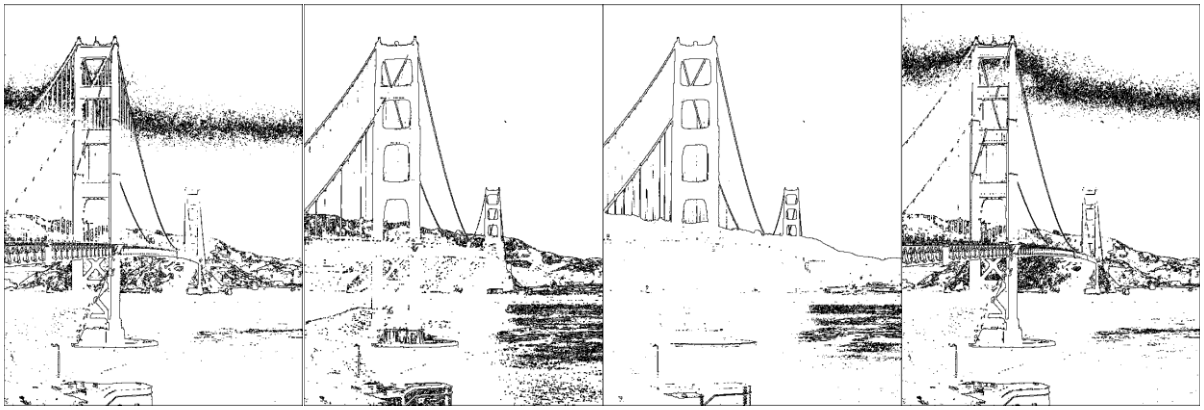

Figure 3: *Drawings generated using only the red, green, blue, or a mixture of the channels.*

5 Development and discussion of a new model based on global and local data

As has been seen, the original model which was based solely on global data did not allow for the consideration of features that are only locally important that potentially are critical to constructing art-like drawings from digital images. Based on an idea concerning how a human artist would use both an overall perspective and a local one in determining the importance of various features in the subject of their art work, we want to modify the model so that it uses input from both global and local data. Similar to what a human artist would do, we want the method (in addition to looking at the image as a whole) to look at a small region surrounding each point when deciding whether to include an object boundary at that point as part of what is to be drawn on the canvas. To achieve this, Equation (9) shows a modification of Equation (7) using a weighted sum of local and global terms based on the color value averages (locally and globally) in the image:

$$\begin{aligned}\frac{\partial\phi}{\partial t} = \delta_\epsilon(\phi)\Big[\mu p\left(\int_\Omega \delta_\epsilon(\phi)\left|\vec{\nabla}\phi\right| d\,\vec{x}\right)^p \vec{\nabla}\cdot\left(\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}\right) - \nu \\ +\beta_{global}\sum_{i=1}^{3}\left\{-\lambda_{1,i}\left(u_0 - c_{1,i}\right)^2 + \lambda_{2,i}\left(u_0 - c_{2,i}\right)^2\right\} \\ +\beta_{local}\sum_{i=1}^{3}\left\{-\lambda_{1,i}\left(u_0 - c_{1,i}^{local}\right)^2 + \lambda_{2,i}\left(u_0 - c_{2,i}^{local}\right)^2\right\}\Big]\end{aligned} \tag{9}$$

In Equation (9), the two new weighting parameters are chosen such that $\beta_{global} + \beta_{local} = 1$ (allowing the user to assign the relative importance of the local and global perspectives), and $c_{1,i}^{local}$ is the average value for color channel i inside the zero level set contour in a small region surrounding the point under consideration (with $c_{2,i}^{local}$ being the corresponding local average outside the contour). The size of the local region we want to use to assess the importance of a specific feature could be anything from the entire image (in which case one recovers the original model in Equation (7)) down to a single pixel. The choice of the size of this local region corresponds to the resolution at which one wants to look when determining the relative importance of a local feature. Similarly, the weighting of the global and local terms determines how much focus the "artist" should give to regional (local) issues when determining how important a feature is to the overall image. Based on numerical explorations, it was found that including the pixel of interest as well as its eight closest neighbors was sufficient to result in the effective use of the local data to assess feature importance. As a result, in the remainder of this work, a small (nine pixel) region will be used in computing the local terms near each point in the examined images. Clearly, the weighting of the global and local terms (the β_{local} and $\beta_{global} = 1 - \beta_{local}$ parameter values) will be critical in the determination of the final drawing. To assess the impact of this weighting, we will now look at the application of Equation (9) to Figure (1) using the same color mixture that led to last pane in Figure (3). Figure (4) shows the results for $\beta_{local} =$ 0%, 25%, 50%, or 75% (note that the 0% case reproduces the same drawing as that resulting from the use of Equation (7) shown in Figure 3 [right panel]).

