

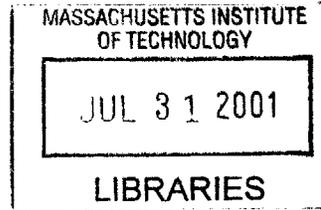
An Application of Nonlinear Resistive Networks in Computer Vision

by
Bikui Chen

Submitted to the Department of Electrical Engineering and Computer
Science and Department of Nuclear Engineering
in partial fulfillment of the requirements for the degrees of
Master of Science in Electrical Engineering and Computer Science
and
Master of Science in Nuclear Engineering

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May 2000

©Massachusetts Institute of Technology, 2000



Author
Department of Electrical Engineering and Computer Science and
Department of Nuclear Engineering

BARKER

Certified by
Berthold K. P. Horn
Department of Computer Science
Thesis Supervisor

Certified by
Richard Lanza
Senior Research Scientist of Nuclear Engineering
Thesis Supervisor

Read by
Ichiro Masaki

Principal Researcher
Laboratory
Thesis Reader

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Students, EECS

Accepted by
Sow-Hsin Chen
Chairman, Department Committee on Graduate Students, NED

An Application of Nonlinear Resistive Networks in Computer Vision

by

Bikui Chen

Submitted to the Department of Electrical Engineering and Computer Science and
Department of Nuclear Engineering
on May 3, 2000, in partial fulfillment of the
requirements for the degrees of
Master of Science in Electrical Engineering and Computer Science
and
Master of Science in Nuclear Engineering

Abstract

In this thesis, an improved optical flow algorithm is presented, as well as a hardware called “Tanh” component. The new approach performs an optimization that reduces the error at spatial discontinuities, and increases the computational speed using analog circuit implementation.

Simple simulation of this design is tested using HSPICE. We also build a simulator for a complicated circuit using C, which focuses more on speed, and less on transient.

For image smoothing and segmentation problems and optical flow problems, a series of test images are fed to both the resistor network and the so-called “TANH” network to determine how effective the Tanh network is in image analysis.

Thesis Supervisor: Berthold K. P. Horn

Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Richard Lanza

Title: Senior Research Scientist of Nuclear Engineering

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Professor Berthold K. Horn, for giving me the opportunity and support to work on this thesis, for providing me with crucial guidance and encouragement throughout years, and for serving as a model of an enthusiastic teacher and a dedicated scientist. I am also grateful to Dr. Richard Lanza, my thesis co-supervisor, for providing invaluable insight, ideas, and support. I have learned much from their expertise and am grateful for the time they have devoted to me.

This work has benefited from many discussions with Professor John L. Wyatt about non-linear networks, Professor Hae-Seung Lee about analog circuit design, Professor Rahul Sarpeshkar about non-linear analog circuit design and Dr. Christopher Terman about non-linear circuit simulation. I thank all of them for their enthusiasm, patience, and advice.

I would like to express my gratitude to all the people from my research group, Dr. Ichiro Masaki, Paul Fiore, Nicole Love and Yajun Fang, for their help and invaluable advice in the past three years.

I would also like to thank all my friends in MIT and elsewhere for the great moment we have shared in the past years, especially thank Benjamin Walter, for his advice and help in thesis revising.

I would like to express my deep love to my family, especially my parents, who have granted me unconditional love throughout my life; my sisters Bizhou and Jin, my brothers Bihui and Juhui, who have given me all kinds of support in both my career and my life.

Finally and most importantly, I would like to thank my girl friend, Xiang Xian, for her love and encourage.

Contents

- 1 Introduction** **9**
- 1.1 Optical Flow 9
- 1.2 Goal 10
- 1.3 Thesis Organization 10

- 2 Theory and Algorithm** **12**
- 2.1 The Optical Flow Problem 12
- 2.2 Previous Work 17
- 2.3 Our Work 17
- 2.4 Comparison of Smoothing Terms 23

- 3 Hardware Design and Simulation** **25**
- 3.1 Hardware Design 25
- 3.1.1 Goal 25
- 3.1.2 Previous work 25
- 3.1.3 Tanh Resistive component 26
- 3.2 Simulation 28
- 3.2.1 Transfer Function of TANH Circuit 28
- 3.2.2 Solving Nonlinear Circuit 30
- 3.2.3 Solving Sparse Matrix 31
- 3.2.4 Image Segmentation Simulation 37
- 3.2.5 Optical Flow Simulation 44

4 Conclusion	51
4.1 Summary	51
4.2 Recommendation	51

List of Figures

1-1	Structure of Computer System	10
2-1	Optical Flow Field	13
2-2	1-D BCCE Equation	14
2-3	Concave Smooth Function	19
2-4	Ideal Transfer Function of Interconnection	20
2-5	Resistive Grid	21
2-6	A Node in the Resistive Network	21
2-7	Comparison of Four Smoothing Terms	24
3-1	Simplified Mead's Saturating Resistor	26
3-2	Simplified TANH circuit	26
3-3	Detail of TANH Circuit	27
3-4	Transfer Function of TANH circuit (from HSPICE)	29
3-5	Transform a Nonlinear Resistor into Linear Components	31
3-6	Resistor Chain	37
3-7	1-D Voltage Input	38
3-8	Comparison of Smoothing Effect (1-D)	39
3-9	Image Input (64x64)	41
3-10	Simulation Result of Image Segmentation	42
3-11	Comparison of Smoothing Effect	43
3-12	2-D Gaussian Filters	45
3-13	Pixel Cube Used to Estimate the Three Partial Derivatives	46
3-14	Input Images of Optical Flow Simulation	47

3-15 Simulation Result of Optical Flow Algorithm with Tanh Network and Resistor Network 48

3-16 Comparison of Simulation Results from the Tanh Network and the Resistor Network 49

List of Tables

3.1	1-D Image Segmentation Error Comparison between Tanh Network and Resistor Network	40
3.2	Optical Flow Error Comparison between Tanh Network and Resistor Network	50

Chapter 1

Introduction

Twenty years have passed since Professor Berthold K.P. Horn and Brian Schunck published their influential paper[6] on the calculation of optical flow. Since then, a substantial amount of research has been devoted to finding ways to calculate optical flow more efficiently and more accurately. Despite volumes of research that have been published on this topic, the best current approach remains computationally intensive, because the vector field must be computed pixel by pixel. In the thesis, an analog resistive network is presented that will increase the processing speed and reduce the error at discontinuities.

1.1 Optical Flow

We can get much more information from an image sequence than from a single image. When a camera has motion relative to an object, a change of the brightness pattern in the image can be observed. This is called optical flow[5]. If we assign a velocity vector to each point in the image, we can get a motion field[5]. Since optical flow is a very good approximation of the motion field, optical flow extraction has been proposed as a preprocessing step for many high level vision algorithms. Knowledge of the optical flow field can provide the basis for the calculation of important parameters such as the Time-to-Collision (TTC) and the Focus-to-Expansion (FOE).

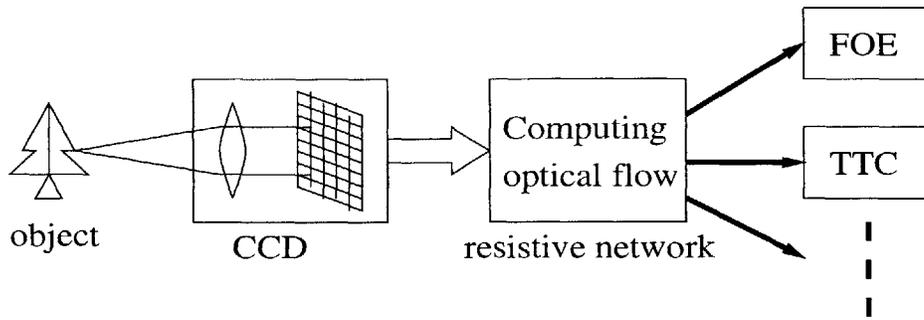


Figure 1-1: Structure of Computer System

Figure 1-1 shows the structure of a computer vision system. The Charge-Coupled Device (CCD) grabs the image of objects, converts them into voltage signals, then feeds them into a resistive network. The resistive network computes the optical flow as the intermediate result, and provides the input for other applications, such as computing TTC and computing FOE. Our work mainly focuses on the implementation the resistive network.

1.2 Goal

In this research, two approaches are considered. One is to implement the optical flow algorithm using an analog resistive network, which can increase speed and save power significantly. The other is to improve the algorithm, and design a new nonlinear resistive component, which can reduce the error near the spatial discontinuity, and simultaneously solve the local minima problem[1].

1.3 Thesis Organization

Algorithm improvement and hardware implementation are presented separately. Chapter 2 will introduce the theoretical framework, the optical flow algorithm and assumption that are used in this thesis. Based on this framework, Chapter 3 describes the hardware implementation, hardware architecture and the so-called TANH non-linear

resistive circuit. Results obtained using software simulation are then presented. Finally, Chapter 4 closes with conclusions and future work.

Chapter 2

Theory and Algorithm

2.1 The Optical Flow Problem

“Brightness patterns in the image move as the objects that give rise to them move. Optical Flow is the apparent motion of the brightness pattern. Ideally the optical flow will correspond to the motion field.”[5]. From this definition, we can see that we can compute the optical flow field based on the differences in brightness between two frames of image. Figure 2-1 gives an example of an optical flow field.

In Figure 2-1, the top two images are taken at the beginning and the end of a short time interval. We can tell that the two donuts are moving toward each other, because the area of overlap is greater in the second image. In the bottom image, the arrows show the optical flow field, with the velocity at each point being proportional to the length of the arrow. We need to pay attention to two things here. The first is that the optical flow field gives useful information about how those objects are moving. The second is that some non-zero velocity vectors appear in the static background. This is happened because in computing optical flow we assume that the velocity varies smoothly, which is not true at spatial discontinuities. The result is that the velocity vector dies slowly, which causes in significant error. In this thesis, we are going to focus on this problem, and look for an effective solution.

There are several methods to compute the optical flow field from given images, one of which is a gradient-based method. We can assume the lighting condition does not

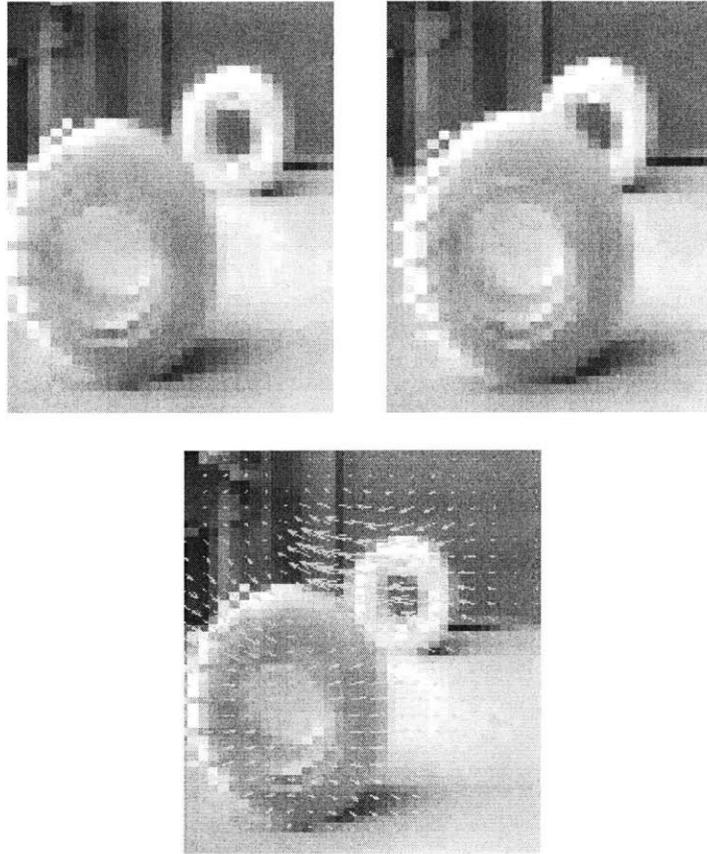


Figure 2-1: Optical Flow Field

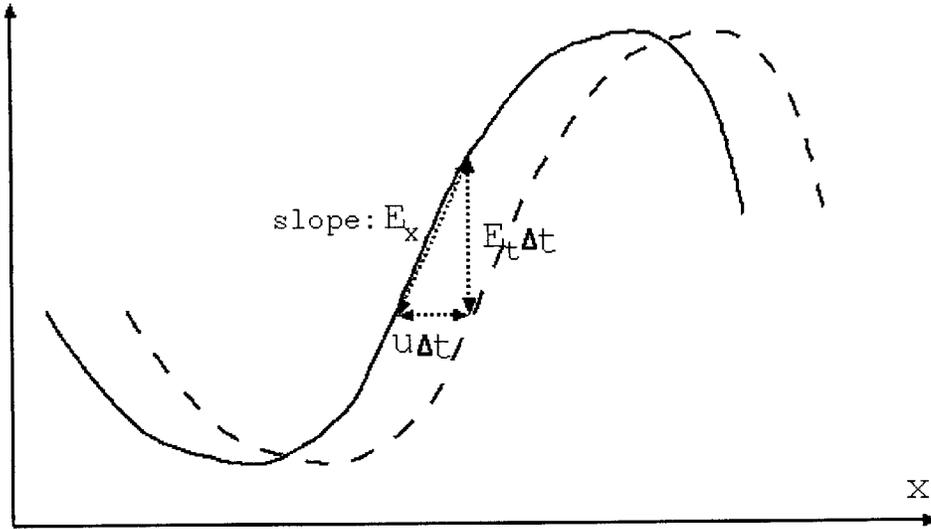


Figure 2-2: 1-D BCCE Equation

change. If the time interval between two frames of image is very small. Under these conditions, we can derive the Brightness Constant Constraint Equation (BCCE)[5]. Figure 2-2 shows the 1-D case.

