Background Subtraction with Streaming DPGMMs

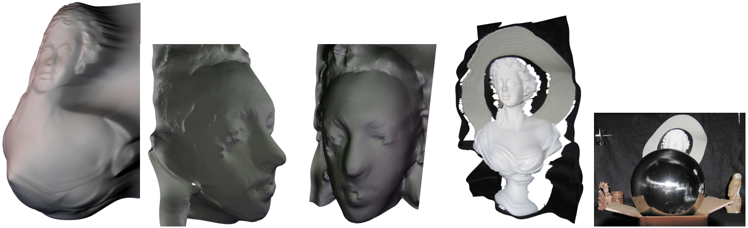
Tom S. F. Haines
thaines@cs.ucl.ac.uk

9$^{\text{th}}$ September 2016

Contents

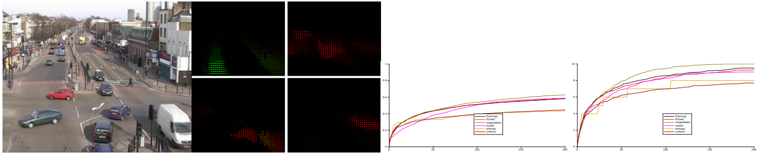
- Summary of past work.
- Background subtraction.

PhD



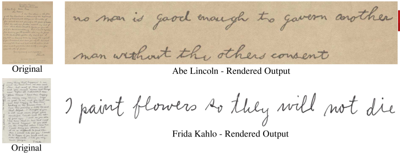
- Shape from Shading [SfS], SfS with Stereopsis and Light Source Estimation from Stereopsis.
- Belief propagation: Discrete, Gaussian and with directional statistics (Fisher-Bingham 8 parameter distribution).
- Generalised Hough transform, Bayesian with branch and bound.

Postdoc I



- Background subtraction – Streaming Dirichlet process [DP] mixture models.
- Abnormal behaviour detection – Various topic models, inc. DP based.
- Active learning – DP model, rare class discovery with simultaneous boundary refinement.

Postdoc II



BBC Sign in News Sport Weather iPlayer TV Radi

NEWS

Home UK World Business Politics Tech Science Health Education Entertain

Technology

Can a computer copy your handwriting?

Rory Cellan-Jones
Technology correspondent

12 August 2016 | Technology

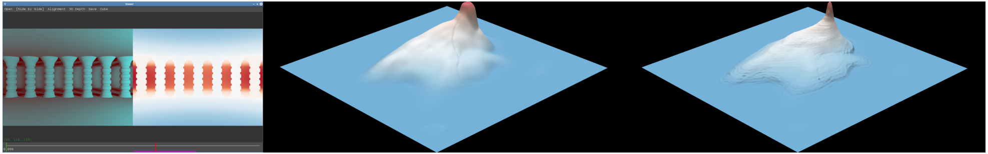
Researchers at University College London have taught a computer to imitate anyone's handwriting.

They have created an algorithm that can take a sample of handwritten text, examine its qualities, and then write any text in the same style.

There are already typefaces in word processing programs that produce text in a fairly uniform handwritten style. But what Tom Haines and his fellow UCL researchers have done is create software that they claim reproduces the messy details of any individual writer's hand.

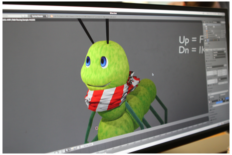
They call their system My Text In Your Handwriting and have tried it out on samples of handwritten text from historical figures such as Abraham Lincoln and the creator

- Handwriting project – received media attention, commercialisation likely.
- Automatic generation of tileable textures.
- Probabilistic and robust camera path estimation.
- Realistic landscape models, by combining real data with an artists rough model.



Outreach


- Director of **3Dami**, an educational non-profit.
- Summer school for college students.
- They make a 3D animated film in a week – have created 14 films!
- Teach them how - process I know in detail.




Background Subtraction

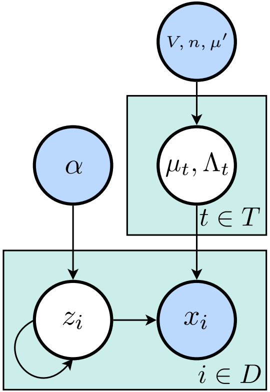
- Finds the interesting regions of a video.
- 'Blue screening without a blue screen'.
- Below by row: Input, ground truth, presented.

time of light foreground
moved object day switch waving trees camouflage bootstrap aperture

Method

- Construct a per-pixel model of the background. . .
  . . . using a Dirichlet process Gaussian mixture model (DPGMM).
- Bayes rule on each pixel obtains class membership probability (foreground or background).
- Construct a Markov random field and regularise; solve using belief propagation with momentum (GPU friendly).

