효율적인 비디오 카투닝을 위한 인터랙티브 시스템

홍성수[1], 윤종철[2], 이인권[3]
연세대학교 컴퓨터과학과[1 2 3]
maeno@cs.yonsei.ac.kr[1] media19@cs.yonsei.ac.kr[2] iklee@yonsei.ac.kr[3]

Interactive System for Efficient Video Cartooning

Sung-Soo Hong[1],Jong-chul Yoon[2], In-Kwon Lee[3]
Department of Computer Science, Yonsei University[1 2 3]

요 약

Mean shift 는 데이터의 특징을 잘 살려내는 None-parametric 방법으로, 특히 영상처리분야에서 많은 각광을 받아왔다. 하지만 좋은 결과를 보장하는 뛰어난 성능에도 불구하고, 높은 메모리소요와 긴 처리시간에 기인하여, 비디오처리 등의 분야에 적용하기엔 현실적인 제약점이 있다.

상기한 제약점을 극복하기 위해, 본 시스템은 비디오를 분석하여 전경과 후경으로 나눈다. 본 논문은 전경으로 분류된 부분에 대해 각 분리된 개체를 구분하고, 좌표변환(coordinate shift)을 실행하여 연산을 할 비디오의 연산의 규모를 줄이는 방법론을 제시한다. 이러한 처리로 매우 많은 처리시간이 단축됨을 실험을 통해 알 수 있었다. 다음으로, 나뉘어진 전경에 3D mean shift 를 적용하여 생성된 결과물에 대하여 3D cluster data structure 를 생성하고, 이를 이동하여 인터랙티브 에디팅이 가능하도록 하였다.

후경으로 나뉜 데이터는 이미지 한 장으로 축약이 되며, 2D mean shift 기반의 interactive cartooning system 을 통하여 만화화가 된다. 본 논문은 만화 특유의 단순한 톤을 표현하기 위해, 세밀한 분할이 필요한 부분과 그렇지 않은 부분을 따로 구분하여 처리하는 레이어처리방법을 제안한다.

위의 과정을 여러 실사이미지에 적용, 실험해본 결과 기존의 연구결과에 비해 매우 짧은 시간 내에 대상의 특징이 잘 나타낸 양질의 결과물이 생성되었다. 이러한 결과물은 출판, 영상편집분야 등 여러 분야에서 요긴하고 간편하게 사용될 수 있을 것으로 생각된다.

Keyword : Non-photorealistic Rendering, Video Cartooning, Mean-shift, Bezier curve, illustration

1. 서론

만화나 일러스트는 주로 표현하고자 하는 대상물의 관찰을 시작으로 제작된다. 제작자는 대상의 형태와 그 특징을 면밀히 살펴보고, 대상 고유의 특징적인 요소들이 잘 반영이 되도록 결과물을 제작하게 된다. 즉, 카투닝이 된 결과물은 대상의 특징을 잘 표현해야 한다는 것이며, 이러한 관점에서 실사 이미지와 비디오는 제작자가 표현하고자 하는 대상의 있는 그대로를 잘 나타내고 있는, 많

본 연구는 한국전자통신연구원의 정보통신연구개발사업
위탁연구과제지원으로 이루어졌음.

은 의미 있는 정보를 담고 있는 좋은 자료가 된다고 볼 수 있다.

하지만 이러한 일련의 과정을 따라 양질의 결과물을 나타내기 위해서 투입되어야 하는 시간과 인력의 비용은 매우 크다. 영상의 경우, 단 수 초 분량의 영상을 제작하기 위해서도 수 백장의 이미지가 필요하며, 이러한 수 초 분량의 결과물 제작을 위해 짧게는 몇 시간에서부터, 길게는 수 개월의 시간이 걸린다. 이러한 카투닝 작업의 노동집약적인 특징으로 인하여, 실사 이미지와 비디오를 바탕으로 자동화된 공정을 통한 카투닝 결과를 제작하는 방법은 꾸준히 연구의 대상이 되어왔다.

