テクスチャ解析を用いたQuickBird画像の土地被覆分類の可能性

東京大学大学院農学生命科学研究科
大久保　悟

報告の内容

1. Object-based分類とtexture解析
2. Texture情報の種類
3. Texture解析の単位
4. QuickBird Panchromatic データを用いたGLCMによるtexture解析の実態
   - 適正解析範囲（window size）の設定
   - 有効texture情報の抽出
5. 画像分類におけるCARTの有効性
6. Definiensでできること/できないこと

超高解像度衛星画像の分類

- 高解像度と低解像度
  - 1ピクセルの意味（Woodcock & Strahler, 1987）
    - 高解像度：実世界における1つの対象物を視覚的に構造化する複数ピクセルの1つ
    - 低解像度：実世界における複数の小さな対象物の集合体のシグナル
      - Cf. 10m幅を持つ樹冠を1つの対象物とした場合，高解像度では樹冠を構成する葉や枝，影が個々のピクセルで，その集合で対象物が構成。低解像度では，1つのピクセルが樹冠1つ，もしくは森林を構成
- 高解像度画像におけるimage-object
  - 分類の対象物＝空間的自己相関の高い空間的に集塊したピクセル群（Hay et al. 2003）

Object-based分類とtexture解析

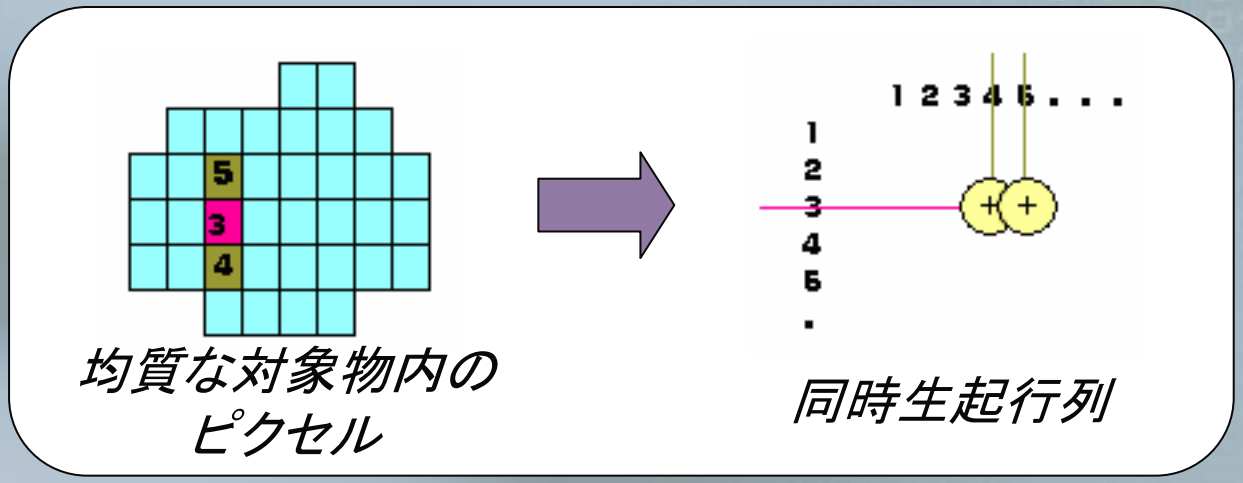
- 空中写真等判読の手動分類
  - 人間による画像の総合判断(Hay et al., 2003)
    - "… humans can cognitively group similar toned and spatially arranged pixels into meaningful *image-objects* that correspond to real-world entities within the geographic extent of the scene being assessed."
  - 判断材料(例えば，大矢ら，1998)
    - 注目物の形態(大きさ，形，陰影)
    - 色調(色の彩度，明度，濃度)
    - **きめ:テクスチャー**
    - 模様:パターン(線状，方形状，モザイク状)

Texture情報の種類

- 分類対象物のtexture情報の計量化
  - 1st-order statistics
    - 平均値，分散，度数分布から得られる各種情報（尤度，歪度，レンジなど）など，対象物を構成するピクセル輝度値の単純統計量
    - 利点：簡便；欠点：空間的なパターンは無視
  - 2nd-order statistics
    - GLCM (Grey level co-occurrence matrices)
    - Geostatistical measures
    - 自己相関指数
    - フーリエ変換の指標
    - 利点：空間パターンをある程度考慮；欠点：複雑

GLCMによるtexture情報(1)

- GLCM（同時生起行列）とは



均質な対象物内のピクセル

同時生起行列

  - ある一定距離にあるピクセル間で，組み合わせのピクセル値の頻度を示す正方行列
  - 任意のピクセルの上，右上，右，右下方向，の4方向で個別に行列作成することもあるが，4方向まとめて一つの行列を作るのが推奨
  - セル間距離は1とするのが一般的

GLCMによるtexture情報(2)

- GLCMを用いた主なtexture measures
  - Homogeneity: $\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j) \Big/ 1+|i-j|$
  - Dissimilarity (k = 1), Contrast (k = 2): $\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j)|i-j|^k$
  - Entropy: $\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j)\log(p(i,j))$
  - Angular second moment: (Uniformity or energy) $\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j)^2$

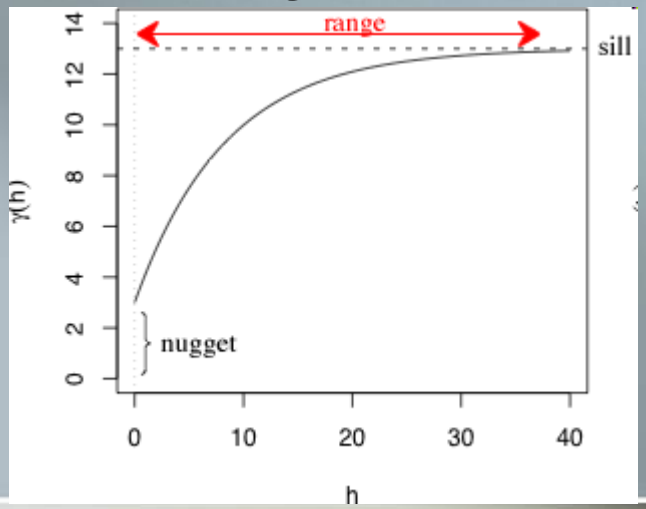
$p(i,j)$はGLCMにおける行列(i,j)の相対頻度

Geostatistical measure: Semivariogram

- Semivariogramとは
  - Semivariance:

$$\gamma(\mathbf{h}) = \frac{1}{2N(\mathbf{h})} \sum_{i=1}^{N(\mathbf{h})} \{DN(x+\mathbf{h}) - DN(x)\}^2$$

    - *DN*はピクセル値，*N*(**h**)は任意の2点間の距離が**h**となる組み合わせ数
  - Semivarianceとhのグラフ（経験的variogram）
    - 一般的に右図のような形状になることが知られている。この形状が対象物の肌理によって異なることが経験的に知られており，nugget, sill, rangeが情報量となる。



Texture解析の単位

- Texture情報の利用
  - Per-pixel解析にも利用可能
    - ただし，それぞれのピクセルに対しmoving windowを設定し，window内で解析
    - 適切なwindow sizeは？
      - 小さいと適切な情報が得られない
      - 大きすぎると異質な土地被覆を含むwindowができあがってしまうため，こちらも適切な情報にならない
  - Object単位の場合
    - Image-objectが解析単位
    - 異質な空間を含むことがない→土地被覆の境界効果を気にしなくてよいが，適正最小objectサイズは考慮する必要がある

Texture解析と分類の試行

- 対象地:
  - インドネシア西ジャワバンドン市郊外
  - 土地利用/被覆の特徴
    - 竹林を休閑とした切替畑システム
      - ステージの違う畑地と竹林の存在
    - 形状や生育段階の異なる水田
      - 沖積低地:区画面積の広い水田
      - 丘陵地や火砕流台地の開析谷:棚田状の水田
      - 作期の異なる水田が隣接
    - 竹林と樹木園
    - ホームガーデンを有する集落/有しない集落
  - 使用画像
    - 2004年8月28日撮影QuickBird画像
      - パンクロ0.6m, スペクトル2.4m
      - GLCMにはパンクロを11-bit→8-bitに線形変換して使用

土地被覆区分

- 樹林・タケ林（多年生作物地）
  - 竹林と樹木園
- 水田
  - 区画の大小×生育ステージ（計4区分）
- 畑地
  - 竹林伐採中（枯れたタケの葉が覆う）
  - 耕作準備完了（土壌水分・有機物が多い）
  - 2年目以降（切替畑と永久耕作地含む）
  - 耕作放棄初期（タケの再成長）
- 集落
  - ホームガーデンを有する/有しない集落
- 開放水域