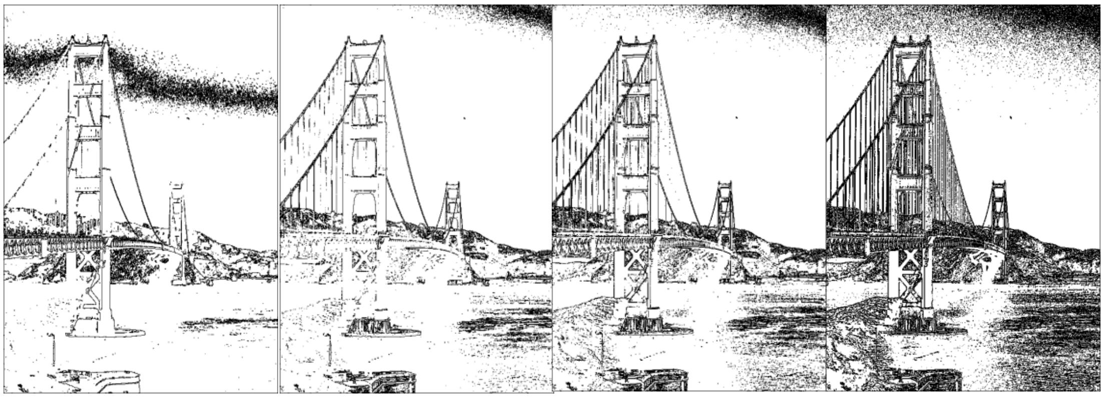

Figure 4: *Final drawings resulting from the variation of the relative importance of the local data: 0% (left), 25%, 50%, 75% (right).*

Clearly, the inclusion of the local data impacts the quality of the final drawing, and in the 75% case results in one with qualities that, at least in the opinion of this author, are "art-like". It is instructive to recall at this point that these figures are the result of solving a differential equation. While those of us who are involved in modeling are used to using differential equations to examine such things as the mechanical motion of a bridge due to natural vibrations and/or external forces, it is, for some reason, far more surprising (at least to this author) to realize that they can also be used to interpret (in an artistic manner) images of that same bridge. That such an equation can not only be used to find boundaries of objects in an image, but can also assess the relative importance of those boundaries with respect to the image as a whole is a unique and fascinating application.

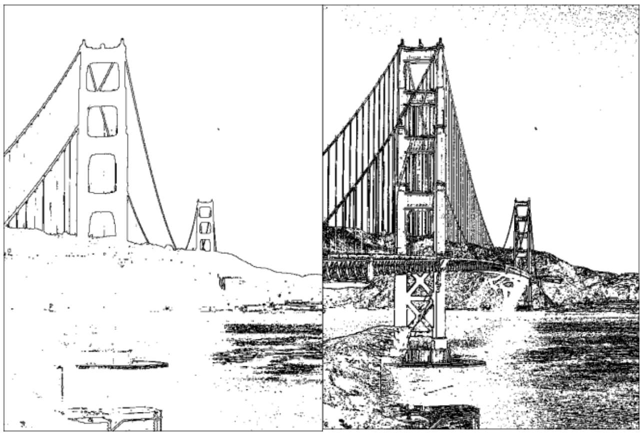

Figure 5: *Results of using only global data (left) and including local data (right).*

The impact of using local data on the final image shown in Figure (4) is critical to changing a not-very-complete representation of object borders into an art-like drawing. This transformation is even more impressive in other cases, such as when one uses a different mixture of the color channels. For example, if instead of the mixture of color channels used in Figure (4) (which was $\lambda_{red} = 0.30$, $\lambda_{green} = 0.66$, $\lambda_{blue} = 0.04$) we use an equal mixing of the colors ($\lambda_i = \frac{1}{3}$), the drawing obtained ignoring local data (left-hand panel in Figure 5) is dramatically improved by using a 75% local data weighting (right-hand panel in Figure 5).