In figure 2-2, u is the velocity along the x direction, E_x is the derivative of brightness in the x direction, E_t is the derivative of brightness with respect to time.

After time Δt , the brightness curve changes from the solid curve to the dashed curve. There are two ways to calculate the brightness change of a particular point. $(u \cdot \Delta t) \cdot (E_x)$ and $-\Delta t \cdot E_t$. Equating these, we obtain

$$(u \cdot \Delta t) \cdot (E_x) = -\Delta t \cdot E_t \quad (2.1)$$

or

$$u \cdot E_x + E_t = 0 \quad (2.2)$$

Similarly, we can obtain the equation for the 2-D case[5],

$$u \cdot E_x + v \cdot E_y + E_t = 0 \quad (2.3)$$

where u is the velocity in the x direction, v is the velocity in the y direction. This equation holds for each pixel.

We get one equation with two unknown variables. We need another constraint, which is a regularization term, to solve it. We can add a rigid body assumption, because most objects we deal with are rigid body. When objects are rigid bodies, the motion field varies smoothly across the image. We therefore seek to minimize a measure of departure from smoothness.

$$e_s = \int \int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy$$

The integral of square of the magnitude of the gradient of the optical flow. The error in the brightness constraint equation,

$$e_c = \int \int (u \cdot E_x + v \cdot E_y + E_t)^2 dx dy$$

should also be small.

Overall, the optical flow problem is to minimize the total error of these two terms.

$$\begin{aligned} E &= e_f + \lambda \cdot e_s & (2.4) \\ &= \int \int (u \cdot E_x + v \cdot E_y + E_t)^2 dx dy + \lambda \cdot \int \int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy \\ &= \int \int ((u \cdot E_x + v \cdot E_y + E_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2)) dx dy \end{aligned}$$

Similarly, we can write it in discrete form.

$$\begin{aligned}
E &= e_{f_{i,j}} + \lambda \cdot e_{s_{i,j}} \tag{2.5} \\
&= \sum_i \sum_j ((u_{i,j} \cdot E_{x_{i,j}} + v_{i,j} \cdot E_{y_{i,j}} + E_{t_{i,j}})^2 + \lambda(u_{x_{i,j}}^2 + u_{y_{i,j}}^2 + v_{x_{i,j}}^2 + v_{y_{i,j}}^2))
\end{aligned}$$

Differentiating the error with respect to u_{ij} and v_{ij} yields

$$\begin{aligned}
\frac{\partial E}{\partial u_{ij}} &= 2\lambda(u_{i,j} - \bar{u}_{i,j}) + 2(u_{i,j}E_{x_{i,j}} + v_{i,j}E_{y_{i,j}} + E_{t_{i,j}})E_{x_{i,j}} = 0 \tag{2.6} \\
\frac{\partial E}{\partial v_{ij}} &= 2\lambda(v_{i,j} - \bar{v}_{i,j}) + 2(u_{i,j}E_{x_{i,j}} + v_{i,j}E_{y_{i,j}} + E_{t_{i,j}})E_{y_{i,j}} = 0
\end{aligned}$$

From its Euler-Langrange Equations, we get,

$$\begin{aligned}
(\lambda + E_{x_{i,j}}^2)u_{i,j} + E_{x_{i,j}}E_{y_{i,j}}v_{i,j} &= \lambda\bar{u}_{i,j} - E_{x_{i,j}}E_{t_{i,j}} \\
E_{x_{i,j}}E_{y_{i,j}}u_{i,j} + (\lambda + E_{y_{i,j}}^2)v_{i,j} &= \lambda\bar{v}_{i,j} - E_{y_{i,j}}E_{t_{i,j}}
\end{aligned}$$

Where $\bar{u}_{i,j}$ and $\bar{v}_{i,j}$ are local average of u and v . These equations can be solved with an iterative scheme[5] as

$$\begin{aligned}
u_{i,j}^{n+1} &= \bar{u}_{i,j}^n - \frac{\bar{u}_{i,j}^n E_{x_{i,j}} + \bar{v}_{i,j}^n E_{y_{i,j}} + E_{t_{i,j}}}{\lambda + E_{x_{i,j}}^2 + E_{y_{i,j}}^2} E_{x_{i,j}} \tag{2.7} \\
v_{i,j}^{n+1} &= \bar{v}_{i,j}^n - \frac{\bar{u}_{i,j}^n E_{x_{i,j}} + \bar{v}_{i,j}^n E_{y_{i,j}} + E_{t_{i,j}}}{\lambda + E_{x_{i,j}}^2 + E_{y_{i,j}}^2} E_{y_{i,j}}
\end{aligned}$$

where

$$\begin{aligned}
\bar{u}_{i,j}^n &= \frac{u_{i-1,j}^n + u_{i+1,j}^n + u_{i,j-1}^n + u_{i,j+1}^n}{4} \\
\bar{v}_{i,j}^n &= \frac{v_{i-1,j}^n + v_{i+1,j}^n + v_{i,j-1}^n + v_{i,j+1}^n}{4}
\end{aligned}$$

From the recursive formula, we can see that when it converges, $\bar{u}_{i,j}E_{x_{i,j}} + \bar{v}_{i,j}E_{y_{i,j}} +$

$E_{t\ i,j} = 0$, which is exactly the brightness change constraint equation. Then $u_{i,j} = \bar{u}_{i,j}$ and $v_{i,j} = \bar{v}_{i,j}$.

2.2 Previous Work

Solution of optical flow problem has two intrinsic difficulties. Firstly, it is computationally expensive, because the algorithm involves iterative pixel-wise computation. Secondly, it introduces the smoothness term, which assumes the motion field varies smoothly. In Figure 2-1, the error that can result from this assumption is visible in the spurious background motion vectors.

There are several hardware implementations of optical flow algorithm, such as Tanner and Mead's Optical Motion Sensor[7], and Alan Stocker's optical computation circuit [14].

Strictly speaking, Tanner and Mead's chip is not a implementation of the optical flow algorithm, because they only produce a single unified global velocity instead of a pixel-wise velocity field. Alan Stocker improved Tanner and Mead's work, but his approach still suffers from the local minima problem, which we will discuss in Chapter 3.

There is also related research done on the image smooth and segmentation problem [4][8][10][11][13], such as Professor John Wyatt[1][2][9], Dr. Pietro Perona and Dr. J. Malik's resistive fuse[12], Dr. John Harris' tiny-tanh resistive element[3].

2.3 Our Work

Our goal was to design an algorithm that reduces error at spatial discontinuities, and allows large velocity gradients in those areas.

Another important problem in computer vision is the image smoothing and segmentation problem, which mathematically is to minimize the object function,

$$\begin{aligned}
E &= \int \int (u - e)^2 dx dy + \lambda \cdot \int \int (u_x^2 + u_y^2) dx dy \\
&= F(u|e) + S(u)
\end{aligned}$$

where u is the smoothed output, e is image input.

The object function of optical flow problem is

$$\begin{aligned}
E &= \int \int (u \cdot E_x + v \cdot E_y + E_t)^2 dx dy + \lambda \cdot \int \int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy \\
&= F(u, v|E_x, E_y, E_t) + S(u, v)
\end{aligned}$$

We can see some similarities between these two functions. They both consist of two terms, a fidelity term F and a smoothness term S . F is trying to get the best estimation of some parameter from input, and at the same time, S is smoothing the estimate[15].

In the recursive algorithm for the optical flow problem, the effect of the regularization term is to smooth the motion field. However in our approach, we want to preserve the edge features while smoothing out the noise. So I apply the network we designed to both the image smoothing and segmentation problem, and the optical flow problem.

A great deal of work has been done to improve the performance of image smoothing algorithms. One approach is to use a different smoothing term. The most common smoothing term is a quadratic smoothing function, such as $\int \int (u_x^2 + u_y^2) dx dy$. Dr. John Harris used an absolute value smoothing function, $\int \int (|u_x| + |u_y|) dx dy$ [3]. Professor John Wyatt, Dr. Pietro Perona and Dr. J. Malik proposed a concave smoothing function[1][9][12][15], which is shown in Figure 2-3.

In Figure 2-3, the solid line represents the smoothing function, and the dashed line is its first order derivative. Wyatt, Perona and Malik also implemented the smoothing function using a resistive-fuse circuit, which produces very good result in

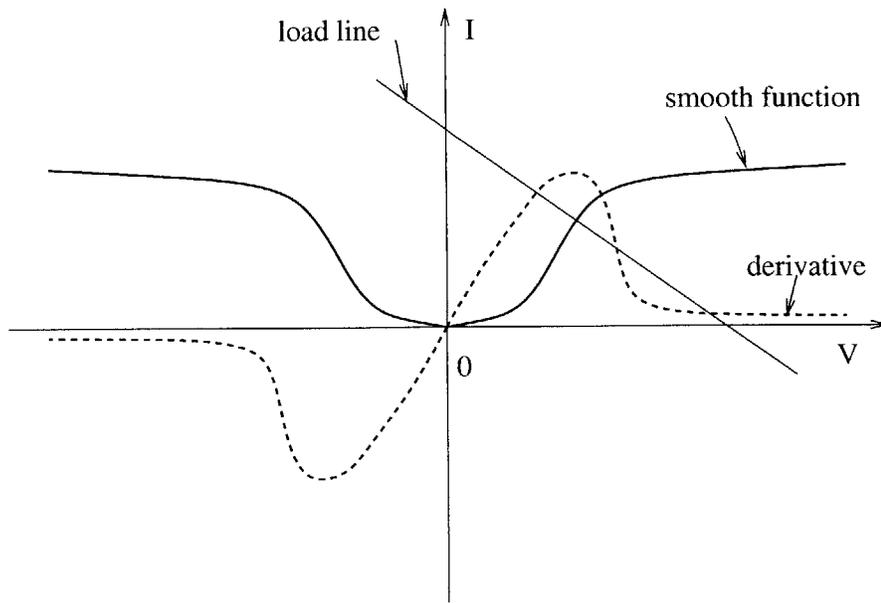


Figure 2-3: Concave Smooth Function

some images.

However, their circuit has local minima problem[1]. The derivative of the concave smoothing function is the V-I characteristics function of the resistive-fuse element. The straight line in figure 2-3 is the load line in the physical circuit. It intersects with the V-I curve more than once. Thus, there are multiple solutions to the object equation. The circuit may just converge to a local minimum.

