

Localization1

6.localization.

로봇이나 자율주행차량이 주어진 공간(Map)의 어디에 있는지를 예측하는 방법을 말한다.

PDF(probability density function)

추측한 로봇의 위치를 나타낸다.

xy 좌표가 가리키는 함수값은 로봇이 그 위치에 있을 가능성을 나타낸다.

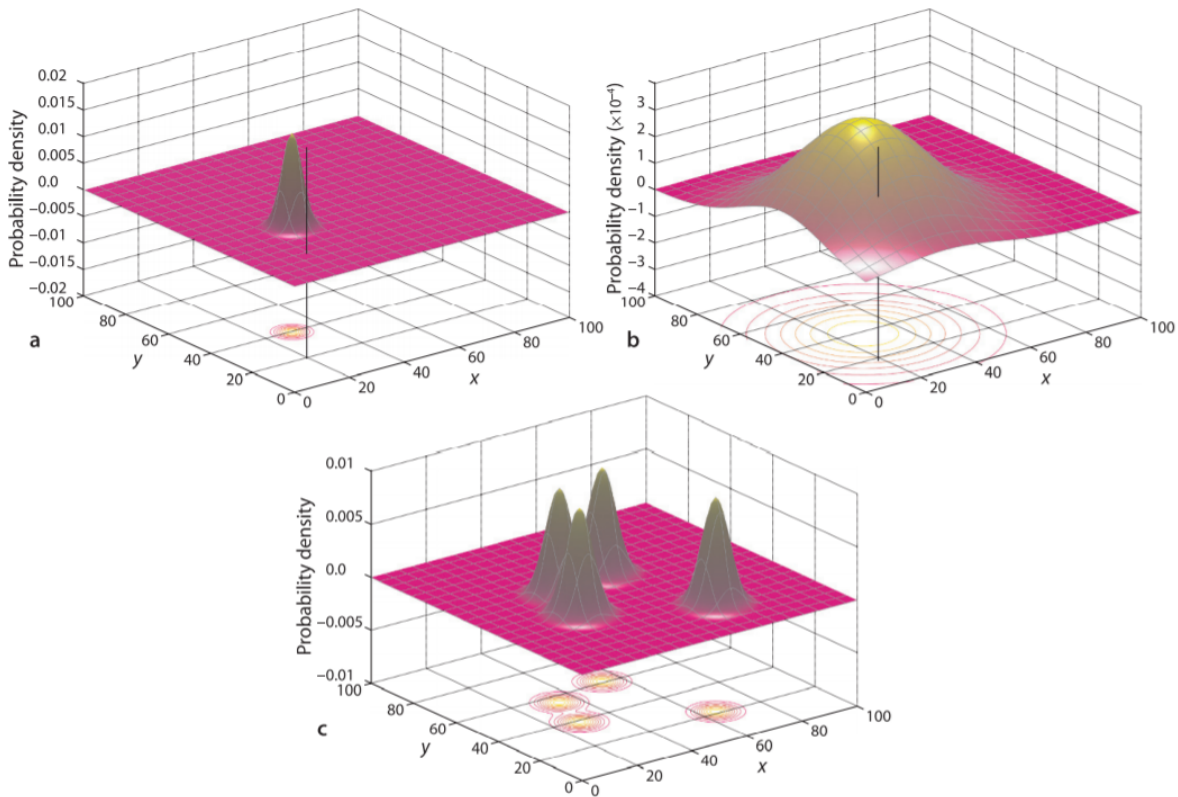


그림 a는 로봇의 위치에 대한 불확실성 $\sigma = 1$ 이다.

그림 b는 로봇의 위치에 대한 불확실성 $\sigma = 20$ 이다.

그림 c는 로봇의 위치에 대한 불확실성 $\sigma = 1$ 이고 로봇이 있을 수 있는 곳은 4군데 이다.

6.1 Dead Reckoning

추측방법은 이미 아는 출발위치에서 침로와 속력을 계산하여 자신의 위치를 추측하며 항해하는 방법이다.

추정위치는 해류, 바람 등을 비롯한 그밖의 인자가 미치는 영향을 보정한 추측항법위치이다. 추측항법은 시간과 거리가 증가함에 따라 불확실해진다. 따라서 지속적인 위치 보정이 필요하다. 차량의 경우 자이로 스코프 또는 왼쪽,오른쪽 바퀴의 각속도 차이를 이용해 방향변경 값을 구하고 이동 방향은 전자 나침반을 사용하여 측정한다.

