

Finding a Needle in a Haystack: An Image Processing Approach

Emily Beylerian
Advisor: Hayden Schaeffer

University of California, Los Angeles
Department of Mathematics

Abstract

Image segmentation (also known as object/edge detection) is the process of dividing an image into its constituent parts using information about the boundaries between objects, edges within objects, variations in intensity, et cetera. Often, the human eye can easily recognize salient information from an image; however, background variations in intensity, noise and other degradations, and other highly oscillatory features make the process of image segmentation challenging. This work is unique because we propose using a cartoon-texture-noise separation to remove highly oscillatory features from the image prior to segmentation. The cartoon and texture components can be used to analyze important information from the original image; specifically, by applying a segmentation algorithm on the cartoon component, we can extract objects from the original image. A new numerical implementation is provided for one of the two decompositions used as well as various experimental results. The method is applied to the classic example of finding a needle in a haystack, as well as real images where the texture component and noise causes problems for standard techniques.

1 Introduction

The goal of this paper is to use modern imaging techniques to solve the difficult problem of locating a needle within a haystack. Issues in locating a needle in an image of hay exemplify the open problems in image segmentation. Image perspective introduces scales (varying size levels) of the hay, shape similarity makes it difficult to distinguish between needle and hay, and the large number of objects in the image causes many false edges. From the perspective of imaging science, the problem of locating a needle in a haystack can be formulated as locating an object hidden by texture, noise, and occlusion.

Over the past couple of decades, there have been many segmentation models proposed in the literature. A brief summary of a few major models is provided here. In [12], the authors proposed an energy minimization method (known as the Snakes model) to detect the edges in an image. For the rest of this work, we will take f to represent the given image. Then the problem of segmenting an image can be formulated as finding the set of edges in f .

Let Γ be the edge set, then the general energy based segmentation models can be written as $E = P(\Gamma) + R(\Gamma)$, where P is a penalty term that typically involves geometric aspects of the edge

set and R is the region fitting term.

Let $\Gamma(s)$ be a parameterized curve ($s \in [0, 1]$) which defines the edge set, then the Snakes energy is given by the following equation,

$$E_S(\Gamma) := \int_0^1 (\beta|\Gamma'|^2 + \gamma|\Gamma''|^2 - |\nabla f(\Gamma)|^2) ds, \quad (1)$$

where β and γ are strictly positive. To find the curve Γ , the energy is minimized over the set of all possible curves. The first two terms in equation (1) keep the curve smooth, while the last term attracts the curve to high gradients (one definition of an edge). The image f must be treated with a Gaussian filter before applying the Snakes method, in order to remove noise and texture.

The Geodesic Active Contours model [5], is an energy minimization method which is similar to Snakes,

$$E_{GAC}(\Gamma) = \int_{\Gamma} g(|\nabla f(\Gamma)|) ds, \quad (2)$$

where g is a continuous edge detector (0 near sharp gradients and 1 in low gradient regions). The minimizing curve is attracted to regions where the gradient of the image is sharp, while also having small length.

In contrast to the previous methods, the model developed in [6], which we refer to as the Chan-Vese model, uses regional information rather than sharp gradients. The method detects the object boundaries and then reconstructs a two-region piecewise constant image of the detected object(s). The Chan-Vese energy is as follows,

$$E_{CV}(\Gamma, C_1, C_2) = \int_{In(\Gamma)} (f - C_1)^2 dx dy + \int_{Out(\Gamma)} (f - C_2)^2 dx dy + \lambda \int_{\Gamma} ds,$$

where $\lambda > 0$. The method fits two constants (C_1 and C_2) to the given image, while using the minimal amount of the partitioning curve Γ . The Chan-Vese model is a simple case of the Mumford-Shah model [16] :

$$E_{MS}(u, \Gamma) = \mu \int_{\Gamma^c} |\nabla u|^2 dx dy + \int (u - f)^2 dx dy + \lambda \int_{\Gamma} ds. \quad (3)$$

The Mumford-Shah model looks for a smooth u with jumps along Γ which best fits the given image.

