Smart Ubiquitous Projection: Discovering Surfaces for the Projection of Adaptive Content

F. Matulic[1], W. Büschel[1], M. Y. Yang[2], S. Ihrke[2], A. Ramraika[2], C. Rother[2], R. Dachselt[1]

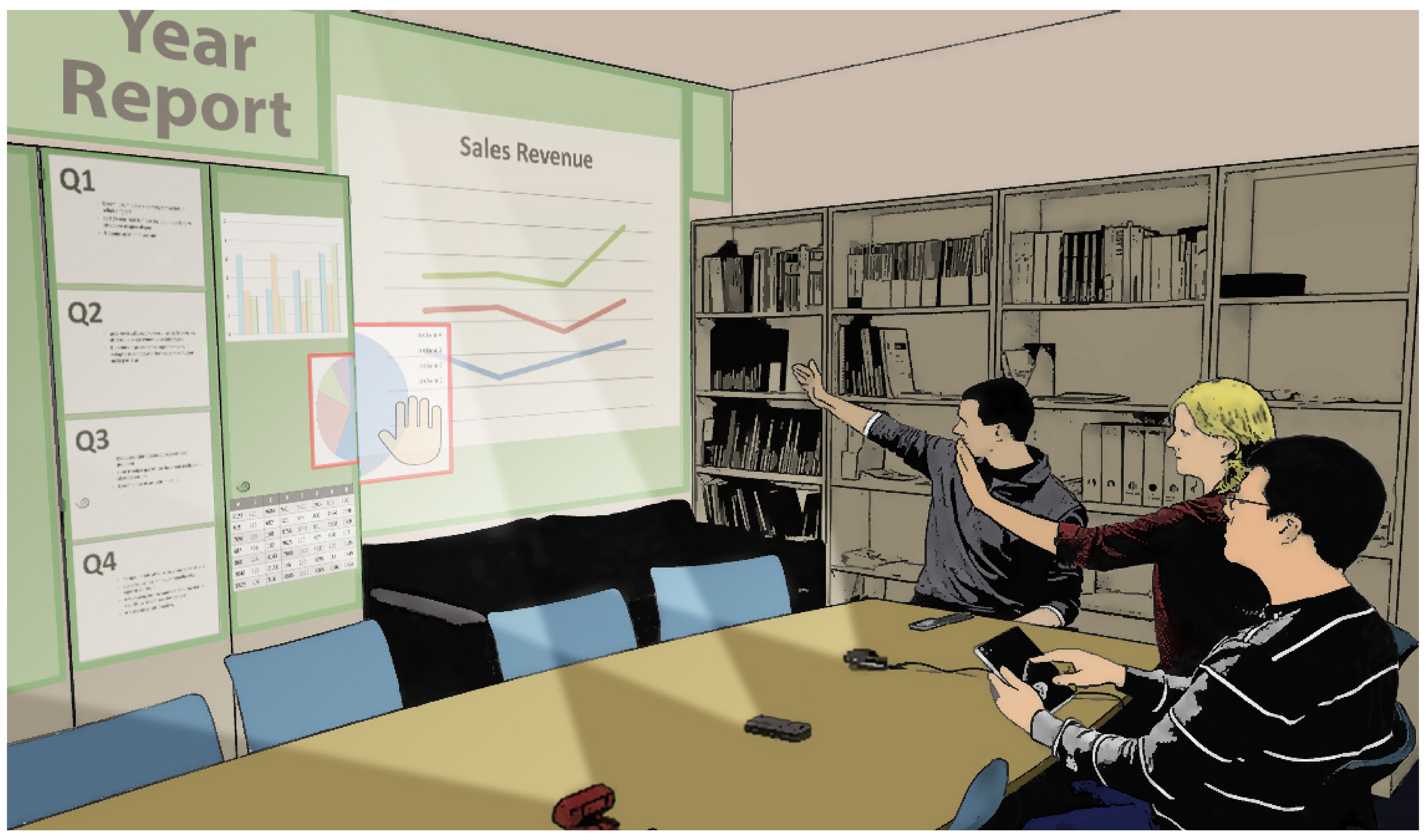
2016 #chi4good

Introduction

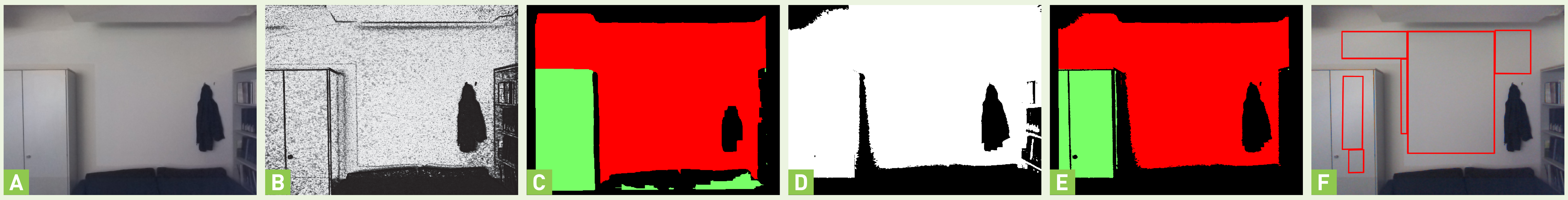
In this work, we seek to determine areas that are suitable for the projection and interaction with digital information. We describe a novel computer vision-based technique to automatically detect rectangular surface regions that are deemed adequate for projection and mark those areas as available placeholders for users to use as "clean" displays. As a proof of concept, we show how content can be adaptively laid out in those placeholders using a simple tablet UI.

Our concept:

- Sharing of digital content in a non-instrumented room
- Use mobile procam units to capture environment and project content
- Instead of projecting everywhere, detect areas suitable for projection
- Indicate those areas to users by means of projected frames
- Users can assign digital content, e.g. presentation elements to the frames.



Projection Surface Detection



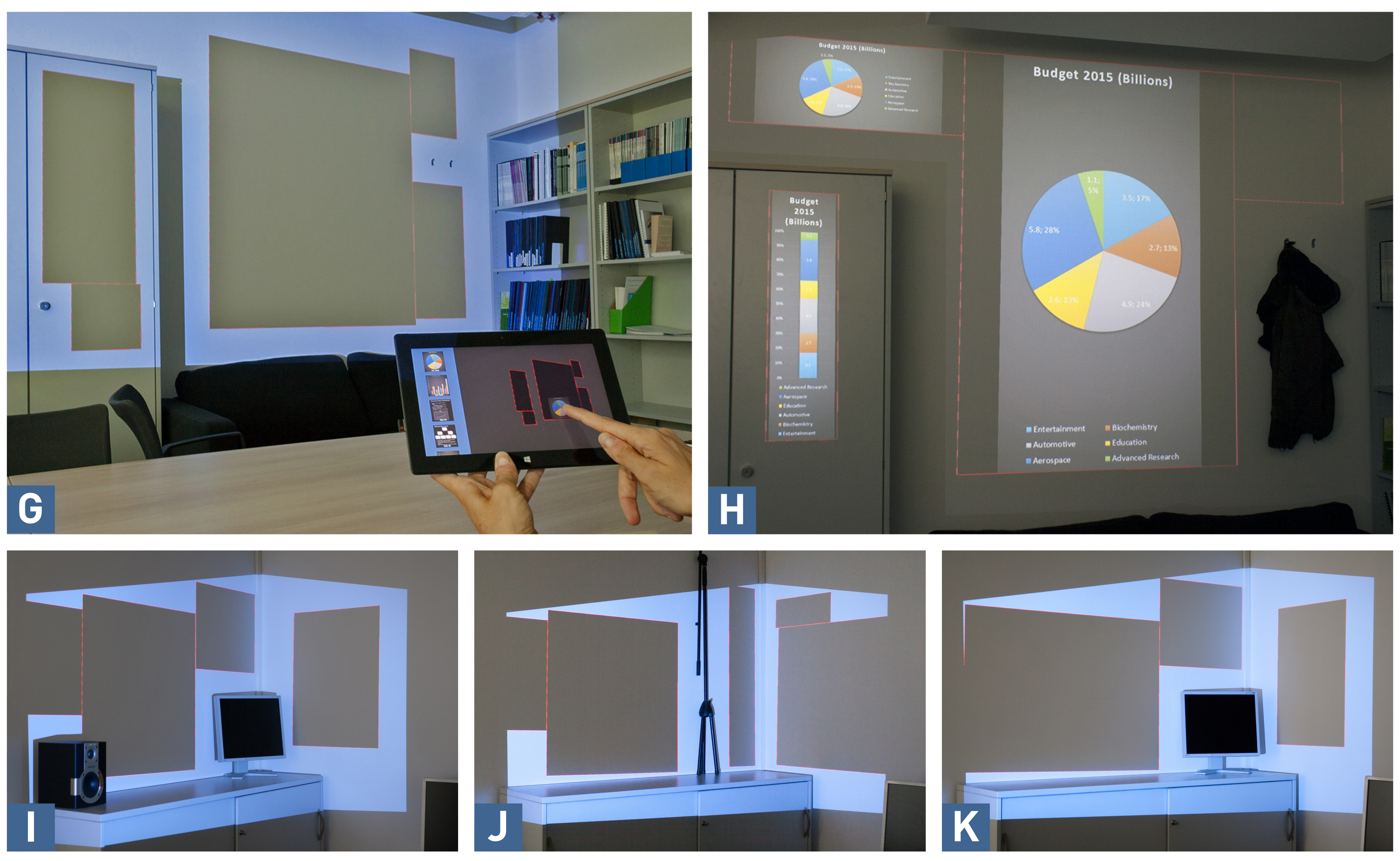
Random Forest Classification: Our algorithm works on RGB-D input (colour and depth) in form of a multistep pipeline. In the first step, a Randomised Decision Forest, trained with RGD-B data from multiple sample rooms, is used to classify pixels as projectionable. The pictures show **(A)** the RGB part of the input and **(B)** the generated probability map for the projectionability.