최근, Wang[3]은 Dorin Comaniciu 가 제안한 Mean-shift 기반 이미지분할방법을 사용하여 Video tooning 을 시도하였으며, Kang[5]은 Intelligent scissor 방식을 사용하여 인터랙티브한 이미지 카투닝을 시도하였다. 각각의 연구는 충분한 자동화를 취하기도 하지만, 사용자의 인터랙션에 매우 큰 비중을 두고 있는 특징을 가지고 있다.

본 논문에서는, 이미지와 비디오에서 얻은 정보를 바탕으로, 그 정보의 특징을 잘 살릴 수 있는 카투닝 방법을 소개한다.

Mean-shift 방식[1, 4]은 비모수(none-parametric) 통계적 추론방식을 사용한 이미지, 비디오 분할방법이다. 비모수적 추론방식의 특성상, 연산하고자 하는 결과에 대한 데이터의 분포를 입력된 값을 통해 추론하게 되므로, 대상의 형태와 고유적인 특징을 잘 나타낼 수 있는 매우 적합한 방식이라고 볼 수 있다. Mean shift 방식은 또한, 대역폭의 조정이 가능하여 세밀한 정도를 조정할 수 있으며, 차원의 확장 역시 가능하여, 이미지와 비디오에 고루 사용될 수 있는 장점이 있다. 이러한 여러 장점에도 불구하고, Mean-shift 는 현실적으로 사용하기 곤란한 성능적 한계가 있다. 비디오 프로세싱의 경우 데이터셋은 매우 방대하다. 하지만 Mean-shift 는 global optimization 문제를 해결하기 위한 방식으로, 모든 비디오의 픽셀들을 읽어야 하기 때문에 엄청난 메모리와 시간을 소요한다. 이러한 Mean-shift 의 특징으로 인해, 저해상도의 비디오 단 십 수 프레임을 프로세싱할 수 있을 뿐이라고 알려져 있다[3]. 본 논문은, 이러한 한계점을 해결하기 위한 데이터셋의 축소에 대한 방법론을 제시하여, 고해상도의 비디오도 Mean-shift 프로세싱을 할 방법을 제안한다.

또한, 본 논문에서는 이미지, 비디오 카투닝에서 사용자 인터랙션을 통한 결과물의 완성도를 높이는 방법론을 제안한다.

2. Mean- shift

k-means 등의 기존에 존재하던 이미지 분할 방식은 입력된 데이터에 대한추정(seeding point 등)을 통하여 결과가 어떠할 지 추측을 한다. 하지만 이러한 방식과 달리, Mean-shift 는 어떠한 초기적인 추측도 하지 않는다. 이러한 성질은, Mean-shift 가 최대한 대상의 형태와 고유적인 특징을 잘 나타낼 수 있도록 한다. Mean-shift 는 각각의 데이터 포인트(비디오나 이미지의 픽셀)에서 비슷한 포인트들간의 지역적인 조밀도를 바탕으로 결과를 추측한다. 더욱 자세히 이야기하자면, mean shift 알고리즘은 비슷한 데이터들 간의 지역적인 조밀도의 gradient 를 추측한다. (여기에서 지역적인 조밀도는 입력된 이미지나 비디오에서 구성된 확률밀도함수로 구성된다.) 계산된 gradient vector 는, 초기 위치에서 Mean-shift point 를 도출시키며, 연산된 포인트에서 또 다시 gradient vector 를 구하는 반복적인 실행을 통하여 지역적 조밀도 분포에서 가장 높은 정점(peak)를 찾게 된다. 같은 정점에 수렴되는 모든 포인트들은 같은 세그먼트에 속한 멤버로 취급된다. 모든 포인트가 각각 지역적인 정점을 찾음으로 인해 데이터는 분할이 될 수 있다.

