



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

변 해 원 교수지도  
석사학위 청구논문

특징 기반의  
점묘화 동영상 생성 기법 연구

2015

성신여자대학교 대학원

컴퓨터학과

오 슬 기

특징 기반의  
점묘화 동영상 생성 기법 연구

변 혜 원 교수지도

이 논문을 석사학위논문으로 제출함

2014년 11월

성신여자대학교 대학원  
컴퓨터학과  
오 슬 기

# 인 준 서

오늘기의 석사학위 논문으로 인준함.

2014년 11월

심 사 위 원 홍 의 석 인

심 사 위 원 홍 승 필 인

심 사 위 원 김 태 훈 인

성신여자대학교 대학원

## 논문개요

본 논문에서는 영상의 특징 요소가 반영된 점묘화 동영상 생성하는 시스템을 제안한다. 그동안 컴퓨터 그래픽스(Computer Graphics) 분야에서 점묘 기법 연구는 대부분 단일 이미지를 대상으로 한 연구가 주를 이루었으며, 상대적으로 동영상을 대상으로 한 연구는 미비한 상태이다. 기존의 동영상을 대상으로 한 점묘화 연구[1]는 이미지의 밝기에 따라 점의 밀도를 조절하여 점을 고르게 분포하고, 부가적으로 영상의 특징 표현을 위해 에지(edge) 부분에 점이나 패턴을 추가하는 기법을 제시했다. 하지만 이 방법만으로는 움직이는 대상의 특징을 효과적으로 나타내기 어렵다. 본 연구에서는 단일 이미지 점묘화 기법 중 헤드컷(head-cut) 일러스트 작품처럼 방향성 있는 점 배치를 통해 영상의 특징을 강조한 연구[2]에서 착안하여, 움직이는 대상의 특징 요소를 반영한 점묘화 동영상 생성 시스템을 제안한다. 점묘화 동영상 생성 시스템은 크게 이미지 전처리, 특징 요소를 반영한 점의 배치, 점묘화 동영상 렌더링의 세 단계로 구성된다. 이미지 전처리 단계에서는 점의 밀도를 결정하기 위한 그레이 이미지와 특징 흐름을 반영하기 위한 특징 라인 이미지, 그리고 동영상의 시간적 일관성을 유지하기 위한 프레임 간 차이 이미지를 생성한다. 특징 요소를 반영한 점의 배치 단계에서는 영상의 밝기와 에지 정보를 사용하여 점의 초기 위치를 결정하고, 특징 라인에 영향을 받도록 점을 분포시킨다. 점묘화 동영상 렌더링 단계에서는 각 프레임 별 점의 좌표와 색상 정보를 바탕으로 점을 렌더링 하고 점묘화 동영상을 생성한다. 본 논문에서 제안하는 기법을 통해 움직이는 대상의 특징 표현이 향상된 점묘화 동영상을 생성할 수 있다.

# 목 차

## 논문개요

I. 서론	1
II. 관련연구	4
1. 단일 이미지 점묘화 생성 기법 연구	4
2. 동영상 점묘화 생성 기법 연구	9
III. 문제점 분석 및 연구 방법	10
IV. 점묘화 동영상 생성 시스템	11
1. 시스템 개요	11
2. 이미지 전처리	13
3. 특징 요소를 반영한 점의 배치	20
4. 점묘화 동영상 렌더링	32
V. 실험 및 결과	33
VI. 결론 및 향후 연구	44

## 참고문헌

## ABSTRACT

## 표 목 차

[표1] 특징 요소 구성.....	13
[표2] 점의 초기 집합 구성.....	21
[표3] Site 점(좌)과 Voronoi 점의 정보(우).....	22
[표4] 프레임 별 Site 점의 정보를 기록하는 텍스트 파일 구조.....	32
[표5] 그림21의 점묘화 이미지 파라미터.....	33
[표6] 그림23의 점묘화 이미지 파라미터.....	35
[표7] 점묘화 동영상 렌더링 파라미터.....	37
[표8] 수행 시간 측정 결과.....	42

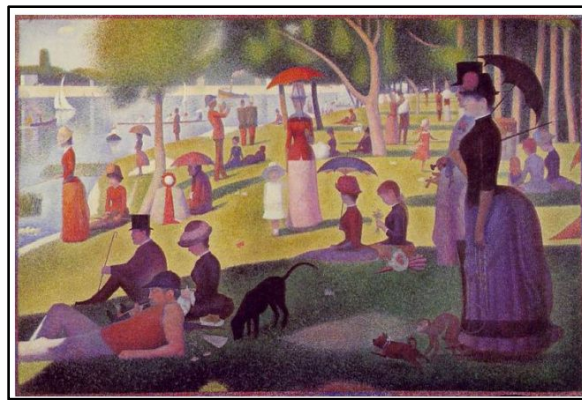
## 그림 목 차

[그림1] 조르주 쇠라의 포인틸리즘 작품.....	1
[그림2] 스티플링 기법으로 그린 점묘화 작품.....	2
[그림3] 조르주 쇠라 작품의 특징을 표현한 결과[4].....	4
[그림4] 조르주 쇠라 작품에 나타나는 색상 분포를 적용한 결과[5].....	5
[그림5] 타원 점을 이용한 포인틸리즘 기법[6].....	5
[그림6] 보로노이 다이어그램을 이용한 반자동 점묘화 시스템[7].....	6
[그림7] 가중치 보로노이 다이어그램을 이용한 점묘화 생성 기법[8].....	6
[그림8] 그래프를 이용한 점묘화 생성 기법[9].....	7
[그림9] 특징 흐름을 이용한 헤드컷 스타일 생성 기법[2].....	7
[그림10] 예제 기반의 스티플링 기법[11].....	8
[그림11] 스트럭처 그리드를 이용한 헤드컷 스타일 생성 기법[10].....	8
[그림12] 동영상에 특화된 컬러 점묘화 연구[1].....	9
[그림13] 점묘화 동영상 생성 시스템 개요.....	12
[그림14] 그레이 이미지 생성 과정.....	14
[그림15] 특징 라인 이미지 생성 단계.....	15
[그림16] 반전된 에지 이미지.....	16
[그림17] 거리 변환 이미지.....	16
[그림18] 특징 라인 이미지.....	17
[그림19] 점의 추가를 위한 이미지(좌)와 삭제를 위한 이미지(우).....	19
[그림20] 점의 배치 단계.....	20
[그림21] 입력 이미지, [1]의 방법 결과, 본 연구 방법 결과.....	34
[그림22] 점의 크기에 따른 결과.....	34

[그림23] 자동으로 초기 Site 점 개수를 설정한 점묘화 결과.....	36
[그림24] 점묘화 동영상 결과1.....	38
[그림25] 점묘화 동영상 결과2.....	39
[그림26] 점묘화 동영상 결과3.....	40
[그림27] 점묘화 동영상 결과4.....	41
[그림28] 누적 수행 시간 측정 결과 .....	43

## I. 서론

예술에서의 점묘는 펜이나 잉크를 이용하여 무수히 많은 점을 찍어 표현하는 미술 기법이다. 점묘는 표현 방식에 따라 크게 두 가지로 구분할 수 있다. 첫 번째는 포인틸리즘(Pointillism)이다. 포인틸리즘은 균일한 필촉으로 작은 색점을 찍는 표현 기법이다. 이 기법은 19세기 화가인 조르주 쇠라의 작품[그림1]에서 나타나는 순수한 색점들 간의 상호 작용으로, 색채를 물리적으로 혼합하는 방식 보다 더욱 강렬하게 색채의 시각적 진동효과를 창출하는 과학적 원리에 기초를 두고 있다.<sup>1)</sup>



[그림1] 조르주 쇠라의 포인틸리즘 작품

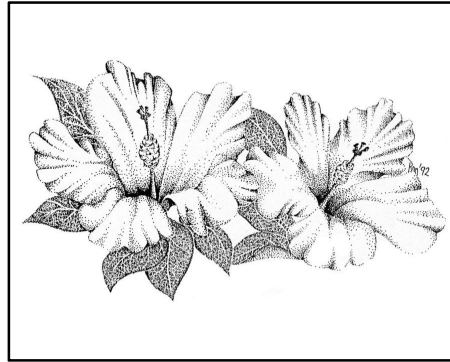
※출처: 네이버캐스트, “조르주쇠라”,

[http://navercast.naver.com/contents.nhn?rid=51&contents\\_id=2690](http://navercast.naver.com/contents.nhn?rid=51&contents_id=2690), 2014년 12월 06일

두 번째는 점의 크기와 간격을 달리하여 점을 촘촘히 찍어 여러 가지 다양한 톤을 나타내는 방법인 스티플링(Stippling)이다.<sup>2)</sup> [그림2]은 점의 밀도를 조절하여 톤을 표현한 스티플링 작품이다.

1) “점묘주의 Pointillism”, <세계미술용어사전>, 월간미술, 1999, p.401.

2) 이안 심프슨, 최기득 역, 『드로잉 기법』, 예경, 1994, p.62.



[그림2] 스티플링 기법으로 그린 점묘화 작품

출처: 구글 이미지 검색, “stippling technique”,

<http://mist-weave.deviantart.com/art/L-O-N-G-I-O-S-T-161764280>, 2014년 12월 06일

화가가 점을 일일이 그려 하나의 점묘화를 완성하기 위해서는 많은 시간과 노력이 필요하다. 뿐만 아니라 그림을 수정하는 작업도 쉽지 않다. 이와 같은 단점을 해소하기 위해, 컴퓨터 그래픽스의 한 분야인 비사실적렌더링(Non-Photorealistic rendering, NPR)[3]에서는 컴퓨터 기술을 사용하여 미술 기법을 재현하는 연구가 최근까지 활발히 진행되고 있다. 비사실적렌더링(NPR) 분야의 포인틸리즘 기법 연구에서는 특히 신인상주의 대표 화가인 조르주 쇠라의 독특한 미술 기법을 표현하기 위한 다양한 연구[4][5]가 이루어졌다. 고전적인 포인틸리즘 기법의 재현뿐만 아니라 최근에는 다양한 스타일의 점을 이용한 포인틸리즘 기법[6]이 제안되었다. 스티플링 기법 연구에서는 보로노이 다이어그램(Voronoi Diagram)을 이용한 반자동(semi-auto) 점묘화 생성 시스템[7], 입력 영상의 톤을 반영한 점묘화 생성 기법 연구[8], 그래프를 이용하여 에지 부분에 특징을 강조한 연구[9], 영상의 방향성 있는 특징을 표현한 연구[2][10], 특정 화가의 작품을 예제로 하여 점묘화를 생성하는 연구[11] 등 다양한 연구가 제안되었다. 단일 이미지 기반의 연구뿐만 아니라 점차적으로 비디오 기반의 연구들이 등장하기 시작

했다. 하지만 비디오에 특화된 점묘 기법에 관한 연구는 아직 부족한 실정이다. 또한 기존의 동영상 점묘화 연구 방법[1]은 에지 부분에 점을 추가로 생성하거나, 패턴을 추가하는 방식으로 특징을 강조했지만, 이 방법만으로는 움직이는 대상의 특징을 효과적으로 표현하는데 한계가 있다. 본 연구에서는 특징 기반의 점묘화 동영상 생성 기법을 제안한다. 영상으로부터 추출한 다양한 특징 요소를 점의 배치에 적용하여, 특징 표현이 개선된 점묘화 동영상을 얻는다.