RANSAC Plane Detection: In a second step, a RANSAC (RANdom SAmple Consensus) algorithm is used to detect all planes in the 3D point cloud that have significant inliers. RANSAC is applied multiple times to find all sufficiently large planar surfaces **(C)**.

Combination of Intermediary Results: The individual masks for the planes detected by RANSAC **(C)** are joined with a binarised mask based on **(B)** as well as a projector mask **(D)** through a Boolean AND-operation. The projector mask provides information about the bounds of the projection area and occluding elements. It is computed by thresholding the difference image of a fully black and a fully white projection.

Rectangle Fitting: In the final step, for each mask representing a 3D plane, 2D rectangles are identified in the 3D plane. To this end, the contour of each mask **(E)** is transformed into a 3D polygon. The largest rectangles are then found using a rectangle-finding algorithm. The resulting rectangles, projected into the original picture, are shown in **(F)**.

Prototype



Implementation & Evaluation:

- The algorithm was implemented in Matlab and integrated in a C# program.
- We use one procam unit consisting of a Kinect v1 and a full HD projector, calibrated using a standard projector-camera calibration method.
- The Random Forest was trained on 100 images, the validation set consisted of 50 images.
- Pixel-wise classification achieved a training accuracy of 84% and a validation accuracy of 80%.

User Interface:

- We developed a tablet UI **(G)** with sample content to drag and drop into the identified placeholders.
- Content such as charts can be made adaptable to different rectangle sizes and aspect ratios **(H)**.
- Examples of detected rectangles in a test scene with differently arranged occluding objects can be seen in **(I, J, K)**.

www.cvlab-dresden.de
www.imld.de

TECHNISCHE UNIVERSITÄT DRESDEN

Technische Universität Dresden
[1] Interactive Media Lab Dresden [2] Computer Vision Lab Dresden
Contact: bueschel@acm.org

COMPUTER VISION LAB DRESDEN

INTERACTIVE MEDIA LAB DRESDEN