

무선 센서 네트워크 영상을 위한 모자이크 기법

An Image Mosaic Technique for Images Transmitted by Wireless Sensor Networks

전 상 은*, 어 진 우**
Sangeun Jun*, Jinwoo Eo**

요 약

무선 센서 네트워크(WSN)은 비교적 대역폭이 좁고 데이터 저장 메모리 용량에 한계가 있다. 인접한 센서노드들로부터 전송되는 영상을 모자이크하여 하나의 파노라믹 영상으로 만들어 저장하면 필요한 저장 영상의 수도 감소하고 보다 크고 넓은 시야각의 영상을 얻을 수 있어 센서노드 주변의 환경을 이해하는 데 더욱 효율적일 것이다. 대부분의 WSN이 탐색용으로 사용되기 때문에 실시간 처리가 가능하도록 영상획득 주기가 충분히 작아야 한다. 본 논문에서는 센서 네트워크에서의 노드들의 위치를 미리 알 수 있으며 고정되어 있다는 점을 이용하여 빠른 처리가 가능한 모자이크 알고리즘을 제안한다. 제안된 알고리즘은 기존에 방법에 비해 계산량이 매우 적으므로 빠르게 영상 모자이크를 수행할 수 있어 빠른 주기로 여러 개의 센서노드에서 들어오는 영상을 처리할 수 있다. 실험 결과 제안하는 알고리즘이 빠른 처리 능력을 보이면서도 모자이크된 화질이 기존의 방법에 비해 크게 떨어지지 않음을 보여주었다.

Abstract

Since wireless sensor networks (WSN) have relatively narrow bandwidth and have limited memory space. Mosaic by inlaying images transmitted by adjacent sensors can provide wider field of view and smaller storage memory. Most WSN are used for surveillance purpose, image acquisition period should be sufficiently short, so that mosaic algorithm has to be run in real time. Proposed algorithm is derived by using the fact that position of sensor nodes are fixed and known. Transformation matrix can be calculated by using distance between sensor nodes and distance between sensor nodes and predefined object. Simulation result shows that proposed algorithm provides very short processing time whereas it preserves image quality.

Key words : WSN, USN, image mosaic, image stitching, panoramic image

I 서론

센서 네트워크는 빛, 온도, 습도, 압력과 같은 데이터를 센서노드(Sensor Node)를 통해 수집하여 베이스 노드(Base Node) 혹은 게이트웨이로 전달함으로써 센서노드 주변의 정보 획득을 위한 기반을 제공한다 [1]. 이러한 무선 센서 네트워크는 대상 지역에 대한 충분한 환경정보를 제공하지만, 대상 지역에 존재하는 객체에 대한 정보를 제공하지 못한다. 대상 지역의 환경정보는 존재하는 객체에 따라 의미가 달라지

* 檀國大學校 電子컴퓨터工學部
(School of Electronics and Computer Engineering, Dankook University)

★ 교신저자 (Corresponding author)

※ 감사의 글 (Acknowledgment)

이 연구는 2006학년도 단국대학교 대학 연구비의 지원으로 연구되었음

接受日:2007年 12月 2日, 修正完了日: 2007年 12月 27日

기 때문에 센서 네트워크가 제공하는 정보는 주변 상황을 인지하기 위한 정보로서 충분하지 않다[2]. 이러한 문제를 해결하기 위해 센서 네트워크에 카메라를 설치하여 영상을 전송 받을 수 있도록 센서노드를 구성하였다. 이렇게 만들어진 센서 네트워크는 일정 주기로 다수의 센서노드에서 전송되는 영상의 수가 너무 많고, 이러한 전송된 영상을 통해서 센서노드들 간의 관계를 파악하기가 힘들다. 그러므로 전송된 각각의 영상을 효율적으로 관리하기 위해 영상 모자이크가 필요하다.

