Intelligent Visual Tracking Based on Realtime Particle Filters

강 훈
Intelligent Robot & Vision Lab.
중앙대 전자전기공학부
Email: hkang@cau.ac.kr
URL: http://sirius.cie.cau.ac.kr

Chung Ang University

지능로봇비전 연구실 최근 연구내용 요약 및 성과

	연구 내용 및 목표	연구 성과
시각적 물체 추적	▪ 실시간 Particle Filter 시스템 구현 - Webcam 기반 (15fps) ▪ Unified Histogram Equalizer 설계 ▪ 이동하는 사람의 얼굴 영상 추적 ▪ Geodesic Level Set기반 이동물체 윤곽 추출 ▪ PTZ 카메라의 시각적 물체 추적	▪ 물체의 특징점(IFAD, 색상, 외곽선)을 Particle Filter로 적용, 샘플링, 예측, 측정단계를 반복하여 해당 물체추적 ▪ 조명과 가려짐에 강인한 실시간 물체 추적 (5 Lux) ▪ 템플릿 기반으로 실내조명에서 이동하는 사람 얼굴을 추적한 영상 획득 → 얼굴 이미지 추출 및 저장 (JPEG) ▪ 유한 제어 점의 레벨 집합 이론을 적용, 연산속도 개선 ▪ PTZ 카메라의 Pan, Tilt 변화에 따른 IFAD 영상 개선

	향후 연구 계획	연구 계획 내용
물체 추적 및 인식	▪ 실시간 (2+) 채널 Stereo Visual Tracking 시스템의 시작품 제작 ▪ PTZ 카메라의 실시간 추적 및 인식 ▪ Particle Filter와 SIFT 기법에 의한 동시 추적 및 인식 시스템의 구현 ▪ 3차원 Human Motion Tracking	▪ Disparity Map의 실시간 구현 및 상대 z-축 깊이 정보에 의해 가려진 물체들의 구분 추적 ▪ SIFT (Scale-Invariant Feature Transform)의 실시간 설계 및 구현 ▪ 신경회로망기반 물체 인식 및 추적 시스템의 구현 ▪ 3차원 Human Model기반의 동작 추적 시스템 구현

Particle Filter: Definition & Problem Statement

- **Definition & Synopsis**
  - **Particle Filter** (PF) is a class of stochastic approximations of the state posterior with a set of N weighted particles (samples) $\{X^{(i)}, \pi^{(i)}\}$ where $X^{(i)}$ is a possible state and $\pi^{(i)}$ is the associated weight. So, we perform the **Monte-Carlo simulation** of underlying **probability distributions** which may take arbitrary form, instead of deriving the analytic solution.
  - **The state** may be any measurable contexts, e.g., position, velocity, rotating angle, scaling factor, tilting, etc.
  - **The measure** may be any quantities, e.g., contrast, digital image subtraction, edge-detected silhouette, 2D or 3D contours, RGB or HSV colors, etc.
- **Problem Statement**
  - **Particle Filter** (PF): Given the old particle set $\{X^{(i)}, \pi^{(i)}\}_{t-1}$, compute a new particle set $\{X^{(i)}, \pi^{(i)}\}_t$
    - Initialize the weights of a likelihood $\{\pi^{(i)}\}_0 = 1/N$, and its cumulative distribution $\{c^{(i)}\}_0 = i/N$.
    - Each particle represents a possible state with the associated weight of a likelihood which is measurable or computable.
    - Simulate deterministic or probabilistic motion of particles, updating the weights using measurement likelihood.
    - The prior density is replaced by the posterior density.

Particle Filter: Basics

- **Particle Filter Basics**
  - The posterior distribution is approximated by the ensemble of weights on all of these sampled particle sets.

    (Factored Sampling)

    $P(x_t|Z_t) = k\,P(Z_t|x_t)\,P(x_t|Z_{t-1})$, $\quad P(x_t|Z_{t-1}) \leftarrow$ **Prior Density**
  - Keep track of the state samples with non-zero posterior probability
  - More particles, better approximations.
    - But there's no rule for the right number of particles (we cannot keep track of state samples if N is small; while we may have oversamples or misleading if N is too large) → "impoverishment or depletion problems"
    - The number of effective weights:

$$N_{eff} = 1 \Big/ \sum_{i=1}^{N} \left\{\pi_t^{(i)}\right\}^2 \quad \left(1 \le N_{eff} \le N\right)$$