<https://www.youtube.com/watch?v=feKF-4Osl0U>

6.1.1 Modeling the Vehicle

k가 time step 일때 $\delta\langle k \rangle = (\delta d, \delta\theta)$ 는 이동한 미소 위치변화 값과 미소 방향전환 값이다. 초기 pose를 2차원으로 나타내면 다음과 같다.

$$\xi\langle k \rangle \sim \begin{pmatrix} \cos\theta\langle k \rangle & -\sin\theta\langle k \rangle & x\langle k \rangle \\ \sin\theta\langle k \rangle & \cos\theta\langle k \rangle & y\langle k \rangle \\ 0 & 0 & 1 \end{pmatrix}$$

(좌표계의 pose를 ξ 로 나타낸다.)

차량을 δd 만큼 전진하고, $\delta\theta$ 만큼 회전하면 다음과 같은 배열로 나타낼 수 있다.

$$\begin{aligned} \xi\langle k+1 \rangle &\sim \begin{pmatrix} \cos\theta\langle k \rangle & -\sin\theta\langle k \rangle & x\langle k \rangle \\ \sin\theta\langle k \rangle & \cos\theta\langle k \rangle & y\langle k \rangle \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \delta_d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\delta_\theta & -\sin\delta_\theta & 0 \\ \sin\delta_\theta & \cos\delta_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &\sim \begin{pmatrix} \cos(\theta\langle k \rangle + \delta_\theta) & -\sin(\theta\langle k \rangle + \delta_\theta) & x\langle k \rangle + \delta_d \cos\theta\langle k \rangle \\ \sin(\theta\langle k \rangle + \delta_\theta) & \cos(\theta\langle k \rangle + \delta_\theta) & y\langle k \rangle + \delta_d \sin\theta\langle k \rangle \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

3차 벡터 $x = (x, y, \theta)$ 로 간결히 나타내면 다음과 같다.

$$\mathbf{x}\langle k+1 \rangle = \begin{pmatrix} x\langle k \rangle + \delta_d \cos \theta\langle k \rangle \\ y\langle k \rangle + \delta_d \sin \theta\langle k \rangle \\ \theta\langle k \rangle + \delta_\theta \end{pmatrix}$$

로봇의 오도메트리 에러를 포함한 다음 time step 배열은 다음과 같다.

$$\mathbf{x}\langle k+1 \rangle = \mathbf{f}(\mathbf{x}\langle k \rangle, \delta\langle k \rangle, \mathbf{v}\langle k \rangle) = \begin{pmatrix} x\langle k \rangle + (\delta_d + v_d) \cos \theta\langle k \rangle \\ y\langle k \rangle + (\delta_d + v_d) \sin \theta\langle k \rangle \\ \theta\langle k \rangle + \delta_\theta + v_\theta \end{pmatrix}$$

k 는 time step이고, $\delta\langle k \rangle = (\delta_d, \delta_\theta)^T$ 오도메트리 측정량이고

$\mathbf{v}\langle k \rangle = (v_d, v_\theta)^T$ 는 이전 간격의 랜덤 measurement noise 이다.

noise : 차량 이동에서 발생하는 noise이다.

6.1.2 Estimating Pose

우리는 칼만 필터를 통해 우리의 현재 위치와 그 정도에 대한 최선의 추정치 알 수 있다.

칼만 필터는 noise가 포함되어 있는 측정치를 바탕으로 선형역학계의 상태를 추정하는 재귀필터이다. 칼만 필터는 과거에 수행한 측정값을 바탕으로 현재의 상태 변수의 결합분포를 추정한다. 알고리즘은 예측과 업데이트의 두 단계로 이루어진다. 예측 단계에서는 현재 상태 변수의 값과 정확도를 예측한다.

현재 상태 변수의 값이 실제로 측정된 이후, 업데이트 단계에서는 이전에 추정한 상태 변수를 기반으로 예측한 측정치와 실제 측정치의 차이를 반영해 현재의 상태 변수를 업데이트한다.