Gibbs Sampling



- Gibbs sample the Chinese restaurant model.
- Integrate out μ_t and Λ_t - conjugate prior means we can use the student-t distribution and update incrementally.
- Sample the z_i using

$$P(z_i = t) \propto \begin{cases} \frac{m_t}{\alpha + \sum_{i \in T} m_i} P(x|V_t, n_t, \mu_t) & t \in T \\ \frac{\alpha}{\alpha + \sum_{i \in T} m_i} P(x|V, n, \mu') & t \notin T \end{cases}$$

Infinitely Long Videos

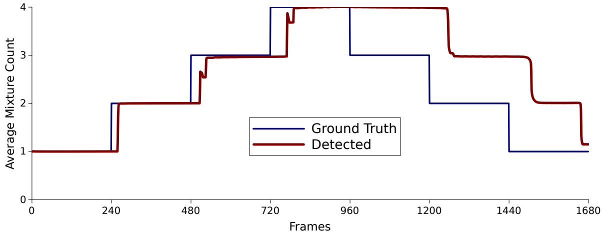
- **Problem:** We have an infinite number of samples.
  Solution: Sample each new feature vector once only. Can therefore throw them away; keep sample counts and incremental Student-t parameters at each mixture component only.
  Comment: This is not a problem because we slowly forget (next slide).

- **Problem:** With infinite data comes infinite mixture components.
  Solution: Clamp number of mixture components. When a new mixture component is created replace component with lowest sample count.
  Comment: Works, but hard to justify.

Forgetting

- As time passes the background can change – the model needs to forget the old background.
- This is achieved by capping the "confidence" and scaling it back when the threshold is passed.
  - "Confidence" $=$ number of exemplars in highest probability mode. This is tied with the Student-t parameters that are also counting exemplars.
- This causes older sample to be repeatedly scaled to irrelevance as time passes.
- Note that we are capping individual modes, not the model as a whole (which would be the expected solution!).

Behaviour



- Graph shows synthetic input, that is changing its mode count.
- Mode count estimated by thresholding the size of each mixture!
- It discovers new modes quickly, forgets old modes slowly.

Regularisation

- Standard Markov random field over image.
- Have: $P(\text{data}|\text{background})$; Need: $P(\text{background}|\text{data})$ – assume $P(\text{data}|\text{foreground})$ is uniform and apply Bayes rule.
- An edge preserving cost is used between pixels, with a Cauchy distribution-like cost that depends on colour difference:

$$P(l_a = l_b) = \frac{h}{h + m \times ||c_a - c_b||_2} \tag{1}$$

  h is the "half life", the distance at which the cost is exactly half. m is decreased when a pixel is not like any of its neighbours, to discourage salt and pepper noise.
- Solved with belief propagation – graph cuts would be optimal, but slow on GPU. Done hierarchically for speed.

Further Details

- Background subtraction is an old area – it takes a certain amount of engineering to be competitive . . .
  - Weight updates by probability of belonging to the background from last frame.
  - Update model with Gaussian distributions instead of point estimates. Distributions calculated with neighbour pixels for robustness to camera shake.
  - Compensate for lighting change, using a mean shift based estimate of global change from all local estimates (involves adjusting all model parameters!).
  - Custom colour model to reduce the effect of shadows.
  - GPU implementation for speed.

Quantitative Results

- Big charts of numbers can be found in paper...
- ...executive summary:

  SABS (synthetic): 27% improvement.
  Wallflower: 33% less mistakes.
  Star: 4% improvement.
  Change detection: 2% improvement.

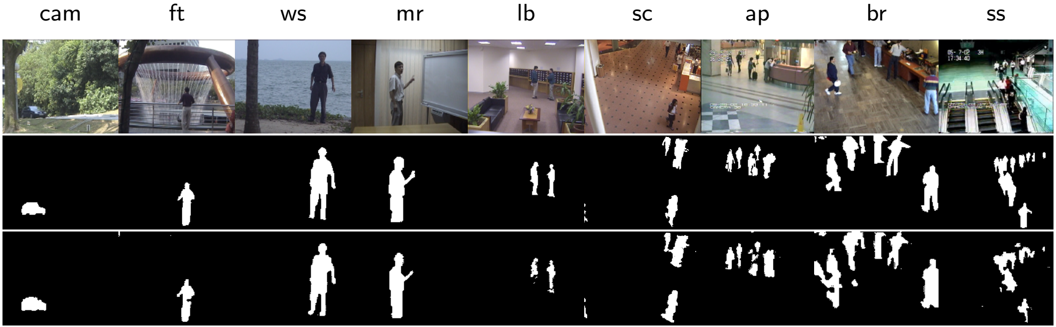
  (Compared to nearest competitor in each case, at time of publication)

Output - Wallflower

moved object	time day	of light switch	waving trees	camouflage	bootstrap	foreground aperture

(First row = input; second row = ground truth; third row = output)

Output - Star



(First row = input; second row = ground truth; third row = output)

Conclusions

- The DPGMM works well in this situation – it models multi-modal distributions and learns how much noise there is.
- Consequentially, it does really well at dynamic backgrounds that stump other algorithms. It's also great with camouflage.
- The method of forgetting learns model changes quickly, but keeps the old model around for a long time, to be reused if needed.