IMAGE PROCESSING PROJECT

FLOW BASED IMAGE ABSTRACTION

GROUP NUMBER 8

Content

1. Introduction

In the field of image processing, the objective is often to enhance or manipulate images to make them appear more realistic. But sometimes, we want to make them look different, more like drawings or paintings. This is where non-photorealistic rendering (NPR) comes in. Instead of copying every detail, NPR simplifies things to highlight what's important.

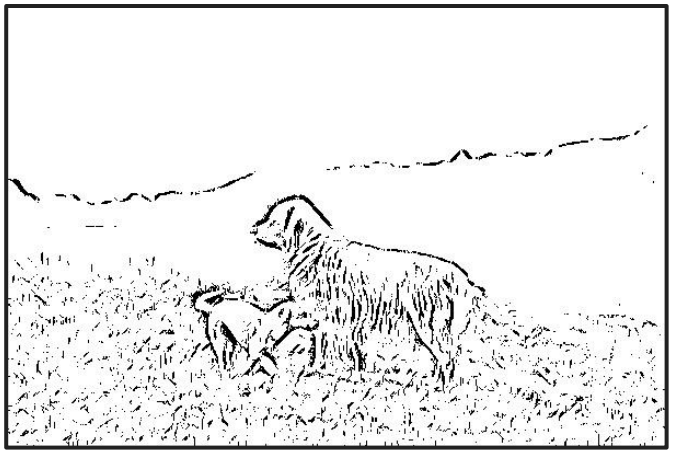
Our project aims to make this process easier by creating a tool that can turn photos into stylized drawings. We use filters to focus on shapes and colors, guided by image's edge flow. This helps to keep the important features while making the image look unique and artistic.

2. Problem Statement

We have to generate a stylized image while preserving its main features. Lines are very useful in presenting the shape of image.

NPR also includes the painting of object surface with a limited palette of colors for better visual information transfer.
So the image abstraction is divided into two parts

1. Line drawing
2. Region smoothing



Overview

We have implemented a flow-driven approach for both line extraction and region smoothing. this involves

1. Flow construction
2. Line extraction based on the Edge Tangent Flow
3. Region smoothing based on the Edge Tangent Flow

We have implemented the DoG filter for line extraction and the bilateral filter for region smoothing. The main difference is that it takes into account the “direction” of the local image structure in shape and color filtering, rather than looking in all directions.

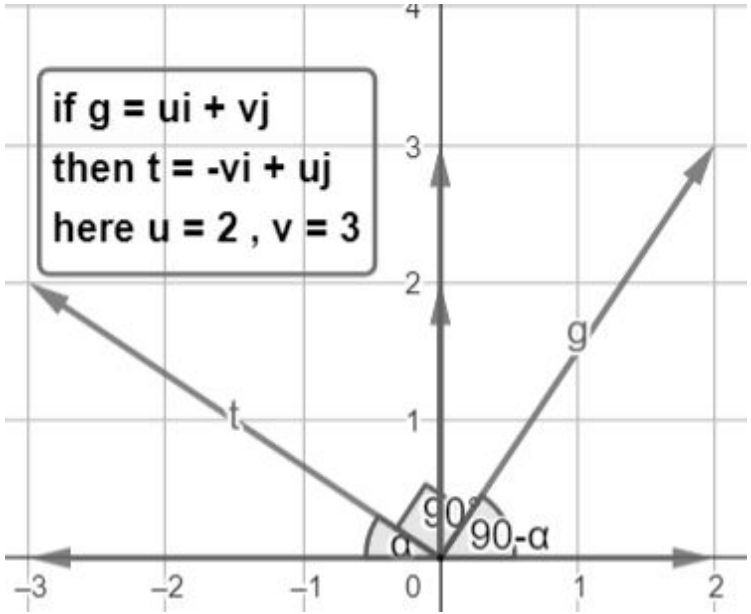
Flow Construction

We have to produce a tangent field t(x) for image I(x) where x is a pixel.
For better results

(1) the flow must describe the salient edge tangent direction in the neighbourhood
(2) The neighbouring vectors must be smoothly aligned except at sharp corners
(3) Important edges must retain their original directions

Steps involved in edge tangent flow construction

1. Initaly gradient map g(x) = ∇I(x), is computed by using sobel operator($sobel_x$ and $sobel_y$) and also normalised gradient magnitude is stored separately.
2. Initial tangent field t(x) is constructed by taking perpendicular to the g(x).



