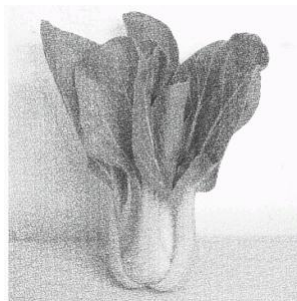


# 作成手順を考慮した 切り絵制作支援ツール

システム情報工学研究科 CS専攻  
博士前期課程1年 201120729 中島健次郎

# はじめに(1/3)

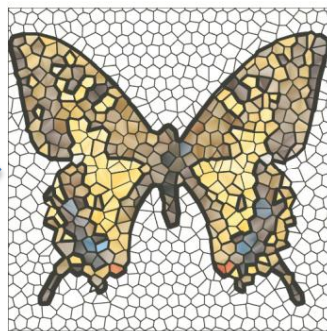
- ノンフォトリアリスティックレンダリング  
– 写実的でない絵画技法等の計算機上での再現



鉛筆画風画像生成



油絵風画像生成



ステンドグラス風画像生成



切り絵風画像生成

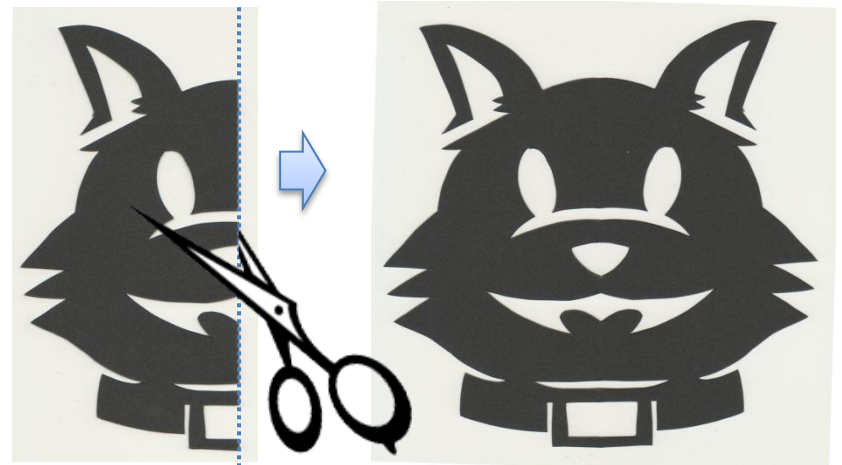
# はじめに(2/3)

- 切り絵

- 紙の不要部分を切り抜いて作り上げる絵画技法
- 黒い紙を切り抜き白い台紙に貼り白黒で表現
- 紙を折りハサミで切り抜く「紙切り」とは異なる



切り絵作品



紙切り作品

# はじめに(3/3)

- 切り絵に関する既存研究

- Computer Generated Papercutting [Xu et al. , 2007]
- Artistic Paper-Cut of Human Portraits [Meng et al. , 2010]



画像から切り絵

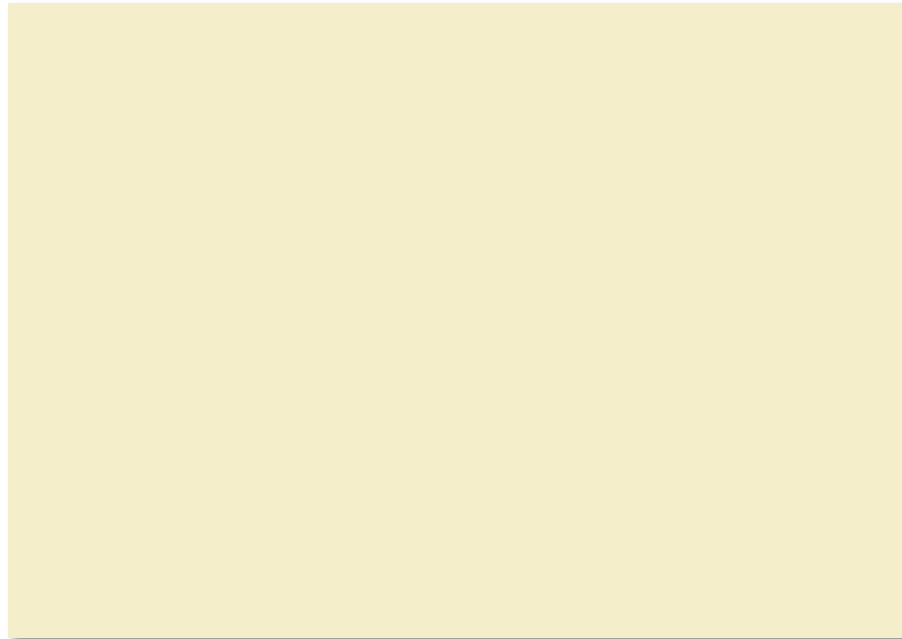


顔画像から切り絵

既存研究では実際の制作手順が考慮されていない

# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵



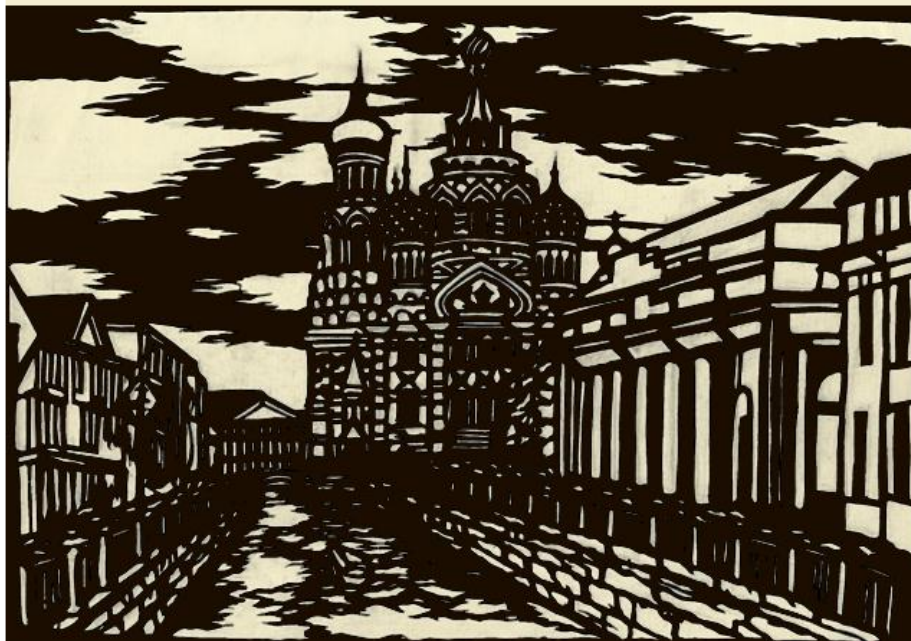
# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵



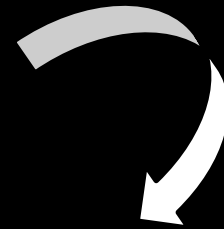
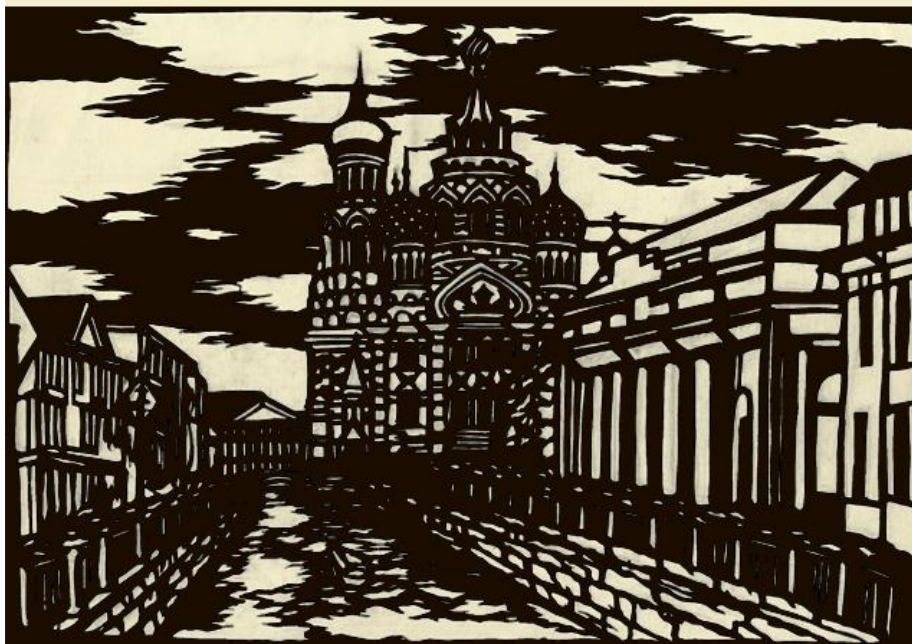
# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵



# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵



# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵



# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵



# 作成手順

- デッサン ⇒ 下絵の作成 ⇒ 切り ⇒ 台紙貼り
  - 下絵とは、切り抜く領域を明確にするため白黒塗り分けられた絵

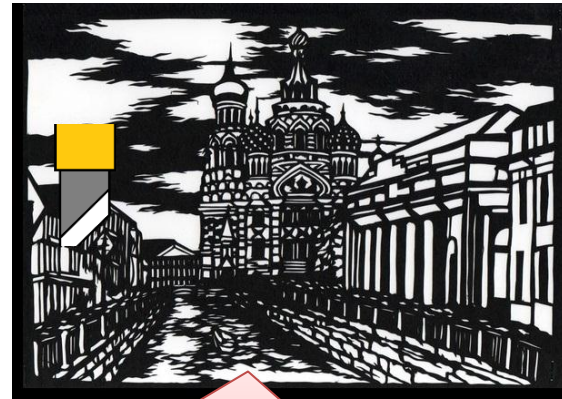


# 制作上の課題・問題

- 下絵の黒い領域は一繋がりの必要あり
- 切りっていく際中、部分的に紙がずれる  
– ずれて紙の細部がちぎれるなどの影響



オリジナルなデザインには  
経験と知識が必要



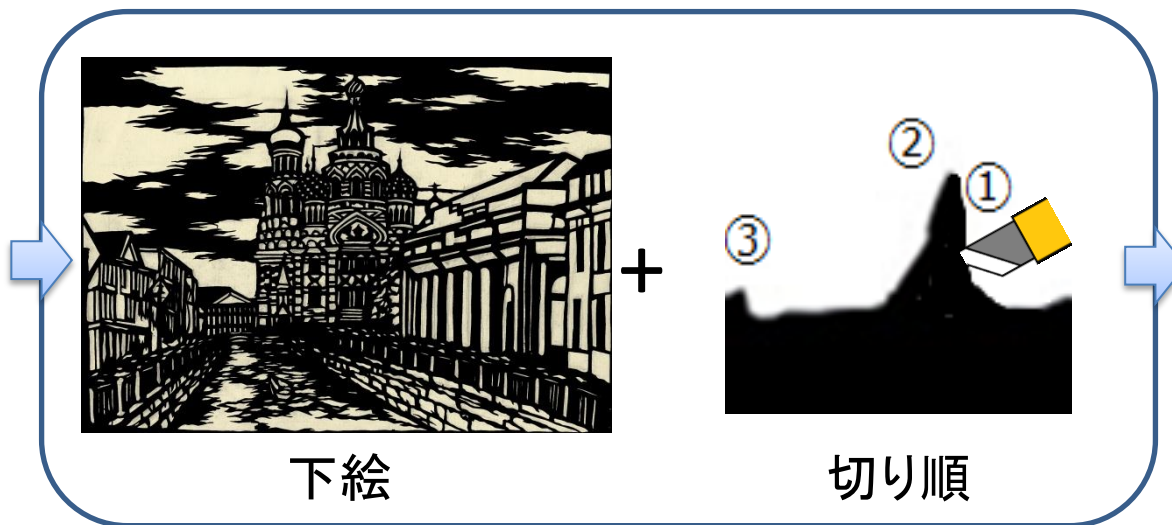
適切な順序で切るには  
経験が必要

# 研究目的

- 作成手順を考慮した上で切り絵の制作を支援するツールの提案
  - スケッチ画からの下絵作成
  - 適切な切り順の提示



スケッチ画像



# 関連研究(1/2)

## Computer-Generated Papercutting

[Jie Xu, Craig S. Kaplan, Xiaofeng Mi, 2007]

- 写真などから黒が一繋がりの下絵を作り出す手法を提案



入力: 写真



出力: 下絵

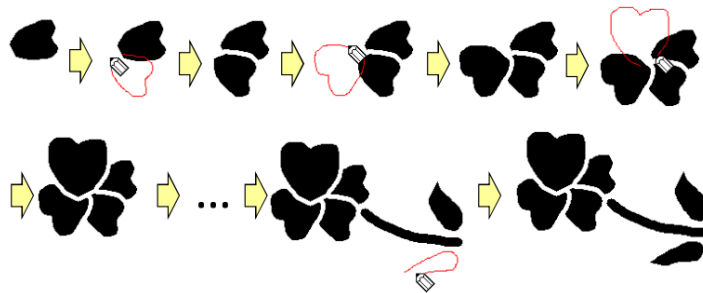
本研究では  
入力をスケッチ画

# 関連研究(2/2)

## Holly: A Drawing Editor for Designing Stencils.

[Igarashi, Y. and Igarashi, T., 2010]

