Face Tracking with Guaranteed Framerates on Mobile Phones

Rymut Bogusław and Bogdan Kwolek

This paper addresses the problem of face tracking with guaranteed framerates at mobile devices. The frame rate of the computationally inexpensive algorithm is not affected by the image content. An ellipse with fixed orientation is used to model the head. The position and the size of the ellipse are determined with respect to intensity gradient near the edge of the ellipse and skin color probability in the ellipse's interior. The tracking is done using particle swarm optimization. The experiments were done using Lenovo S10 netbook and Nokia N900 smart phone.

Aganda

- The problem
- Particle Swarm Optimization
- The Algorithm
- Experiments
- Summary

The problem

- Limited resources (small computational power, quality of cameras) of mobile devices.
- Real-time face tracking, with guaranteed frame-rates.
- Tracking in long image sequences, varying illumination conditions.

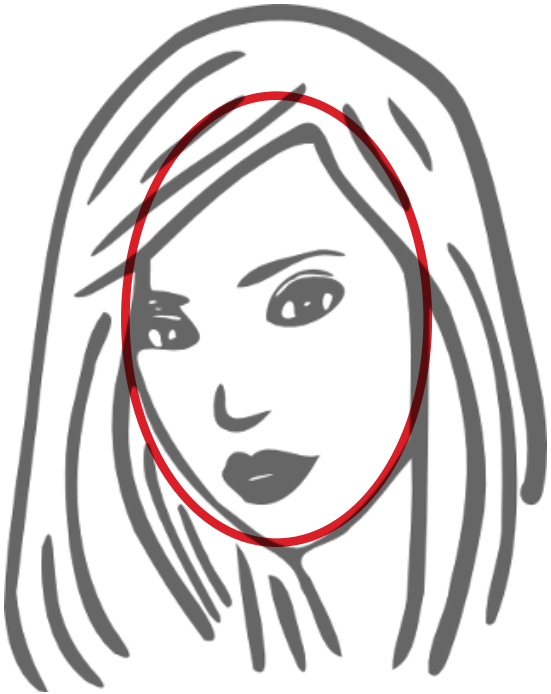
Specification: Nokia N900

- Arm Cortex-A8 600 MHz
- 256 MB SDRAM + 768 MB virtual NAND
- Front camera: Microelectronics VS6555 VGA ST 640x488@30
- Rear camera: Toshiba ET8EK8 640x492@30
- Operating system: Maemo 5

Specification: Lenovo S10

- Intel Atom N270 1.6 GHz
- 1 GB DDR RAM
- Front camera : Chicony 640x480@20
- Operating system: MeeGo 1.0

Shape model



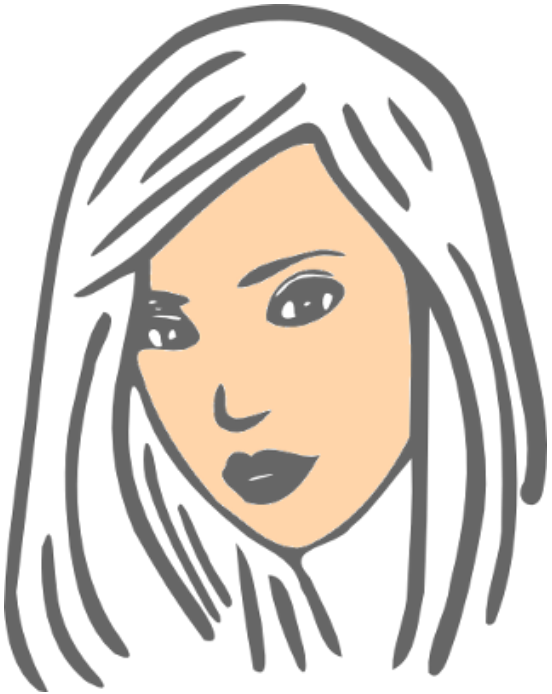
- The head is modeled as vertical ellipse with fixed aspect ratio.

$$f_k(x) = a \cdot f_i(x) + (1-a)\, f_e(x)$$

$$f_i(x) = \sum_{j \in E(x)} \sqrt{\left(\alpha_x(j) \cdot G_x(j)\right)^2 + \left(\alpha_y(j) \cdot G_y(j)\right)^2}$$