What do we expect the behavior of a nonlinear resistor to be? When the input voltage difference across its two nodes is small, we assume that they are from pixels corresponding to the same object. In this case, the resistor should act like a normal resistor, so that it can conduct the filtering work. When the voltage difference is large, we may conjecture that they are from different objects. The resistor should cut off communication between these two neighboring nodes. There are at least two methods to implement this. We can cut off the current completely, as in John Wyatt and Malik's work, which is known to suffer from local minima[1]. We can also instead limit their relationship by limiting the current flow.

In Figure 2-4, the dashed line shows the transfer function for an idealized inter-

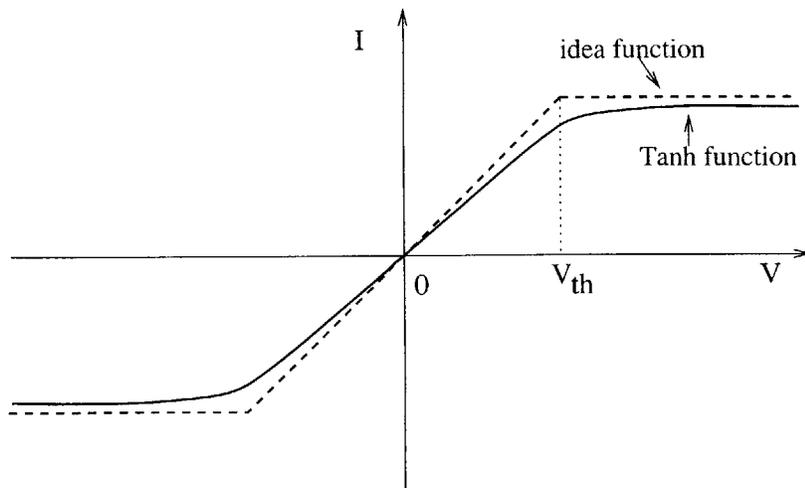


Figure 2-4: Ideal Transfer Function of Interconnection

connection component. When the voltage is smaller than V_{th} , the current increases linearly, just as in a normal resistor; when the voltage exceeds V_{th} , the current stops increasing. The ideal transfer function is not physical implementable, because its first order derivation at V_{th} is not continuous. We can use the hyperbolic tangent function (\tanh) to approximate it. It is shown as the solid line in figure 2-4. The Tanh function is even better, because its first order derivative is always continuous.

Figure 2-5 shows the basic structure of a resistive network in our approach. And Figure 2-6 shows detail at a node in the grid.

Each node is connected to its four neighbors by four resistive elements. At each node, a voltage-controlled current source injects current, which is zero when the circuit settles to its final state.

We now prove that the TANH circuit network will reach the optimal solution for the optical flow problem when the circuit settles. Similarly, we also prove that it solves the image smoothing and segmentation problem.

One concept is very important in nonlinear network analysis. That is the Co-Content of a resistive element[15]. For a resistor with a voltage-controlled constitutive relation

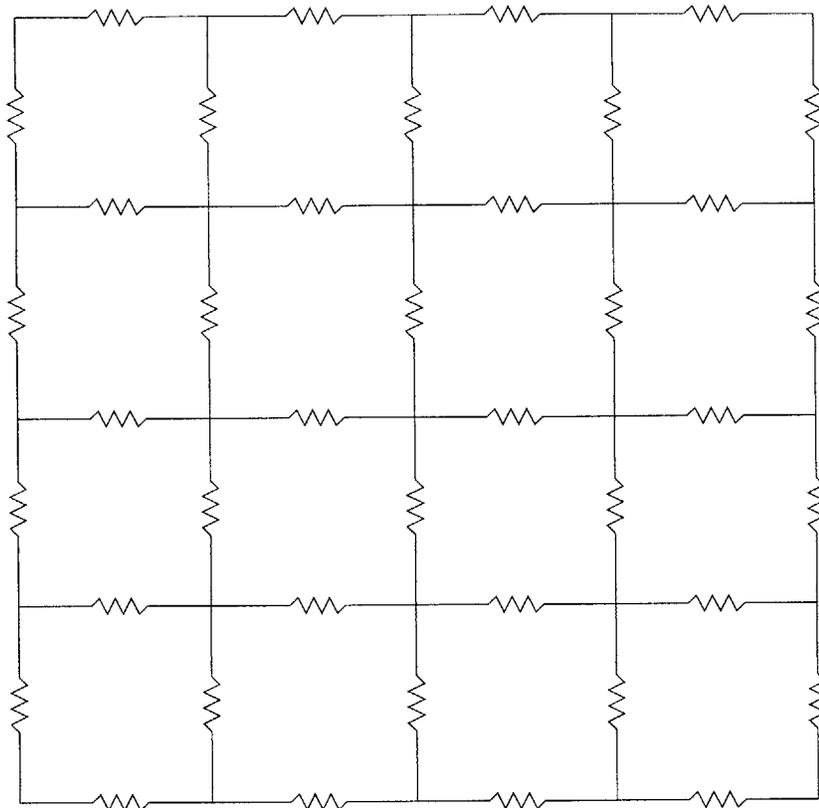


Figure 2-5: Resistive Grid

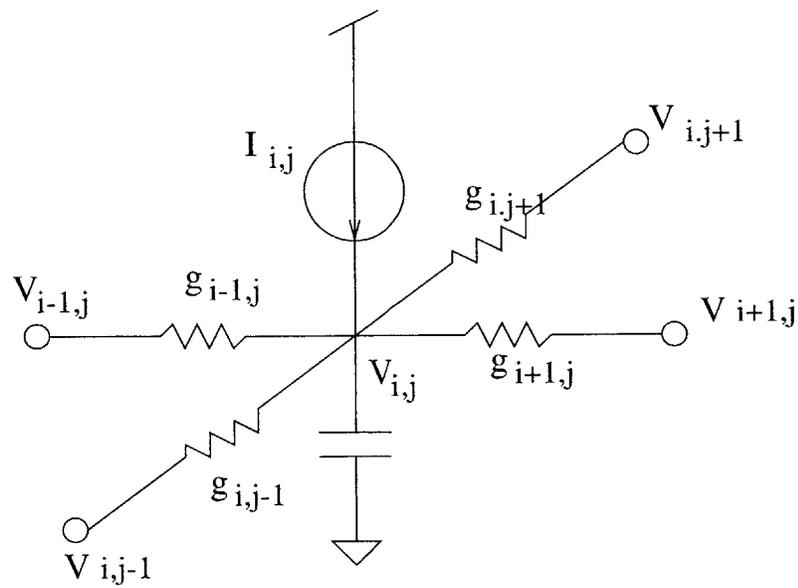


Figure 2-6: A Node in the Resistive Network

$$i = g(v),$$

the co-content is defined as

$$J(v) = \int_0^v g(v') dv'.$$

In particular, for a linear resistor, the co-content is simply half of the dissipated power. Although power and co-content have the same units, they are distinct in non-linear circuits. According to the Minimum Co-Content Principle[15], while the minimum power dissipation property fails for circuits with nonlinear resistive components, the total co-content is minimized instead[15].

If we use the integral of the *TANH* function as the smoothing term, then the error function of the optical flow problem becomes

$$\begin{aligned} E &= \int \int (u \cdot E_x + v \cdot E_y + E_t)^2 dx dy \\ &+ \lambda \int \int \left(\int_0^{u_x} \tanh\left(\frac{v}{\delta}\right) dv + \int_0^{u_y} \tanh\left(\frac{v}{\delta}\right) dv \right. \\ &\left. + \int_0^{v_x} \tanh\left(\frac{v}{\delta}\right) dv + \int_0^{v_y} \tanh\left(\frac{v}{\delta}\right) dv \right) dx dy \end{aligned}$$

Its discrete case is

$$\begin{aligned} E &= \sum_i \sum_j [(u_{i,j} \cdot E_{x_{i,j}} + v_{i,j} \cdot E_{y_{i,j}} + E_{t_{i,j}})^2 \\ &+ \lambda \left(\int_0^{u_{i+1,j} - u_{i,j}} \tanh\left(\frac{v}{\delta}\right) dv + \int_0^{u_{i,j+1} - u_{i,j}} \tanh\left(\frac{v}{\delta}\right) dv \right. \\ &\left. + \int_0^{v_{i+1,j} - v_{i,j}} \tanh\left(\frac{v}{\delta}\right) dv + \int_0^{v_{i,j+1} - v_{i,j}} \tanh\left(\frac{v}{\delta}\right) dv \right)] \end{aligned} \quad (2.8)$$

Differentiating it with respect to $u_{i,j}$ and $v_{i,j}$, we get the Euler-Langrange equations,

$$\begin{aligned}
\frac{\partial E}{\partial u_{ij}} = 0 &= \lambda(\tanh(\frac{u_{i,j} - u_{i+1,j}}{\delta}) + \tanh(\frac{u_{i,j} - u_{i-1,j}}{\delta}) + \tanh(\frac{u_{i,j} - u_{i,j+1}}{\delta}) \\
&+ \tanh(\frac{u_{i,j} - u_{i,j-1}}{\delta})) + 2(u_{i,j}E_{x\ i,j} + v_{i,j}E_{y\ i,j} + E_{t\ i,j})E_{x\ i,j} \\
\frac{\partial E}{\partial v_{ij}} = 0 &= \lambda(\tanh(\frac{v_{i,j} - v_{i+1,j}}{\delta}) + \tanh(\frac{v_{i,j} - v_{i-1,j}}{\delta}) + \tanh(\frac{v_{i,j} - v_{i,j+1}}{\delta}) \\
&+ \tanh(\frac{v_{i,j} - v_{i,j-1}}{\delta})) + 2(u_{i,j}E_{x\ i,j} + v_{i,j}E_{y\ i,j} + E_{t\ i,j})E_{y\ i,j}
\end{aligned} \tag{2.9}$$

These are transcendental equations, so that we can not get analytical solutions. But we can design a physical circuit, which will satisfy these equations when they are in stable state. Let's use the u component as the example.

We can build a resistive grid with $TANH$ components. One node from that grid is shown in the figure 2-6, the voltage of the central node is $u_{i,j}$, and its neighbors' voltage are $u_{i-1,j}$, $u_{i+1,j}$, $u_{i,j-1}$, $u_{i,j+1}$. They are all connected by the $TANH$ components, which satisfy $I = \lambda \tanh(\frac{V}{\delta})$. A voltage-controlled current source $I_{i,j}$ is connected to this node. The current it injects is $2(u_{i,j}E_{x\ i,j} + v_{i,j}E_{y\ i,j} + E_{t\ i,j})E_{y\ i,j}$. Each node satisfies the Euler-Langrange equation, and the co-content of this nonlinear resistive network is the same as the error function for the optical flow problem. According to the Minimum Co-content Property, the voltage distribution $u_{i,j}$ and $v_{i,j}$ at the stable state minimize the error function. Because the $\tanh()$ function is a strictly increasing convex function, the circuit always settles at the unique optimal point in the stable state.

2.4 Comparison of Smoothing Terms

Figure 2-7 compares four known smoothing functions. What characteristics should a good smoothing term have?

(a) When ΔV is small, it acts similar to a linear resistor, and far from the line $I=kV$ (in this case, the current will saturate immediately at 0^+ and no smoothing effect at all).