## II. 관련연구

본 장에서는 논문과 관련하여, 단일 이미지를 대상으로 하는 점묘화 생성 기법과 동영상에 대상으로 하는 점묘화 생성 방법에 관한 연구를 소개한다.

### 1. 단일 이미지 점묘화 생성 기법 연구

포인틸리즘 기법에 관한 연구에서 Yang[4]등은 신인상주의 화가 ‘조르주 쇠라’의 작품을 분석하여, 작품에 나타나는 중요한 특징인 보색 (complementary color)과 할로 효과(halo effect)를 표현하는 방법을 제안했다. 이 연구는 점묘주의에 관한 효과를 시뮬레이션 하는 첫 번째 연구라는 점에서 의미 있는 연구이다. 하지만 보색을 표현하는데 있어서 여전히 쇠라의 접근 방법과 차이가 있고, 많은 파라미터들을 조절해야 했다[그림3].



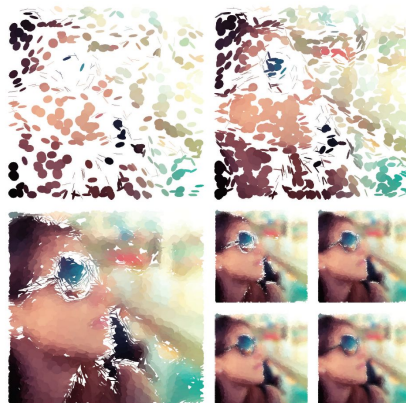
[그림3] 조르주 쇠라 작품의 특징을 표현한 결과[4]

Wu[5]등은 조르주 쇠라가 그린 그림의 색상 분포를 통계적으로 분석하여 색상 모델을 만들었다. 색상 모델을 기초로 조르주 쇠라 스타일의 점묘화를 생성하는 방법을 제안했다. 이 연구에서는 실제 작품과 비교를 통해 신뢰성 있는 결과를 제시했다[그림4].



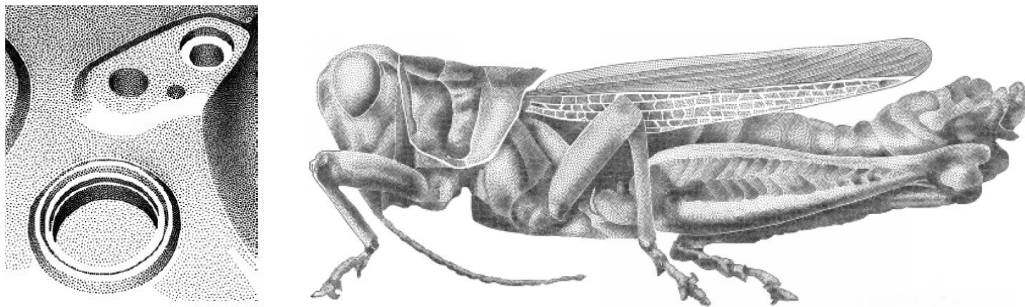
[그림4] 조르주 쇠라 작품에 나타나는 색상 분포를 적용한 결과[5]

Dongxiang[6]은 타원 모양의 점을 이용하여 자연스러운 포인틸리즘 스타일의 점묘화를 만드는 기법을 제시했다. 이 연구에서는 방향 맵(direction map)과 돌출 맵(Saliency map)을 이용하여 타원의 속성을 조절하고, 레이어 계층 별로 점을 렌더링 하였다. 고전적인 포인틸리즘 기법과 달리, 타원의 속성을 조절하여 스타일 변화를 줄 수 있는 방법이다[그림5].



[그림5] 타원 점을 이용한 포인틸리즘 기법[6]

스티플링(Stippling)에 관련된 연구를 살펴보면, Deussen[7]등은 반자동 점묘화 생성 시스템을 제안했다. 디더링(dithering) 알고리즘을 이용하여 초기 점을 생성하고, 보로노이 다이어그램을 사용하여 점을 고르게 분포시켰다. 이 방법은 실제 화가가 점묘화를 그리는 데 소비하는 시간을 수 시간으로 크게 단축하였다. 그러나 여전히 오랜 시간이 소요되며, 사용자가 조절해야 하는 부분들이 많다는 것이 단점이다[그림6].



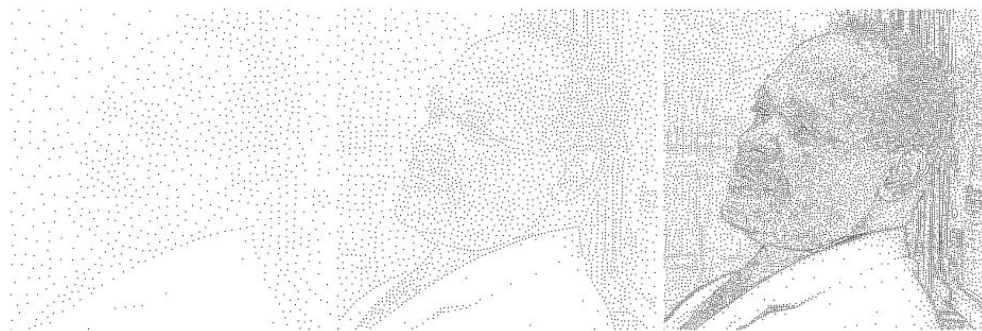
[그림6] 보로노이 다이어그램을 이용한 반자동 점묘화 시스템[7]

Secord[8]는 [7]의 후속 연구로, 가중치 기반의 보로노이 다이어그램을 사용한 점묘화 생성 기법을 제안했다. 이 기법은 이미지 톤에 따라 점의 밀도를 조절하는 방식이며, 자동으로 점묘화를 생성하여 사용이 편리해졌다. 하지만 이미지 톤에 의존하여 단순히 명암을 표현하는데 머물렀다[그림7].



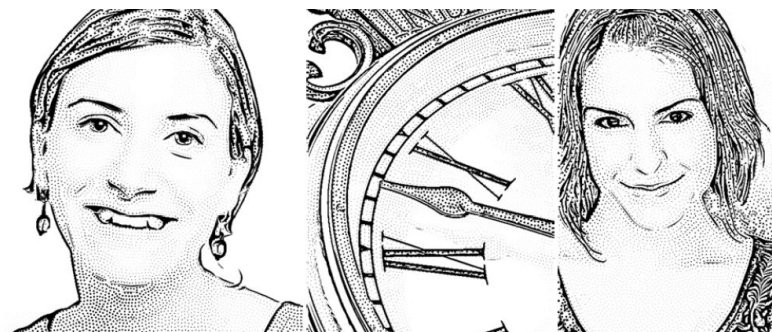
[그림7] 가중치 보로노이 다이어그램을 이용한 점묘화 생성 기법[8]

이 후 특징 표현을 강화하려는 다양한 연구가 제안되었다. Mould[9]등은 그래프를 이용하여 점을 배치하는 알고리즘을 제안했다. 특히, 영상의 특징을 강조하기 위해 점들이 에지(edge)를 따르도록 가중치를 주었다. 이미지 톤을 가중치로 부여한 보로노이 기반의 방법[8]과 달리 에지 부분이 강조함으로써 특징 표현을 개선하였다[그림8].



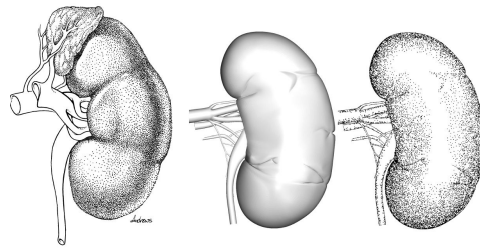
[그림8] 그래프를 이용한 점묘화 생성 기법[9]

Kim[2]등은 헤드컷 일러스트 작품을 모티브로 특징의 흐름을 점의 배치에 적용한 점묘화 기법을 제안했다. 전체적으로 점들을 균일하게 분포시키고, 특징의 방향에 따라 위치하는 점과 라인을 이용하여, 헤드컷 일러스트 작품과 비슷한 결과를 얻었다[그림9].



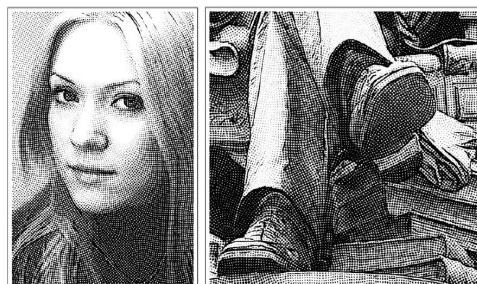
[그림9] 특징 흐름을 이용한 헤드컷 스타일 생성 기법[2]

Kim[11]등은 개인 작가의 점묘화 작품을 참고로 예제 기반의 점묘화 기법을 제안했다. 화가가 그린 점묘화의 톤 맵(tone map)을 추출하고, 점의 분포 방식을 분석하여, 텍스처 합성 알고리즘을 통해 생성된 텍스처를 기반으로 점을 표현한다. 이 방법은 입력 조건이 자유롭지만, 손으로 그린 점묘화에서 텍스처를 추출하는 것이 항상 가능하지는 않다는 한계점이 있다[그림10].



[그림10] 예제 기반의 스티플링 기법[11]

Son[10]등은 스트럭처 그리드(structure grid)를 사용하여 방향성 있는 점의 배치(placement)와 해칭(hatching) 기법을 제안했다. 이 방법은 특징 벡터 필드로부터 특징 방향에 따라 부드럽게 변형된 스트럭처 그리드를 생성하고, 그리드의 교차점에 점을 렌더링 한다. 이전의 헤드컷 스타일 점묘화 기법 연구[2]에서 보다 헤드컷 스타일의 모든 중요한 방향으로 점들을 엄격하게 맞춘다는 점에서 실제 헤드컷 일러스트 작품과 더욱 유사한 결과를 보여주었다[그림11].