Mean-shift 를 사용하여 이미지와 비디오를 분할하기 위하여, 우리는 전체적인 절차를 두 단계로 나누어 생각해 볼 수 있다. 첫 번째 단계는 각 pixel 에 영향을 미치는 범위를 나타내는 kernel 을 설정하는 것 이다. 이 kernel 은 픽셀들간의 정점(peak)을 찾기 위한 거리를 설정한다고 볼 수 있다. 거리는 공간적인(비디오의 경우 시간축도 포

함된다.)거리와 컬러값에 관련된 거리로 구성이 된다. 이 값에 따라 각 픽셀이 정점을 찾을 때 참조하는 범위가 달라진다. 두 번째 단계에서, 각 픽셀들에 대해 Mean-shift point $M(X_i)$ 를 찾고, 그 포인트에서 지역적인 조밀도의 gradient 를 찾은 다음, 정점으로 이동할 때까지 반복적으로 두 번째 단계를 반복적으로 실행시켜준다.

수학적으로, 포인트 x 에서의 다변수 kernel density estimator 는 아래와 같이 정의된다.

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_H(x - x_i) \quad (1)$$

n 은 참고하고자 하는 데이터 셋들의 숫자이며, x_i 는 각 데이터 포인트를 나타낸다.

$$K_H(x) = |H|^{-1/2} K(H^{-1/2}x) \quad (2)$$

H 는 d*d 의 대칭 대역폭 행렬이며, K(z)는 kernel 을 나타내는 function 으로,

$$K(x) = ck(\| x \|^2) \quad (3)$$

와 같은 형태로 나타나게 된다. c 는 정규화를 위한 상수이며, 만약 방사형으로 영향력이 같은 구형의 kernel 을 사용하게 될 경우, $H = h^2 I$ 가 성립이 되며, (1)번식은 아래와 같이 고쳐진다.

$$\hat{f}(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \quad (4)$$

이미지와 비디오를 위한 프로세싱의 경우, 고려하고자 하는 space 는 두 개의 독립적인 영역으로 나뉘게 된다. 하나는 spatial/lattice 영역이고, 다른 하나는 range/color 영역이다.

영상의 spatial/lattice 영역을 지정하기 위해서, 각 포인트 들을 p 차원의 공간상에 있는 격자에 넣게 된다. 이미지의 경우 p=2 (x, y)가 되고, 비디오의 경우 p=3 (x, y, t)이 될 것이다. 각 pixel 간의 거리는 spatial domain 을 구성하게 된다. 또, 이미지의 range/color 영역을 구성하기 위해서, RGB 의 컬러값을 LUV 로 고쳐서 사상시키게 된다. 이는 컬러값의 Euclidian Distance 가 RGB 일 경우보다 LUV 일 경우 인간이 느끼는 Distance 와 가깝기 때문이다[1]. 두 영역의 다른 성질에 기인하여, kernel 은 두 부분으로 나뉘게 되어 곱으로 표현되게 된다.

$$K_{h_s,h_r}(x) = \frac{C}{h_s^{\,p} h_r^{\,q}} k^s(\| \frac{x^s}{h_s} \|^2) k^r(\| \frac{x^r}{h_r} \|^2) \quad (5)$$

x^s 와 x^r 은 spatial 과 range 의 벡터가 되고, h_s 와 h_r 은 각각 두 영역에서의 지정한 대역폭이 된다. (5)번의 커널을 사용하면, kernel density estimator 는 아래와 같이 재정의 된다.

