Structure from Motion

CS 763

Ajit Rajwade

Problem definition

- **Structure from motion** refers to the inference of the object’s 3D structure or shape (i.e. the X,Y,Z coordinates of several points on the object’s surface) given a sequence of the object’s images when the object is in relative motion w.r.t. a camera.

Human perception of motion

- We humans have the ability to do this inference – see below:

https://www.youtube.com/watch?v=zdKX7Xo3Cb8&feature=player_detailpage#t=270

Contents of the lecture

- We are going to study an interesting algorithm called as **factorization**.
- It was developed by Tomasi and Kanade and was published in the early 90s.
- The algorithm is simple and elegant.

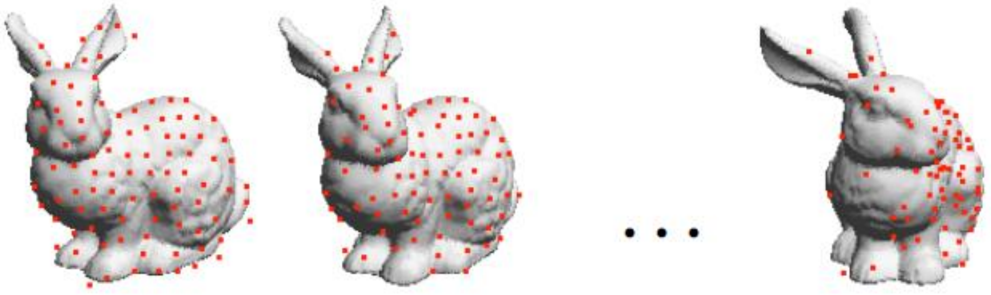
Tomasi and Kanade, “Shape and motion from image streams under orthography: a factorization method”, International Journal of Computer Vision, 1992.
http://link.springer.com/article/10.1007%2FBF00129684#page-1

Algorithm input and assumptions

- Given: A sequence of some $F \geq 3$ images of a non-planar object acquired under an **orthographic** camera moving relative to the object.
- The camera may be actually moving and the object could be still, or vice-versa, or both could be in motion.
- For simplicity but without loss of generality, we will assume the former.

Algorithm input and assumptions

- Let the object consist of $n \geq 3$ non-coplanar points – $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$ measured in some world coordinate system.
- We will assume that (1) these *n* 3D points are visible in each of the *F* frames, and (2) the corresponding *n* image points are tracked and marked out in each of the *F* frames.



Algorithm: input and assumptions

- We are assuming an orthographic camera.
- We are assuming that the whole video sequence is obtained a priori with points tracked.

Algorithm: input and assumptions

- Let $\mathbf{p}_{ij}$ = (x_{ij}, y_{ij}) = *j*-th image point (*j* = 1 to *n*) in the *i*-th frame (*i* = 1 to *F*).
- Assemble matrix **W** (size 2*F* x *n*) as follows:

$$\mathbf{W} = \begin{pmatrix} x_{11} & x_{12} & . & . & x_{1n} \\ x_{21} & x_{22} & . & . & x_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ x_{F1} & x_{F2} & . & . & x_{Fn} \\ y_{11} & y_{12} & . & . & y_{1n} \\ y_{21} & y_{22} & . & . & y_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ y_{F1} & y_{F2} & . & . & y_{Fn} \end{pmatrix}$$

Algorithm: input and assumptions

- Consider the following matrix (size 2*F* x *n*) as follows:

$$\tilde{\mathrm{W}} = \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & . & . & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & . & . & \tilde{x}_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \tilde{x}_{F1} & \tilde{x}_{F2} & . & . & \tilde{x}_{Fn} \\ \tilde{y}_{11} & \tilde{y}_{12} & . & . & \tilde{y}_{1n} \\ \tilde{y}_{21} & \tilde{y}_{22} & . & . & \tilde{y}_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \tilde{y}_{F1} & \tilde{y}_{F2} & . & . & \tilde{y}_{Fn} \end{pmatrix}$$

For each frame, compute the centroid of the 2D points. Deduct the centroid from the points in every frame to create the new matrix on the left. Why do we do this? We will see soon.

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_i, \bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}, \tilde{y}_{ij} = y_{ij} - \bar{y}_i, \bar{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{ij}$$

We will prove that this matrix actually has rank at the most 3 under ideal conditions (no noise in point coordinates). This is called the **Rank Theorem**.

