

“영화 만화” 생성을 위한 말풍선 자동 배치 기법

황원일^o, 천봉경, 류동성, 조환규

부산대학교 컴퓨터공학과

{wihwang, bkchun, dsryu}@pearl.cs.pusan.ac.kr, hgcho@pusan.ac.kr

An Automated Procedure for Word Balloon Placement in Cinema Comics

Won-Il Hwang^o, Bong-Kyung Chun, Dong-Sung Ryu and Hwan-Gue Cho

Department of Computer Engineering, Pusan National University

Abstract

영화와 같은 사실적인 영상은 너무 많은 정보를 포함하고 있어서 때로는 사용자에게 제작자의 의도를 쉽게 전달하기 어려울 수 있다. 이와 달리 만화와 같은 비사실적인 영상은 중요하지 않은 부분을 아주 간단하게 표현하거나 혹은 중요한 부분만을 강조해서, 작가의 의도를 직관적으로 훨씬 잘 전달해준다. 만화는 스피드라인, 말풍선, 배경 텍스트 등과 같은 만화 고유의 표현 기법들을 통하여 일반 정지 영상들보다 많은 부가적인 정보를 전달해 준다. 특히, 말풍선은 장면 내의 인물들의 대사를 전달하는 매체로, 올바른 내용 전달을 위해 읽기 순서(reading order)에 따른 말풍선간의 관계 및 말풍선과 인물간의 관계를 고려하여 적절한 위치에 배치 되어야 한다. 본 논문에서는 비디오 영상에서 추출된 장면의 만화적 변환을 위한 기법들 중에서 말풍선 자동 배치 방법을 제안한다. 제안한 방법은 지도에서 지명을 배치할 때 흔히 사용하는 특징점 라벨 배치(Point-Feature Label Placement) 방법에 기반하였다. 또한, 본 논문에서는 말풍선 배치가 얼마나 잘 되었는가를 평가하기 위한 객관적인 평가 함수를 제안하였다. “Love Actually”, “살인의 추억” 등의 다양한 영화 장면들을 이용하여 실험한 결과, 제안한 방법의 말풍선 배치 결과가 사람이 수작업으로 배치한 것과 비교해 크게 부족함이 없는 것을 알 수 있었다.

1 서론

컴퓨터 그래픽스 기법으로 제작된 사실적인 영상들을 보면, 실제와 매우 유사하고 깨끗하다. 하지만, 사물이나 장면의 모든 정보를 표현하는 것은 상당히 많은 연산을 필요로 할 뿐만 아니라, 인지적인 관점에서도 효율적이지 못하다. 이러한 이유로 최근에는 장면을 강조하거나 생략하는 등의 기법을 사용하여 비사실적으로 표현하는 NPR(Non-Photorealistic Rendering, 이하 NPR)분야에 대한 연구가 활발히 이루어지고 있다.

만화는 NPR분야에서 가장 오랫동안 연구되어 온 분야이며, 실생활에서 다양하게 응용되고 있어 컴퓨터 그래픽스를 연구하지 않는 일반 사람들에게도 친숙한 매체이다. Scott McCloud의 저서 “Understanding Comics” [1]와 “Reinventing Comics” [2]에서는 일반적인 만화의 기본 구조와 다양한 특수 기법, 그리고 의사전달 도구로서의 효율성에 대해 설명하고 있다. 만화는 과장된 동작이나 배경 텍스트, 그리고 말풍선 등을 이용하여 인물의 감정을 쉽게 표현할 수 있다. 그러므로, 만화에 사용되는 특수 효과와 구성 요소들을 다른 매체에 적용하면 인지적인 측면에서 장면을 이해하는데 도움이 된다.

영화와 같은 비디오 영상을 자동화된 방법으로 만화책으로 변환하는 것은 상당히 흥미로운 주제이다. 만화는 정지 영상의 한정된 매체에서 인물의 동적인 움직임이나 시간의 흐름, 그리

고 청각적인 요소인 소리까지 표현하기 위해 다양한 만화적 기법들을 사용한다. 가장 대표적인 기법으로는 객체의 움직임을 표현하기 위한 스피드라인과 장면의 분위기를 표현하기 위한 배경 효과, 그리고 소리를 전달하기 위한 말풍선이 있다. 영화를 만화책으로 자동으로 변환하기 위해서는 이런 다양한 만화적 기법들을 자동화하는 연구가 필요하다.

이전부터 많은 연구자들이 만화가 가지는 재미있는 특징들에 대해서 연구하였는데, 사실적인 영상을 추상화하여 만화적 영상으로 변환하는 카툰닝(cartooning) 연구 [3, 4, 5]와 정지 영상에서 객체의 동적인 움직임을 표현하는 연구 [6, 7, 8]들이 주를 이루었다.

만화 장면에서 인물들 간의 대사 전달을 위한 말풍선에 관한 연구를 살펴보면 *Comic Chat* [9]이 유일한 연구이다. 이 연구는 인터넷 만화 채팅 프로그램에 관한 연구로 말풍선 배치에 관한 언급을 하고 있으나, 장면에서 말풍선을 배치를 위한 영역이 따로 존재한다. 따라서, 말풍선을 위한 영역이 따로 존재하지 않는 영화와 같은 영상에서는 적용하기 어렵다. 또한, 여기에서는 객체와 말풍선간의 관계를 고려하지 않고, 단순히 말풍선들간의 읽기 순서(reading order)만을 고려한 배치 방법으로 배치 결과가 실제 사람이 그린 만화와 차이가 나며, 인지적인 측면에서 좋지 않다.

본 논문에서는 비디오 영상에서 추출된 장면에 말풍선을 배

치하기 위해 지도에서 지명을 배치할 때 흔히 사용하는 특징점 라벨 배치(Point-Feature Label Placement) [10, 11] 방법에 기반한 말풍선 자동 배치 방법을 제안한다. 제안하는 방법은 두 단계로 구성된다. 첫 번째 단계는 특징점 라벨 배치 방법을 확장하여 말풍선들간의 읽기 순서를 만족하면서 해당 인물에 위치시키는 것이다. 두 번째 단계는 말풍선 배치 결과가 좀 더 자연스럽게 보이도록 하기 위해 본 논문에서 제안한 말풍선 배치 결과와 평가 함수에 따라 말풍선들의 위치를 조정하는 것이다. 제안한 방법은 말풍선간의 읽기 순서 관계뿐만 아니라 말풍선과 인물간의 관계를 함께 고려하기 때문에 사람이 그린 만화와 비슷한 배치 결과를 도출하며, 인지적인 측면에서 이전 연구보다 훨씬 좋다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 일반 정지 영상에서 말풍선 배치를 위한 문제를 정의하고, 3장에서는 본 논문에서 제안한 특징점 라벨 배치 방법을 확장한 말풍선 배치 방법을, 그리고 4장에서는 인물의 외침과 같은 말풍선을 위한 모델을 소개한다. 그 후, 5장에서는 다양한 영화 장면들에서의 말풍선 배치 결과를 살펴보고, 끝으로 6장에서 결론 및 향후 과제에 대해서 논한다.