영상 모자이크 방법에는 크게 화소 기반 모자이크 제작 방법과 특징 기반 모자이크 제작 방법으로 나뉜다[3]. 일반적으로 화소에 기반한 방법은 양쪽 영상에 존재하는 대응되는 화소들에 대해 그 차이가 최소가 되도록 하는 위치를 반복적으로 찾는다. 화소 기반 방법은 정확하지만, 계산상 복잡도가 상당히 높다. 반면, 특징 기반 모자이크 방법에서는 화소 기반 방법에 비해 계산상의 복잡도가 상대적으로 낮다는 장점이 있지만, 이를 위해서는 적절한 특징점을 찾아야 하는 문제점이 있다[4][5].

본 논문에서 제안하는 방법은 특징 기반 모자이크 방법에서 센서노드를 통해 수집된 대상물체의 기하학적인 위치 정보를 이용하여 영상에서의 특징점의 위치를 구하였다. 그러므로 특징점을 찾기 위한 알고리즘이 필요없게 되어 기존의 방법들보다 변환 행렬을 구하는 시간을 크게 단축할 수 있다.

본 논문은 2장에서 기존의 영상 모자이크의 방법에 대해 기술하고, 3장에서 새로운 영상 모자이크 방법을 제안한다. 4장에서 실험 및 분석 결과를 기술하고 5장에서 결론 및 후후 과제에 대해 언급한다.

II 관련 연구

영상 모자이크는 고해상도의 대규모 영상을 획득하는 영상 기반 렌더링 기법 중의 하나로서, 가상현실, 대화형 멀티 플레이어 게임, 가시화 분야 등과 같은 많은 응용분야에 사용되고 있다[6]. 여러 카메라로부터 획득되는 영상들을 하나의 커다란 모자이크 영상으로 정합하고 합성함으로써 높은 해상도를 가진 영상을 완성시킬 수 있다. 현재 사용되고 있는 영상 모자이크 방법으로는 화소 기반 방법과 특징 기반 방법으로 나눌 수 있다.

2.1 화소 기반 방법

화소 기반 모자이크는 두 영상의 화소를 직접 비교하여 오차가 최소가 되는 값을 반복적으로 찾는 방

법이다. 일반적으로 가장 많이 사용되는 화소 기반 방법은 Szeliski가 평면 투영 변환을 계산하기 위하여 두 영상 사이의 밝기 값의 오차의 제곱을 최소화하는 변환 계수를 비선형 최소 자승 오차법(nonlinear least square method)인 Levenberg-Marquardt(LM)를 이용한 방법이다[6][7]. 여기서 중요한 것은 변환 행렬 M 을 구하는 것이다.

2차원 좌표계의 임의의 점을 $\mathbf{x}=(x,y,1)$ 이라고 할 때 2차원 투영변환에 의한 새로운 점의 좌표 \mathbf{x}' 는 다음과 같이 계산된다.

$$\mathbf{x}' = M\mathbf{x} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

여기서 M 은 변환행렬, M 의 요소 m_0, m_1, m_3, m_4 는 회전과 크기 요소, m_6, m_7 은 이동요소, m_2, m_5 는 비례변환요소를 나타낸다.

변환행렬 M 을 구하는 것이 필요한 데 그 중에서 LM 방법을 이용하여 오차를 최소화하는 방법은 증첩 영역에서 최소가 되는 오차 e 를 구해 새로운 변환 행렬 M 을 결정하는 방법으로 미지의 운동 요소 $\{m_0, \dots, m_7\}$ 에 대한 e 의 편미분 값을 이용한다.

$$e_i = I'(x'_i, y'_i) - I(x_i, y_i) \quad (2)$$

e_i 는 (x_i, y_i) 화소에서의 밝기 차이이다. 위의 식 (2)에서 LM 방법을 적용하여 평면 투영변환계수를 구하는 방법에 대하여 설명한다. 먼저 미지수인 변환 행렬 계수 m_i , ($i = 0, \dots, 7$)에 대한 편미분을 구한다.