DIGITALGLOBE
DEFINIENS
Understanding Images

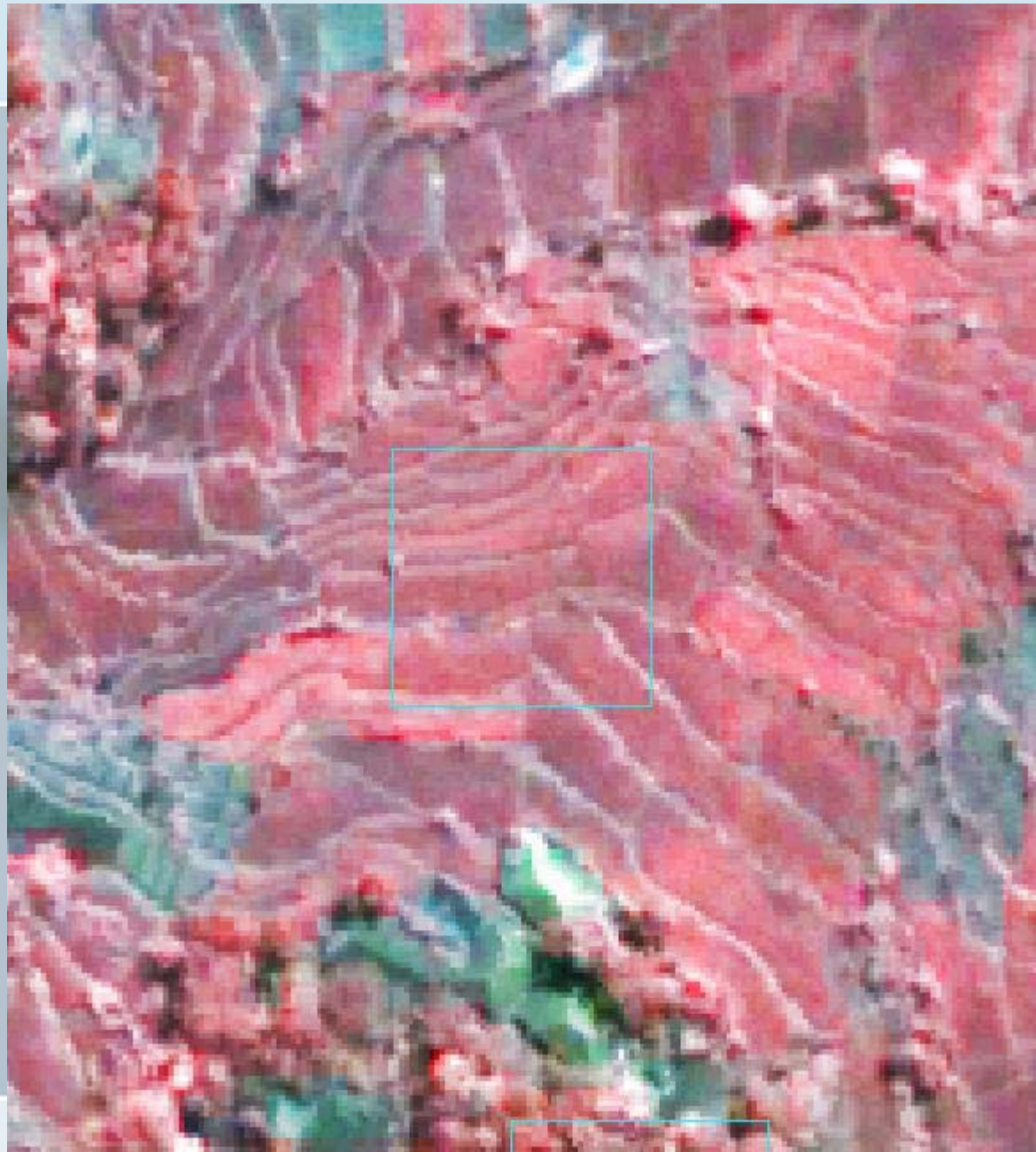
樹木園

タケ林

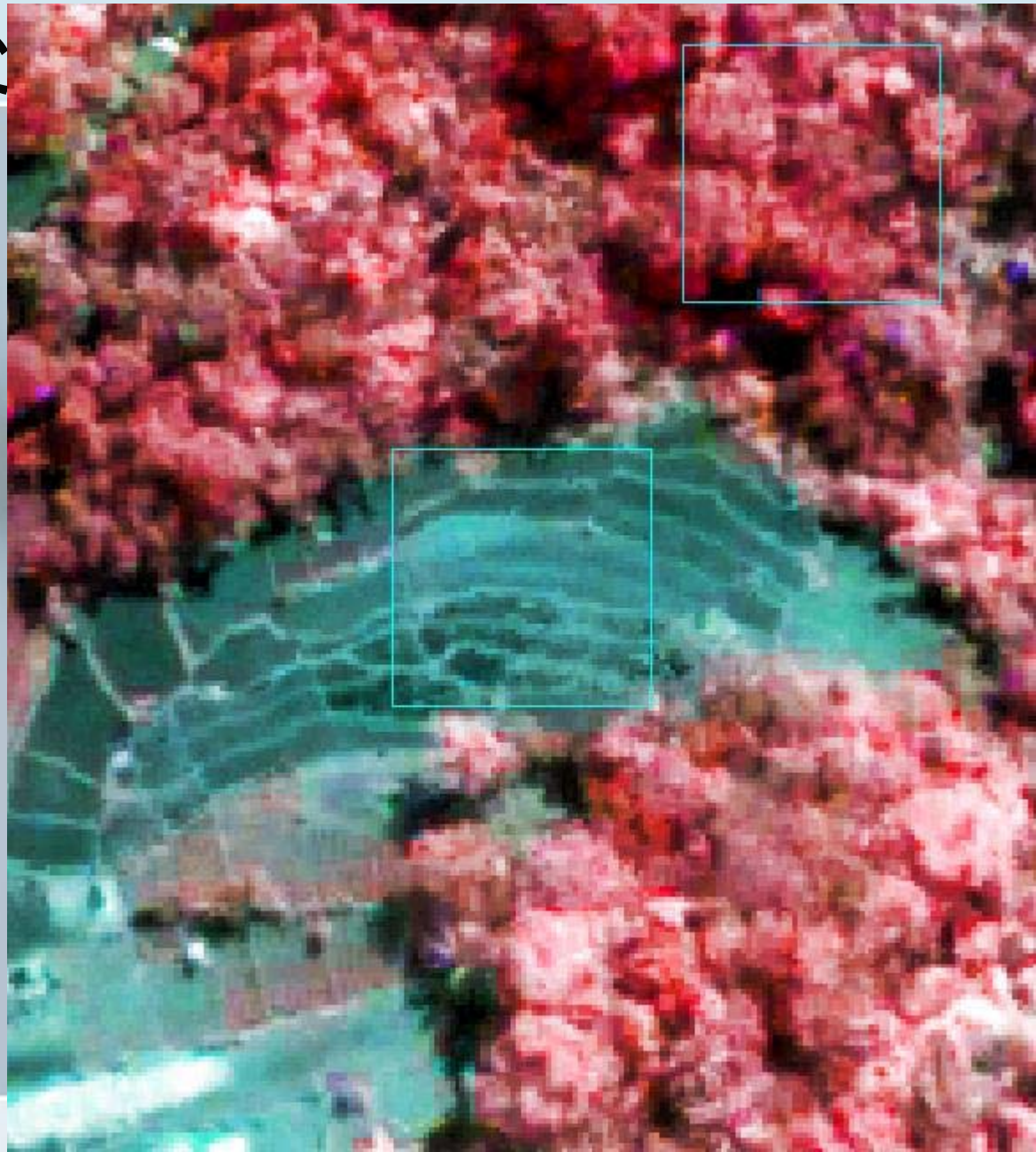
大区画水田
収穫前

大区画水田
移植前後

棚田
収穫前



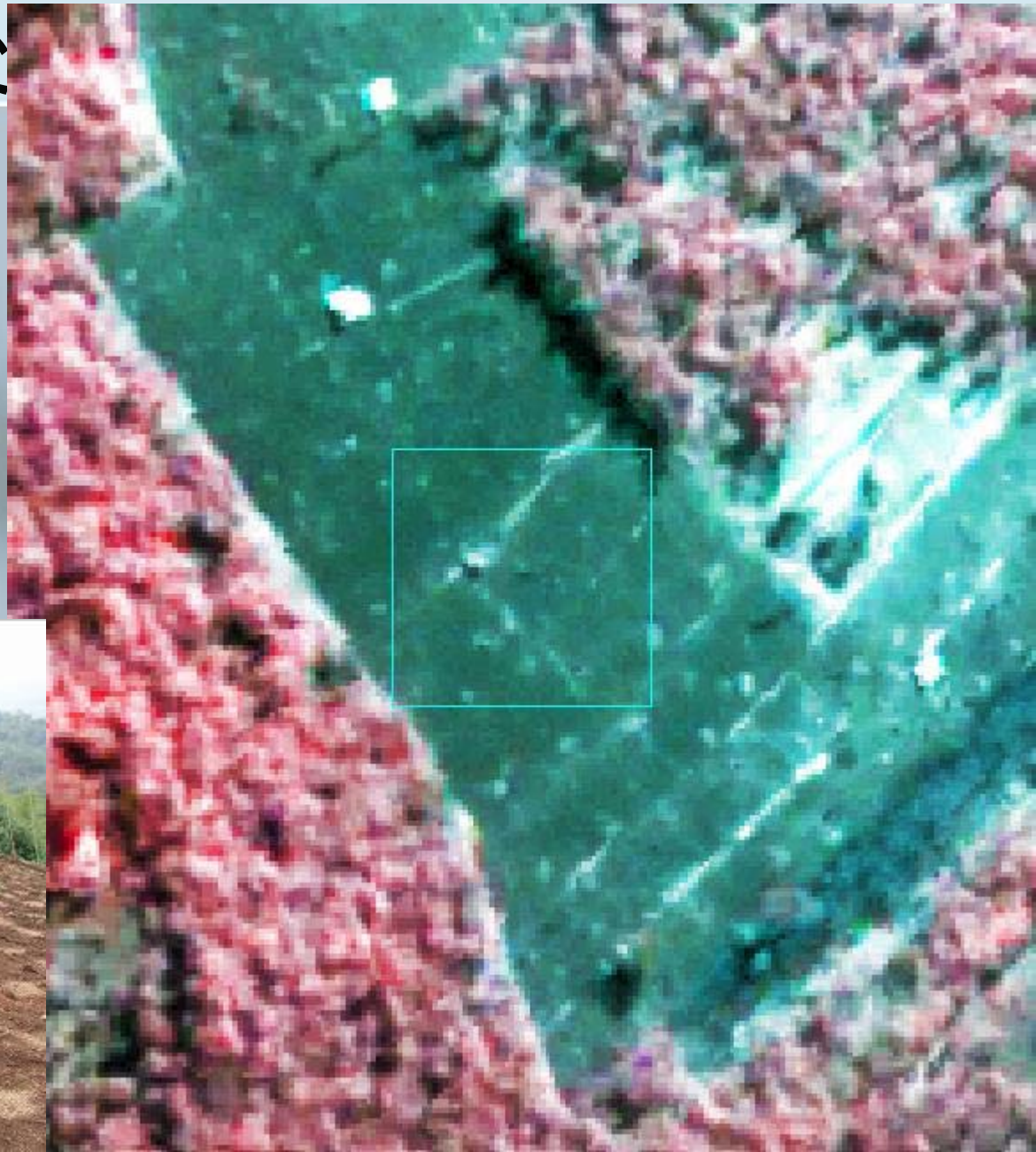
棚田
移植前後



DIGITALGLOBE
DEFINIENS
Understanding Images

畑地
切替中

畑地
切替直後



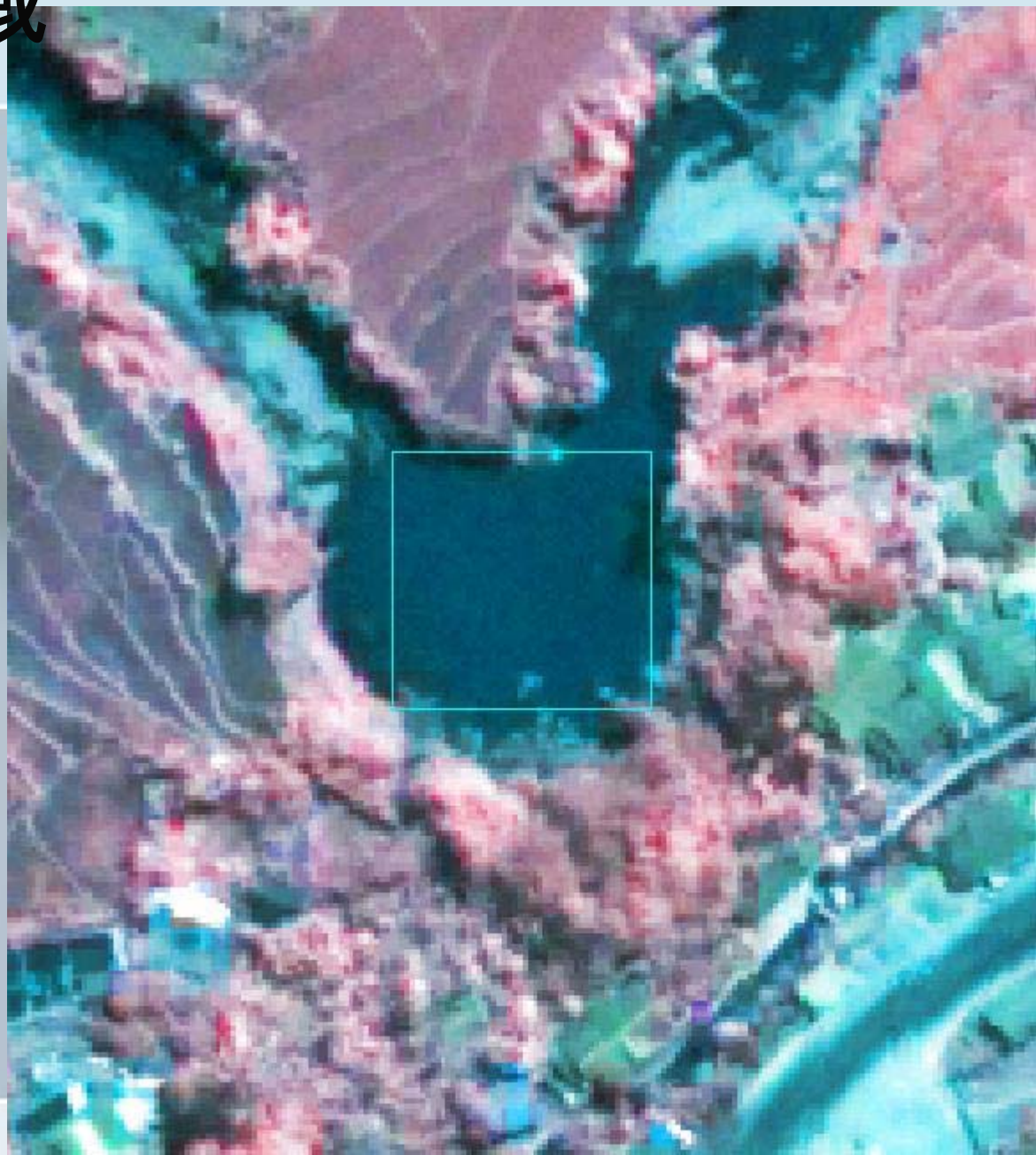
畑地
耕作2年以降

DIGITALGLOBE

DEFINIENS
Understanding Images

畑地
放棄地

開放水域