2 말풍선 배치 문제 정의

NPR 분야에서 논의되는 대부분의 문제들은 객관적인 기준보다는 사람의 직관에 의한 주관적인 판단에 의해 결과가 평가된다. 말풍선 배치 문제도 표준화된 기준이 정립되어 있지 않아 결과를 평가하는 것은 상당히 까다롭다. 하지만, 오랫동안 만화가들은 말풍선들간의 읽기 순서(reading order)를 유지하면서 해당 인물을 주위에 적절하게 말풍선들을 배치하는 규칙들을 지켜왔다. 이러한 관습적인 규칙을 기반으로 하여, 말풍선 배치 문제를 정의하고, 말풍선 배치 작업 시 지켜야하는 규칙 및 말풍선 배치 결과 측정 함수에 대해서 설명한다.

2.1 장면 구성 요소

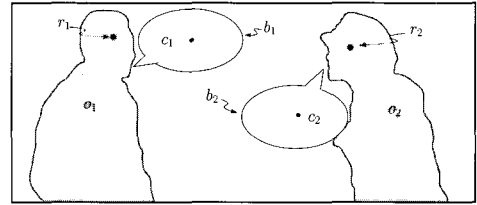
말풍선 배치 작업은 장면을 구성하는 개체(인물)와 말풍선, 그리고 장면의 경계 등의 정보를 토대로 이루어진다. 따라서 말풍선을 배치하기 위해서는 장면 내의 구성요소들을 명확히 정의할 필요가 있다. 그림 1은 장면을 구성하는 요소들을 보여준다. 먼저, o_i 는 i 번째 말풍선의 소유자인 특정 개체를 의미한다. r_i 는 최종적으로 말풍선 배치 시 기준이 되는 o_i 내의 참조점(reference point)을 의미하며, 인물의 얼굴 영역 혹은 단순히 눈이나 입의 위치가 될 수 있다.

전체 장면에서 배경 이미지로부터 주요 개체 o_i 를 추출하는 알고리즘과 이를 구현해놓은 도구들은 이미 많이 존재한다. 따라서, 본 논문에서는 장면의 주요 구성 요소인 o_i 와 r_i 는 이미 발표된 논문이나 프로그램을 이용하여 얻을 수 있다고 가정한다. 각각의 개체 o_i 에는 중심이 $c_i = (x_i, y_i)$ 인 하나의 말풍선 b_i 가 할당된다. 하지만, 이는 반드시 지켜야할 규칙은 아니며, 특정한 경우에 한하여 하나 이상의 말풍선이 개체에 할당되기도 한다. 이런 경우에는 각각의 b_i 마다 r_i 를 추가하여 참조점과 해당 말풍선이 1:1 맵핑 관계를 유지하도록 하면 된다. 그리고 각각의 b_i 내에는 말풍선이 전달할 내용에 해당하는 대사 d_i 가 포함된다. 즉, b_i 와 d_i 의 관계는 $Map(d_i) = b_i$ 로 나타낼 수 있다(표 1). 대사 d_i 의 순서는 시간 순서에 따라 결정되는데, 이는

말풍선 배치 시 중요한 제약 조건이 된다. 즉, d_{i+1} 는 d_i 보다 시간적으로 다음에 나와야 하는 대사가 된다.



(a) 입력 영상



(b) 입력 영상의 구성 요소 표현

Figure 1: 말풍선 배치를 위한 장면의 구성요소

d_i	SCRIPT WORDS	Map(d_i)
d_1	LUFFY is going to do something funny again.	b_2
d_2	Ha Ha Ha ! Go for it.	b_1
d_3	Let's see what you're gonna do!	b_1

Table 1: 대사 d_i 와 주된 말풍선 b_i 의 함수 관계, $Map(d_i) = b_i$.

2.2 말풍선 배치를 위한 일반적인 규칙

말풍선은 텍스트 형식의 인물의 대사나 생각을 그래픽적인 형식으로 독자에게 전달하는 역할을 담당한다. 소리가 시간의 순서에 따라 차례로 전달되는 것과는 달리 텍스트는 독자가 그 순서를 선택하여 읽는다. 일반적인 문장의 경우, 독자는 문장을 읽을 때 위에서 아래로, 그리고 왼쪽에서 오른쪽 순서로 읽는다. 말풍선 내의 대사를 올바르게 전달하기 위해서는 이와 같은 말풍선들간의 읽기 순서(reading order)에 맞게 배치하는 것이 중요하다.

장면 내 말풍선들의 집합을 b_i 라고 가정하고, b_i 의 읽기 순서를 $RO(b_i)$ 로 정의하자($RO(b_i)$ 값이 작은 말풍선이 먼저 읽어야 되는 말풍선이 된다). 그리고 그림 2에서 보는 것과 같이 b_i 를 둘러싸는 최소한의 수평 사각 영역을 $Zone(b_i)$ 라고 하자. 기본적으로 위에서 아래 그리고 좌에서 우로 말풍선을 읽는다고 가정할 때, 말풍선들의 위치에 따라 $RO(b_i)$ 를 결정하는 규칙은 다음과 같다.

- b_i 의 중심 좌표 c_i 가 b_j 의 중심 좌표 c_j 보다 높은 경우
 - x_i 가 x_j 보다 작으면 $RO(b_i)$ 가 $RO(b_j)$ 보다 작다.

- x_i 가 x_j 보다 크고, c_i 가 $Zone(b_j)$ 보다 높으면 $RO(b_i)$ 가 $RO(b_j)$ 보다 작다.
 - x_i 가 x_j 보다 크고, c_i 가 $Zone(b_j)$ 에 포함되면 $RO(b_i)$ 가 $RO(b_j)$ 보다 크다.
2. b_i 의 중심 좌표 c_i 가 b_j 의 중심 좌표 c_j 보다 낮은 경우
- x_i 가 x_j 보다 작고, c_i 가 $Zone(b_j)$ 에 포함되면 $RO(b_i)$ 가 $RO(b_j)$ 보다 작다.
 - x_i 가 x_j 보다 작고, c_i 가 $Zone(b_j)$ 보다 낮으면 $RO(b_i)$ 가 $RO(b_j)$ 보다 크다.
 - x_i 가 x_j 보다 크면 $RO(b_i)$ 가 $RO(b_j)$ 보다 크다.

그림 2은 앞서 설명한 규칙에 따라 말풍선들의 위치에 따른 RO값을 보여준다.

