Texture Modelling with Non-contiguous Filters

Ralph Versteegen, Georgy Gimel’farb and Pat Riddle

Department of Computer Science, The University of Auckland
Goal: improved texture modelling with Markov–Gibbs random fields (MGRF)

A generalisation/unification of FRAME/MGRF nesting and Field-of-Experts

Learning square and non-contiguous linear filters

Evaluation using texture synthesis
Markov-Gibbs random fields (MGRF)

MGRFs (a.k.a. MRFs, Markov networks, undirected graphical models):
- Probabilistic models based on statistics of local features $f$
- Given statistics $S_f(x_{obs})$, learn distribution $P$ such that $\mathbb{E}_P[S_f(x)] \approx S_f(x_{obs})$

![Diagram of training image, statistics, and model sample]
MGRFs: energy-based models

- MGRFs are energy-based, i.e. defined by a Gibbs distribution:

\[ P(x) \propto \exp(\sum_i E(-f_i(x))) \]

Image \quad Features \quad Potentials

\[
\begin{align*}
\lambda_f & \quad \lambda_E \\
x & \quad f_i(x) & \quad E(f_i(x))
\end{align*}
\]

E.g. \( \lambda_f \) are the coefficients of a linear filter.
Field-of-Experts is a popular MGRF image prior for e.g. denoising

Learn filters (features) $\lambda_f$ and potentials $\lambda_E$ simultaneously using stochastic gradient descent... not so easy.

Only small filters used, e.g. $7 \times 7$

Usually simple potentials (experts), not very suitable for complex texture
Nesting MGRFs

- Greedily add new features/potentials to an existing model
- These are corrections to the base model $P_{i-1}$ to meet new constraints:
  \[ \mathbb{E}_{P_{i-1}}[S_f(x_i)] \approx S_f(x_{\text{obs}}) \] (\(x_i\): image generated from \(P_{i-1}\), \(x_{\text{obs}}\): training image)

Iterate:
- Generate (sample) image \(x_i\) from \(P_{i-1}\)
- Select candidate feature \(f\) with largest disagreement (distance) in statistics \(\text{dist}(S_f(x_i), S_f(x_{\text{obs}}))\).
- Learn new parameters \(\lambda_f\) to meet the new constraints
Nesting MGRFs

- Greedily add new features/potentials to an existing model $P_{i-1}$ to meet new constraints $\mathbb{E}_{P_i}[S_f(x_i)] \approx S_f(x_{\text{obs}})$ ($x_i$: image generated from $P_{i-1}$, $x_{\text{obs}}$: training image)

- Iterate:
  - Generate (sample) image $x_i$ from $P_{i-1}$
  - Select candidate feature $f$ with largest disagreement (distance) in statistics $\text{dist}(S_f(x_i), S_f(x_{\text{obs}}))$.
  - Learn new parameters $\lambda_E$ to meet the new constraints
Nesting MGRFs

- Greedily add new features/potentials to an existing model
- These are corrections to the base model $P_{i-1}$ to meet new constraints $\mathbb{E}_{P_i}[S_f(x_i)] \approx S_f(x_{\text{obs}})$ ($x_i$: image generated from $P_{i-1}$, $x_{\text{obs}}$: training image)
- Iterate:
  - Generate (sample) image $x_i$ from $P_{i-1}$
  - Select candidate feature $f$ with largest disagreement (distance) in statistics $\text{dist}(S_f(x_i), S_f(x_{\text{obs}}))$.
  - Learn new parameters $\lambda_E$ to meet the new constraints
Nesting MGRFs

- Greedily add new features/potentials to an existing model
- These are corrections to the base model $P_{i-1}$ to meet new constraints $\mathbb{E}_{P_i}[S_f(x_i)] \approx S_f(x_{\text{obs}})$ ($x_i$: image generated from $P_{i-1}$, $x_{\text{obs}}$: training image)
- Iterate:
  - Generate (sample) image $x_i$ from $P_{i-1}$
  - Select candidate feature $f$ with largest disagreement (distance) in statistics $\text{dist}(S_f(x_i), S_f(x_{\text{obs}}))$.
  - Learn new parameters $\lambda_E$ to meet the new constraints
Example: Nesting iterations

Top row: model samples.
Below: the filter learnt and added on each iteration.
Heterogeneous MGRFs

- Can build heterogeneous models.
- During nesting, first add simple grey-level difference (GLD) features (learnt offsets)

\[ f(x_1, x_2) = x_1 - x_2 \]

- Then add higher-order features.