ホームガーデン集落

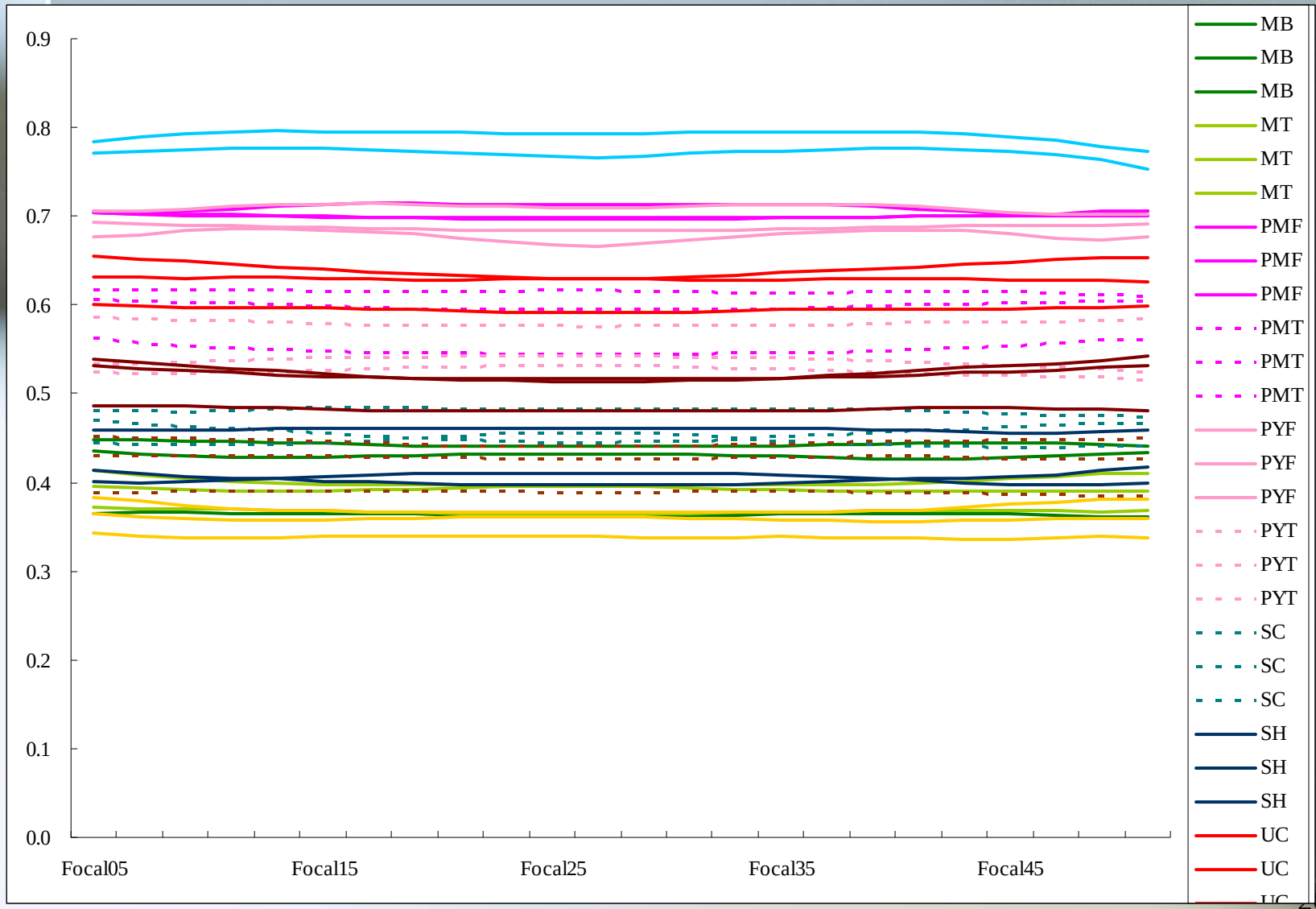
市街地
集落

解析の目的

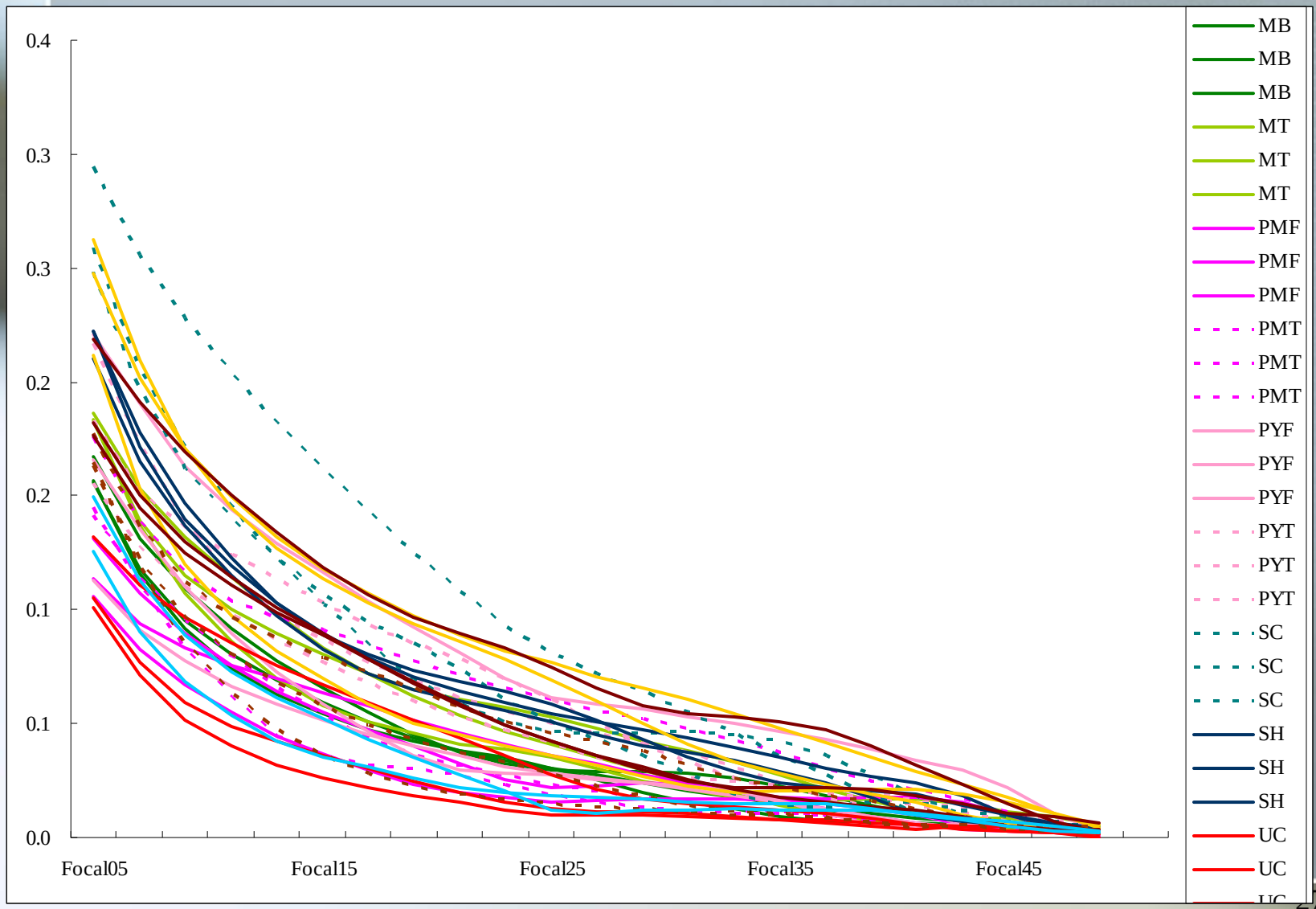
- Per-pixel分類の向上を目指して
  - Texture情報の適正なwindow sizeは？
    - GLCMでwindow sizeを変えたときに，各土地被覆タイプの平均値およびバラツキはどのような挙動を示すか（cf Puissant et al. 2005）？
    - Semivariogramで，rangeがどの程度か（cf Dell'Acqua et al 2006）？
  - どのtexture情報が有効？
    - CART（Classification and regression tree）を用いて，土地被覆の分類に有効な情報および閾値の抽出