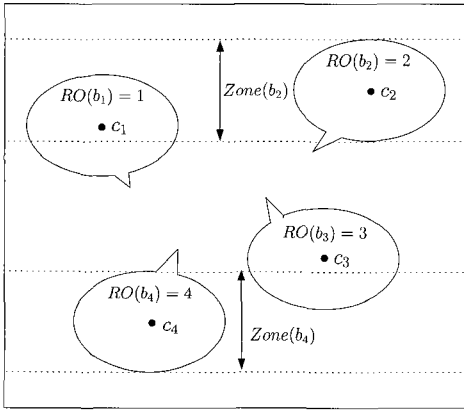


Figure 2: 만화에서 말풍선의 읽기 순서의 예

말풍선 배치 작업 시 말풍선들간의 읽기 순서를 지키는 규칙 외에 추가적으로 다음과 규칙들을 만족하여야 한다. 이 규칙들은 되도록 지켜져야 하겠지만, 예외적으로 장면내 말풍선 배치를 위한 공간이 적을 경우 부분적으로 위배될 수 있는 규칙이다.

1. $b_i \cap b_j = \emptyset$: 말풍선들은 서로 겹치지 않는다.
2. $b_i \cap o_j = \emptyset$: 모든 말풍선은 개체의 참조점(일반적으로 얼굴 영역)을 가리지 않는다.

2.3 말풍선 배치 결과 평가 함수

말풍선 배치가 잘 이루어졌는가를 판단하는 것은 상당히 주관적인 문제이다. NPR분야의 연구 자체가 주관적인 성향이 강하기 때문에 표준을 정하거나, 성능 평가 기준을 세우기 어렵다. 본 논문에서는 말풍선 배치의 기본적인 특성을 고려한 배치 측정 함수를 제안한다. 올바른 말풍선 배치를 위해서는 다음의 두 가지 규칙이 잘 지켜져야 한다.

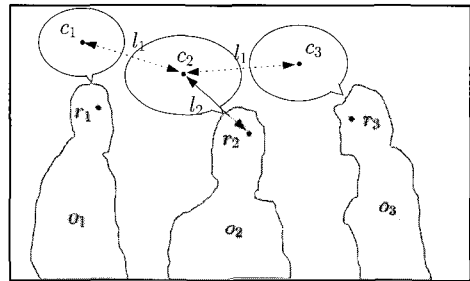
1. 말풍선-개체 관계: b_i 는 가능한 o_i 의 참조점 r_i 에 가깝게 배치되어야 한다.

2. 말풍선-말풍선 관계: b_i 는 가능한 다른 말풍선 b_j 와 멀리 배치되어야 한다.

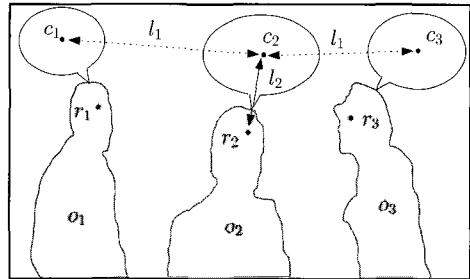
첫 번째 규칙은 모든 만화에서 공통적으로 지켜지고 있는 것으로 말풍선이 해당 인물 가까이 위치하는 것이 인지적인 측면에서 바람직하다. 두 번째 규칙은 여유 공간에 따라 유동적으로 적용되는 규칙으로 장면 내에 말풍선을 자유롭게 배치할 수 있는 공간이 충분하지 않은 경우에는 결과에 크게 영향을 미치지 않는다. 즉, 첫 번째 규칙과 두 번째 규칙을 모두 반영하되 두 가지 요소에 가중치를 적용할 수 있도록 측정 함수를 정의할 필요가 있다. 이러한 점들을 고려하여 말풍선 배치 측정 함수를 정의하면 다음과 같다.

$$QLayout(\{b_i\}) = k_1 \cdot L_1 - k_2 \cdot L_2$$

L_1 은 서로 다른 개체 o_i 에 할당된 모든 말풍선 b_j 들간의 사이의 거리의 합이다. L_2 은 r_i 와 c_i 와 사이의 거리 합이다. k_1 과 k_2 는 L_1 과 L_2 사이의 균형을 조절하기 위한 상수 값들로 $k_1 + k_2 = 1$ 을 만족한다. 그림 3는 동일한 장면에서 서로 다른 배치 결과와 그에 따른 $QLayout(\{b_i\})$ 값을 보여준다. 표 2는 $QLayout(\{b_i\})$ 을 구하는 과정에서 계산하는 세부 값들을 상세히 보여준다. $QLayout(\{b_i\})$ 값이 클수록 결과가 좋은 것이기 때문에 그림 3 (b)는 (a)보다 배치가 잘 되었다고 할 수 있다.



(a) $QLayout(\{b_i\}) = 753.4$



(b) $QLayout(\{b_i\}) = 1358.5$

Figure 3: 동일한 장면에서 말풍선의 다른 배치 결과 예. $QLayout(\{b_i\})$ 이 높은 (b)가 (a)보다 좋은 결과를 보여준다.

말풍선 배치 문제는 최종적으로 최적화된(또는 최적에 가까운) 말풍선 배치를 구하는 방법이다. 본 논문에서는 이 문제를 OBL(Optimal Balloon Layout)이라 명하며, OBL은 다음과 같이 정의된다.

Input: $\{< o_i, r_i >, \{d_j\}\}$: 배경으로부터 분리된 개체들과 모

Distance	Figure 3 (a)	Figure 3 (b)
$d(c_1, r_1)$	77.36	89.00
$d(c_2, r_2)$	101.98	91.59
$d(c_3, r_3)$	87.00	109.60
$d(c_1, c_2)$	123.01	234.11
$d(c_1, c_3)$	252.24	411.20
$d(c_2, c_3)$	134.62	179.04
L_1	1019.74	1648.70
L_2	266.34	290.19
$Qlayout(\{b_i\})$	753.40	1358.51

Table 2: 말풍선 배치 작업을 수행한 후 측정 함수 결과 비교

든 o_i 에 할당된 일련의 대사들의 집합

Output: $b_i = \langle c_i(x_i, y_i) \rangle : RO()$ 를 만족하면서 $Qlayout(\{b_i\})$ 을 최대화 하는 말풍선들의 배치 위치

3 말풍선 배치를 위한 자동화 기법

3.1 특징점 라벨 배치

특징점 라벨 배치(Point-Feature Label Placement, 이하 PFLP)는 지도나 다이어그램의 가독성을 높이기 위해 특징점 주위에 텍스트 라벨을 배치시키는 문제를 말한다 [10, 11]. PFLP 문제는 텍스트 라벨의 배치를 최적화하는 문제로 생각할 수 있는데, 이를 위해 찾을 영역(search space)과 목적 함수(objective function) 두 가지 측면에 대해서 정의되어야 한다.

찾을 영역은 주어진 K 개의 특징점에 대해 텍스트 라벨을 위치시키기 위한 후보 위치라고 할 수 있다. 그림 4은 하나의 특징점에 대해 텍스트 라벨을 배치시키기 위한 전형적인 여덟개의 후보 위치를 보여준다.

목적 함수는 텍스트 라벨 배치 결과의 성능을 향상 시키기 위해 찾을 영역(라벨 후보 위치들)에 대해 할당할 상대적인 가중치 값을 산출한다. 하지만, 라벨링 결과의 성능은 사람의 인지적인 측면의 영향을 많이 받기 때문에 목적 함수를 정의하는 것은 상당히 어려운 문제이다. 따라서 목적 함수는 다음과 같은 일반적인 세 가지 요소를 기본적인 평가 잣대로 하여 정의 하되, PFLP 알고리즘이 사용되는 용도에 따라 평가 잣대를 변경하여 사용한다.

