Deep Learning as Substitute for CRF Regularization in 3D Image Processing

Daniel Soukup, Svorad Štolc, Petra Thanner

Center for Vision, Automation & Control
AIT Austrian Institute of Technology GmbH
Giefinggasse 4, 1210 Vienna
daniel.soukup@ait.ac.at
svorad.stolc@ait.ac.at
petra.thanner@ait.ac.at

Abstract. For calculating 3D information with stereo matching, usually correspondence analysis yields a so-called depth hypotheses cost stack, which contains information about similarities of the visible structures at all positions of the analyzed stereo images. Often those cost values comprise a large amount of noise and/or ambiguities, so that regularization is required. The Conditional Random Field (CRF) regularizer from Shekhovtsov et al. [Sh16] is a very good algorithm among various methods. Due to the usual iterative nature of those regularizers, they often do not meet the strict speed and memory requirements posed in many real-world applications. In this paper, we propose to substitute Shekhovtsov's CRF algorithm with an especially designed U-shaped 3D Convolutional Neural Network (3D-CRF-CNN), which is taught proper regularization by the CRF algorithm as a teacher. Our experiments have shown, that such a 3D-CRF-CNN is not only able to mimic the CRF's regularizing behavior, but - if properly setup - also comprises remarkable generalization capabilities compared to a state-of-the-art 2D-CNN that is trained on a slightly different, yet equivalent, task. The advantages of such a CNN regularizer are its predictable computational performance and its relatively simple architectural structure, which allows for easy development, speed up, and deployment. We demonstrate the feasibility of the concept of training a 3D-CRF-CNN to take over CRF's regularizing functionality on the basis of available test data and show that it pays off to invest special effort into tailoring an according CNN architecture.

1 Introduction

In the process of calculating 3D information with stereo matching, correspondence analysis is used for analyzing the plausibility of a number of predefined depth hypotheses for each acquired object point. This step usually yields a data structure called the *Depth Hypotheses Cost Volume (DHCV)*. Therein, each value reflects the similarities of visible data structures at according image positions in the compared stereo images. An estimate of the object's underlying 3D surface structure, the so-called *depth map*, is derived from the DHCV. For each object

point, the depth is derived by selecting the depth hypothesis value which showed the most plausible similarities at according image locations in the different stereo images.

Due to insufficient texture in some parts of an acquired scene, the DHCV often is very noisy and/or does not yield unambiguous depth optima. For solving that problem by transferring depth information from more reliable object regions, various depth regularizers are used (e.g. SGM [Hi08], MGM [FDFM15], TRW-S [Ko06]), whereas most of them involve iterative algorithms for energy minimization with some priors. One of the best, which exceeds many other methods in terms of quality of results and computational performance, is from Shekhovtsov et al. [Sh16]. That algorithm (hereinafter referred to as *CRF regularizer*) is based on a conditional random field formulation (CRF) of the depth labeling problem and yields a very good approximation of the global solution.

As the CRF regularizer has shown superior results over various compared alternative methods and the fact that we had an actual highly optimized implementation of it at our disposal, we integrated it into our 3D image processing pipeline. While we are striving to satisfy real-world processing time requirements, it turned out that the CRF regularizer, while providing very satisfying results, is not fast enough. Moreover, due its considerable algorithmic as well as implementation complexity, we found it difficult and infeasible to achieve further optimization. Therefore, we decided trying to take a "shortcut" via deep learning and training a Convolutional Neural Network (CNN) on regularizing DHCVs. The CRF regularized DHCVs were considered the ground-truth data, i.e. the CRF regularizer serving as teacher for that CNN.

Zbontar and LeCun [ZL16] presented a CNN system that learns initial guesses for the matching costs from rectified input image pairs. In FlowNet [Do15] and FlowNet2.0 [Il17], solutions for optical flow computation with deep networks were proposed. Other prior work focused on the prediction of depth maps directly from two- or multi-view image stacks with deep networks in an end-to-end manner (e.g. CRL [Pa17], DeMoN [Um17], Deep-MVS [Hu18]).