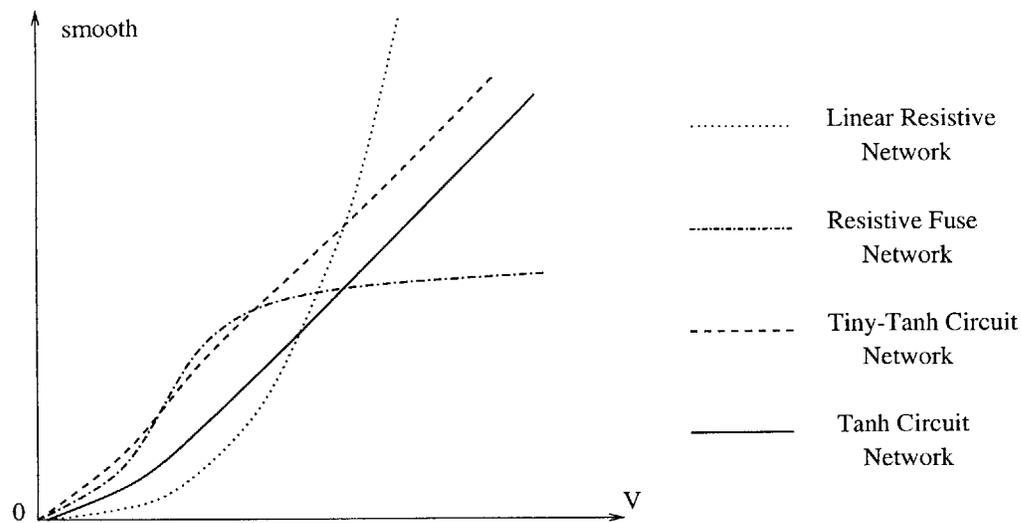


Figure 2-7: Comparison of Four Smoothing Terms

(b) When ΔV is large, it should be small, in order to preserve the edge features.

Observing Figure 2-7, we can see that the Tanh function is nearly linear for small Δv , so that it has more smoothing effect when Δv is small, which we suppose to correspond to noise, and has less smoothing effect when Δv is large, which occurs at velocity discontinuities. In this sense, *integral - tanh* function beats both quadratic function and absolute-value function as the regularization term.

Chapter 3

Hardware Design and Simulation

3.1 Hardware Design

3.1.1 Goal

In our design, the resistive network is actually a grid of resistive components. Resistive grids are well known as an efficient method of providing local interaction between cells with minimum requirement in terms of space and interconnection. In this paper, we will investigate the application of nonlinear resistive networks to vision problems.

Ideally, we would like the resistive components to act like linear resistors for small potential differences, and cut off when voltage is large. We also desire the circuit to be compact, since we are going to duplicate it for each pixel, and integrate them into one single chip. It should also work robustly in case of transistor mismatching. One of the easiest function of this kind is $TANH$ function. So that we choose $TANH$ as our solution for this problem.

3.1.2 Previous work

The next step is to implement the nonlinear resistive component. Dr. Carver Mead designed a saturating resistor[7]. It can be simplified to two transistors and two voltage sources.

In this design, two NMOS transistors are connected in serial. Two voltage sources

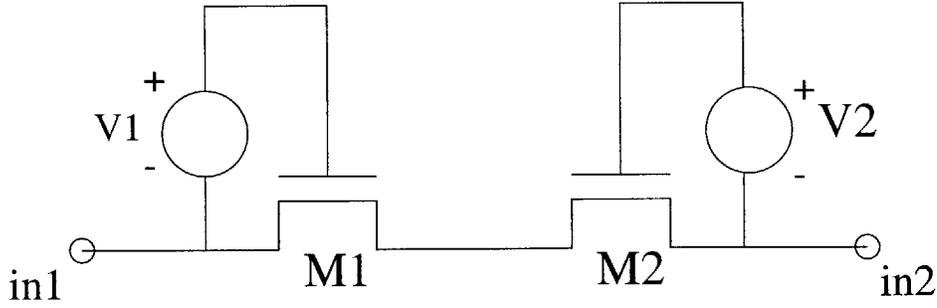


Figure 3-1: Simplified Mead's Saturating Resistor

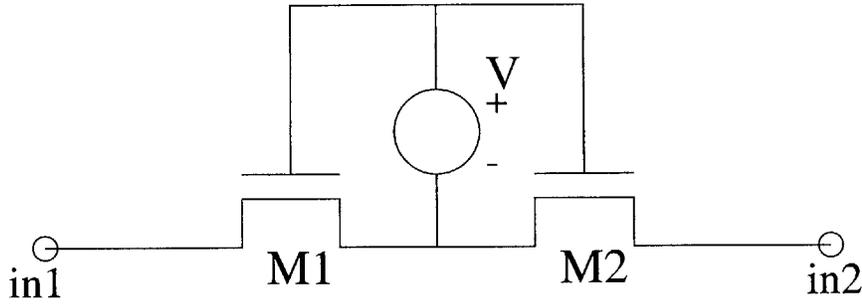


Figure 3-2: Simplified TANH circuit

$V1$ and $V2$ provide offset voltage with value of a V_{gs} . When $|V_{in1} - V_{in2}|$ is small, both $M1$ and $M2$ are in linear region, they act like linear resistor. When $|V_{in1} - V_{in2}|$ is large – let's assume V_{in1} is larger than V_{in2} , – $M1$ is in linear region, and $M2$ is in saturated region, the current flowing through is limited by $M2$, and vice versa when V_{in2} is larger than V_{in1} [9].

3.1.3 Tanh Resistive component

Following a suggestion by Professor Rahul Sarpeshkar, we combine the function of those two voltage sources into a single source. The voltage-current characteristics curve of the resulting circuit is a good approximation of the TANH function.

The detail of TANH component is shown in Figure 3-3. V_{bias} controls the conductance of this circuit in the linear region. Higher values of V_{bias} lower the conductance of this circuit. Raising V_{bias} will reduce the current flowing through the left branch,

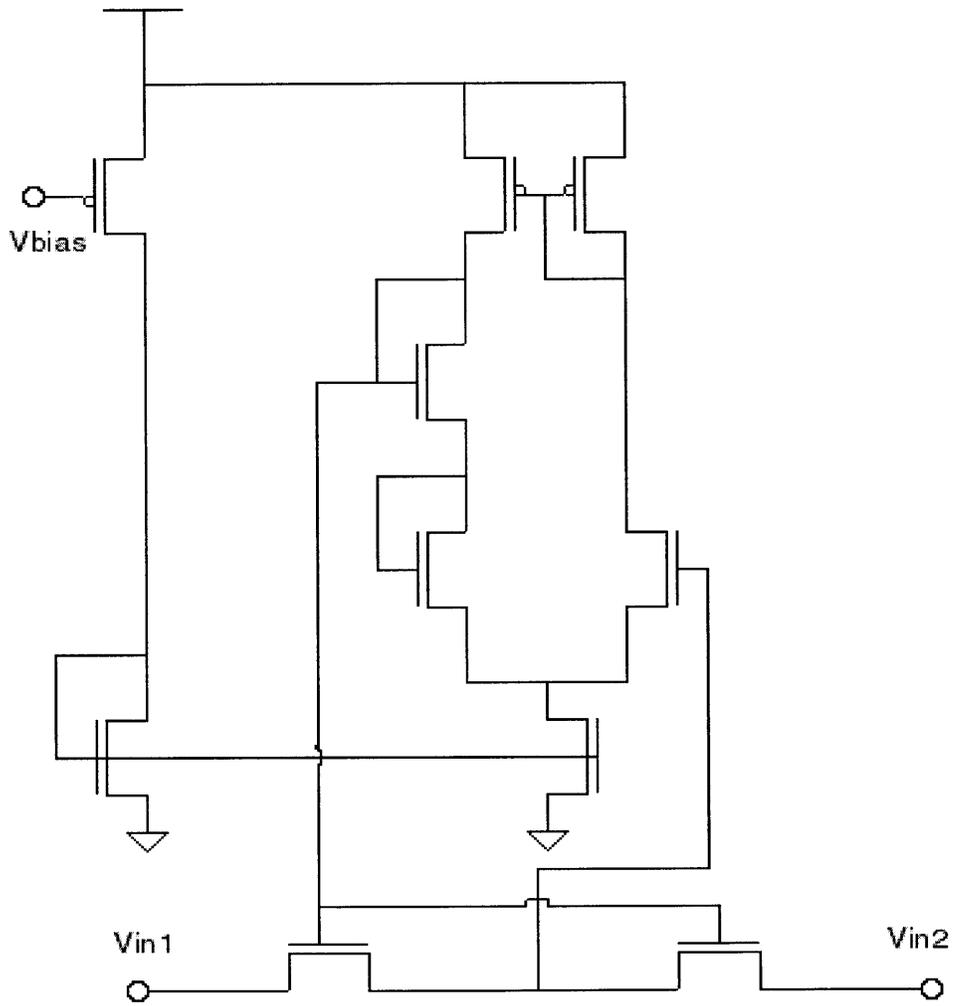


Figure 3-3: Detail of TANH Circuit

which will cause the V_{gs} of the bottom two NMOS to drop. Then the current flowing through the right part of circuit will decrease also. This causes that all the V_{gs} on the right part of the circuit to drop. As the voltage that the upper part of the circuit provides drops, the conductance of the bottom two horizontal NMOS decreases. We tried several other designs, but this one has an important virtue. The current flowing in from one side is always equal to the current flowing out from the other side. We will not have current leakage or transistor mismatching problem.

3.2 Simulation

We use several methods to simulate our implementation of the algorithm. We use HSPICE to simulate the $TANH$ resistive component and grids with less than 20 $TANH$ components. For grid including more than 20 $TANH$ resistive components, we simulate using C/C++ and the transfer function of $TANH$ circuit.

3.2.1 Transfer Function of TANH Circuit

We use HSPICE to simulate the $TANH$ circuit we described in Chapter 3. Figure 3-4 is the HSPICE simulation result of the $TANH$ circuit transfer function.

In this simulation, we set one side of input to $3.0V$, and change the voltage on the other side from $1.2V \sim 5.0V$. We can see that when ΔV is lower than $0.3V$, current I changes linearly, which is like a normal resistor. When ΔV is higher than $0.3V$, current I starts to increase more slowly with the increase of ΔV . Furthermore, after a narrow transient region, the rate of increase drops significantly. The transfer function looks like a $Tanh()$ function.

Because HSPICE is very slow when simulating a complicated circuit, we use the transfer function instead of the real circuit in simulations of larger networks. We sample 50 points from the HSPICE simulation result, and use MATLAB to do second order interpolation, producing an array of 1000 samples of $TANH$ transfer function. We use this transfer function in all the simulations later on. We can use the formula

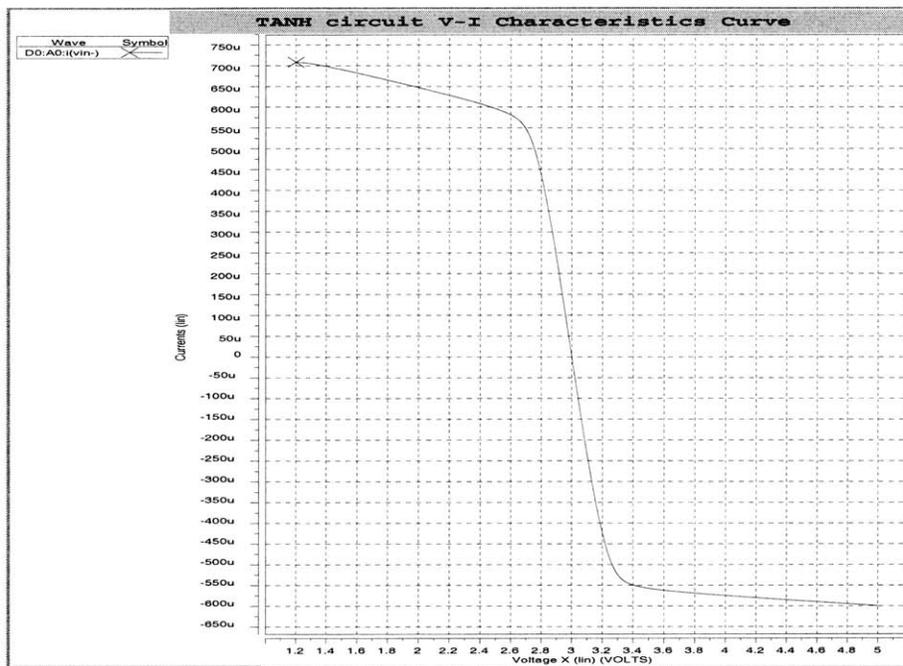


Figure 3-4: Transfer Function of TANH circuit (from HSPICE)

$$I(a) = V[n] * (n + 1 - a) + V[n + 1] * (a - n) \quad (3.1)$$

$$n \in N, n < a < n + 1$$

to calculate the current I when the input voltage a is not one of the given samples.

