IEEE Computer Society Conference on Computer Vision and Pattern Recognition, July 2007.

To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.

Accurate, Dense, and Robust Multi-View Stereopsis (PMVS)

Yasutaka Furukawa and Jean Ponce, *Fellow, IEEE*

2009.12.1

Abstract

This article proposes a novel algorithm for multiview stereopsis that outputs a dense set of small rectangular patches covering the surfaces visible in the images. Stereopsis is implemented as a match, expand, and filter procedure, starting from a sparse set of matched keypoints, and repeatedly expanding these before using visibility constraints to filter away false matches. Simple but effective methods are also proposed to turn the resulting patch model into a mesh which can be further refined by an algorithm that enforces both photometric consistency and regularization constraints. A quantitative evaluation on the Middlebury benchmark shows that the proposed method outperforms all others submitted so far for four out of the six datasets.

Key procedures of PMVS

(1) *Feature detection and matching*:

features found by Harris and Difference-of-Gaussians operators are matched across multiple pictures, yielding a sparse set of patches.

The procedure (2) and (3)iterative 3 times

(2) *Expansion:*

a technique is used to spread the initial matches to nearby pixels and obtain a dense set of patches.

(3) *Filtering:*

visibility constraints are used to eliminate incorrect matches lying either in front or behind the observed surface.

(4) *Polygonal surface reconstruction*

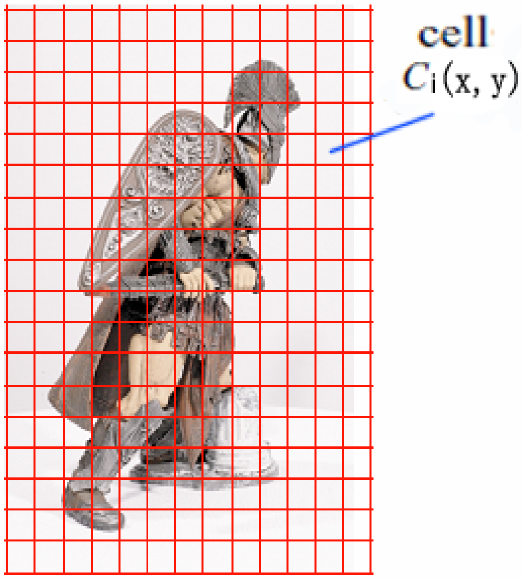
Some definitions

- **Cell**: we associate with each image a regular grid of $\beta_1 \times \beta_1$ pixels cells $C_i(x, y)$

- ***V*(*p*) and *V*(p)*:** $V(p)$ denote a set of images in which patch p is visible.

$$V^*(p) \quad = \quad \{I | I \in V(p), h(p, I, R(p)) \leq \alpha\}$$

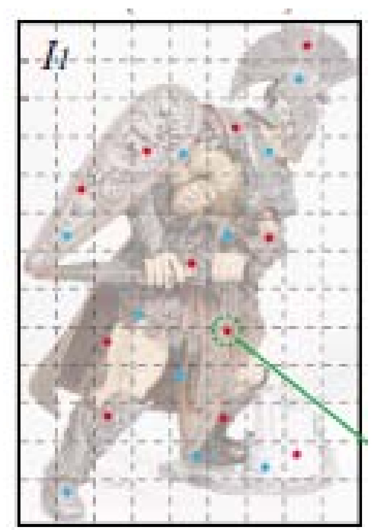
- ***Q$_i$(x,y)* and *Q*$_i$* (*x, y*)** : given a patch p and its visible images $V(p)$, we simply project p into each image in $V(p)$ to identify the corresponding cell. Then, each cell $C_i(x, y)$ remembers the set of patches $Q_i(x, y)$ that project into it. Similarly, we use $Q^*_i(x, y)$ to denote the patches that are obtained by the same procedure, but with $V*(p)$ instead of $V(p)$.



Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

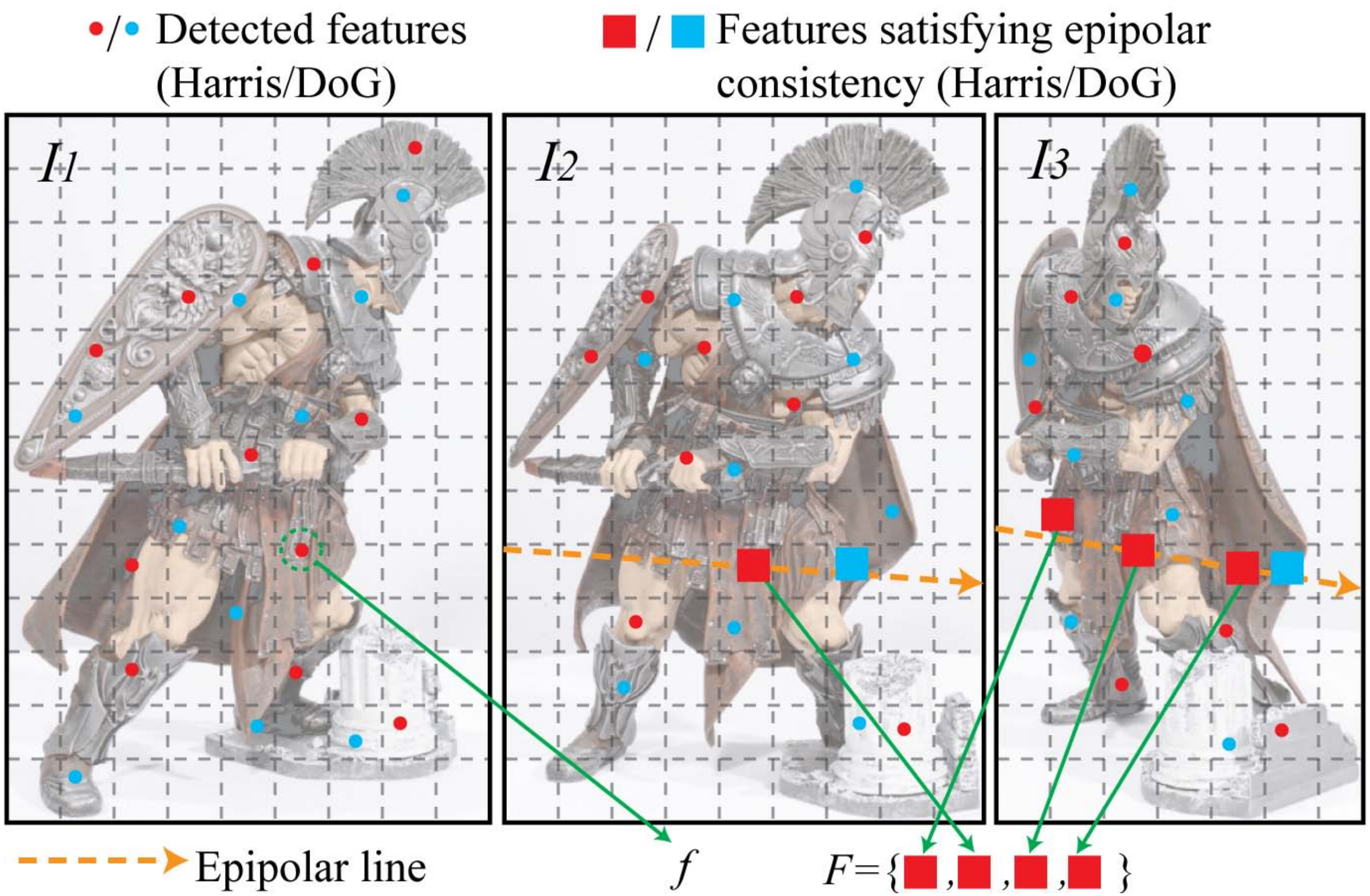
$P \leftarrow \phi$;
For each image I with optical center $O(I)$
    For each feature f detected in I
        $F \leftarrow$ {Features satisfying the epipolar consistency};
        Sort F in an increasing order of distance from $O(I)$);
        For each feature $f' \in F$
            // Test a patch candidate p;
            Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4, 5, 6)
            Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
            Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
            Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
            If $|V^*(p)| < \gamma$
                Go back to the innermost *For* loop (failure);
            Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
            Remove features from the cells where p was stored;
            Add p to P;
            Exit innermost *For* loop;



Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying the epipolar consistency};
    Sort F in an increasing order of distance from $O(I)$);
    For each feature $f' \in F$
      // Test a patch candidate p;
      Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4, 5, 6)
      Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
      Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      If $|V^*(p)| < \gamma$
        Go back to the innermost *For* loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;