Particle Filter: Basics

- **Bayesian Inference Rule of "Factored Sampling"**
  - **Conditional Posterior Density Propagation**

$$
\begin{aligned}
P\left(X_t \mid Z^t\right) &= \kappa P\left(Z_t \mid X_t\right) P\left(X_t \mid Z^{t-1}\right) \\
&= \kappa P\left(Z_t \mid X_t\right) \int_{X_{t-1}} P\left(X_t \mid X_{t-1}\right) P\left(X_{t-1} \mid Z^{t-1}\right) dX_{t-1} \\
&\cong \kappa P\left(Z_t \mid X_t\right) \sum_{i=1}^{N} P\left(X_t \mid X_{t-1}^{(i)}\right) P\left(X_{t-1}^{(i)} \mid Z^{t-1}\right) \\
&= \kappa P\left(Z_t \mid X_t\right) q\left(X_t\right) \\
P\left(X_t^{(i)} \mid Z^t\right) &= \frac{P\left(Z_t \mid X_t^{(i)}\right) q\left(X_t^{(i)}\right)}{\sum_{j=1}^{N} P\left(Z_t \mid X_t^{(j)}\right) q\left(X_t^{(j)}\right)} = \frac{\pi_t^{(i)} q\left(X_t^{(i)}\right)}{\sum_{j=1}^{N} \pi_t^{(j)} q\left(X_t^{(j)}\right)}
\end{aligned}
$$

X_t : state, $Z^t = \{Z_1, \ldots, Z_t\}$: total observation, κ: propotional constant,
$P\left(X_t \mid Z^t\right)$: posterior density, $P\left(X_t \mid Z^{t-1}\right)$: prior density
$P\left(Z_t \mid X_t\right)$: measurement density, $P\left(X_t \mid X_{t-1}\right)$: Markov chain motion model,
$\pi_t^{(i)} = P\left(Z_t \mid X_t^{(i)}\right)$: weight of likelihood for the ith particle $X_t^{(i)}$,
$q\left(X_t\right) = \sum_{i=1}^{N} P\left(X_t \mid X_{t-1}^{(i)}\right) P\left(X_{t-1}^{(i)} \mid Z^{t-1}\right)$: proposal density

Particle Filter: Basics

- **"Importance Weight" (Proposal Density Available)**
  - **Target Density p(x), Proposal Density q(x)**

$$
\begin{aligned}
E\left[f(x)\right] &= \int f(x) \frac{p(x)}{q(x)} q(x) dx \\
&= \int f(x) W(x) q(x) dx \\
&= \int_{X_t} f\left(X_t\right) \frac{P\left(X^t \mid Z^t\right)}{q\left(X_t \mid X_{t-1}, Z_t\right)} q\left(X_t \mid X_{t-1}, Z_t\right) dX_t \\
&= \frac{1}{P\left(Z^t\right)} \int_{X_t} f\left(X_t\right) \left[\frac{P\left(Z_t \mid X_t\right) P\left(X_t\right)}{q\left(X_t \mid X_{t-1}, Z_t\right)}\right] q\left(X_t \mid X_{t-1}, Z_t\right) dX_t \\
W_t\left(X_t\right) &= \frac{P\left(Z_t \mid X_t\right) P\left(X_t\right)}{q\left(X_t \mid X_{t-1}, Z_t\right)}
\end{aligned}
$$

  - **Recursive Update for Importance Weights**

$$
W_t^{(i)} = W_{t-1}^{(i)} \frac{P\left(Z_t \mid X_t^{(i)}\right) P\left(X_t^{(i)} \mid X_{t-1}^{(i)}\right)}{q\left(X_t^{(i)} \mid X_{t-1}^{(i)}, Z_t\right)} \qquad \sum_{i=1}^{N} W_t^{(i)} =
$$

Visual Tracker (PF Algorithm)

- **Primary Updating Steps of PF Tracking (3 Steps)**
  - **(1) Sampling (Selection)**
    - randomly select N particles based on weights
    - use of prior distribution
    - same or nearby particle may be chosen multiple times
    - Selective Resampling – effective particle size N_{eff}, use of threshold in N_{eff}
  - **(2) Predicting (Particle Dynamics)**
    - prediction model (ARMA, Kalman or unscented filter)
    - move particles according to drift and diffusion
    - deterministic drift (known dynamics)
    - stochastic diffusion (unknown statistics)
  - **(3) Measuring (Output)**
    - get a likelihood for each new sample by making a prediction about the image's local appearance and comparing
    - update the associated weight on a particle from the measured posterior distribution from the particle set
    - estimate the states (moments, translation, rotation, scaling, etc.) from the particle set

Particle Filter: CONDENSATION Algorithm

Iterate

From the "old" sample-set $\{s_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}, n = 1, \ldots, N\}$ at time-step $t-1$, construct a "new" sample-set $\{s_t^{(n)}, \pi_t^{(n)}, c_t^{(n)}\}, n = 1, \ldots, N$ for time t.