Based on the results we have shown here, a natural question is whether one should use an even higher weighting for the local data: perhaps even just use local data. After reflection, however, it can be easily seen that if one went to the extreme of only using local data, and did so for a local region that was only one pixel in size, then EVERY pixel would end up being a boundary in the image, and the entire canvas would be black. If we use a slightly larger local area (say the 9 pixel area used in this work), then similar considerations would lead the algorithm to pick out some features (at least one) in that small region that it would draw as boundaries, and the result would again be a very noisy picture. Only if the algorithm has something against which to judge the importance of the boundaries in that small region can it assess whether they are important enough in the global sense to be included in the final product. Thus, while one may use more or less weighting of the local data in a particular case, it is critical to always give some weighting to the global data so that the algorithm has this broader perspective as a guide in constructing the overall image.

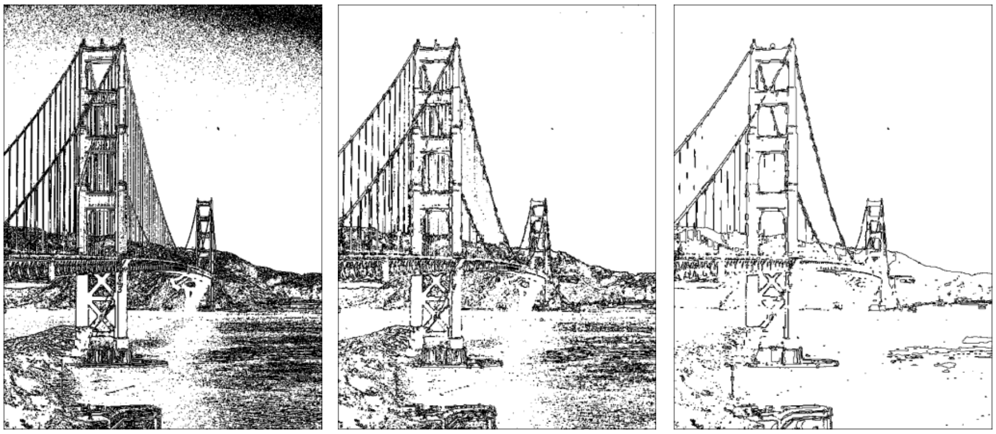

Figure 6: *Restriction on the total length of all the edges: $\mu = 0.0$ (left), 0.001 (center), 0.1 (right).*

While the case we have looked at so far has been enlightening (and led to a very engaging, final drawing), this analysis of the use of Equation (9) would not be complete if we did not look at the impact of limiting the total amount of ink available to complete the drawing, as well as the application of the algorithm to various types of images, especially some that have "softer" edges than something like the steel beams and cables making up a bridge. Further exploring the use of Equation (9), we next examine the impact of nonzero values of μ (recall that this parameter multiplies the term involving the total length of the $\phi = 0$ contour, and so increasing its relative importance in Equation (9) corresponds to decreasing the total amount of ink available to draw the selected edges). As can be seen in Figure (6), increasing the importance of this term forces the algorithm to assess which features it sees as critical to the image due to no longer having an unlimited supply of ink to draw as many edges as would be drawn without this constraint. Interestingly, even in the most limited case shown in Figure (6), many of what even a human artist might think of as the key parts of the image have been retained. It is intriguing to think that perhaps what we perceive when we look at something may also be dependent on an internal assessment of the importance of edges based on both local and global perspectives, conceptually not that different from what is being done by Equation (9).

We will conclude the examination of this new algorithm by looking at its application to several other types of images. Figure (7) examines its performance when applied to faces (those of two famous women), where we see that even in the absence of the "sharp" edges common in man-made objects like the focal point of Figure (1), the algorithm still generates drawings that many would consider "artistic" in nature. Figure (8) shows the results of processing photos of various subjects (a cityscape, a tree, and a still-life). In the author's opinion, these computer generated drawings all convey a quality that is more reminiscent of what is typically meant by "art" than are figures generated with typical edge detection software (such as that in Figure 1).