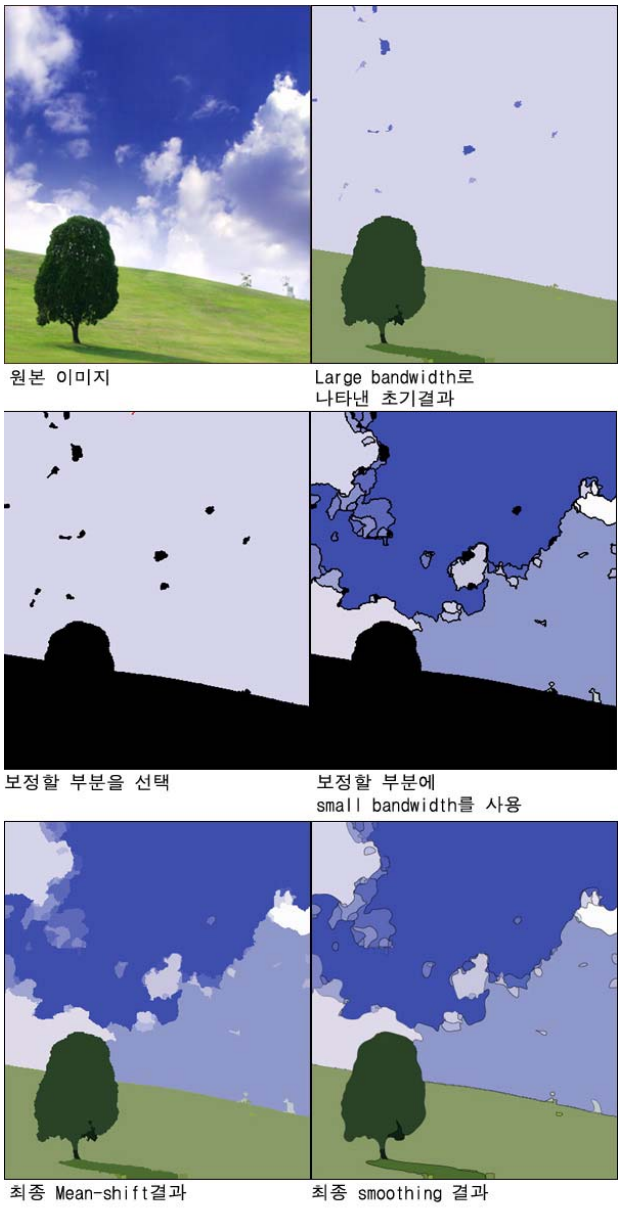
$$\hat{f}(x) = \frac{c}{n(h^s)^p (h^r)^q}\sum_{i=1}^{n} k^s(\| \frac{x^s - x_i^s}{h^s} \|^2) k^r(\| \frac{x^r - x_i^r}{h^r} \|^2) \quad (6)$$

(6)의 kernel density estimator 로 각 픽셀의 $M(X_i)$ 와 정점을 찾을 수 있으며, 픽셀에 대한 분할이 가능해진다.

3. 이미지 카투닝

앞서 언급하였듯, Mean-shift 는 대역폭(bandwidth)을 통하여 각 point 가 density estimation 에서 참조를 하는 데이터들의 범위를 지정할 수 있고, 이로서 결과의 세밀도를 조절할 수 있기 때문에 카투닝에 있어서 적합한 방법이라고 하였다. 본 논문에서는, 사용자의 간단한 인터랙션으로, 세밀하게 표현하고 싶은 부분을 선택하여 세밀하게 결과를 제작하여 주는 인터페이스를 개발하였다.

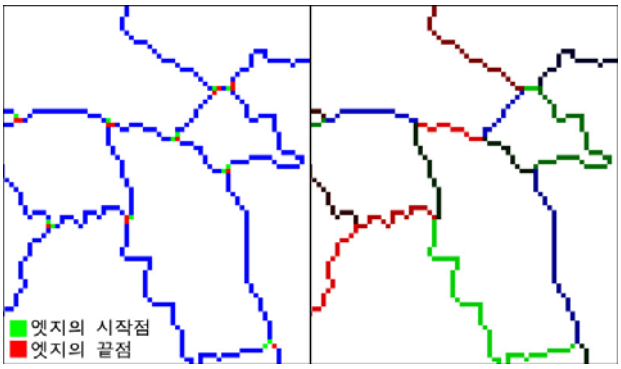
그림 1 은 이러한 프로세스를 잘 설명해 준다. 사용자는 입력된 이미지에 큰 bandwidth 를 적용하여, 각 segment 의 크기가 크게 결과를 설정해준다. (그림 1 의 右上) 사용자는 초기결과에 대해, 더 자세히 표현하고 싶은 부분을 선택, (그림 1 의 中左) 선택한 부분에 대한 레이어를 생성시키고, 적당한 bandwidth 로 세밀도를 조절한다. (그림 1 의 中右) 이때, 새로 생성된 레이어에서, 사용자에 의해 선택이 되지 않았던 부분(고쳐지지 않을 부분)은 선택된 부분과 color distance 가 가장 큰 컬러값으로 설정, mean-shift 작업 이후에도 고쳐진 부분과의 경계면이 바뀌지 않도록 처리를 해준다.