3.The ETF construction filter is thus defined as follows

$$t'(x) = \frac{1}{k} \iint_{\Omega_\mu} \phi(x,y) t(y) w_s(x,y) w_m(x,y) w_d(x,y) dy$$

$\Omega_\mu(x)$ denotes the kernel of radius μ at x, and k is the vector normalisation term.
w_s is spatial weight function which is a box filter of radius μ

$$w_s(x,y) = \begin{cases} 1 & if\ |x-y| < \mu \\ 0 & otherwise \end{cases}$$

w_m is the magnitude weight function, which is defined as

$$w_m(x,y) = [\hat{g}(y) - \hat{g}(x) + 1]/2$$

w_d is the direction weight function

$$w_d(x,y) = |t(x).t(y)|$$

ϕ(x,y) is used to change the direction of tangent to ensure that negihbours with similar orientation but opposite direction can contribute in the edge flow.

$$\phi(x,y) = \begin{cases} 1 & if\ t(x).t(y) > 0 \\ -1 & otherwise \end{cases}$$

4. Filter can be iteratively applied to update ETF incrementally: $t^i(x) \rightarrow t^{i+1}(x)$

How iterations affects the flow.

By keeping other factors same, if the iterations of ETF filters are increased then the dominant edges become more dominant and the vector field gets smoother.

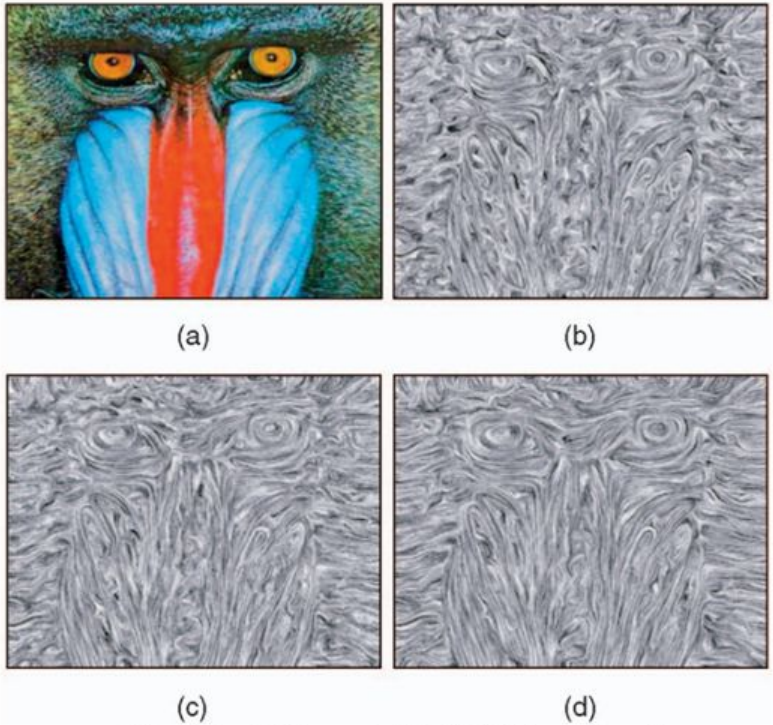
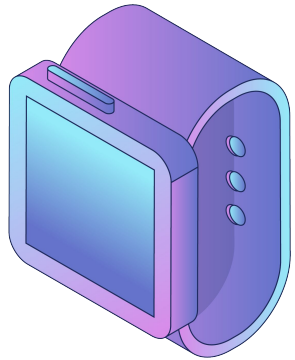


Fig 2.1. (a) input (b) ETF iteration-1
(c) ETF iteration-2 (d) ETF iteration-3

1. Line Extraction

Line extraction or Edge Detection is a low-level operation in Image Processing that involves finding lines in digital images. Conventional edge (or line) detectors are often employed and adapted, such as Canny Edge Detection, Difference of Gaussians filtering, Laplacian of Gaussians and so on. Here we use an updated version of DoG Filter so that the quality of lines is enhanced by steering the DoG filter along the ETF flow.