When simulating a nonlinear resistive network, we are faced with two problems, solving nonlinear circuit, since the *TANH* circuit has nonlinear transfer function, and solving sparse matrix, because each component only has connections to a few neighboring nodes.

3.2.2 Solving Nonlinear Circuit

In order to solve nonlinear equations using a matrix, we need to first linearize those equations. To linearize, we use the Newton-Raphson algorithm. For a nonlinear resistor, its current I and voltage V satisfy $I = f(V)$. The Taylor Series Expansion of this equation about the operating point v_0 is

$$i = f(v) = f(v_0) + f'(v_0)(v - v_0) + o(v - v_0)^2$$

We can derive the recursive scheme based on its Taylor Series Expansion,

$$\begin{aligned} i^{(l+1)} &= f(v^{(l+1)}) = f(v^{(l)}) + f'(v^{(l)})(v^{(l+1)} - v^{(l)}) \\ &= f'(v^{(l)})v^{(l+1)} + (f(v^{(l)}) - f'(v^{(l)})v^{(l)}) \\ &= f'(v^{(l)})v^{(l+1)} + I(v^{(l)}) \end{aligned} \quad (3.2)$$

where $I(v^{(l)}) = f(v^{(l)}) - f'(v^{(l)})v^{(l)}$.

Then we can replace a nonlinear resistor with a normal resistor and a voltage-controlled current source in parallel (see Figure 3-5).

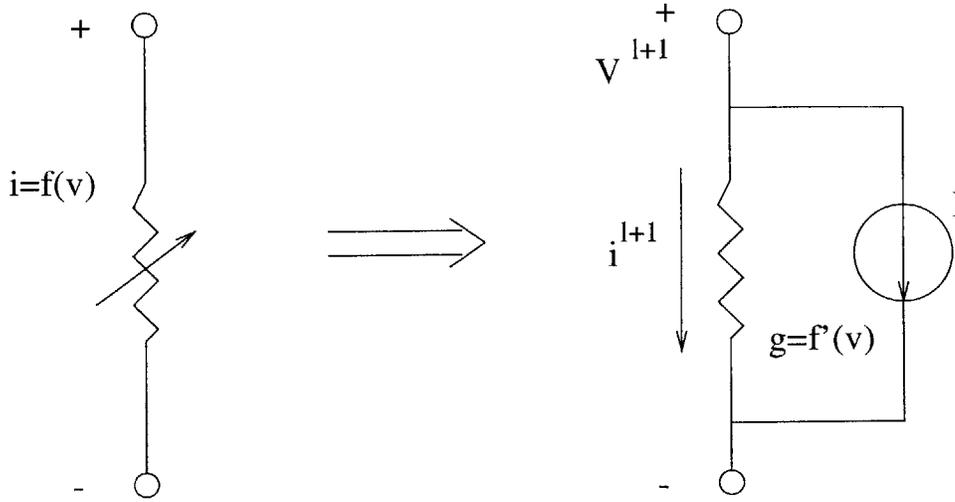


Figure 3-5: Transform a Nonlinear Resistor into Linear Components

The value of resistor is the first order derivative of transfer function $f'(v^{(l)})$, and the value of current source is $f(v^{(l)}) - f'(v^{(l)})v^{(l)}$, where $v^{(l)}$ is the solution from the last iteration.

3.2.3 Solving Sparse Matrix

Node A in the resistive grid is connected to four neighbors B, C, D, E with four nonlinear resistive components. The conductance between node A and its neighbors are $g_{AB}, g_{AC}, g_{AD}, g_{AE}$. It is also linked to load or ground with a normal resistor R . Following Kirchoff's Voltage Law, we get equation:

$$V_A \cdot (g_{AB} + g_{AC} + g_{AD} + g_{AE} + 1/R) - V_B \cdot g_{AB} - V_C \cdot g_{AC} - V_D \cdot g_{AD} - V_E \cdot g_{AE} = I_A \quad (3.3)$$

We can gather all the equations and write the result in the form

$$G_{n \times n} \cdot V_{n \times 1} = I_{n \times 1}$$

Element g_{ij} in $G_{n \times n}$ shows the conductance between node i and node j , $V_{n \times 1}$ is the voltage at the nodes, and $I_{n \times 1}$ represents the current source associated with each node.

If we have a resistor grid, with $N \times N$ nodes, we label each node with integers $1, 2, 3, 4, \dots, N \times N$. Each node in the grid has connection to at most 4 of its neighbors. So G is a highly sparse matrix, with dimension $(N^2 \times N^2)$, it has at most five lines of non-zero elements diagonal. A recursive methods yields an efficient solution to the sparse matrix, provided we can guarantee convergence. The positive conductance of those resistive components ensures that the recursive algorithm converges.

We are going to use Gauss-Siedel Recursive methods to solve these equations. If we divide matrix G into three matrixes, $G = D - L - U$, the recurrence is

$$V^{k+1} = (D - L)^{-1}U \cdot V^k + (D - L)^{-1} \cdot I \tag{3.4}$$

where

$$D = \begin{bmatrix} g_{11} & & & & \\ & g_{22} & & & \\ & & \dots & & \\ & & & & g_{nn} \end{bmatrix}, L = - \begin{bmatrix} 0 & & & & \\ g_{21} & 0 & & & \\ g_{31} & g_{32} & 0 & & \\ \dots & \dots & \dots & \dots & \\ g_{n1} & g_{n2} & & g_{nn-1} & 0 \end{bmatrix},$$

$$U = - \begin{bmatrix} 0 & g_{12} & g_{13} & \dots & g_{1n} \\ & 0 & g_{23} & \dots & g_{2n} \\ & & 0 & \dots & \dots \\ & & & \dots & g_{n-1n} \\ & & & & 0 \end{bmatrix}$$

Below is the proof of the convergence.

Lemma A: If matrix $G_{n \times n}$ satisfies $|g_{ii}| > \sum_{1 \leq j \leq n, i \neq j} |g_{ij}|$, then $\det(G) > 0$.

Proof:

Let's assume $\det(G) = 0$, then there is nonzero solution for equation $Gx = 0$. We denote the solution as

$$x = (x_1, x_2, x_3, \dots, x_n)^T \text{ and } |x_k| = \max_{1 \leq i \leq n} |x_i| \neq 0$$

From $Gx = 0$, we get $\sum_{j=1}^n g_{kj}x_j = 0$, then,

$$|g_{kk}x_k| = \left| \sum_{j=1, j \neq k}^n g_{kj}x_j \right| \leq \sum_{j=1, j \neq k}^n |g_{kj}||x_j| \leq |x_k| \sum_{j=1, j \neq k}^n |g_{kj}|$$

$$|g_{kk}| \leq \sum_{j=1, j \neq k}^n |g_{kj}|$$

This contradicts with the assumption: $|g_{ii}| > \sum_{\substack{1 \leq j \leq n \\ i \neq j}} |g_{ij}|$, so we know that $\det(G) \neq 0$.

Theorem: If matrix $G_{n \times n}$ satisfies $|g_{ii}| > \sum_{\substack{1 \leq j \leq n, \\ i \neq j}} |g_{ij}|$, all the eigenvalues, λ , of its recursive matrix $R = (D - L)^{-1}U$ satisfies $|\lambda| < 1$.

Proof:

In the equation $G_{n \times n} \cdot V_{n \times 1} = I_{n \times 1}$, we know that $|g_{ii}| \neq 0 (i = 1, 2, \dots, n)$ and $|g_{ii}| > \sum_{\substack{1 \leq j \leq n, \\ i \neq j}} |g_{ij}|$. The recursive matrix for Gauss-Siedel method is $R = (D - L)^{-1}U$. Its eigenvalues should satisfy

$$\begin{aligned} \det(\lambda I - R) &= \det(\lambda I - (D - L)^{-1}U) \\ &= \det(D - L)^{-1} \cdot \det(\lambda(D - L) - U) = 0 \end{aligned} \tag{3.5}$$

that equals to

$$\det(\lambda(D - L) - U) = 0$$

Let's denote $C = \lambda(D - L) - U = \begin{bmatrix} \lambda g_{11} & g_{12} & g_{13} & \dots & g_{1n} \\ \lambda g_{21} & \lambda g_{22} & g_{23} & \dots & g_{2n} \\ \lambda g_{31} & \lambda g_{32} & \lambda g_{33} & \dots & g_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ \lambda g_{n1} & \lambda g_{n2} & \dots & \lambda g_{nn-1} & \lambda g_{nn} \end{bmatrix}$ if $|\lambda| \geq 1$,

$$|c_{ii}| = |\lambda| \cdot |g_{ii}| > \sum_{j=1}^{i-1} |\lambda g_{ij}| + \sum_{j=i+1}^n |g_{ij}| = \sum_{j \neq i} |c_{ij}|$$

from Lemma A, we know $\det(C) = \det(\lambda(D - L) - U) \neq 0$, Therefore, all λ must satisfy $|\lambda| < 1$. In our problem, the conductance matrix has the form

$$\begin{bmatrix} G & -R \\ -R & R \end{bmatrix} \quad (3.6)$$

where

$$G = \begin{bmatrix} g2[0] & g3[0] & \dots & g4[0] & \dots & 0 \\ g1[1] & g2[1] & g3[1] & \dots & \dots & \dots \\ \dots & g1[2] & \dots & \dots & \dots & g4[n-1-N] \\ g0[2] & \dots & \dots & \dots & g3[n-3] & \dots \\ \dots & \dots & \dots & g3[n-2] & g2[n-2] & g3[n-2] \\ 0 & \dots & g0[n-1] & \dots & g1[n-1] & g2[n-1] \end{bmatrix}_{n \times n}$$

$$R = \begin{bmatrix} 1/r & 0 & \dots & \dots & 0 & 0 \\ 0 & 1/r & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & 1/r & 0 \\ 0 & 0 & \dots & \dots & 0 & 1/r \end{bmatrix}_{n \times n}, n = N \times N$$

Only five lines of non-zero elements appear in the diagonal direction. Each row of the conductance matrix corresponds to an equation in the form of equation 3.3. Because

all the elements have positive conductance, for matrix G , those elements along the diagonal are larger than the sum of all other elements in the same row. That is to say,

$$|g_{ii}| > \sum_{\substack{1 \leq k \leq N \times N \\ i \neq k}} |g_{ik}|$$

Then it fits Theorem B, so that the absolute values of all the eigenvalues of its recursive matrix $(D - L)^{-1}U$ are less than 1. We know that the recursive method converges if and only if the absolute value of all the eigenvalues of the matrix are less than 1. Therefore the Gauss-Siedel method converges for this problem.