In respect to our method, Wang et al. [WS18] and GC-NET [Ke17] are of particular interest, as their architectures contain deep networks, which use DHCV as input for a CNN rather than initial stereo image stacks. Wang et al.'s *MVDepthNet* is a 2D-CNN (utilizing 2D convolutional kernels) and is trained on the task of directly predicting depth maps from DHCV, in a DHCV-to-end manner so to say. GC-NET contains a stage, where DHCVs are processed through U-shaped 3D-CNNs in order to predict depth maps. They use 3D convolutional kernels like we do, which seems to be a natural choice as the DHCVs are 3D data structures as well. However, they operate on DHVC that were provided by a previous 2D-CNN from an input image pair, whereas the 2D- and 3D-CNN stages were trained in an end-to-end manner. The actual DHCV regularization takes place implicitly in those methods, while our proposed procedure is purely concentrated on the DHCV regularization step. In order to have a means to compare our proposed *3D-CRF-CNN* architecture (utilizing 3D convolutional kernels)

in this work with similar state-of-the-art, we used MVDepthNet’s depth map predictions on available test object acquisitions as reference to compete with.

In contrast to the state-of-the-art, we merely focus on substituting the single step of DHCV regularization with deep learning. In doing so, we make use of the knowledge that is revealed by the traditional correspondence analysis from the stereo images’ texture patterns, which we value as worthwhile prior knowledge for a deep learning method. Moreover, the calculation of depth maps from regularized DHCVs in a traditional non-machine learning way works perfectly well in our pipeline. Those two steps that perform well, anyway, the CNN would have to learn itself if trained in an end-to-end manner. We consider that a greater risk, as then the entire pipeline would solely depend on available training data and the additional prior knowledge was unnecessarily and wastefully discarded. If we only substitute a single step in our pipeline, then a CNN only needs to come up with solutions for a simpler task. That makes learning and/or retraining the inherent coherences easier for a CNN, e.g. if training data are limited or the CNN should be possibly small.

In Sec. 2, we go into the details of DHCVs and the impact that respective DHCV regularization has on the final estimates of depth maps. The core idea of this work, the concept of substituting the CRF regularization algorithm with a deep neural network is presented in Sec. 3. Actual results of conducted feasibility experiments on the basis of two available test objects are discussed in Sec. 4. We conclude in Sec. 5 and discuss lessons learned.

2 Role of the Depth Hypotheses Cost Volume and its Regularization

We have developed an *Inline Computational Imaging (ICI)* system, which utilizes a multi-line scan camera for acquiring linear light-field (LF) stacks of objects being transported in front of the camera [St14]. A LF stack is comprised of multiple images of the same object region observed from various different angles. The goal is to recover the acquired object region’s underlying 3D surface structure, the so-called *depth map*, from that multi-view image stack.

In the processing pipeline for determining the object’s depth map (Fig. 1), the so-called depth hypotheses cost volume (DHCV) is calculated from the LF stack and evaluated to get an estimate of the 3D surface structure. Due to the parallax principle, each object point is depicted at a slightly different image position in each of the view images of the LF stack. The extent of that positional deviation is directly related to the object point’s distance from the camera, i.e. its depth in the scene. Consequently, if the exact image positions of an object point’s depictions were known in all of the images in the LF stack, the object point’s depth is known as well. So, for each object point, a number of predefined, plausible depth hypotheses are analyzed, each being related to a different positional disparity structure. Each such depth hypothesis is assigned a cost value by means of correspondence analysis, indicating how similar the texture patterns at according image locations mutually are. Such that, for each object point, a vec-

Fig. 1. ICI 3D image processing pipeline: an object is acquired in a line-scan process with a multi-line scan camera yielding a 3D light-field (LF) stack of images. By means of correspondence analysis, a depth hypotheses cost volume is calculated from the LF stack, from which the underlying depth map (3D surface structure) of the object can be derived.

tor of cost values according to the depth hypotheses is obtained, which is stored into the third dimension of respective DHCV at the according image location (Fig. 1). The object points' depths can be derived from that by evaluating the depth values comprising minimal costs.

