

포즈 예측을 이용하는 자동 파노라마 영상 생성

정다운, 김만배

강원대학교 IT대학 컴퓨터정보통신공학과

manbae@kangwon.ac.kr

Automatic Panorama Image Generation Using Pose Estimation

Daun Jeong and Manbae Kim

Dept. of Computer and Communications Engineering, Kangwon National University

요약

본 논문에서는 하나의 카메라로 일정 영역에서 랜덤하게 촬영되어 얻어진 영상으로 파노라마 영상을 제작하는 방법을 제안한다. 실험 영상으로는 디지털카메라로 일정 영역을 랜덤하게 촬영한 여러 장의 영상을 사용한다. 여러 장의 영상에서 제어점이 될 특징 점들을 검출한 후, 유클리드의 좌표로 바꿔준다. 이 좌표들을 통해 각각의 제어 점에 가장 인접해 있는 좌표 4개를 추출한다. 이 인접 좌표들이 서로 다른 각각의 영상에서 매칭 되는 확률을 계산하여, 가장 높은 매칭 확률을 갖는 영상과 매칭 되는 pose값을 이용하여 각각의 영상을 회전 및 이동하여 매칭시킴으로써 촬영된 영상들의 파노라마 영상을 구현한다.

1. 서론

본 논문에서는 일정 영역을 촬영한 여러 장의 정지 영상을 이용해 파노라마 영상을 제작하는 기법을 제안한다 [1]. 입력받은 여러 장의 정지 영상에서 특징점으로 사용하기 위해 각각 영상의 코너 픽셀을 추출한다. 추출한 코너 픽셀을 각 영상의 제어점으로 사용한다. reference 영상의 제어점을 다른 영상으로 변환시켜 매칭하고 다른 영상의 제어점과 변환되어 매칭된 reference 영상의 제어점의 distance ϵ 내에 최대 매칭 되는 회전각도와 이동거리를 추출한다. 얻어진 transform pose (R, t)를 이용하여 reference 영상의 모든 제어점을 다른 영상으로 변환하여 매칭된 제어점으로 확률을 계산한다. 각각의 영상에서 확률이 가장 높은 영상을 매칭 영상으로 정하여 영상을 매칭시켜 파노라마 영상을 생성한다.

본 논문의 구성은 다음과 같다. 2절에서 제안 방법을 설명한다. 3절에서는 파노라마 영상 생성을 설명하며, 실험결과는 4절에서 보여준다. 마지막으로 5절에서 결론을 맺는다.

2. 제안 방법

여러 장의 입력 영상에서 파노라마 영상을 생성하기 위해 gray-scale 영상으로 변환한 후에 Harris corner detection을 통해 제어점으로 사용할 코너점을 생성한다. 먼저 하나의 제어점과 가장 인접한 4개의 제어점을 다른 영상의 제어점과 그 인접점들과 매칭시키면서 scale factor를 구하고, scale factor를 이용해 포즈 예측을 단계를 거쳐 transform pose (R, t)를 구한다. 제어점들을 변환시켜 매칭 되는 점들을 확인하고 확률을 구한 뒤, 각각의 이미지들의 제어점을 변환하여 매칭시키면서 가장 알맞은 이미지를 찾는다.

그림 1은 본 논문에서 제안하는 전체 블록도를 보여준다.

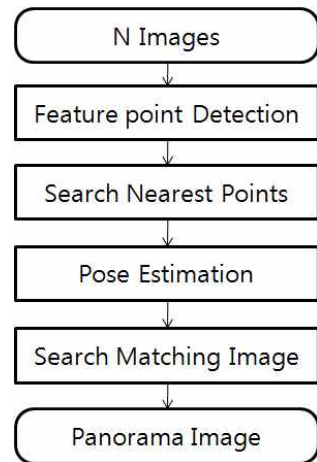


그림 1 파노라마 영상 생성의 블록도

3. 파노라마 영상 생성

영상에서 코너점은 직선과 더불어 영상분석을 위한 정보를 제공한다. 코너 검출을 통해서 영상 내에 존재하는 특정 객체의 형태와 위치에 대한 정확한 정보를 얻을 수 있다. 본 논문에서는 코너를 검출하기 위해 Harris corner detection을 이용하였다. Harris corner detection은 영상의 명암 값 변화에 의해 코너를 검출하는 알고리즘이다 [2]. 이 알고리즘은 2D gray-scale 영상에서 Δx 와 Δy 에 의해 움직여지는 영상의 patch인 W 가 있다고 한다. W 가 Δx 와 Δy 에 의해서 움직임에 따라 생기는 명암 값의 차이를 통해 얻는다.

그림 2는 입력영상을 gray scale로 변환 한 뒤 Harris corner detection을 통해 얻은 코너점을 표시한 영상이다.