In the work we will focus on the Chan-Vese algorithm since it is known to locate local edges. Normally, finding local (as opposed to global) edges is an undesired property in a segmentation algorithm; however, since we are interested in locating a specific object, we would like a method that does not over-segment the image. Rather than filtering out noise like the Snakes or Geodesic Active Contours methods, we separate the image into cartoon and texture components. The contribution of this work is to perform segmentation on only the cartoon component of the image. In section 2, more details on the Chan-Vese algorithm are provided with a description of the Cartoon-Texture decomposition used to precondition the image for the segmentation step. For a needle in a haystack, the hay is removed (as the texture), which makes the needle much more prominent, resulting in better segmentation results. In section 3, the algorithms for the Cartoon-Texture decomposition, the Chan-Vese segmentation and the sequential implementation of these methods are given. In section 4, results are given on the needle in a haystack image as well as other images. We conclude the work in section 5.

2 Description of the Method

We propose a two step method for feature and structure extraction by first removing texture using an image decomposition method, segmenting the cartoon, and then recombining the cartoon and texture (without the noise) in the segmented region. In this section, we first discuss Cartoon-Texture decomposition methods in general and our choice of model and algorithm. We also detail the Chan-Vese segmentation method and the motivation behind its proposed use.

2.1 Cartoon-Texture Decomposition

The Gaussian filter can be thought of as the first cartoon algorithm, since it filters out all high frequency information in the image and leaves only large scale features. Given an image f which can contain degradation, the Gaussian filtered image u is given by

$$u = G_\sigma * f, \quad (4)$$

where $*$ denotes convolution and G_σ is a Gaussian kernel with zero mean and a standard deviation of σ (see [11]). Equation 4 is the solution of the heat equation

$$\frac{\partial u}{\partial t} = \Delta u, \quad (5)$$

at time $t = \sqrt{2\sigma}$ with initial data $u(0) = f$. Therefore, removing high frequency data can be done by a finite time diffusion process. However, this type of process does not retain much of the important information from the original data. On the other hand, adding a forcing term to the right hand side of equation 5 can partially reintroduce information to the diffusion process. This can take on the following form:

$$\frac{\partial u}{\partial t} = \Delta u + \lambda(f - u) \quad (6)$$

for $\lambda > 0$. As the equation moves forward in time, the solution diffuses by the Laplacian term, while the difference terms add back some of the lost information. It is well known that equation 6 provides the gradient flow for the following energy:

$$\begin{aligned} E(u) &= \int (|\nabla u|^2 + \lambda(u - f)^2) \, dx \, dy \\ &= \|u\|_{H^1}^2 + \lambda \|u - f\|_{L^2}^2 \end{aligned}$$

where $\|\cdot\|_{L^2}^2 = \int |\cdot|^2 \, dx \, dy$ and $\|\cdot\|_{H^1} = \|\nabla \cdot\|_{L^2}$ are norms for the function spaces L^2 and H^1 respectively. Thus minimizers of the energy above provide smooth approximations of f .

However, the diffusion term over-smooths the image, removing noise, texture and edges. The reason for this is due to the isotropic nature of the Laplacian, it smooths all information regardless of its local characteristics. In order to preserve edge information and provide a more reasonable cartoon, Rudin, Osher and Fatemi ([18]) proposed the following energy:

$$E_{ROF}(u) = \|u\|_{TV} + \frac{\lambda}{2} \|u - f\|_{L^2}^2$$

where $\|u\|_{TV} := \int |\nabla u| dx dy$ is the total variation of u . Minimizers of this energy are known to be nearly piece-wise constant, where the jumps correspond to edges in the image. This avoids the issue of over-smoothing edges; however, there is still a loss of texture in the reconstructed image.

In [15], special norms were introduced to specifically capture and extract the behavior of texture. With these norms, the problem becomes finding the cartoon u and the texture v which best approximates the given image f by minimizing the new energy:

$$E_{CT}(u, v) = \|u\|_{TV} + \mu \|v\|_{Texture} + \frac{\lambda}{2} \|u + v - f\|_{L^2}^2 \quad (7)$$

where $\mu, \lambda > 0$. The parameter μ plays the role of a texture scale parameter and λ acts as the fidelity parameter between the approximation $u + v$ and the original f .