For the image segmentation problem, the conductance should satisfy the equation,

$$\begin{bmatrix} G & -R \\ -R & R \end{bmatrix} \cdot \begin{bmatrix} V_{out} \\ V_{in} \end{bmatrix} = \begin{bmatrix} I_{node} \\ 0 \end{bmatrix} \quad (3.7)$$

where

$$G = \begin{bmatrix} g2[0] & g3[0] & \dots & g4[0] & \dots & 0 \\ g1[1] & g2[1] & g3[1] & \dots & \dots & \dots \\ \dots & g1[2] & \dots & \dots & \dots & g4[n-1-N] \\ g0[2] & \dots & \dots & \dots & g3[n-3] & \dots \\ \dots & \dots & \dots & g3[n-2] & g2[n-2] & g3[n-2] \\ 0 & \dots & g0[n-1] & \dots & g1[n-1] & g2[n-1] \end{bmatrix}_{n \times n}$$

$$R = \begin{bmatrix} 1/r & 0 & \dots & \dots & 0 & 0 \\ 0 & 1/r & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & 1/r & 0 \\ 0 & 0 & \dots & \dots & 0 & 1/r \end{bmatrix}_{n \times n},$$

$$V = \begin{bmatrix} V_{out(0)} \\ \dots \\ V_{out(n-1)} \\ V_{in(0)} \\ \dots \\ V_{in(n-1)} \end{bmatrix}_{2n \times 1}, I_{node} = \begin{bmatrix} I_{(0)} \\ \dots \\ I_{(n-1)} \end{bmatrix}_{n \times 1}$$

Then,

$$G \cdot V_{out} - R \cdot V_{in} = I_{node},$$

$$G \cdot V_{out} = R \cdot V_{in} + I_{node}$$

Solving this equation, we get the V_{out} for each iteration. We also need to update the conductance matrix,

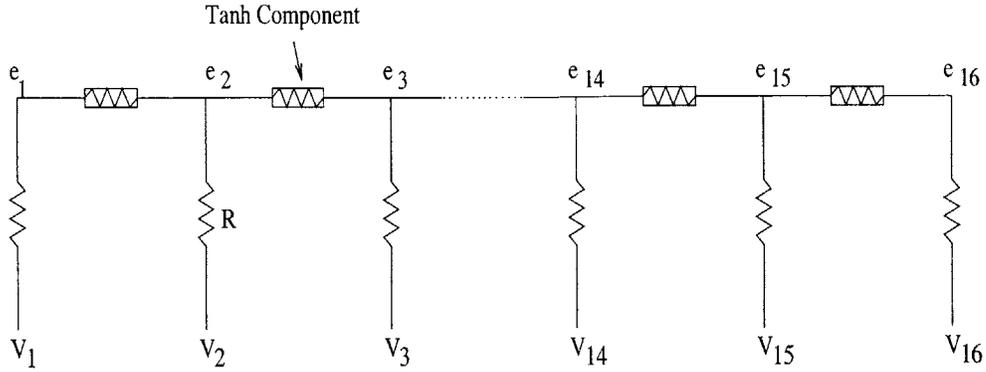


Figure 3-6: Resistor Chain

$$\begin{aligned}
 g1[i] &= g3[i - 1] = -f(|V_{out(i)} - V_{out(i-1)}|); i \geq 1 \\
 g0[i] &= g4[i - N] = -f(|V_{out(i)} - V_{out(i-1)}|); i \geq N \\
 g2[i] &= g0[i] + g1[i] + g3[i] + g4[i];
 \end{aligned}
 \tag{3.8}$$

where $i = f(v)$ is the transfer function of *TANH* component, and V_{out} is from last iteration. When the iteration converges, we will have found the solution for this problem.

3.2.4 Image Segmentation Simulation

1-D Simulation

We start with a 1-D simulation. With 15 *TANH* components, we build up a chain with 16 nodes. Figure 3-6 shows the structure of the resistor chain. e is output, and V is input.

In Figure 3-7, we see how we generate an input signal. We have a step signal, which is in the range $2.0V \sim 3.0V$. It is corrupted by noise r , which satisfies the distribution $r \sim N(0, 0.05)$.

Figure 3-8 shows the HSPICE simulation of the resistor chain. We can divide the

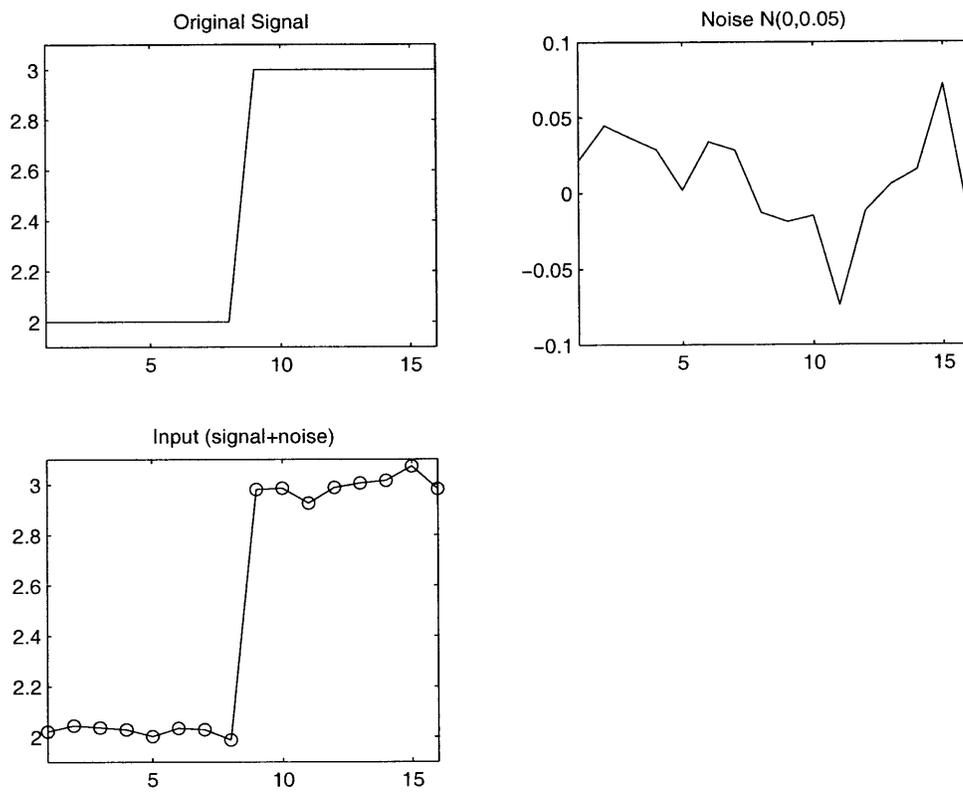


Figure 3-7: 1-D Voltage Input

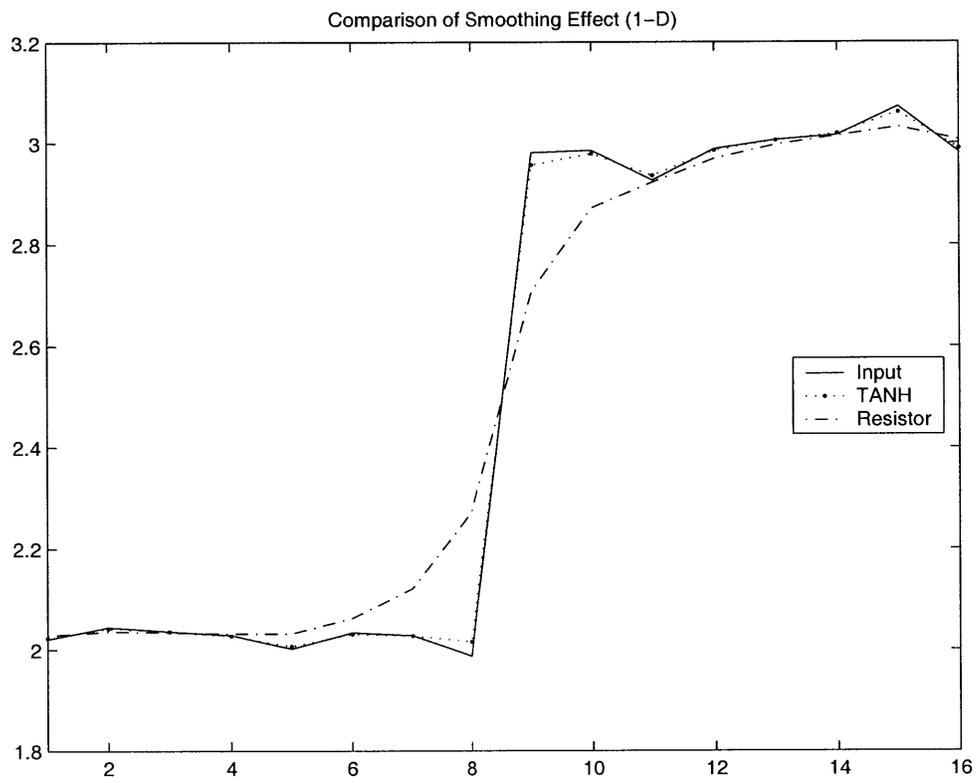


Figure 3-8: Comparison of Smoothing Effect (1-D)

	TANH	Resistor
error	0.0173	0.2077

Table 3.1: 1-D Image Segmentation Error Comparison between Tanh Network and Resistor Network

input signal into three parts, two relatively flat regions, and one transition region. We can see that, while smoothing the flat regions, the *TANH* chain also keeps the edge feature. A normal resistor chain tends to smooth all the humps, including the edge.

We use $\sum(V_{out} - V_{original})^2$ as an error criterion, to get the data shown in the Table 3.1, from which we can see that *TANH* function reduces error significantly.

2-D Simulation

In 2-D simulation, we use a 64×64 input image. Due to the huge computation load for HSPICE, we use C/C++ language to implement the simplified model for simulation.

Figure 3-9 is the surface plot of 64×64 input image. Input image voltage range is $1.2V \sim 4.0V$.

Figure 3-10 shows the simulation results of image segmentation. When we feed the same image to one network consisting of Tanh components, and one resistor network, we get two different results. The top two graphs are the output images, and the bottom two are graphs a single row of data selected from each image. By comparing them side by side in Figure 3-11, we can see the different smoothing effects of *TANH* component grid and normal resistor grid.

From Figure 3-11, we can see that in the flat region the output signal from the *TANH* network is similar to the output signal from the normal resistor network, that is to say that the *TANH* network has the same effect as normal resistor in smoothing. The output signal from the *TANH* network is similar to the input signal in the edge region which shows that it keeps the edge feature from the input, as desired.

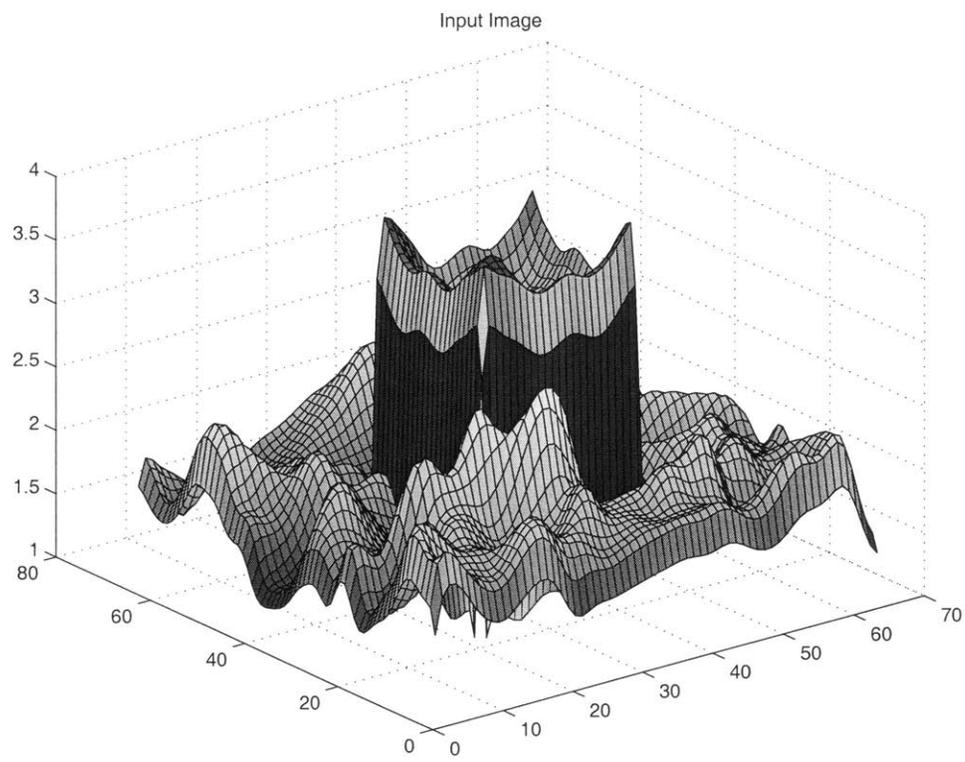


Figure 3-9: Image Input (64x64)