- ステンシルのデザインのためのドローエディタを提案
- 図柄の形状の制約は切り絵と同じ
- 切り抜く領域をユーザが自由に描くことができる



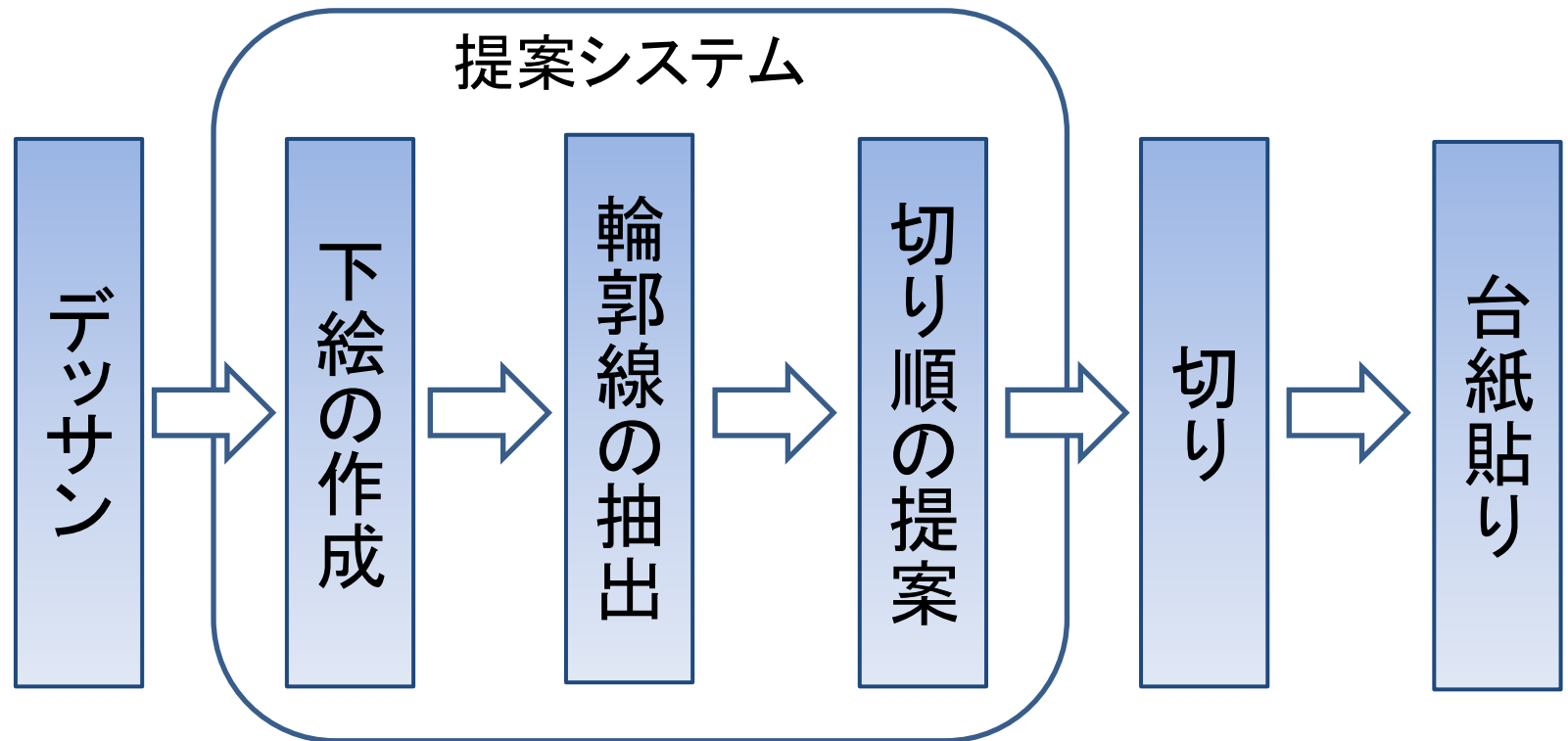
ドローエディタによるデザイン例

領域の周りは  
切り抜かない

本研究では  
切り絵の表現を考慮

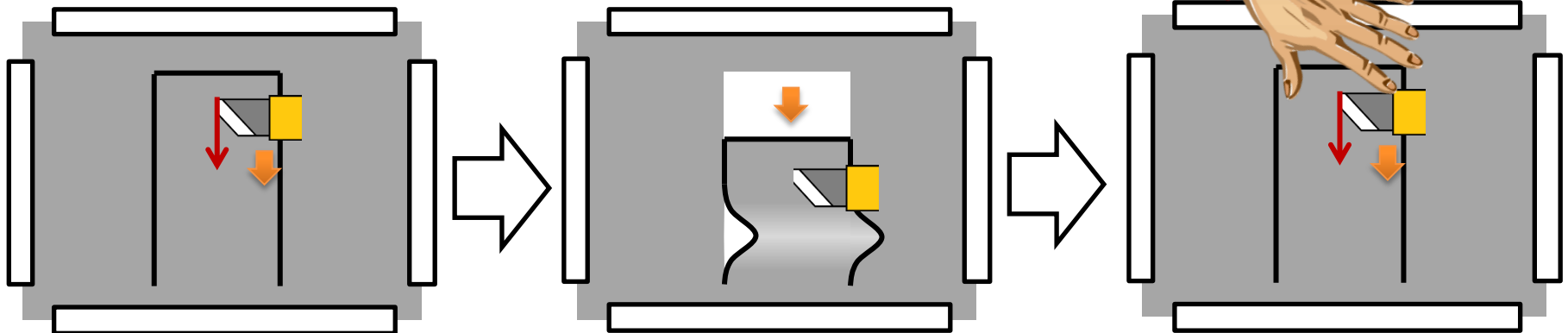
# 提案システムの概要

- スケッチ画を入力として下絵を作成し、切り順を提示した画像を出力する



# 切るときはずれ

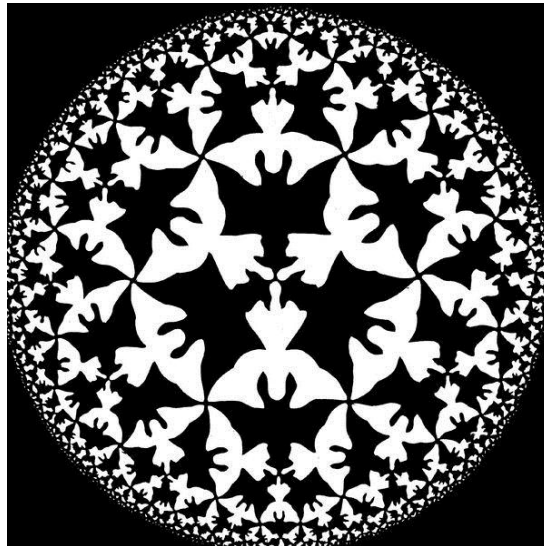
- 切る対象の紙の端は固定される
  - 紙全体がずれることはない
  - 部分的には**ずれてしまう**



ずれる場合の例

# ずれの影響

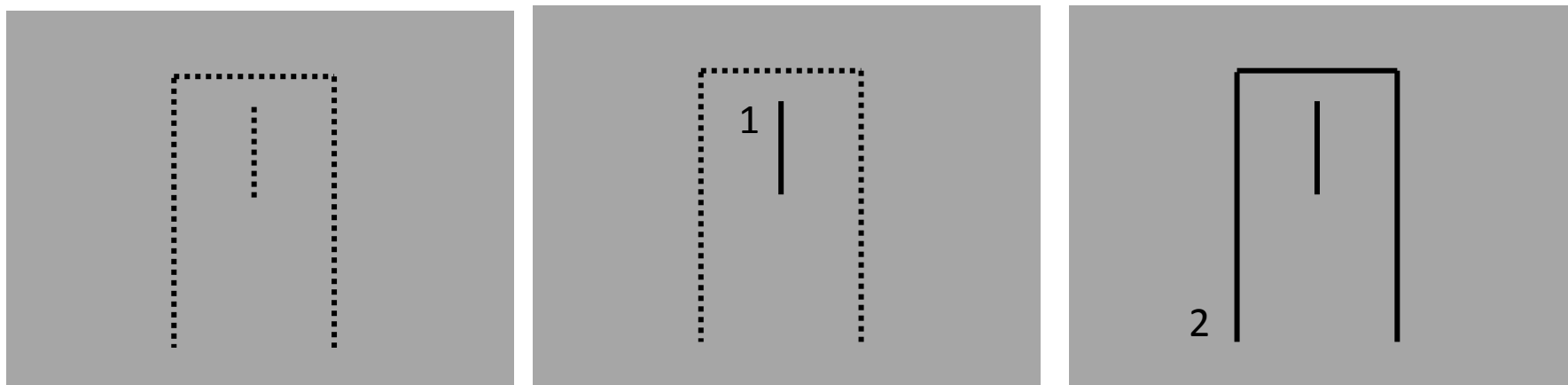
- 複雑な作品において部分的なずれが多発
- 紙がずれて一部がちぎれたり、作品の質が下がる
- 細かい部分は手で押さえるのも困難
- 手で押さえる作業で、完成までの時間の増加



