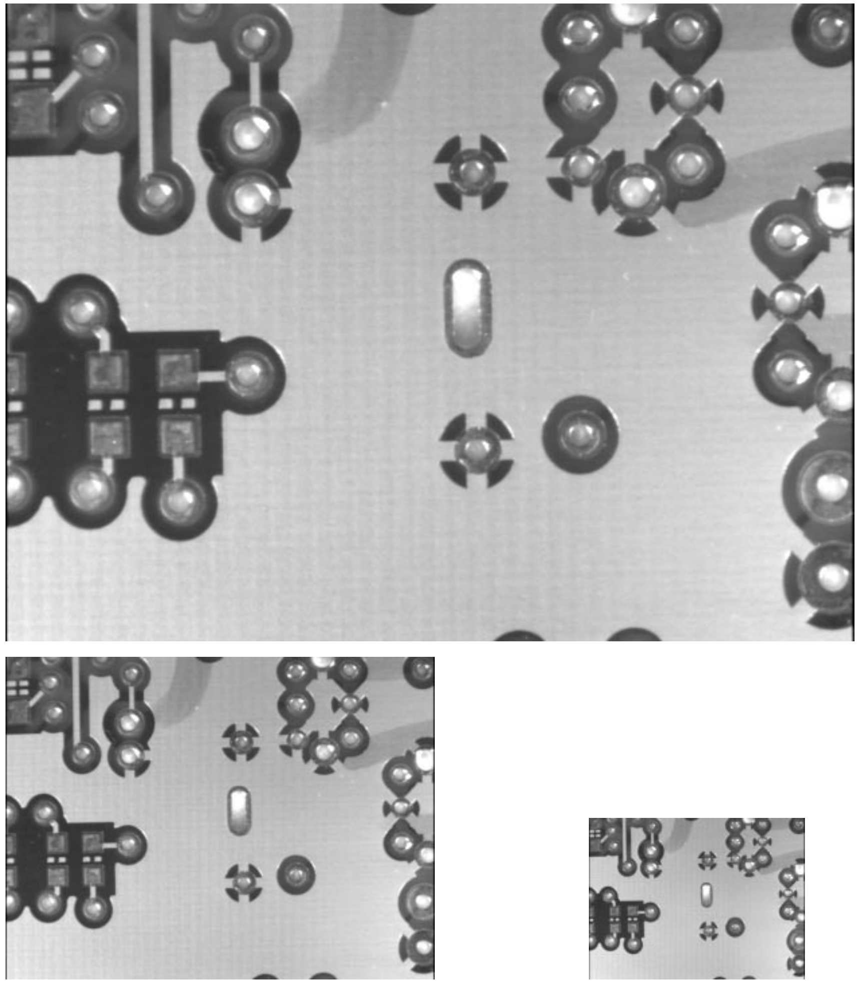
The Problem

- Need fast (~10–50 mS) search
- Arbitrary Patterns
- Non-specialized hardware
- Normalized greyscale correlation too slow (or is it?)
- Search for target $M(x, y)$ in image $I(x, y)$

The Approach

- **Manage complexity by using pyramid image representation: speed-up by factor of** $2^{4(k_{\max}-1)}$
  - **Do cheap search at coarse scale, refine at each level of the pyramid**
  - **Use estimates of correlation gradient to guide search at each level**
  - **Keep "good" matches at finest level**

Building the Pyramid



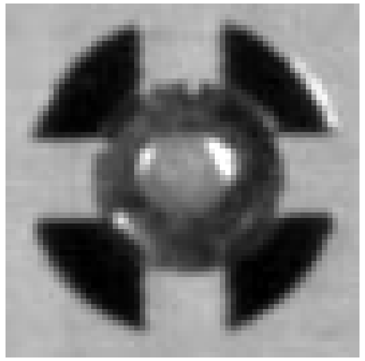
- Image dimensions decrease by factor of 2 at each level
- Each pixel in level *k* is the average of 4 pixels in level *k-1*
- Can be built just using add and shift
- $k_{\max} \le \log_2 \min\left(I_w, I_h\right)$