•/• Detected features (Harris/DoG)
■ / ■ Features satisfying epipolar consistency (Harris/DoG)
I_1
I_2
I_3
Epipolar line
f
$F=\{$■,■,■,■$\}$

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying the epipolar consistency};
    Sort F in an increasing order of distance from $O(I)$);
    For each feature $f' \in F$
      // Test a patch candidate p;
      $\boxed{\text{Initialize } \mathbf{c}(p), \mathbf{n}(p) \text{ and } R(p); \text{ // Eqs. (4, 5, 6)}}$
      Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
      Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      If $|V^*(p)| < \gamma$
        Go back to the innermost *For* loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying the epipolar consistenc
    Sort F in an increasing order of distance from C
    For each feature $f' \in F$
      // Test a patch candidate p;
      Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4, 5, 6)
      Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
      Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      If $|V^*(p)| < \gamma$
        Go back to the innermost *For* loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;

$$\mathbf{c}(p) \leftarrow \{\text{Triangulation from } f \text{ and } f'\}, \quad (4)$$

$$\mathbf{n}(p) \leftarrow \overrightarrow{\mathbf{c}(p)O(I_i)} / |\overrightarrow{\mathbf{c}(p)O(I_i)}|, \quad (5)$$

$$R(p) \leftarrow I_i. \quad (6)$$

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying the epipolar consistency};
    Sort F in an increasing order of distance from $O(I)$);
    For each feature $f' \in F$
      // Test a patch candidate p;
      Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4, 5, 6)
      $\boxed{\text{Initialize } V(p) \text{ and } V^*(p)\text{; // Eqs. (2, 7)}}$
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
      Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      If $|V^*(p)| < \gamma$
        Go back to the innermost *For* loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying the epipolar cons
    Sort F in an increasing order of distance fi
    For each feature $f' \in F$
      // Test a patch candidate p;
      Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4, 5, 6)
      Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
      Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      If $|V^*(p)| < \gamma$
        Go back to the innermost *For* loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;

$$V^*(p) = \{I | I \in V(p), h(p, I, R(p)) \leq \alpha\}, \quad (2)$$

$$V(p) \leftarrow \left\{ I | \mathbf{n}(p) \cdot \overrightarrow{\mathbf{c}(p)O(I)} / |\overrightarrow{\mathbf{c}(p)O(I)}| > \cos(\iota) \right\}. \quad (7)$$

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
    For each feature f detected in I
        $F \leftarrow$ {Features satisfying the epipolar cons
        Sort F in an increasing order of distance fr
        For each feature $f' \in F$
            // Test a patch candidate p;
            Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4,
            Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
            Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
            Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
            If $|V^*(p)| < \gamma$
                Go back to the innermost *For* loop (failure);
            Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
            Remove features from the cells where p was stored;
            Add p to P;
            Exit innermost *For* loop;

optimization   The geometric parameters, $\mathbf{c}(p)$ and $\mathbf{n}(p)$, are optimized by simply minimizing the photometric

$$g^*(p) = \frac{1}{|V^*(p) \setminus R(p)|} \sum_{I \in V^*(p) \setminus R(p)} h(p, I, R(p)).$$

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

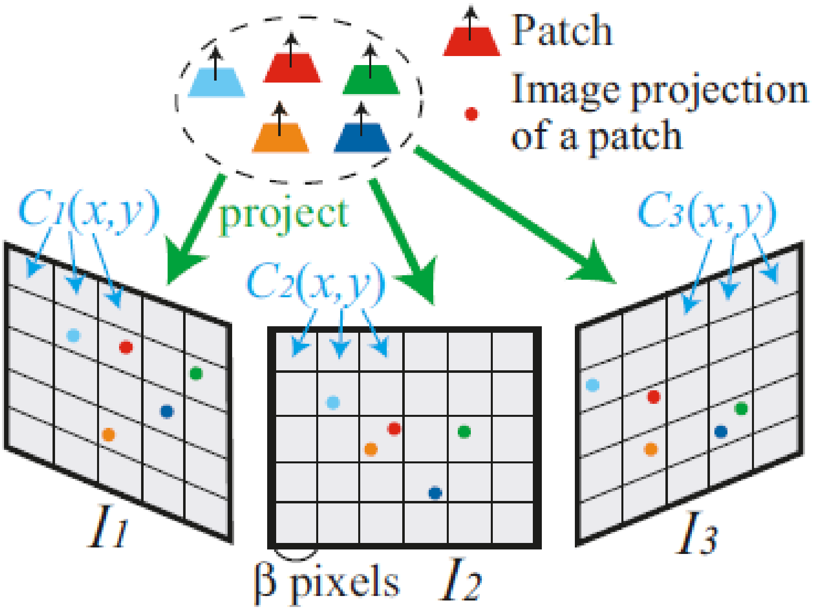
$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying the epipolar consistency};
    Sort F in an increasing order of distance from $O(I)$);
    For each feature $f' \in F$
      // Test a patch candidate p;
      Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and $R(p)$; // Eqs. (4, 5, 6)
      Initialize $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; // (Sect.II-C)
      Update $V(p)$ and $V^*(p)$; // Eqs. (2, 7)
      If $|V^*(p)| < \gamma$
        Go back to the innermost *For* loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q^*_j(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;