편미분을 이용하여 변환 행렬 계수 m_i 에 대한 이차 편미분 행렬인 Hessian 행렬 A 와 가중 그래디언트 벡터 \mathbf{b} 를 구하여 식 (3)과 같이 변환 행렬 계수를 반복적으로 구한다.

$$m^{t+1} = m^t + \Delta m \quad (3)$$

여기서 Δm 은 다음과 같다.

$$\Delta m = (A + \lambda I)^{-1} \mathbf{b}$$

$$A_{k \times l} = a_{kl} = \sum_i \frac{\partial e_i}{\partial m_k} \frac{\partial e_i}{\partial m_l}$$

$$\mathbf{b} = \mathbf{b}_k = -2 \sum_i e_i \frac{\partial e_i}{\partial m_k}$$

i 는 1부터 4까지의 대응점들, k 와 l 은 0부터 7까지의 변환행렬 계수를 나타내는 첨자이다[8].

이 결과를 이용하여 LM 알고리즘의 변위 계산식 $\Delta m = (A + \lambda I)^{-1} \mathbf{b}$ 에 대입하여 운동 변수 Δm 을 새로 계산하는 것이 가능하다. 여기에서 λ 는 반복 계산에 대한 안정화 변수이다. 이 값은 식(2)에서 오차를 측정했을 때 이전 단계에서 최소화를 수행한 것보다 예러가 증가하면 그 값을 감소시키고, 감소하면 증가시켜 새로운 운동변수 Δm 을 계산해 최적의 변환행렬 M 을 구한다.

2.2 특징 기반 모자이크 방법

특징 기반 모자이크 방법에서는 기준영상과 영상 모자이크를 할 대상 영상에서 추출된 특징점들을 토대로 상관 관계를 계산하여 변환 행렬을 만든다. 특징 기반 모자이크 방법 중 Yao Li와 Lizhuang Ma의 'A Fast and Robust Image Stitching Algorithm'을 이용하여 변환행렬을 구하였다[9].

특징점 추출에 이용될 몇 개의 함수를 정의한다. 우선, 2D 가우시안 스무딩 함수(2D Gaussian smoothing function)는 다음과 같다.

$$G^\sigma(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (4)$$

여기서 σ 는 표준 편차이다. x 와 y 축으로 $G^\sigma(x, y)$ 에 의해 스무딩된 영상 $I(x, y)$ 의 그래디언트는 각각 다음과 같이 정의된다.

$$\begin{aligned} I'_x(x, y) &= I * G'_x(x, y) \\ I'_y(x, y) &= I * G'_y(x, y) \end{aligned} \quad (5)$$

여기서 $*$ 는 컨벌루션 연산자를 의미한다. 이 때, $I(x, y)$ 의 그래디언트 벡터의 절대값은 다음과 같다.

$$|\nabla I^\sigma(x, y)| = \sqrt{|I'_x(x, y)|^2 + |I'_y(x, y)|^2} \quad (6)$$

만약 $|\nabla I^\sigma(x, y)|$ 의 모든 지역 극대값(local maximum)이 문턱치(threshold) 값보다 크다면 크기 비교(scale scanning)을 통해 $I(x, y)$ 의 모든 에지(edge) 값들이 검출되어 진다. 여기서 특징점은 로컬 지역에서 가장 큰 값을 갖는 점을 특징점으로 하였다.

변환 행렬을 생성하기 위해서는 두 영상간에 대응되는 점의 값을 정하여야 한다. 첫 번째 영상 I 에서의 점 \mathbf{p}_i 에 상응하는 두 번째 영상 I' 에서의 한 점 \mathbf{p}'_i 를 찾기 위해 밝기에 대한 상호 상관(cross correlation)을 이용한다. 매칭되는 점을 찾기 위해 검색할 영역인 사각 탐색 창(rectangular search

window) Γ 으로 하고, 사각 상관 마스크(rectangular correlation mask)는 A 라고 정의한다.

Γ 안의 각각의 화소에 대해 마스크 A 는 두 개의 영상에 대해 각각을 정의한다. 하나는 마스크의 중심이 \mathbf{p}_i 에 다른 하나의 마스크의 중심은 \mathbf{p}'_i 에 위치한다. \mathbf{p}_i 는 영상 I 에서 임의적으로 선택되고, 두 번째 영상의 검색창(search window) 내에서 \mathbf{p}'_i 를 찾는다.

상관계수는 단지 검색창(search window) 내에서 가장 큰 값, 즉 특징점이 위치한 점에서만 계산하였다. 그러므로 검색창(search window) 내의 대부분에 점들은 계산에서 제외된다.