However, without any further post-processing of the DHCV, resulting depth maps are usually noisy (cmp. results in Fig. 4 for COIN and CAN data in Fig. 3). On the one hand, this is due to the fact, that the correspondence analysis only gives a very local, rather point-wise estimate of the underlying texture similarities, which yields a certain level of average noise also in well textured regions (Fig. 4, COIN data). On the other hand, the correspondence analysis for deter-

Fig. 2. Extended ICI 3D image processing pipeline: a cost volume regularizer is required in order to consolidate the rather local, thus noisy depth hypotheses cost estimates over larger regions. Consequently, noise-free, more reliable depth maps can be obtained.

mining the DHCV relies on texture similarities, which cannot give reasonable cost values in the case of homogeneous, totally untextured regions, which can be seen in the case of CAN data (Fig. 4). The initial DHCV has to be post-processed in order to consolidate depth hypotheses costs over larger regions into each point, in order to reduce noise and include reliable estimates of neighboring regions into untextured regions, where the correspondence analysis grasps at nothing, i.e. the DHCV needs to be regularized (Fig. 2).

We had decided to use the *Conditional Random Field (CRF)* regularizer presented in [Sh16]. The algorithm is a discrete method based on a conditional random field formulation of the depth labeling problem and has shown to yield good approximations of the global solution. If it converges, then it is guaranteed to converge towards the global solution. Fig. 4 shows depth maps derived from CRF regularized DHCV. Obviously, the average noise has entirely been removed for both test objects. Moreover, it is remarkable how well the CRF was able to come up with absolutely plausible estimates of the 3D surface structure in the homogeneous regions of the CAN data, where no meaningful results could be obtained from the un-regularized DHCV. The CRF regularizer was able to consolidate reliable depth cost values into those regions. However, it also yields undesirable, unnaturally looking blocking artifacts which are due to its discrete nature.

Fig. 3. Test objects **COIN** and **CAN**. The yellow rectangles indicate the actually acquired object regions.

3 A Practical Problem with CRF and a Possible Solution with a "3D-CRF-CNN"

Shekovtsov's CRF regularizer does a remarkable job regularizing DHCVs. Moreover, we had its implementation that we could integrate into our 3D processing pipeline. However, that implementation's computational complexity was too large for our requirements. Unfortunately, the CRF regularizer is also very complex, algorithmically and implementation-wise, respectively. A lot of theoretic

Fig. 4. Depth maps derived from un-regularized DHCV (*1st and 3rd images*). Depth maps derived from CRF regularized DHCV (*2nd and 4th images*). Additionally to average noise in the entire depth maps in the case of un-regularized DHCV, the CAN data comprise absolutely unreliable depth estimates in some regions (bottom, above the middle, inner region of the clip) due to underlying totally homogeneous textures. The CRF regularizer has cured the average noise as well as the severe misestimations.

concepts are utilized conceptually and the implementation itself, while thoroughly elaborated, is rather bulky. Working into the details would require a lot of effort in order to achieve further optimization.

Consequently, we were forced to search for an alternative that could more easily be achieved: using *CRF as a teacher for a Convolutional Neural Network (CNN)* that should be trained with the goal to predict the CRF’s regularized DHCV as closely as possible. A CNN is comprised of very simple building blocks (i.e. convolutions, non-linearities), that can be easily developed and implemented. Moreover, the processing of a CNN can easily be parallelized in order to speed up processing significantly.

Our *3D-CRF-CNN* is setup to solve the regression task of transforming an un-regularized DHCV into a regularized version as similar as possible to the CRF’s result. Such a regression task is typically solved by a U-shaped network comprising a down-sampling path of convolutional layers and an according up-sampling path. As the DHCVs are 3D structures, naturally, we chose to use *3D convolutional kernels*. According to those coarse architectural structures, 3D-CRF-CNN is comparable to GC-NET [Ke17]. In order for the 3D convolutional kernels being able to integrate information over possibly large image regions, we also utilized *dilated filters* in the spatial domain of the down-sampling path, which consists of 3 convolutional layers. Their far-reaching view without increasing the number of network parameters is crucial for reproducing those aspirational bridging capabilities of CRF at handling homogeneously textured areas. The down-sampling path inherently is comprised of *strided convolutions*, which increases the filters’ fields of view even more. The up-sampling consists of 3 strided, transposed convolutional layers with the goal to restore the corresponding DHCV size, the 3D-CRF-CNN’s actual regularization prediction. The entire 3D-CRF-CNN architecture is specially designed for increasing each filter’s field of view.