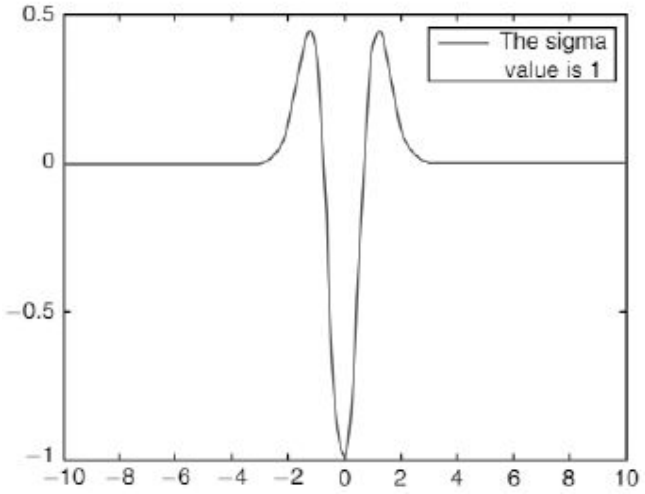
WHY DoG FILTER?

Difference of Gaussians (**DoG**) is a feature enhancement algorithm that involves the subtraction of one Gaussian blurred version of an original image from another, less blurred version of the original.

We build on the DoG edge detection mainly due to its simplicity and the stylistic nature that suits the purpose.

$$\Gamma_{\sigma_1,\sigma_2} = I * (G_{\sigma_1} - G_{\sigma_2})$$

$$G_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$



FLOW BASED DIFFERENCE OF GAUSSIANS

The central idea is to apply a linear DoG filter in this gradient direction as we move along the edge flow. We then accumulate the individual filter responses along the flow, as a way of collecting enough evidence before we draw the conclusion on the “edge-ness.” This allows us to exaggerate the filter output along genuine edges.

The filtering scheme is :

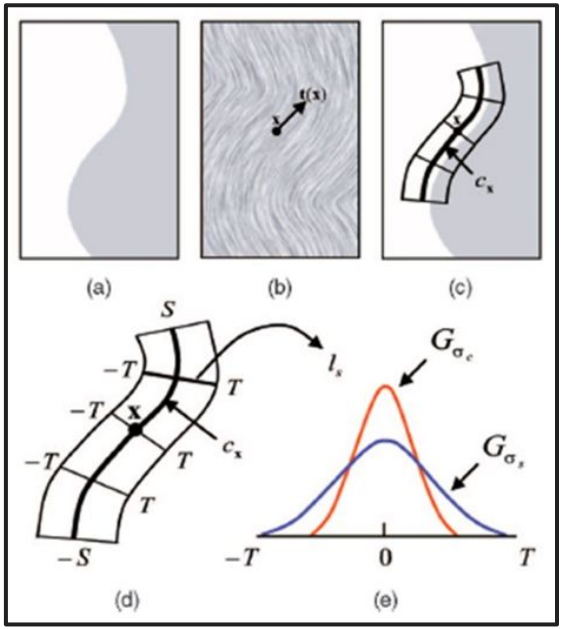
$$H(x) = \int_{-S}^{S} \int_{-T}^{T} I\left(l_{x,s}(t)\right) f(t) G_{\sigma m}(s)\, dt\, ds$$

$$f(t) = G_{\sigma_c}(t) - \rho G_{\sigma_s}(t)$$

$C_x(s)$: flow curve at x, where s is an arc-length parameter.

$l_{x,s}$: A Line segment that is perpendicular to $t(c_x(s))$ and intersecting $c_x(s)$.

$l_{x,s}(t)$: A point on the line $l_{x,s}$ at t distance from the curve.