Construct the n^{th} of N new samples as follows:

1. **Select** a sample $s_t'^{(n)}$ as follows:
   (a) generate a random number $r \in [0, 1]$, uniformly distributed.
   (b) find, by binary subdivision, the smallest j for which $c_{t-1}^{(j)} \geq r$
   (c) set $s_t'^{(n)} = s_{t-1}^{(j)}$
2. **Predict** by sampling from

$$p(x_t | x_{t-1} = s_t'^{(n)})$$

to choose each $s_t^{(n)}$. For instance, in the case that the dynamics are governed by a linear stochastic differential equation, the new sample value may be generated as: $s_t^{(n)} = A s_t'^{(n)} + B w_t^{(n)}$ where $w_t^{(n)}$ is a vector of standard normal random variates, and BB^T is the process noise covariance — see section 5.

3. **Measure** and **weight** the new position in terms of the measured features z_t:

$$\pi_t^{(n)} = p(z_t | x_t = s_t^{(n)})$$

then normalise so that $\sum_n \pi_t^{(n)} = 1$ and store together with cumulative probability as $(s_t^{(n)}, \pi_t^{(n)}, c_t^{(n)})$ where

$$c_t^{(0)} = 0,$$
$$c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)} \quad (n = 1, \ldots, N).$$

Once the N samples have been constructed: estimate, if desired, moments of the tracked position at time-step t as

$$\mathcal{E}[f(x_t)] = \sum_{n=1}^{N} \pi_t^{(n)} f\left(s_t^{(n)}\right)$$

obtaining, for instance, a mean position using $f(x) = x$.

Figure 6: The CONDENSATION algorithm.

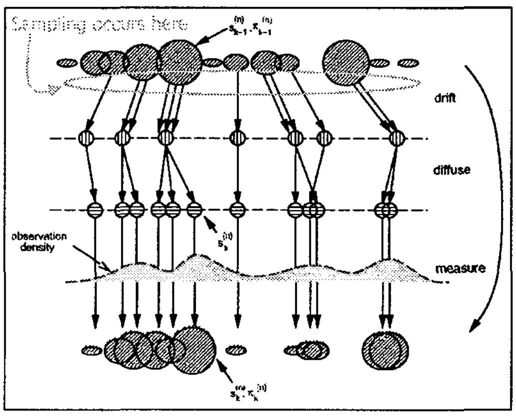
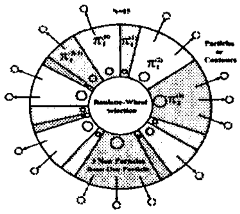


Figure 5: One time-step in the CONDENSATION algorithm. *Each of the three steps — drift-diffuse-measure — of the probabilistic propagation process of figure 2 is represented by steps in the* CONDENSATION *algorithm.*

Particle Filter: (1) Sampling

- (1) Sampling
  - Normalize N particle weights (Σ w(i) = 1 or Σ π(i) = 1)
  - Select N particles by picking randomly and uniformly in [0,1]
  - Similar to spinning a roulette wheel with arc-length of bins equal to particle weights
  - Resample adaptively focusing on promising areas of the state space
  - Steps:
    - Generate r in U[0,1].
    - Find the smallest j for {c(j)}t-1 > r.
    - Set St(i) = Xt-1(j).