These norms (which we will denote by $\|\cdot\|_{Texture}$) can be difficult to characterize since at best they are generalized functions (weak functions). For example, in [17] the texture norm is defined as $\|v\|_{H^{-1}}^2 = \|\nabla \Delta^{-1} v\|_{L^2}^2$. To examine this norm more closely, let us look at a one dimensional “texture” $v(x) = \sin(\alpha x)$ and where $\alpha > 0$ $x \in [0, 2\pi]$, then

$$\|\sin(\alpha x)\|_{H^{-1}}^2 := \int_0^{2\pi} \left(\frac{d}{dx} \left(\frac{d^2}{dx^2} \right)^{-1} \sin(\alpha x) \right)^2 dx$$

Since $\frac{d^2}{dx^2} \sin(\alpha x) = -\alpha^2 \sin(\alpha x)$, we can formally invert the differential operator (assuming periodic boundary conditions), thus $\left(\frac{d^2}{dx^2}\right)^{-1} \sin(\alpha x) = -\frac{1}{\alpha^2} \sin(\alpha x)$.¹ Therefore,

$$\begin{aligned} \|\sin(\alpha x)\|_{H^{-1}}^2 &= \int_0^{2\pi} \left(\frac{1}{\alpha^2} \frac{d}{dx} \sin(\alpha x) \right)^2 dx \\ &= \int_0^{2\pi} \left(\frac{1}{\alpha} \cos(\alpha x) \right)^2 dx \\ &= \frac{1}{\alpha^2} \int_0^{2\pi} \cos^2(\alpha x) dx \\ &\leq \frac{2\pi}{\alpha^2} \end{aligned}$$

Thus the norm decreases for large α , which translates to lower energy for high frequency. In terms of images, minimizers of equation 7 with similar texture norms extract highly oscillatory information into the texture component. For more on this type of texture norm see [1, 13, 14, 21].

However, this can be unsatisfactory in practice, since noise is also highly oscillatory. Models which use this type of texture norm are unable to properly denoise images, therefore we choose to use two texture norms which are not inversely proportional to the frequency. The first is the L^1 norm which is defined as $\|\cdot\|_{L^1} := \int |\cdot| dx dy$, which was used for texture modeling in [22]. The second, more recent model is the patch-rank norm of [19], which is defined $\|\cdot\|_T := \|\mathcal{P} \cdot\|_*$, where \mathcal{P} maps a matrix into another matrix whose columns are all the non-overlapping patches (blocks) in the original matrix and $\|\cdot\|_*$ is the sum of the singular values of the matrix. The

¹In general, we can invert this second order differential operator for sufficiently smooth functions with the appropriate boundary conditions.

number of non-zero singular values gives the rank of a matrix, which corresponds to a measure of the repetitive, or patterned, texture in the image.

Comparing the L^1 texture to the low patch-rank texture model, there are two main differences. The first is that the L^1 texture model works on scale while the low patch-rank model works on pattern. The second is that, for noisy images, the low patch-rank model is better suited for texture-noise separation. This is useful in images where the texture and noise are on the same scale. On the other hand, when there is little noise, the L^1 model converges to a decomposed solution faster.

In either case, let us define $\|\cdot\|_t$ to be the texture norm (either L^1 or low patch-rank), then it can be shown that for a given f the following holds:

$$\|f\|_t \leq c\|f\|_{L^2} \leq C\|f\|_{TV}$$

which gives a natural ordering between the norms (where $c, C > 0$). The texture norm is bounded by the noise norm which is bounded by the cartoon norm. This is important since following through the theory of texture norms in [9], one can show that for certain parameters μ and λ , thin objects can be considered part of either the cartoon or texture component. Of particular interest is that the width scale can be used as a characteristic of the decomposed components, see [9]. This motivates the use of Cartoon-Texture Decomposition for the separation of a needle from a haystack. For more on applications of texture extraction see: [2-4, 8, 9].

2.2 Chan-Vese Segmentation Method

From an input image f , the Chan-Vese method will detect boundaries of objects in the image and use the boundaries to reconstruct a two-phase piecewise constant image. Ideally, for our application, the method would detect the needle (or main object) as one region and the complement as the second region. Recall that the method minimizes the energy:

$$E_{CV}(\Gamma, C_1, C_2) = \int_{R_1} (f - C_1)^2 \, dx \, dy + \int_{R_2} (f - C_2)^2 \, dx \, dy + \lambda \int_{\Gamma} ds$$

For the remainder of the paper, we set the region inside of Γ as R_1 and the region outside of Γ as R_2 . By minimizing the length term, the resulting curve Γ will remain smooth. The first two terms are error terms between the constants (C_1 and C_2) and the given image. Minimizing the error terms ensures that the intensities of the pixels in each region are similar to the constant assigned to that region. We will also denote the first term as E_1 and the second term as E_2 .