$$f_e(x) = \sum_{j \in E(x)} G_{xy}(j)$$

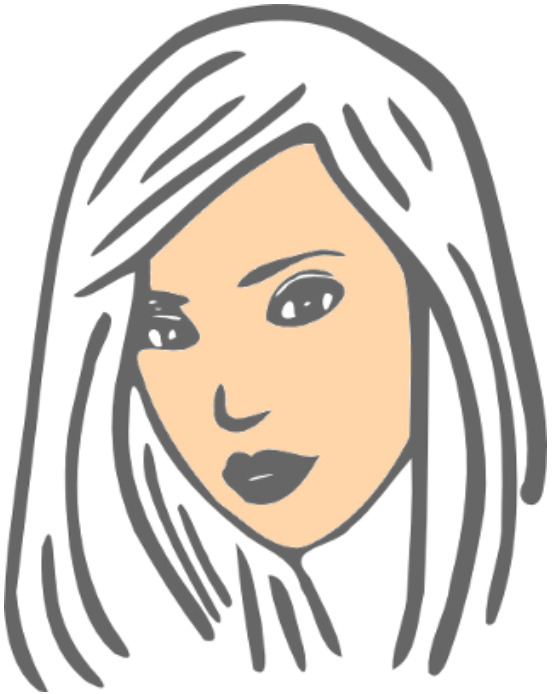
Skin probability model



- Histogram back-projection.
- Histogram constructed in YCbCr or HSI color space.

$$f_s(x) = \sum_{j \in N(x)} H(j)$$

Skin probability model adaptation



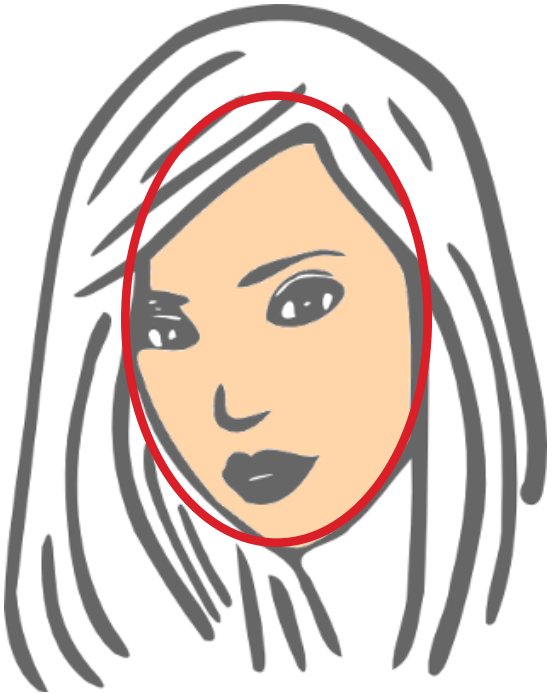
- Histogram adaptation

$$h_{b,t+1} = \beta h_{b,t} + (1-\beta) h_{b,t+1}$$

aided by skin locus

$$R \geq G \wedge R \geq B$$

The object model



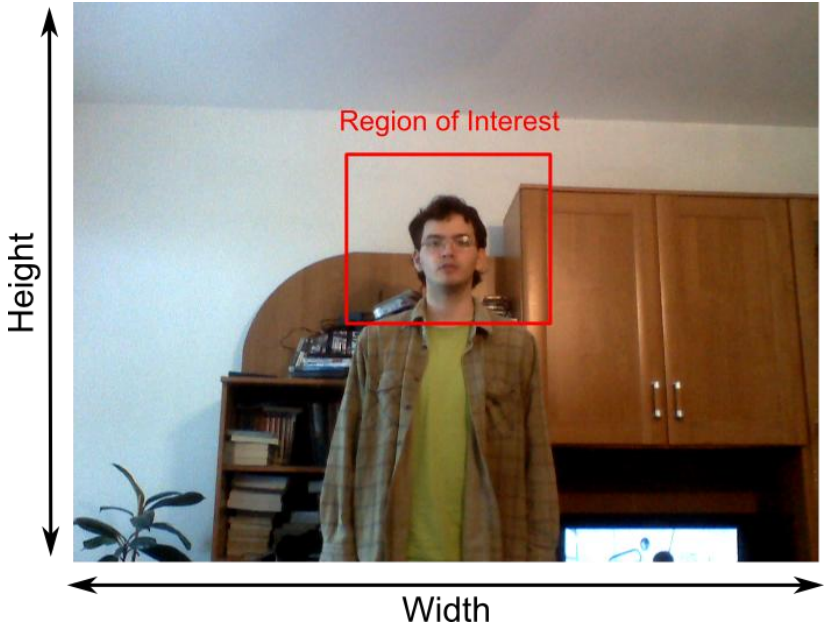
$$F(x) = \left[a \cdot f_i(x) + (1-a)\, f_e(x) \right]^{w_1} \cdot f_s(x)^{w_2}$$