<그림 1: 인터랙티브한 레이어기반 Mean-shift>

결과로 제작된 Mean-shift 가 완료된 이미지(그림 1 의 左下)는 각 분할된 세그먼트 하나를 graph 의 node 로 하는 자료구조를 생성시키고, 인접한 node 들의 좌표를 추출, Bezier curve 화 시켜 최종적인 결과물을 만들게 된다. (그림 1 의 右下)

그림 2 는 Bezier curve 로 바뀌면서 smoothing 이 되는 process 를 보여준다. 각 분할된 세그먼트 사이를 찾아가면서 좌표화하여, curve approximation 을 실행한다.



<그림 2: segment 와 edge 의 Bezier curve 화>

4. 비디오 카투닝

비디오는, 이미지에 비해 훨씬 많은 데이터를 가지고 있다. Mean-shift 의 구현측면에서의 특성상, 고려하고자 하는 데이터를 모두 메모리에 올려야 하므로, 많은 메모리의 공간이 필요하며, 시간 역시 많이 걸린다. 이러한 한계점은, mean-shift 의 비디오 프로세싱에서의 큰 한계점으로 알려져 있다.

이러한 한계점을 극복하기 위해서, 본 논문에서는 데이터의 규모를 줄여서 연산시간과 메모리를 효율적으로 사용할 수 있는 방법론을 제안한다

데이터의 규모를 줄이기 위해, 본 논문에서는, 비디오를 분석, 전경과 후경으로 구분한다. 비디오가 고정이 되어있고, 물체가 움직이고 있다는 가정하에, 후경은, 비디오의 모든 프레임의 같은 위치에 존재하는 픽셀의 중간값이라고 추정할 수 있다.

$$BackGround = \sum_{j=1}^{Height} \sum_{i=1}^{Width} \left(\sum_{f=1}^{TotalFrame} Mean(f, i, j) \right) \quad (7)$$

위의 식을 기초로, 구해진 후경 이미지는 앞서 언급하였던 이미지 카투닝시스템에 의해 만화화가 된다.

그림 3 은 위에서 설명한 비디오에서 배경을 적출하여, 카툰화를 실행하는 과정을 잘 나타내어 주고 있다. 각각, 소스비디오와 적출된 배경 이미지, 그리고 카툰화된 이미지를 보여주고 있다. (左上, 右上, 下 순서)