- 텍스트 라벨과 장면의 그래픽적인 요소와의 겹치는 횟수
- 그림 4에서 보여주는 전형적인 후보 위치의 우선 순위

그림 4에서 각 라벨의 후보 위치들의 수치값은 상대적인 가중치를 뜻한다. 수치값이 낮은 라벨 위치가 수치값이 높은 라벨 위치보다 위치 선정시 상위 우선순위를 가지게 된다. 즉, PFLP는 상대적인 수치값의 합을 최대 또는 최소화 하기 위한 각 특징점의 후보 위치를 찾는 문제라고 할 수 있다. 단, 선택된 후보 위치들은 다른 특징점의 선택된 후보 위치들과 겹치지 않도록 해야 한다.

본 논문에서 해결하고자 하는 말풍선 배치 문제(Word Balloon Placement, 이하 WBP)도 위에서 설명한 PFLP문제로 생

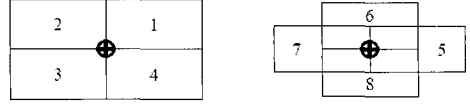


Figure 4: 특징점에 대한 전형적인 여덟 개의 후보 라벨 위치 집합 [10]

각할 수 있다. 그림 5에서 보여주는 것처럼, 인물의 참조점인 r_k 와 해당 말풍선의 잠재적인 위치들은 특징점과 그 특징점의 잠재적인 라벨 위치들로 간주하면 PFLP 문제와 동일하다. 만 화에서 말풍선이 해당 인물과 멀리 떨어져 긴 말풍선 꼬리를 가지는 경우도 존재하지만, 이러한 경우는 시각적으로 독특함을 보여줄 수 있지만 가독성 측면에서는 좋지 못하다. 따라서 각각의 말풍선이 해당 인물의 참조점에 가까이 위치하는 것을 올바른 배치 결과로 여길 수 있다. 이러한 특징을 기반으로 하여 WBP는 PFLP를 이용하여 다음과 같이 정의된다.

- 주어진 K 개의 참조점 r_k 와 각각의 r_k 에 대해서 가능한 위치들의 개수를 N_k 라고 하면, 각각의 잠재적인 말풍선 위치는 변수 $P_{k,i}$ 이다($1 \leq k \leq K$ 이고, $1 \leq i \leq N_k$).
- 각각의 $P_{k,i}$ 는 해당 위치상에서의 다른 말풍선(b_k)의 존재 또는 부재를 나타내는 1 또는 0의 값을 가진다.
- 하나의 말풍선은 해당 r_k 의 후보 위치 중 반드시 한 곳에 만 위치하여야 한다.

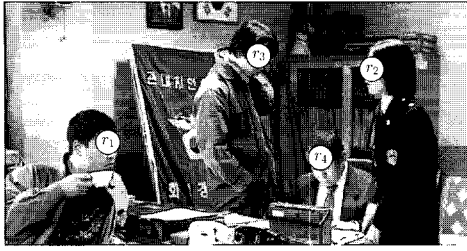
$$\sum_{i=1}^{N_k} P_{k,i} = 1, \text{ for } 1 \leq k \leq K$$

- 서로 다른 말풍선들의 후보 위치들 사이의 겹치는 후보 위치들의 경우의 쌍들을 Q 라고 하였을 때, 서로 다른 말풍선들은 겹치지 않아야 된다:
각각의 겹치게 되는 후보 위치에 대해 $P_{r_q, s_q} + P_{r'_q, s'_q} \leq 1, 1 \leq q \leq Q$ 이다.
- 목적 함수는 $Qlayout(\{b_i\})$ 값이 최대가 되게 하는 것이다. 더 큰 목적 함수는 더 나은 배치 결과를 얻게 한다.

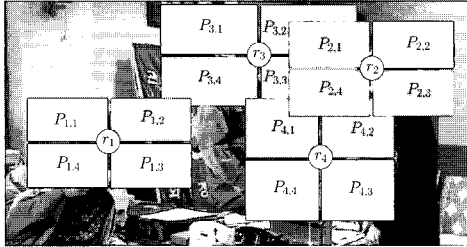
그림 5는 간단한 라벨링 알고리즘을 이용하여 WBP문제를 해결하는 예를 보여준다. 이는 앞에서 설명한 바와 같이 여러 $P_{k,i}$ 집합 중에 $Qlayout(\{b_i\})$ 를 최대화 하면서 아래의 조건을 만족하는 $P_{k,i} = 1$ 의 집합을 구하는 것이다.

$$\begin{aligned} P_{1,1} + P_{1,2} + P_{1,3} + P_{1,4} &= 1, \\ &\vdots \\ P_{4,1} + P_{4,2} + P_{4,3} + P_{4,4} &= 1, \text{ and} \\ P_{1,2} + P_{3,4} &\leq 1, \quad P_{3,4} + P_{4,1} \leq 1, \\ P_{3,2} + P_{2,1} &\leq 1, \quad P_{3,3} + P_{2,1} \leq 1, \\ &\vdots \\ P_{2,4} + P_{4,2} &\leq 1, \quad P_{2,3} + P_{4,2} \leq 1 \end{aligned}$$

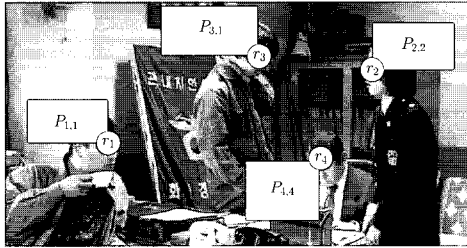
그림 5(a)는 입력 이미지와 r_k 를 보여주며, 각각의 r_k 에 대한 $P_{k,i}$ 를 구하는 단계는 그림 5(b)에서 보여주며, 최종결과는 그



(a) 각각의 개체에 대해 r_k 의 위치를 정한다.



(b) 각 r_k 에 대한 말풍선 후보 위치를 생성한다.



(c) 말풍선 배치 결과

Figure 5: 라벨링 알고리즘을 이용한 말풍선 배치 과정

림 5(c)에서 보여준다. 이 예에서 r_k 의 해당 말풍선은 b_k 이고, $RO(b_k)$ 는 k 라고 가정하자. 즉, $RO(b_k) < RO(b_{k+1})$ 이다. 그림 5와 같이 r_k 의 위치가 주어질 경우, 어떤 PFLP 알고리즘을 적용하더라도 말풍선들간의 읽기 순서 제약 조건을 만족하지 못한다. 이는 r_k 의 위치가 고정되어 있기 때문에 b_k 가 위치할 수 있는 영역이 제약받기 때문이다. 그러므로, 영화의 장면과 같이 r_k 의 위치가 미리 정해진 영상의 경우, PFLP 알고리즘을 말풍선 배치 문제에 적용하기 위해서는 말풍선들간의 읽기 순서를 유지하기 위한 방법의 추가가 필요하다.