$f_i(x)$ - normalized intensity gradient along ellipse boundary

$f_e(x)$ - normalized angle between the normal of the ellipse and gradient

$f_s(x)$ - normalized skin probability in ellipse interior

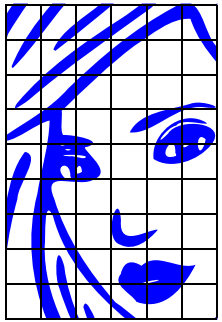
Region of Interest

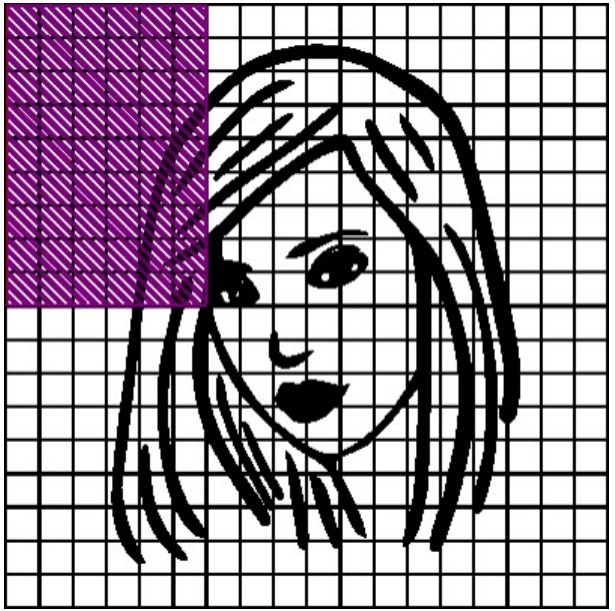


- The image processing is done in a sub-region of the input image.
- Size of the clipping window is 176x144px.

Particle Swarm Optimization (PSO)

- Stochastic optimization algorithm
- The optimization is achieved via a set of particles
- Particles collaborate each other in optimization process





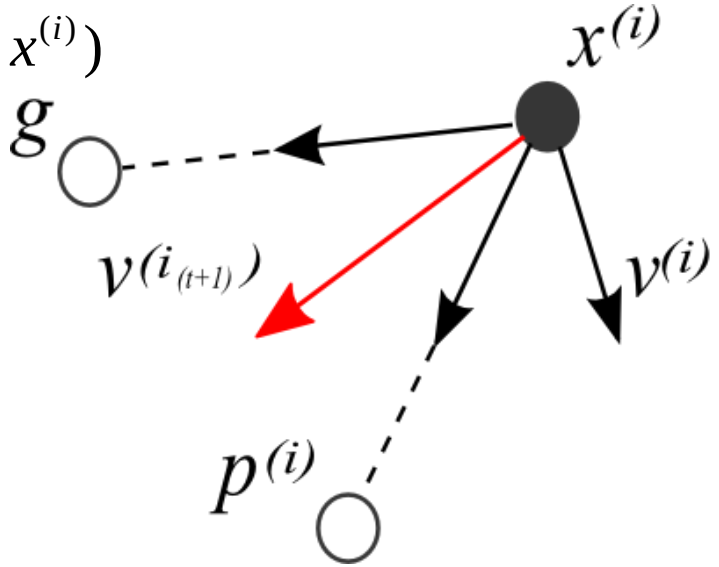
Particle Swarm Optimization

$$v^{(i)} \leftarrow \omega v^{(i)} + c_1 r_{1,j}\left(p^{(i)} - x^{(i)}\right) + c_2 r_{2,j}\left(g - x^{(i)}\right)$$

$$x^{(i)} \leftarrow x^{(i)} + v^{(i)}$$

$$p^{(i)} \leftarrow \begin{cases} x^{(i)}, & \text{if} \quad f\left(x^{(i)}\right) < f\left(p^{(i)}\right) \\ p^{(i)}, & \text{otherwise} \end{cases}$$