<그림 3: 전경의 추출과 카툰화>

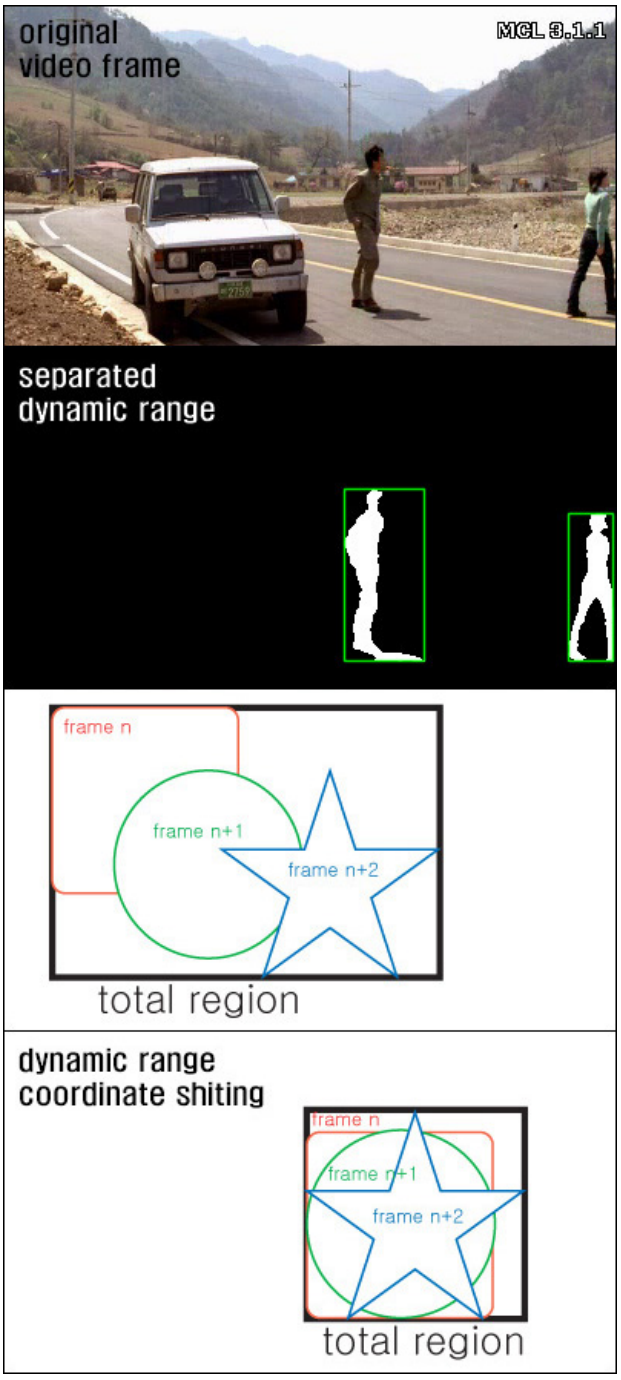
생성된 배경이미지를 기반으로, color distance 를 고려하여 움직이는 영역을 연산하면, 비디오는 각각 동적 영역(dynamic range)과 정적 영역(static range)로 나뉘어진다.

여기에서, 데이터의 규모를 줄이기 위해 나뉘어진 동적 영역을 연결된 세그먼트 별로 구분하여 개체화 시키고, 각 개체 별로 Mean-shift processing 을 실행시킴으로써, 많은 계산적 이득을 볼 수 있었다.

$$M(D_{Video}) = num(\sum_{f=1}^{F_{Total}} \sum_{i=1}^{F_h} \sum_{j=1}^{F_w} D(f,i,j))$$
$$M(D_{Segm}) = num(\sum_{f=S_s}^{S_e} \sum_{i=1}^{S_h} \sum_{j=1}^{S_w} D(f,i,j) \quad (8)$$

$M(D_{Video})$ 은 비디오를 Mean shift 하였을 경우의 데이터의 숫자를 나타내며, $M(D_{Segm})$ 는 개체를 구분하여 Mean shift 하였을 경우의 데이터 셋의 숫자를 나타낸다. F_h, F_w 는 각각 비디오프레임의 넓이와 높이를 나타내고, F_{Total} 은 비디오의 총 프레임 수를 나타낸다. 또한, S_h 와 S_w 는 각 개체의 넓이와 높이 값을, S_s 와 S_e 값은 개체의 시작프레임과 끝 프레임을 나타낸다. $\frac{M(D_{Segm})}{M(D_{Video})}$ 는 데이터의 감소율을 나타내며, 일반적으로 고정된 비디오라는 가정하에 1/2~1/20 정도의 값이 산출되었다.