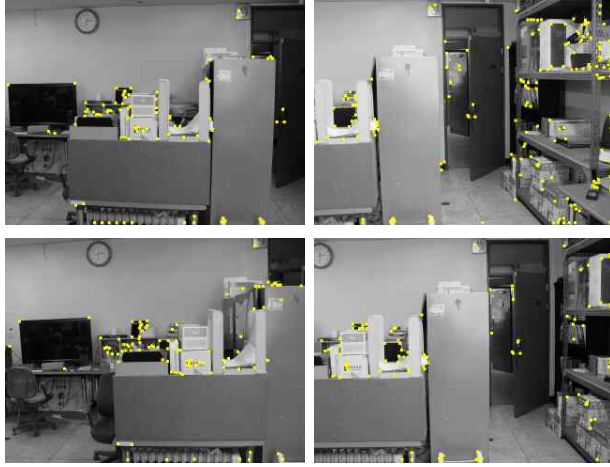


그림 2. 코너점 출력 영상

입력된 여러 장의 영상의 제어점을 모두 생성하면 제어점과 인접한 4개의 점을 선택하여 자신을 포함하지 않은 다른 영상들의 제어점과 인접한 4개의 점들과 비교해 각각의 scale factor를 식 (1)을 이용하여 구한다. 여기서 인접한 점 4개를 사용하기 때문에 N은 4이다.

$$D = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (1)$$

$$avg = \frac{\sum_{n=0}^N D}{N}$$

$$s.f = \frac{avg_n}{avg_m}$$

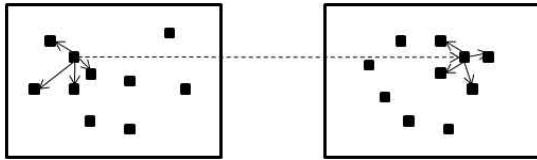


그림 3. 제어점과 인접한 4점의 비교

그림 3처럼 reference 영상 m과 다른 영상 n의 각각의 제어점과 인접점을 이용하여 scale factor를 구하고 이를 이용해 포즈 예측을 하여 transform pose (R, t)를 구한다. transform pose (R, t)를 이용하여 reference 영상 m의 모든 제어점을 변환해서 n영상으로 매칭 시키고 그림 4처럼 매칭된 좌표의 distance ϵ 내에 매칭 되는 제어점을 확인한다.

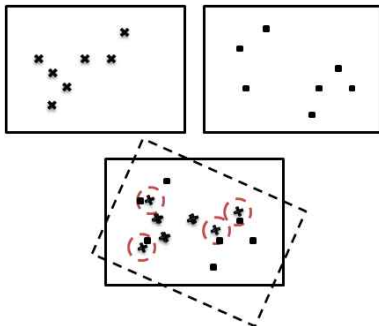


그림 4. 영상 간에 제어점 매칭

이때 매칭된 제어점을 식 2를 이용해서 매칭 확률을 구할 수 있다.

$$P_{mn} = \frac{n_n}{N_m} \quad (2)$$

N_m 는 reference 영상 m의 제어점 수이고 n_n 은 매칭된 영상 n에서 매칭된 제어점 수이다.

매칭 확률이 계산한 뒤, n영상의 다른 제어점과 인접점을 이용하여 reference 영상 m이 n영상으로 매칭 되는 다른 확률과 transform pose (R, t)를 구할 수 있고 m영상과 n영상에서의 포즈 예측이 끝나고 나면 reference 영상 m은 m과 n을 제외한 다른 영상에서 같은 작업을 하게 된다. 모든 영상에 대해 매칭 될 확률이 계산되면 매칭 될 확률이 가장 높은 영상과 transform pose (R, t)를 구할 수 있게 되며 다른 영상에서도 동일한 연산을 수행하여 각각의 확률을 계산하며, 이 연산 작업은 각각의 모든 이미지에서 확률이 가장 높은 영상을 찾을 때까지 반복한다.

다음은 포즈 예측의 단계를 보여준다. 두 영상의 제어점이 각각 x_n, y_n 이 주어지고, R, t가 있으면, residual error는 다음 식처럼 계산된다 [3][4][5].

$$\epsilon^2 = \sum_{n=1}^N w_n \| y_n - (Rx_n + t) \|^2 \quad (3)$$

여기서 가중치 평균은 다음과 같이 계산된다.

$$\bar{x} = \sum_{n=1}^N w_n x_n \quad \bar{y} = \sum_{n=1}^N w_n y_n \quad (4)$$

위의 식에서 계산된 가중치 평균값과 다음의 식을 이용하여 포즈를 예측 한다.

$$\bar{y} = R\bar{x} + t \quad (5)$$

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (6)$$

$$x_n = \begin{pmatrix} x_{1n} \\ x_{2n} \end{pmatrix}, \quad y_n = \begin{pmatrix} y_{1n} \\ y_{2n} \end{pmatrix} \quad (7)$$