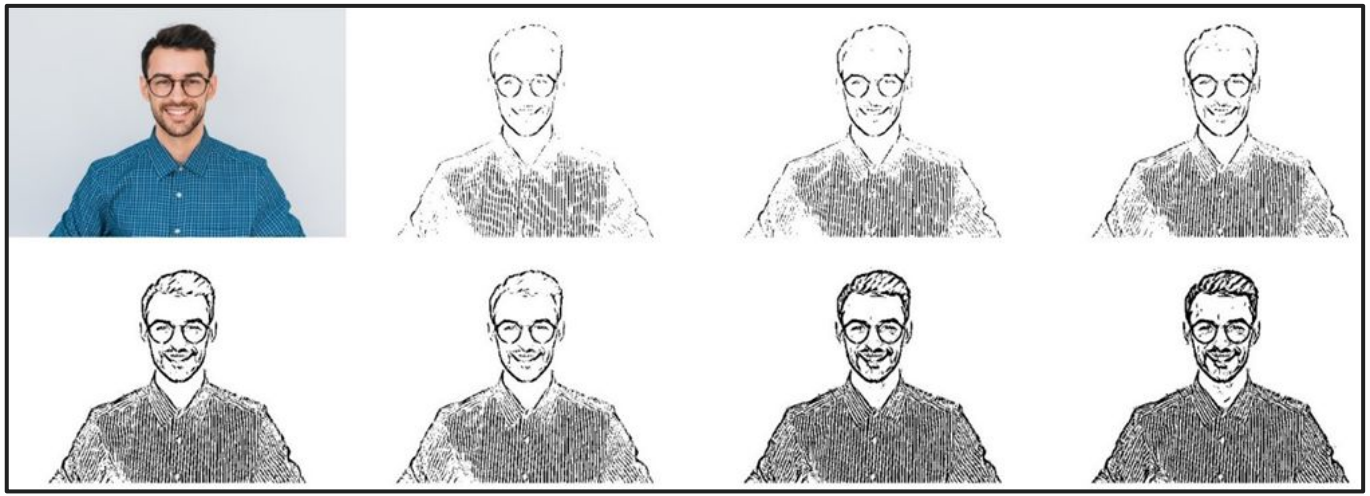


FLOW BASED DIFFERENCE OF GAUSSIANS

Once we obtain the H(x) for a point x, we perform binary thresholding as:

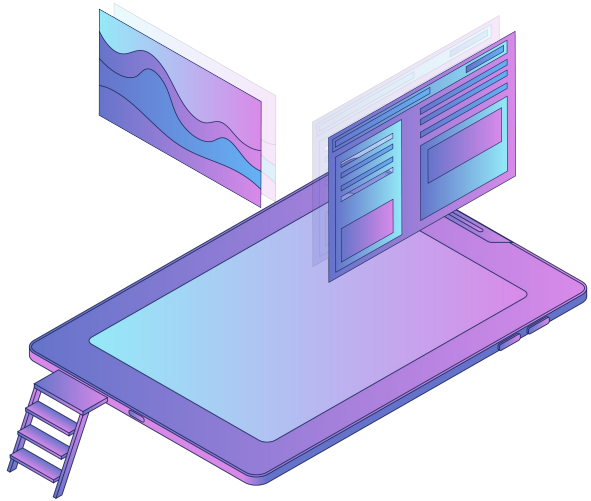
$$\overline{H}(x) = \begin{cases} 0, & if\ H(x) < 0\ , 1 + \tanh\big(H(x)\big) < r \\ 1, & otherwise \end{cases}$$

where r is a threshold in [0, 1]. This binary output is the targeted line illustration.



COMPARISON OF DoG AND FDoG FILTERS

The DoG filter is called the FDoG Filter since it is driven by the vector flow. When compared to the DoG filter, the line coherence is better with FDoG filtering. More lines are included in the picture as the ρ value is increased. In contrast to traditional DoG filters, the FDoG filter uses a large ETF kernel size to extract ETF from a Gaussian-smoothed input image, allowing lines to be constructed from a collection of disconnected points. The FDoG Filter has limits when it comes to handling small-scale non-directional structures, but it is excellent for safeguarding directed structures.



ALTERNATIVE METHODS TO DoG FILTER

Apart from the Difference of Gaussians suggested in the paper, we tried to implement the same idea using other two filters :

1) Laplacian of Gaussians

$$\nabla^2(h(m,n)) = \frac{1}{\sigma^2}\left[\frac{r^2}{\sigma^2} - 1\right]\exp\left(-\frac{r^2}{2\sigma^2}\right)$$

2) Extended Difference of Gaussians

$$XDoG(x) = (1+p)G_\sigma(x) - pG_\sigma(x)$$

5. Region Smoothing

Region smoothing is a technique that Is used in image processing to smooth or blur specific regions or segments of an image while preserving the overall structure and edges. The goal of this technique is to remove unimportant details from the region's interiors while preserving important shapes.

There are multiple purposes of region smoothing technique, some of them are –
1. Enhancing Features
2. Noise Reduction
3. Edge Preservation
4. Selective Smoothing

Among the filters mentioned above for region smoothing in image processing, the bilateral filter is often considered one of the most popular and widely used techniques. Its popularity comes from its effectiveness in smoothing images while preserving edges and fine details.