"Roulette-Wheel Selection"
Again

Particle Filter: (2) Prediction

- (2) Prediction
  - Place each particle from the generative form of dynamics

$$S_{t+1}^{(i)} = F[S_t^{(i)}] + \varepsilon_t$$

random component (diffusion)

deterministic component (drift)

  - Nonlinear drift of each particle with different displacement
  - Independent diffusion of each particle typically modeled with a Gaussian
    - Obtain predicted samples:
    - Often use Kalman filters (KF) as a prediction model
    - Linear Systems:

$$X_{t+1} = A X_t + B U_t + v_t \quad E(X_0) = X_0$$
$$Y_t = H X_t + w_t \quad v_t, w_t \sim N(0,\sigma^2)$$

Particle Filter: (3) Measurement

- (3) Measurement
  - For each particle s(i), compute new weight π(i) using measurement likelihood

    $\pi_t(i) = P(Z_t|X_t(i) = s_t(i))$
  - Plausibility conditions
    - particles with impossible configurations are given 0 likelihood

      (ex: positions outside of image π(i)=0)
    - number of effective weights are important !

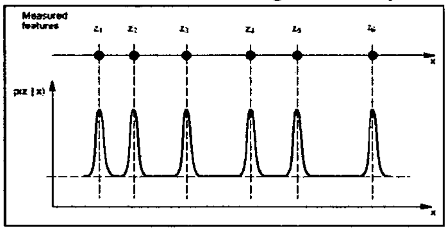


Figure 7: One-dimensional observation model. A probabilistic observation model allowing for clutter and the possibility of missing the target altogether is specified here as a conditional density p(z|x).

Observation Model (Example) :

Parametric Curves: B-Spline Snake with normal search

Particle Filter: State Estimate

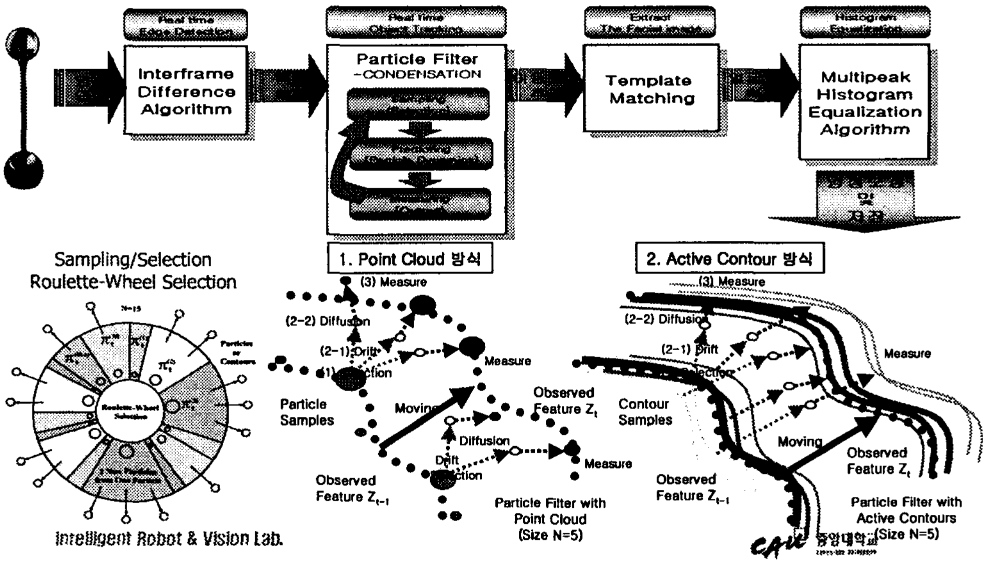
- Obtaining a State Estimate (Output) $E[g(X_t)]$

$$E\left[g(X_t)\right] = \sum_{i=1}^{N} \pi_t^{(i)} \cdot g\left(S_{t|t-1}^{(i)}\right)$$

  - Note that there's no explicit state estimate maintained
    - just a cloud of particles
  - We can obtain an estimate at a particular time by querying the current particle set
  - Some approaches
    - "Mean" particle
      - Weighted sum of particles (1st moment)
      - Confidence: inverse variance (2nd moment)
    - Really want a mode finder
      - Mean of tallest peak

시각적 추적연구 도식 및 Particle Filter 알고리즘

- **Block Diagram of Visual Object Tracking** (Facial Tracking Example)



Intelligent Robot & Vision Lab.