ちぎれやすい  
複雑な作品例

# 切り順によるずれの回避

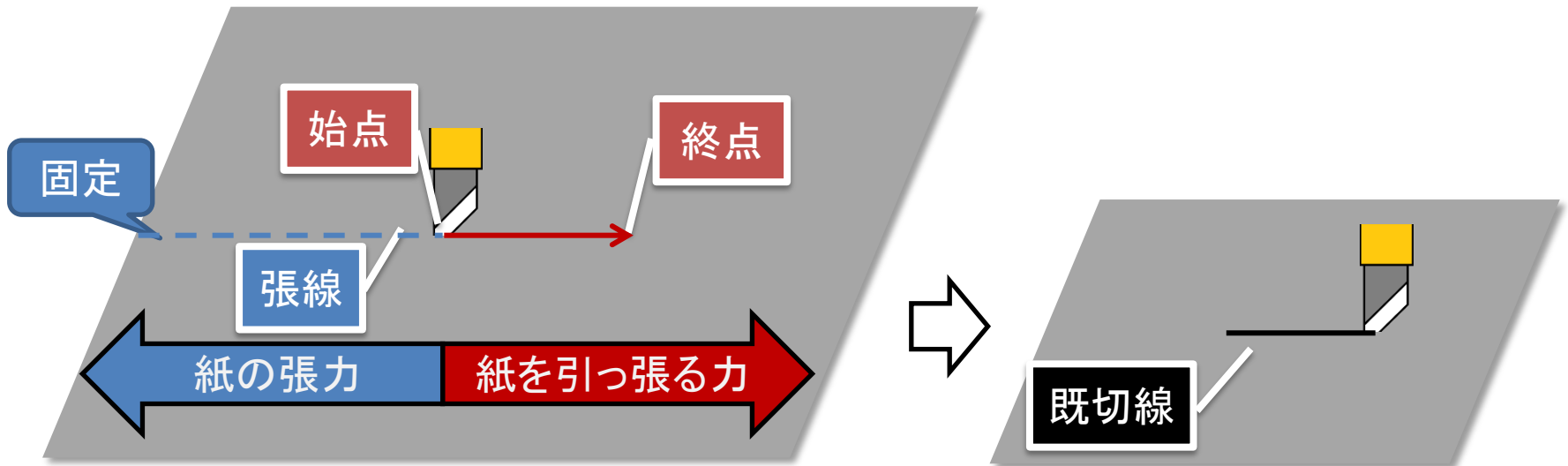
ずれは切り順により回避可能



ずれの回避

切るべき線が与えられたら  
ずれることのない切り順を提案

# 紙の切れ方



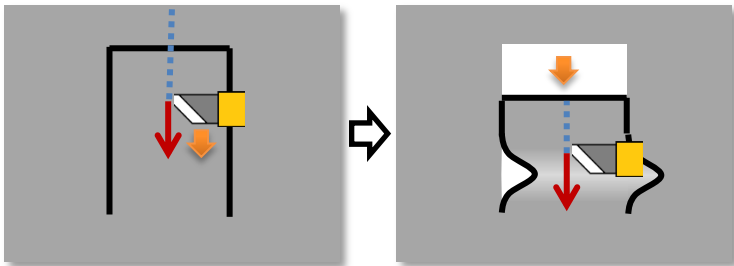
力が釣り合い紙がずれない

- : 切る線(方向付)
- - : 紙を引っ張る力が働く線
- : 既に切られた線

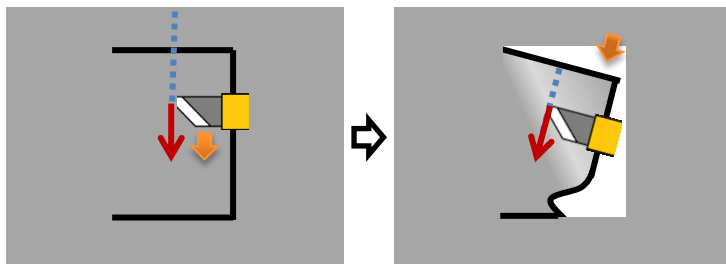
- ... (始点から終点)
- ... 張線
- ... 既切線

# ずれる状況の考察(1/4)

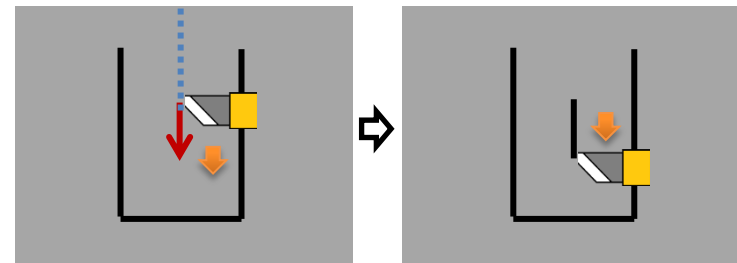
- 張線・既切線的位置から分類した以下の状況を比較・検討する



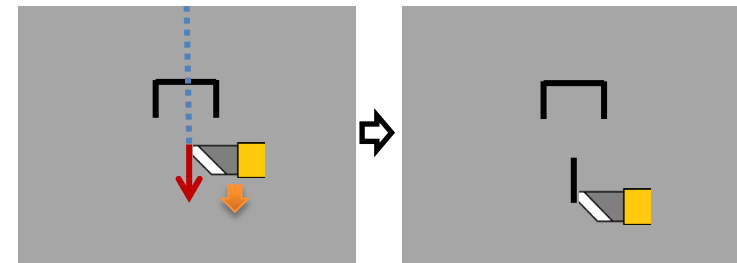
(a) ずれる



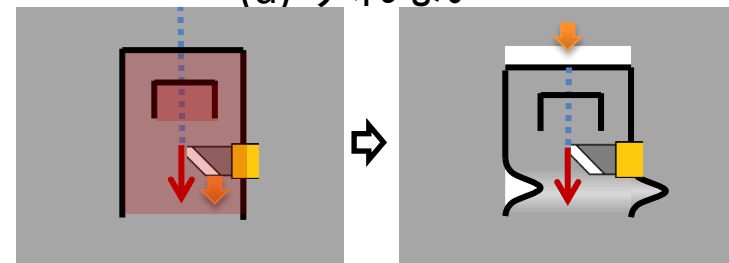
(b) ずれる



(c) ずれない



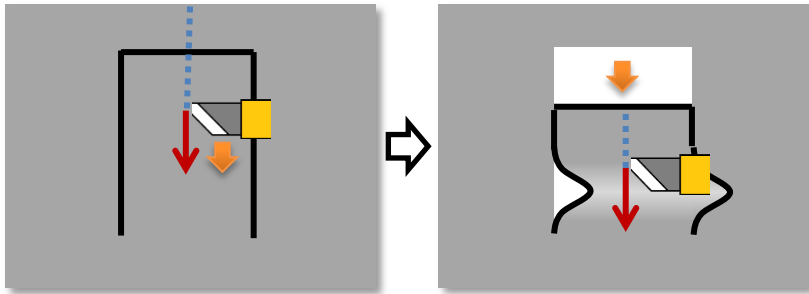
(d) ずれない



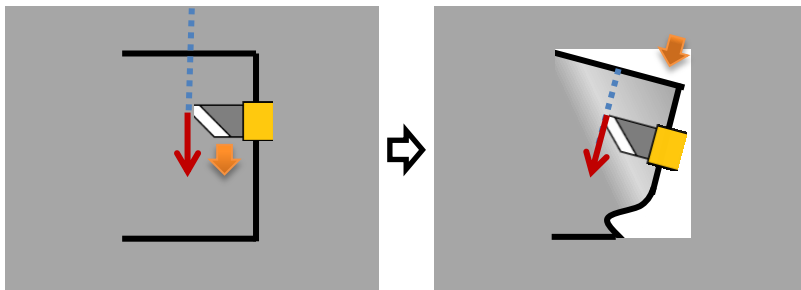