Proof: Rank Theorem

- The 3D object is stationary and the camera is moving (performing rotation and translation).
- Each time the camera moves, its extrinsic parameters change, i.e. the **rotation transformation** between the camera axes and the world coordinate axes changes, and also the **translation vector** between the origin of the camera coordinate system and the origin of the world coordinate system changes.

Proof: Rank Theorem

- In the *i*-th frame, let the translation vector be given as $\mathbf{t}_i$. Let the axes of the camera as measured in the *world* coordinate system be given as $\mathbf{i}_i$, $\mathbf{j}_i$, $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$.

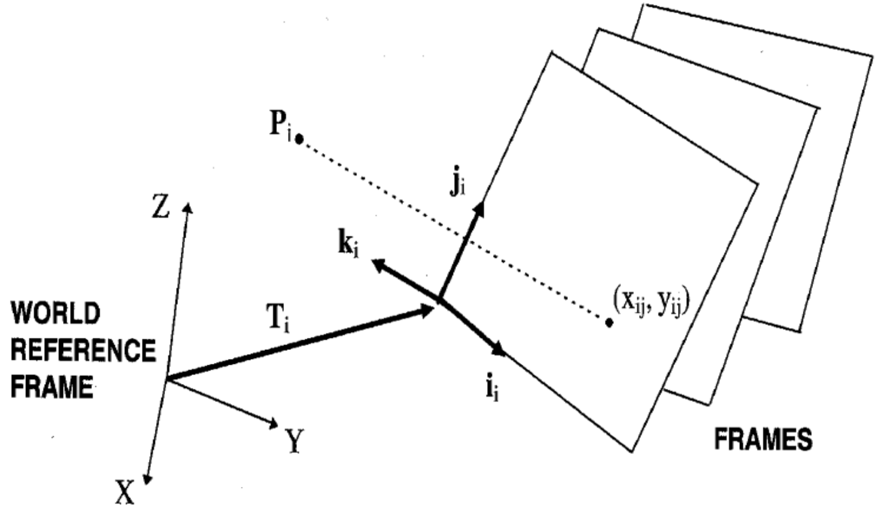


Figure 8.10 The geometry of the factorization method.

The image coordinates are thus given as follows:

$$x_{ij} = \mathbf{i}_i^t(\mathbf{P}_j - \mathbf{t}_i)$$

$$y_{ij} = \mathbf{j}_i^t(\mathbf{P}_j - \mathbf{t}_i)$$

Proof: Rank Theorem

- Without loss of generality, we will assume that the origin of the world coordinate system is at the centroid of the 3D object.
- In other words, we have

$$\frac{1}{n}\sum_{j=1}^{n}\mathbf{P}_j = \mathbf{0}$$

Proof: Rank Theorem

- Now consider the following equations:

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_i, \bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij},$$

$$\tilde{y}_{ij} = y_{ij} - \bar{y}_i, \bar{y}_i = \frac{1}{n}\sum_{j=1}^{n} y_{ij}$$

$$\frac{1}{n}\sum_{j=1}^{n} \mathbf{P}_j = \mathbf{0}$$

$$x_{ij} = \mathbf{i}_i^t(\mathbf{P}_j - \mathbf{t}_i)$$

$$y_{ij} = \mathbf{j}_i^t(\mathbf{P}_j - \mathbf{t}_i)$$

- Combining them, we have

$$\tilde{x}_{ij} = \mathbf{i}_i^t(\mathbf{P}_j - \mathbf{t}_i) - \frac{1}{n}\mathbf{i}_i^t\sum_{m=1}^{n}(\mathbf{P}_m - \mathbf{t}_i)$$

$$\tilde{y}_{ij} = \mathbf{j}_i^t(\mathbf{P}_j - \mathbf{t}_i) - \frac{1}{n}\mathbf{j}_i^t\sum_{m=1}^{n}(\mathbf{P}_m - \mathbf{t}_i)$$

$$\Longrightarrow$$

$$\tilde{x}_{ij} = \mathbf{i}_i^t\mathbf{P}_j$$

$$\tilde{y}_{ij} = \mathbf{j}_i^t\mathbf{P}_j$$

Proof: Rank Theorem

- Reconsider the following equations:

$$\tilde{x}_{ij} = \mathbf{i}_i^t \mathbf{P}_j$$

$$\tilde{y}_{ij} = \mathbf{j}_i^t \mathbf{P}_j$$

$$\tilde{\mathrm{W}} = \mathbf{R}\mathbf{S}$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{i}_1^{\mathrm{t}} \\ \mathbf{i}_2^{\mathrm{t}} \\ . \\ . \\ \mathbf{i}_{\mathrm{F}}^{\mathrm{t}} \\ \mathbf{j}_1^{\mathrm{t}} \\ \mathbf{j}_2^{\mathrm{t}} \\ \\ \mathbf{j}_{\mathrm{F}}^{\mathrm{t}} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} \mathbf{P}_1 & \mathbf{P}_2 & . & . & \mathbf{P}_n \end{pmatrix}$$

R has size 2F x 3 and has rank 3 as F ≥ 3.

