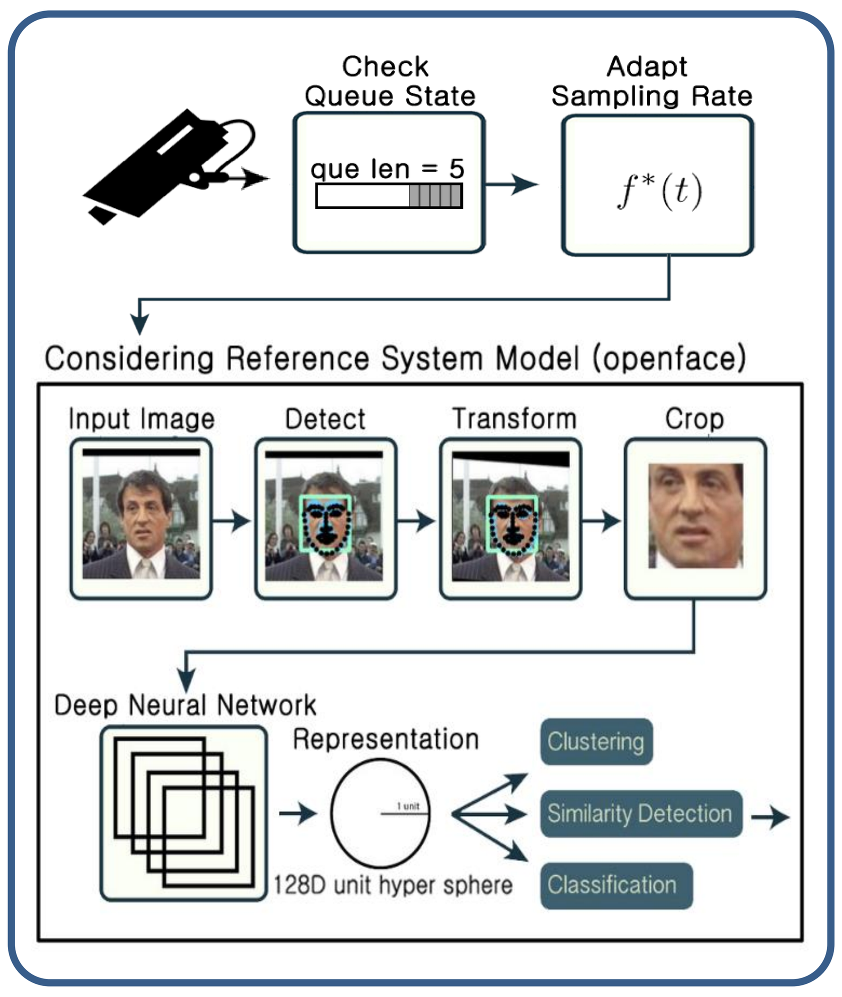
A Reliable, Self - Adaptive Face Identification Framework via Lyapunov Optimization

Dohyeon Kim, Joongheon Kim (Chung-Ang University, Seoul, Republic of Korea)
Jae Young Bang (Kakao Corporation, Seongnam, Republic of Korea)
E-mails: dohyeonkim0412@gmail.com, joongheon@cau.ac.kr, jae.bang@kakaocorp.com

Introduction and Backgrounds

Introduction and Reference FID System

Real-Time Face Identification System



Necessity of Dynamic Adaptation

→ In the FID system, processing time is proportional to number of faces in the frame ,so it is difficult to find a reliable and effective static low sampling rate.

Trade-off

→ With high sampling rate
- Low Queue stability
- High Hit rate

→ With low sampling rate
- High Queue stability
- Low Hit rate

Lyapunov Optimization

Objective Function:
Time-Average Recognition-Accuracy Maximization subject to Stability

$$\max: \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} S[\tau]$$

$$\text{subject to: } \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[Q[\tau]]$$

Lyapunov Optimization

$$D_o \leftarrow \arg\max_{D_i \in \mathcal{D}} \{V \cdot S(D_i) + Q[t] \cdot P(D_i)\}$$

Lyapunov Optimization

→ optimizing a time-average utility subject to queue stability when the objective function and the queue stability constraints are in a trade-off relationship.