Feature detection and matching

Input: Features detected in each image.
Output: Initial sparse set of patches P.

$P \leftarrow \phi$;
For each image I with optical center $O(I)$
  For each feature f detected in I
    $F \leftarrow$ {Features satisfying th
    Sort F in an increasing ord
    For each feature $f' \in F$
      // Test a patch candidate
      Initialize $\mathbf{c}(p), \mathbf{n}(p)$ and
      Initialize $V(p)$ and $V^*(p$
      Refine $\mathbf{c}(p)$ and $\mathbf{n}(p)$; //
      Update $V(p)$ and $V^*(p)$;
      If $|V^*(p)| < \gamma$
        Go back to the innermost For loop (failure);
      Add p to the corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;
      Remove features from the cells where p was stored;
      Add p to P;
      Exit innermost *For* loop;

$Q_j(x,y)$ and $Q_j^*(x,y)$ Given a patch p and its visible images $V(p)$, we simply project p into each image in $V(p)$ to identify the corresponding cell. Then, each cell $C_i(x,y)$ remembers the set of patches $Q_i(x,y)$ that project into it. Similarly, we use $Q_i^*(x,y)$ to denote the patches that are obtained by the same procedure, but with $V^*(p)$ instead of $V(p)$.

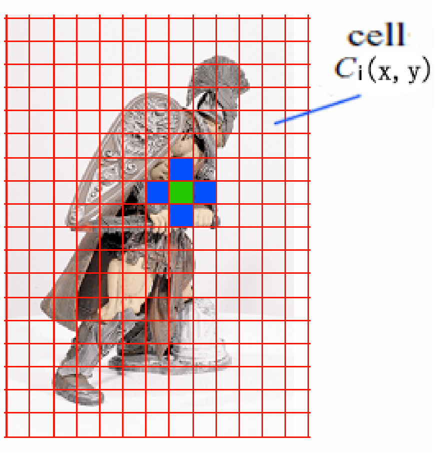


Patch
Image projection of a patch
project
C1(x,y)
C2(x,y)
C3(x,y)
I1
I2
I3
β pixels

Expansion

- Identifying Cells for expansion
- Expansion procedure

Identifying Cells for expansion



Given a patch p ,we initialize $C(p)$ by collecting the neighboring image cell in its each visible image:

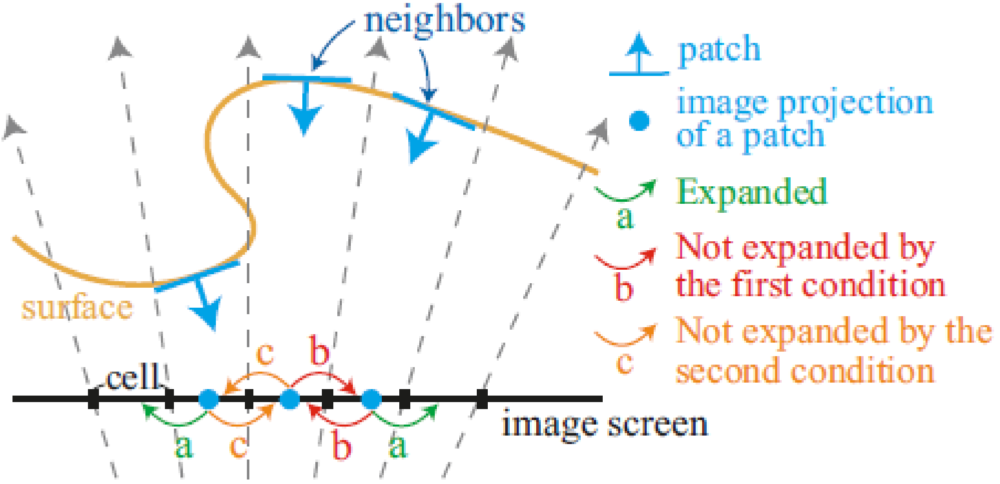
$$\mathbf{C}(p) = \{C_i(x', y') | p \in Q_i(x, y), |x - x'| + |y - y'| = 1\}$$