    →画像分類のrule setとして活用可能

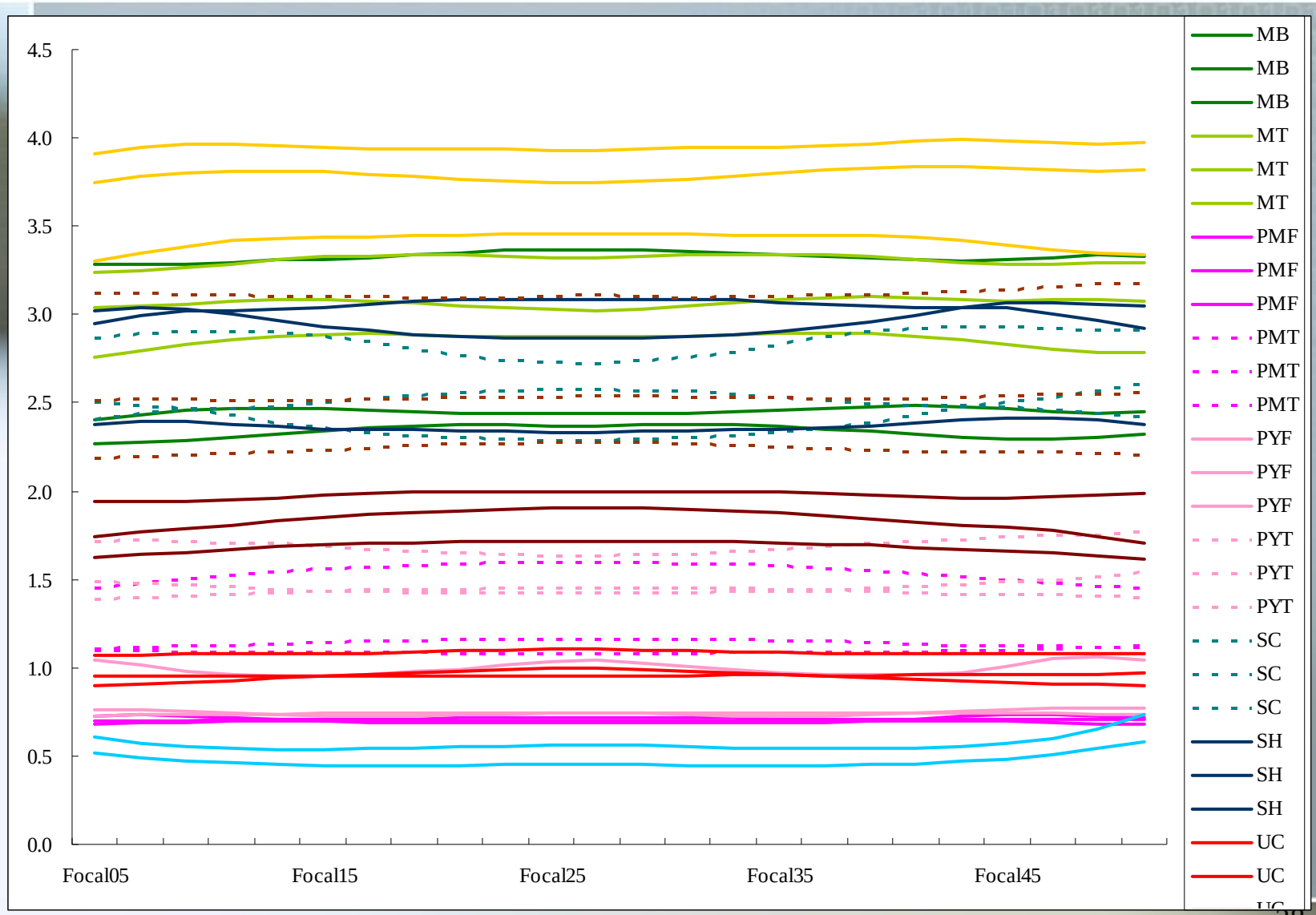
結果：GLCMとwindow size Homogeneity



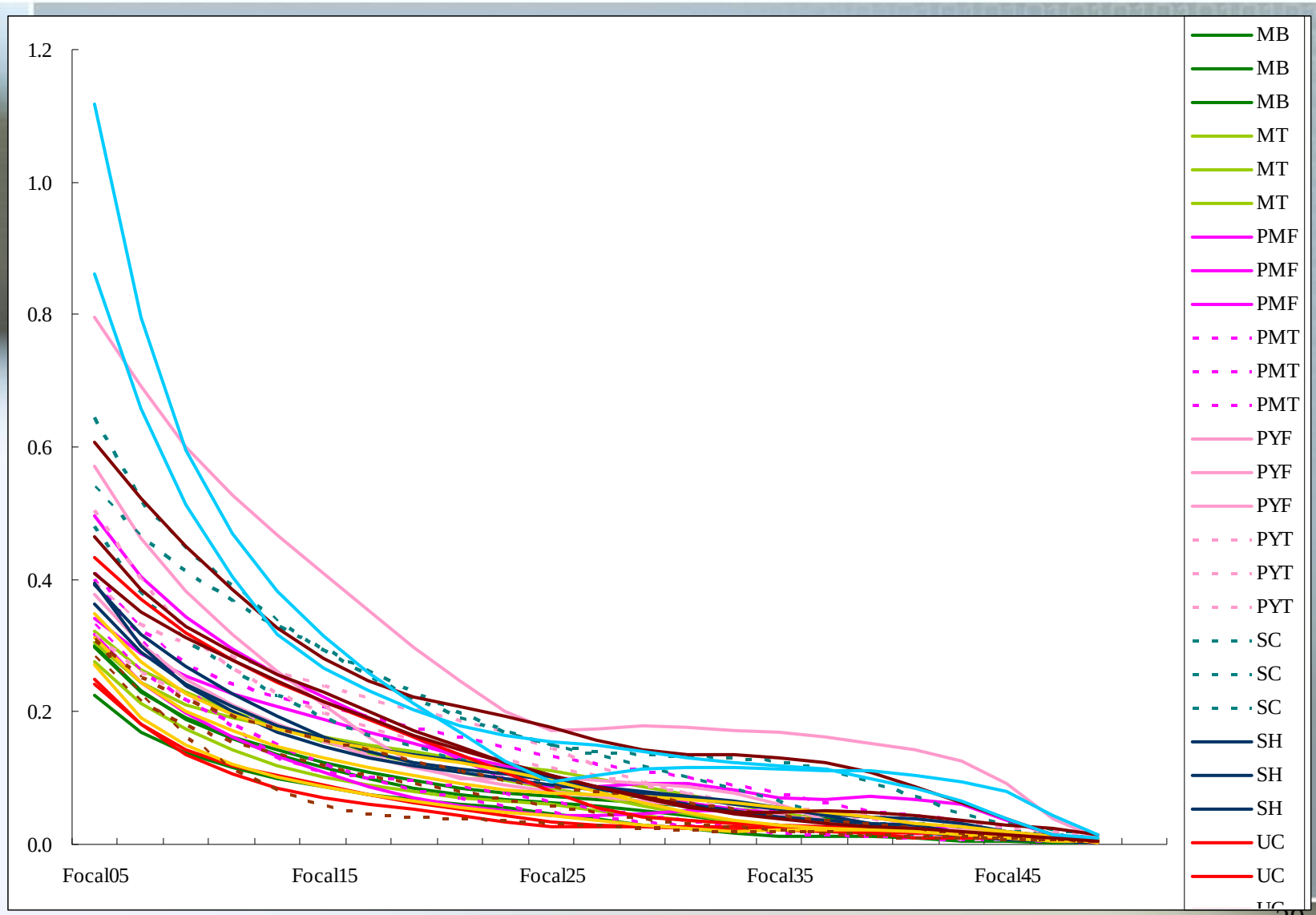
結果：GLCMとwindow size Homogeneity



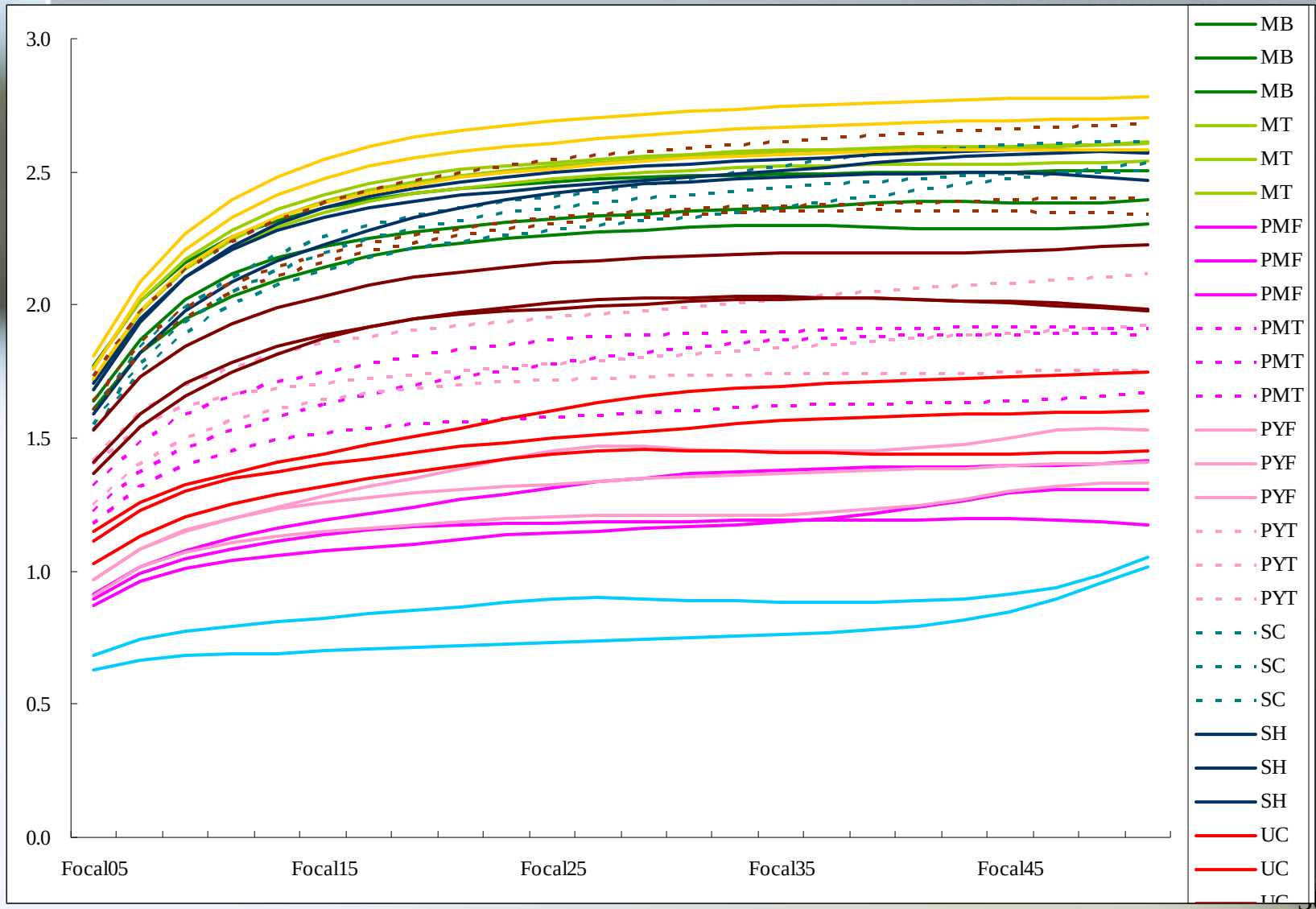
結果：GLCMとwindow size Dissimimarity



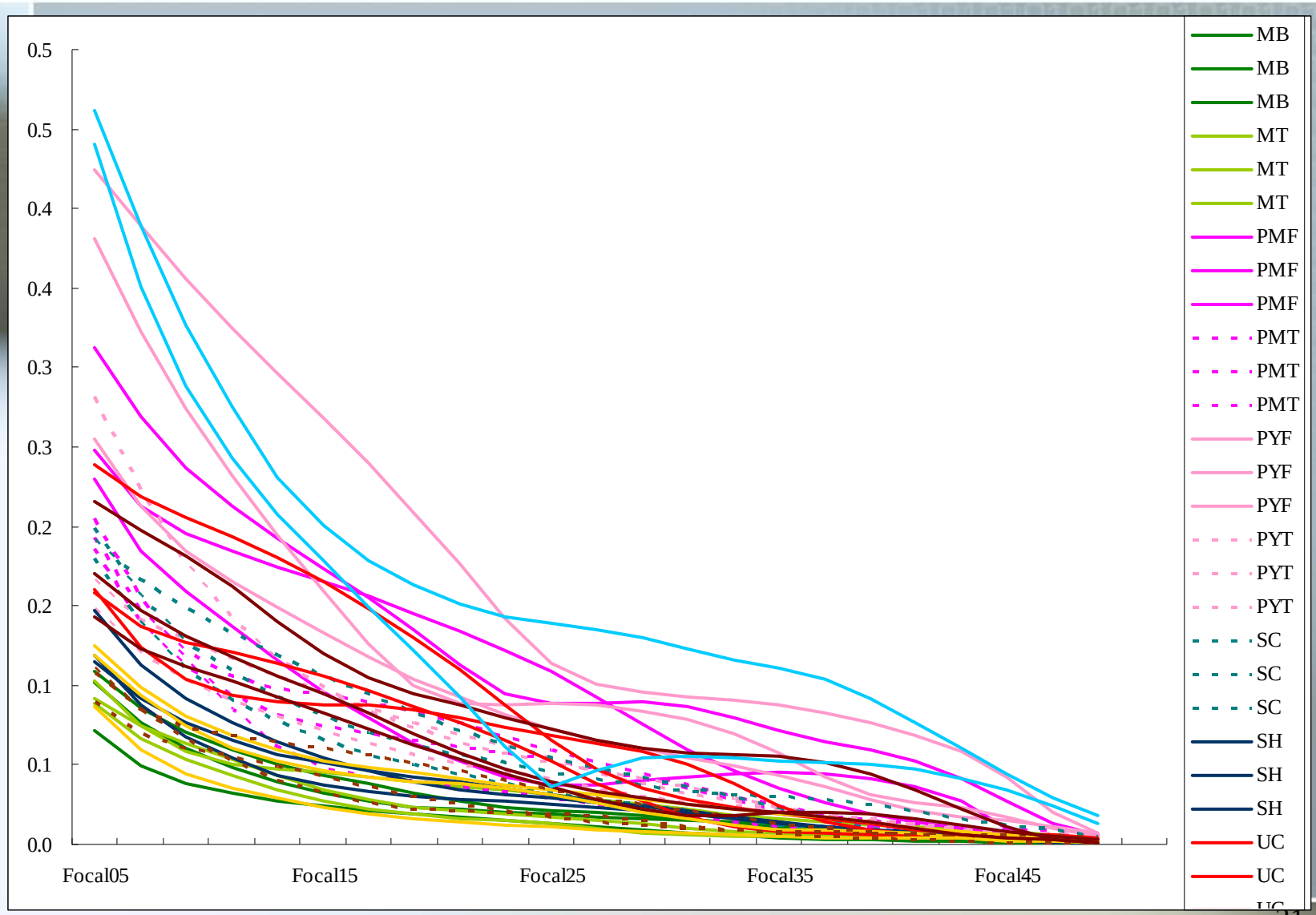