Bilateral Filter

The bilateral filter is a nonlinear filtering technique used in image processing for smoothing images while preserving edges. It is designed to reduce noise and blur images while retaining important structural information, such as edges and boundaries. The bilateral filter achieves this by considering both the spatial proximity and the intensity similarity of neighboring pixels when computing the weighted average for each pixel.

Working of Bilateral Filter-

• Spatial Proximity Weighting: The filter considers the spatial distance between the center pixel and its neighboring pixels. Pixels that are closer to the center pixel are given higher weights in the averaging process, while pixels farther away receive lower weights. This spatial proximity weighting ensures that the filter focuses more on nearby pixels during the smoothing process.

• Intensity Similarity Weighting: In addition to spatial distance, the filter also considers the similarity in intensity or color between the center pixel and its neighbors. Pixels with similar intensity values to the center pixel are given higher weights, while pixels with significantly different intensity values are given lower weights. This intensity similarity weighting helps preserve edges and fine details in the image.

Flow Based Bilateral Filter (FBL)

***Ce* (Edge Direction Filter)**

$$C_e(x) = \frac{1}{v_e}\int_{-S}^{S} I\big(c_x(s)\big)G_{\sigma_e}(s)h(x, c_x(s), r_e)ds$$

Notations:

***Ce* (*x*):** Output of the linear bilateral filter at pixel x

***ve* :** Weight normalization term. S: Domain over which the integral is calculated.

***cx* (*s*):** Flow curve of the Edge Tangent Flow (ETF).

***I*(*cx* (*s*)):** Input image or signal.

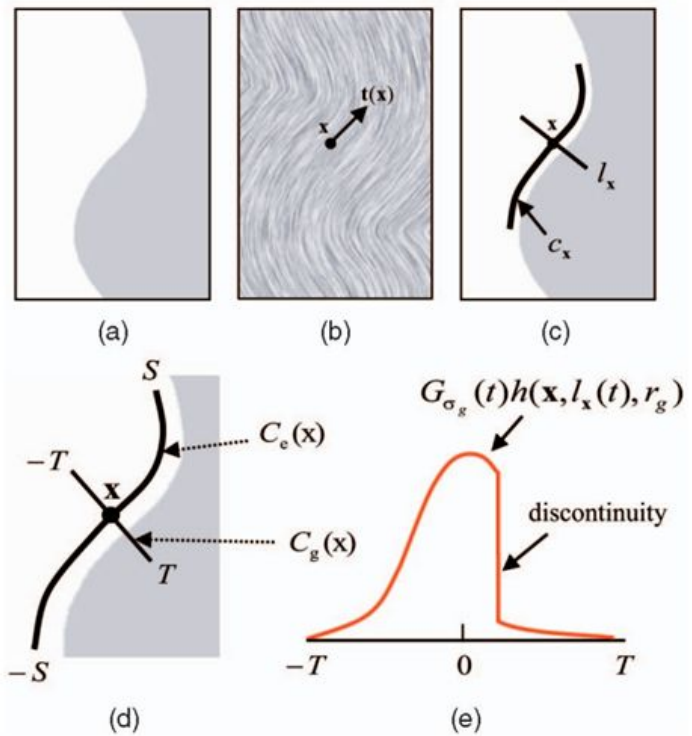
***Gσe* (*s*):** Spatial weight function along the flow axis cx (s).

***h*(*x*, *cx* (*s*), *re*):** Color weight function.

***re* :** Parameter related to color similarity.

***ds* :** Infinitesimal element of integration over the spatial domain S.

***x*(*s*):** Point in the spatial domain within the filter kernel at location s.



$$h(x, y, \sigma) = G_\sigma(||I(x) - I(y)||)$$

The similarity weight function h is defined similarly to the original bilateral filter, except that we compare the colors between the center point x and the points along the main axis cx.

• Ce operates primarily along the edge directions, focusing on preserving and enhancing shape boundaries in the image.

• It protects and cleans up the shape boundaries by emphasizing edges and reducing noise along edge directions.

• The spatial weight function $G\sigma e\,(s)$ along the edge direction helps in preserving edge information, while the color weight function $\hbar(x, cx\,(s), re\,)$ ensures that edges are maintained by considering color similarity.