The expansion procedure is not performed for an image cell if

1. an image cell $C_i(x', y') \in C(p)$ contains a patch p', which is a neighbor of p , $C_i(x', y')$ is removed from the set of $C(p)$

$$|(\mathbf{c}(p) - \mathbf{c}(p')) \cdot \mathbf{n}(p)| + |(\mathbf{c}(p) - \mathbf{c}(p')) \cdot \mathbf{n}(p')| < 2\rho_1$$

2. $C_i(x', y')$ already contains a patch whose photometric discrepancy score associated with I_i is better than the threshold α.


neighbors
patch
image projection
of a patch
a
Expanded
b
Not expanded by
the first condition
c
Not expanded by the
second condition
surface
c
b
cell
image screen
a
c
b
a

Expansion procedure

Input: Patches P from the feature matching step.
Output: Expanded set of reconstructed patches.

While P is not empty
  Pick and remove a patch p from P;
  For each image cell $C_i(x,y)$ containing p
    Collect a set $\mathbf{C}$ of image cells for expansion;
    For each cell $C_i(x',y')$ in $\mathbf{C}$
      // Create a new patch candidate p'
      $\mathbf{n}(p') \leftarrow \mathbf{n}(p)$, $R(p') \leftarrow R(p)$, $V(p') \leftarrow V^*(p)$;
      Update $V^*(p)$; // Eq. (2)
      Refine $\mathbf{c}(p')$ and $\mathbf{n}(p')$; // (Sect.II-C)
      Add visible images (a depth-map test) to $V(p')$;
      Update $V^*(p')$; // Eq. (2)
      If $|V^*(p')| < \gamma$
        Go back to *For*-loop (failure);
      Add p' to P;
      Add p' to corresponding $Q_j(x,y)$ and $Q_j^*(x,y)$;

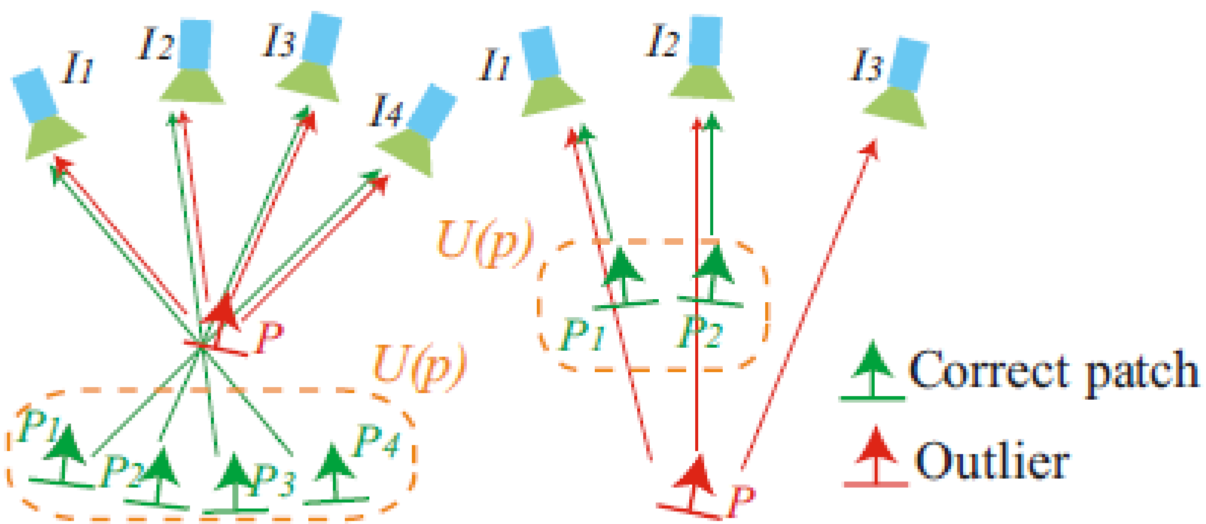
Filtering

A patch p will be removed in the following three situations:

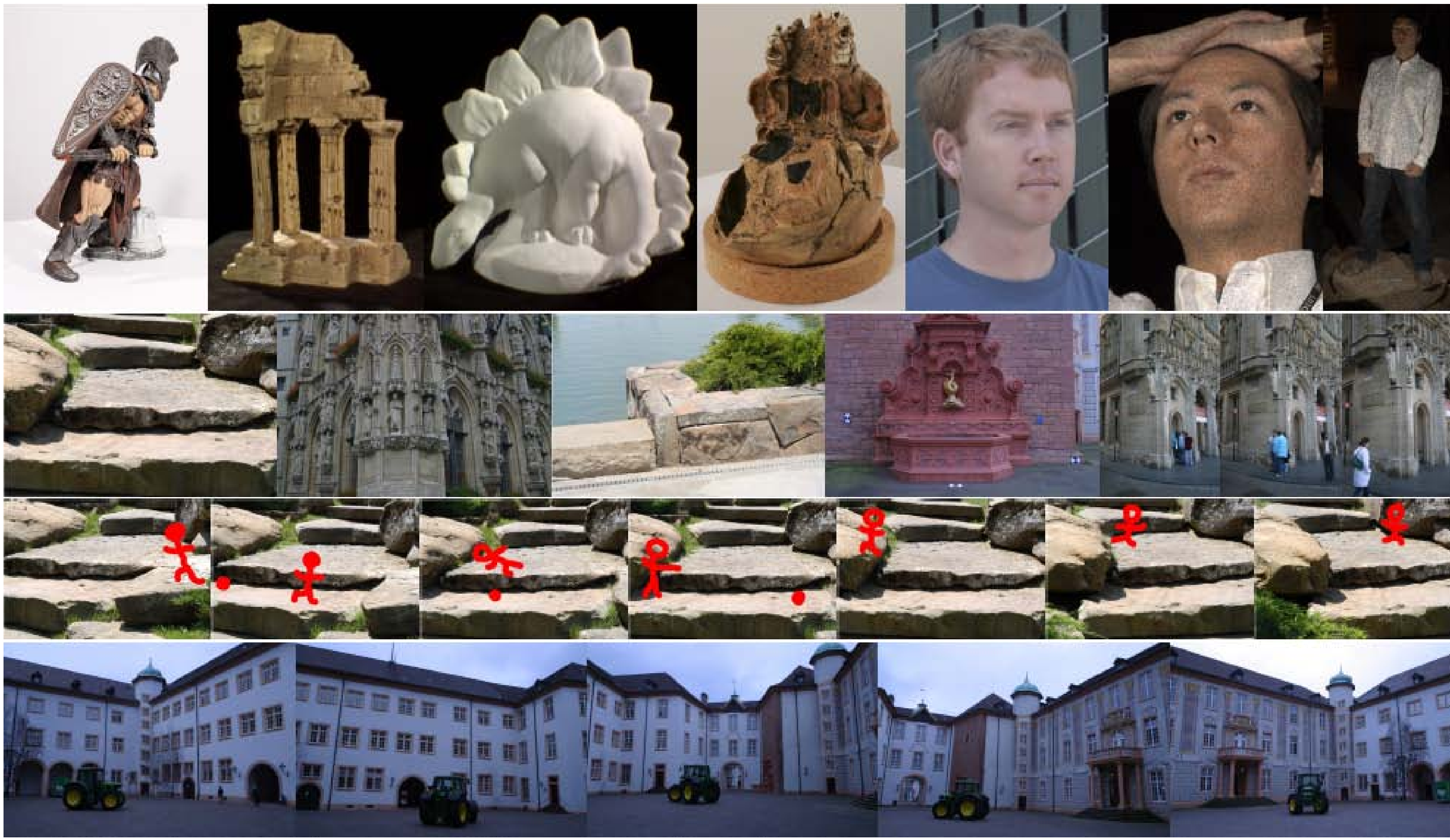
1. p and p' are not neighbors,but are stored in the same cell of one of the images where p is visible. Then p is filtered out as an outliers if the following inequality holds