[그림11] 스트럭처 그리드를 이용한 헤드컷 스타일 생성 기법[10]

## 2. 동영상 점묘화 생성 기법 연구

Houit[1]은 동영상에 특화된 컬러 점묘화 연구를 제안했다. 점을 배치하는 단계에서 로이드 알고리즘[12]을 개작하여 이미지 밀도를 근본적으로 점의 분포에 반영했다. 로이드 알고리즘은 무게중심 보로노이 다이어그램(Centroid Voronoi Diagram)을 만들기 위한 알고리즘 중 하나로, 점묘화 기법 연구에서 많이 사용되었다. 이 연구에서는 기본적으로 이미지의 밝기에 비례한 확률을 이용하여 점을 분포시키고, 날카로운 부분을 표현하기 위해서 에지부분에 추가로 점이나 패턴을 렌더링 하는 방법을 사용했다[그림12].



[그림12] 동영상에 특화된 컬러 점묘화 연구[1]

### Ⅲ. 문제점 분석 및 연구 방법

점묘화 연구는 단일 이미지 기반뿐만 아니라 동영상 기반의 연구로 적용 대상이 확대되었다. 하지만 동영상 기반의 점묘화 기법 연구는 단일 영상 기반 연구에 비해 활발히 이루어지지 못했다. 뿐만 아니라 그동안 제안된 점묘화 동영상 생성 기법 연구[1]는 근본적으로 이미지의 톤에 따라 점의 밀도를 조절함으로써 대상을 표현하기 때문에 움직이는 물체의 특징을 보존하는 데 한계가 있다.

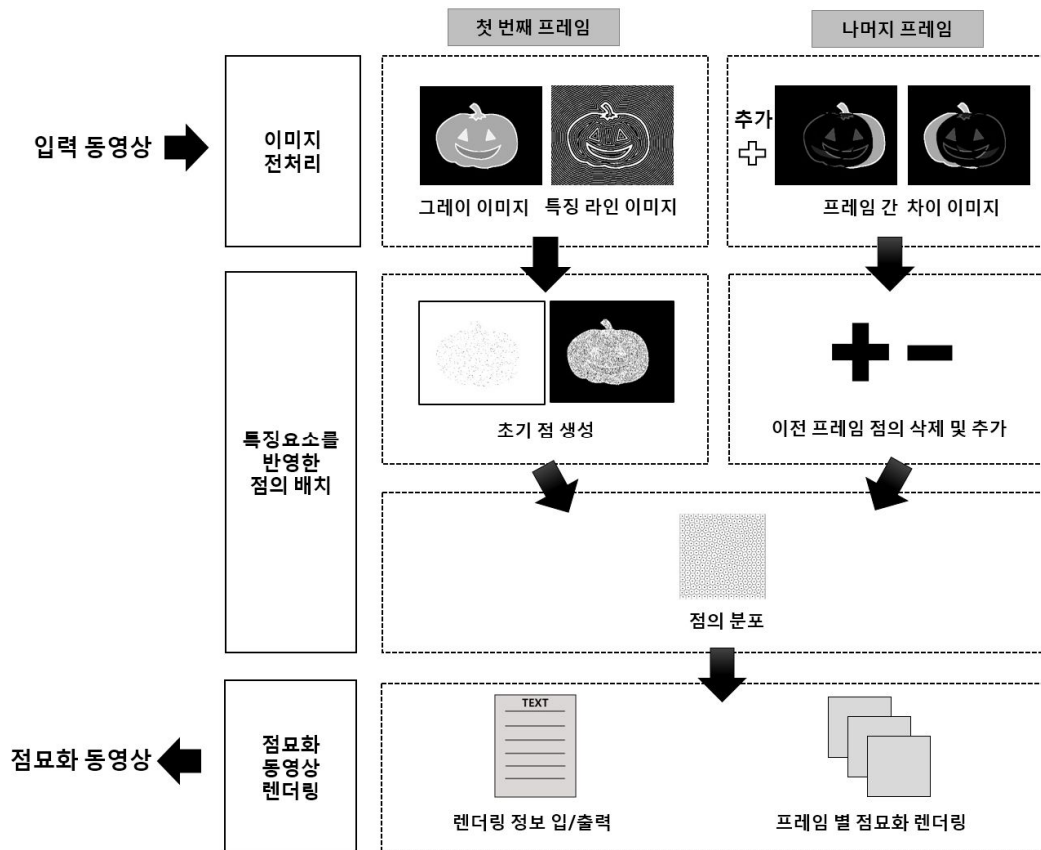
본 논문에서 제안하는 방법은 밝기, 예지, 특징라인과 같은 특징 요소를 점의 배치에 반영하여, 특징 표현이 개선된 점묘화 동영상을 결과로 출력하는 것이다. 다양한 관점에서 제안된 단일 이미지 기반 연구 중 헤드컷 일러스트 스타일의 점묘화 기법 연구[2]에서 착안하여, 특징 라인 이미지를 생성하고 점의 배치에 영향을 주고자 한다. 동영상 기반의 NPR 연구의 가장 중요한 이슈는 프레임 간의 시간적 일관성(time coherence)을 유지하는 것이다. 프레임 간의 일관성을 고려하기 위해, 첫 번째 프레임에 대해서만 단일 이미지 점묘화 기법을 적용하고 두 번째 이후 프레임에 대해서는 이전 프레임에 배치된 점을 수정하는 방식[1]을 사용한다. 다만 본 연구에서는 특징 요소를 반영하기 때문에, 원본 동영상의 중요한 특징이 훨씬 더 두드러진 점묘화 동영상 결과를 얻는다. 또한 본 연구에서는 자동으로 점묘화 동영상을 생성해주는 시스템 구현을 통해, 간단한 파라미터(parameter)를 입력만으로 손쉽게 점묘화 동영상을 생성할 수 있도록 한다.

## IV. 점묘화 동영상 생성 시스템

### 1. 시스템 개요

본 논문에서 제안하는 점묘화 동영상 생성 시스템은 크게 세 가지 단계로 구성된다. 첫 번째는 이미지 전처리 단계이고, 두 번째는 특징 요소를 반영한 점의 배치 단계이며, 세 번째는 점묘화 동영상 렌더링 단계이다.

이미지 전처리 단계에서는 먼저 입력동영상으로 부터 이미지 프레임을 추출한다. 각 이미지 프레임으로부터 영상의 밝기와 에지를 나타낸 그레이 이미지, 영상의 특징 흐름이 표현된 특징 라인 이미지, 현재 프레임과 이전 프레임 간의 그레이 이미지의 밝기 차이 값을 나타낸 프레임 간 차이 이미지를 생성한다. 특징 요소를 반영한 점의 배치 단계에서는 이미지 밀도에 맞게 제작된 로이드 알고리즘 방법[1]을 특징 라인에 영향을 받도록 변형하여 사용한다. 동영상의 시간적 일관성을 유지하기 위해 첫 번째 프레임과 나머지 프레임의 초기 점 생성 방식을 다르게 한다. 첫 번째 프레임의 초기 점은 그레이 이미지의 밝기 값에 비례한 확률로 생성한다. 나머지 프레임에 대해서는 이전 프레임의 점을 수정하는 과정을 통해 초기 점을 설정한다. 초기 점을 배치한 후에는 특징라인에 영향을 받아 고르게 분포시키는 과정을 거쳐 점의 최종 위치가 결정된다. 점묘화 동영상 렌더링 단계에서는 실제 점묘화 이미지를 생성한다. 각 프레임 별 점의 위치가 최종 결정되면 점의 좌표, 색상과 같은 정보를 텍스트 파일에 입력한다. 그리고 점의 정보를 각 프레임의 흰색 캔버스 위에 렌더링 한다. 마지막으로 모든 프레임을 동영상으로 합성하여 점묘화 동영상을 생성한다. [그림13]는 점묘화 동영상 생성 시스템 개요를 나타낸 것이다.



[그림13] 점묘화 동영상 생성 시스템 개요

## 2. 이미지 전처리

이미지 전처리는 영상의 특징 요소가 표현된 그레이 이미지와 특징 라인 이미지를 생성하며, 동영상의 시간적 일관성 유지를 위한 프레임 간 차이 이미지를 생성하는 단계이다. 먼저 입력 동영상에서 이미지 프레임을 추출하여, JPG 파일 포맷으로 저장한다. 특징 요소의 구성은 [표1]과 같다.

특징 요소	밝기	에지	특징 라인
의미	1채널 그레이 스케일 이미지의 강도에 따라 0~255 값으로 표현	영상의 날카로운 부분을 표현	에지 라인에 따라 선을 이루는 특징의 흐름을 표현
사용	점의 초기 집합 생성 시 점의 밀도 조절		점의 분포 단계에서 점의 위치에 영향
표현 수단	그레이 이미지		특징 라인 이미지

[표1] 특징 요소 구성

### 2.1 그레이 이미지 생성

그레이 이미지는 밝기 이미지와 에지 이미지를 합성한 이미지로, 점의 초기 집합 생성 시 점의 밀도를 결정하기 위해 참고로 사용된다. 밝기 이미지는 3개의 컬러 채널(R, G, B)을 가진 입력 이미지를 변환한 1채널 이미지이다. 밝기 이미지의 강도(intensity)는 R, G, B 값의 평균을 구하고 255로부터 뺀 값이다. 그 식은 (1)과 같다.

$$\text{Intensity}(x, y) = 255 - (R(x, y) + G(x, y) + B(x, y)) / 3 \quad (1)$$

에지 이미지를 생성하기 위해서는 먼저 입력 이미지를 그레이 스케일로 변환한다. 다음은 변환된 그레이 스케일 이미지로부터 에지를 검출하는 단

계이다. 에지는 주로 영상의 밝기가 급격하게 변하는 지점에서 존재한다. 에지 이미지는 캐니 에지 알고리즘[13]을 사용하여 생성한다. 이 알고리즘은 잡음에 민감하지 않기 때문에 에지 검출 시 많이 사용하는 방법이다. 캐니 알고리즘은 잡음 제거를 위한 스무딩(smoothing), 그래디언트(gradient) 계산, 비 최댓값 제거(non-maximum suppression), 상단 임계값과 하단 임계값을 이용해 에지를 판단하는 4가지 단계로 구성된다. 생성된 캐니 이미지의 에지부분은 흰색, 나머지 부분은 검정색으로 결과가 나타난다. 그레이 이미지는 밝기 이미지에서 에지에 해당하는 픽셀을 흰색으로 변환하는 과정을 통해 생성된다. [그림14]은 그레이 이미지의 생성 과정을 나타낸 것이고, [알고리즘1]은 다음과 같다.