CAU 중앙대학교

PF Tracking: Solution to Re-Initialization Problem & Priority-based Subtractive Potential for Occlusion

- **Solution to Re-Initialization Problem**
  - Competitive Adaptive Vector Quantization (c-AVQ)
  - Region of Interest: Boundary Stripes (10-pixel wide)

c-AVQ for PF Re-Initialization

- Check boundary stripes P_B with threshold (3%~4% IFD data), if the condition holds,
- For all $p_i(k)$ in P_B > threshold at time t, find the winner (the i^{th} cluster)

$$w_i^*(k) = \arg\min_j \{\|w_j(k) - p_i(k)\|_2\} \quad j\#(\text{cluster})$$
$$w_i^*(k+1) = w_i^*(k) + \mu\left(p_i(k) - w_i^*(k)\right) \quad k \rightarrow$$

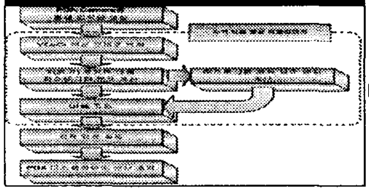
-Update the i^{th} particle state s.t. the average is shifted to the i^{th} cluster center (the winner)

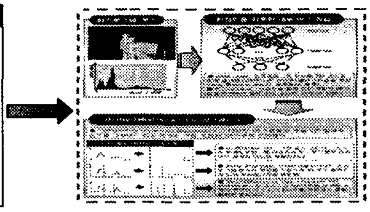
- **Priority-based Subtraction of Background IFD Potential**

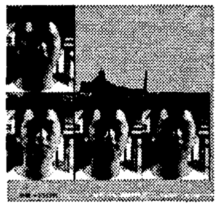
$V_k^* = \Delta P$

  - 차감 이미지(IFD)를 Potential Field (V)로 간주, 다수의 입자필터 들에 대해 우선 순위(Priority)를 정하고 차감 클러스터링 기법과 유사하게 IFD 버퍼에서 우선 순위에 의해 차감하며 순차적 샘플링, 예측, 출력을 반복하는 것
  - 즉, 두 물체가 교차하여 가려지더라도 각기 독립적으로 다른 방향으로 추적이 가능함

Self-Calibrated Multipeak Histogram Equalization





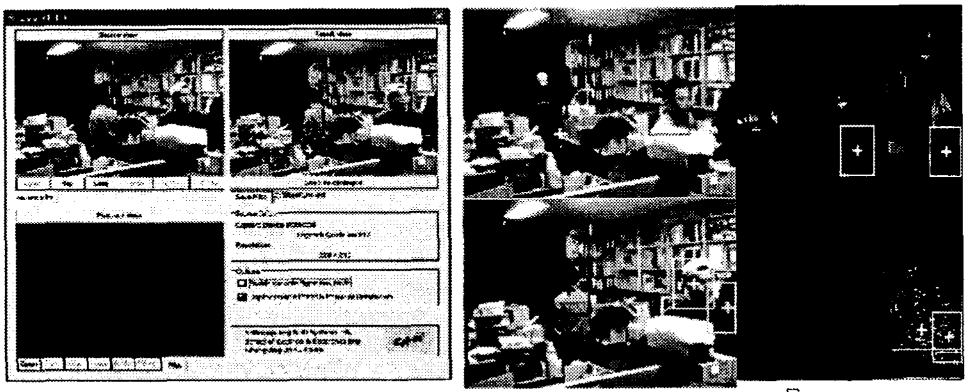


Intelligent Robot & Vision Lab.

CAU 중앙대학교

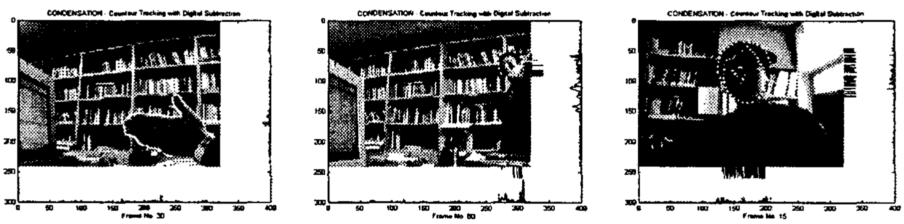
이동 물체 실시간 추적 기술 개발 (Particle Filter/Evol. Strategy 기법)

- **Particle Model 1 (Initial Particle Set)**
  - **Point Clouds 방식**
    - Gaussian or uniform priors with large covariance (initial uncertainty)
  - **실시간 다중물체추적: Pentium4 (2.4GHz), WebCam (15~30fps), Direct-X 9.0**
    - Particle Sets : 5(sets) x (100 particles), 조도 5 Lux에서의 물체 추적



이동 물체 추적 기술 개발 (Particle Filter/Evol. Strategy 기법)