結果：GLCMとwindow size Dissimimarity



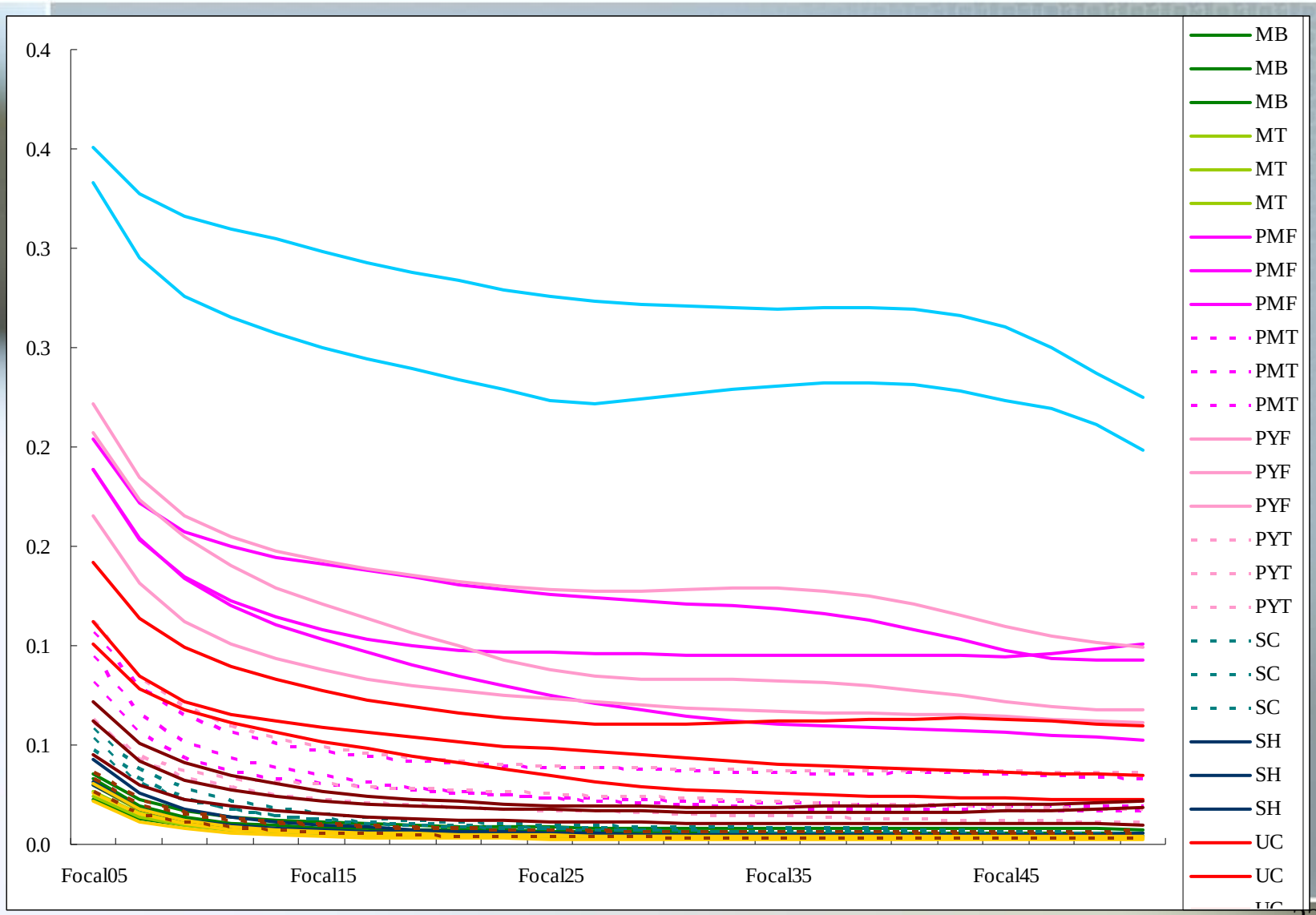
結果：GLCMとwindow size Entropy



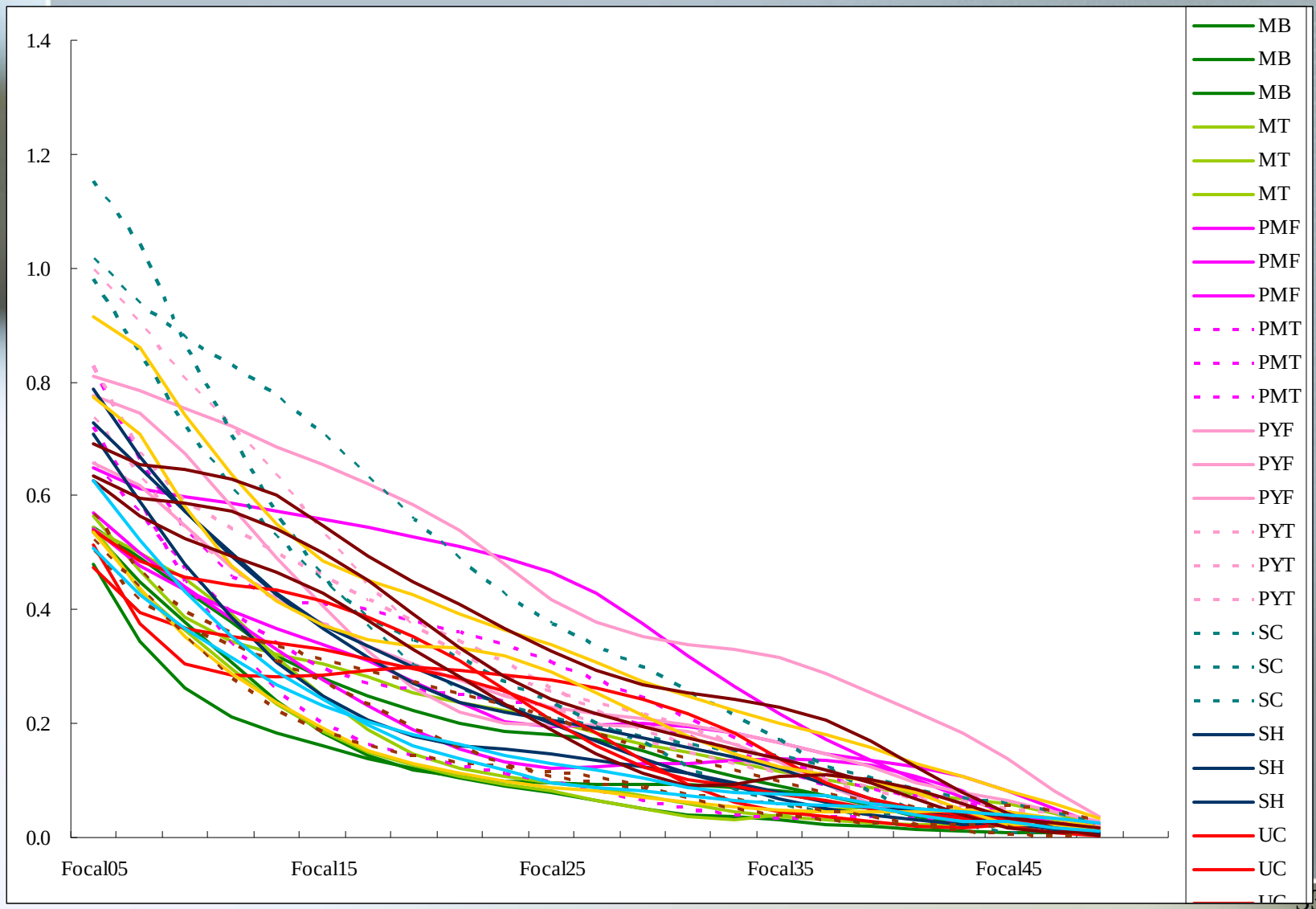
結果：GLCMとwindow size Entropy



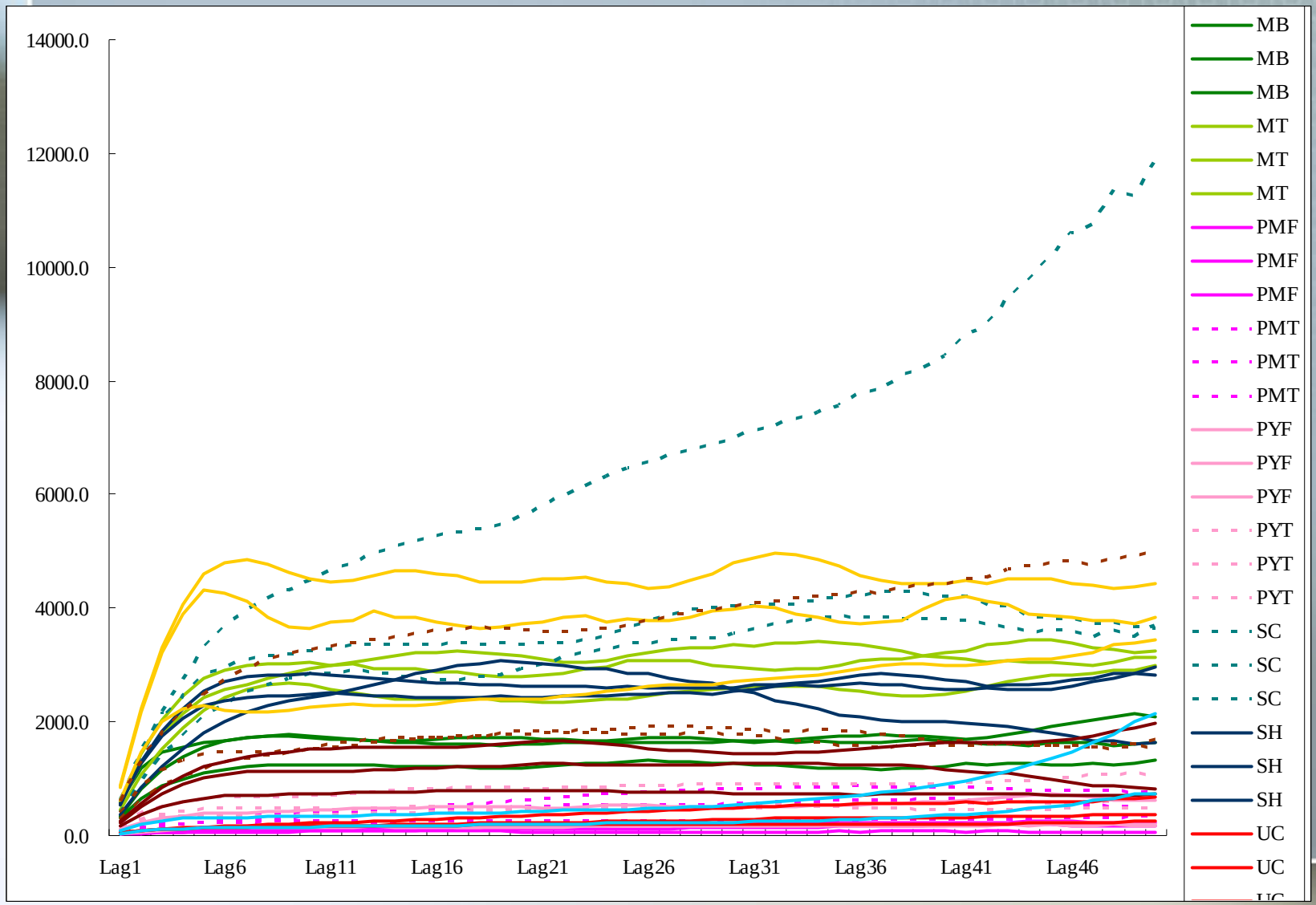
結果：GLCMとwindow size Moment



結果：GLCMとwindow size Moment