To see why the minimizers give an appropriate segmentation we provide a simple example. Figure 1 shows that the only Γ that minimizes both of the error terms is that which lies on the boundary of object. The average intensity value inside the curve in Figure 1(a) is much greater than the intensities at each of the light gray pixels in R_1 , yielding a large error term in R_1 . Similar principles apply to Figures 1(b) and 1(c). In Figure 1(d), the average intensity in each region is equal to the intensity values at each pixel in the respective regions.

One problem in segmentation is that the variables are of different types: curves versus functions. This is avoided by using the level set method. Let ϕ be a Lipschitz function where $\phi(x, y) > 0$ if (x, y) is inside Γ (in R_1), $\phi(x, y) < 0$ if (x, y) is outside Γ (in R_2), and $\phi(x, y) = 0$ on the curve Γ . Let $H(z)$ be the Heaviside function, which returns 1 if $z \geq 0$, and returns 0 if $z < 0$. With this definition, we have the regional indicator functions $H(\phi)$ and $1 - H(\phi)$. Also from [7], the length of the curve is given by

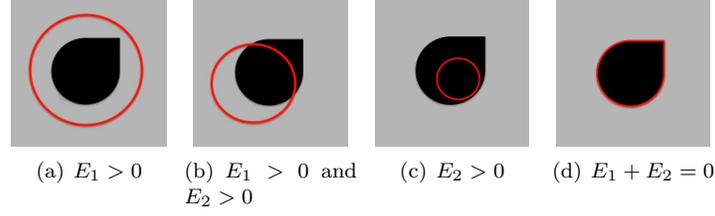


Figure 1: The curve Γ shown in red. In each image except 1(d), $E_1 > 0$ or $E_2 > 0$ or both. In 1(d), $E_1 = E_2 = 0$ because $C_1 = f$ at each point in region 1 and $C_2 = f$ at each point in region 2.

$$\begin{aligned} \text{Length}\{\phi = 0\} &= \int |\nabla H(\phi)| \, dx dy \\ &= \int \delta(\phi) |\nabla \phi| \, dx dy, \end{aligned} \quad (8)$$

where $\nabla \phi$ is the gradient of $\phi(x, y)$ and $\delta = H'$. Using equation 8 and the indicator functions, the Chan-Vese energy becomes:

$$E_{CV}(\phi, C_1, C_2) = \int (\lambda \delta(\phi) |\nabla \phi| + H(\phi)(f - C_1)^2 + (1 - H(\phi))(f - C_2)^2) \, dx dy \quad (9)$$

Minimizing equation 9 with respect to the constants C_1 and C_2 is easily done and has the following form:

$$C_1 = \frac{\int H(\phi) f \, dx dy}{\int H(\phi) \, dx dy} \quad (10)$$

$$C_2 = \frac{\int (1 - H(\phi)) f \, dx dy}{\int (1 - H(\phi)) \, dx dy} \quad (11)$$

Equations 10 and 11 are the regional averages. The smooth approximation used for $H(\phi)$ is given by $H_\varepsilon(\phi) = \frac{1}{2} + \frac{1}{\pi} \arctan(\frac{\phi}{\varepsilon})$ and the smooth version of the delta function is taken to be $\delta_\varepsilon = H'_\varepsilon$, which are strictly positive and bounded above by 1. This avoids dividing by zero in the equations 10 and 11. Lastly, the minimizing equation in terms of ϕ is given by the steady state of:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left(\text{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \lambda (-(f - C_1)^2 + (f - C_2)^2) \right).$$

3 Algorithm

3.1 Cartoon-Texture Decomposition

The L^1 texture energy is written as

$$E_{CT, L^1}(u, v) = \|u\|_{TV} + \mu \|v\|_{L^1} + \frac{\lambda}{2} \|u + v - f\|_{L^2}^2 \quad (12)$$