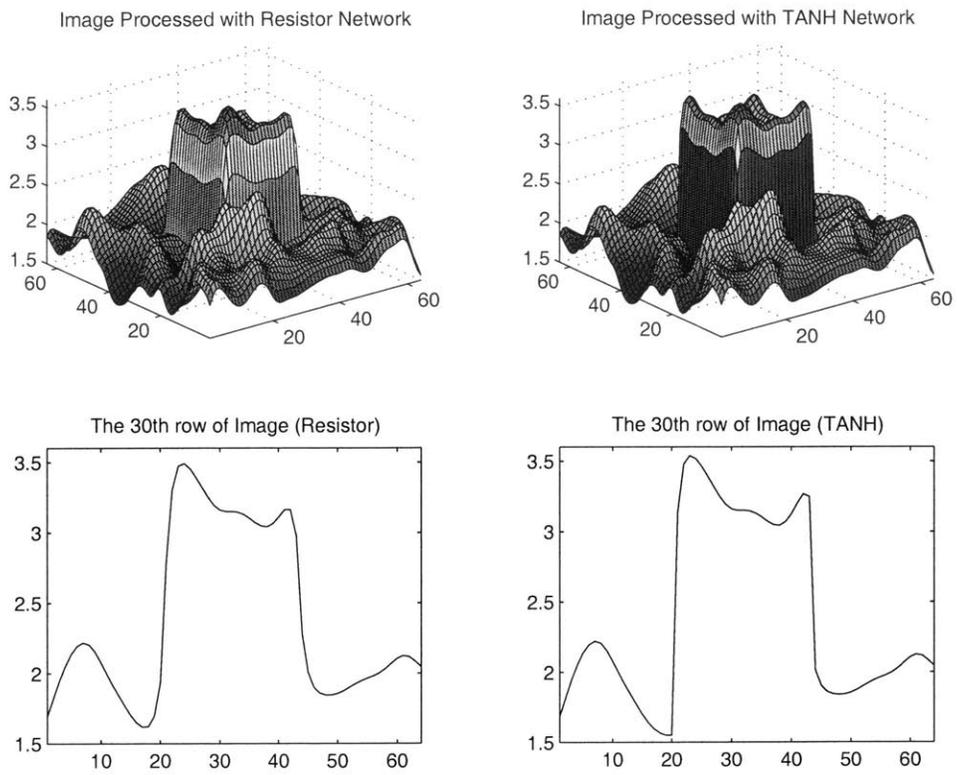


Figure 3-10: Simulation Result of Image Segmentation

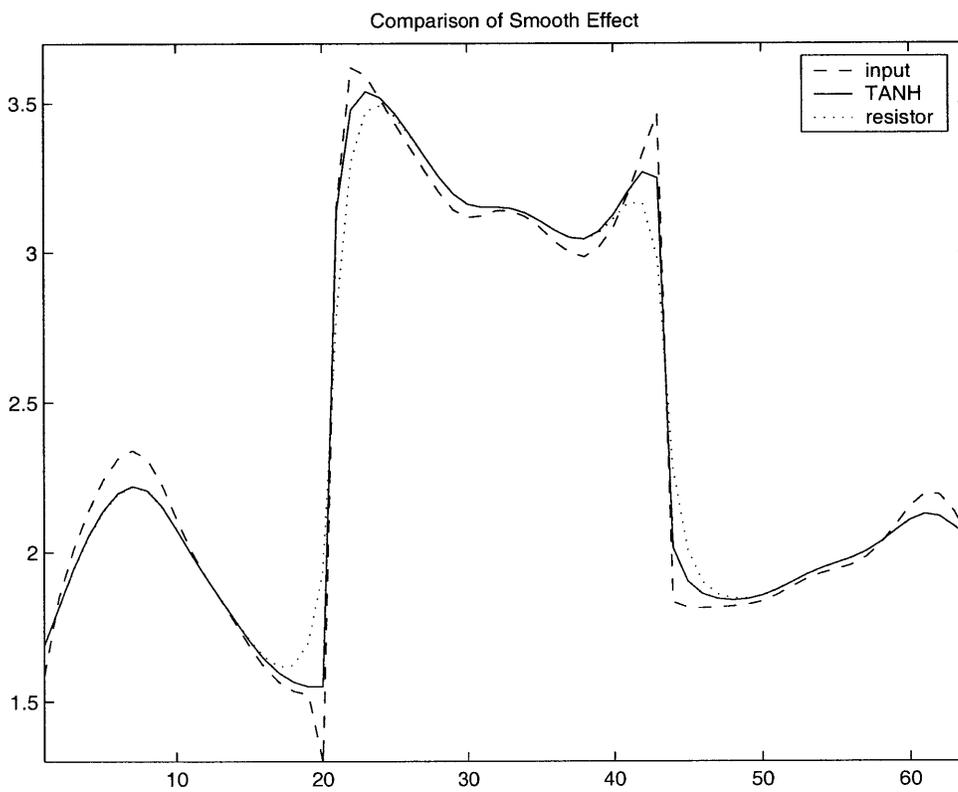


Figure 3-11: Comparison of Smoothing Effect

3.2.5 Optical Flow Simulation

Preprocess Image

Input images are discrete in time and space. Gradients of the image brightness are needed for the optical flow algorithm, which requires that the image brightness be differentiable. Therefore, a smoothing process was applied to the input images to improve the subsequence derivative estimates. Because there is effectively a built-in low pass filter in time – the video camera smears the input image sequence and decreases aliasing, – we are more interested in spatially smoothing the image. The input image can be pre-smoothed with a Gaussian filter,

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\left(\frac{x^2}{2\sigma^2} + \frac{y^2}{2\sigma^2}\right)\right] \quad (3.9)$$

where the mean of x and y are both zero, and the standard deviation is σ . We can choose proper σ according to our need for the smoothness of the input images. When using a Gaussian filter to smooth images, the larger the value σ is, the larger the smoothing window will be. The shape of Gaussian filter is shown in the figure 3-12.

Another need for smoothing image is when the displacement between two frame of images is large. We know that the optical flow algorithm assumes small change of brightness, so that it more accurate for a smaller displacement. But we can also apply it to large displacement with the hierarchical scheme. We can first smooth the input images with a Gaussian filter, then downsample images and get a lower-resolution image with less motion. After we get the estimation of motion variables, we can apply them to higher level of estimations, then we can reach the optimal estimation of large displacement.

Gradient Derivation

The optical flow algorithm involves the computation of spatial and temporal partial derivatives (E_x, E_y, E_t) of the brightness at each pixel in the image. To get the

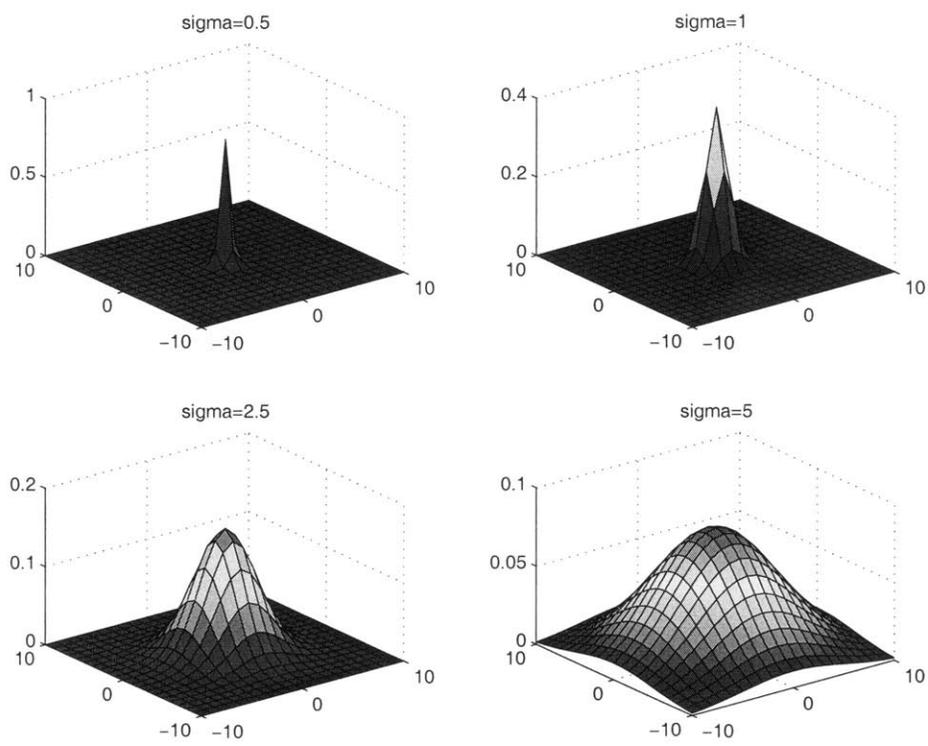


Figure 3-12: 2-D Gaussian Filters

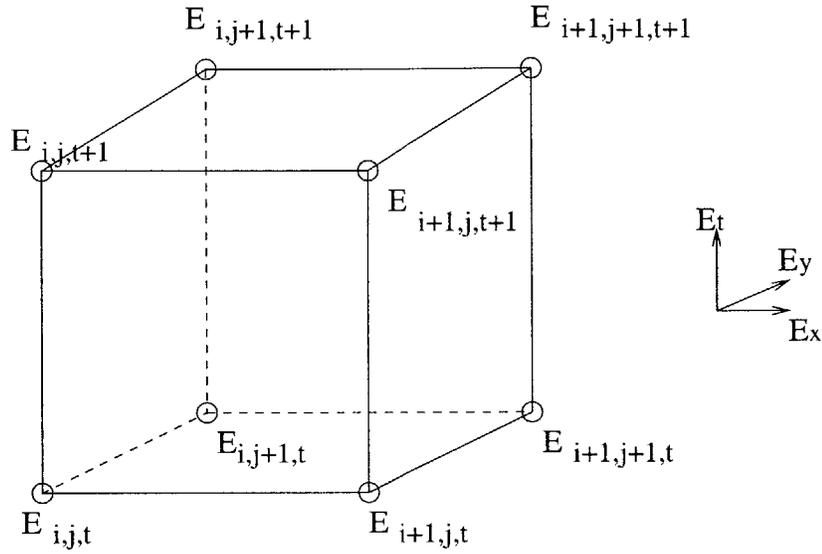


Figure 3-13: Pixel Cube Used to Estimate the Three Partial Derivatives

derivative E_t , we also need an image pair taken sequentially in time. The partial derivatives of image brightness are computed with a first order derivative. The first order difference approximations are

$$\begin{aligned}
 E_x|_{i,j,k} &= \frac{E(i+1) - E(i)}{\Delta x} \\
 E_y|_{i,j,k} &= \frac{E(j+1) - E(j)}{\Delta y} \\
 E_t|_{i,j,k} &= \frac{E(k+1) - E(k)}{\Delta t}
 \end{aligned} \tag{3.10}$$

In our approach, we instead get the derivatives as below[5],

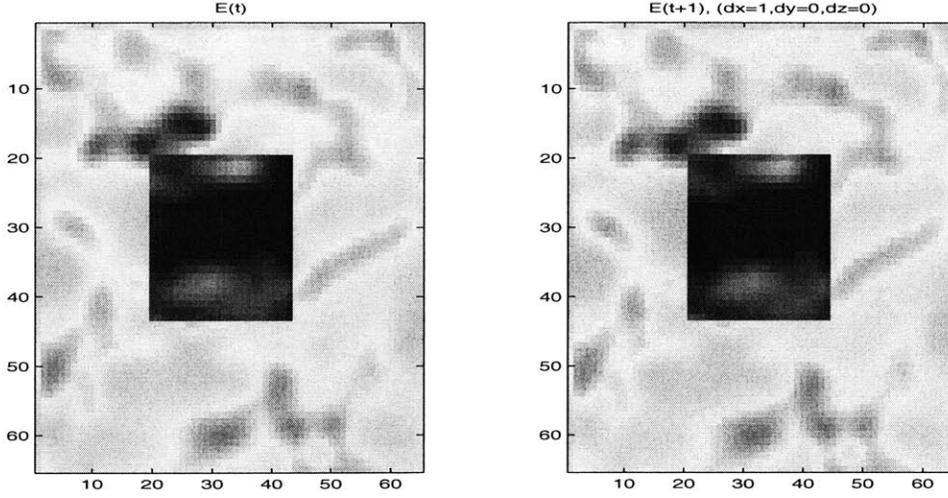


Figure 3-14: Input Images of Optical Flow Simulation

$$\begin{aligned}
 E_x(i,j,k) &= \frac{1}{4\Delta x} [(E_{(i+1,j,k)} + E_{(i+1,j+1,k)} + E_{(i+1,j,k+1)} + E_{(i+1,j+1,k+1)}) \\
 &\quad - (E_{(i,j,k)} + E_{(i,j+1,k)} + E_{(i,j,k+1)} + E_{(i,j+1,k+1)})] \\
 E_y(i,j,k) &= \frac{1}{4\Delta y} [(E_{(i,j+1,k)} + E_{(i+1,j+1,k)} + E_{(i,j+1,k+1)} + E_{(i+1,j+1,k+1)}) \\
 &\quad - (E_{(i,j,k)} + E_{(i+1,j,k)} + E_{(i,j,k+1)} + E_{(i+1,j,k+1)})] \\
 E_t(i,j,k) &= \frac{1}{4\Delta t} [(E_{(i,j,k+1)} + E_{(i+1,j,k+1)} + E_{(i,j+1,k+1)} + E_{(i+1,j+1,k+1)}) \\
 &\quad - (E_{(i,j,k)} + E_{(i+1,j,k)} + E_{(i,j+1,k)} + E_{(i+1,j+1,k)})]
 \end{aligned} \tag{3.11}$$

where $E(i, j, k)$ corresponds to the brightness of pixel (i, j, k) . Here i is in the x direction, j is in the y direction, k is in the t direction. The three partial derivatives of image brightness at the center of the cube are estimated from the average of the four differences along the four parallel edges.