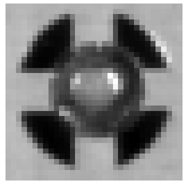
4

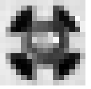
Tuning the Pyramid

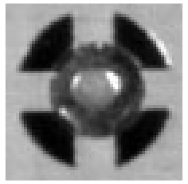
- **How many levels to use in pyramid?**
- **Can build pyramid "too deep" (1×1 at top? Pathological?)**
- **Use analysis of target to decide: "worst case" scenario**
  - **Construct "maximum height" pyramids for target and <u>shifted versions</u> of target**
  - **Use correlation between original and shifted versions to determine worst possible scores**
  - **Use scores to limit pyramid depth**

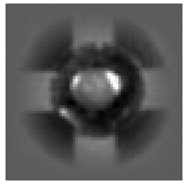
Target Representation

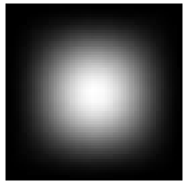

Original Target

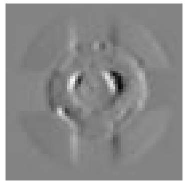

Target Pyramid Level 1

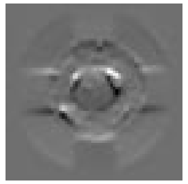

Target Pyramid Level 2


Model


Windowed Model


Window Function


X Component of Gradient


Y Component of Gradient

Correlation Gradient

- **At each level the location estimate is refined using steepest descent search (4 terms to compute)**

$$C(u,v)=\frac{\iint I(x,y)(MW)(x-u,y-v)\,dx\,dy}{\sqrt{\iint I^2(x,y)W(x-u,y-v)\,dx\,dy}}$$

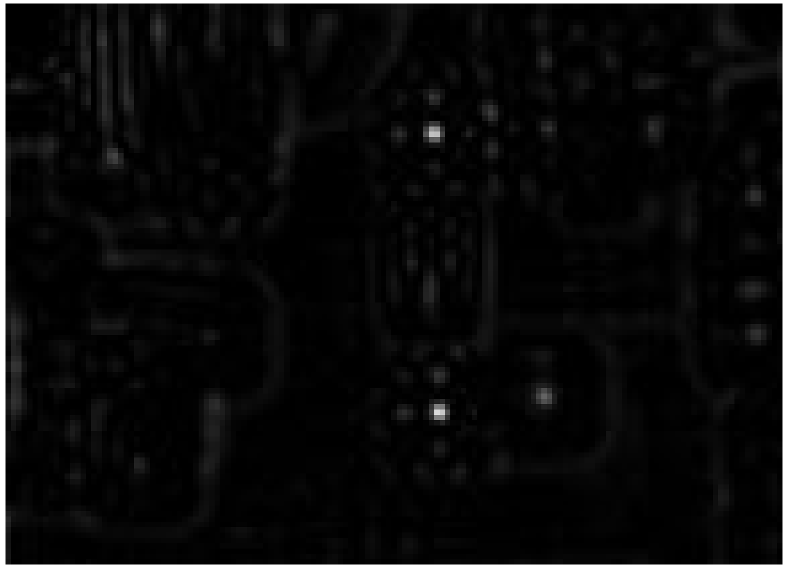
$$\nabla C^2(u,v)=-2\frac{\iint I(x,y)(MW)(x-u,y-v)\,dx\,dy}{\iint I^2(x,y)W(x-u,y-v)\,dx\,dy}\times\iint I(x,y)\nabla(MW)(x-u,y-v)\,dx\,dy$$

$$+\left[\frac{\iint I(x,y)(MW)(x-u,y-v)\,dx\,dy}{\iint I^2(x,y)W(x-u,y-v)\,dx\,dy}\right]^2\iint I(x,y)\nabla W(x-u,y-v)\,dx\,dy$$

The Search

- Do "inexpensive" correlation at top-level of pyramid
- Identify "candidates" from peaks in top-level correlation
    - For each candidate, descend through pyramid (coarse-to-fine), revising location estimate
    - Use correlation gradient estimate to minimize steps at each level
    - Reject candidates whose score drop below threshold

Top Level Correlation



Top Level Correlation Surface

- **Inexpensive due to reduced image size at top of pyramid**
- **May lead to false matches, so call peaks "candidates"**
- **Inhibit candidate peaks after use**