- **D : selectable models**
- **S(Di) : objective function with model Di**
- **P(Di) : processing ability with model Di**
  - P(Di) = input(Di) – output(Di)
- **Q[t] : length of queues' backlog**
- **V : weight of objective function**
- **Do : Optimized model**

Dynamic Sampling Rate Adaptation via Lyapunov Optimization

Basic Algorithm

- ***f(t)*** **: selectable sampling rate during time t**
- **S(*f(t)*) : objective function**
  - Hit rate at sampling rate f(t)
- **λ(*f(t)*)** : processing ability with rate f(t)
- **Q[t] : length of queues' backlog at time t**
- **V : weight of objective function**
- ***f**(t)** **: Optimized sampling rate**

Lyapunov Optimization

$$f^*(t) \leftarrow \arg\max_{f(t)\in\mathcal{F}} \{V \cdot S(f(t)) - Q(t) \cdot \lambda(f(t))\}$$

Simple Form

$x^* \leftarrow \arg\max_{x\in X} \{V \cdot x - Q(t) \cdot x\}$ [x = sampling rate = *f(t)*]

Algorithm 1 Frame rate control via Lyapunov optimization

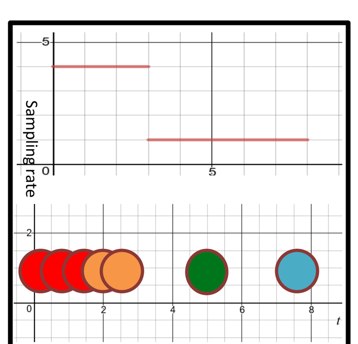
Initialize:
1: $Q(t) \leftarrow 0$ and $t \leftarrow 0$
Main algorithm:
2: **while** $t \leq T$ **do** // T: operation time
3: Observe $Q(t)$ and $\mathcal{T}^* \leftarrow -\infty$
4: **for** $f(t) \in \mathcal{F}$ **do** $\mathcal{T} \leftarrow V \cdot S(f(t)) - Q(t) \cdot \lambda(f(t))$
5: **if** $\mathcal{T} \geq \mathcal{T}^*$ **then** $\mathcal{T}^* \leftarrow \mathcal{T}$ and $f^*(t) \leftarrow f(t)$

Problems of Basic Algorithm

Necessity of Considering Penalty

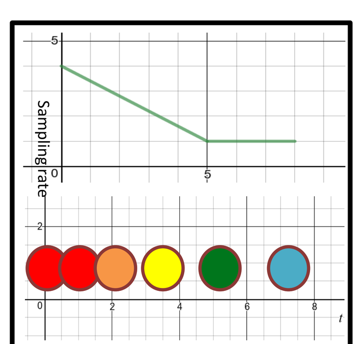
→ The higher the sampling rate, the faster the exhaustion of queue

→ The faster the exhaustion of queue, the lower the value of hit items.



With Basic Algorithm

VS



Hit value considered model

➲ Multiply a **Penalty function *P(f(t))*** to objective function

Improved Algorithm

$$x^* \leftarrow D(\ \mathbf{argmax}_{x\in X}\{L_{max} \cdot x \cdot P(x)^{Q(t)/Lmax} - Q(t) \cdot x\}\ )$$

- ***x*** : selected sampling rate at time t
- ***x**** : Optimal sampling rate at time t+1
- ***X*** : Candidate set of sampling rate
- L_{max} : Max size of Queue
- ***P(x)*** : Penalty function for sampling rate.
  (If it is hard to predict accurately, just set it to P(x) = $-x/L_{max} + 1$)
- ***Q(t)*** : Size of queue at time *t*
- ***D(x)*** : Distribution function for sampling rate. (If it is not considerable, just set it to 1)