상태 벡터는 차량의 구성이고 다음과 같이 나타난다.

$$\mathbf{x} = (x_v, y_v, \theta_v)^T$$

예측 방정식은 다음과 같다.

$$\hat{\mathbf{x}}^{+}_{\langle k+1 \rangle} = \mathbf{f}(\hat{\mathbf{x}}_{\langle k \rangle}, \hat{\mathbf{u}}_{\langle k \rangle})$$

$$\hat{\mathbf{P}}^{+}_{\langle k+1 \rangle} = \mathbf{F}_x \hat{\mathbf{P}}_{\langle k \rangle} \mathbf{F}_x^T + \mathbf{F}_v \hat{\mathbf{V}} \mathbf{F}_v^T$$

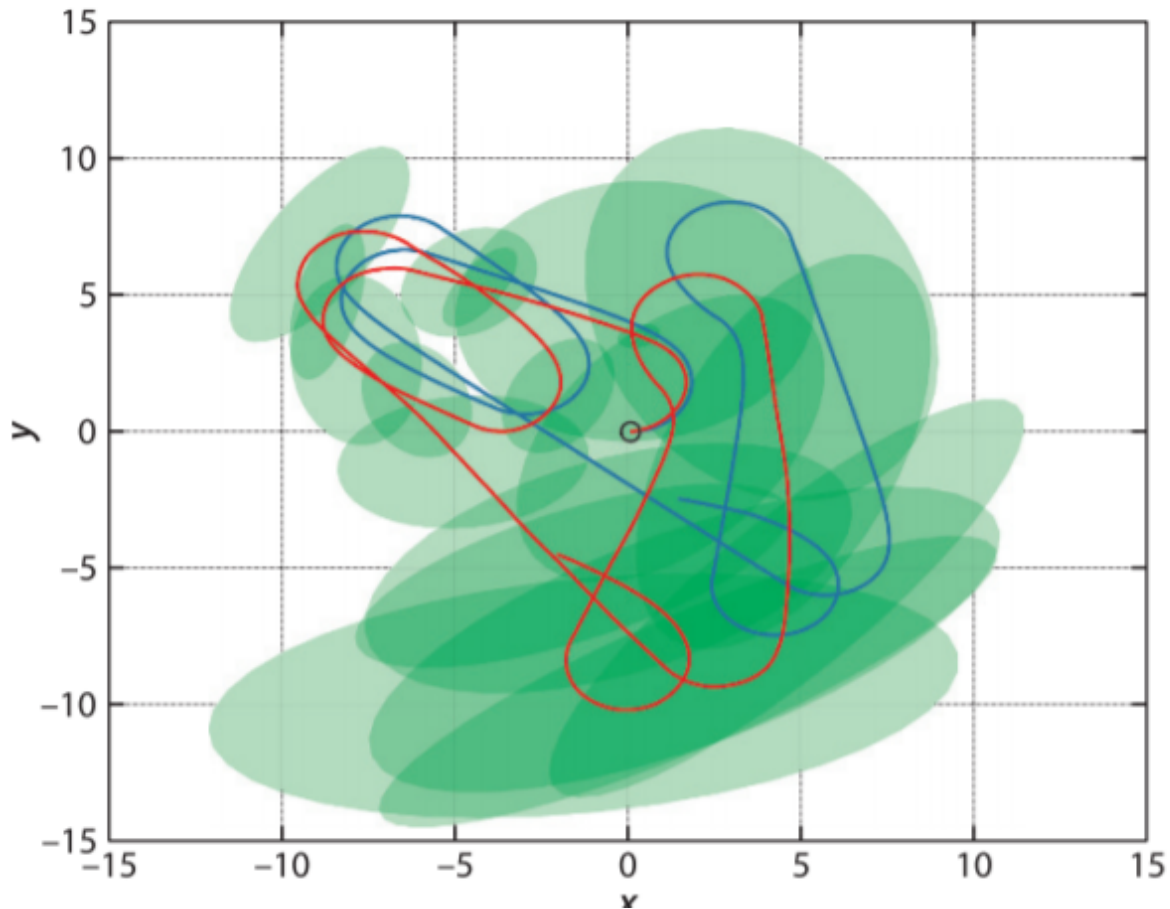
$\hat{\mathbf{u}}_{\langle k \rangle}$ 은 프로세스의 입력 값이다.