- **Particle Model 2 (Initial Particle Sets)**
  - **B-Spline Snakes in Active Contours (Blake)**
    - Q' = WY + Q → Affine Shape Space (W: shape matrix; Y: shape-space vector; Q', Q: spline vectors)
    - Measure: (1) IFD+Init. Contour, (2) IFD+Init. Contour+HSV, (3) IFD+Init. Features



$$Q' = W \cdot Y + Q$$

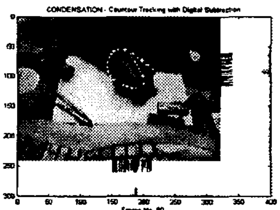
$Q = \left[Q_x^T \quad Q_y^T \right]^T$ ref. control pt. vector

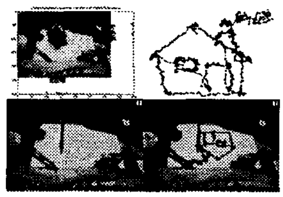
$Q' = \left[Q_x'^T \quad Q_y'^T \right]^T$ updated control pt. vector

$Q' = \left[x_{r,1}(t), \ldots, x_{r,n}(t); y_{r,1}(t), \ldots, y_{r,n}(t) \right]^T$ (n: total no. of control pts.)

$W = \begin{bmatrix} 1 & 0 & Q_x & 0 & 0 & Q_y \\ 0 & 1 & 0 & Q_y & Q_x & 0 \end{bmatrix}$ shape matrix

$Y = \left[\bar{x}_r, \bar{y}_r, \cos\theta - 1 + \alpha, \cos\theta - 1 + \alpha, \sin\theta, \sin\theta \right]^T$ shape-space vector

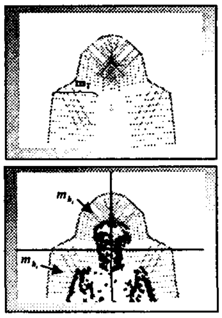




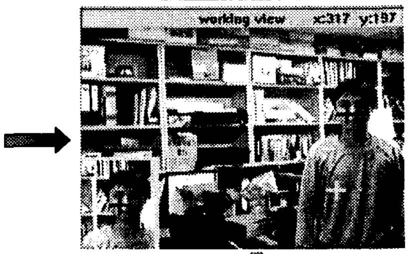
Particle Filter & Real-time Template Matching

- IFD (InterFrame Difference) 및 Template matching
  - 움직이는 사람의 얼굴 이미지를 PF를 이용하여 실시간 검출
  - 추적된 영상으로부터 얼굴 이미지 획득 및 보정
- Real-time Template Matching
  - Control point : $c_0, c_1, \ldots, c_n$
  - Measurement line : $m_0, m_1, \ldots, m_n$
  - Template Control Point의 Normal 방향의 Measurement Line을 이용하여 Sub-control Point를 동적으로 생성
- 얼굴 이미지 위치 결정

$$P_{head}(x,y) = \frac{\sum_{head}(sub-control\ point(x,y))}{N_{head}}$$



얼굴 이미지 위치 획득



Geodesic Level Set: Derivation

- Define F as the speed in the normal direction:

$$F = \vec{x}'(t) \cdot \vec{n} = \vec{x}'(t) \frac{\nabla \phi}{|\nabla \phi|} \Rightarrow F|\nabla \phi| = \vec{x}'(t) \nabla \phi \qquad (1)$$

- A particle on the front with path x(t) is on the zero level set:

$$\phi(\vec{x}(t), t) = 0 \xrightarrow{\text{by chain rule}} \frac{d\phi}{dt} = \phi_{\vec{x}} \cdot \vec{x}'(t) + \phi_t = 0$$

$$\nabla \phi(\vec{x}(t), t) \cdot \vec{x}'(t) + \phi_t = 0 \qquad (2)$$

(1) into (2): $\boxed{F|\nabla \phi| + \phi_t = 0, \text{ given } \phi(\vec{x}, t=0), \text{ "Continuous Level Set Equation"}}$