3.2 말풍선 배치를 위한 확장된 특징점 라벨 배치 방법

본 논문에서는 말풍선 배치를 위해 확장된 특징점 라벨 배치(Extended Point-Feature Label Placement, 이하 EPFLP) 방법에 기반한 발견적 기법(heuristic algorithm)을 제안한다. 이를 위해 그림 6(a)에서와 같이 확장된 말풍선 후보 영역($EP_{k,i}$)을 정의하였다. $EP_{k,i}$ 는 $RO(b_k)$ 를 지키면서 b_k 를 배치할 수 있도록 추가적인 영역을 제공한다. 이는 PFLP와는 달리 WBP에서는 말풍선들간의 읽기 순서가 매우 중요한 제약 조건이기 때문

이다. b_k 는 앞에서 설명한 $P_{k,i}$ 와 같이 한 위치에 고정되는 것이 아니라 $EP_{k,i}$ 영역을 따라 이동하여 배치될 수 있다. 그림 6(a)에서 각각의 사각 영역은 $EP_{k,i}$ 를 나타낸다.

본 논문에서 제안하는 EPFLP 기반의 말풍선 배치 알고리즘은 두 단계로 구성된다. 첫 번째 단계에서는 EPFLP를 적용하여 $RO(b_k)$ 를 유지하면서 각각의 b_k 를 r_k 에 위치시킨다. 두 번째 단계에서는 말풍선 배치 결과 측정 함수인 $Qlayout(\{b_i\})$ 를 이용하여 $Qlayout(\{b_i\})$ 가 최대가 되도록 b_k 의 위치를 보정한 다. $Qlayout()$ 값이 클수록 해당 개체의 참조점에 가깝고, 다른 말풍선들과 멀리 떨어지기 때문에 보다 자연스럽게 보인다.

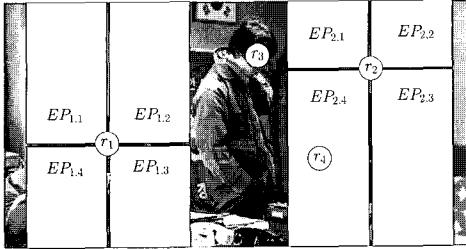
첫 번째 단계에 대해 자세히 알아보면 다음과 같다.

1. 각 r_k 에 대한 $EP_{k,i}$ 영역을 생성한 후, $RO(b_k)$ 의 순서에 맞게 각 r_k 의 말풍선 b_k 의 배치 작업을 수행한다. (r_k 의 말풍선은 b_k 이고, b_k 의 reading order인 $RO(b_k)$ 는 k 이다.)
2. r_1 는 가장 낮은 $RO()$ 값을 가지기 때문에 확장 후보 위치 영역인 EP_1 을 정할 때 $RO()$ 를 고려하지 않는다.
3. EP_1 을 정한 후, 그림 6(b)와 같이 b_1 을 위치시킨다.
4. 그림 6(b)에서 노란색 사각형 영역이 b_k 를 나타낸다.
5. 1 ~ 5번까지의 과정을 모든 말풍선의 배치가 끝날 때까지(k 번) 반복한다.

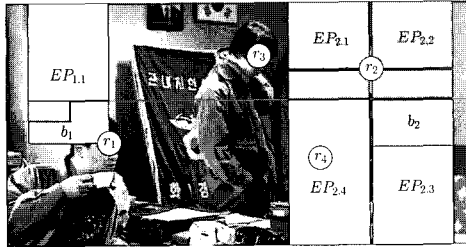
위의 과정에서는 각각의 말풍선 b_k 의 위치를 결정할 때 말풍선들간의 읽기 순서를 고려하지 않았다. 올바른 말풍선 배치를 위해서는 b_{k+1} 의 위치를 결정할 때 b_k 의 $RO(b_k)$ 를 고려하여 후보 위치를 제한해야 한다. b_{k+1} 이 배치 시 $RO(b_{k+1})$ 의 값을 만족시키기 위하여 b_k 로부터 b_{k+1} 가 배치되어야 할 제한 영역인 $BZone(b_{k+1})$ 를 계산할 수 있다. 그림 6(b)에서 빨간색 선은 $BZone(b_2)$ 을 나타낸다. 따라서, b_2 는 빨간색 선 아래에 배치되어야 한다. b_2 의 위치를 결정할 때, 먼저 $BZone(b_2)$ 영역에 따라 확장 후보 영역 $EP_{2,i}$ 을 활성화된 영역과 그렇지 않은 영역으로 나눈 뒤, 비활성 영역은 말풍선 후보 영역에 제외시킨다. 그리고 활성화된 EP_2 영역 중에서 가능한 한 r_2 과 가깝고, b_1 과 먼 영역에 위치시킨다. 이 단계는 그림 6(b)에서 확인할 수 있다. 비활성 $EP_{2,i}$ 영역은 회색으로 표현되어 있다. 이 과정을 모든 b_k 에 대해서 반복한다. 그림 6(c)는 첫 번째 단계를 마친 후의 결과를 보여준다.

앞에서 제안한 말풍선 배치 결과 측정 함수의 경우, r_k 와 b_k 사이의 거리, 그리고 b_k 와 b_l 사이의 거리에 따라 계산했다. 하지만, 직관적으로 서로 다른 말풍선들간의 거리보다 참조점과 해당 말풍선 사이의 거리가 말풍선 배치 결과에 더 큰 영향을 미친다. 그림 8(a)에서 $\sum d(b_i, r_i)$ 와 그림 8(b)에서 $\sum d(b_i, r_i)$ 값은 같지만, 배치 결과는 (b)가 (a)보다 직관적으로 낫다는 것을 알 수 있다. 즉, 측정 함수의 결과 값이 동일한 경우, $d(b_i, r_i)$ 값이 길면 길수록 배치 결과에 영향을 많이 미친다. 따라서, 말풍선과 특징점 사이의 거리에 더 큰 가중치를 두기 위해 $d(P_{k,i}, r_k)^2$ 와 같이 제곱형태로 표현하였다.

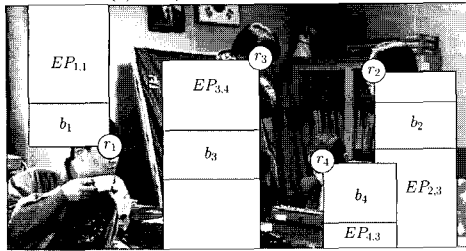
EPFLP 기반의 말풍선 배치 알고리즘의 첫 번째 단계에서 모든 말풍선 b_k 의 위치를 $RO(b_k)$ 에 맞게 정하는 작업을 수행하였다. 모든 말풍선은 해당 개체의 참조점을 가리키면서 말풍선들간의 읽기 순서를 만족하기 때문에 말풍선 배치 작업을 완료하였다고 볼 수도 있다. 하지만, 첫 번째 단계에서 결정된 말풍선의 위치는 $RO(b_k)$ 제약에 의해 배치 결과 측정값이 좋지 못한 경우가 많다. 따라서 보다 자연스러운 말풍선 배치 결과를



(a) 확장된 잠재적 말풍선 영역 $EP_{k,i}$ 를 생성한다.