S has size 3 x n
and will have rank 3 if the points in **S** are non-coplanar.

So **W̃** has rank 3.

What does the rank theorem tell you?

- Given the matrix $\tilde{\mathbf{W}}$, we compute its SVD as follows:

Reduced form of the SVD as $\tilde{\mathbf{W}}$ has rank only 3

$$\tilde{\mathbf{W}}_{2F\times n} = \mathbf{U}_{2F\times 2F}\mathbf{D}_{2F\times n}(\mathbf{V}_{n\times n})^T = \underbrace{\mathbf{U}_{2F\times 3}\mathbf{D}^{1/2}_{3\times 3}}_{\mathbf{R}}\underbrace{\mathbf{D}^{1/2}_{3\times 3}(\mathbf{V}_{n\times 3})^T}_{\mathbf{S}}$$

- For *i* = 1 to *F*, the *i*th and (*F+i)*th rows of **R** give you the vectors $\mathbf{i}_i$ and $\mathbf{j}_i$ respectively. Since $\mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i$, we have axes of the camera coordinate system in the *i*-th frame. Comparing the camera coordinate systems across consecutive frames tells you how much the camera rotated from one frame to another.
- The columns of **S** give the 3D point coordinates.

Problem!

- But the obtained **R** and **S** are not unique because for any invertible 3 x 3 matrix **Q**, we have:

$$\tilde{W} = RS = R(QQ^{-1})S = RQQ^{-1}S$$

- How do we resolve this?

Problem solution

- But the obtained **R** and **S** are not unique because for any invertible 3 x 3 matrix **Q**, we have: $\tilde{\mathbf{W}} = \mathbf{RS} = \mathbf{R}(\mathbf{QQ}^{-1})\mathbf{S} = \mathbf{RQQ}^{-1}\mathbf{S}$
- Observe that the rows of a rotation matrix (here **RQ**) must have unit magnitude. Any two rows must be perpendicular to each other. So we solve for **Q** by observing that:

$$\left.\begin{aligned} \mathbf{i}_i^t \mathbf{QQ}^T \mathbf{i}_i &= 1 \\ \mathbf{j}_i^t \mathbf{QQ}^T \mathbf{j}_i &= 1 \\ \mathbf{i}_i^t \mathbf{QQ}^T \mathbf{j}_i &= 0 \end{aligned}\right\}$$

These 3 equations are true for all *i* from 1 to *F* (i.e. for each frame). These equations are called the **metric properties** or metric constraints on **R**. Recall that $\mathbf{i}_i$ and $\mathbf{j}_i$ are obtained from the **R** matrix that you get from the SVD of $\tilde{\mathbf{W}}$.

Problem solution: not so soon!

- We can solve for **Q** which will satisfy the following equations using Newton's method (details later):

$$\left.\begin{aligned}
\mathbf{i}_i^t \mathbf{Q}\mathbf{Q}^T \mathbf{i}_i - 1 &= 0 \\
\mathbf{j}_i^t \mathbf{Q}\mathbf{Q}^T \mathbf{j}_i - 1 &= 0 \\
\mathbf{i}_i^t \mathbf{Q}\mathbf{Q}^T \mathbf{j}_i &= 0
\end{aligned}\right\}$$

These 3 equations are true for all *i* from 1 to *F* (i.e. for each frame). Recall that $\mathbf{i}_i$ and $\mathbf{j}_i$ are obtained from the **R** matrix that you get from the SVD of $\tilde{\mathbf{W}}$.

- The final **R** and **S** matrices will be as follows:

$$\begin{aligned}
\mathbf{R} &\leftarrow \mathbf{R}\mathbf{Q} \\
\mathbf{S} &\leftarrow \mathbf{Q}^{-1}\mathbf{S}
\end{aligned}$$

- But these solutions are also unique only up to some unknown orthonormal transformation $\mathbf{R}_0$, i.e. $\tilde{\mathbf{W}} = \mathbf{R}\mathbf{S} = \mathbf{R}\mathbf{R}_0\mathbf{R}_0^T\mathbf{S}$

Problem solution: not bad after all!