Subpixel Localization

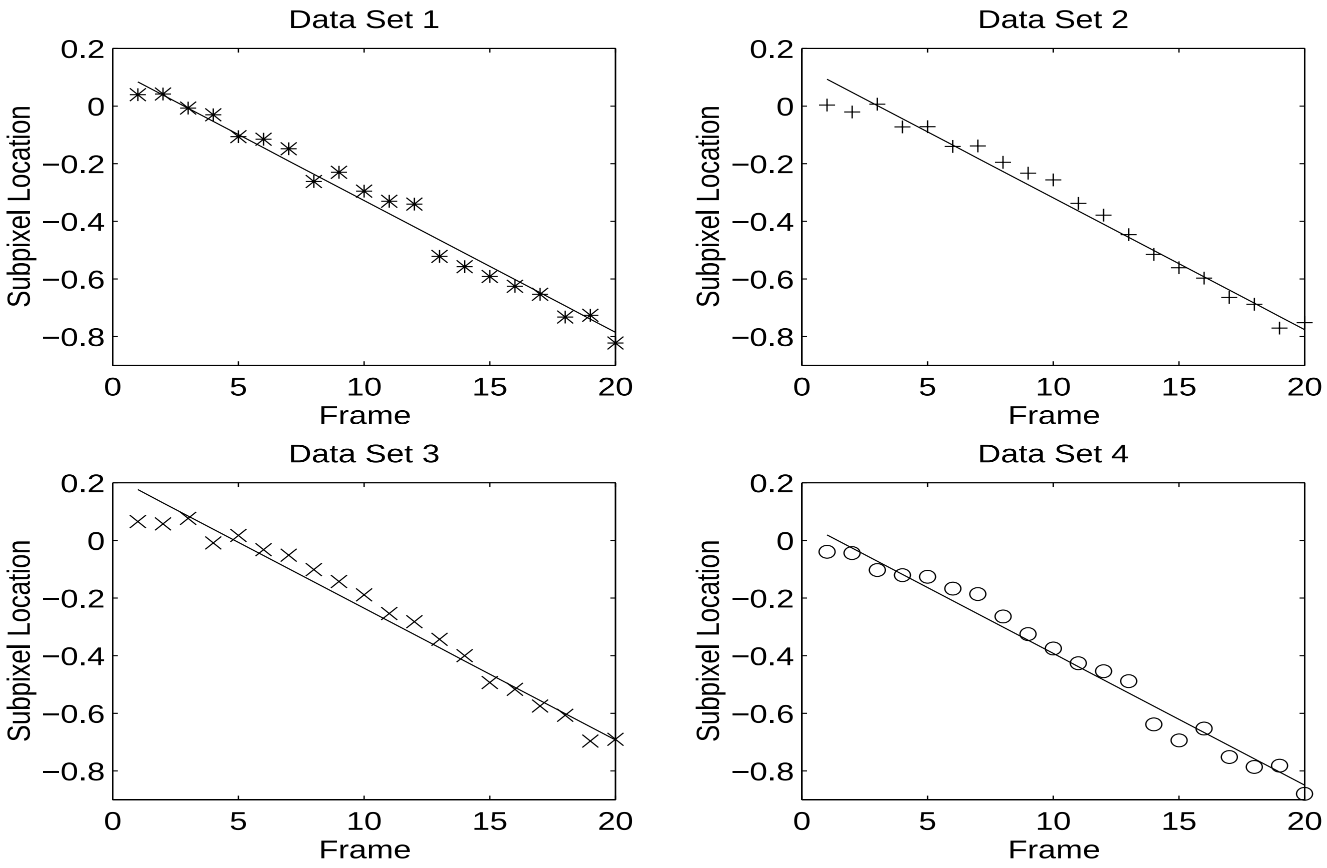
- We can locate a target with better than 1 pixel resolution
- Model correlation surface in 3x3 neighbourhood with a biquadratic surface: $\hat{C}(x,y) = \alpha x^2 + \beta y^2 + \gamma x + \delta y + \xi xy + \varphi$
- Fit model parameters via least squares (cheap/fast)
- Solve for maximum:

$$\frac{\partial \hat{C}}{\partial x} = 0 \Rightarrow 2\alpha x_c + \gamma + \xi y_c = 0 \qquad \frac{\partial \hat{C}}{\partial y} = 0 \Rightarrow 2\beta y_c + \delta + \xi x_c = 0$$