For the training procedure, we used the mean squared error (MSE) as a loss function and the Adam optimizer to perform the gradient descent. Moreover, we included *dropout* (50%) in the layers of the down-sampling path, which simulates noisy cost estimates in the training phase, so that the 3D-CRF-CNN has to explicitly learn the handling of problematic cost profiles.

4 Feasibility Experiments

We performed following two initial experiments, each with both of the two data sets COIN and CAN (Fig. 3), respectively, with the goal to determine the feasibility of the concept in general. The test objects are a coin and the top-view onto a can containing some fish snack. Both are metal surfaces with 3D structure comprising texture to different extents. They were acquired with our ICI camera setup and according DHCVs calculated from the respective LF stacks. Those were used as inputs for the 3D-CRF-CNN as well as for CRF, whose regularized versions were used as ground-truth data for the CNNs to learn.

As a reference to the state-of-the-art, we implemented MVDepthNet [WS18] which directly estimates depth maps from DHCV. While GC-NET [Ke17] also contains a processing section with a 3D-CNN being used to estimate depth maps from DHCV, they rely on DHCV that were provided by a previous CNN from two views only, whereas both CNNs were trained in an end-to-end manner. MVDepthNet and GC-NET do not focus on the DHCV regularization explicitly like our 3D-CRF-CNN, however, they have to perform some DHCV regularization implicitly in order to come up with a de-noised depth map. The chosen reference algorithm MVDepthNet is trained with the goal to reproduce a depth map that has been derived from a CRF regularized DHCV.

The two feasibility experiments' training procedures were performed on randomly sampled 256×256 training data patches from predefined training areas:

- *Experiment 1 "Basic Feasibility"*: training data set consists of patches from the upper region of the COIN data as well as from a small area in the CAN data, which is positioned over one of its problematic homogeneously textured regions (Fig. 5). The goal is to provide the CNN with both non-problematic as well as problematic data cases and see what will be the baseline performance given sufficient data.
- *Experiment 2 "Generalization capabilities"*: training data set only contains training data from the COIN stack, so that the CNNs have to be able to generalize for handling the difficult CAN data correctly, because they do not actually "see" them in their training phases (Fig. 6). The goal is to analyze the generalization performance.

The fully trained CNNs were subsequently applied to the entire DHCV stacks of both data sets in a slided window manner, whereas those regions that have not been used in the training process served as test data. They reveal the CNNs' actual capabilities to generalize their learned knowledge to yet unseen data, which is a crucial property of any machine learning method.

Fig. 5. Training data regions used for training the CNNs in Experiment 1 "Basic Feasibility": randomly sampled patches from the upper part of the COIN and some patches sampled from a problematic region of the CAN data.

Fig. 6. Training data regions used for training the CNNs in Experiment 2 "Generalization capabilities": ONLY randomly sampled patches from the upper part of the COIN. NO samples from the CAN data are used for training at all!

4.1 Results for Experiment 1 "Basic Feasibility"

The quality of the CNNs' results is measured by the root mean squared error (RMSE) of the depth map that can be obtained from the application of according CNN w.r.t. to the depth map that was obtained by using the CRF regularizer. [Fig. 7](#) shows those depth maps together with the respective RMSE values. Structurally, it is perceivable that both CNNs, the state-of-the-art MVDepthNet as well as our 3D-CRF-CNN are capable of getting rid of the average noise in both data sets and with both it is possible to come to quite reasonable estimates of the underlying 3D surface structures in the mentioned difficult areas of the CAN data set. While MVDepthNet over-estimates the depth of the pit too much, the 3D-CRF-CNN generates some small bumps at the edge of the pit ([Fig. 8](#)). However, in terms of RMSE, both CNNs do similarly well on both data sets, when they have been shown samples from both data sets in the training procedures. Moreover, both CNNs do not cause those unnaturally looking blocking artifacts as the CRF method does ([Fig. 9](#)).