- Note that this $\mathbf{R}_0$ cannot be uniquely obtained by exploiting the metric properties unlike the case of **Q** (why?).
- All this means is that the if you assumed all the camera positions were rotated by some **fixed** $\mathbf{R}_0$ in every frame, the object coordinates would rotate by a fixed $(\mathbf{R}_0)^{-1}$ in every frame.
- This can be resolved by assuming that in the first frame, the world coordinate system is aligned with the camera coordinate system.

What about camera translation from frame to frame?

- This is orthographic projection: so we can never determine the Z component of the translation vector in any frame.
- The X and Y components of the translation vector (in frame *t*) are obtained by the difference between the image centroids in frame *t* and those in frame *t*-1.

$$x_{ij} = \mathbf{i}_i^t(\mathbf{P}_j - \mathbf{t}_i)$$
$$y_{ij} = \mathbf{j}_i^t(\mathbf{P}_j - \mathbf{t}_i)$$

$$\overline{x}_i = -\mathbf{i}_i^t\mathbf{t}_i$$
$$\overline{y}_i = -\mathbf{j}_i^t\mathbf{t}_i$$

What about camera rotation from frame to frame?

- Compare the **i**,**j**,**k** axes of the camera in frame *t* and frame *t*-1.

Measurement noise

- The rank theorem says that $\tilde{\mathbf{W}}$ has rank 3.

- But that is true only when there is no noise in measuring the coordinates of the tracked points in every frame.

- What if there is noise? One can attempt to “filter out” the noise in $\tilde{\mathbf{W}}$ by considering its rank 3 approximation.

Measurement noise

- Consider the SVD:

$$\tilde{\mathbf{W}}_{2F\times n} = \mathbf{U}_{2F\times 2F}\mathbf{D}_{2F\times n}(\mathbf{V}_{n\times n})^T \approx \mathbf{U}_{2F\times 3}\mathbf{D}_{3\times 3}(\mathbf{V}_{n\times 3})^T$$

- Due to noise, the rank exceeds 3, but we can create a rank-3 approximation by considering only the 3 largest singular values in **D** (and their corresponding columns in **U** and **V**).

- This is the best rank-3 approximation to **W̃** as per the well-known Eckart-Young Theorem on SVD.

How to estimate **Q**?

- Look at the following equations (totally 3*F* in number):

$$\begin{aligned}
\mathbf{i}_i^t \mathbf{Q}\mathbf{Q}^T \mathbf{i}_i - 1 &= 0 \\
\mathbf{j}_i^t \mathbf{Q}\mathbf{Q}^T \mathbf{j}_i - 1 &= 0 \\
\mathbf{i}_i^t \mathbf{Q}\mathbf{Q}^T \mathbf{j}_i &= 0
\end{aligned}$$

- This is a system of non-linear equations, the variables being the 9 entries of **Q** which we rearrange to yield vector **q**. We will label each equation as $f_k(\mathbf{q}) = 0$ ($k = 1$ to $3F$).
- No closed-form solution unlike linear case ☹

How to estimate $\mathbf{Q}$?

1. Start with an initial guess for $\mathbf{q}$, for example $\mathbf{q}_t$ = vectorized form of identity matrix.
2. If $\mathbf{q}_t$ is the true solution, then $f_k(\mathbf{q}_t) = 0$ for all *k* from 1 to 3*F*, and you stop (this won't happen in the first step when $t = 0$!).
3. Instead we want to find vector $\boldsymbol{\delta}$ such that $f_k(\mathbf{q}_t + \boldsymbol{\delta}) = 0$ for all *k*.
4. We seek to find $\boldsymbol{\delta}$ by approximating each f_k as a linear function in the neighborhood of $\mathbf{q}_t$.

How to estimate **Q**?

5. The linear approximation is given as:

$$f_k(\mathbf{q}_t + \boldsymbol{\delta}) = f_k(\mathbf{q}_t) + \boldsymbol{\delta}^t \boxed{\left(\frac{\partial f_k}{\partial \mathbf{q}}\right)_{\mathbf{q}=\mathbf{q}_t}}$$

This is a 9 x 1 vector of first derivatives. Remember that **δ** is a 9 x 1 vector.

6. But we want $f_k(\mathbf{q}_t + \boldsymbol{\delta}) = 0$ for all *k*. Hence for a given *k*, we have

$$\boldsymbol{\delta}^t \left(\frac{\partial f_k}{\partial \mathbf{q}}\right)_{\mathbf{q}=\mathbf{q}_t} = -f_k(\mathbf{q}_t)$$

How to estimate **Q**?