Training image + GLD model = 6 × 6 linear filters model = Heterogeneous model
Single-texture models allow customisation of features:

- Combine Field-of-Experts and model nesting: add new features during learning
- Use learnt non-contiguous filters:
  - Can model long range and large-scale interactions
  - Much cheaper or simpler than large square filters, multiple scales or layers of filters.
  - Learn the filters
- Learn heterogeneous MGRFs by nesting models
- (Feature selection is $\lambda_f$ selection.)
Filter learning

- Learn filter by gradient descent: *maximise* error in statistics
- Fast: no sampling, not stochastic.
- To learn non-contiguous filters:
  - Heuristically select offsets (shape) of the new filter:
  - Assume high-order interactions usually also induce pairwise interactions
  - Pick the offsets with the largest errors in greylevel difference pairwise
Learn filter by gradient descent: *maximise* error in statistics

Fast: no sampling, not stochastic.

To learn non-contiguous filters:

- Heuristically select offsets (shape) of the new filter:
- Assume high-order interactions usually also induce pairwise interactions
- Pick the offsets with the largest errors in greylevel difference pairwise
Examples of learnt $6 \times 6$ square filters
Examples of learnt non-contiguous filters
Comparison to local binary pattern features

- Our previous work used LBPs as MGRF features.
- LBPs compare surrounding pixels $p_1, \ldots, p_k$ to a central pixel $p_0$, result is a binary vector $[p_1 \neq p_0, \ldots, p_k \neq p_0]$.
- Example: selected 13-th order LBP features ($2^{12} = 4096$ $\lambda_E$ parameters each):
Comparing MGRFs with different feature types:
- Original
- Learnt non-contiguous
- Learnt $6 \times 6$
Comparing MGRFs with different feature types:

- Original
- Learnt non-contiguous
- Learnt $6 \times 6$
Comparisons - with GLD features

Comparing MGRFs with different feature types:

<table>
<thead>
<tr>
<th>Original</th>
<th>Previous work: LBPs</th>
<th>Learnt non-contiguous</th>
<th>Learnt $6 \times 6$</th>
<th>Portilla &amp; Simoncelli</th>
</tr>
</thead>
</table>

Comparisons - with GLD features

Comparing MGRFs with different feature types:

Original
Previous work: LBPs
Learnt non-contiguous
Learnt $6 \times 6$

Portilla & Simoncelli

Comparisons - with GLD features

Comparing MGRFs with different feature types:

Original

Previous work: LBPs

Learnt non-contiguous

Learnt $6 \times 6$

Portilla & Simoncelli

Comparisons - with GLD features

Comparing MGRFs with different feature types:

Original  Previous work: LBPs  Learnt non-contiguous  Learnt $6 \times 6$  Portilla & Simoncelli

Comparisons - with GLD features

Comparing MGRFs with different feature types:

Original   Previous work: LBPs   Learnt non-contiguous   Learnt $6 \times 6$   Portilla & Simoncelli

Comparisons - with GLD features

Comparing MGRFs with different feature types:

<table>
<thead>
<tr>
<th>Original</th>
<th>Previous work: LBPs</th>
<th>Learnt non-contiguous</th>
<th>Learnt $6 \times 6$</th>
<th>Portilla &amp; Simoncelli</th>
</tr>
</thead>
</table>

Comparisons - with GLD features

Comparing MGRFs with different feature types:

<table>
<thead>
<tr>
<th>Original</th>
<th>Previous work: LBPs</th>
<th>Learnt non-contiguous</th>
<th>Learnt $6 \times 6$</th>
<th>Portilla &amp; Simoncelli</th>
</tr>
</thead>
</table>

Discussion & Conclusion

- Our novelty: filter learning plus nesting
  - Combines advantages from FoE and FRAME
  - Fast separate filter learning
  - Trialed non-contiguous linear filters

- Specialising and selecting features especially effective for texture modelling

- Comparing learnt filters to local binary patterns for synthesis:
  - Better texture details, worse for complex structure.

- Non-contiguous filters mostly equivalent & redundant to square ones given long-range GLDs.
Discussion & Conclusion

- Our novelty: filter learning plus nesting
  - Combines advantages from FoE and FRAME
  - Fast separate filter learning
  - Trialed non-contiguous linear filters

- Specialising and selecting features especially effective for texture modelling

- Comparing learnt filters to local binary patterns for synthesis:
  - Better texture details, worse for complex structure.

- Non-contiguous filters mostly equivalent & redundant to square ones given long-range GLDs.

Thanks for listening