(b) $EP_{k,i}$ 안에 b_k 를 배치시킨다.



(c) 첫 번째 단계를 마친 결과

Figure 6: EPFLP 기반의 말풍선 배치 알고리즘의 진행 과정(1단계)

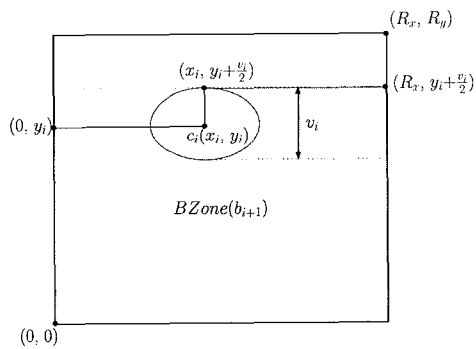
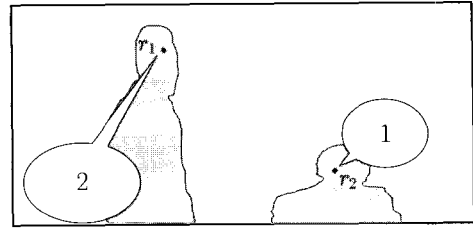
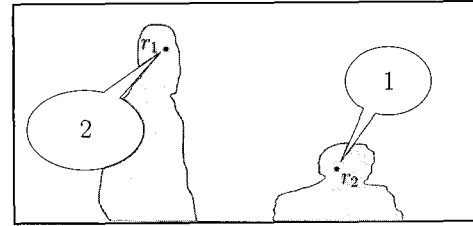


Figure 7: b_{i+1} 이 배치될 수 있는 제한 영역 $BZone(b_{i+1})$

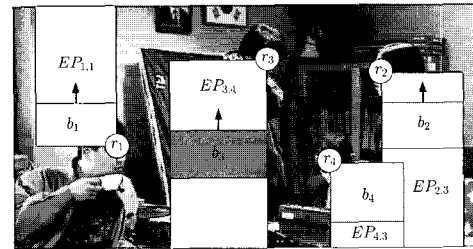


(a)

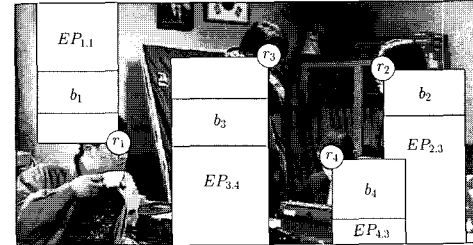


(b)

Figure 8: $d(b_k, r_k)$ 의 중요성: $\sum d(b_k, r_k)$ 값이 같지만 (b)가 (a)보다 더 나은 결과이다.



(a) $Qlayout()$ 에 기반하여 b_k 의 위치를 보정하는 과정.



(b) 최종 결과

Figure 9: EPFLP 기반의 말풍선 배치 알고리즘의 진행 과정(2단계)

생성하기 위해 $Qlayout(\{b_k\})$ 값에 기반하여 각각의 b_k 의 위치를 보정하는 작업을 수행하여야 한다.

말풍선의 위치를 보정하기 위해서는 말풍선 배치 결과 측정 함수인 $Qlayout(\{b_k\})$ 을 향상시키기 위하여 가장 큰 영향을 미치는 $d(b_k, r_k)$ 에 관심을 둔다. 먼저, $d(b_k, r_k)$ 값이 최대한 b_m 을 찾은 다음, b_m 을 ΔD 만큼씩 r_m 쪽으로 이동시킨다. 그리고 $RO(b_m)$ 보다 낮은 $RO(b_k)$ 값을 가지는 다른 말풍선들도 $RO(b_k)$ 를 유지하면서 해당 특징점 r_k 쪽으로 이동시킨다. 이 작업을 $QLayout(\{b_k\})$ 값이 증가할 때까지 반복한다. 그림 9(a)는 이 과정을 보여준다. 빨간색 사각형 b_3 는 $max(d(b_k, r_k))$ 인 b_m 을 나타내고, 화살표는 이동거리 ΔD 이다. 그림 9(b)는 2단계 과정까지 완료한 최종 결과를 보여준다.

지금까지 설명한 내용에서는 각 특징점 r_k 에 대해서 하나의 말풍선 b_k 만을 할당한다고 가정하였었다. 하지만, 이러한 가정은 실제 장면에서 한 인물이 여러 번의 대사를 전달하는 것도 동일하게 처리할 수 있기 때문에 유용하다. 즉, 한 인물이 하나 이상의 대사를 가지는 경우, 말풍선의 개수만큼 중복된 특징점을 생성하면 알고리즘 상의 아무런 변경 없이 정상적인 결과를 생성할 수 있다.

4 외침 말풍선을 위한 모델

일반 매체와 구분되는 만화의 특징 중 하나는 말풍선을 통해 장면의 상황을 표현할 수 있다는 것이다. 사람의 말이 아닌 소리 들 또는 기호들도 말풍선으로 표현하는 것이 가능하다 [1, 2]. 가장 일반적인 형태의 말풍선인 그림 10의 첫 번째 말풍선과 그 말풍선으로부터 모양이 변형된 여러 종류의 말풍선들은 일반 대화 뿐만 아니라 소리의 크기, 어투나 분위기, 그리고 소리를 내는 매체의 종류 등을 직관적으로 나타낸다. 특히 세 번째 종류의 말풍선은 주로 사람이 큰소리로 외치는 대사나 효과음을 표현하는데 쓰이는 것으로 만화에서 일반 말풍선 다음으로 많이 쓰인다. 한국과 일본의 만화들을 조사한 결과, 소리 말풍선은 다음과 같은 다섯 종류의 매개변수를 이용하여 특징 지을 수 있다.