7. Collecting together 3*F* such equations, we have:

$$\boldsymbol{\delta}^t \boxed{\left(\frac{\partial f}{\partial \mathbf{q}}\right)_{\mathbf{q}=\mathbf{q}_t}} = -f(\mathbf{q}_t)$$

The yellow box contains a 9 x 3*F* matrix called the Jacobian. Again, **δ** is a 9 x 1 vector and f($\mathbf{q}_t$) is also a 9 x 1 vector.

8. One can solve for **δ** by pseudo-inverse.

9. But this solution will not exactly satisfy all the equations as we performed a linear approximation which was not fully accurate, and also because a least squares solution for **δ** is not guaranteed to yield $f_k(\mathbf{q}_t + \boldsymbol{\delta}) = 0$ for all *k*.

How to estimate **Q**?

10. Hence we update our solution from $\mathbf{q_t}$ to $\mathbf{q_{t+1}} = \mathbf{q_t} + \boldsymbol{\delta}$.

- We repeat the previous steps with *t* = 0, 1, 2,… and so on until we reach a time when $f_k(\mathbf{q}_t + \boldsymbol{\delta}) \approx 0$ for all *k*.
- This overall method is called Newton-Raphson method of root-finding.

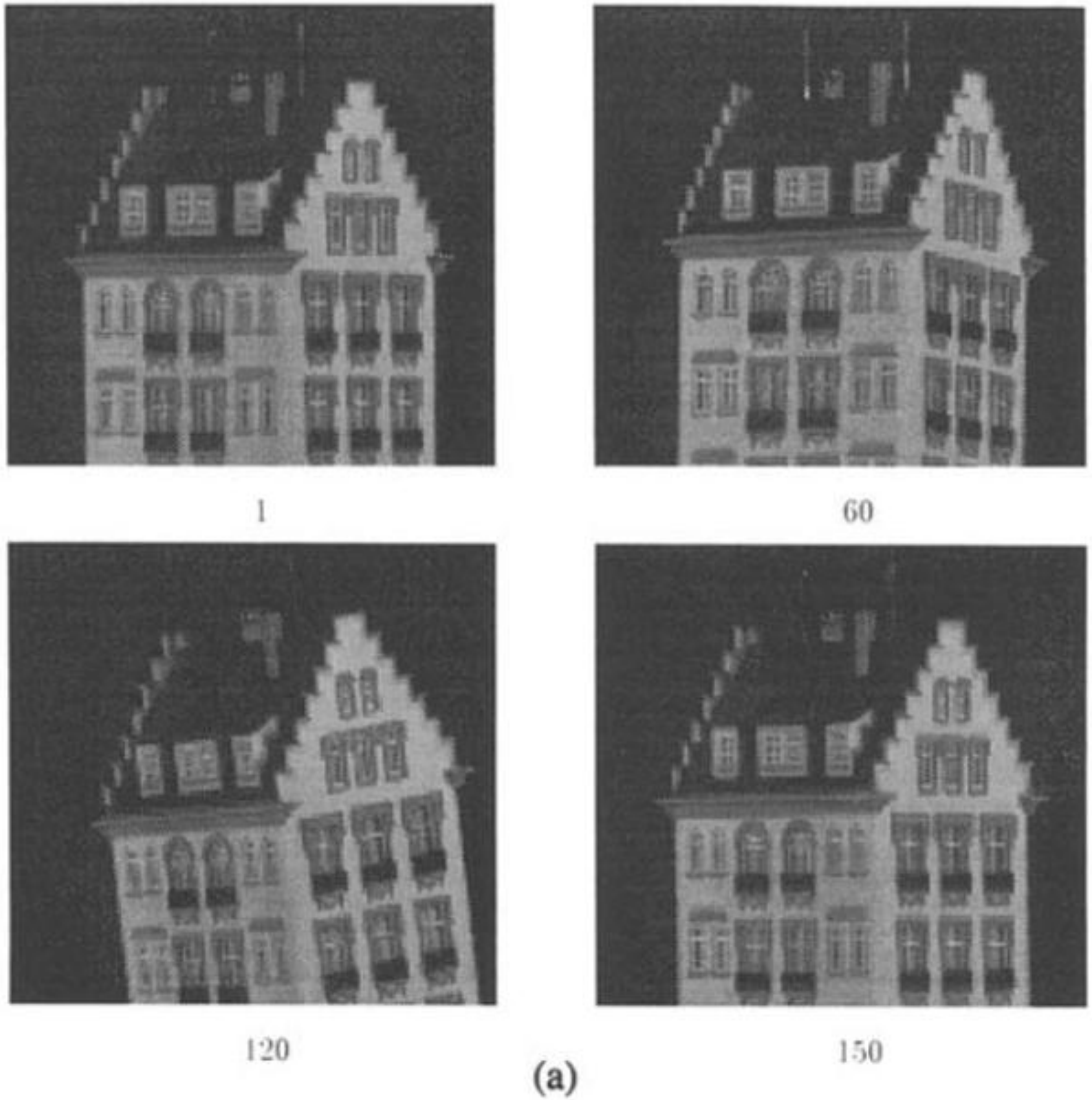
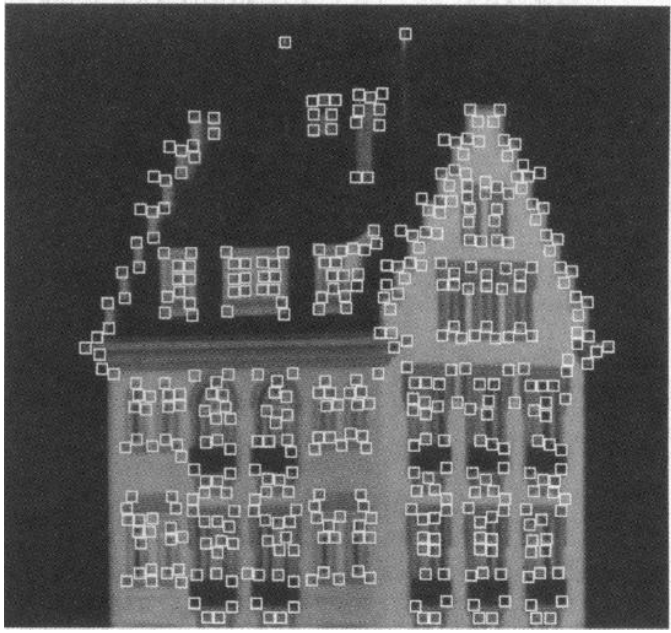
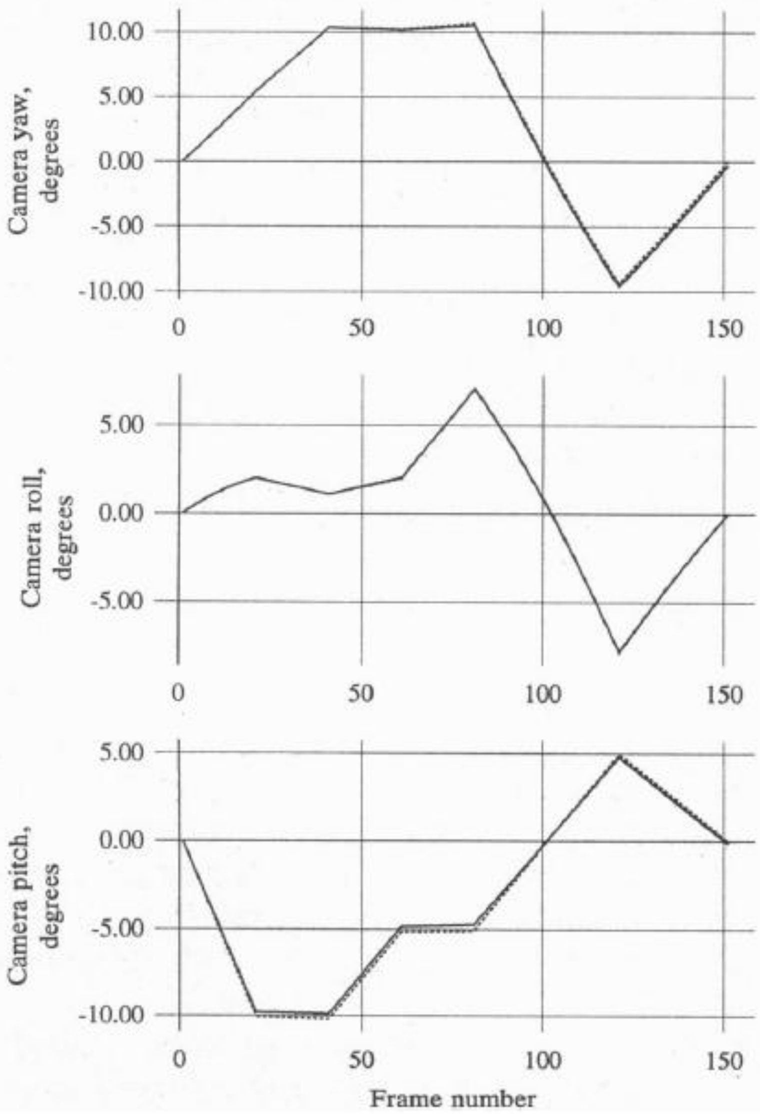
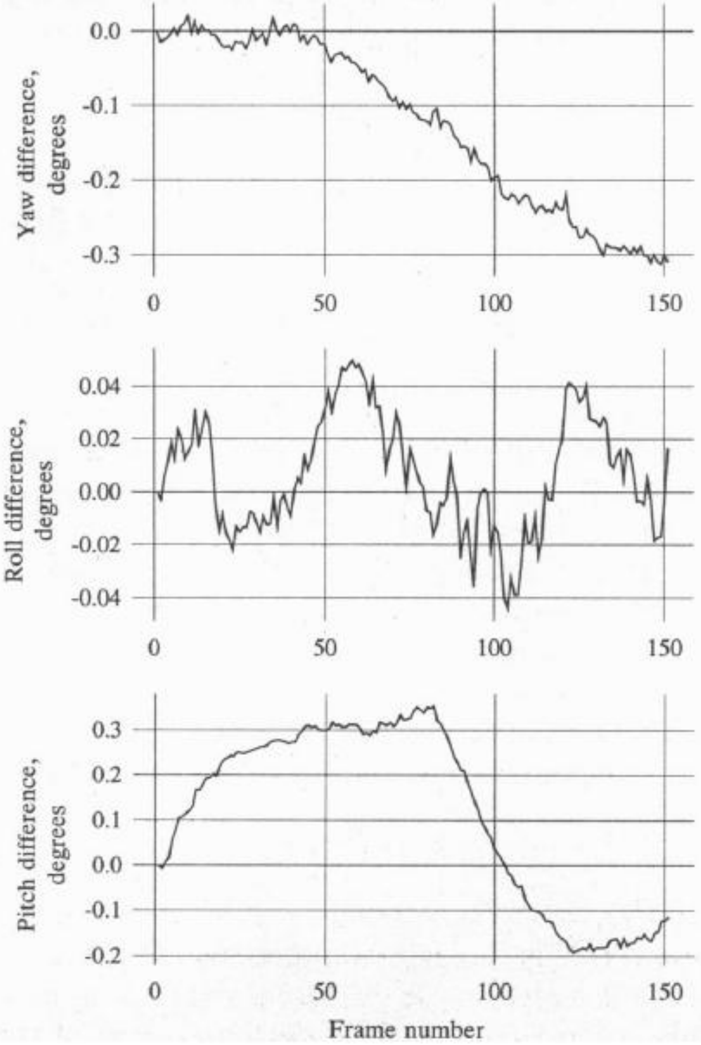