Since the Cartoon-Texture energy is convex in u , convex in v , and L^1 like norms, we can use the Split Bregman method [10]. The method splits the equation from a non-linear problem to a linear problem and a shrink function. First we can write the two subproblems as:

$$u^{n+1} = \arg \min_u \|u\|_{TV} + \frac{\lambda}{2} \|u + v^n - f\|_{L^2}^2 \quad (13)$$

$$v^{n+1} = \arg \min_v \mu \|v\|_{L^1} + \frac{\lambda}{2} \|u^{n+1} + v - f\|_{L^2}^2 \quad (14)$$

To solve the equation for u^{n+1} for fixed v^n , we use the standard Split Bregman for TV problems (see [10]). To solve for v^{n+1} , the shrink function is applied:

$$v^{n+1} = S(f - u^{n+1}, \frac{\mu}{\lambda})$$

where $S(x, c)$ is $x - c$ if $x > c$ and $x + c$ if $x < -c$. The algorithm is summarized below.

Cartoon-Texture Decomposition with L^1 texture model

```

Initialize  $u^0 = f, v^0 = 0$ 
while Not Converged do
    Solve for  $u^{n+1}$  from equation 16
    Solve for  $v^{n+1}$  via shrinkage
end while

```

The substep for the u^{n+1} term adds back the edges, while the shrinkage term for the texture component thresholds via scale. For details on the Cartoon-Texture algorithm using the low patch-rank texture norm see [19].

3.2 Chan-Vese Algorithm

Let $\phi_{i,j}^n$ denote the value of the level set function at position (i, j) at iteration n . We follow a similar discretization as in [6, 20] to iterate forward to ϕ^{n+1} . Recall that the equation of motion for the level set function ϕ is as follows:

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left(\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \lambda \left(-(f - C_1)^2 + (f - C_2)^2 \right) \right)$$

By discretizing forward in time using a first order difference and first and second order differences in space yields the following scheme:

$$\phi_{i,j}^{n+1} = \frac{\phi_{i,j}^n + g(\phi^n) + \lambda \cdot \delta(\phi^n) \cdot [K_1 \phi_{i+1,j}^n + K_2 \phi_{i-1,j}^n + K_3 \phi_{i,j+1}^n + K_4 \phi_{i,j-1}^n]}{1 + \Delta t \delta(\phi^n) \lambda (K_1 + K_2 + K_3 + K_4)}. \quad (15)$$

where the function g and scalars K, K_1, K_2, K_3, K_4 are defined as:

$$\begin{aligned}
g &= \delta(\phi^n) \Delta t [-(f - C_1)^2 + (f - C_2)^2] \\
K &= K_1 + K_2 + K_3 + K_4 \\
K_1 &= \frac{1}{\sqrt{\varepsilon^2 + (\phi_{i+1,j}^n - \phi_{i,j}^n)^2 + \frac{1}{4} \cdot (\phi_{i,j+1}^n - \phi_{i,j-1}^n)^2}} \\
K_2 &= \frac{1}{\sqrt{\varepsilon^2 + (\phi_{i,j}^n - \phi_{i-1,j}^n)^2 + \frac{1}{4} (\phi_{i-1,j+1}^n - \phi_{i-1,j-1}^n)^2}} \\
K_3 &= \frac{1}{\sqrt{\varepsilon^2 + \frac{1}{4} (\phi_{i+1,j}^n - \phi_{i-1,j}^n)^2 + (\phi_{i,j+1}^n - \phi_{i,j}^n)^2}} \\
K_4 &= \frac{1}{\sqrt{\varepsilon^2 + \frac{1}{4} (\phi_{i+1,j-1}^n - \phi_{i-1,j-1}^n)^2 + (\phi_{i,j}^n - \phi_{i,j-1}^n)^2}}
\end{aligned}$$

Simplifying the denominator of equation 15 we get

$$\phi_{i,j}^{n+1} = \frac{\phi_{i,j}^n + g(\phi^n) + \lambda \cdot \delta(\phi^n) \cdot [K_1 \phi_{i+1,j}^n + K_2 \phi_{i-1,j}^n + K_3 \phi_{i,j+1}^n + K_4 \phi_{i,j-1}^n]}{1 + \Delta t \delta(\phi^n) \lambda K}. \quad (16)$$

Now ϕ^{n+1} can be used to find the next iteration of the curve, given ϕ^n . The parameter λ influences the smoothness of the curve, while Δt controls the stability. The parameter ε is used to avoid dividing by zero. In this work we set $\varepsilon = 10^{-5}$ for all experiments. Altogether, the algorithm is as follows.