- Simple computation, much done offline in advance

10

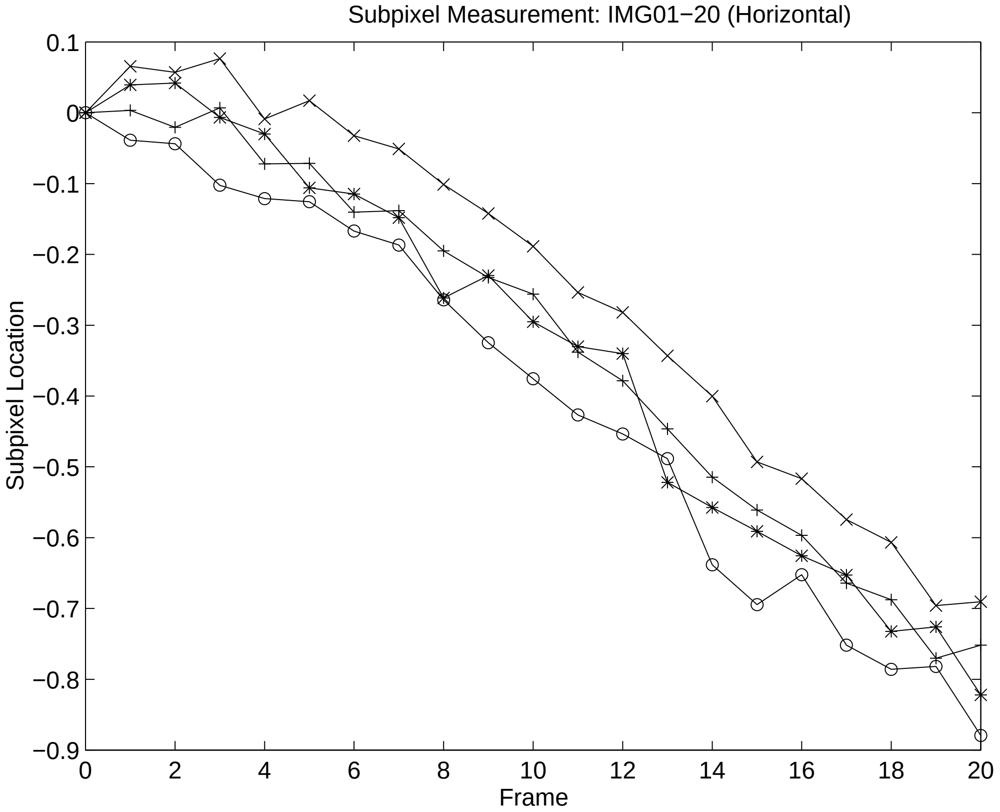
•Subpixel Results



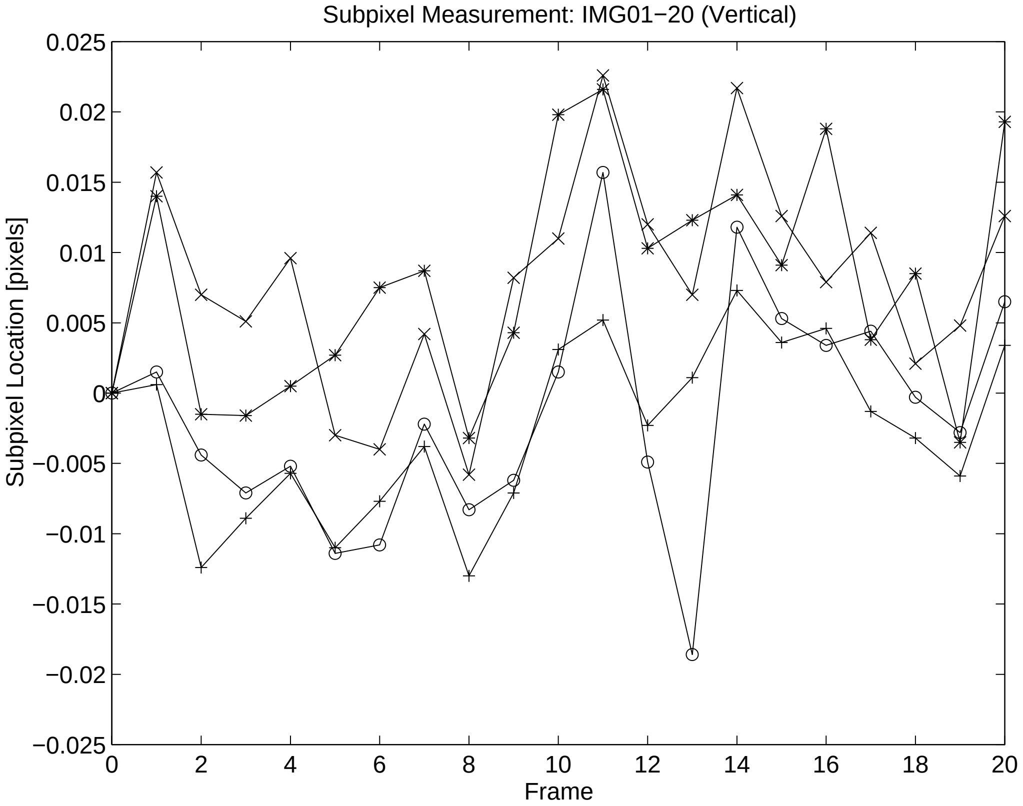
- **Search region moved in approximately 1/20th pixel increments horizontally with camera stationary**
- **Target detected with subpixel localization, results plotted**