[그림14] 그레이 이미지 생성 과정

```

[알고리즘1] 그레이 이미지 생성
for x = 0 to image width-1 do
  for y = 0 to image height-1 do
    Intensity(x, y) = 255 - (R(x, y)*G(x, y)*B(x, y)) / 3 ∈[0;255]
    if P(x, y) == edge pixel then
      P(x, y) = white color
    else
      P(x, y) = Intensity(x, y)
    end if
  end for
end for
end for

```

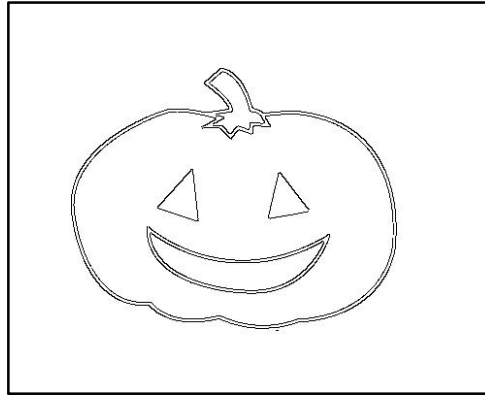
## 2.2 거리 변환을 이용한 특징 라인 이미지 생성

특징 라인 이미지는 영상의 특징 흐름에 따라 선을 이루는 흑백 이미지이다. 특징흐름을 추출하는 방법[2]를 참고하여 특징 라인 이미지를 생성한다. [그림15]는 특징 라인 이미지를 생성하는 단계를 나타낸다.



[그림15] 특징 라인 이미지 생성 단계

에지 추출 단계에서는 그레이 이미지 생성 단계에서 생성한 에지 이미지를 반전(reverse)하여 에지부분의 픽셀이 0값을 갖도록 변환한다[그림16].



[그림16] 반전된 에지 이미지

다음 단계는 에지 이미지의 거리 변환(distance transform)[14]이다. 거리 변환은 모든 픽셀에 대해서 픽셀 값이 0인 지점과의 최단거리를 계산하는 과정이다. 거리 변환을 통해 얻은 거리 값은 특징 라인을 추출하는 데 사용된다. 거리변환 이미지의 픽셀은 에지로부터 가까울수록 0에 가까운 값을 갖는다. 본 연구에서는 거리 변환을 수행하기 위해, OpenCV라이브러리[15]에서 제공하는 함수인 `cvDistTransform()`을 사용하여 [그림17]과 같은 거리 변환 이미지를 생성하였다.



[그림17] 거리 변환 이미지

마지막으로 특징 라인 추출 단계에서는 거리 변환된 이미지의 거리 값을 일정 간격으로 샘플링 하는 방법[2]을 사용하여 특징 라인 이미지를 생성하며, 검정색과 흰색 라인으로 구분하여 영상의 특징 흐름을 표현한다. 샘플링 간격을  $s$ , 흰색 라인의 두께를  $w$ , 검정색 라인의 두께를  $b$ 라 할 때,  $s = w + b$ 이며 거리 값을  $s$ 로 나눈 나머지 값이  $w$ 보다 작을 경우 흰색, 그렇지 않으면 검정색으로 칠해준다[그림18] [알고리즘2].



[그림18] 특징 라인 이미지

[알고리즘2] 특징 라인 추출

```

for int x = 0 to image width-1 do
  for int y = 0 to image height-1 do
    remainder = Distance(x, y) % s
    if ( remainder < w) then
      P(x, y) = White Color
    else
      P(x, y) = Black Color
    end if
  end for
end for

```

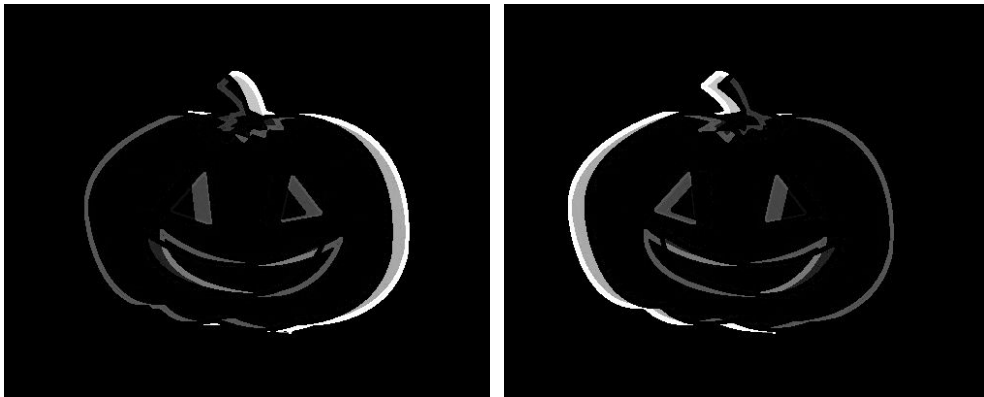
### 2.3 프레임 간 차이 이미지 생성

점의 배치과정에서 이전 프레임의 점을 유지할 것인지 아닌지 결정하기 위해, [1]의 방법을 사용하여 프레임 간 차이 이미지를 생성한다. 다만 본 연구에서는 밝기와 에지를 동시에 나타낸 그레이 이미지를 사용하여 프레임 간 차이 이미지를 생성한다. 점의 추가와 삭제를 위해, 두 개의 프레임 차이 이미지를 생성한다[알고리즘3].

#### [알고리즘3] 프레임 차이 이미지 생성

```
for  $i = 0$  to image width - 1 do
  for  $j = 0$  to image height-1 do
     $D_{add}(i, j) = \text{Current image color}(i, j) - \text{Previous image color}(i, j)$ 
     $D_{remove}(i, j) = \text{Previous image color}(i, j) - \text{Current image color}(i, j)$ 
    if  $D_{add}(i, j) \leq 0$  then
       $P_{add}(i, j) = \text{Black Color}$ 
    else
       $P_{add}(i, j) = D_{add}(i, j)$ 
    end if
    if  $D_{remove}(i, j) \leq 0$  then
       $P_{remove}(i, j) = \text{Black Color}$ 
    else
       $P_{remove}(i, j) = D_{remove}(i, j)$ 
    end if
  end for
end for
```

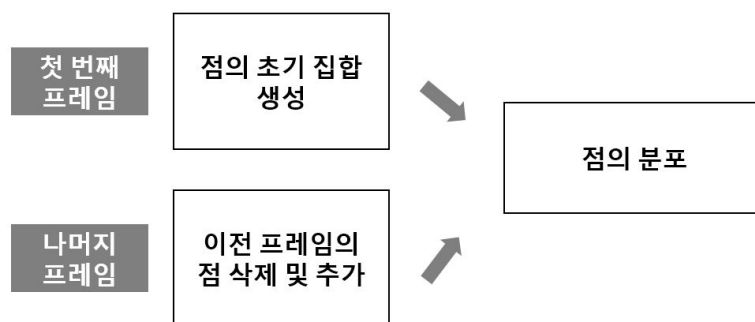
[알고리즘3]에서  $P_{add}$ 는 점의 추가를 위한 차이 이미지의 픽셀 값이며,  $P_{remove}$ 는 점의 삭제를 위한 차이 이미지의 픽셀 값이다.  $D_{add}$ 는 현재 그레이 이미지의 픽셀 값에서 이전 그레이 이미지의 픽셀 값을 뺀 값,  $D_{remove}$ 는 그 반대로 뺀 값이다.  $P_{add}$ 은  $D_{add}$ 값이 속하는 범위 조건에 따라 결정한다.  $D_{add}$ 가 0보다 작거나 같으면, 이전 프레임과 밝기가 동일하거나 감소한 부분이므로 점을 추가하지 않으므로 검정색으로 칠한다. 그렇지 않은 경우 밝기가 증가한 영역이므로  $P_{add}$ 의 픽셀 값을  $D_{add}$ 로 정한다.  $P_{remove}$ 도  $P_{add}$ 와 동일한 방식으로 구한다. 그 결과 [그림19]과 같이, 프레임 간 차이 이미지가 생성된다.



[그림19] 점의 추가를 위한 이미지(좌)와 삭제를 위한 이미지(우)

### 3. 특징 요소를 반영한 점의 배치

본 장에서는 점의 초기 집합 생성 방법, 점의 분포 방법, 점의 삭제 및 추가 방법에 대해 다룬다. 특징 요소를 반영하여 점을 배치하기 위해서 본 연구는 Houit[1]이 제안한 알고리즘을 변형하여 사용한다. 이 알고리즘은 근본적으로 이미지 밀도에 맞춘 무게중심 보로노이 다이어그램(Centroid Voronoi Diagram)을 만들기 위한 방법이다. 무게중심 보로노이 다이어그램을 생성하기 위해서는 두 개의 점의 집합을 사용한다. 하나는 보로노이 다이어그램의 영역을 이루는 픽셀에 해당하는 점의 집합  $V$ 이고, 나머지 하나는 실제 점묘화 렌더링에 사용하는 Site 점의 집합  $S$ 이다. 동영상의 시간적 일관성을 고려하기 위해 첫 번째 프레임에 대해서만 점의 초기 집합을 생성한다. 그레이 이미지의 밝기 값에 따른 확률로  $V$ 와  $S$ 을 생성한다. 다음으로 두 개의 집합을 연관시키고, 무게중심으로 점을 이동시키는 과정을 반복적으로 수행하여 점을 분포시킨다. 본 연구에서는 특징 요소를 반영하도록 알고리즘을 변형했다. 결과적으로 점의 밀도는 에지(edge)와 강도(Intensity)에 의해 결정되고, 점의 위치는 특징 라인에 영향을 받도록 한다. 첫 번째 프레임을 제외한 나머지 프레임에 대해서는 이전 프레임의 점을 삭제하고 추가하는 과정을 통해 점의 집합을 구성한다[그림20].