Fig. 7. Depth maps obtained from the CNNs trained on COIN and CAN data (Experiment 1). *1st and 3rd images*: results with state-of-the-art MVDepthNet. *2nd and 3rd images*: results with our 3D-CRF-CNN with specially tailored architecture. Both perform similarly in terms of visual appearance and RMSE.

Fig. 8. Zoom to a special structure in depth maps at the edge of the pit in the CAN data set. *Left*: depth map derived from un-regularized DHCV. *Middle Left*: depth map derive from CRF regularized DHCV. *Middle Right*: depth map from state-of-the-art MVDepthNet. Note the erroneously enhanced pit depth. *Right*: depth map derived from the 3D-CRF-CNN regularized DHCV. Note the small bumps at the edge of the pit.

Fig. 9. Blocking artifacts in the depth maps derived from CRF regularized DHCVs, which are due to the CRF regularizer’s discrete nature (*left*). Smoother depth maps without unnaturally looking blocking artifact obtained from CNN processing (*right*).

4.2 Results Experiment 2 ”Generalization capabilities”

The results of Experiment 2, where the two networks were trained on parts of the COIN data only without presenting any samples of the problematic CAN data during training, are presented in Fig. 10. This experiment really reveals in how far the two networks are capable of generalizing: while the 3D-CRF-

Fig. 10. Depth maps obtained from the CNNs trained solely on COIN, while no CAN data were used in training (Experiment 2). *1st and 3rd images:* results with MVDepthNet. Note, that due to overfitting, it totally fails on the CAN data, which were not present in the training phase. *2nd and 3rd images:* results for 3D-CRF-CNN with specially tailored architecture. It shows a much better generalization capability, as it still can handle the CAN data robustly, despite it hasn't explicitly seen the respective noisy, unreliable CAN DHCVs during training.

CNN performs similarly well on both data sets, the MVDepthNet totally fails to yield reasonable estimates of the CAN's underlying surface structure. It seems to have totally over-fitted to the COIN data it was solely trained on and cannot handle the CAN data with its comprised difficult homogeneously textured regions correctly. Our 3D-CRF-CNN, whose architecture has carefully been tailored specially to the problem of integrating reasonable cost profile estimates over larger regions, is perfectly able to generalize to those CAN data. The fact that MVDepthNet contains much more network parameters (< 6 GB mem. requirement in inference phase) than 3D-CRF-CNN (< 0.9 GB mem. requirement in inference phase) makes the it much more prone to overfitting. Again, both CNNs do not cause those unnaturally looking blocking artifacts like the CRF regularizer (Fig. 9).

5 Conclusion

We have tackled a practical computational performance problem in our 3D processing pipeline by means of deep learning. The processing pipeline comprises a step of regularizing a data structure called the depth hypotheses cost volume in order to eliminate noise and consolidate measurements over larger spatial areas into each object point. An available implementation of a state-of-the-art cost volume regularizer, the discrete CRF regularizer, solved the problem satisfactorily, but it's available implementation turned out to be computationally infeasible for our practical applications.

Due to a high algorithmic complexity of the existing CRF implementation, we decided to try a "shortcut" by utilizing a 3D-CNN, for which the CRF regularizer served as a teacher. That 3D-CRF-CNN was trained to reproduce the regularization result of the CRF regularizer. As a CNN only consists of rather

simple building blocks, it is relatively easy to develop, deploy, and speed up, e.g. using parallelization techniques. We presented the respective obtained depth maps (3D surface structures) of a first feasibility study, in which we have tailored a special 3D-CRF-CNN architecture to the given task.