(e) ずれる

# ずれる状況の考察(2/4)

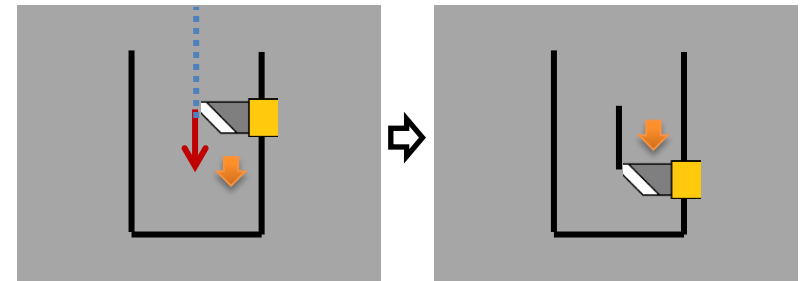
- (a),(b),(c)において始点が既切線に3方囲まれている



(a) 張線と既切線が交差する  
→ずれる

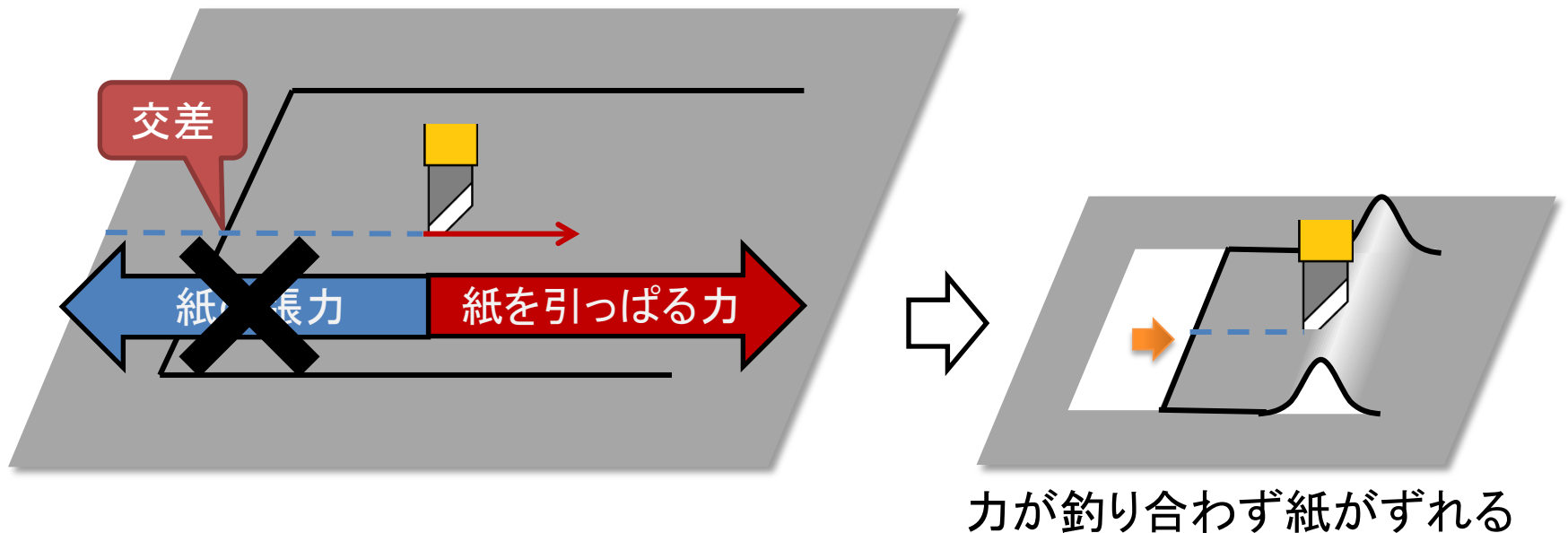


(b) 張線と既切線が交差する  
→ずれる



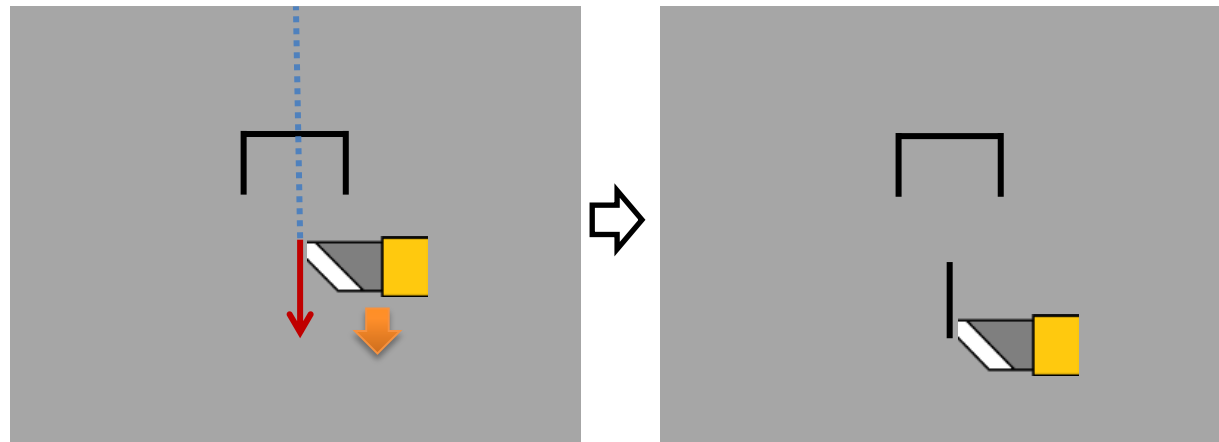
(c) 張線と既切線が交差しない  
→ずれない

# 張線と既切線の交差



張線と既切線が交差しなければ  
ずれない

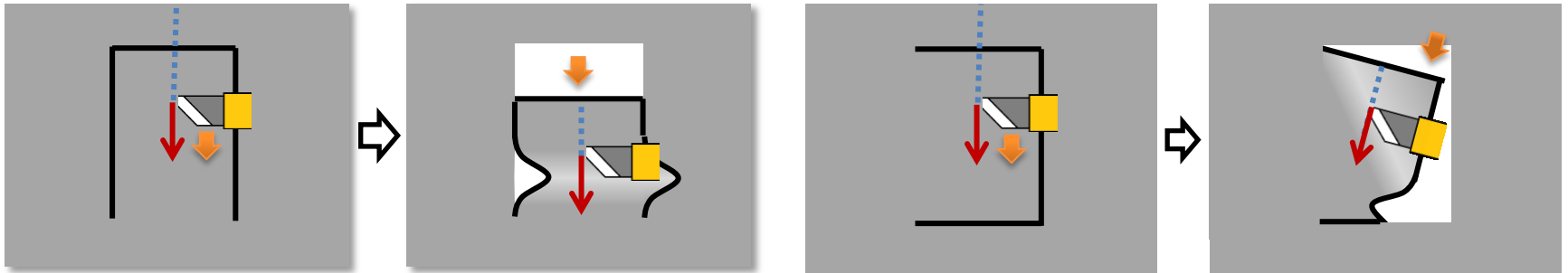
# 張線と既切線の交差



(d) 張線と既切線が交差する  
なのに、ずれない

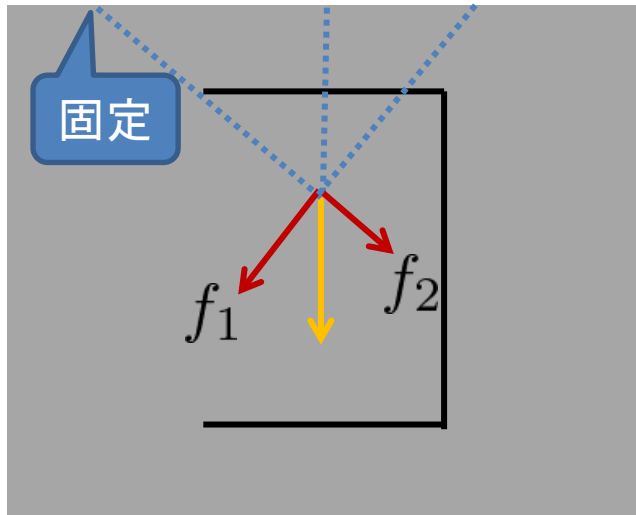
張線と既切線が交差しなければ  
ずれない

# ずれる状況の考察(3/4)



(a) 下にずれる

(b) 斜めにずれる



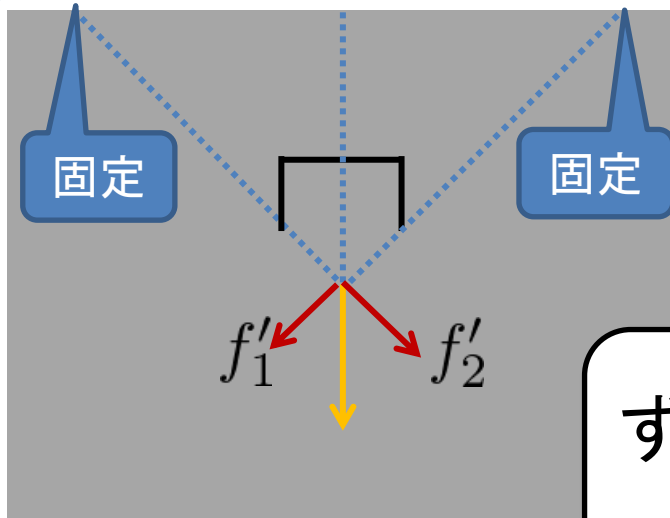
- $f_1, f_2$  に分解できる
  - $f_2$  だけ紙の張力と釣り合う
- ⇒  $f_2$  と既切線の接点を中心に  $f_1$  方向に回転