[그림20] 점의 배치 단계

### 3.1 점의 초기 집합 생성

Site 점의 집합 S의 원소 개수는 사용자 입력에 따라 다양하게 조절한다. 보로노이 영역에 해당하는 점의 집합 V의 점의 개수는 S의 개수\*100으로 정했다. S와 V의 기본적인 점 생성 방식은 동일하다. 이미지 전처리 단계에서 구한 그레이 이미지에서 랜덤으로 추출한 픽셀의 값이 [0:255] 사이의 랜덤 값보다 클 경우 집합의 원소로 추가한다. 즉, 이미지의 밝기 값이 큰 픽셀 일수록 점이 집합에 속할 확률이 높아진다. 단 Site 점의 집합을 생성하는 과정에서는 같은 위치에 점이 위치하는 것을 방지하기 위해, 이미 생성된 점의 좌표가 표시된 이미지를 참조한다[알고리즘4]. 점의 초기 집합 구성은 다음 [표2]와 같다.

	Site 점의 집합 (S)	Voronoi 점의 집합(V)
점의 개수	사용자 지정	S의 원소 개수*100
중복 여부	중복 불가	중복 허용
생성 방법	그레이 이미지의 밝기에 비례한 확률로 무작위하게 생성	

[표2] 점의 초기 집합 구성

각 Site 점과 Voronoi 점을 구성하는 정보는 [표3]와 같다.

Site 점의 정보	Voronoi 점의 정보
① 고유 id ② 점의 위치 좌표 (x, y) ③ 무게 중심 좌표 (cx, cy) ④ 연관된 Voronoi 점 x값의 합 ⑤ 연관된 Voronoi 점 y값의 합 ⑥ 연관된 Voronoi 점 개수 ⑦ 수렴 조건 충족 여부(T/F)	① 연관된 Site 점의 id ② 점의 좌표 (x, y)

[표3] Site 점(좌)과 Voronoi 점의 정보(우)

Site 점은 식별 가능한 고유id와 이미지의 x, y축에 해당하는 위치 좌표를 갖는다. 점을 분포시키는 과정에서 필요한 무게중심 계산을 위해, 무게 중심 위치 정보와 연관된 Voronoi 점의 x, y 좌표 값의 합, 연관된 보로노이 점 개수 정보를 가진다. 또한 Site 점이 수렴 조건에 충족하는지 확인하기 위해, 수렴조건 충족 여부에 대한 정보를 포함한다. Voronoi 점은 점의 좌표와 연관된 Site id 정보를 가진다.

[알고리즘4] 점의 초기 집합 생성

Voronoi 점 집합 V 생성:

```
for  $i = 1$  to V의 개수 do
   $rx =$  랜덤 정수  $\in [0; \text{width}-1]$ 
   $ry =$  랜덤 정수  $\in [0; \text{height}-1]$ 
   $r =$  랜덤 정수  $\in [0; 255]$ 
  Intensity( $rx, ry$ ) = 그레이 이미지의 밝기 값  $\in [0; 255]$ 
  if  $r > (255 - \text{Intensity}(rx, ry))$  then
    V집합에 point( $rx, ry$ )를 추가
     $i \leftarrow i+1$ 
  end if
end for
```

Site 점 집합 S 생성:

```
for  $j = 1$  to S의 개수 do
   $rx =$  랜덤 정수  $\in [0; \text{width}-1]$ 
   $ry =$  랜덤 정수  $\in [0; \text{height}-1]$ 
   $r =$  랜덤 정수  $\in [0; 255]$ 
  intensity( $rx, ry$ ) = 그레이 이미지의 밝기 값  $\in [0; 255]$ 
  if point( $rx, ry$ )이 중복이 아닌 경우 then
    if  $r > (255 - \text{Intensity}(rx, ry))$  then
      S집합에 point( $rx, ry$ )를 추가
       $j \leftarrow j+1$ 
    end if
  end if
end for
```

### 3.2 점의 분포

이 절에서는 에지(edge)와 강도(Intensity)에 의해 결정된 초기 점들이 특징 라인에 영향을 받아 고르게 분포하는 과정에 대해 설명한다.

먼저 Voronoi 점에 대해서 Site 점들을 순회하면서 가장 가까운 Site 점을 찾는다. 두 점사이의 거리  $d$ 를 계산하기 위해서는 유클리드 거리 계산식 (2)을 사용한다. Voronoi 점은 가장 가까운 거리에 위치한 Site 점의 고유id를 가짐으로써 Site 점과 연관(association)된다. 이 때, 연관된 Voronoi 점의 위치가 특징 라인 이미지의 흰색 영역에 해당할 경우만 Site 점의 무게중심 계산 정보에 추가한다. 결과적으로 점들이 흰색 라인에 영향을 받아 분포하도록 한다.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

다음은 각 Site 점의 무게 중심 계산 정보를 이용하여 무게중심을 계산하고, 무게중심으로 Site 점을 이동시키는 단계이다. 이 때 무게중심과 Site의 이전 위치를 비교하여, 수렴 조건에 충족하는지 확인한다. Site 점이 수렴조건에 만족할 때까지 점들을 연관시키고 무게중심으로 이동하는 과정을 반복한다[알고리즘5].

**[알고리즘5] 점의 분포**

**① 최단 거리 계산 및 연관(association)**

*min* :최단거리 비교 값( 초기 값: INT\_MAX)

*id* :Site 점의 고유 번호

feature color :특징 라인 이미지의 픽셀 색상

```
for i = 1 to Voronoi 점의 개수 do
  for j = 1 to Site 점의 개수 do
     $V_i$ 와  $S_j$ 의 최단 거리 d를 계산
    if  $d < min$  then
       $min \leftarrow d$ 
       $id = j$ 
    end if
     $j \leftarrow j+1$ 
  end for
   $V_i$ 와 연관된 Site  $id = id$ 
  if feature color( $V_i$ ) == White color then
    Site 점의 무게 중심 계산 정보에  $V_i$ 를 추가
  end if
   $i \leftarrow i+1$ 
end for
```

② 무게 중심 계산 및 이동

*convergence* : 모든 Site 점이 무게중심 범위에 해당하는지 판단(T/F)

**for**  $j = 1$  to Site 점의 개수 **do**

$S_j$ 의 무게중심 좌표  $C_j$ 을 계산

$S_j$ 의 좌표를  $C_j$ 로 이동

**if**  $S_j$ 의 좌표가 중복이 아님 **and** 연관된 Voronoi 점의 개수  $\geq 1$

**then**

**if**  $S_j$ 의 좌표가 무게중심 범위에 해당하지 않음 **then**

*convergence* = *false*

**end if**

$S_j$ 의 무게중심 계산 정보를 리셋(reset)

**else**

$S_j$ 을 삭제

**end if**

$j \leftarrow j+1$

**end for**

③ 수렴 기준 충족 여부 확인

**while** *convergence* = *false* **and** 반복 횟수 < threshold **do**

①, ② 수행

**end while**

### 3.3 점의 삭제 및 추가 기법

이미지의 두 번째 프레임부터는 동영상의 시간적 일관성 고려를 위해, 새로 점의 초기 집합을 생성하지 않는다. 대신 앞서 이미지 전처리 단계에서 구한 두 개의 프레임 간의 차이 이미지를 이용하여 이전 프레임의 Site 점과 Voronoi 점을 삭제 및 추가하는 과정을 통해, 현재 프레임의 초기 점을 구성한다.

#### 3.3.1 점의 개수

두 번째 이후 프레임의 점의 개수를 구하기 위해, 기존 점묘화 동영상 연구[1]의 이미지의 밀도와 점의 개수를 같은 비율로 유지하는 방법을 사용한다. 본 연구에서는 첫 번째 프레임 그레이 이미지의 전체 밀도(global density)와 점의 개수, 그리고 현재 프레임의 밀도(current density)를 이용하여 점의 개수를 결정한다. 이미지 밀도를 구하는 식은 (3)과 같다.

$$density = \frac{\sum_{i=1}^{width*height} Gray(i)}{255*width*height} \quad (3)$$

$i$ 번째 프레임의 점의 개수  $N_i$ 을 구하는 식은 (4)과 같다. 예를 들어, 첫 번째 프레임의 점의 개수가 100개이고, 밀도가 0.5라 할 때, 두 번째 프레임의 밀도가 0.25라면 점의 개수는 50개로 결정된다.

$$N_i = (N_1 * current\ density) / global\ density \quad (4)$$

### 3.3.2 점의 삭제

Site 점과 Voronoi 점은 이전 프레임의 모든 Site점과 Voronoi 점을 순회하면서, 삭제할 것인지 결정한다. 밝기 값을 [0:100]의 범위로 변환한 후, 0에서 99사이의 랜덤 정수를 생성하여 점의 밝기 값과 비교하여 점의 삭제 여부를 결정한다. 프레임 간 차이 이미지의 밝기 차이 값이 클수록 점이 삭제될 확률이 높다. 즉, 이전 프레임보다 밝기가 어두워진 영역에 해당하거나 더 이상 에지에 해당하지 않을 경우 점을 삭제하기 위함이다. Site 점과 Voronoi 점의 삭제를 위한 과정은 각각 [알고리즘6]과 [알고리즘7]에 기술하였다.

#### [알고리즘6] Site 점의 삭제

```
for  $i = 1$  to Site 점의 개수 do
   $x = i$ 의  $x$ 좌표  $\in [0; \text{width}-1]$ 
   $y = i$ 의  $y$ 좌표  $\in [0; \text{width}-1]$ 
   $r =$  랜덤 정수  $\in [0; 99]$ 
   $\text{diff}(x, y) =$  삭제를 위한 차이 이미지 값
   $\text{probability} =$  삭제될 확률 값  $\in [0; 100]$ 
   $\text{probability} \leftarrow \text{diff}(x, y) * 100 / 255$ 
  if ( $\text{probability} > r$ ) then
    Site point( $x, y$ )를 삭제
  end if
   $i \leftarrow i+1$ 
end for
```

### [알고리즘7] Voronoi 점의 삭제

```
for  $j = 1$  to Voronoi 점의 개수 do
   $x = j$ 의  $x$ 좌표[0;width-1]
   $y = j$ 의  $y$ 좌표 [0;width-1]
   $r =$  랜덤 정수[0;99]
   $\text{diff}(x, y) =$  삭제를 위한 차이 이미지 값
   $\text{probability} =$  삭제될 확률 값 [0;100]
   $\text{probability} \leftarrow \text{diff} * 100 / 255$ 
  if ( $\text{probability} > r$ ) then
    Voronoi point( $x, y$ ) 삭제
  end if
   $j \leftarrow j + 1$ 
end for
```

### 3.3.3 점의 추가

삭제와 마찬가지로 프레임 간 차이 이미지의 밝기에 비례한 확률로 Site와 Voronoi 점을 추가한다. 각각 점을 삭제 한 뒤 점의 개수가 현재 프레임의 점의 개수에 도달할 때까지 반복한다. 먼저 추가를 위한 프레임 간 차이 이미지에서 임의의 위치의 차이 값을 추출한다. 차이 값을 0~100의 범위로 변환한 확률 값(probability)과 0~99의 범위의 랜덤 정수와 비교하여, 해당 위치의 점을 추가할지 결정한다. Site 점의 경우 위치 중복을 방지하기 위해, 중복 확인 조건을 추가한다[알고리즘8] [알고리즘9].