$\hat{\mathbf{P}}$ 은 공분산 로 측정된 차량 구성의 불확실성을 나타낸다.

$\hat{\mathbf{V}}$ 은 odometry의 공분산 배열 값으로 우리가 모르는 값이다.

\mathbf{F}_x 와 \mathbf{F}_v 은 자코비안 배열로 벡터형태의 미분이다.

Extended Karman Filter을 사용한 추측방법



파란선 : 실제 로봇의 로봇 경로

빨간선 : odometry로 측정한 로봇 경로

초록색 영역 : 95%의 신뢰도를 갖는 영역

<https://www.youtube.com/watch?v=mwn8xhgNpFY>

6.2 Localizing with a Map

이번에는 n개의 위치를 아는 landmark가 존재하는 지도를 사용할 것이다. 또한 로봇은 landmark의 범위와 베어링 각도를 인지 할 수 있는 radar, scanning-laser rangefinder 과 같은 센서를 갖추고 있다.

다음의 수식을 사용하여 로봇의 경로를 추정한다.

$$z = h(x, p_i) = \begin{pmatrix} \sqrt{(y_i - y_v)^2 + (x_i - x_v)^2} \\ \tan^{-1}(y_i - y_v)/(x_i - x_v) - \theta_v \end{pmatrix} + \begin{pmatrix} w_r \\ w_\beta \end{pmatrix}$$

$$\mathbf{x} = (x_v, y_v, \theta_v)^T$$

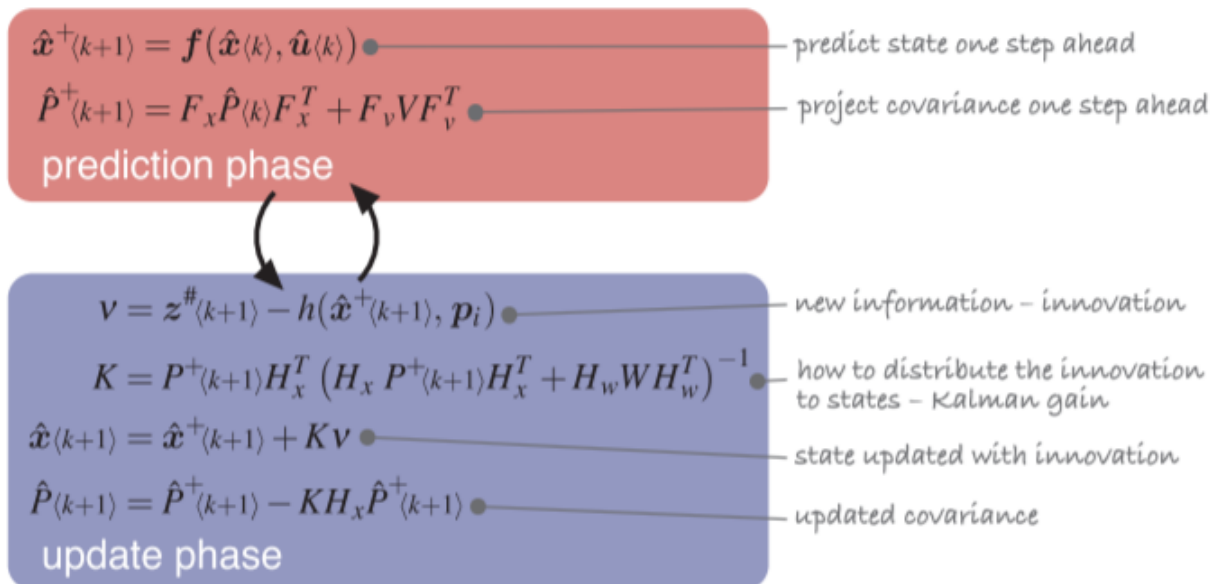
차량의 상태 값을 나타낸다.

$$\mathbf{p}_i = (x_p, y_p)^T$$

n번째 알고 있는 landmark의 위치를 나타낸다.

$$\mathbf{w} = (w_p, w_\beta)^T$$

센서의 오류 값을 나타낸다.



아래 그림은 다수의 landmark를 경로상에 주고 로봇의 경로를 EKF로 추정했을 때 결과 값을 MATLAB

으로 나타낸 그림이다.