# ずれる状況の考察(4/4)



(b) ずれる

(d) ずれない

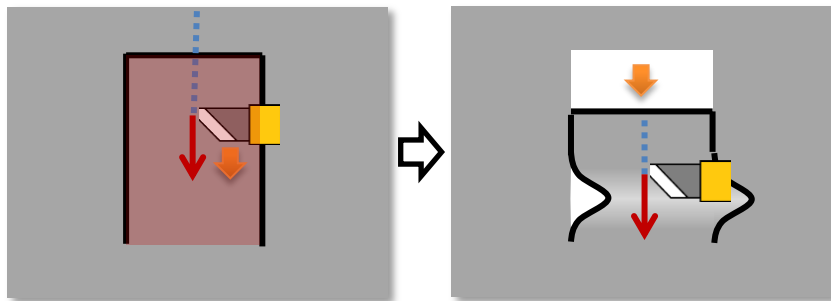


- $f'_1$  ,  $f'_2$  に分解できる
- $f'_1$  も  $f'_2$  も紙の張力と釣り合う

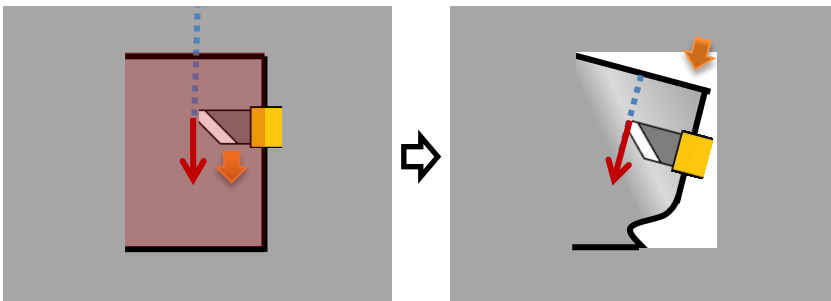
ずれるかずれないかの違いは  
既切線と始点の位置関係

# 始点と既切線の位置関係

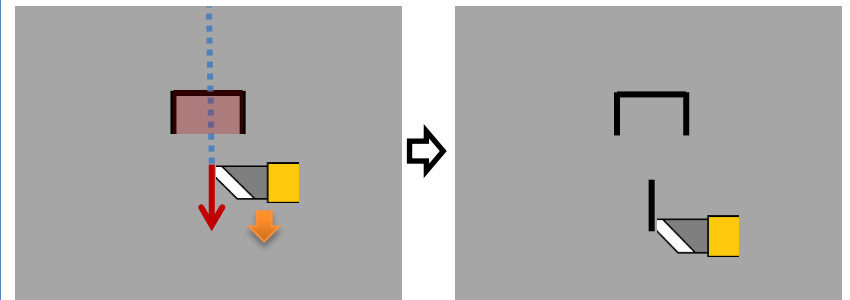
- 始点と既切線で囲まれた領域の関係に注目



(a) 凸包に始点が含まれる  
→ずれる



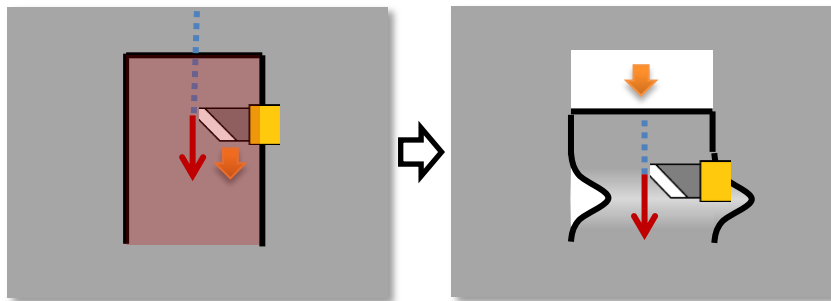
(b) 凸包に始点が含まれる  
→ずれる



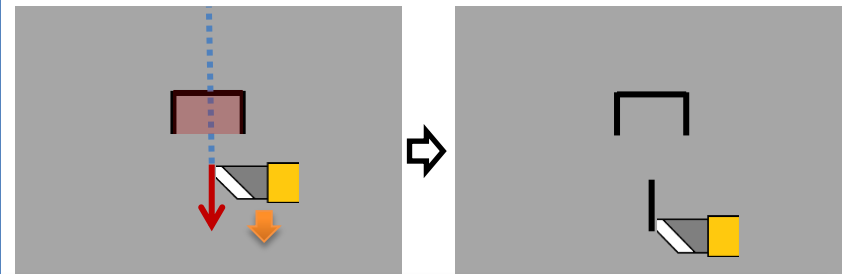
(d) 凸包に始点が含まれない  
→ずれない

# 始点と既切線の位置関係

- 始点と既切線で囲まれた領域の関係に注目



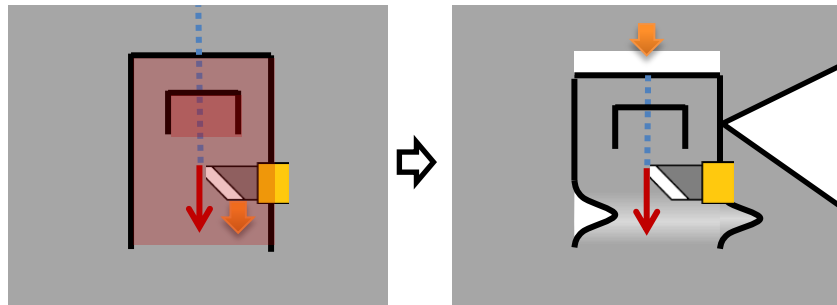
(a) 凸包に始点が含まれる  
→ずれる



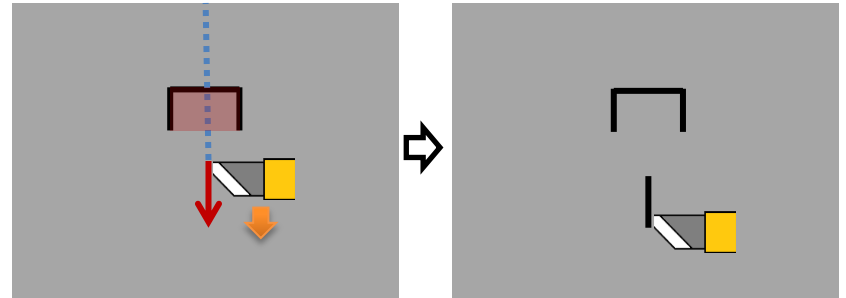
始点が(張線と交わる)既切線の凸包に含まれる  
ならずれる

(b) 凸包に始点が含まれる  
→ずれる

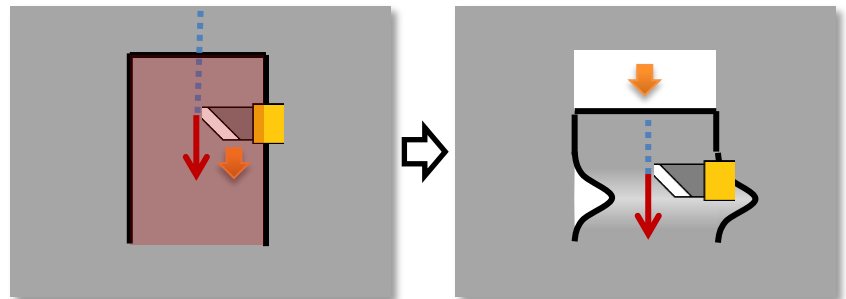
# 考慮すべき凸包



(e) ずれる

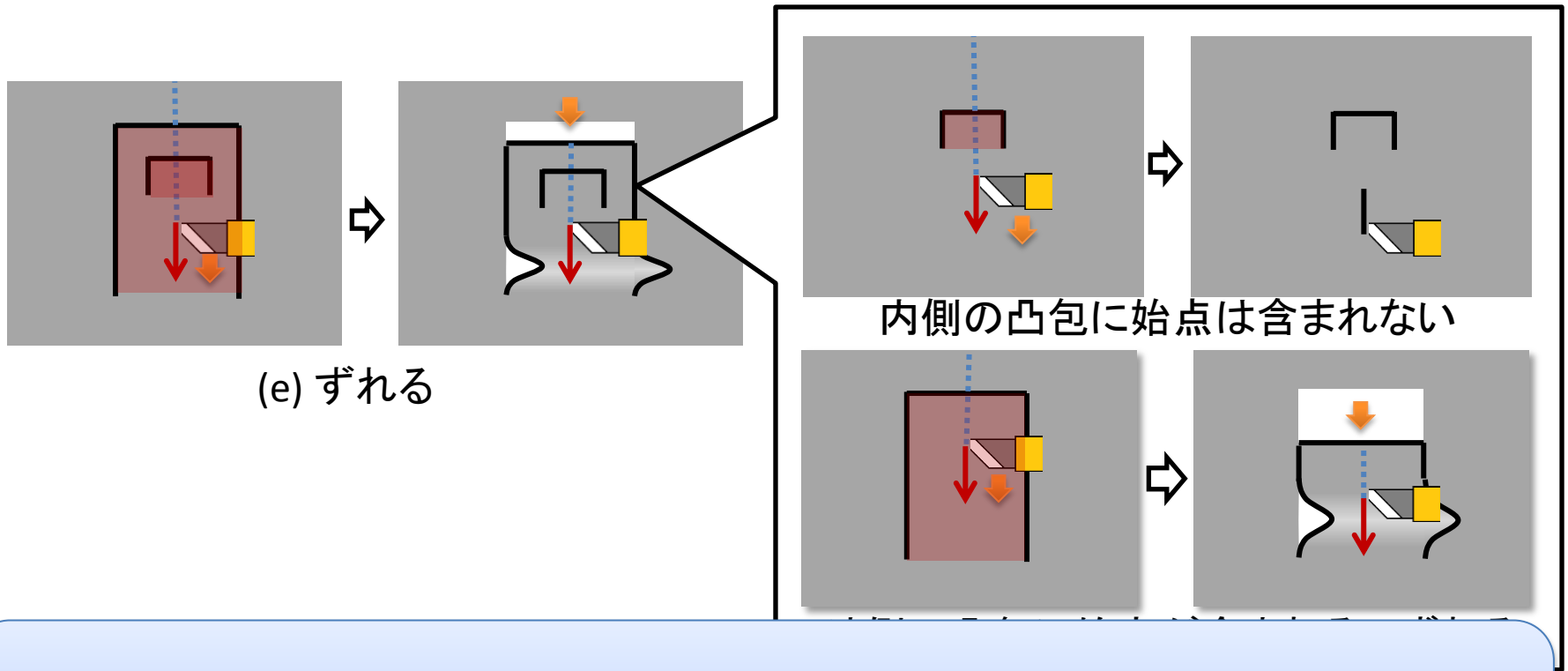


内側の凸包に始点は含まれない



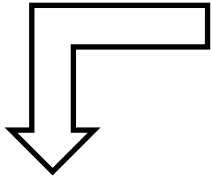
外側の凸包に始点が含まれる→ずれる

# 考慮すべき凸包



(張線と交わる)全ての凸包を調べる必要がある

# ずれ判定の擬似コード



- 張線と既切線が交差しなければずれない
- 始点が既切線の凸包に含まれるならずれる
- 全ての凸包を調べる必要がある

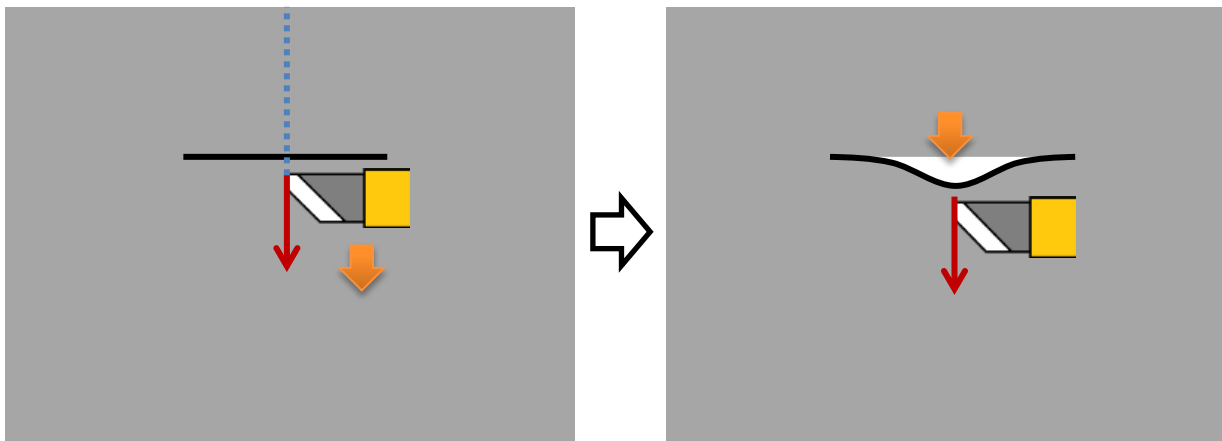
```
PROCEDURE ずれの判定(切り込みの始点, 切り込みの終点)
  ray ← 「切り込みの終点」を端点として「切り込みの始点」を通る半直線
  FOR 各既切線
    IF rayが既切線と交差する
      AND 既切線の凸包に切り込みの始点が含まれる
    THEN
      RETURN ずれる
    END IF
  END FOR
  RETURN ずれない
END PROCEDURE
```