**[알고리즘8] Site 점의 추가**

```
for  $i$  = 삭제 후 Site 점 개수 to 현재 프레임의 Site 점 개수 do  
   $x$  = 랜덤 정수  $\in [0; \text{width}-1]$   
   $y$  = 랜덤 정수  $\in [0; \text{height}-1]$   
   $r$  = 랜덤 정수  $\in [0; 99]$   
   $\text{diff}(x, y)$  = 추가를 위한 차이 이미지 값  
   $\text{probability}$  = 추가될 확률 값  $\in [0; 100]$   
   $\text{probability} \leftarrow \text{diff} * 100 / 255$   
  if  $\text{probability} > r$  and point ( $x, y$ )가 중복이 아님 then  
    point ( $x, y$ )를 S에 추가  
     $i \leftarrow i+1$   
  end if  
end for
```

**[알고리즘9] Voronoi 점의 추가**

```
for  $j$  = 삭제 후 Voronoi 점 개수 to 현재 프레임의 Voronoi 점 개수 do  
   $x$  = 랜덤 정수  $\in [0; \text{width}-1]$   
   $y$  = 랜덤 정수  $\in [0; \text{height}-1]$   
   $r$  = 랜덤 정수  $\in [0; 99]$   
   $\text{diff}(x, y)$  = 추가를 위한 차이 이미지 값  
   $\text{probability}$  = 추가될 확률 값  $\in [0; 100]$   
   $\text{probability} \leftarrow \text{diff} * 100 / 255$   
  
  if  $\text{probability} > r$  then  
    point  $(x, y)$ 를  $V$ 에 추가  
     $j \leftarrow j+1$   
  end if  
end for
```

#### 4. 점묘화 동영상 렌더링

점묘화 동영상 렌더링은 동영상 파일로 합성하는 과정이다. 프레임별로 점묘화를 렌더링하기 위해서 각 프레임의 점의 배치 단계가 끝날 때마다 최종 결정된 Site 점의 정보를 텍스트 파일에 기록한다. 기록하는 정보는 다음 [표4]와 같다.

프레임 별 Site 점의 정보
① 프레임 번호
② 프레임별 점의 개수
③ 프레임 별 고유 번호
④ x, y좌표
⑤ 원본 색상의 컬러 R, G, B 값
⑥ 그레이 밝기 값

[표4] 프레임 별 Site 점의 정보를 기록하는 텍스트 파일 구조

모든 프레임의 점의 배치가 끝난 후, 기록된 Site 점의 정보를 이용하여 점을 렌더링 한다. 점의 색상은 원본 이미지의 색상을 사용한다. 프레임 별로 이미지 프레임은 JPG확장자 형태로 저장한다. 저장된 이미지 프레임을 동영상 파일로 합성하여 최종 점묘화 동영상을 생성한다.

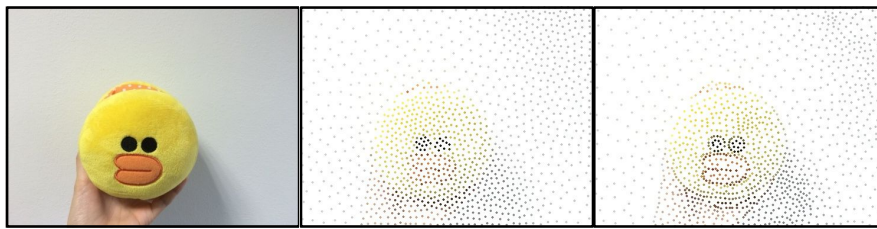
## V. 실험 및 결과

### 1. 단일 이미지 점묘화 렌더링 결과

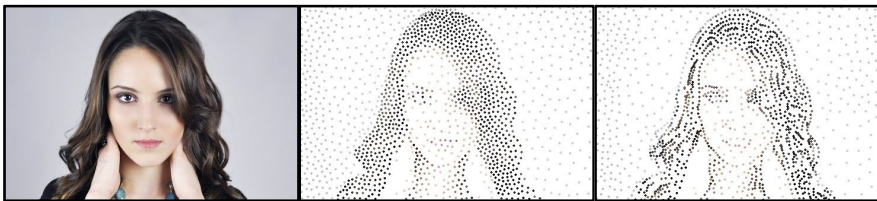
본 연구에서는 점묘화 동영상 렌더링 실험에 앞서, 단일 이미지를 대상으로 실험을 진행하였다. [표5]와 같이, 입력 이미지와 점의 개수를 동일하게 설정한 후, [1]의 개작된 로이드 알고리즘을 사용하여 점을 분포한 방법과 본 연구에서 제안하는 방법으로 결과를 생성했다. [그림21]는 다양한 종류의 단일 이미지를 입력으로 생성한 점묘화 이미지들을 나타낸다. [그림21]의 (a)는 캐릭터 사진을 입력으로 한 결과이다. 캐릭터의 라인, 눈, 입 등 중요한 특징이 잘 나타나는 것을 확인 할 수 있다. [그림21]의 (b)는 인물 사진을 입력으로 한 결과이다. 인물 사진은 단순한 캐릭터 사진보다 더 복잡한 형태이다. 인물 사진에서도 머리카락, 눈썹 등과 같은 입력 영상의 특징이 잘 보존된 점묘화 이미지를 생성할 수 있다. [그림21]의 (c)는 자연의 꽃 이미지를 입력으로 한 결과이다. 입력 이미지의 잎사귀 부분의 잎맥과 같은 특징 부분이 잘 표현되었음을 확인 할 수 있다. 결과적으로 기존 방법[1]과 비교했을 때, 입력 이미지의 대상의 특징이 두드러지는 점묘화 이미지 결과를 나타낸다.

	종류	초기 점 개수
그림 (a)	캐릭터	1200개
그림 (b)	인물	1500개
그림 (c)	자연	1200개

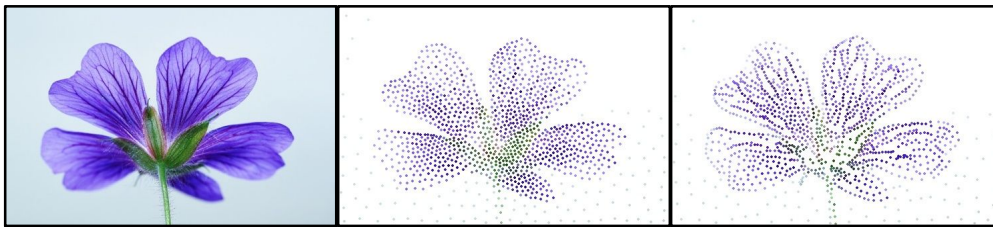
[표5] 그림21의 점묘화 이미지 파라미터



(a)



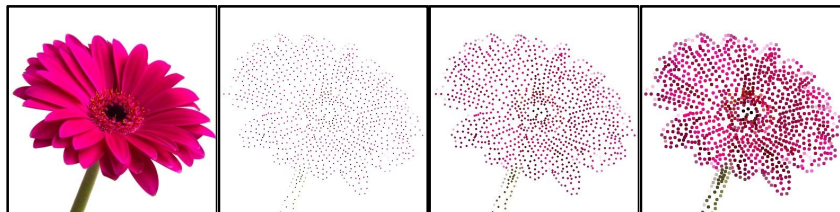
(b)



(c)

[그림21] (왼쪽부터) 입력 이미지, [1]의 방법 결과, 본 연구 방법 결과

추가적인 실험으로, 동일한 개수의 점을 세 가지 크기로 구분하여 점묘화 이미지를 생성하였다[그림22]. 사용자의 기호에 따라 점의 크기를 다양하게 조절하여 결과를 만들 수 있다.



[그림22] 점의 크기에 따른 결과

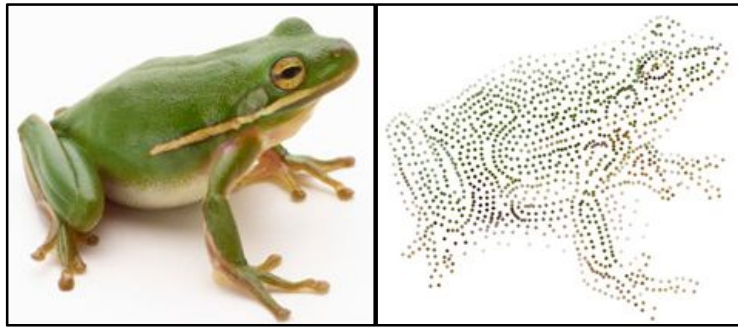
초기 Site 점의 개수는 사용자가 지정할 수 있다(4장 3.1절 참고). 하지만 점의 개수가 너무 적을 경우에 특징을 표현하기 충분하지 않을 수 있다. 반대로 점의 개수가 너무 많을 경우에는 특정 영역에 점이 밀집할 가능성이 높기 때문에 원치 않는 결과가 나타날 수 있다. 따라서 본 연구에서는 적절한 초기 Site 점의 개수를 결정하기 위해, 그레이 이미지의 전체 밀도와 입력 이미지의 전체 픽셀 수를 이용해 계산식 (5)을 만들었다.

$$\text{initial site number} = N_{input} * density * C \quad (5)$$

$N_{input}$ 는 입력이미지의 픽셀 수이며,  $density$ 는 4장의 3.3.1절에서 구한 그레이 이미지의 전체 밀도이다.  $C$ 는 초기 점의 개수를 결정하기 위한 상수이다. 다양한 크기의 입력 동영상을 사용하여 실험한 결과를 바탕으로,  $C$ 의 값을 0.03으로 설정하여 초기 Site 점의 개수를 결정하였다[표6]. [그림23]는 자동으로 초기 Site 점의 개수를 결정하여 생성한 점묘화 이미지이다.