Chan-Vese Algorithm

```

Initialize  $\phi^0$ 
while  $\phi^n$  not converged do
    Compute  $C_1$  and  $C_2$  from the  $\phi^n$ 
    Find  $\phi^{n+1}$  by equation 16
    Reset boundary conditions of  $\phi^n$  (Neumann BC)
end while

```

Computing the regional constants C_1 and C_2 follows easily from equations (10) and (11) using H_ε in place of the sharp Heaviside function. Convergence is achieved when ϕ^n minimizes the Chan-Vese energy for the given parameters. An additional reinitialization process may be needed to return the function ϕ^n to a distance function to the zero level curve; however, for our applications this is not necessary.

When the preceding algorithms are run in sequence, they yield a process that can detect an object's boundaries even when applied to a textured image. The complete algorithm is below.

4 Experimental Results

The following images display the results of the proposed object detection and segmentation method. For a simple noisy image the Chan-Vese algorithm can be used directly without the component

Complete Algorithm

Input noisy and highly textured data f

Run Cartoon-Texture Decomposition from algorithm on page 7: $f \rightarrow u + v + noise$

Input u into the Chan-Vese Segmentation algorithm from page 8: $u \rightarrow (C_1, C_2, \phi)$

Output characteristic function for the located region: $H(\phi)$

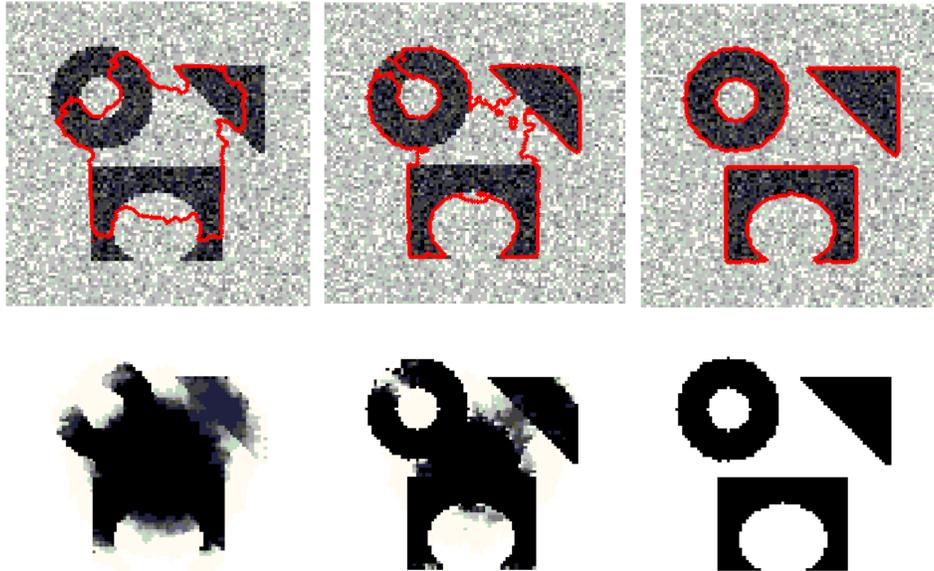


Figure 2: Above: Original image Ω and the contour Γ in red after 15, 25 and 100 iterations. Below: Piecewise constant reconstruction.

decomposition (see Figure 2). Also in the noise and texture free case the segmentation is similar to edge detecting. In Figure 3, the Chan-Vese algorithm is applied to an image of a needle (on a slightly varying background). The red curve (the segmenting curve) moves to the edges of the needle. The object is properly located at the steady state.