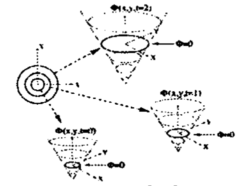
가변 윤곽선의 측지 레벨집합에 기반한 물체추적

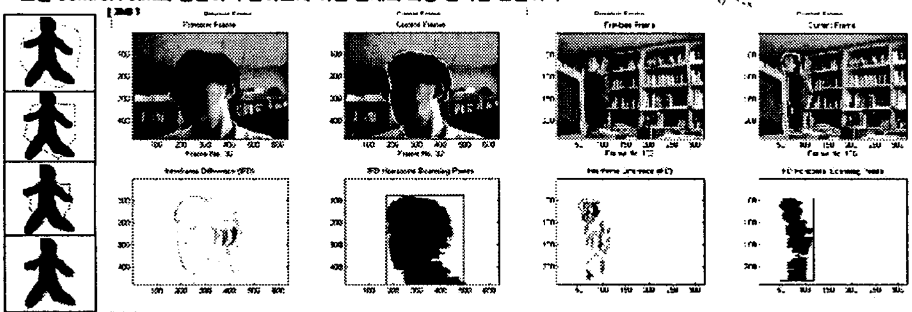
- **Deformable B-Snakes based on Geodesic Level Set**

- 유한한 가변 제어점으로 레벨집합 이론의 연산을 수행하여 연산속도를 획기적으로 개선

1. 입력 영상으로부터 초기폐곡선을 전개, 객체를 추출하기 위한 경계함수를 구한다. 경계함수는 Inter-frame Difference Algorithm을 이용하여 얻는다.
2. 추출 하고자 하는 물체의 내부 또는 외부에 초기 폐곡선을 지정하기 위하여, 사용자 입력 혹은 미리 정의된 유한개의 Control point을 설정한다.
3. Level Set 알고리즘을 이용하여 곡선을 전개 시킨다.
4. 인접한 Control Point의 관계를 고려하여 동적으로 point를 생성 또는 소멸 시킨다.
5. 모든 Control Point에서의 Level Set 함수 값이 불변할 때까지 곡선을 전개 시킨다.
6. 모든 Control Point의 집합이 추출하고자 하는 물체의 최종 윤곽선 점들이다.

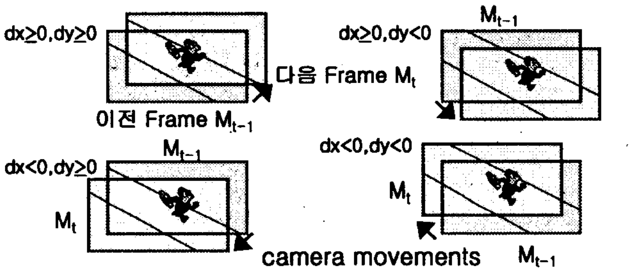
$$F \cdot |\nabla \Phi| + \frac{d\Phi}{dt} = 0,$$
$$\{\Phi(x(t), t=0) \text{ given}\}$$

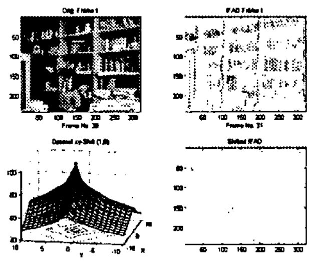




PT camera-based Shift-Invariant Tracking

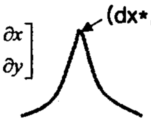
- **Optimal xy-Shifts** → 4 cases: (dx,dy) has different boundaries





- **Gradient Search**
  - MIT Rule for Optimal xy-Shift: (dx•,dy•)
  - Problems:
    - Sharp Peak & Slow Neighbor
  - Solution:
    - Adaptation Gain inversely proportional to Gradient
    - Reduce Gain to 1 if Fitness Ratio > 90%

$$\begin{bmatrix} dx_t \\ dy_t \end{bmatrix} = \begin{bmatrix} dx_{t-1} \\ dy_{t-1} \end{bmatrix} + \begin{bmatrix} \mu_x & 0 \\ 0 & \mu_y \end{bmatrix} \begin{bmatrix} \partial f_R / \partial x \\ \partial f_R / \partial y \end{bmatrix}$$