C_g (Gradient Direction Filter)

$$C_g(x) = \frac{1}{v_g}\int_{-T}^{T} I(l_x(t))G_{\sigma_g}(t)h(x, l_x(t), r_g)dt$$

Notations:
This formulation can also be applied to color images by employing a distance metric in the color space. Similarly, the following equation defines the other linear bilateral filter along the gradient direction. Where **lx(t)** is again an abbreviated notation for **lx,0(t)**, which is the gradient axis at x.

• Cg suppresses color differences within regions and smooths out the interiors of regions.

• It operates along the gradient direction, reducing color variations within homogeneous regions and promoting smoother color transitions.

• The spatial weight function Cg along the gradient direction helps in smoothing out color transitions, while the color weight function $h(x, lx\,(t), rg)$ suppresses color differences within regions.

FBL Filter – Combination of Ce and Cg

The FBL (Flow Bilateral) filter is a combination of Ce and Cg applied iteratively.

By alternately applying Ce and Cg in an iterative fashion, the FBL filter effectively balances the preservation of edges and smoothing of regions.

The iterative application of Ce and Cg allows for efficient edge preservation and region smoothing, making it faster than the full-kernel bilateral filter.

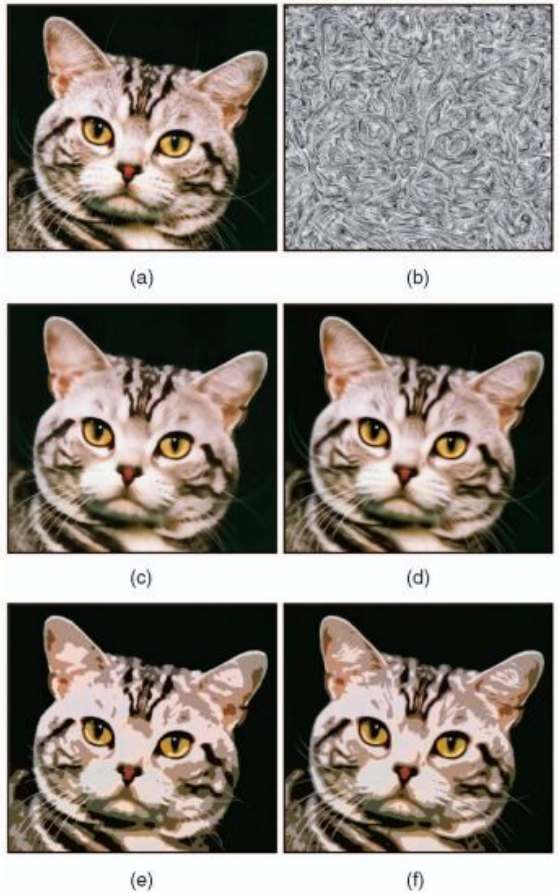
The FBL filter is particularly useful in scenarios where both edge preservation and region smoothing are desired, such as in image abstraction, stylization, or denoising tasks.

In summary, Ce and Cg play distinct roles in edge preservation and region smoothing, respectively. Their combination as the FBL filter allows for efficient and effective image filtering, balancing the preservation of shape boundaries and the smoothing of region interiors.

Comparison of Bilateral and FBL Filter

FBL: Shape enhancement.
(a) Input image.
(b) ETF.
(c) Bilateral filter (σd = 2.0, σr =10, three iterations).
(d) FBL (σe =2.0, re = 10, σg =0.3, rg = 10, five iterations).
(e) Bilateral filter (flattened).
(f) FBL (flattened).

While both the full-kernel bilateral filter and the FBL filter blur the image without destroying the original edges, the full-kernel bilateral filtering does not remove the blocking artifact. On the other hand, the FBL filter restores clean and smooth shape boundaries.



6. Conclusion

We introduced an automated method for image abstraction utilizing a flow-based filtering framework. Specifically, we introduced the (FDoG) and (FBL) filters as innovative solutions to two key challenges in image abstraction: line drawing and region smoothing. Leveraging the Edge Tangent Flow (ETF) for feature preservation, these filters surpass their traditional counterparts by enhancing features and stylization, thereby generating high-fidelity image abstractions. This approach effectively communicates shapes and color, ensuring a superior level of abstraction quality.