11

Subpixel Results

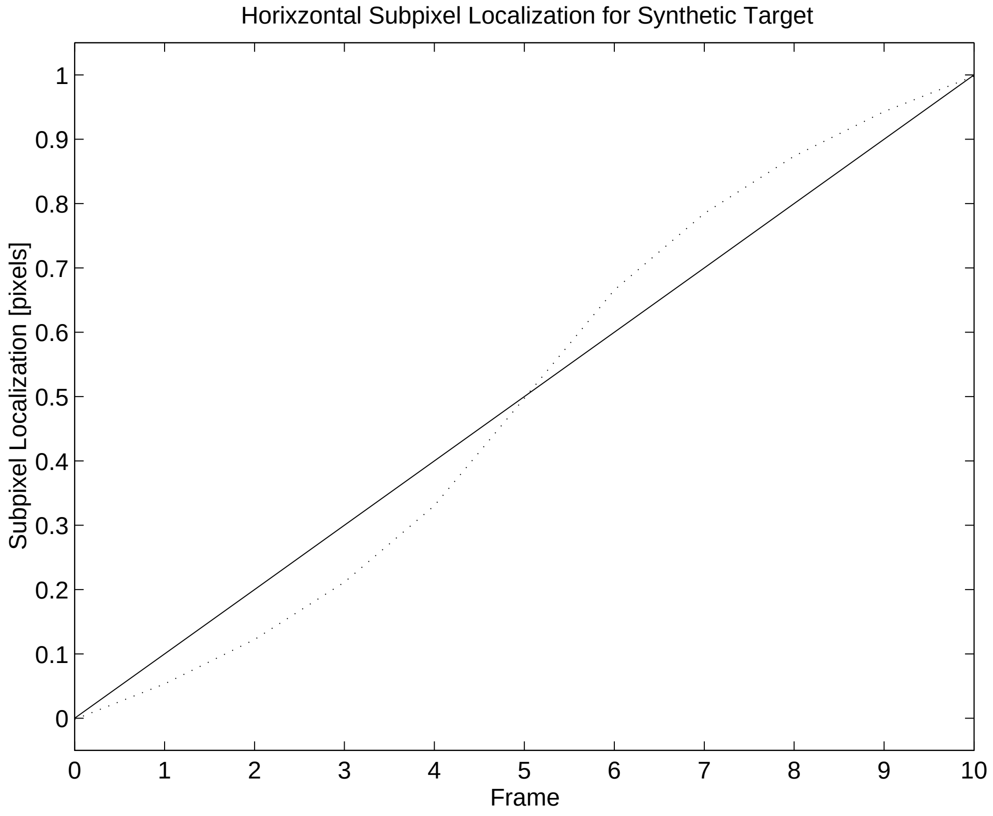


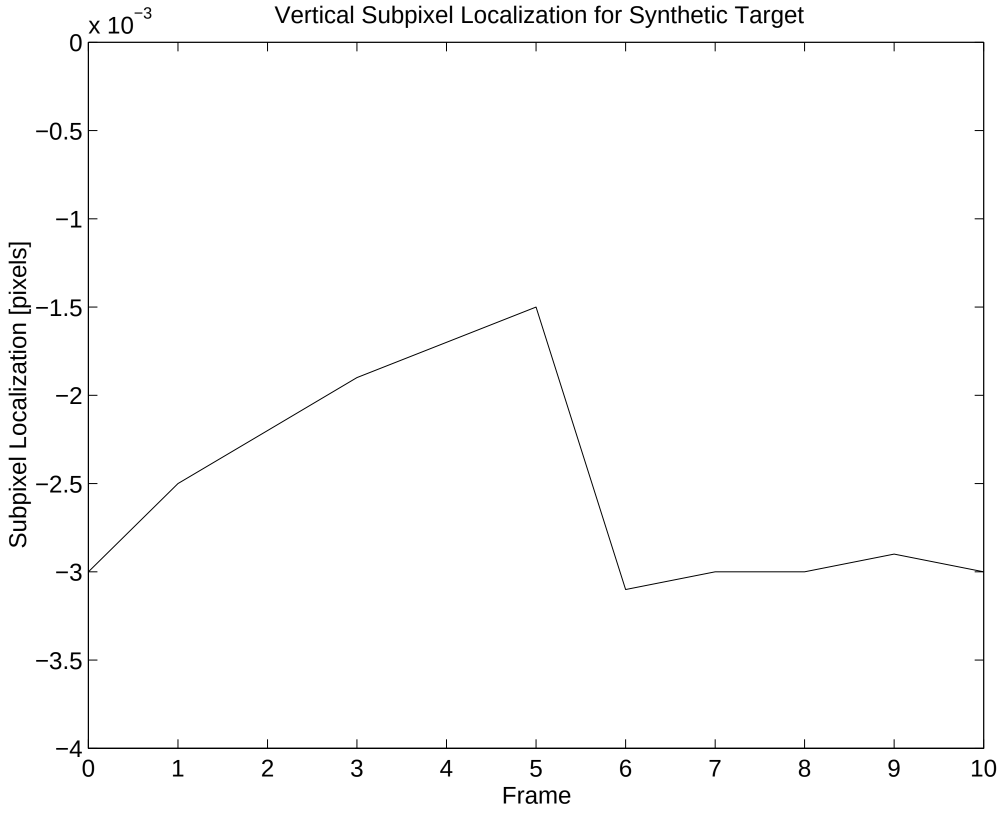
Composite of Horizontal Data



Composite of Vertical Data

Subpixel Results: Synthetic Data


Horixzontal Subpixel Localization for Synthetic Target


Vertical Subpixel Localization for Synthetic Target

Results

- 10 to 50 mS search times on 200 MHz PC running Windows NT, 64MB RAM
- Testing with accept threshold set to 0.95:
  - False positives typically less than 0.1% (varies with accept threshold)
  - False negative rate similar (rises if accept threshold set too high)
- Search time rises if many candidates
- Larger images have similar search times; pyramid just gets deeper

Applications

- **Machine Vision inspection & registration**
- **Searching for images on the Internet (insensitive to watermark removal)**
  - **Many images to search, so speed important!**
  - **Searching for "abused images" (defaced trademarks) on Internet**

Future Directions

- **Size invariance (pyramid representation well suited to this)**
- **Rotation invariance**
- **Occlusion**
- **Don't care pixels**

Conclusions

- **Novel use of pyramid representation in correlation image search**
- **Use of "worst case" analysis to decide depth of target pyramid representation**
- **Use of correlation gradient estimate to guide location refinement at each level of pyramid**
- **Implementation of simple but effective subpixel localization**