張線

# 考慮に入れないずれる状況

- 紙がたわんでずれる状況

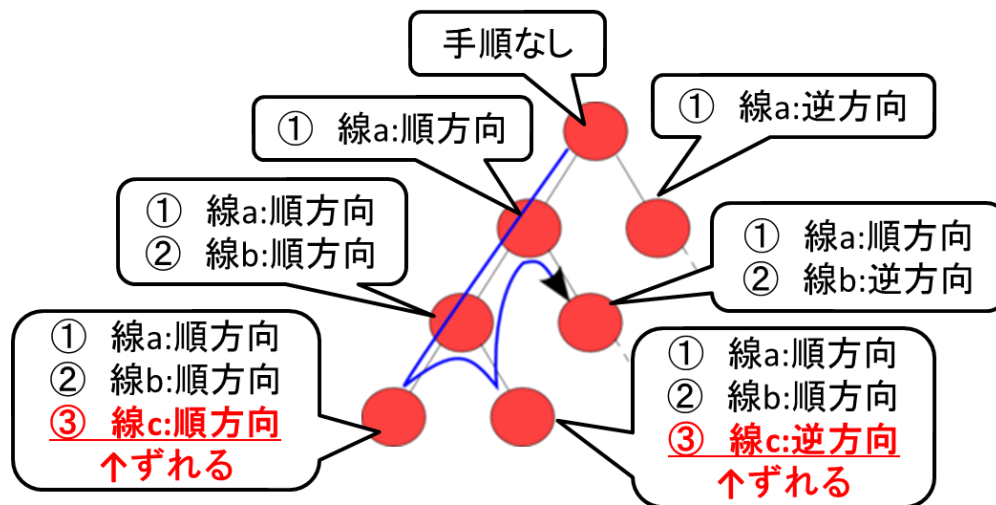
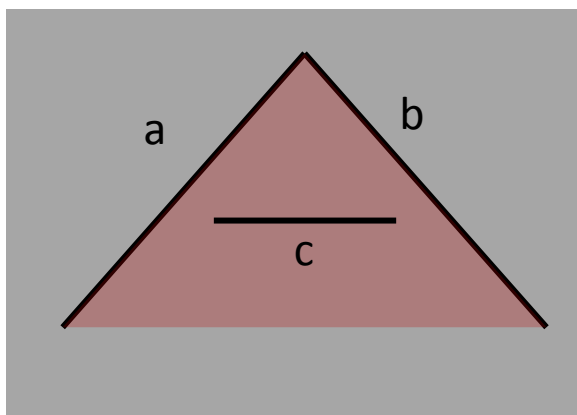
- 例: 切る線の張線が既切線に交わり始点が既切線に近い状況
- 少なくとも、紙(媒体)のたわみやすさが始点との近さに関する



今回は考えないずれる状況の例

# 切り順の探索アルゴリズム

- 一つ一つ線分を切っていき探索を進める
- ずれたら、異なる切り方を探索
- 全ての線分切ることができたら解として出力



探索の例

# デモ(1/2)



補足:  
言語はjava

# デモ(2/2)



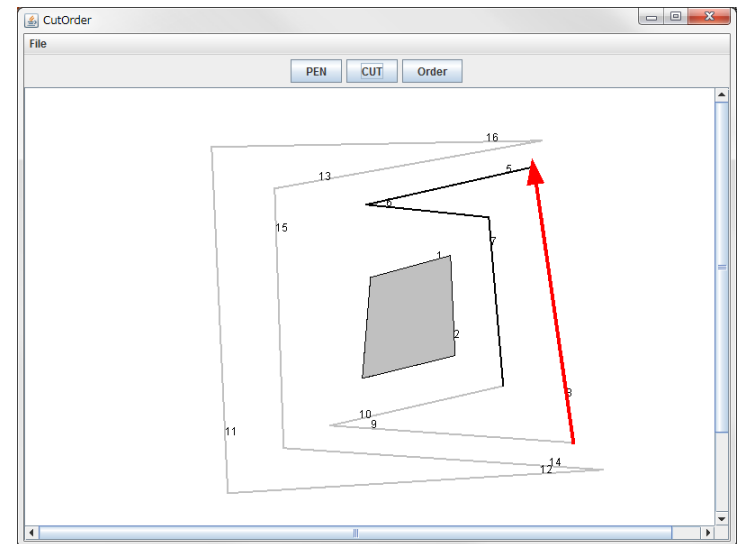
補足：  
探索法は  
深さ優先

# まとめ

- 切り絵制作手順を考慮した支援ツールを提案
  - ずれ発生の判定アルゴリズム考案
  - 切り順の探索

```
PROCEDURE ずれの判定(切り込みの始点, 切り込みの終点)
  ray ← 「切り込みの終点」を端点として「切り込みの始点」を通る半直線
  FOR 各既切線
    IF rayが既切線と交差する
      AND 既切線の凸包に切り込みの始点が含まれる
    THEN
      RETURN ずれる
    END IF
  END FOR
  RETURN ずれない
END PROCEDURE
```

ずれ判定の擬似コード



システムの実行画面

# 課題

- 線の数により探索時間がかかりすぎる
  - ヒューリスティック関数で目標に近いかを評価
- 現在の探索では切るときの効率の良さは無視
  - コスト関数で切るときの効率のよさを評価
  - コスト関数は切り絵経験者の観点を参考にする
- 提案された切り順の評価
  - 切り絵素人の方に対してのユーザテスト
  - 多少複雑な切り絵のサンプルを用意
  - 切り順を提案した場合と、しない場合どちらが切りやすかったかを判断

# 今後の展望

- カuttingプロッタへの応用



カuttingプロッタとは  
任意の形状を切り抜く機械  
台には粘着性のあるビニール

- 台上の粘着が弱まってしまった、または、粘着が効かない素材の場合、ずれが起こりうる
- 効率の良い切り方は作業時間を削減する可能性

**ご清聴ありがとうございました**

**▼ 以下は補足スライド**

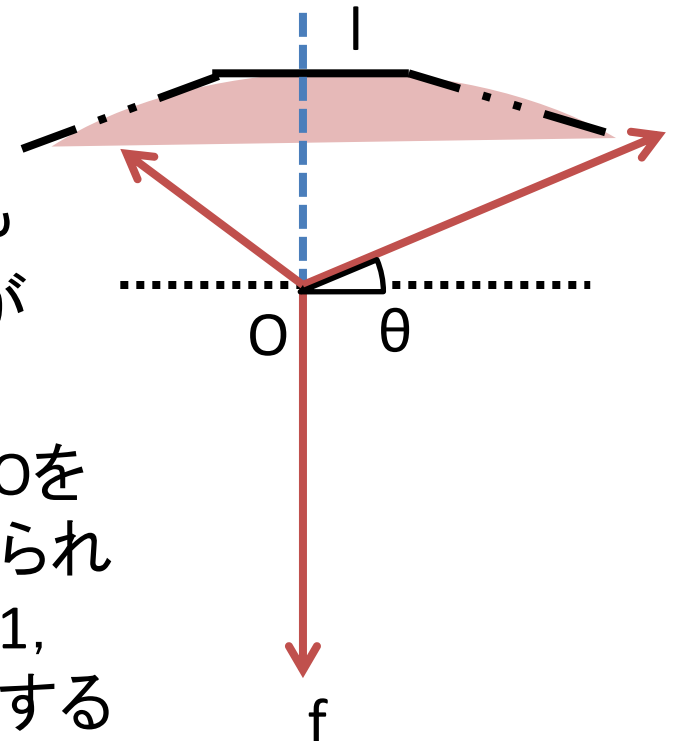
# ずれに凸包を用いる理由(証明)

・ある力 $f$ と釣り合うのに必要な力を考える.

$f$ と逆方向に力があればいいのだが、  
それはないと仮定する.

・すると、力の合成の関係上、少なくとも  
 $0 < \theta < \pi/2$ に一つ、 $\pi/2 < \theta < \pi$ に一つの力が  
なければ $f$ と釣り合う力は合成できない

・今 $O$ が線 $l$ の凸包に含まれるとすると、 $O$ を  
通りと交わらずに接する直線は二つ得られ  
 $l_1, l_2$ とする。その二つの $O$ での角度を $\theta_1,$   
 $\theta_2$ とする。 $0 < \theta_1 < \pi/2$ 、 $\pi/2 < \theta_2 < \pi$ と仮定する  
と、 $O$ は $l$ の凸包に含まれることと矛盾する