In Figure 4, the texture pattern of the straw causes the curve to falsely locate the straw as well as the needle. However, the cartoon is easily segmented and the reconstructed image clearly defines the needle. The parameters are more sensitive in this case than with our non-textured images; we choose $\lambda = .04 * 255^2$ and $\Delta t = .03$. Although separating the cartoon component adds time to the procedure, the whole process takes only a few minutes for all of our test images. One of the advantages of using a non-convex method like Chan-Vese is that the steady state curve is only a local minimizer. This is preferred for object detection, since we only want one specific feature or structure. This also makes us rethink the saying, since it does not seem that hard to find a needle in a haystack.

Of course, many images contain texture and noise that make segmentation difficult. In Figure 5, the grass introduces intensity variations which causes the segmentation algorithm to falsely locate a



Figure 3: Locating a needle with the Chan-Vese algorithm.

region near the tank. After separation, the tank is detected and the texture can be reintroduced to display the final results (Figure 5(f)). In this case we use parameters $\lambda = .0035 * 255^2$ and $\Delta t = .04$.

5 Conclusions

We proposed using a cartoon-texture-noise decomposition to separate features for a segmentation algorithm. We use two models for the texture component (the L^1 norm and the low patch-rank) and the total variation norm for the cartoon. In both cases, the cartoon provides an image which is easier to segment. The experimental results support this claim as well as the ability to locate a needle in a haystack.

References

- [1] J-F Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *Journal of Mathematical Imaging and Vision*, 22:71–88, January 2005.
- [2] J-F Aujol and T.F. Chan. Combining geometrical and textured information to perform image classification. *Journal of Visual Communication and Image Representation*, 17(5):1004–1023, October 2006.
- [3] J-F Aujol and S.H. Kang. Color image decomposition and restoration. *Journal of Visual Communication and Image Representation*, 17(4):916–928, August 2006.
- [4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, August 2003.
- [5] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
- [6] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.
- [7] L. C. Evans and R. F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, 1992.

- [8] J. Gilles. Noisy image decomposition: A new structure, texture and noise model based on local adaptivity. *Journal of Mathematical Imaging and Vision*, 28:285–295, August 2007.
- [9] J. Gilles and Y. Meyer. Properties of BV-G structures and textures decomposition models: Application to road detection in satellite images. *IEEE Transactions on Image Processing*, 19(11):2793–2800, November 2010.
- [10] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323, 2009.
- [11] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 2008.
- [12] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.
- [13] Y. Kim and L.A. Vese. Image recovery using functions of bounded variation and Sobolev spaces of negative differentiability. *Inverse Problems and Imaging*, 3:43–68, February 2009.
- [14] L.H. Lieu and L.A. Vese. Image restoration and decomposition via bounded total variation and negative Hilbert-Sobolev spaces. *Applied Mathematics and Optimization*, 58:167–193, May 2008.
- [15] Y. Meyer. Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures. *American Mathematical Soc.*, 2001.
- [16] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, July 1989.
- [17] S. Osher, A. Solé, and L.A. Vese. Image decomposition and restoration using total variation minimization and the H1. *Multiscale Modeling & Simulation*, 1:349, 2003.
- [18] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, November 1992.
- [19] H. Schaeffer and S. Osher. A low patch-rank interpretation of texture. CAM Report, November 2011.
- [20] L.A. Vese. Numerical discretization of the Rudin-Osher-Fatemi model, time-dependent semi-implicit discretization. Notes from UCLA Math 285J, Section 1, Fall 2009.
- [21] L.A. Vese and S. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal of Scientific Computing*, 19:553–572, 2002.
- [22] W. Yin, D. Goldfarb, and S. Osher. Image Cartoon-Texture decomposition and feature selection using the total variation regularized l-1 functional. In *Variational, Geometric, and Level Set Methods in Computer Vision*, volume 3752, pages 73–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