그림23	입력 이미지 픽셀 수	전체 밀도	초기 Site점의 개수
(a)	139600	0.293870	1230
(b)	166000	0.268799	1338
(c)	240000	0.206804	1488

[표6] 그림23의 점묘화 이미지 파라미터



(a)



(b)



(c)

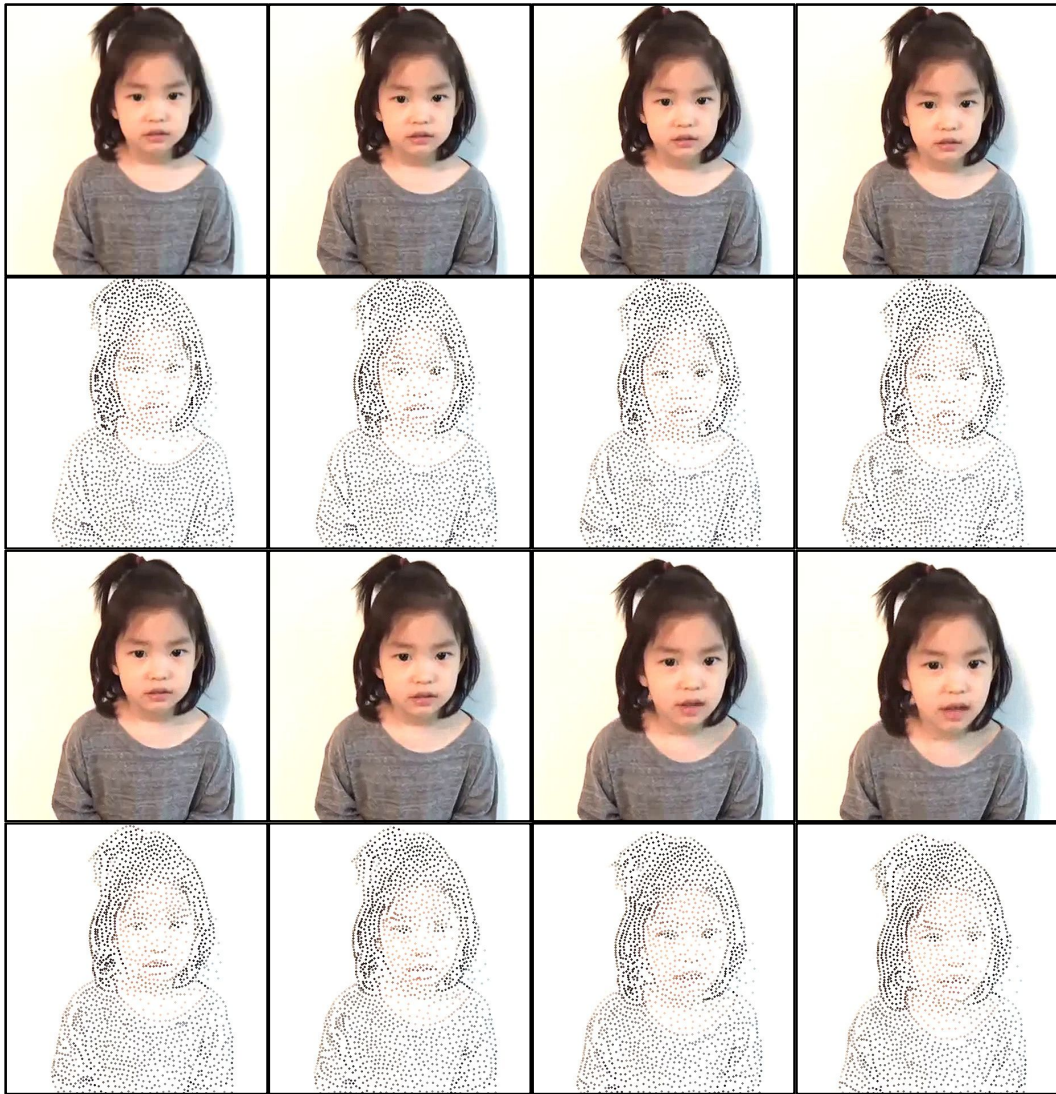
[그림23] 자동으로 초기 Site 점 개수를 설정한 점묘화 결과

## 2. 점묘화 동영상 렌더링 결과

본 절에서는 동영상을 입력으로 하여, 본 연구에서 제안하는 시스템을 사용하여 생성한 점묘화 동영상 결과를 보여준다. 실험을 위해 웹에서 다양한 동영상을 수집하였다. [표7]은 그림 별로 초기 점의 개수와 프레임 개수를 나타낸다. [그림24]는 인물이 노래하는 동영상을 입력으로 한 결과이다. 본 논문에서 제안하는 특징 요소를 반영한 점의 배치 방법을 통해, 움직이는 대상의 특징들이 잘 표현된 결과를 보여준다. [그림25]은 물고기가 꼬리를 흔드는 동영상을 입력으로 한 결과이다. 물체의 색상이 비슷한 톤을 나타내지만, 점들이 특징 라인에 영향을 받기 때문에 디테일한 형태를 잘 표현할 수 있다. [그림26], [그림27]는 애니메이션 동영상을 입력으로 하여, 기존 애니메이션 입력 영상과는 또 다른 느낌의 점묘화 스타일 애니메이션을 생성하였다.

	초기 점 개수	프레임 개수
그림24	1600개	8 프레임
그림25	600개	12 프레임
그림26	800개	12 프레임
그림27	1500개	9 프레임