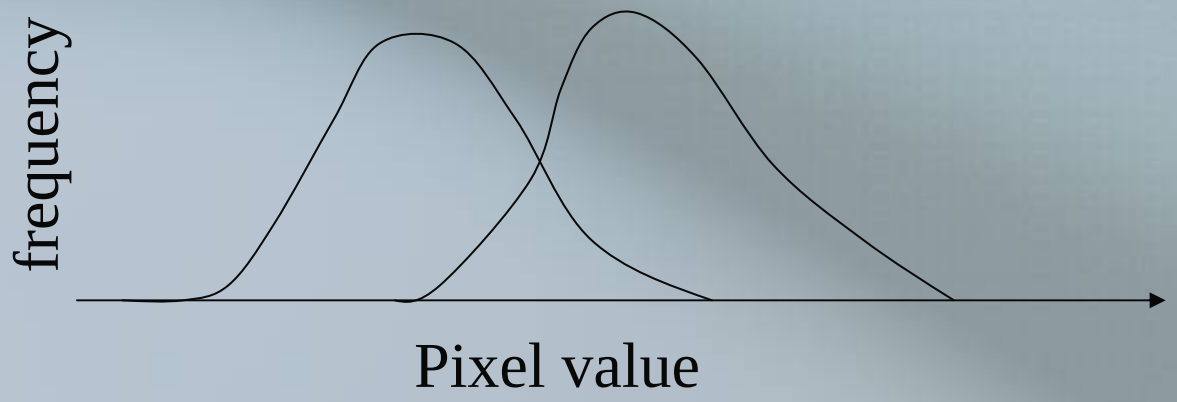
結果:
SemigariogramとWindow size



HDIによる最適window size

- HDI (Histogram Distance Index)

$$HDI = \sum_{m=0}^{nc} \sum_{n=0}^{nc} \left(1 - 2 \sum_{\Omega m, \Omega n} \min(f_m(\vec{x}), f_n(\vec{x})) \Big/ \sum_{\Omega m, \Omega n} (f_m(\vec{x}) + f_n(\vec{x})) \right)$$