# 切り絵の基本表現(1/2)

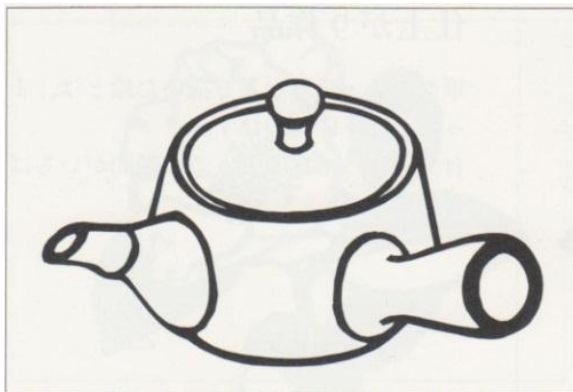
線によって分けられた閉領域を白か黒に割り当てる

白・・・切り抜く領域

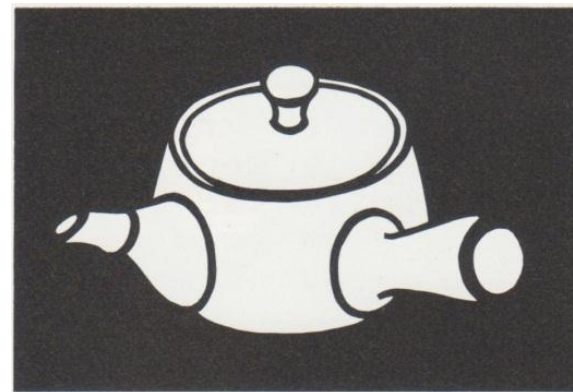
黒・・・切り抜かず残す領域

## 基本手法

- 白領域と黒領域の境界線はそのまま切る
- 白領域と白領域の境界線は黒い線として残す



線と閉領域

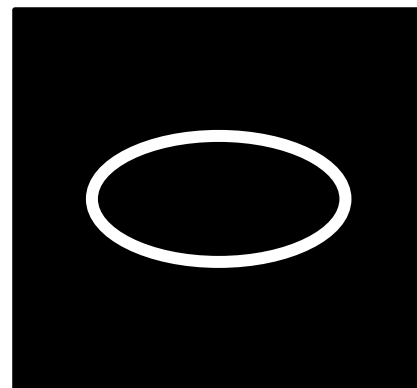


白い領域の隣接

# 切り絵の基本表現(2/2)

- 黒領域と黒領域の境界線は白い線として切り抜く

※境界線をすべてそのまま切り抜くと  
中心の領域は残せない



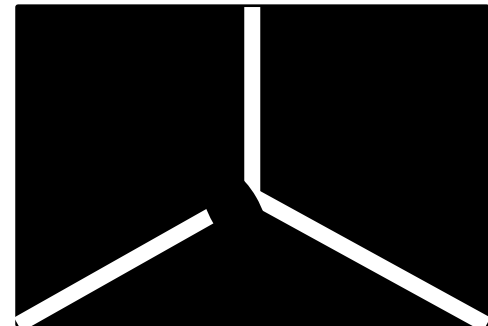
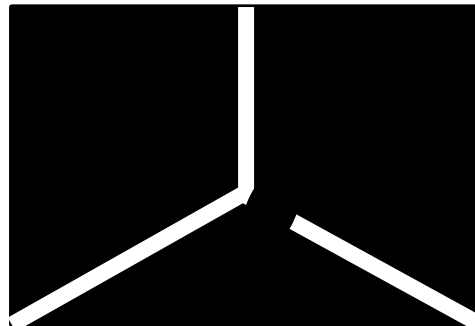
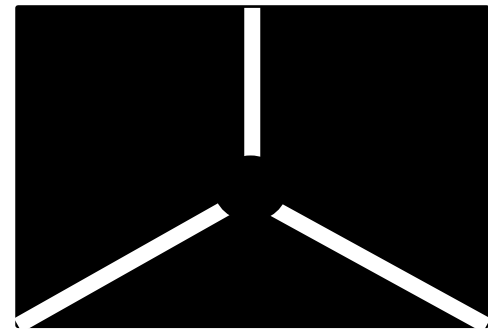
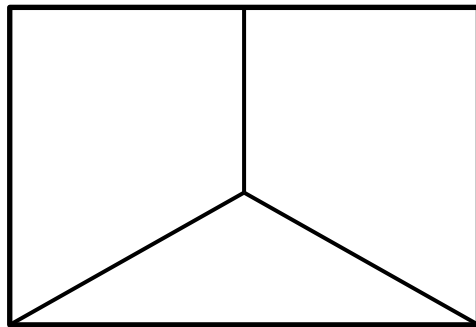
不自然でない様に繋げる必要がある  
- 基本は線の交点で繋げる



黒い領域の隣接

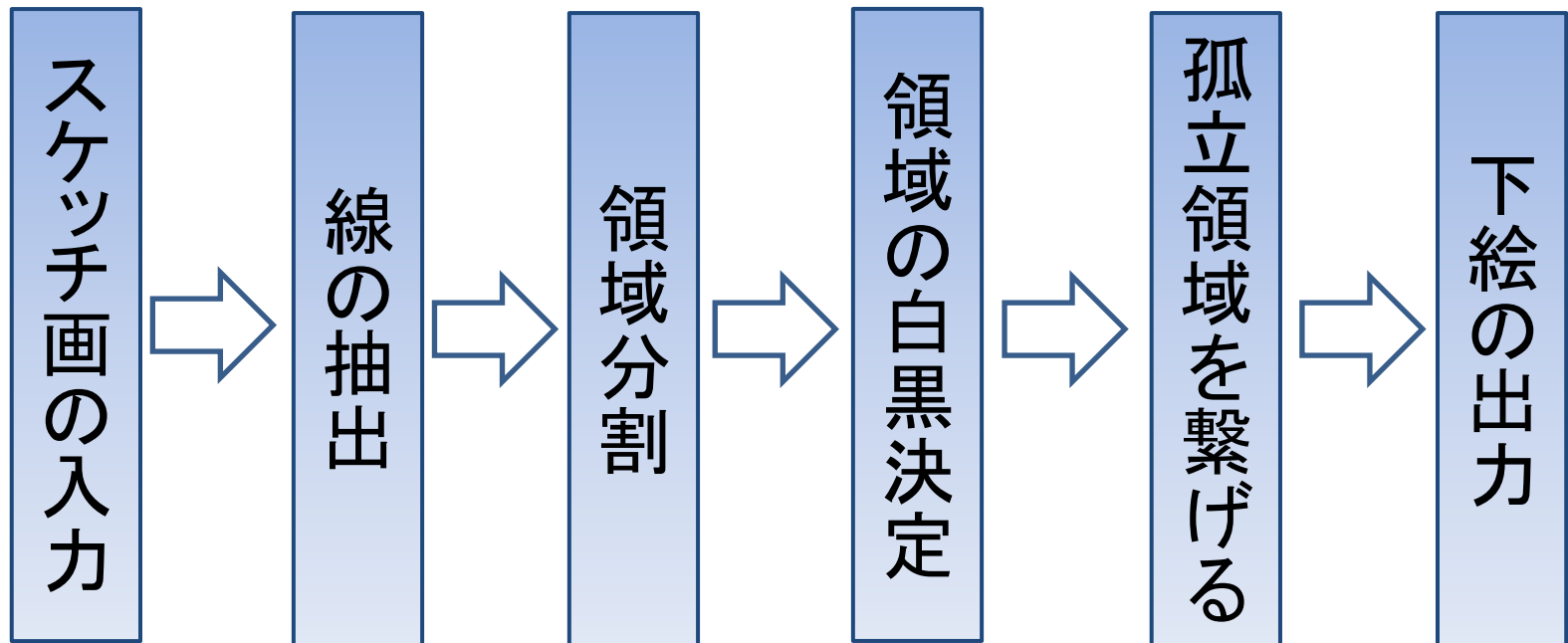
# 黒い領域のつなげ方

- 基本的に交点で繋げる
  - どう繋げるかが問題



# 下絵提案の流れ

スケッチから切り絵の表現を考慮して下絵を提案するため  
以下のような流れになる



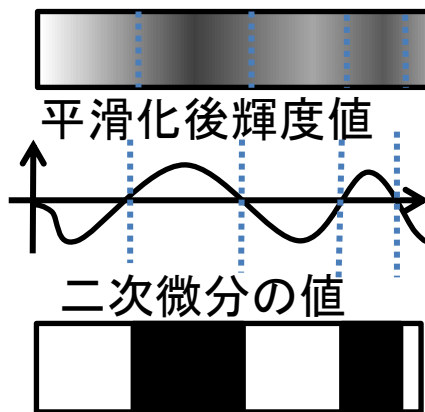
# 線の抽出

スケッチ画からLaplacian of Gaussianフィルタでストローク抽出

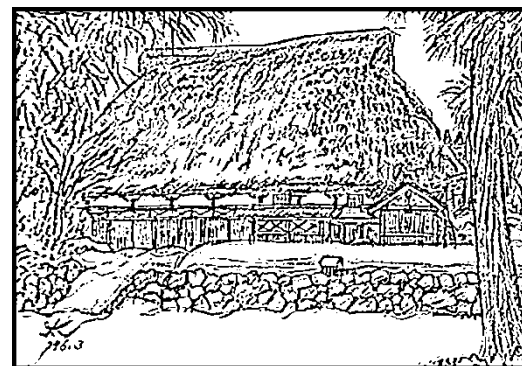
- フィルタ後の値が0以上なら黒、0未満なら白
- 黒い領域がストロークを表す



入力



二値化



ストローク抽出

- スケッチ画では輪郭線以外にも陰影や色の濃さもストロークで表現される
  - 輪郭線のみを抽出できてはいない

# 輪郭線の抽出

- 入力は完成された下絵
  - 2値化されたラスタ形式の画像
- 8近傍の輪郭抽出アルゴリズム
  - 連結情報を残しておく
- 線の長さが短く、数が多く、滑らかな折れ線ではない
- 1. Laplacian smoothingで平滑化を行う
- 2. 隣接する二つの線分が平行に近いならば
- 間の頂点を削除し、線を一つにし、線の数を減らす
- この処理1.2. を繰り返す。

# 課題

- 下絵の提案
  - ストロークから陰影、色の濃さではなく輪郭線のみ抽出
  - 線の交点で孤立領域を繋ぐだけではユーザが望む表現ではない
- 探索時間と平滑化・線の削除
  - 線が多ければ探索に時間がかかる
  - 平滑化・線の削除を繰り返すと元の下絵から誤差が多くなる
  - 探索時間を考え、目的の形の維持に必要な線の数の設定