상관계수 값은 -1에서 1 사이의 실수값이다. $(\mathbf{p}_i, \mathbf{p}'_i)$ 의 쌍을 찾는 방법은 검색창(search window) 안을 full search 방법으로 해서 찾는다. 이렇게 하여 3개의 매칭되는 점 $(\mathbf{p}_i, \mathbf{p}'_i)_{i=1, \dots, 3}$ 를 구한다[9].

이렇게 구해진 3개의 매칭되는 쌍을 이용하여 구한 변환행렬 M 을 이용하여 두 개의 영상을 같은 2차원의 영상 좌표로 만들 수 있다. 3개의 \mathbf{p}_i 중 중간에 위치한 점을 기준으로 두 영상을 스티칭(stitching) 하였다. 본 논문에서는 위의 특징 기반 모자이크 방법을 제안하는 알고리즘과 동일한 환경 하에서 구현하여 제안하는 모자이크 알고리즘과 비교하고자 한다.

III 제안하는 영상 모자이크 기법

실험을 위한 영상 모델을 표준 pin-hole 카메라 영상 모델로 하였다. 그리고 카메라의 중심축을 z 축으로 하고, $z=1$ 인 면을 retina plane으로 하였다. 공간상의 점 $\mathbf{P}=(X, Y, Z)$ 는 retina point $\mathbf{p}=(x, y, z)$ 로 투사된다.

$$\begin{cases} x = X/Z \\ y = Y/Z \end{cases} \quad (7)$$

\mathbf{p}' 를 \mathbf{p} 로 바꾸는 과정은 아래와 같은 변환 식으로 표현된다.

$$\mathbf{p} = R \cdot (\mathbf{P} - \mathbf{t}) \quad (8)$$

여기서 R 은 변환 행렬(rotation matrix)이고 \mathbf{t} 는 3D 전이 벡터(translation vector)이다. $retina(\cdot)$ 는 식 (7)을 이용하여 위치 값을 retina plane값으로 바꾼다. Retina point \mathbf{p} 와 pixel 좌표 (u, v) 와의 관계는 다음과 같이 정의할 수 있다[10].

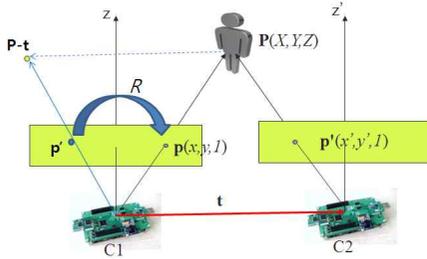


Fig. 1. Standard pin-hole camera imaging model
그림 1. 표준 핀홀 카메라 영상 모델

$$\begin{cases} x = (u - u_0)/f_u \\ y = (v - v_0)/f_v \\ z = 1 \end{cases} \quad (9)$$

여기서 (u, v) 는 \mathbf{p} 의 화소 좌표이고, f_u 와 f_v 는 각각 u 와 v 축의 scale factor이다. 그리고 (u_0, v_0) 는 투사된 영상의 중심점이다. 그러므로 영상은 다음과 같은 행렬식으로 쓸 수 있다.

$$\mathbf{i} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \cdot \text{retina}(\mathbf{P}) \quad (10)$$

여기서 \mathbf{i} 는 영상 좌표 $(u, v, 1)$ 에 해당하는 좌표를 갖는 벡터이다. 그리고 K 는 카메라 고유 행렬(camera intrinsic matrix)이며 다음과 같다.

$$K = \begin{pmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (11)$$

두 개의 센서노드 C1과 C2에 대해 모자이크를 하면, C1은 기준이 되는 센서노드이므로, 변환행렬은 단위행렬인 I (Identity Matrix)가 되고, 전이벡터 값은 0이 된다. 따라서 다음과 같이 쓸 수 있다[11].

$$\begin{aligned} \mathbf{t} &= 0 \\ R &= I \end{aligned} \quad (12)$$

이 때 구해진 좌표를 \mathbf{i} 이라고 하면 수식 (13)로 나타나게 된다.