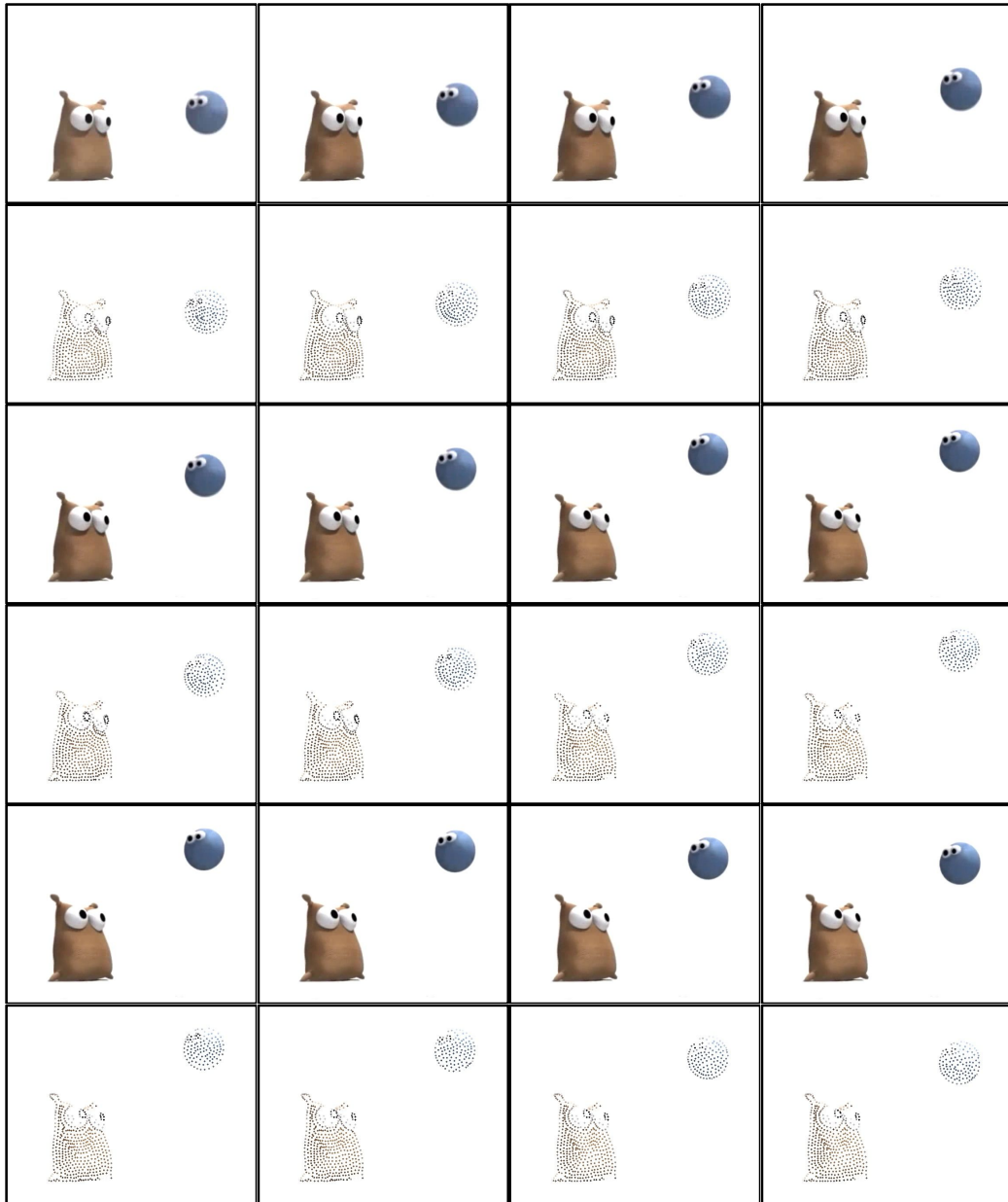
[표7] 점묘화 동영상 렌더링 파라미터



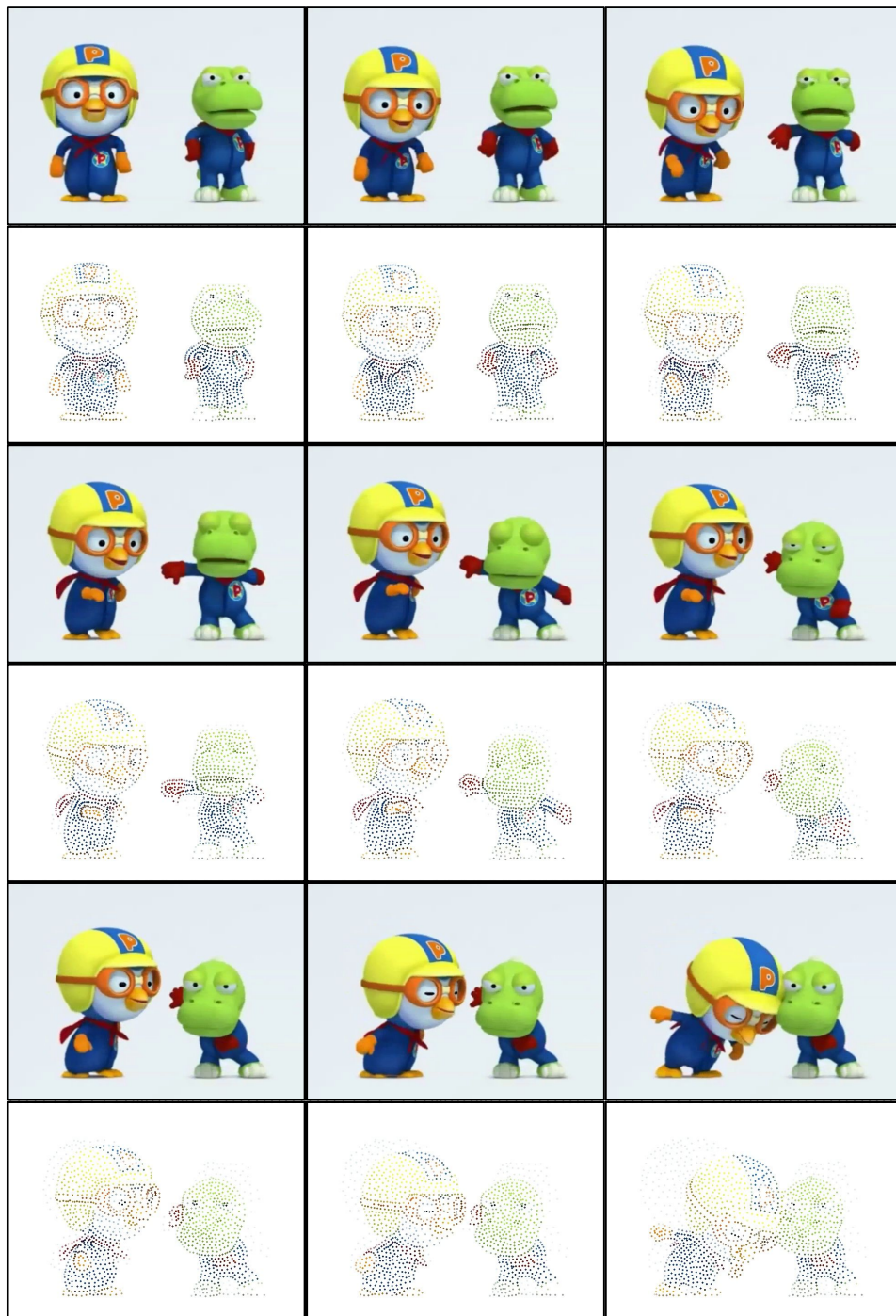
[그림24] 점묘화 동영상 결과1



[그림25] 점묘화 동영상 결과2



[그림26] 점묘화 동영상 결과3



[그림27] 점묘화 동영상 결과4

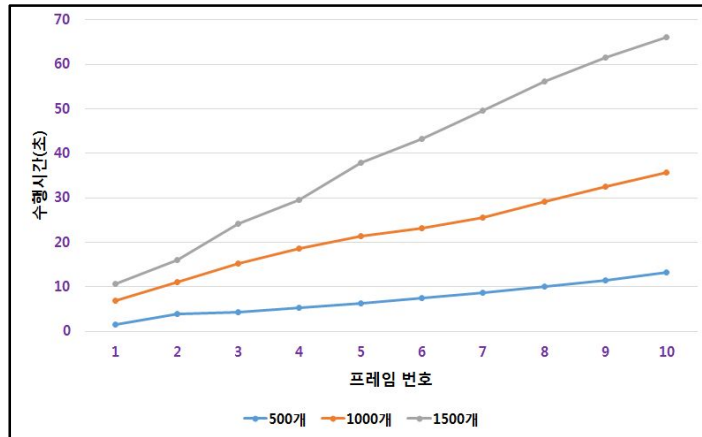
### 3. 수행 시간

본 연구에서 제안하는 방법을 사용하여 점묘화 동영상 생성하는데 걸리는 시간을 측정하였다. 입력 동영상은 440x320 픽셀 크기의 컬러 영상을 사용하였다. [표8]은 프레임 별로 점묘화를 생성하는데 소요되는 시간을 측정 한 결과이다. 1번째 프레임부터 10번째 프레임까지 수행 시간을 측정하였으며, 초기 점이 500개인 경우, 프레임 당 평균 소요 시간은 약 1.4초였다. 점의 개수가 1000개, 1500개인 경우는 프레임마다 평균적으로 각각 3.5초, 6.5초가 소요되었다.

프레임 번호	500개	1000개	1500개
1	1.6	6.9	10.6
2	2.3	4.1	5.4
3	1.4	4.2	8.2
4	1.0	3.4	5.4
5	1.1	2.8	7.2
6	1.0	1.8	5.4
7	1.2	2.4	6.5
8	1.4	3.5	6.5
9	1.5	3.4	5.3
10	1.8	3.2	4.6

[표8] 수행 시간 측정 결과 (시간 단위: 초)

[그림28]은 프레임별로 측정된 누적 수행 시간을 나타낸 그래프이다. 점의 개수가 500개, 1000개, 1500개 일 때, 각각 약 13초, 35초, 66초가 소요되었다.



[그림28] 누적 수행 시간 측정 결과

## VI. 결론 및 향후 연구

기존의 동영상 점묘화 연구[1]은 점의 밀도를 조절하여 점을 분포시키고, 예지에 추가로 점을 분포하는 방식으로 점묘화 동영상을 생성했다. 본 논문은 이미지의 특징 분석을 통해 특징표현이 강화된 동영상 점묘화 기법을 제안하였다. 이 방법은 특징 라인 이미지를 이용하여 점의 배치에 영향을 줌으로써, 밝기에 따라 점의 밀도를 조절하는 것은 물론 특징을 효과적으로 강조한다. 그리고 자동으로 점묘화 동영상을 생성하기 때문에, 전문가뿐 아니라 일반 사용자도 쉽게 사용할 수 있다. 본 연구 방법으로 생성한 점묘화 동영상 콘텐츠는 광고, 전시 등 다양한 분야에서 활용할 수 있을 것이다.

향후 연구로는 입력 동영상의 제약을 해결할 수 있는 알고리즘의 개선이 필요하다. 본 연구 방법은 입력 영상의 밝기 요소를 점의 밀도에 반영하여 점묘화 동영상을 생성하는데, 원본 동영상의 움직이는 물체의 색상이 흰색에 가까운 경우에는 점이 적게 분포하여, 방향성 있는 점들의 배치를 얻기 어렵다. 또한 프레임 간에 급격한 차이가 있을 경우, 시간적 일관성을 고려하기 위한 방법을 모색해야 할 것이다. 향후 연구로는 GPU 프로그래밍을 이용한 속도 개선이 필요하다. 본 논문에서는 PC기반에서 연구가 이루어졌으나, 스마트폰과 같은 모바일 기기에 특화된 연구도 고려해 볼 수 있다. 그리고 사용자 인터페이스를 구축하여, 사용성을 강화할 수 있을 것이다.

## 참고문헌

- [1] Houit, Thomas, and Frank Nielsen. "Video stippling." *Advances Concepts for Intelligent Vision Systems*. Springer Berlin Heidelberg, 2011.
- [2] Kim, Dongyeon, et al. "Feature guided Image Stippling." *Computer Graphics Forum*. Vol. 27. No. 4. Blackwell Publishing Ltd, 2008.
- [3] Gooch, Bruce, and Amy Gooch. "Non-photorealistic rendering." Vol. 2. Wellesley: AK Peters, 2001.
- [4] Yang, Chuan-Kai, and Hui-Lin Yang. "Realization of Seurat's pointillism via non-photorealistic rendering." *The Visual Computer* 24.5, 2008.
- [5] Wu, Yi Chian, et al. "Generating pointillism paintings based on Seurat's color composition." *Computer Graphics Forum*. Vol. 32. No. 4. Blackwell Publishing Ltd, 2013.
- [6] Dongxiang. "A Natural Image Pointillism with Controlled Ellipse Dots.", *Advances in Multimedia*, 2014.
- [7] Deussen, Oliver, et al. "Floating points: A method for computing stipple drawings." *Computer Graphics Forum*. Vol. 19. No. 3. Blackwell Publishers Ltd, 2000.
- [8] Secord, Adrian. "Weighted voronoi stippling." *Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*. ACM, 2002.
- [9] Mould, David. "Stipple placement using distance in a weighted graph." *Proceedings of the Third Eurographics conference on*

Computational Aesthetics in Graphics, Visualization and Imaging. Eurographics Association, 2007.

[10] Son, Minjung, et al. "Structure grid for directional stippling." *Graphical Models* 73.3, 2011.

[11] Kim, Sung Ye, et al. "Stippling by example." *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*. ACM, 2009.

[12] Okabe, Atsuyuki, et al. "Spatial tessellations: concepts and applications of Voronoi diagrams." Vol. 501. John Wiley & Sons, 2009.

[13] Canny, John. "A computational approach to edge detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6, 1986.

[14] Borgefors, Gunilla. "Distance transformations in digital images." *Computer vision, graphics, and image processing* 34.3, 1986.

[15] Bradski, Gary. "The opencv library." *Doctor Dobbs Journal* 25.11, 2000.

# **ABSTRACT**

## **A Method for Generating Stippled Video Reflecting Feature Elements**

**Seulgi Oh**

**Dept. of Computer Science**

**The Graduate School**

**Sungshin Women's University**

This thesis proposes a system for generating a stippled video expressing feature elements in input video. In many fields of computer graphics, previous stippling techniques have mainly aimed at a single image. On the other hand, studies targeting a video are inadequate. Previous work[1] focused on the brightness of the image in the step of dot distribution and additionally suggested a method to add some of patterns or dots to represent sharp areas of the image. However, this approach is not enough to express the features of moving object effectively. I was inspired from a directional point placement method[2] of the study which is targeting a single image. In this paper, I present a

system for generating stippled video. this system mainly consists of three steps. The first step is image preprocessing that generates four images such as gray image, feature line image, difference image. a gray image determines the density of points and a feature line image is produced in order to affect to feature flow also two difference images between a previous frame and a current frame are created to maintain time coherence of the video. The second step determines the initial position of points by using brightness and edge information of image then distributes the points by using the feature line image. The final step is the rendering of stippled video. This step performs rendering dots for each frame and synthesizing video. This method could make stippled video having the better feature representation of the moving object.