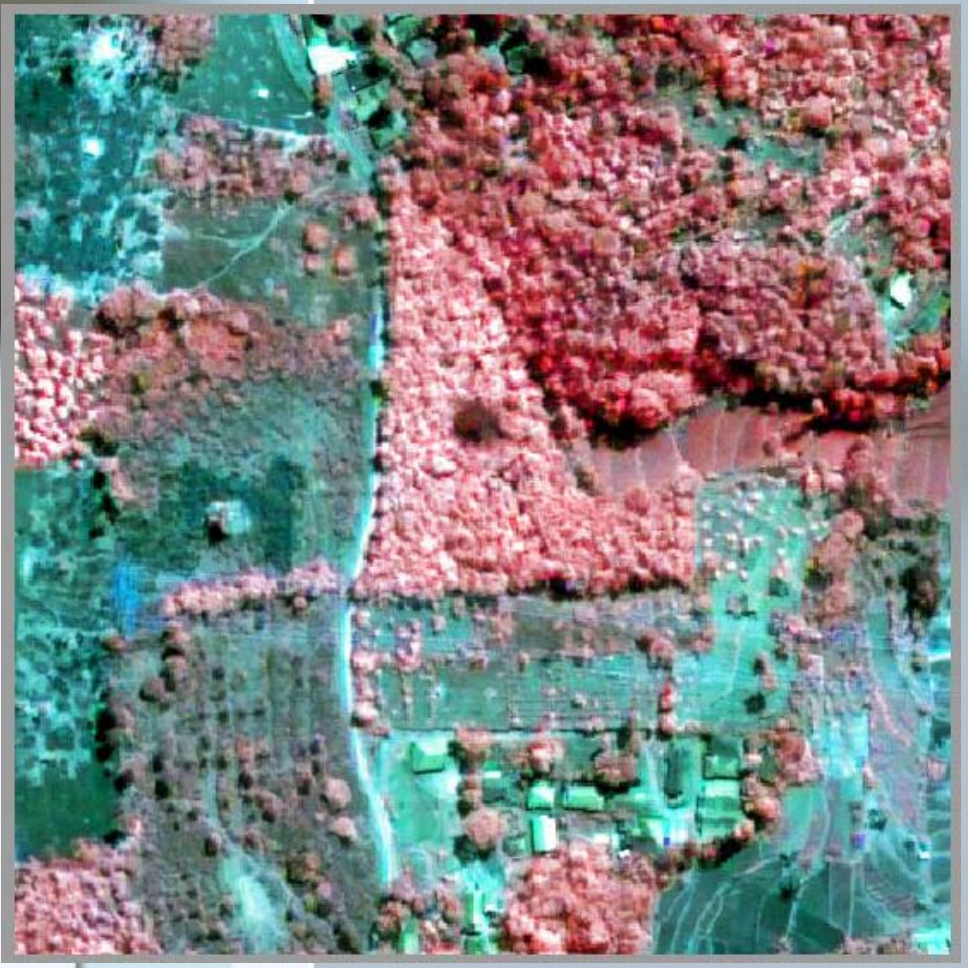
HDIの結果

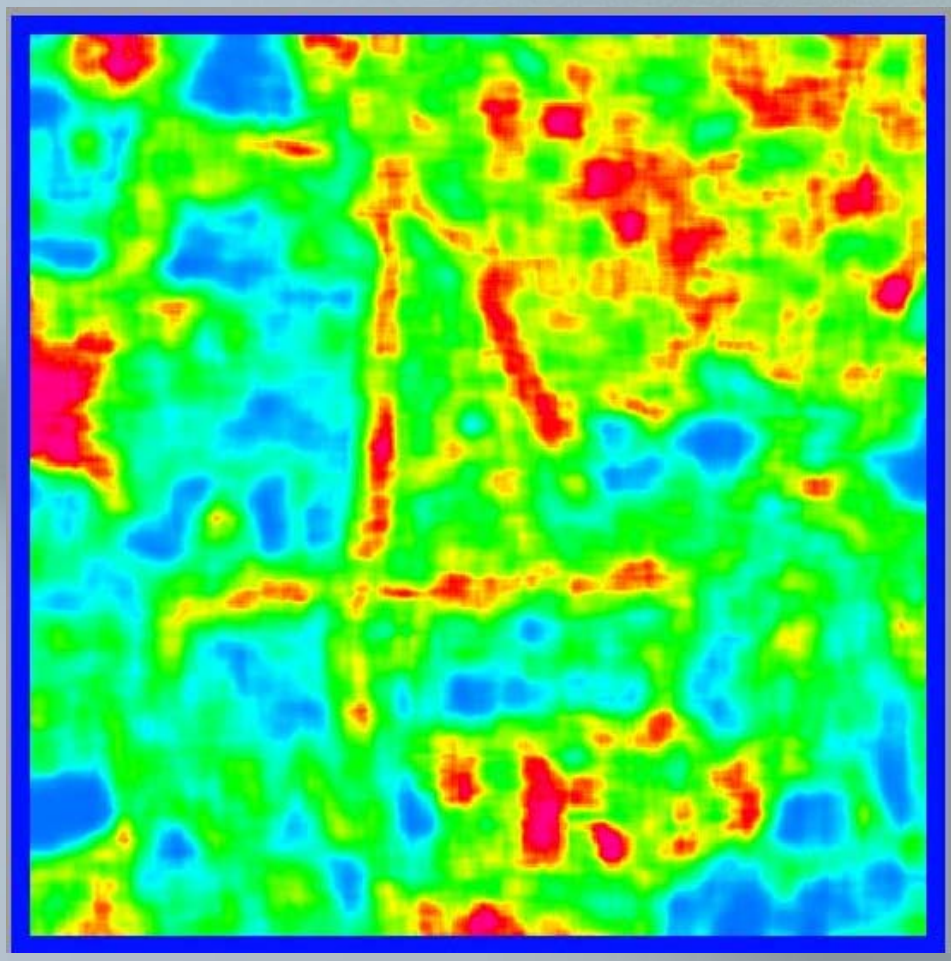
- 大きければ大きいほどよい

小括:テクスチャ解析の適正window size

- GLCMによるテクスチャ情報
  - Window sizeの大小にかかわらず同一土地被覆の平均的な情報量は安定
  - これはsemivariogramによる解析結果でも同様で，11ピクセル以上であるとほぼ安定
  - window sizeを大きくした方が同一土地被覆での情報量の変動が小さくなる
  - つまり，テクスチャ情報を算出するには大きなwindow sizeを設定するべき←私の例だと21×21（12.6×12.6m）以上
- 問題点
  - Window sizeが大きくなると，pixelベースの解析では異なる土地被覆を含むwindowが増えるため，境界部分の「ゴミ」情報が多くなってしまう