$$\mathbf{i} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \cdot R \cdot \text{retina}(\mathbf{P} - \mathbf{t}) \quad (13)$$

C2의 화소 좌표 \mathbf{i} 는 C1의 영상과 모자이크 할 것이

므로 \mathbf{i} 과 같아야 하고, 센서노드가 설치된 위치는 고정되어 있으므로 \mathbf{t} 값을 측정을 통하여 알 수 있다. 그러므로 C2의 변환행렬 R_2 는 수식 (14)을 통하여 구하여 진다.

$$R_2 = K^{-1} \mathbf{I}_1 \cdot (\text{retina}(\mathbf{M}_1))^{-1} \quad (14)$$

여기서,

$$\begin{aligned} \mathbf{I}_1 &= [\mathbf{i}_1 \ \mathbf{i}_2 \ \mathbf{i}_3] \\ \mathbf{M}_1 &= [\mathbf{P}_1 - \mathbf{t} \ \mathbf{P}_2 - \mathbf{t} \ \mathbf{P}_3 - \mathbf{t}] \end{aligned}$$

\mathbf{I}_1 은 C1에서의 3개의 영상의 특징점을 모은 행렬이고, \mathbf{M}_1 은 C1에서의 3개의 위치벡터에 C1에서C2까지의 벡터인 \mathbf{t} 를 뺀 행렬이다.

위의 과정을 반복적으로 실행하여 n 개의 센서노드에 대해서 각각의 변환행렬 R_1 에서 R_n 까지는 다음과 같이 된다.

$$R_n = K^{-1} \mathbf{I}_{n-1} \cdot (\text{retina}(\mathbf{M}_{n-1}))^{-1} \quad (15)$$

식 (15)을 통해 구해진 R_1 에서 R_n 을 이용하여 n 개의 영상이 스티칭 된 모자이크 영상을 구할 수 있다. 이 때, 각각의 센서노드의 변환 행렬 R 을 구하기 위해서는 두 영상 간에 매칭되는 3개 이상의 \mathbf{P} 값을 알아야 한다.

식 (15)을 통해 구해진 R_1 에서 R_n 을 이용하여 n 개의 영상이 스티칭된 모자이크 영상을 구한다. 변환행렬은 대상이 되는 영상의 화소 좌표를 변환행렬에 곱하여 기준이 되는 노드의 화소 좌표값으로 바꾸어 두 개의 영상을 같은 좌표상에 표현한다. 이때 두 영상이 중첩되는 위치의 중앙을 기준으로 좌우로 영상이 스티칭을 하여 하나의 파노라마 영상을 만든다.

그림 2는 제안된 영상 모자이크 과정이다.

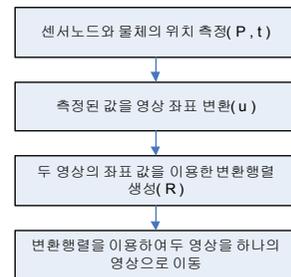


Fig. 2. Proposed image mosaic algorithm
그림 2. 제안하는 영상 모자이크 알고리즘

IV 실험 및 분석

본 논문에서는 제안하는 모자이크 영상 생성 알고리즘은 Intel core2duo 1.83GHz와 1GB의 메모리를 가진 PC에서 Matlab 7.0.1버전을 이용하여 구현하였다. 실험에 사용된 영상은 GPM430 센서노드에 C328 CMOS 카메라 모듈을 통해서 얻어진 320x240 JPEG 영상을 사용하였다.

그림 3은 각각의 센서노드를 같은 높이의 동일 선상에 1m 간격으로 배치한 후에 무선 센서 네트워크를 통하여 획득한 영상을 보여주고 있다.