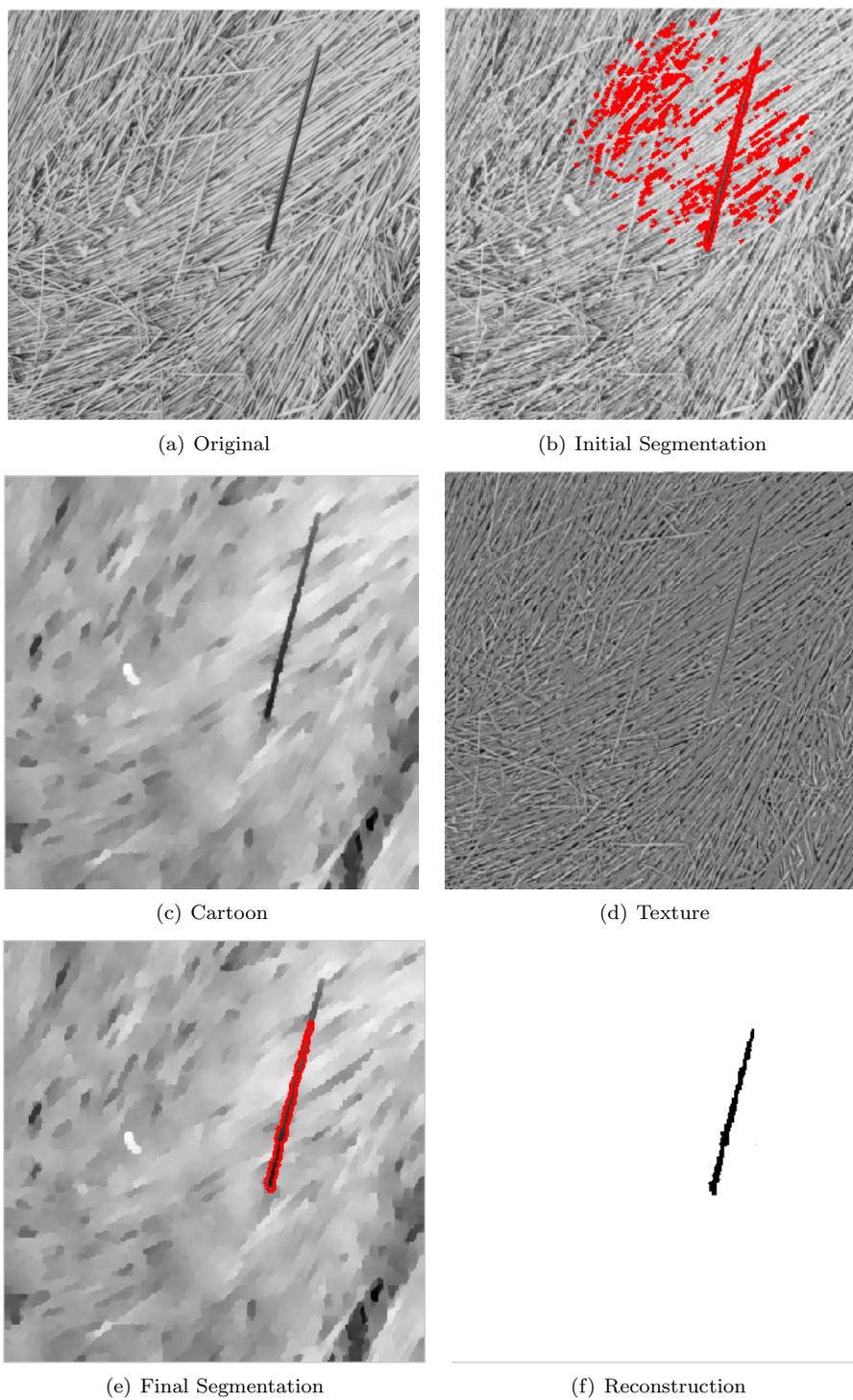


Figure 4: Segmenting a needle in a haystack before and after Cartoon-Texture decomposition.



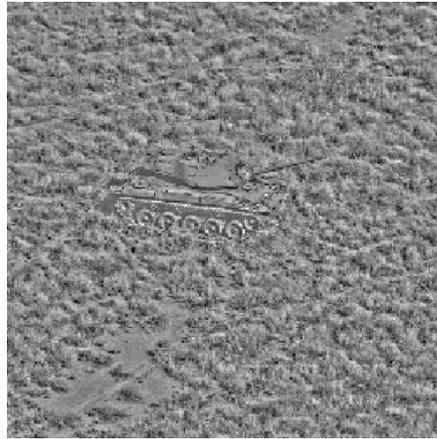
(a) Original



(b) Initial Segmentation



(c) Cartoon



(d) Texture



(e) Final Segmentation



(f) Reconstruction

Figure 5: Segmenting Tank image before and after Cartoon-Texture decomposition.