実際のテクスチャ情報画像



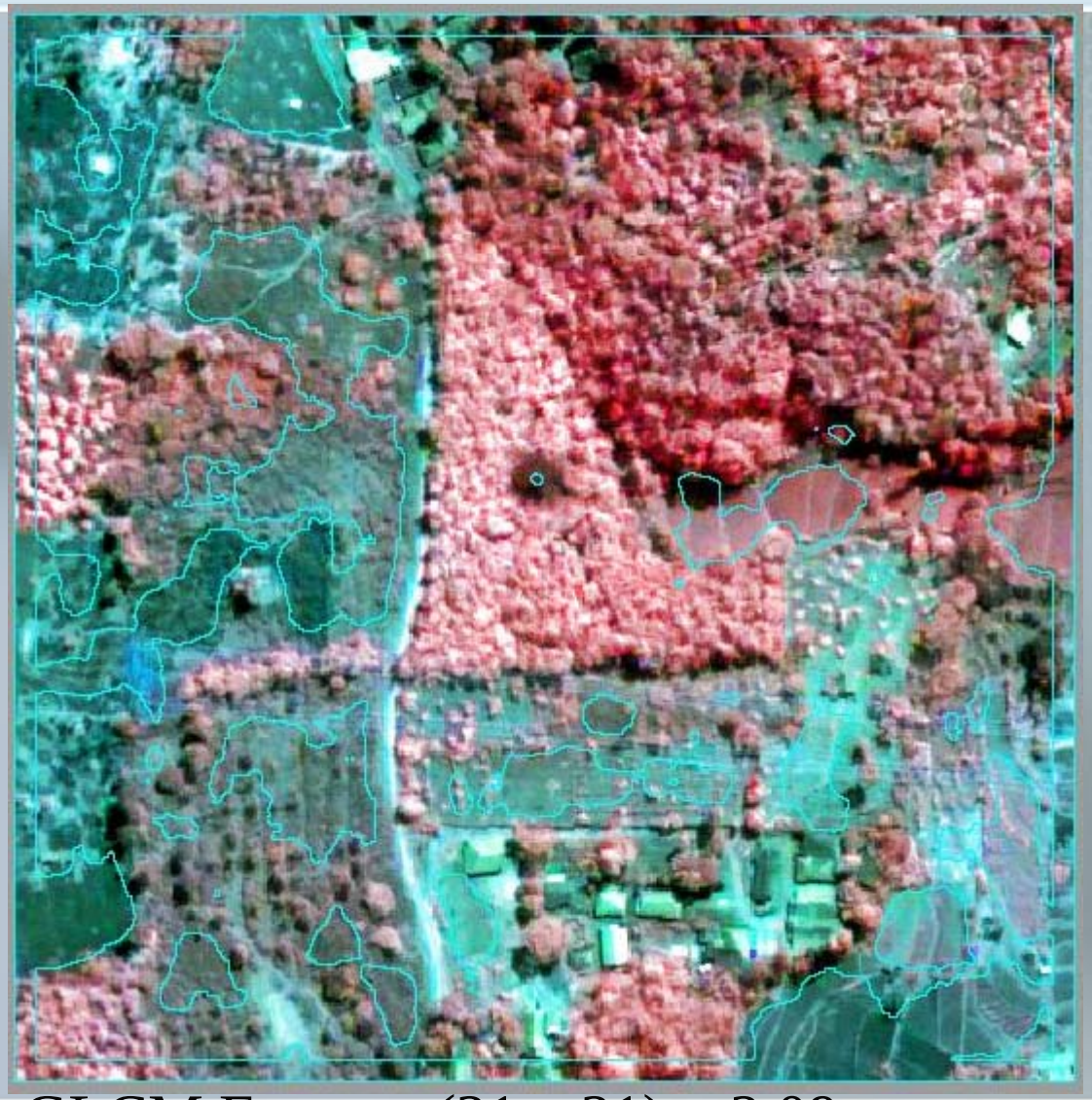


被覆分類に有用なテクスチャ

- CARTの分類樹木による土地被覆分類の鍵となる情報量の抽出
  - Window size：21×21ピクセル
    - GLCM homogeneity（window内の中央セル値）
    - GLCM dissimilarity（同上）
    - GLCM entropy（同上）
    - GLCM moment（同上）
    - Band 1から4のwindow内平均値と標準偏差
  - それぞれwindowが重ならない上記サイズのサンプルエリアを，各土地被覆タイプで30前後のサンプル用意

CARTの結果

閾値は使える？



GLCM Entropy (21 x 21) < 2.09

CARTによるエラーマトリックス

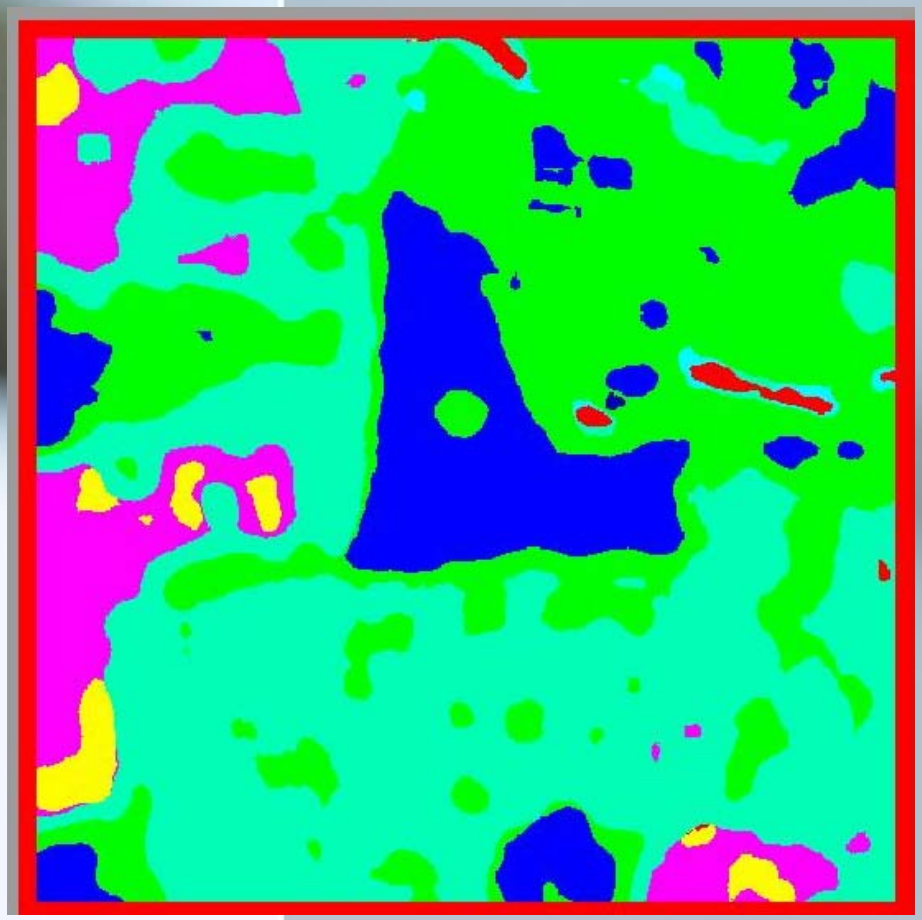
スペクトル＋テクスチャ

	タケ林	樹木園	水田	集落	市街地	畑地	切替直後	耕作放棄直後	開放水域	合計	User's Ac
タケ林	37	4	1	0	0	2	2	0	0	46	80.4%
樹木園	2	35	0	0	0	0	0	0	0	37	94.6%
水田	3	0	168	1	0	10	2	0	0	184	91.3%
集落	0	0	0	40	3	0	0	4	0	47	85.1%
市街地	0	6	0	2	48	0	8	0	0	64	75.0%
畑地	0	0	3	5	0	71	0	0	0	79	89.9%
切替直後	2	3	0	0	1	1	28	0	0	35	80.0%
耕作放棄直後	0	0	0	0	0	0	0	20	0	20	100.0%
開放水域	0	0	0	0	0	0	0	0	8	8	100.0%
合計	44	48	172	48	52	84	40	24	8	520	
Prod's Acc	84.1%	72.9%	97.7%	83.3%	92.3%	84.5%	70.0%	83.3%	100.0%		87.5%

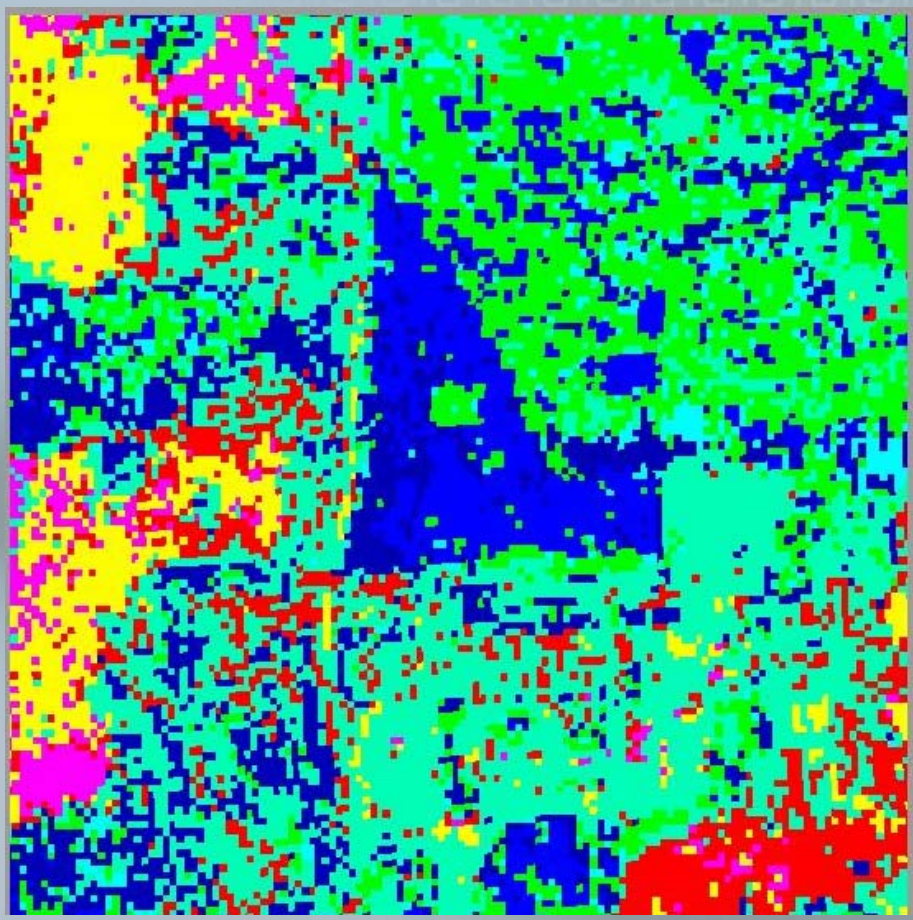
スペクトルのみ

	タケ林	樹木園	水田	集落	市街地	畑地	切替直後	耕作放棄直後	開放水域	合計	User's Ac
タケ林	27	4	1	0	0	0	2	0	0	34	79.4%
樹木園	4	32	2	0	2	0	1	0	0	41	78.0%
水田	12	4	163	0	0	13	1	0	0	193	84.5%
集落	0	0	1	41	4	1	0	1	0	48	85.4%
市街地	0	6	1	0	45	0	4	0	0	56	80.4%
畑地	0	0	2	4	0	70	0	1	0	77	90.9%
切替直後	1	2	2	0	1	0	32	0	0	38	84.2%
耕作放棄直後	0	0	0	3	0	0	0	22	0	25	88.0%
開放水域	0	0	0	0	0	0	0	0	8	8	100.0%
合計	44	48	172	48	52	84	40	24	8	520	
Prod's Acc	61.4%	66.7%	94.8%	85.4%	86.5%	83.3%	80.0%	91.7%	100.0%		84.6%

サンプルエリアでの分類結果



テクスチャ情報による分類結果



スペクトルバンドによる分類結果

総括：今後の展開について

- テクスチャ情報
  - Per-pixelでも分類精度の向上に寄与
  - とくに耕作地（段階の違う水田や畑地）とその他を分類するのに貢献
- 分類方法
  - テクスチャ情報については大きなwindow sizeを用意した方がよい
  - ただし，土地被覆境界部分の扱いが問題
  - テクスチャ情報を得るための，適切なセグメンテーションサイズを決定し，object-based分類を行うのがやはり妥当