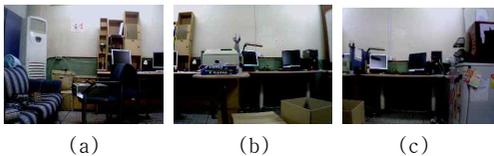


Fig. 3. Original images from three sensor nodes
그림 3. 세 개의 센서노드를 통해 획득한 원영상

3개의 영상을 모자이크를 하는 방법은 그림 3의 (b)를 기준으로 그림 3의 (c)를 영상 모자이크를 하고 다시 합성된 영상과 그림 3의 (a)의 영상을 모자이크를 하였다. 이 때, 변환 행렬을 생성하기 위해 그림 3의 (a)와 (b) 사이와 그림 3의 (b)와 (c) 사이의 공간에 3개의 물체의 좌표를 측정하였고 이 값을 제안한 방법을 이용하여 변환행렬을 생성하였다. 영상합병 방법은 3개의 물체의 좌표 중 중간에 위치한 좌표를 기준으로 하여 좌우로 영상을 스티칭하였다.



Fig. 4. Mosaic image using proposed technique
그림 4. 제안한 기법으로 모자이크된 영상

그림 4는 그림3의 (a)와 (b),(c)가 모자이크된 것을 보여주고 있다. 그림 4를 보면 좌측의 영상을 기준으로 하여 우측의 영상이 변환되어 접합된 것을 알 수 있다.

표 1과 표 2는 각각의 모자이크 방법에 대한 비교를 나타내는 표이다. 3가지의 모자이크 방법은 변환행렬을 구하는 부분이 가장 중요하면서도 3가지 방법의 알고리즘에서 차이가 나는 부분이다. 그래서 표 1은 3가지 모자이크 방법에 따른 변환행렬 생성 알고

리즘의 프로세싱 과정에 소요되는 시간을 비교하였다.

Table 1. Comparison of transformation matrix generation time among various mosaic techniques

표 1. 각 모자이크 방법의 변환행렬 생성 시간 비교 (단위:초)

	화소	특징	제안한 방법
초기행렬생성	0.172	-	-
특징점 추출	-	42	0.01
특징점 매칭	-	7.4	-
변환행렬생성	109	0.001	0.001
전체	109.2	49.4	0.011

화소 기반 방법은 오차가 최소가 되는 값을 찾기 위해 연산이 계속적으로 반복되므로 매우 많은 연산 시간을 소비하는 것을 보여준다. 특징 기반 모자이크 방법은 화소 기반 방법에 비해 상대적으로 적은 연산 수이지만 여전히 영상내에서 특징점을 찾기 위해서는 여전히 많은 연산 시간이 필요함을 표에서 확인할 수 있다. 제안된 방법은 변환행렬을 생성하기 위해서 특징점을 찾기 위한 알고리즘이 없고 단지 단순한 행렬 연산만을 요하므로 다른 방법에 비해 처리시간이 매우 적게 사용되는 것을 알 수 있다.

표 2는 성능 평가를 위해서 중첩되는 부분에서의 평균 제곱오차(mean squared error) 값을 비교하였다. 평균 제곱오차는 변환에 의해 중첩되는 영역에서의 화소의 밝기 값 사이의 오차의 제곱을 중첩되는 영역의 화소로 나눈 것이다. 평균 제곱오차는 작을수록 중첩되는 부분의 유사성이 높은 것을 보여준다.

Table 2. MSE comparison among various mosaic techniques
표 2. 각 모자이크 방법의 MSE값 비교

	화소	특징	제안한 방법
MSE	1185	1214	1371

표 2를 보면 제안하는 방법이 다른 방법에 비해 MSE 값이 높다. 그래서 두 영상을 중첩시켜 모자이크한 영상인 그림 5의 중첩부분은 다소 부자연스러운 영상을 보인다. 하지만 그림 5의 스티칭 부분과 같이 경계를 중심으로 좌우로 영상을 스티칭하였을 때는 각각의 방법 간의 큰 차이가 보이지 않는 것을 알 수 있다. 그리고 본 논문에서는 영상을 모자이크할 때 두 영상을 스티칭하여 모자이크하였으므로 큰 차이를 보이지 않음을 알 수 있다.

제안한 모자이크 방법을 거리에 따라 MSE 값이 변화하는 것을 실험하였다. 이 때, 스티칭을 할 기준

이 되는 물체와 z축 방향으로 4m 거리를 두고 구한 모자이크 영상이 가장 낮은 MSE 값을 가졌다. 따라서 변환행렬을 구할 때 특징점을 구하기 위한 물체의 위치를 4m 거리에 있는 물체를 이용하여 변환행렬을 구하면 상대적으로 좋은 모자이크 영상을 생성할 수 있을 것이다.