6 Conclusion

In this work we have explored the modification of a previous edge-detecting model based on level set methods. It has been demonstrated that by including both local and global data the resulting algorithm generates canvases that are engaging and art-like in nature. The resulting drawings are different in their overall feel and content from those resulting from either the original (global) model or standard (local) edge detection algorithms. While there are other algorithms for generating various types of art-like works from digital images, the method considered in this work is unique with respect to its utilization of a differential equation based model, as well as its use of both local and global image data in determining which features to include in the final drawing.

Figure 7: *Drawings generated using Equation (9) based on photos of the Mona Lisa* (*left*) *and actress Angelina Jolie* (*right*).

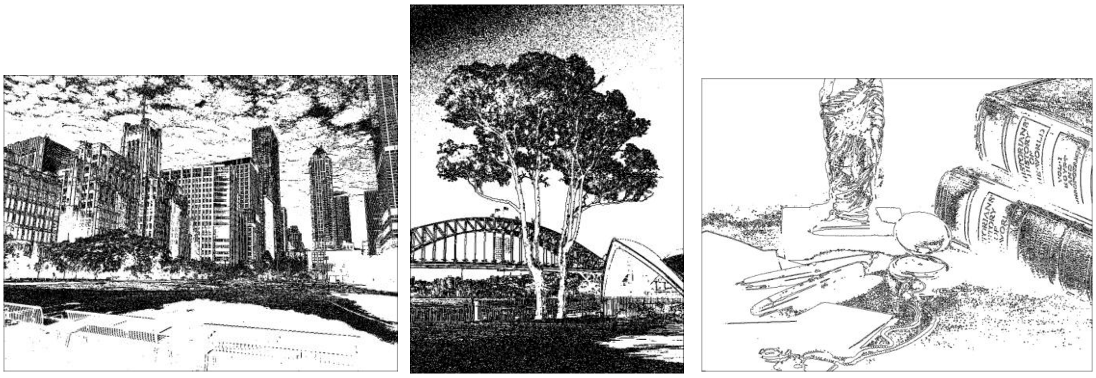

Figure 8: *Drawings generated using Equation* (*9*) *based on photos of the Lake Shore Park in downtown Chicago, at Northwestern University* (*left*), *a view of the Sydney Harbour Bridge and the Sydney Opera House from the Government House lawn (center) and a simple still life* (*right*)

References

[1] T. Chan and L. Vese, *An active contour model without edges, in Scale-Space Theories in Comupter Vision,* Springer-Verlag *Lecture Notes in Computer Science*, No. 1682, M. Nielsen, P. Johansen, O.F. Olsen and J. Weickert, eds., Proceedings of the Second International Conference, Scale-Space '99(1999), 141-151.

[2] Chan and L. Vese, *Image segmentation using Level Sets and the Piecewise-Constant Mumford-Shah Model*, UCLA CAM Report (2000), 1-14.

[3] T. Chan and L. Vese, *Active contours without edges,* IEEE Transactions on Image Processing, 10 (2001), 266-277.

[4] T. Chan and L. Vese, *A level set algorithm for minimizing the Mumford-Shah functional in image processing,* IEEE/Computer Society Proceedings of the 1st IEEE Workshop on 'Variational and Level Set Methods in Computer Vision (2001), 161-168.

[5] T. Chan and L. Vese, *Active Contour and Segmentation Models using Geometric PDE's for Medical Imaging in Geometric Methods in Bio-Medical Image Processing*, R. Malladi, ed., Series: Mathematics and Visualization, Springer (2002), 63-75.

[6] M. Kass, A. P. Witkin and D. Terzopoulos, *Snakes: Active Contour Models*, International Journal of Computer Vision, 1 (1988), 321-331.

[7] D. Mumford and J. Shah, *Optimal approximations by piecewise smooth functions and associated variational problems,* Comm. Pure Appl. Math. 42 (1989), 577-685.

[8] MS Nixon and A.S. Aguado**,** *Feature Extraction and Image Processing*, 2nd Edition. New York : Elsevier LTD, 2008.

[9] S.J. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*, Applied Mathematical Sciences 153, Springer Verlag. New York, 2003.

[10] S.J. Osher and J.A. Sethian, *Fronts propagating with curvature-dependent speed; algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys. 79 (1988), 12-49.