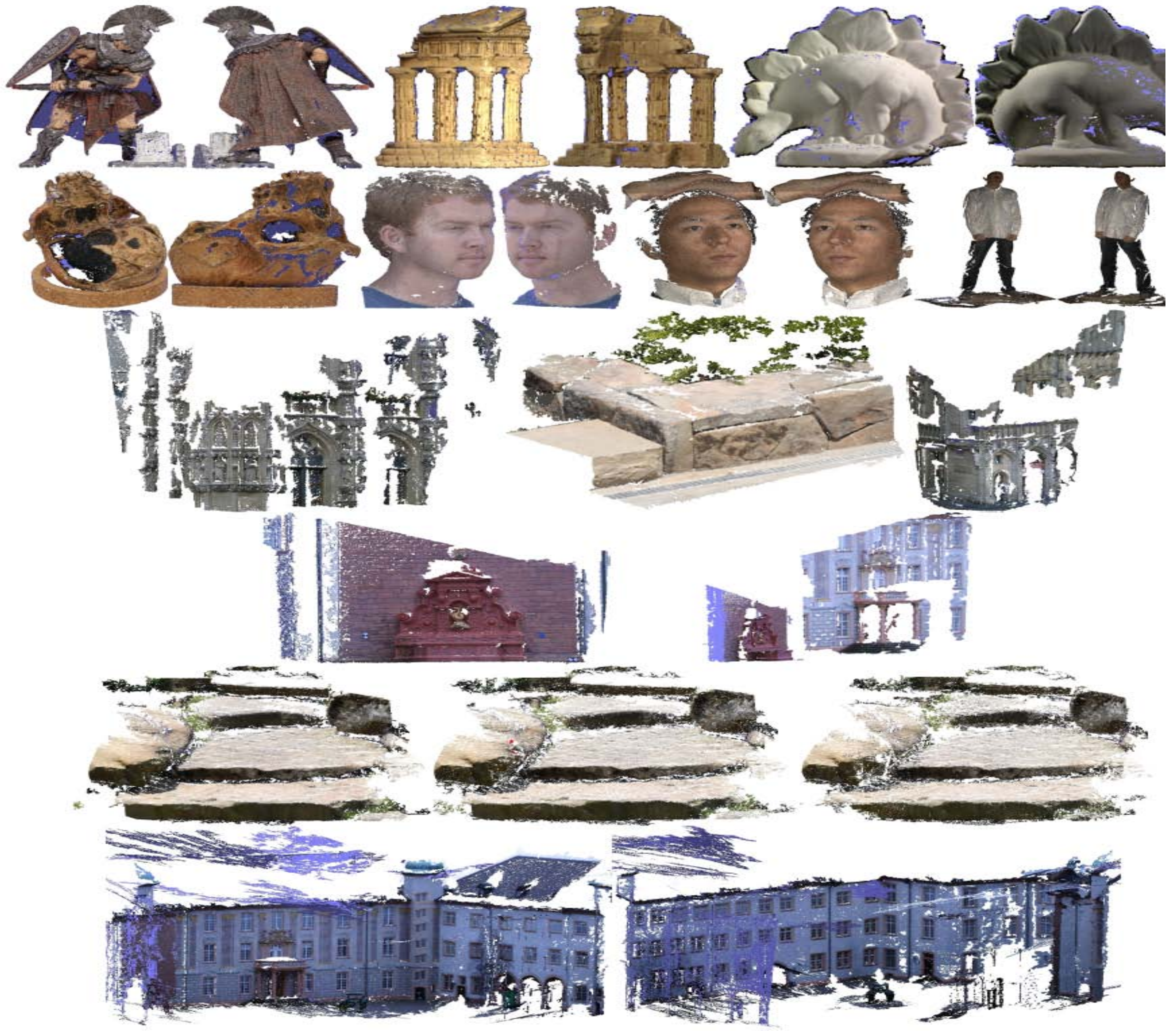
$$|V^*(p)|(1-g^*(p)) < \sum_{p_i \in U(p)} 1-g^*(p_i).$$

2. If the number of images in $V^*(p)$ according to depth-map test is less than γ.
3. If the proportion of patches that are neighbors of p in $V(p)$ is lower than 0.25.


I1
I2
I3
I4
P
U(p)
P1
P2
P3
P4
I1
I2
I3
U(p)
P1
P2
P
Correct patch
Outlier

Some results in the paper





test