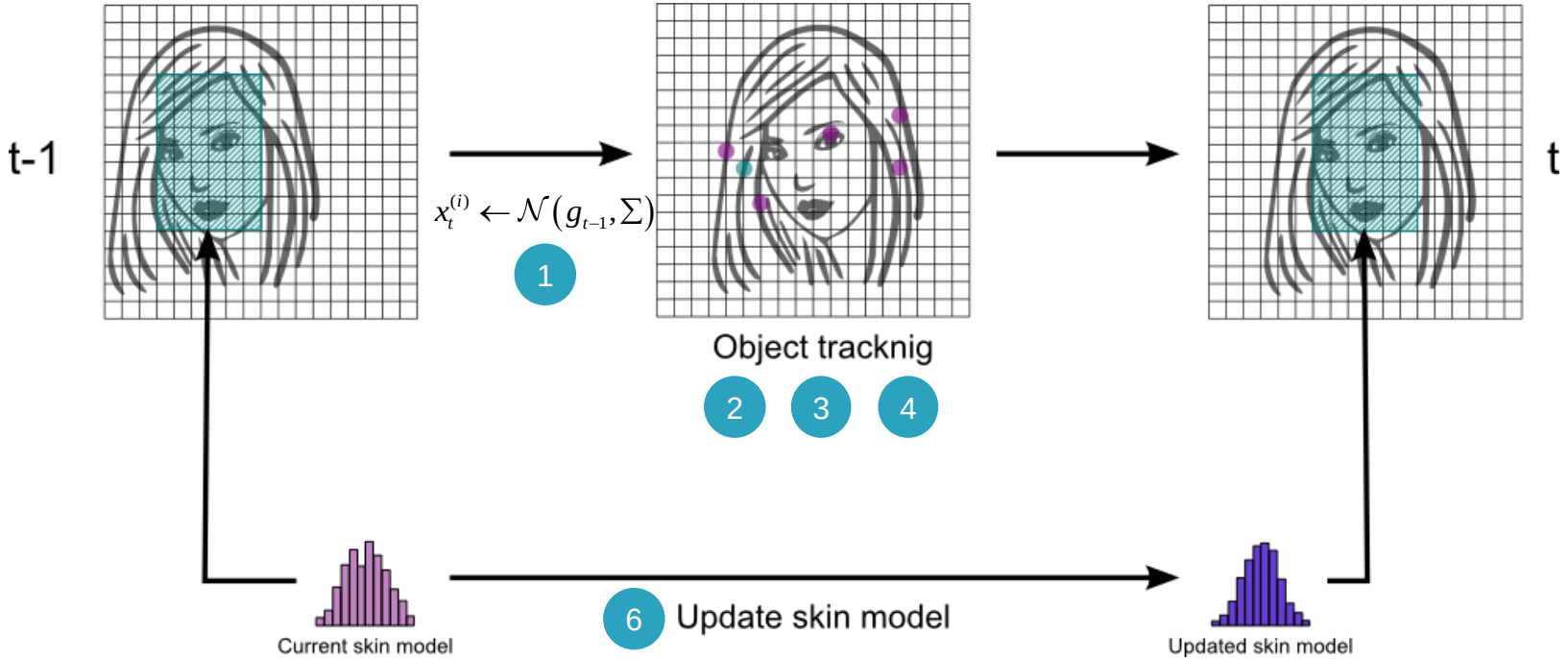
$$g \leftarrow \arg\min_{p^{(i)}} \left\{ f\left(p^{(i)}\right) \right\}$$



Particle Swarm Optimization

1. Assign each particle a random position in the problem hyperspace
2. Evaluate the fitness function and find local best value for each particle
3. Find the particle that has the best fitness value
4. Update the velocities and positions of all particles
5. Repeat steps 2-4 until maximum number of iterations is not attained
6. Update skin model
7. Acquire new image

PSO – based object tracking

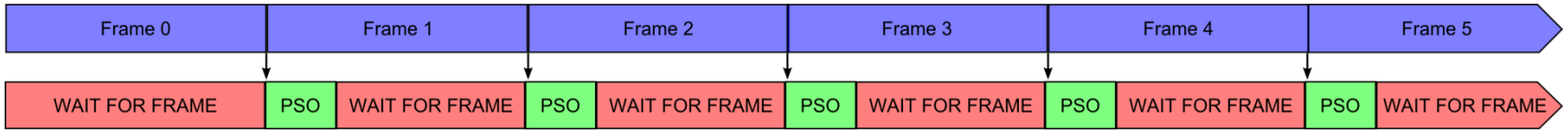


Example – smart phone-based face tracking

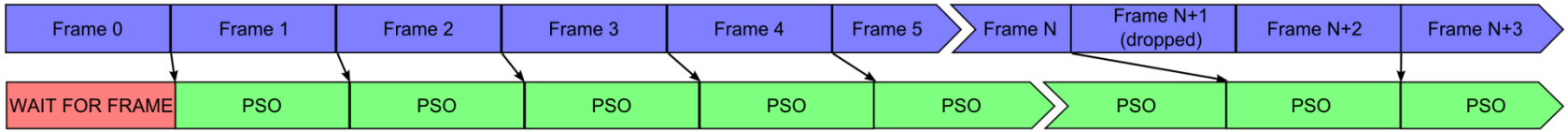
Example – netbook-based face tracking

Tracking with guaranteed frame-rates

	Lenovo S10	Nokia N900
Tracking [ms]	26	49
Frequency [fps]	~20	~20



Nokia N900



Lenovo S10

Summary

- In this paper we presented an algorithm for face tracking with guaranteed frame rates on mobile devices.