Fig. 2a. The "Hotel" stream: four of the 150 frames.




Camera yaw, degrees
10.00
5.00
0.00
-5.00
-10.00
0
50
100
150
Camera roll, degrees
5.00
0.00
-5.00
0
50
100
150
Camera pitch, degrees
5.00
0.00
-5.00
-10.00
0
50
100
150
Frame number


Yaw difference, degrees
0.0
-0.1
-0.2
-0.3
0
50
100
150
Roll difference, degrees
0.04
0.02
0.00
-0.02
-0.04
0
50
100
150
Pitch difference, degrees
0.3
0.2
0.1
0.0
-0.1
-0.2
0
50
100
150
Frame number

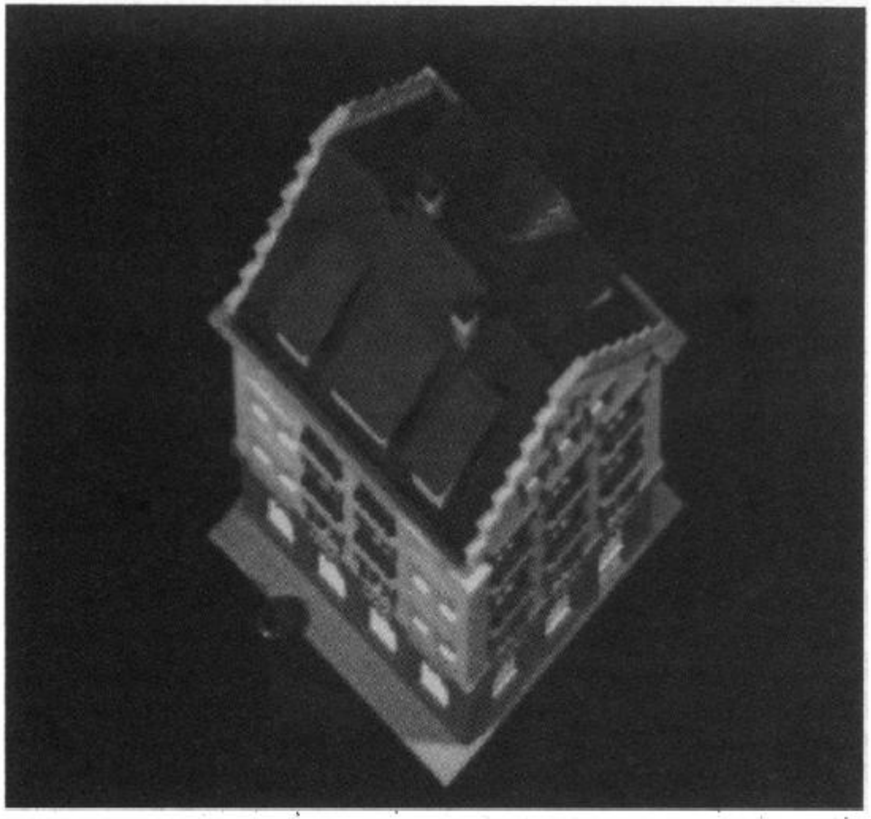
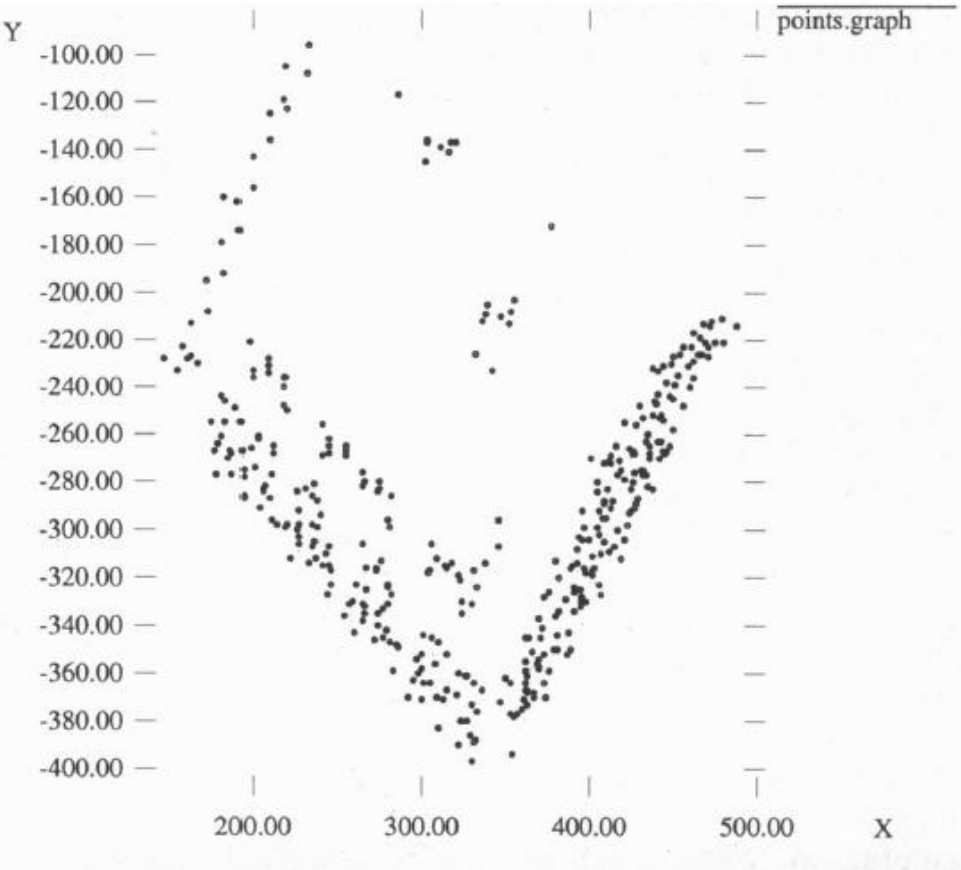


FIG. 4. (*Upper*) View of the computed shape from approximately above the building. (*Lower*) Real picture from above the building.


points.graph
Y
X