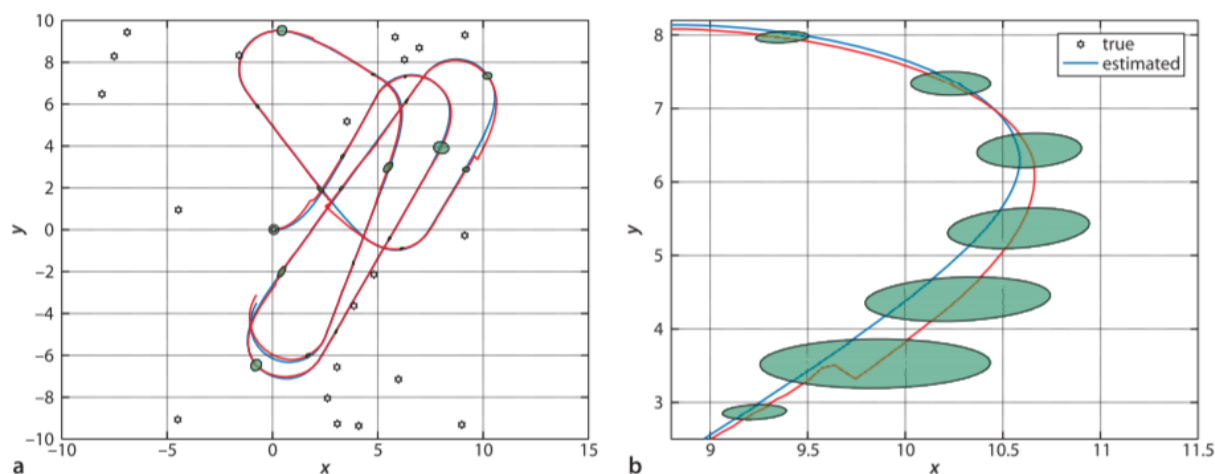
그림 a에서 불확실성 구간이 더이상 증가하지 않고 로봇의 실제 경로를 잘 계산한 것을 볼 수 있다.

그림 b에서 아래로 로봇이 움직일 수 록 에러 타원이 커지는 것을 확인 할 수 있다. 그리고 끝에서 타원이

어늘었는데 그것은 로봇이 landmark를 발견 했기 때문이다. 이것은 로봇이 랜드 마크를 볼 때 추

정 된 공

산을 크게 줄일 수 있기 때문이다. (공분산은 2개의 확률변수의 상관정도를 나타내는 값이다.)



파란선 : 로봇의 실제 경로

빨간선 : odometry에서 landmark를 제외한 경로

검은색 점 : landmark

초록색 영역 : 95%의 신뢰도를 갖는 영역

6.3 Creating a Map

Creating a Map은 landmark가 있는 환경에서 로봇이 이동하며 지도에 landmark의 위치를 작성하는 방법

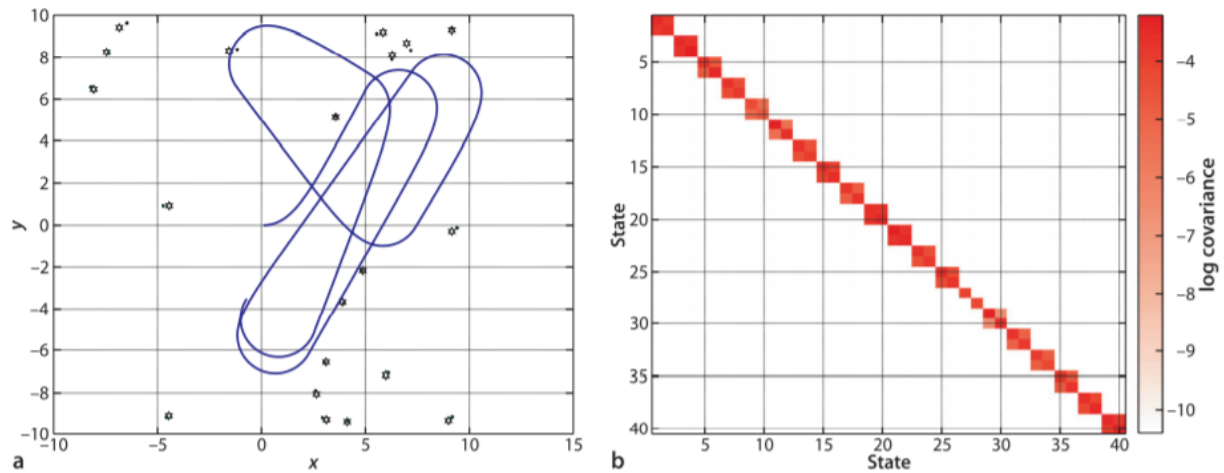
다. 이번에도 로봇은 레인지와 베어링 센서를 장착하고 있어서 landmark를 불확실하게 측정한다. 하지만