Simulation

Figure 3-14 is a pair of synthesized 64×64 input images. The square in the middle is assumed to be an object, which is moving over a static background. In this pair of

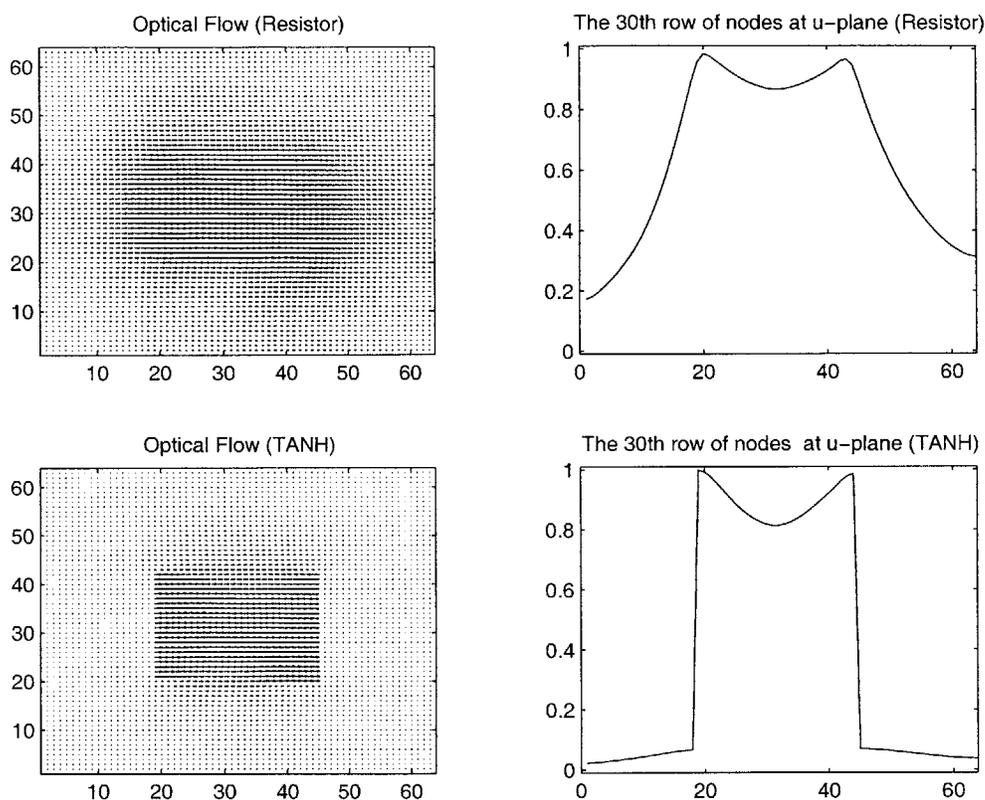


Figure 3-15: Simulation Result of Optical Flow Algorithm with Tanh Network and Resistor Network

the images, it moves exactly one pixel to the right, which is the positive x direction.

Figure 3-15 is the result of simulating optical flow, using both a Tanh network and a resistor network. The top two figures are from the resistor network, while the bottom two from the Tanh network. The right two figures illustrate a single row from the optical flow fields on the left hand side. We can see that the optical flow field obtained using a resistor network has a blurred boundary, while Tanh network yields a relatively clear-cut boundary in optical flow field, which is more consistent with the real motion field. Also in those static background area where we expect the velocity to be zero, the resistor network shows much larger velocity distribution.

Comparing the results of the two simulations, we can see that the *TANH* network out-performs the resistor network. It yields a clear-cut velocity field boundary and its velocity distribution drops rapidly at the edge, while the velocity distribution from

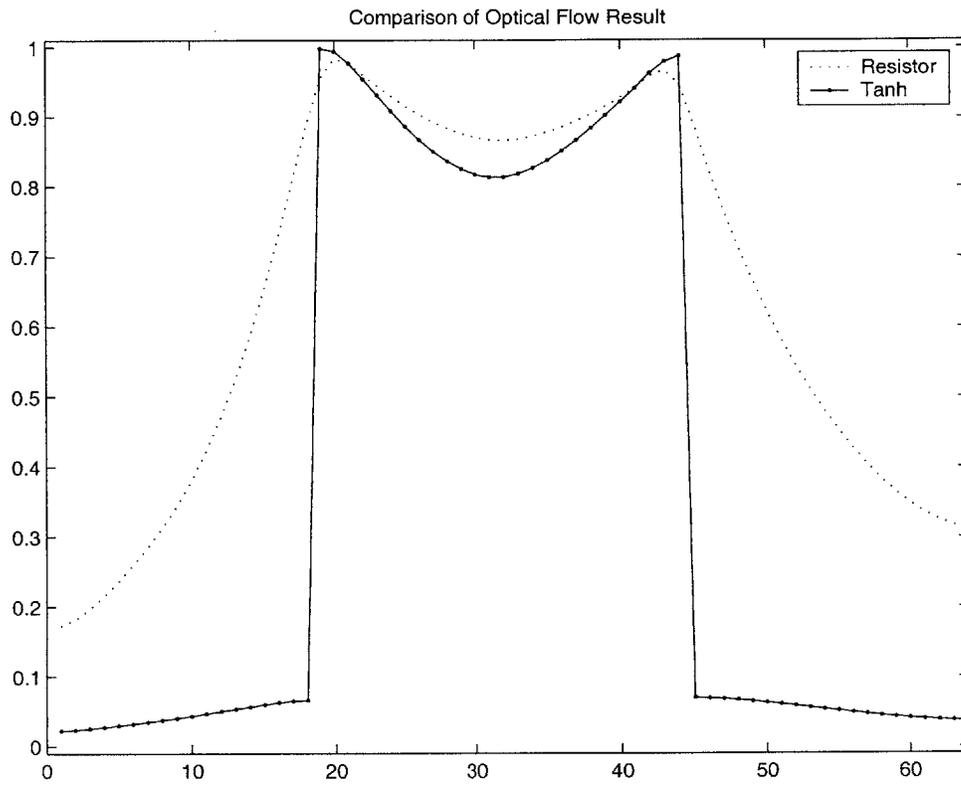


Figure 3-16: Comparison of Simulation Results from the Tanh Network and the Resistor Network

	Resistor	TANH
error	2.7473	1.8121

Table 3.2: Optical Flow Error Comparison between Tanh Network and Resistor Network

resistor network decreases at a much slower rate.

We use the constant brightness constraint, which is $e_f = \sum(u \cdot E_x + v \cdot E_y + E_t)^2$, as the criterion to evaluate the performance.

As the error shown in the Table 3.2, *TANH* network can reduce the error by around 30%.

Chapter 4

Conclusion

4.1 Summary

In this thesis, we investigated the application of non-linear resistive networks to a computer vision problem. Our research suggested that a $TANH$ component is useful in the solution of this problem. We completed a schematic design and simulation of a circuit composed of $TANH$ elements. It is known that analog circuits have advantage in both speed, and power consumption, comparing with digital circuit. This technology makes it possible to realize real-time optical flow computation, which is a computationally intensive algorithm. After reviewing some of the work of pioneers in this field, we proposed a $TANH$ component as the basic element for a resistive network, which can both conduct communication between neighbors when their voltage difference is small, and cut off communication when voltage difference is large. Thus, it can preserve edge features, while smoothing out noise in non-edge regions.

4.2 Recommendation

From our simulation, we can see that we have obtained a performance improvement in optical flow computation, however there is still lots of work to do in the future. For example, looking at the v direction of optical flow simulation result, it does not have clear-cut edge as good as that in u direction. That is because the E_t in this

area varies more slowly than that in spatially discontinuous area, it is harder to gain a large voltage difference to cut off the connection along the edge. In my design of the *TANH* circuit, the linear region is fixed, which means that the threshold value of cut off is fixed. If we have a adjustable threshold value, it can be more flexible. Furthermore, we can combine it with edge detection. We can set the threshold value at those edges smaller, and larger in interior area. So that we can smooth out interior noise feature, while keeping edge feature.

Bibliography

- [1] I. Elfadel A. Lumsdaine, J. Wyatt. Nonlinear analog networks for image smoothing and segmentation. *Journal of VLSI Signal Processing*, pages 53–68, 1991.
- [2] John Wyatt Hae-Seung Lee, Charles F. Sodini. CMOS resistive fuses for image smoothing and segmentation. *IEEE Journal of Solid-State Circuits*, 27, 1992.
- [3] John Harris. An analog network for continuous-time segmentation. *International Journal of Computer Vision* 10:1, pages 43–51, 1993.
- [4] John G. Harris. Discarding outliers using a nonlinear resistive network. *IEEE International Symposium on Circuits and Systems*, pages 501–506, 1991.
- [5] Berthold P.K. Horn. *Robot Vision*. The MIT Press, McGraw-Hill Book Company, 1986.
- [6] B.G.Schunck Horn, B.K.P. Determining optical flow. *Artificial Intelligence*, Vol 17:185–203, 1981.
- [7] Carver A. Mead J. Tanner. An integrated analog optical motion sensor. *VLSI Signal Processing*, 2, page 59, 1986.
- [8] Christof Koch John G. Harris. Analog hardware for detecting discontinuities in early vision. *International Journal of Computer Vision*, 4:211–223, 1990.
- [9] John Wyatt John Harris, Christof Koch. Resistive fuses: Analog hardware for detecting discontinuities in early vision. *Analog VLSI Implementation of Neural Systems*, pages 27–56, 1989.

- [10] John D. Harris Matthew D. Rowley. An edge enhancement technique for analog VLSI early vision application. *IEEE international Symposium on circuits and Systems*, pages 874–879, 1996.
- [11] John G. Harris Matthew D. Rowley. A comparison of three one-dimensional edge detection architectures for analog VLSI vision systems. *IEEE International Symposium on Circuits and Systems*, pages 1840–1843, 1997.
- [12] J. Malik Pietro Perona. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Patter Analysis and Machine Intelligence*, pages 629–639, Vol. 12, 1990.
- [13] John G. Harris Shih-Chii Liu. Generalized smoothing networks in early vision. *Proceedings CVPR '89 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 184–191, 1989.
- [14] Alan Stocker. Smooth optical flow computation by a constraint solving circuit. *INNS Newsletter*, 11, 1998.
- [15] John L. Wyatt. Little-known properties of resistive grids that are useful in analog vision chip designs. *Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits*,, pages 72–104, 1994.