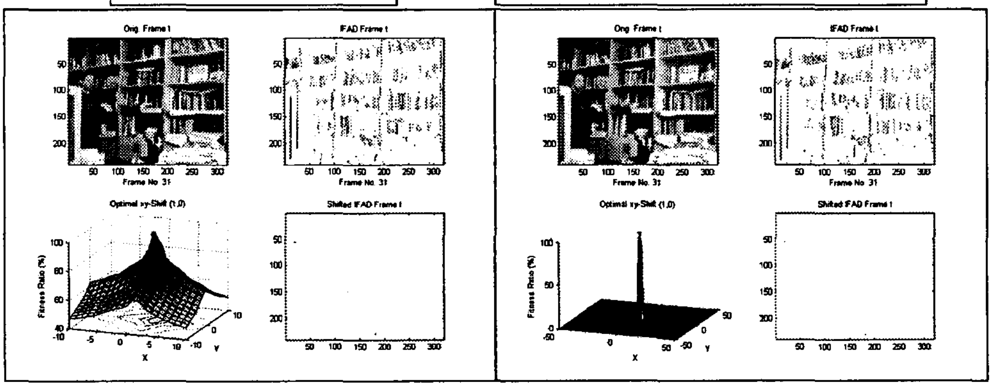


$$\mu_x = \mu_y = 1 + \alpha \cdot \exp -\|\nabla f_R\|^2 / \beta$$

PT camera-based Shift-Invariant Tracking

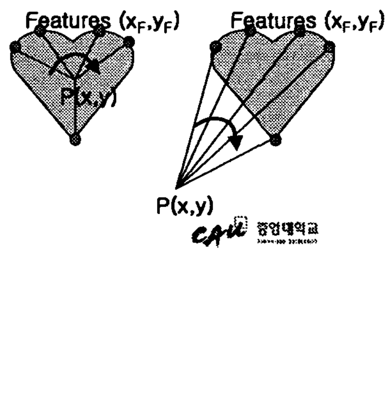
- Gradient Search within 8 Steps (2~8 steps)
  - 4 buffer-to-buffer subtraction per step
  - Min. 8 ~ Max. 32 buffer-to-buffer IFAD computations per frame

Gradient search at each frame | Use of previous optimal xy-shift at each frame



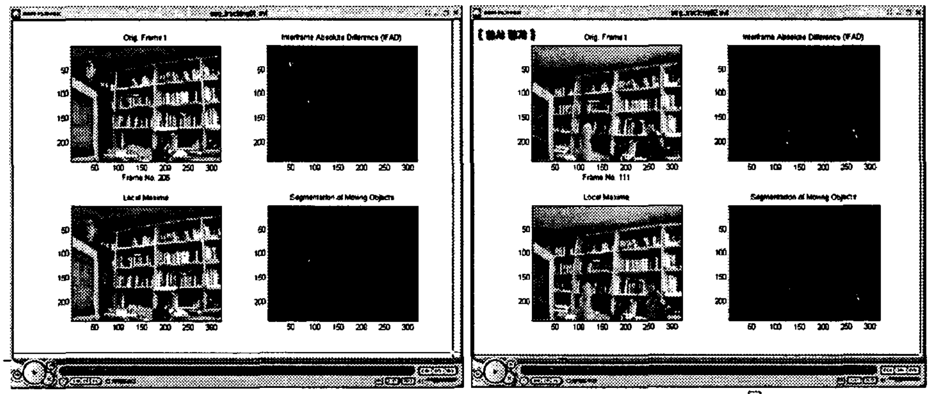
Segmentation Tracking

- Segmentation Tracking
  - To track & distinguish foreground (objects) from background (the environment).
  - For intensity images (with point-wise intensity levels), four popular approaches are:
    - Threshold techniques
    - Edge-based methods
    - Region-based techniques
    - Connectivity-preserving relaxation methods
  - Recent techniques of connectivity-preserving relaxation method
    - *Active contour model*
    - The main idea is to start with some initial boundary shape represented in the form of spline curves, and iteratively modify it by applying various shrink/expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an "elastic" contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is no easy task.
- Stepwise Procedure of Segmentation Tracking
  1. Measured Feature: IFAD or IF2DAD
  2. Preprocessing: Median Filter
  3. Data Compression: Local Maxima
  4. Angular Sorting: Quick-sort
  5. Accumulation of Sliced Angles:
     - Add by comparing with "corner angle"
     - Register if accumulation > (360-corner angle)



Segmentation Tracking

- Real-time Issue
  - Each Frame:
    - Sorting (Total No. of Feature Points), Iterations (Feature수 x Region of Interest)
    - Ex) 200 features x width 80 x height 60 = 960,000 (real-time almost impossible!)
    - Solution: Reduced Resolution (320x240 → 64x48 seg. map)
      Use of PF to obtain region of interest → Localize Regions of Interest



Epilogue

Thank You...