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}, \quad \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix}$$

$$A = \sum_{n=1}^N w_n [(y_{n1} - \bar{y}_1)(x_{n1} - \bar{x}_1) + (y_{n2} - \bar{y}_2)(x_{n2} - \bar{x}_2)] \quad (8)$$

$$B = \sum_{n=1}^N w_n [(y_{n1} - \bar{y}_1)(x_{n2} - \bar{x}_2) + (y_{n2} - \bar{y}_2)(x_{n1} - \bar{x}_1)]$$

그러므로

$$\cos \theta = \frac{-A}{\sqrt{A^2 + B^2}} \quad \sin \theta = \frac{B}{\sqrt{A^2 + B^2}} \quad (9)$$

$$\cos \theta = \frac{A}{\sqrt{A^2 + B^2}} \quad \sin \theta = \frac{-B}{\sqrt{A^2 + B^2}}$$

위와 같은 포즈 예측 연산 후 입력된 각각의 영상마다 매칭 되는 가장 높은 확률의 영상을 찾게 되면 파노라마 영상을 생성하게 되는데, N장의 입력 영상과 각 매칭 영상 및 transform pose (R, t)가 주어지면, 다음과 같은 단계를 수행한다.

- 1) base image n* 선택한다.
- 2) base image n*에 매칭 되는 영상들을 transform pose

(R, t)를 이용하여 변환하여 매칭 한다.

- 3) 2)에서 처리된 영상들로 매칭 되는 영상을 search한 뒤 각 매칭 영상들을 다음 식을 이용하여 변환한다.
- 4) 이전에 매칭된 영상의 transform pose (θ_1, t_1)이고 현재 매칭될 transform pose (θ_2, t_2)이면 변환 pose 값은 다음과 같이 계산된다.

$$\begin{aligned} \theta &= \theta_1 + \theta_2 \\ \vec{t} &= \vec{t}_1 + \vec{t}_2 \end{aligned}$$

이 값들을 이용하여 현재 영상을 base image coordinate로 변환한다.

- 5) go to step 3.

4. 실험 결과

다음의 그림 4는 입력된 4장의 테스트 영상에서 제어점으로 사용될 코너점 검출과 각 코너점의 인접점을 통한 가중치 계산으로 얻은 매칭 변수에 의해 변환되어 가장 높은 확률을 가지는 영상을 매칭 시켜 생성한 파노라마 영상이고, 표 1은 각각의 영상의 매칭영상과 얻어진 transform pose (R, t)와 확률을 보여준다.

reference image와 matching image의 숫자는 각각의 영상을 뜻하며 transform pose (R, t)가 θ 와 t이며 t는 x좌표와 y좌표를 나타낸다. 표 1에서는 reference image 0번이 1번 image로 matching되며 매칭 되는 각도는 -1.17이고 x축으로 -36.04만큼, y축으로 -6.13만큼 움직이며 매칭확률은 0.48임을 의미한다. 이때 이미 매칭 패어가 된 이미지들은 다시 연산하지 않기 때문에 reference image 1번에서는 0번 image를 제외한 다른 이미지들과의 매칭 확률을 계산하게 된다.



그림 5. 가장 높은 확률로 얻어진 파노라마 영상

표 1. 각각 영상에서 매칭 되는 확률과 transform pose

reference image	matching image	θ (deg)	t	확률
0	1	-1.17	(-36.04, -6.13)	0.48
0	2	-0.69	(-184.47, -3.64)	0.34
0	3	-63.03	(214.63, 100.99)	0.43
1	2	3.05	(-209.72, -0.76)	0.36
1	3	-33.08	(288.57, 122.50)	0.38
2	0	0.69	(184.42, 5.86)	0.31
2	1	-3.05	(209.46, -10.42)	0.34
2	3	0.59	(-104.40, 0.07)	0.65
3	0	64.04	(207.06, -173.10)	0.22

5. 결론

본 논문에서는 일정 영역을 촬영한 여러 장의 정지 영상을 포즈에 측기법을 통하여 파노라마 영상을 생성하는 기법을 제안하였다. 영상의 특성과 제어점으로 사용한 코너의 특성 때문에 생성된 파노라마 영상이 완벽한 결과를 보여주지 못하고 있고, 이는 향후 연구에서 해결할 계획이다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구 결과로 수행되었습니다.(NIPA-2010-(CI090-1011-0003))

참고 논문

- [1] Brown, M. and Lowe, D.G. "Recognising Panoramas" Ninth IEEE International Conference on Computer Vision, 2003, pp: 1218-1225 Vol.2
- [2] C. Harris and M. Stephen, "A combined corner and edge detection," Proc. of 4th ALVEY vision conference, 1988.
- [3] P. J. Huber, *Robust Statistics*, NewYork: JohnWiley, 1981.
- [4] R. J. Hogg, *An Introduction to Robust Estimation*,
- [5] R. Haralick, H. Joo, C-N Lee, X. Zjuang, V. Vaidya, M. Kim, "Post estimation from corresponding data," IEEE Trans. On Systems, Man, and Cybernetics, Vol. 19, No. 6, Nov/Dec, 1989.