	중첩	스티칭
화소		
특징		
제안		

Fig. 5. Comparison among various mosaic techniques
 그림 5. 기법별 모자이크 결과 비교

V 결론

본 논문에서는 각각의 센서노드에 카메라를 설치하고 이 카메라를 통해 획득된 영상과 위치 정보를 서버로 전송하는 시스템 상에서 전송되어진 영상을 좀 더 효율적으로 관리하기 위해 영상들을 모자이크 하였다. 모자이크를 위해서 필요한 특징점을 센서노드의 위치정보와 목표 물체의 위치를 측정하여 구하였다. 이를 이용하여 모자이크된 파노라마영상은 저장되는 영상의 수도 감소하고, 주변의 환경을 이해하는데 보다 효율적으로 사용될 수 있을 것이다. 실험결과 제안하는 알고리즘이 빠른 처리 능력을 보이면서도 모자이크된 화질이 기존의 방법에 비해 크게 떨어지지 않음을 보여주었다.

추후에는 카메라가 늘어남에 따라 계산시간이 늘어나는 문제를 해결하기 위한 병렬처리에 대한 고찰이 필요하다.

참고문헌

[1] 특허청, 2004 신기술 동향 조사 보고서 - 유비쿼터스 컴퓨팅 기술, 전기/전자 분야
 [2] 고경철, "RFID 기반 유비쿼터스 센서 네트워크에서의 지능적 상황인지 지원", 한국컴퓨터종합학술대회, 2005 논문집, 2005
 [3] R. Garcia, X. Cufí, X. Muñoz, L. Pacheco, J. Batlle, "An Image Mosaicking Method based on the Integration of

Grey Level and Textural Features," IX Simposium Nacional de Reconocimiento de Formas y Analisis de Imágenes (SNRFAl), Benicassim, Castellón, 2001
 [4] R.Szeliski, "image mosaicing for tele-reality application", Proc. Of the IEEE Workshop on Applications of Computer Vision, p.44-53, 1994
 [5] 김세환, "다수의 3차원 카메라를 이용한 영상 기반 3차원 모자이크", 한국통신학회, pp. 58-61, 2002
 [6] H.-Y. Shum and R. Szeliski, "Panoramic image mosaics," Technical Report MSR-TR-97-23, Microsoft Research, 1997.
 [7] R. Szeliski, "Video mosaics for virtual environments," IEEE Computer Graphics and Applications, pp.22-30, March, 1996.
 [8] 정민영, "Direct X를 이용한 실시간 영상 모자이크", 정보처리학회 논문지 제 10-B권 제7호, pp. 803-810, 2003
 [9] Yao Li, "A Fast and Robust Image Stitching Algorithm", Proceeding of the 6th World Congress on Intelligent Control and Automation, June, 2006
 [10] Changming Sun, "2-point linear algorithm for camera translation vector estimation with known rotation", Robotica(2000) Volume 18, Cambridge Univ. Press,p.557, 2000
 [11] Bo Hu, "Interactive Indoor Scene Reconstruction from Image Mosaics Using Cuboid Structure", Proceeding of the Workshop on Motion and Video Computing (MOTION' 02), 2002

저 자 소 개

전 상 은 (학생회원)



2006년: 단국대학교
 전기전자컴퓨터공학부 학사
 2006년 - 현재: 단국대학교
 전자컴퓨터공학과 (석사과정)
 2006년 - 현재: 전자부품연구원
 (KETI) 위촉연구원
 <주관심분야> 영상처리,
 무선센서네트워크

어 진 우 (정회원)



1979년: 서울대학교
 전기공학과 학사
 1988년: Oregon State University
 전기 및 컴퓨터공학 석사
 1992년: Oregon State University
 전기 및 컴퓨터공학 박사
 1979년 - 1986년: 국방과학연구소 연구원
 1992년 - 현재: 단국대학교 전자컴퓨터공학부 교수
 <주관심분야> 영상신호처리, 비디오 코딩, 의료영상해석