이번엔 로봇은 자신의 위치를 정확하게 알고 있다고 가정한다.

다음 그림은 MATLAB을 이용해 EKF mapping한 결과를 나타낸다.

그림a에서 나타난 것처럼 landmark의 실제 위치와 추정 위치가 완전히 수렴하지 않는 것을 알 수 있다.

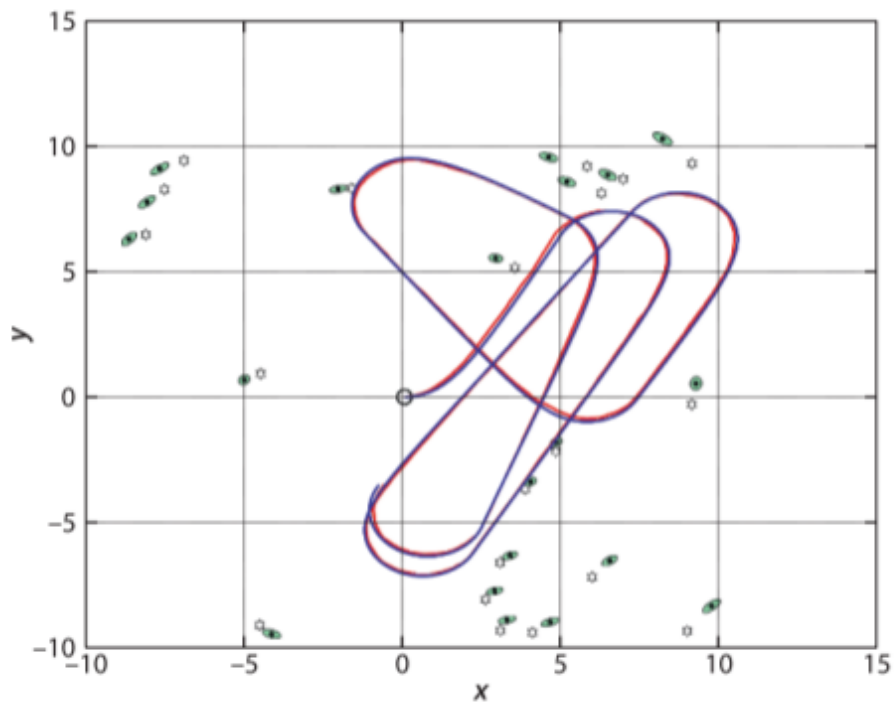
그림b 비 대각선 요소는 0이므로 랜드 마크 추정치가 서로 관련이 없거나 독립적임을 나타낸다. 하나의 랜드 마크를 관찰하면 다른 랜드 마크에 대한 새로운 정보가 제공되지 않기 때문이다.



검은색 점 : 추정한 landmark의 위치
 작은 검은색 점 : 실제 landmark의 위치
 파란색 선 : 로봇의 이동 경로

6.4 Localization and Mapping

마지막으로 Localization and Mapping은 위치를 결정하는 동시에 지도를 만드는 방법이다.
 다음 그림은 Localization과 Mapping을 동시에 실행하여 MATLAB으로 나타낸 이미지이다.



파란색 선 : 로봇의 이동 경로

빨간색 선 : 추정한 로봇의 경로

검은색 별표시 : 실제 landmark 위치

초록색 타원 : 95%신뢰도의 landmark 위치

그림 a에서 시간이 지남에 따라 불확실성이 점점 낮아 짐을 알 수 있다.

그림 b에서 비 대각선 요소는 0이 아니므로 랜드마크 추정치가 서로 관련있음을 나타낸다.