Performance Evaluation and Concluding Remarks

Performance Evaluation

Queue Stability and Performance

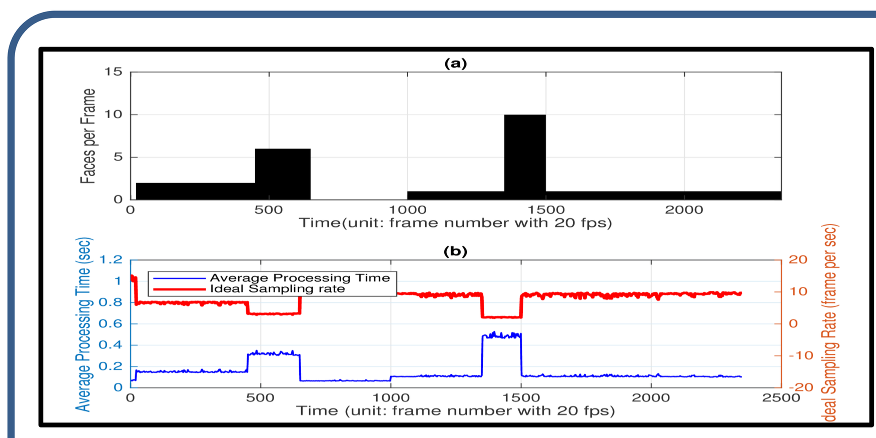


Figure 1. Scenario and Ideal Sampling Rate

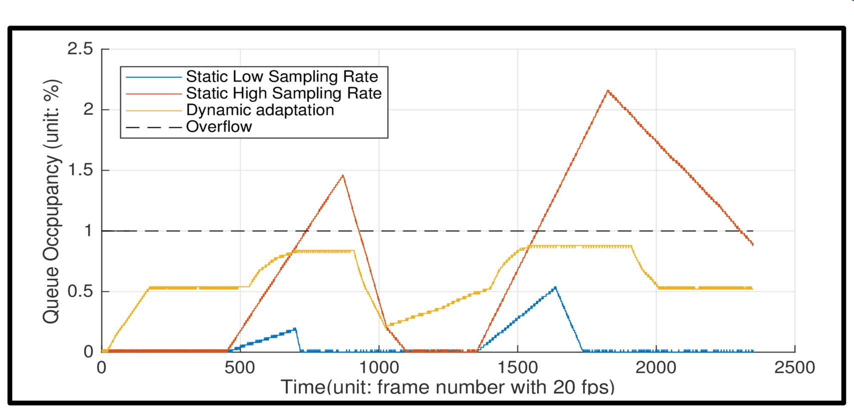


Figure 2. Comparison of Queue Stability

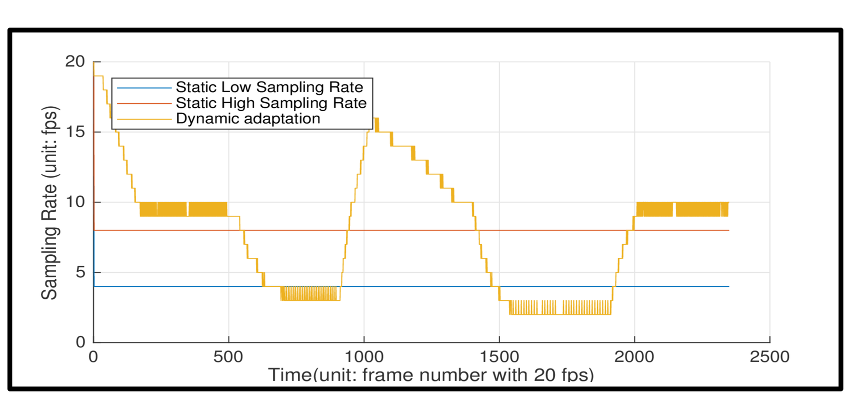


Figure 3. Comparison of Sampling Rate

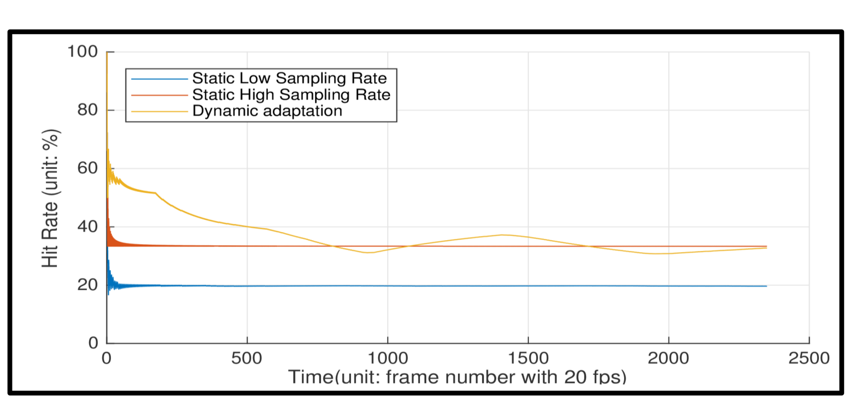


Figure 4. Comparison of Hit Rate

Performance Evaluation

→ **In dynamic adaptation, we assume that we can't guess average processing time.** So we leave it to adapt in range 0 to maximum input rate(in this system, 20 fps) . On **the other hand, with static sampling rate , we have to know specific processing time** before it works.

→ Dynamic adaptation present **higher stability** than static high sampling rate, and **higher hit rate** than static low sampling rate.

→ Even **the more specific** we know the range, **the better the performance** will be

Concluding Remarks

Expectation Effectiveness

→ For stochastic real-time system which uses queues

→ For system that is hard to know specific processing time of each items

Future Work

→ Improving to consider other considerations

→ Generalization this framework for any real-time system which uses queues

Workshop on AI Systems at Symposium on Operation Systems Principles (AISys@SOSP 2017)
October 28, 2017
Shanghai, China