Figure 10: 다양한 종류의 말풍선들

SOUND = < volume, pitch, duration, source, direction >

본 논문에서는 위의 다섯 종류의 매개변수를 이용하여 비디오 영상의 소리를 표현하는 방법을 제안한다. 기본적으로 모든 종류의 말풍선들은 네 개의 타원들 c_1, c_2, c_3 그리고 c_4 안에서 표현된다(그림 11). 말풍선들은 외부의 뾰족한 모서리 점들과 내부의 움푹 들어간 모서리 점들로 이루어진 닫힌 폴리곤들로 간주할 수 있다. 모든 외부 모서리 점들은 c_3 와 c_4 사이에 위치해야 하고, 내부 모서리 점들은 c_1 와 c_2 사이에 위치해야 한다. 정점 p_{2i+2} 와 p_{2i+1} 을 각각 외부 정점과 내부 정점으로 두면, 두 개의 인접한 정점들의 쌍인 p_i 와 p_{i+1} 사이에는 에지가 존재

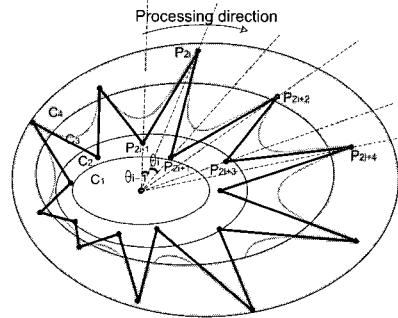


Figure 11: 특수(외침) 말풍선의 구조

한다. 또한, 두 개의 인접한 외부 정점들(p_i 와 p_{i+1}) 사이의 내각을 θ_i 로 나타낸다. 마지막으로 위에서 정의한 인접한 세 개의 정점(두 개의 외부 정점과 하나의 내부 정점)을 잇는 B-spline을 그리면, 말풍선의 모양이 완성된다. 그림 12은 본 논문에서 제안한 알고리즘으로 생성한 특수 말풍선들을 적용한 결과들이다. (a)는 일반 말풍선을 이용하여 인물의 대사를 표현하고 있고, (b)는 둔탁하게 부딪히는 효과음을 소리 말풍선으로 표현하고 있다. (c)와 (d)는 소리 말풍선의 모양을 더욱 뾰족하게 하여 사람의 외치는 소리와 긴박한 상황을 잘 나타내고 있다.

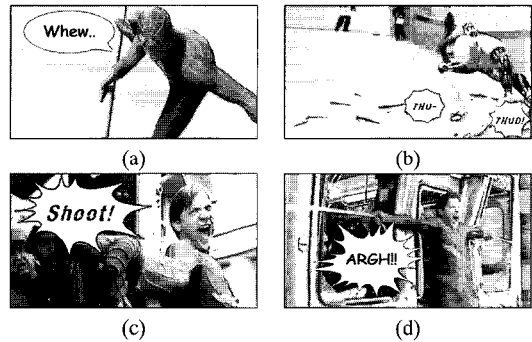


Figure 12: 말풍선의 모양에 따른 효과: 적절한 말풍선의 모양은 독자로 하여금 장면의 분위기를 보다 이해하기 쉽게 도와준다. (a)는 일반 말풍선을 적용한 결과이고, (b), (c), (d)는 매개변수 값을 달리한 외침 말풍선의 적용 결과이다.

5 실험 결과

본 논문에서는 비디오 영상에서 추출된 장면의 만화적 변환을 위한 CORVIS(COMics Rendering system with Video Stream)를 개발하였다. CORVIS에서는 정지 영상의 만화적 변환을 위해서 많은 기능을 제공한다. 필자가 이전 연구에서 제안되었던 정지 영상에서 객체의 만화적 움직임 표현을 위한 방법 및 장면의 분위기 연출을 위한 배경 텍스처 생성 방법뿐만 아니라 본 논문에서 제안한 자동 말풍선 배치 방법을 제공하고 있다.

그림 13는 CORVIS의 실행 화면을 보여주며, C++을 이용하여 개발하였다.

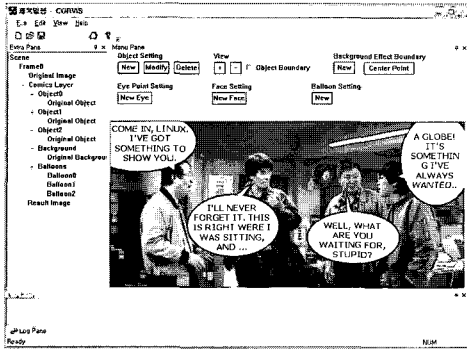


Figure 13: CORVIS 시스템

그림 14은 *ComicChat* [9], *ComicCreator* [12], 그리고 본 연구에서 제안한 알고리즘에 의한 배치 결과들을 보여준다. *ComicChat*의 결과인 그림 14(a)의 말풍선들의 읽기 순서에 맞게 잘 배치되어 있다. 하지만, 말풍선과 해당 개체(인물) 사이의 거리가 멀어서 가독성이 그리 뛰어나지 못하다. 이는 말풍선과 해당 개체 사이의 관계를 단지 말풍선 꼬리로 가리기만 하고, 거리를 전혀 고려하지 않기 때문이다. 그림 14(b)는 사용자가 시스템을 통해 수작업으로 만화 장면을 생성하기 위해 제작된 *ComicCreator*로 만들어진 결과이다. 여기서 말풍선 배치는 수작업으로 이루어지기 때문에 사용자가 말풍선들의 읽기 순서와 해당 개체와의 관계를 고려하여 직접 배치시킬 수 있다. 하지만, 장면내에 많은 수의 인물과 말풍선이 있어서 복잡한 경우 수작업으로 배치하는 것은 상당히 번거로운 작업이 된다. 또한 각각의 말풍선의 최적화된 위치를 한 번에 정하기는 어렵다. 그림 14(c)는 본 연구를 통해 생성된 결과 중 하나이다. 각각의 말풍선들은 읽기 순서와 해당 개체사이의 관계에 적합하게 배치되어 있다. (b)와 유사한 수준의 배치 결과이지만, 수작업 없이 가능하다는 점에서 큰 차이가 난다.

그림 15은 확장된 PFLP기반의 말풍선 배치 알고리즘의 두 번째 단계인 보정 작업을 수행하기 이전과 이후의 결과를 보여준다. 직관적으로 (b)가 (a)의 배치 결과가 낫다는 것을 알 수 있다. 즉, b_k 를 r_k 에 좀 더 가깝게 이동시켜 좀 더 나은 결과를 얻을 수 있다.

그림 16는 동일한 장면에서 서로 다른 말풍선들의 읽기 순서를 적용한 결과를 보여준다. 각각의 말풍선의 번호는 $RO(b_i)$ 를 의미한다. 이 그림을 통해 장면과 대사가 동일하더라도 말풍선들의 위치가 달라질 수 있음을 알 수 있다.

마지막으로 그림 17은 여러 유명한 영화 영상들을 입력으로 하여, 말풍선들을 배치시킨 다양한 결과를 보여준다. 첫 번째 이미지는 애니메이션 장르의 한 장면이고, 나머지 네 개의 이미지는 영화 장르의 장면들이다. 결과 그림들을 통해 본 논문에서 제안한 두 단계의 알고리즘을 이용하면 다양한 장면과 읽기 순서에 대해서 좋은 결과를 생성할 수 있는 것을 알 수 있다. 모든 결과들은 사람이 수작업으로 배치한 것과 비교해 부족함이 없다.



(a) *Comic Chat*에서의 말풍선 배치 결과 [9].



(b) *Comic Creator*에서의 말풍선 배치 결과 [12].