한편, 상기 방법으로 Mean-shift 를 적용시킬 때, 연산을 줄이기 위하여, 각 frame 별 동적 영역을 적당하게 shifting 하는 방법을 고안, 더욱 많은 계산적 이득을 얻을 수 있었다. 이 때, 정적 영역에는 앞서 생성한 카투닝된 이미지를 사용을 해 준다. 위의 방법으로, 예제 비디오에서 데이터의 규모는 8532000 개의 픽셀에서 460160 개의픽셀로 줄었으며, 연산시간은 Mean-shift 연산시간을 기준으로 6 시간에서 13 분으로 단축되었다.



<그림 4: 데이터 규모의 감소>

그림 4 에는 데이터 규모의 감소에 대한 설명이 포함되어 있다. 위에서 두 번째의 그림은 분할된 동적 영역을 나타내고 있으며, 위에서 세 번째와 네 번째의 그림은 동적 영역의 좌표를 수정하여 Mean-shift 연산을 해야 하는 total region 의 크기가 줄어 들었음을 설명하고 있다.

본 논문에서는, 각 개체 별 Mean-shift processing 이후, 시간적 연계성의 보완과, 양질의 결과를 위한 간단한 user interaction 을 위한 인터랙션시스템을 개발하였다. 이러한 인터랙션은, Mean- shift 이후 결과물이 조악하게 보일 수 있는 아주 작은 세그먼트의 병합 등이나, 세그먼트간 곡률의 조정 등의 기능을 포함하고 있다.

5. 결론

Mean-shift 를 기반으로 한 이미지와 비디오 분할은 매우 좋은 성능을 보장하는 것으로 알려져 있지만, 메모리의 시간의 제약 때문에 특히 비디오 프로세싱에서는 사용이 힘들었다. 본 논문에서는 고해상도의 긴 비디오에서도 Mean-shift 를 사용하여 비디오 프로세싱을 할 수 있는 방법론을 제안하였다. $\frac{M(D_{Segm})}{M(D_{Video})}$ 의 값에 따라 제안하는 시스템의 성능의 차이가 나지만, 일반적으로 좋은 성능적 향상을 보여주고 있다. 또한, 이미지 카투닝에서의 레이어 기반 인터랙티브 시스템도 양질의 결과물이 산출될 수 있음을 보여주었다.

비디오 프로세싱에서의 성능향상은 있었으나, $\frac{M(D_{Segm})}{M(D_{Video})}$ 에 종속되어 성능 향상이 나타났으므로 이러한 종속성을 떨어트려, 수치가 높아도 비교적 더 좋은 성능을 보일 수 있는 방법론에 대한 탐구는 의미 있는 연구가 될 것이며, 전경과 후경을 구분하는 프로세싱의 보완은 비디오의 시간 연계성(time coherency)의 향상에 큰 도움이 될 것이다. 또한, 동적 영역의 artistic stylization 에 대한 연구도 좋은 결과물을 내게 할 수 있는 방법이 될 것이다.

참고문헌

[1] Comaniciu ,D. , AND MEER, P 2002. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence 24,5, 603-619

[2] Jue Wang, Bo Thiesson, Yingqing Xu and Michael Cohen., Image and video segmentation by anisotropic kernel mean shift. ECCV 2004, Prague

[3] Jue Wang, Yingqing Xu, Heung-Yeung Shum and Michael Cohen. Video Tooning. ACM Trans. on Graphics (Proc. of SIGGAPH2004), Vol. 23, No. 3, p. 574-583, 2004

[4] Fukunaga, K. , AND Hostetler, L. 1975. The Estimation of the Gradient of a Density Function, with Applications in pattern recognition, IEEE Trans. Information Theory 21,32~40

[5] H. Kang, W. He, C. Chui, U. Chakraborty. "Interactive Sketch Generation". The Visual Computer, Vol. 21, No. 9, pp. 821-830, 2005. (Also presented in Pacific Graphics 2005)

[6] Yin Li, Jian Sun, Heung-Yeung Shum. Video Object Cut and Paste. ACM Trans. on Graphics (Proc. of SIGGAPH2005)

[7] DeManthon, D., Megret, R.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (2000) 142-151

[8] Wnad, M., Jones, M.: Kernel Smoothing.