We compared those results with the CRF regularizer’s results, and with depth map results obtained from a state-of-the-art 2D-CNN. That reference method also operated with un-regularized cost volumes as input, but directly predicted depth maps, rather than only regularizing a cost volume from which the depth map is derived by traditional methods. Our results on the basis of a limited available data set of two objects showed, that the concept of substituting the CRF regularizer in the 3D processing pipeline is feasible. Both CNNs, the reference network and our 3D-CRF-CNN, were able to reduce noise and regularize certain object regions with unreliable underlying cost profiles. Moreover, both CNNs’ depth map results did not consist unnaturally looking blocking artifacts, which are inherent in the CRF regularizer’s depth maps. However, the CRF method is still slightly more robust in transferring reliable neighboring information into large very noisy areas. Our 3D-CRF-CNN outperformed the state-of-the-art 2D-CNN significantly in terms of generalization, as it was able to robustly handle certain extremely noisy object areas being present in one of the test objects, even if such regions were not utilized in the training phase. In this case, the state-of-the-art 2D-CNN totally failed.

In general, carefully collecting a complete training data set for deep learning tasks is a basic requirement. However, in actual real-world applications this might not always be possible. One might even not be aware that some essential object structures are missing in the training data. In this respect, our results demonstrate that it pays off to invest considerable effort into carefully designing the CNN architecture especially to the task at hand. This is safer in respect of reliable online performance, than simply taking a vanilla CNN from the internet, train it on the bunch of data at hand, and hope for the best.

References

- [Do15] Dosovitskiy, A.; Fischer, P.; Ilg, E.; Häusser, P.; Hazirbas, C.; Golkov, V.; v. d. Smagt, P.; Cremers, D.; Brox, T.: FlowNet: Learning Optical Flow with Convolutional Networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 2758–2766, Dec 2015.
- [FDFM15] Facciolo, Gabriele; De Franchis, Carlo; Meinhardt, Enric: MGM: A Significantly More Global Matching for Stereovision. In: BMVC. pp. 90–1, 2015.
- [Hi08] Hirschmuller, Heiko: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on pattern analysis and machine intelligence, 30(2):328–341, 2008.
- [Hu18] Huang, P.; Matzen, K.; Kopf, J.; Ahuja, N.; Huang, J.: Deep-MVS: Learning Multi-View Stereopsis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2018.

- [Il17] Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T.: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
- [Ke17] Kendall, A.; Martirosyan, H.; Dasgupta, S.; Henry, P.: End-to-End Learning of Geometry and Context for Deep Stereo Regression. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 66–75, Oct 2017.
- [Ko06] Kolmogorov, Vladimir: Convergent tree-reweighted message passing for energy minimization. IEEE transactions on pattern analysis and machine intelligence, 28(10):1568–1583, 2006.
- [Pa17] Pang, Jiahao; Sun, Wenxiu; Ren, Jimmy SJ.; Yang, Chengxi; Yan, Qiong: Cascade Residual Learning: A Two-Stage Convolutional Neural Network for Stereo Matching. In: The IEEE International Conference on Computer Vision (ICCV) Workshops. Oct 2017.
- [Sh16] Shekhovtsov, A.; Reinbacher, C.; Graber, G.; Pock, T.: Solving Dense Image Matching in Real-Time using Discrete-Continuous Optimization. In: Proc. of 21st Computer Vision Winter Workshop (CVWW). p. 13, 2016.
- [Št14] Štolc, Svorad; Soukup, Daniel; Holländer, Branislav; Huber-Mörk, Reinhold: Depth and all-in-focus imaging by a multi-line-scan light-field camera. Journal of Electronic Imaging, 23(5):053020, 2014.
- [Um17] Ummenhofer, B.; Zhou, H.; Uhrig, J.; Mayer, N.; Ilg, E.; Dosovitskiy, A.; Brox, T.: DeMoN: Depth and Motion Network for Learning Monocular Stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
- [WS18] Wang, K.; Shen, S.: MVDepthNet: Real-time Multiview Depth Estimation Neural Network. ArXiv e-prints, July 2018.
- [ZL16] Zbontar, Jure; LeCun, Yann: Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches. Journal of Machine Learning Research, 17:65:1–65:32, 2016.

Open Access Dieses Kapitel wird unter der Creative Commons Namensnennung 4.0 International Lizenz (<http://creativecommons.org/licenses/by/4.0/deed.de>) veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Kapitel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.