(c) 본 논문에서 제안한 방법의 말풍선 배치 결과

Figure 14: *Comic Chat*, *Comic Creator*, 본 논문에서 제안한 방법의 말풍선 배치 결과 비교

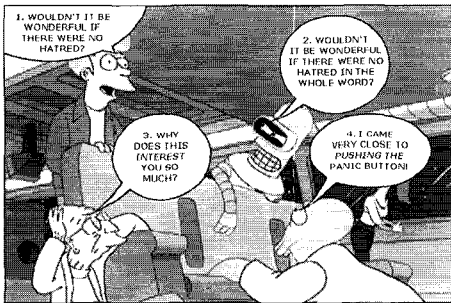


(a) QLayout에 따른 보정 작업 전의 결과

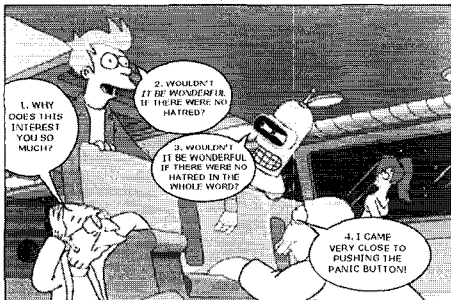


(b) QLayout에 따른 보정 작업 후의 결과

Figure 15: 말풍선 배치 시 QLayout에 따른 보정 작업을 적용하기 전과 후의 결과 비교. 보정 과정을 통하여 보다 나은 배치 결과를 얻을 수 있다.

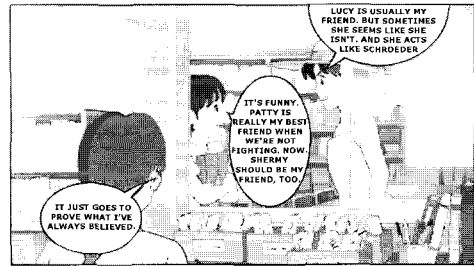


(a)



(b)

Figure 16: 동일한 장면에서 서로 다른 말풍선들의 읽기 순서에 따른 결과 비교. 각각의 말풍선 숫자는 $RO(b_i)$ 를 나타낸다.



(a) 영화 *Whisper of The Heart* 장면의 말풍선 배치 결과.



(b) 영화 *Mr. Hitch* 장면의 말풍선 배치 결과.



(c) 영화 *Mr. Hitch* 장면의 말풍선 배치 결과.



(d) 영화 *Memories of Murder* 장면의 말풍선 배치 결과.



(e) 영화 *Before Sunrise* 장면의 말풍선 배치 결과.

Figure 17: 유명한 영화 장면들에 적용한 말풍선 배치 결과들

6 결론 및 향후과제

영화와 같은 비디오 영상을 자동화된 방법으로 만화책으로 변환하는 것은 상당히 흥미로운 주제이다. 이를 위해 카툰닝, 스피드라인, 말풍선 그리고 배경 텍스처 등과 같은 다양한 만화적 기법들을 자동으로 비디오 영상에 적용하는 방법들이 연구되고 있다. 본 논문에서는 여러 만화적 변환을 위한 기법들 중에서 비디오 영상에서 추출된 장면에 말풍선을 자동으로 배치하는 방법을 제안하였다. 먼저, 장면에서 말풍선들의 좋은 배치를 위한 일반적인 규칙 및 말풍선 배치의 최적화 문제를 정의하였다. 그리고 사용자의 상호작용을 최소화하면서 자동으로 말풍선을 배치하기 위해 특징점 라벨 배치 방법에 기반한 발견적인 알고리즘을 제안하였다. 다양한 실험 결과에서 보듯이, 제안한 말풍선 배치 방법은 읽기 순서에 따른 말풍선간의 관계 및 말풍선과 인물간의 관계를 고려하기 때문에 일반적으로 사람이 수작업으로 수행한 결과와 비슷한 결과를 도출하였다. 또한, 본 논문에서는 이전에 연구되었던 만화적 렌더링을 위한 기법들 [8]과 말풍선 자동 배치 방법을 단일 프로그램으로 구현한 CORVIS(Comics Rendering System for Video Stream) 시스템을 구현하였다.

그러나 비디오 영상을 자동 또는 반자동으로 한 권의 만화책으로 변환하기 위해서는 아직 많은 문제들이 남아 있다. 특히, 카툰닝(cartooning) 작업을 들 수 있다. 본 논문에서는 카툰닝 작업은 수작업 또는 이미 발표된 다른 도구를 이용하여 얻는 것을 가정하고 있다. 예를 들면, 그림 17(e)는 Mean-Shift 알고리즘에 기반한 EDISON 시스템 [13]을 이용하여 만화와 유사한 이미지를 생성한 것이다. 결과에서 알 수 있듯이, 비디오 영상보다 카툰닝된 이미지에 말풍선을 적용하는 것이 보다 효과적인 것을 알 수 있다. 따라서 비디오 영상으로부터 만화책을 생성하기 위한 보다 특화된 이미지 세그멘테이션 알고리즘 개발이 필요하다. 이러한 향후 과제들을 해결한 후, 좀 더 자동화된 버전의 CORVIS를 발표하고, 실제로 다양한 종류의 유명 한 영화들을 만화책으로 제작할 예정이다.

References

- [1] Scott McCloud. *Understanding comics: The invisible art*. Kitchen Sink Press, 1999.
- [2] Scott McCloud. *Reinventing comics: How imagination and technology are revolutionizing an art form*. Perennial, 2000.
- [3] Doug DeCarlo and Anthony Santella. Stylization and abstraction of photographs. *ACM Transactions on Graphics*, 21(3):769–776, 2002.
- [4] Aseem Agarwala, Aaron Hertzmann, David H. Salesin, and Steven M. Seitz. Keyframe-based tracking for rotoscoping and animation. In *Proceedings of SIGGRAPH 2004*, pages 584–591. ACM Press, 2004.
- [5] Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. *ACM Transactions on Graphics*, 23(3):574–583, 2004.
- [6] Maic Masuch, Stefan Schlechtweg, and Ronny Schulz. Speedlines: Depicting motion in motionless pictures. In

Proceedings of SIGGRAPH 1999, Computer Graphics Proceedings, Annual Conference Series, page 277. ACM, ACM Press / ACM SIGGRAPH, 1999.

- [7] Christian Hanl, Michael Haller, and Jeremiah Diephuis. Non-photorealistic rendering techniques for motion in computer games. *Computers in Entertainment*, 2(4):11, 2004.
- [8] W. I. Hwang et al. Cinema comics: Cartoon generation from video stream. In *Proceedings of GRAPP 2006*, pages 299–304, 2006.
- [9] David Kurlander, Tim Skelly, and David Salesin. Comic chat. In *Proceedings of SIGGRAPH 1996*, pages 225–236. ACM Press, 1996.
- [10] Jon Christensen, Joe Marks, and Stuart Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.
- [11] Shawn Edmondson, Jon Christensen, Joe Marks, and Stuart Shieber. A general cartographic labeling algorithm. *Cartographica*, 33(4):13–23, 1997.
- [12] Planetwide Games. Comic book creator, 2005.
- [13] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transaction Pattern Analysis Machine Intelligence*, 24(5):603–619, 2002.